Athletic Recovery Device (ARD)

Amanda Ashmen, Ryan Bosanko, & Herschel Norwitz

ECE-499: Capstone Design Project

Advisor: Professor Takashi Buma, Phd

March 21, 2021

Report Summary

The main goal of this project was to develop a sanitary electrical system that could provide consistent, high quality, cold therapy treatment to consecutive users to cut down on the reliance on single-use plastic ice packs. In order to achieve this, we first got in contact with Cheryl Rockwood, the head athletic trainer at Union College, to help us determine a set design requirements that would need to be met in order for such a device to be practical. We developed a list of design requirements including: reaching and maintaining temperatures between 40 and 50°F in under 5 minutes over a 180° to 360° area, consistently running 10 to 15 minute treatments over a 3 hour duration, being easy and quick to clean, have an easy to use interface, fit in a 2 cubic foot area, and cost under $1500. From this we researched existing products already on the market and brainstormed several design alternatives. In the end we decided to go with thermoelectric cooling modules, known as Peltier coolers, which would enable us to control the temperature electrically. We designed a system which was broken into two parts. The first was a sleeve made from a gel pack that contained the Peltier units to provide the cooling, copper bars to distribute the temperature, and thermistors to monitor the temperature. The second part of the device was a case that contained the drivers to control the peltiers, the circuit to read the thermistors, the Raspberry Pi as the control, with a touch screen attached, and the power supply to run everything. After constructing and testing the completed system, we found that the system satisfied the majority of its requirements. The system only fell short in that only one of the subsystems was able to reach a temperature of 45°F, while the other could only reach 50°F, and that it was not able to achieve coverage of 180° around a treatment area. This said, given more time these issues could have been addressed. Overall, the device that we designed could be a viable alternative to plastic ice packs to provide cold therapy treatment.

Acknowledgements

Professor Takashi Buma: Thank you for guiding us through this project and being our advisor. Your insight really helped us develop our ideas and stay on track.

Union College Athletic Training Room: In particular Cheryl Rockwood and Jill Crary, thank you for taking the time to answer all of our questions about treatment and product requirements for a recovery device.

Gene Davison: Thank you for your assistance in acquiring and ordering our parts and supplying use with needed components that had not arrived yet.

Mark Hooker: Thank you for corresponding with us on the specification of copper plates and for providing a plate for initial testing.

Robert Harlan: Thank you for assistance in the machine shop modifying our case.

Professor Dosiek: Thank you for assisting us in our understanding of our control system.

SRG Committee: Thank you for your generous funding for our ambitious and expensive project.

Table of Contents:

List of Figures

List of Tables

Introduction

Despite many injury prevention programs, acute injuries such as ankle sprains, muscle sprains, and broken bones are inevitable. Risk of injury, associated with competing in a sport at a high level, is a threat that all athletes fear and many come to face. Union College offers 24 varsity sports teams, and like many other colleges across the country, the recovery process can be intense and complicated for many acute injuries. To combat these injuries, cold therapy is frequently used in conjunction with the R.I.C.E. method: Rest, Ice, Compression, and Elevation. If an athlete sustains an injury, they will typically make an appointment to see the athletic trainer, complete their rehabilitation exercises, then sit on a training bench and ice the injured area for ten to twenty minutes.

In the Union College training room, plastic ice bags are used for cold therapy treatment. These recyclable bags are typically applied to the body and strapped on with additional wrap to secure them in place. As shown in the picture below, these bags are even used at the highest level (Fig. 1).

Figure 1: Single-use Ice Bags

After speaking with the head athletic trainer at Union College, we learned that these bags are used because they are easy to set up, inexpensive to use, and sanitary since each bag is only used once [1]. Unfortunately, this single-use characteristic means that hundreds of these bags are used per week, which is unsustainable. While there are reusable gel ice packs on the market, the surface temperature for these products can become too cold or not cold enough for ideal cold therapy treatment. Chemical based ice packs must also be refrozen between uses. Other products which are able to hold a temperature for an extended period of time, such as liquid-based cold therapy machines, do not provide temperature regulation that could be useful for personalization to the athlete or body part. While some products do have the ability to control temperature and apply compression, this comes at a steep cost and makes operation and sanitization cumbersome.

For this project, we maintained communication with Union College's athletic training room to design a device that addressed the drawbacks of the current cold therapy methods being utilized. This product aimed to be sanitary and environmentally-friendly. In order to address the needs of the training room, which deals with many athletes each day, this product needed to have a user-interface that did not require constant professional supervision and had the capability to treat consecutive athletes during the busiest times of the day. While this product might not replace all of the single-use ice bags that student-athletes get after practices, this was intended to be a more sustainable and reliable method of cold therapy. We proposed that an electronically powered ice pack with the ability to control the temperature and duration of the treatment for consecutive uses would be a valuable and sustainable asset to the Union College training room.

This report aims to provide a comprehensive and exhaustive account of the process by which used to attempt to solve this problem. We will first go into detail about the background work that served as inspiration for pursuing this project. This includes motivation for choosing this topic and researching other devices that have been made to address a similar need. From here, we will move on to constraints the device needs to fill, including needs specified by the head athletic trainer. This will be followed by discussions of various methods of addressing and applying our ideas to make a working system. After detailing alternative design ideas, we will go into detail about how we applied the design process to make important decisions about our preliminary design. We detail tests we ran, the results of these tests, and their implications for the final design. Then, we will explain the final design, along with testing results and the performance metrics of this device. Finally, we will end with the proposed production schedule, cost analysis, user manual, discussion and conclusion.

Background

Through various medical studies relating to the effectiveness of different systems for athletes, we saw a variety of different approaches to recovery devices. A popular method is to implement cold therapy, which cools the injury to decrease blood flow to an injured area. Athletes use cold therapy to enhance their performance by increasing the speed of recovery by reducing DOMS (delayed onset muscular soreness), which can occur following high-intensity training and competition [2]. After an athlete sustains an injury, it has been shown that the combination of compression and continuous cryotherapy is an effective method for healing by decreasing inflammation [3]. There are a combination of factors that influence the effectiveness of this process, such as: temperature of the applied surface, duration of treatment, degree of conformity to the surface of the skin, and to the shape of the body part. We will explain in detail the requirements we chose to model our design after in our design requirements section. To determine what goals were realistic and attainable, we did extensive research on products that already have been industrially developed and marketed to audiences seeking cold therapy treatment alternatives to recyclable ice bags.

Currently on the market, most products use one of two different methods of cooling for cold therapy products: chemical packs or systems that pump cold water to an insulated sleeve. Chemical packs are typically filled with a soft gel material and are put in the freezer for a few hours before being applied to the injured area. As previously stated, these devices can either be above or below the desired temperature, rendering them less effective. Additionally, they must be refrozen, so they are not useful in Union College's training room environment which must treat consecutive athletes. The same negative characteristics apply to the PowerPlay device (Fig. 2), which is a machine that holds a gel pack and allows the user to simultaneously provide compression to the injured area [4].

Figure 2: PowerPlay Device

Liquid circulation systems, on the other hand, can be used consecutively over a span of hours. Some of these products are very expensive, and others lack the ability to provide the user with specific temperature control. One example of a liquid circulation product is the GameReady system; this product works by circulating cold air through tubes connected to the compression sleeve [5]. Not only is this system messy and cumbersome to use, as stated by the Union College head athletic trainer, but it is also very expensive; these devices can cost from two to three thousand dollars. A more portable and less expensive device on the market is the Breg Polar

Care Kodiak Cold Therapy Units, shown in Figure 3 below, which does not provide the option to control the specific temperature [6]. Different individuals, different injuries, and different body parts all require different temperature profiles for optimal recovery. As such, there are inherent advantages to any system that can provide temperature regulation.

Figure 3: Breg Polar Care Kodiak Cold Therapy Unit

One device that was able to provide temperature control is the ThermalZone Thermal Therapy system, shown in Figure 4 [7]. This is a wall pluggable device that uses cold water in order to create cooling. It can regulate set temperatures between 34 and 125 degrees Fahrenheit. This device looked to be the closest to what we were thinking of what we wanted our device to be like because it is well sized, appears fairly easy to use, is reasonably priced and can provide good temperature control. Similar to the Breg Polar Care Kodiak Cold Therapy Unit, it used pumping cold water through a tube to an insulated sleeve as its method to apply cold treatment. While this unit has the ability to control its temperature, it comes at a higher cost than most cold therapy units, and users who purchased the device from Amazon.com complained that this device was too loud, leaked water, and did not maintain temperatures within the cold range settings.

Figure 4: ThermaZone Thermal Therapy System

Another common cooling method is refrigeration. The main goal of refrigeration is the removal of heat from the system and prevention of heat from entering a system. Common refrigerators prevent heat from entering by being made of insulating materials. In order to remove heat, refrigerators apply a similar method of cooling as the human body. As the body gets hot, it produces the liquid sweat. As sweat evaporates and turns into vapor, heat is removed and the body is cooled. In order to mimic this behavior, fridges pump in liquid chemicals that can very easily evaporate. As it flows throughout the system, it absorbs the heat, evaporates, and turns into gas. This gas is brought out of the system where the excess heat is siphoned off by discharge lines. As shown in Figure 5, the vapor is passed through a compressor where it is returned to its liquid form before it is pumped back in and the process repeats [8]. This is the most common method used today to induce cooling in a non-medical environment.

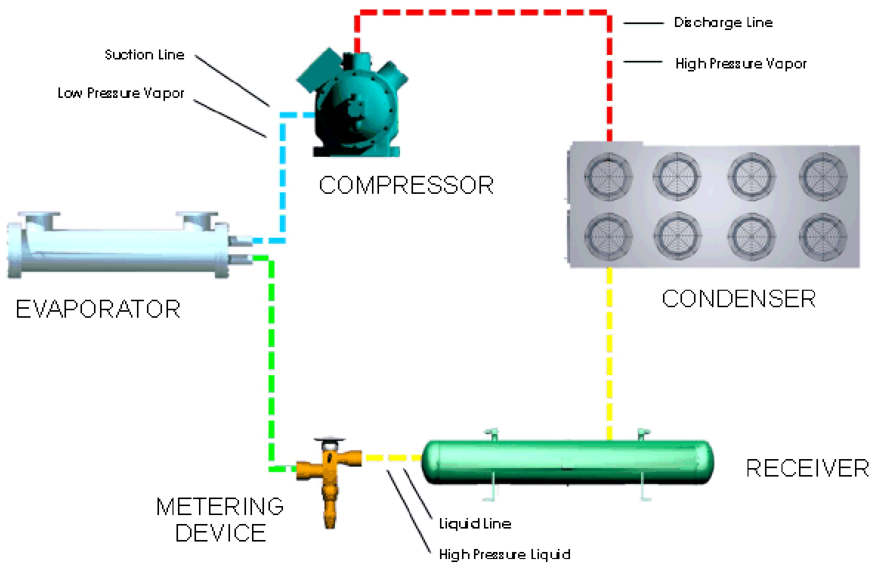


Figure 5: Basic Refrigeration System

An alternate form of cooling that is found in many systems is the use of thermoelectric modules, which often use the Peltier Effect, which tells us that under certain circumstances, an applied current can directly cause a temperature gradient [9][10]. As displayed in the diagram in Figure 6, this can only occur when the current is applied through the transition, or "junction", between two different conductive materials [9][10]. Each material can be given its own "Peltier coefficient", which represents how much thermal current a given electrical current carries in different metals [10]. Here, thermal current can be thought of as a sort of potential energy carried by electric currents, not in their magnetic fields, but caused by the electrons interacting with their material. When this coefficient varies at a junction, the energy entering the junction no longer is equal to the energy leaving the junction, so the excess energy is dissipated as heat, or absorbed in a cooling effect [10]. Whether absorption or dissipation occurs at the junction relies on both the direction of electric current and on which Peltier coefficient of the materials is larger.

Of course, the energy for this must come from somewhere, so the other junction for this operation will either dissipate the absorbed energy as heat, or absorb energy itself to power the junction's heat dissipation. This means that however much energy is being produced by the first junction, that much heat is absorbed by the other junction. A medical recovery device is a viable application for thermoelectric cooling, which needs to be regulated for long periods of time.

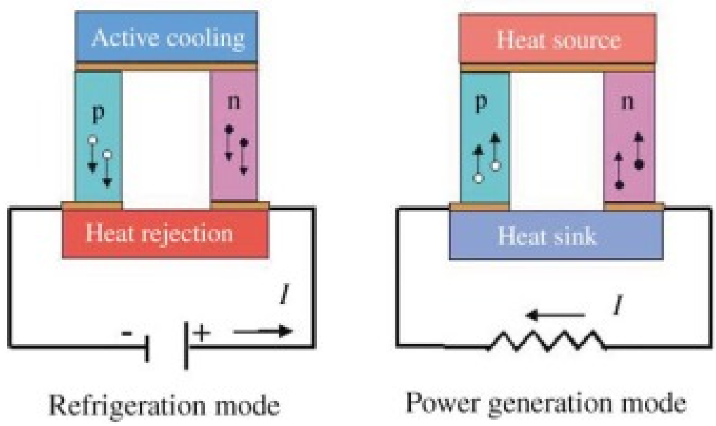


Figure 6: Peltier Thermoelectric Couple

In order to get a better understanding of how to drive Peltier units, we found previous works on Peltier-controlled heating/cooling systems. For example, we found a design project report that created a small Peltier controlled device using an atmega-1248 microcontroller and Peltier unit with a heatsink; the device had a digital graphical user interface powered using MATLAB [11]. We also found work done by an Iranian team who made a Peltier system based on warming what was essentially a bag of water. The team used a simple bag of water with a single Peltier and heat exchanger on top of the bag of water, which they called an "ice pack" [12]. An image of their system is shown below (Fig. 7).

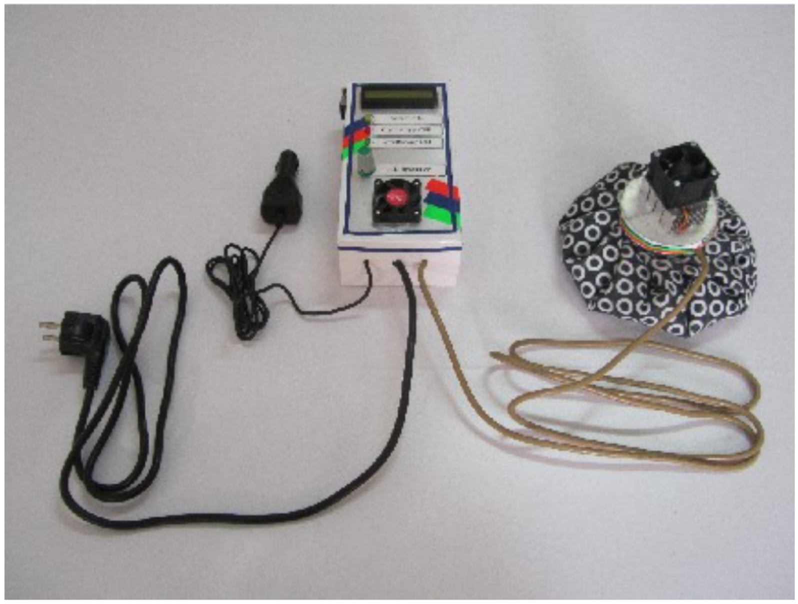


Figure 7: Peltier Cooling Device

The main components here are their control system in the center and their ice bag, shown to the right. The team was able to keep the water at 3-5ºC consistently and only took around 15 minutes to cool down the water to the desired temperature [12]. Because the Peltier was on top of the "ice pack", with the heat sink and fan included and kept relatively small, we took this as further evidence that it may be possible to keep the Peltiers cool using reasonably sized heat sinks while still delivering the required power.

This is not the only research that exists detailing the use of Peltier coolers for cryotherapy. A team from The University of Texas at Austin was also able to demonstrate the viability of Peltier units for more explicitly medical cooling applications [13]. The team constructed a device with two Peltier coolers and used it to perform cryotherapy on nine healthy subjects [13]. Their device in operation is shown below in Figure 8.

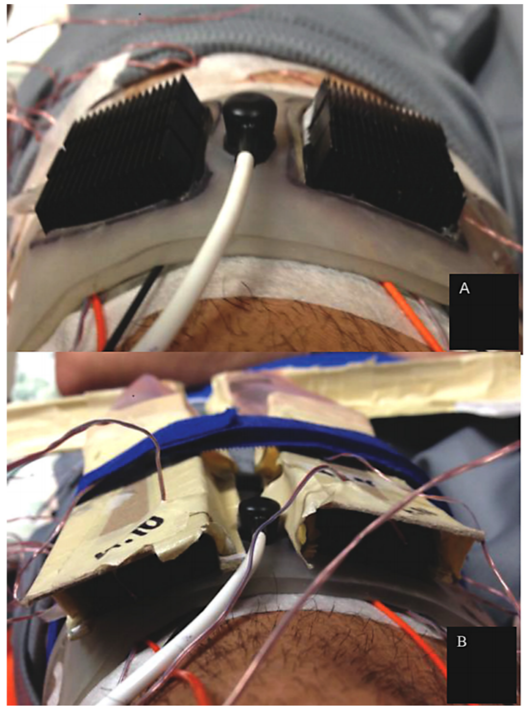


Figure 8: Operational Peltier System

In the upper photo (A), the two Peltiers with heat sinks are shown flanking a blood perfusion probe. The bottom photo (B) shows the system with air circulation to help the heat sinks. As is evident, this device was far from a commercial product and was intended for injuries in a medical context [13]. They were able to show, however, that the device was able to cool the skin by 6ºC and largely control blood perfusion around the affected area [13]. With such a small temperature change on the surface of the skin, they were able to reduce blood perfusion by 60% [13]. Their device was also able to provide heating, which they used to increase blood perfusion by 200%, with an upwards increase of 6ºC on the temperature of the skin[13]. A graph of their results is shown below in Figure 9, with blood perfusion in the upper graph and the skin temperature in the lower graph [13].

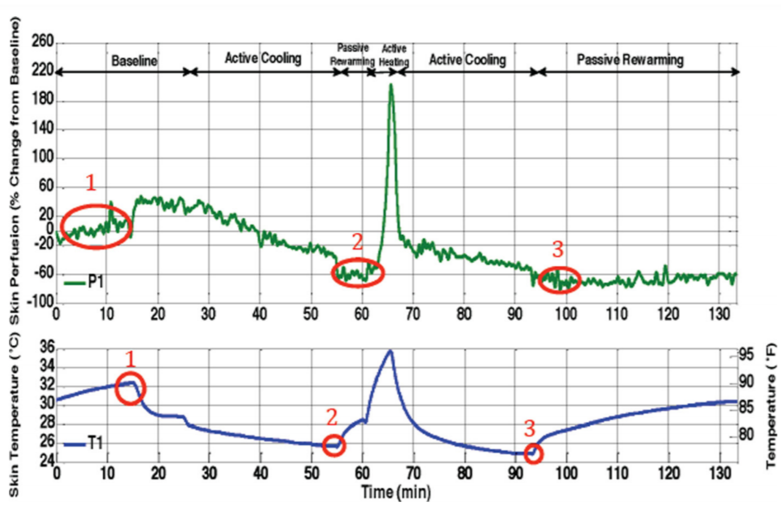


Figure 9: Blood Perfusion and Temperature

They ran their system for about two hours, mostly performing cooling but with a brief warming period in the middle [13]. While the need we were trying to address aimed to operate for a shorter treatment period (less than thirty minutes), it was good to have an understanding of what a longer time interval would look like. Overall, their system does a fantastic job of proving the viability of using Peltier coolers to help with recovery from injuries. Such a system can have very precise temperature regulation while staying relatively small and light.

Design Requirements

Before we could start thinking about different ways to approach the problem we aimed to address, we first had to gain a solid understanding of what the requirements of the device were. To begin, we needed to have in depth communication with the trainer to see what they were looking for in an athletic recovery device. They laid out several requirements relating to performance and safety. From these specifications, we developed several of our own more specific requirements for how we would operate the system, what safety features were needed to account for this, and what the overall cost and final appearance should be.

Performance:

The trainer specified that the device would need to maintain temperatures between 4 and 10 degrees Celsius at the point of contact between the gel pack and the skin [14]. Our team decided that to best achieve this, our system should maintain a measured temperature to within 2ºC of the desired temperature at the point of contact on the surface of the skin. While they did not specify exactly how fast it needed to reach those temperatures during the initial start-up period, they stated that faster is better and anything more than 5 minutes would likely be impractical for use in the busy training room environment. As for overall duration, the training room is typically open for treatment during the week during the hours of 9-12pm and 1-4pm. That is, for two different three hour periods during the day, athletes can come in for their appointments with athletic trainers to do rehabilitory or recovery activities. After communicating with the athletic trainers to get a rough estimate of how many athletes they treat each day, we were told that an average of 40 athletes use an ice bag at a training bench each day during the fall term, and about 20 athletes each day for each of the winter and spring terms [1]. This defines the requirement that the athletic recovery device be able to operate for a three hour period with only a few short, approximately 5-10 minute, breaks in between athletes. During this break, the device will have to be cleaned and sanitized.

Physical Parameters:

We then decided to determine the shape, size, and form that would be ideal for the training room environment. The head trainer of Union College told us that smaller is better for these devices. Though no specified dimensions were given, we aim to have the device fit within at most a 2 cubic foot volume to let it sit comfortably underneath the training bench. In order for this device to be practical, it needed to cover an area greater than 180 degrees around the area being treated, ideally wrapping around the body part at or near 360 degrees. For instance, for knee injuries, the device would have to actively cool an area stretching at least halfway around the knee. Furthermore, the cooling applied on the injured area needed to be fairly uniform throughout. This further implies that the device itself should be malleable and flexible enough to conform to the surface of the skin and to the appendage being treated.

Interface:

As for the process of using the device, it would ideally not require excessive supervision or direction to use, given that qualified athletic trainers can sometimes be heavily outnumbered by injured athletes requiring treatment. Since treatment temperatures vary per athlete, injury, and body part, there is no "one size fits all" treatment temperature/duration combination. The training room communicated that a few preset modes, such as 5 ºC for 15 minutes or 8 ºC for 20 minutes, would be convenient. To address unique treatment cases, a custom mode setting to adjust temperature or duration, which would be password locked, would be a useful function. Ideally, after multiple uses, an athlete could be given basic instruction to inform them of the treatment specifications they needed, and they would be able to input their setting for their treatment and subsequent appointments, without the need of supervision from the trainer. If the device added more complications for the trainer who needs to set-up the device for each student, then it would no longer be practical. We discussed the needs regarding the interface for the device. Given that cold therapy is a routine and low-risk procedure, the trainer specified that there would not be a need for a database or data collection system to store the treatment periods of the device per athlete [1]. Additionally, they stated it would be most convenient to have the controls of the system be directly operable from the device itself, rather than being on an external computer or phone application elsewhere[1].

Safety:

Another important area that the trainer provided specifications for was safety. One key reason that ice bags are used is because they are one-use, and thus do not need to be cleaned, nor do they pose a no risk of transmitting diseases. The trainer made it clear that whatever replaces ice packs must be very easy to clean, to ensure that the device is sanitary between uses [1]. This means that the material must be made of something that is not damaged by cleaning fluids. Furthermore, the device must have safety features to prevent any harm to the user that may come from some sharp or moving component, or electrical system.

Power:

In order to get an idea of how much power our system would need in order to function, we performed some basic calculations. Regardless of the method we eventually choose to deliver this power, some source of power will still need to be provided. We first need to know the area that our system will need to cool. In order to be able to provide cooling to injuries of moderate size, we decided our device needed to be able to cool an area of at least 400 cm^2. For context, a typical circumference of the human leg 7cm above the knee is 26cm [15]. Assuming 360 degree cooling, that would mean our device would be capable of cooling about 15cm of the leg with an average circumference of 26cm.

With an area, we can now calculate how much cooling power our system must provide. We know from previous work that the thermal conductivity of various layers near the surface of human skin are shown Table 1 below:

	Thickness, mm	Thermal conductivity, W/(m K)
Epidermis	0.1	0.235
Papillary dermis	0.55	0.445
Reticular dermis	0.55	0.445
Fat	2.0	0.185
Muscle	6.8	0.51

Table 1: Properties of Skin Surface[4]

If we assume this small region can be approximated by a rectangular prism, with our device on top providing cooling power at 10ºC, and the body below at 37ºC, we know the temperature difference across the region and we know the area of the region. We calculate the thermal resistivity as follows:

Loading…

Loading…+Loading…

Loading…

Where R_T is the thermal resistance, x_i is the thickness of the layer in meters, λ_i is the thermal conductivity of each layer in watts per meter kelvin, and A is our area in square meters. We then can divide the temperature gradient by this thermal resistance to get the minimum cooling power required to power our device:

Loading…

Here, P is the power in Watts ΔT is the temperature difference in kelvin, and R_{Th} is the thermal resistance in degrees Kelvin per Watt.

Economic:

In order to ensure that our device meets the specifications set out by the trainer and to ensure the product we create is viable, our team came up with additional design requirements. Firstly, we decided on a cost to aim for. During the 2018-2019 academic year at Union College, the training room recorded that at least 3088 ice bags were used for treatment. Note that this does not account for instances where an athlete used two ice bags in a single treatment session, nor does it include ice bags used on-the-go after practices. Cramer ice bag rolls, as used in the training room, are sold in rolls of 750 for approximately one hundred dollars. If the training room purchases five of these 750 count rolls a year, to make this a financially valuable investment within three years, the product would need to cost less than $1500. We have kept in mind that the initial prototype will be more expensive than a design ready for manufacture. We also wanted to emphasise the importance of our device being self contained. A lot of products currently on the market require ice to be provided in order to function properly. In order to eliminate this burden from the training room, our device will not require other devices like ice machines to function.

Requirements Summary:

In order to make our product fit the needs that the training room requires, we set out to combine the functionality of athletic recovery devices currently on the market, while reducing the price by eliminating ability to perform the compression. We planned to focus on creating a product that can last for hours during the training rooms 3-hour treatment periods. It should have the ability to be set to a certain temperature or duration based on the specific athlete or body part.

Additionally, it needs to be sanitary and easy to use so it does not compromise the benefits of the current method. Communication with the customer confirmed the need for this type of device; a preference for a touch screen display over buttons or a phone app was indicated.

Design Alternatives

In order to make our product fit the needs that the training room required, we had to combine the functionality of athletic recovery devices that were on the market, while keeping the price low. We focused on creating a product that could last for hours during the training room's 3-hour treatment periods. It had to have the ability to be set to a certain temperature curve or duration based on the specific athlete or body part. Additionally, it needed to be sanitary and easy to use so it did not compromise the benefits of the current method. Communication with the customer confirmed the need for this type of device; a preference for a touch screen display over buttons or a phone app was indicated. To begin our design process, we first had to brainstorm various methods that could be used in order to achieve our main objectives. This meant we had to consider different methods of cooling, how to control all those cooling systems, and how to power all of them. Our three major functions were broken down into the following: temperature control, user interface, and power systems. Temperature regulation focused on the method of cooling, adjusting temperature, and how to drive the cooling system. The user interface looked at how to control and monitor the system, as well as how to interact with users. The power system's main objective was to provide power to the other two systems such that their needs were met while not affecting the other systems. It also looked at how to connect everything together physically.

Thermal Regulation:

The role of the temperature regulation system is simple and self evident. It needs to take the control signal from the user interface and use it to provide the desired temperature to the surface of the skin. There were many approaches to doing this but all of them needed at least to be able to accomplish the goal shown in the Figure 10 below:

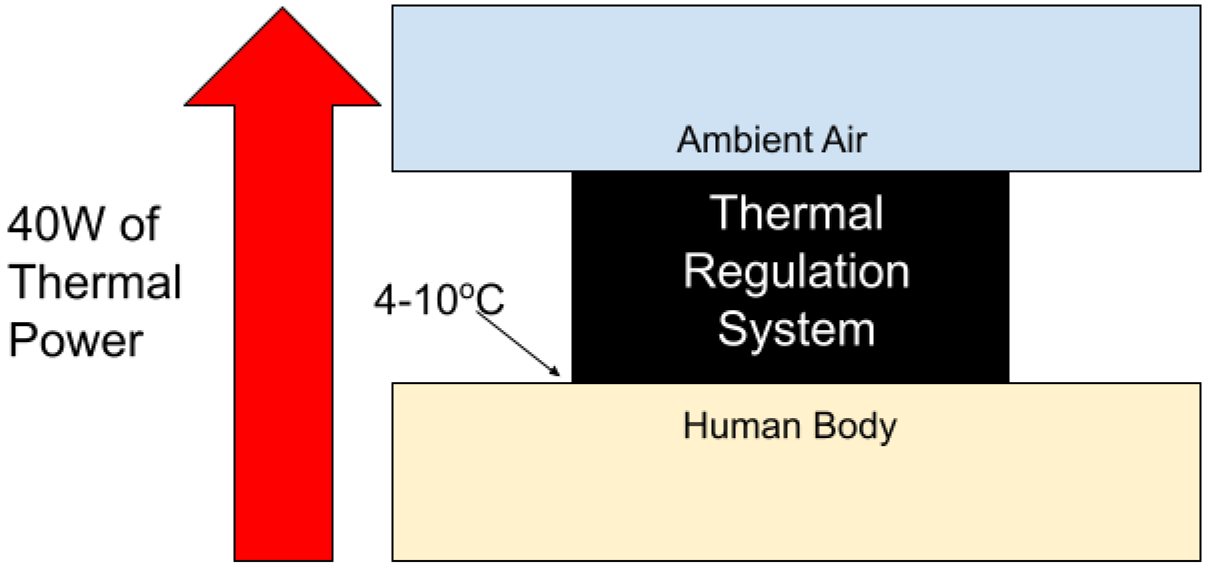


Figure 10: Thermal Regulation Goal

There are many different broad approaches to accomplishing this goal but we mainly considered three. These were air cooling, water cooling, and direct electric cooling. The idea for air cooling derived from the concept of air conditioning. Specifically, if it was possible to decrease the temperature in one's home with an air conditioner something similar could be done for applying cold treatment; this could allow for fast temperature regulation. The only device that we found that used entirely air cooling for medical purposes was the Zimmer Cryo 6 [16]. This device pumps cold air through a tube and can provide precision cold treatment to reach temperatures of -30 degrees Celsius. While this was an effective icing device, its price was between 6 and 7 thousand dollars. Furthermore, the temperature is way too cold for our purposes, it is massive, requires training to use, and it is meant for treating a very small precise spot rather than an overall area or appendage.

Figure 11: Cryo 6 [16]

Unfortunately there were too many drawbacks for us to realistically consider an air cooling system. These include cost, size, and implementation constraints. Specifically, making a reliable air conditioner relies heavily on fluid dynamics which none of us had experience in. At the same time we could not buy one because any that could meet the requirements that we needed were way too big and expensive.

Water cooling showed slightly more promise for the following reasons. The main advantage of water cooling is its ability to keep the electronics of the device far from the treatment area. The idea is you cool down the water in the case of your device and then circulate the cooled water in watertight tubes which filter through a cuff and cool the skin. One such cuff which implements a liquid cooling system is shown below (Fig. 12):

Figure 12: Water Cooling Cuff [7]

This cuff is sold with a separate cooling device which has tube outputs which connect to the cuff. A design like this allows for the cuff to be relatively lightweight and simple, and comes with a controlling device that allows for temperature control. Some existing products are also reasonably priced, costing hundreds, not thousands, of dollars. The Thermalzone system that goes with the cuff shown above is $550, for instance. However, the existing products would not be able to meet all of the training room's desires for a product. The temperature regulation that exists lacks quick and easy presets which follow a temperature curve optimal for healing. The device also requires the user to add water for the product to work. The use of water also leads to the potential for the system to leak, which can be messy and hard to repair. Using water also increases the time the system needs in order to get to the desired temperature. One system diagram we considered is shown below in Figure 13:

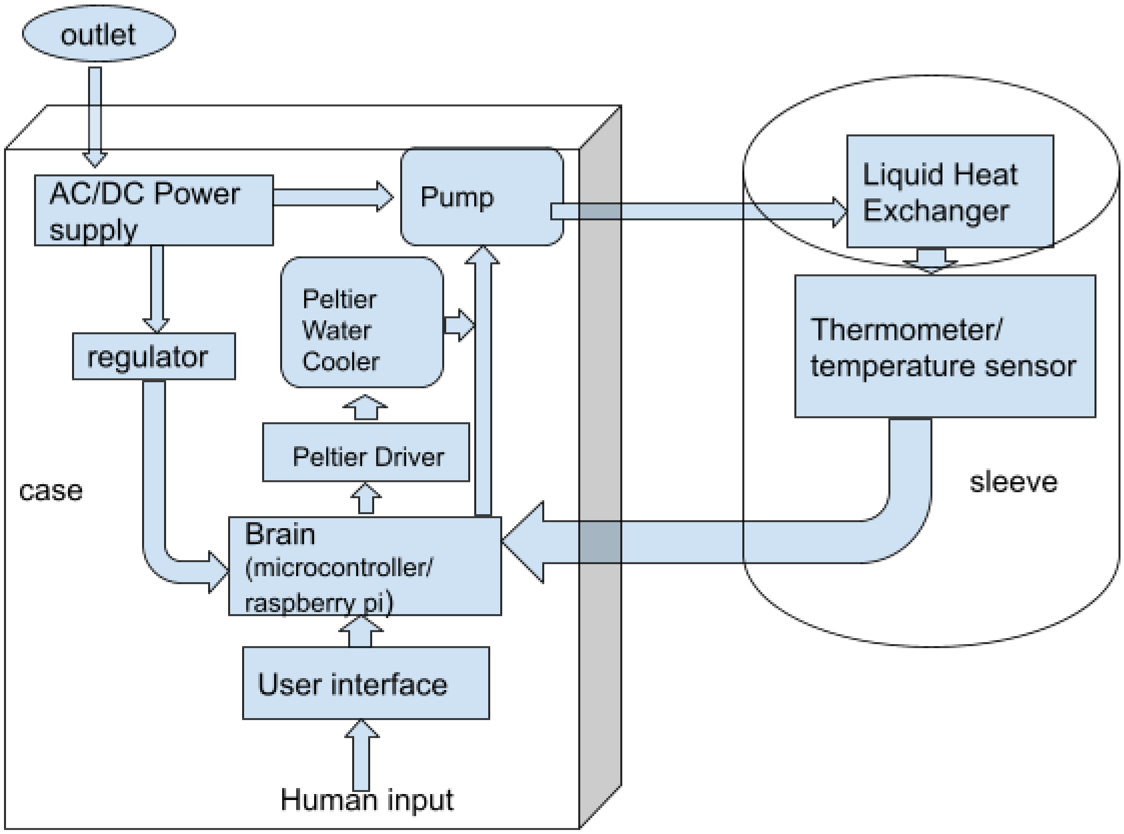


Figure 13: Water Cooling Approach

The last approach we considered was direct electric cooling, in which thermoelectric modules would be applied to the surface of the cooling cuff. Of the three approaches we considered, this was the only one where we were unable to find a commercial product which used it. There were, however, many studies which showed the feasibility of the method using Peltier coolers[12][13]. There was also a pre-commercial product, the Recoverx [17]. The main disadvantage of this method is its requirement that the Peltier coolers be close to the surface of the skin. This means that both the Peltier coolers and the heat sinks would have to be located in the cooling cuff, which increases its weight and bulkiness. However, this comes with key advantages to offset this drawback. Firstly, the system does not use any water to operate. All it needs is a wall outlet and the system does the rest--very few systems on the market can say the same. Another advantage of the system is its simplicity. Every method has to cool things down, but air and water based methods also have to transport the cooled fluid to the area that needs to be cooled. The elimination of this step makes the implementation easier. This method can also provide temperature regulation, be easy to clean, and run for long stretches of time. For these reasons, we choose this general approach to providing cooling.

With this decision made, there were a lot of smaller alternate designs that we considered. A very important question was how to drive the Peltier coolers. There are many chips that already exist that are designed specifically to drive Peltier coolers. We were also capable of building our own simpler driving device. Buying a device to drive the Peltier units comes with many advantages. Firstly, buying a system allows us to have access to something many times more complicated than anything we could hope to build. This complexity can give the chip things like thermal shutdown capabilities, fine-tuning control options, and enhanced efficiency compared to a system we might build. It also could come with application notes, which detail recommended ways to design Peltier systems around the chip, and which could help us make design decisions. We researched some options for possible TEC, or thermoelectric cooler, drivers.

The first option we considered was the chip used by the Texas Instruments application report on driving Peltier units, the TSP54200 [18]. However, this driver was designed for driving LEDs and would require a lot of alteration to make it applicable to our project [19]. On the expensive side, we were able to find a 28V 15A Peltier driver with many customizable features [20]. However, these units were over $300 each and would require powerful and expensive Peltier units, and we assessed that the cost would not be worth it. It was also apparent that buying one of these would require a great deal of correspondence with the manufacturer, which was undesirable. The MAX 1969 Peltier driver occupies a middle ground between these two extremes [21]. They are around $20 each, capable of providing up to 30W of power each and include detailed application notes. Their main downside is they are only capable of using low voltages below 6V. It became clear that if we were going to buy a TEC driver, it would be the MAX 1969. The application notes that our design would have been based on had we used this chip are shown below (Fig. 14):

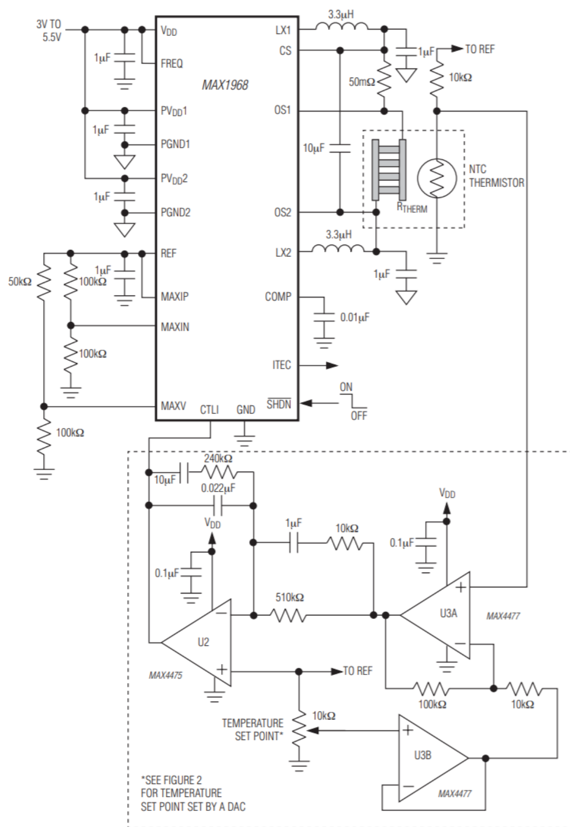


Figure 14: Alternate Design Application Notes

However, as mentioned before, driving the Peltiers with a prebuilt chip comes with many downsides, and we were eager to have alternatives. For this reason, we also considered possible ways to build our own Peltier drivers, which proved to be feasible. In the end, we did decide to build our own Peltier driver, for a number of reasons. The biggest of these was the difficulty of building around a chip like the ones we had been looking at. The package these chips come in was simply not very conducive to small scale soldering. Trying to obtain the chip pre-soldered to all the relevant components would also be expensive and require correspondence with a company to achieve, which, again, is undesirable. The complexity of this chip, which earlier was an advantage, can also be a downside. The larger and more complex the chip, the more difficult it is to connect and the harder the system as a whole is to build. There is also the possibility that the complexity could reduce our understanding of the minutiae of our system, which would make troubleshooting and testing more difficult than they would otherwise be. For these reasons, we decided to forego the benefits of buying a TEC driver and build one ourselves, which we detail further in the design section.

User Interface:

In order to control the various components of the product, some kind of programmable device would need to be used to serve as the brain of the machine. Therefore, research was done to choose the type of processor needed, given the specifications required. We considered the pros and cons of each option, most heavily considering an Arduino Uno Wifi or a Raspberry Pi 4. The machine would need to complete multiple processes simultaneously, such as read an input temperature from a thermistor, output a signal to the temperature control unit, and power the controls of the interface. It would also need to not consume an excessive amount of power, be available at a reasonable price, and be able to handle the commands needed to be executed. First, when considering the power consumption of each device, a Raspberry Pi can consume up to 3A, while an Arduino can consume up to 500mA [22]. Though the Arduino draws significantly less current, it would not be unreasonable to supply the Raspberry Pi with 3A if needed. Therefore, either could be used. Next, we found that both fell within a price range of less than $100, which we felt was reasonable, and were compatible with various accessories such as displays and chips. The most important requirement was the device's ability to complete the required tasks such as reading signals and writing output signals while maintaining a functional user interface. The Raspberry Pi is characterized as a mini-computer with its own operating system that can execute complex tasks simultaneously, while the Arduino is a microcontroller that can complete simple tasks one at a time [23]. Given the complexity of our intended system, the Raspberry Pi was the clear choice for the brain of the athletic recovery device since it will have to monitor and control multiple systems simultaneously.

There were a few different directions we could go in terms of designing a physical user interface. For example, we could develop a web site that the Raspberry Pi could connect to via the internet, or we could create a graphical user interface (GUI) program to be executed. Specifically, the technical work behind a website and a GUI would involve a combination of the HTML/CSS and Python programming languages, respectively. First, we considered the main characteristics of web design: accessibility from different devices and navigation tools. The nature of the internet lends itself to being available on all sorts of hosts, such as laptops, phones, and touch screen tablets. In the training room environment, it was communicated by the trainer that controlling the machine from an external device would not be convenient or useful [1]. Additionally, designing for a web page would not have an advantage in this situation given that the controls would only be accessed by the recovery device itself. A characteristic of web design is that it allows the user to be in control of their navigation; since the only application that the user needs will be the interface itself, there is no need for the user to have this ability to engage with external pages [24]. To understand not just why the web page wouldn't be ideal, but why a GUI would be, we must consider the qualities of a GUI. There are a multitude of frameworks that support GUI development for the Raspberry Pi. For example, guizero library, Kyvi Python, Electron, and Tkinter. The ideal specifications for a GUI framework compatible with a Raspberry Pi would be available in Python, offer many different widget options, and have a reasonable learning curve. To summarize the disadvantages of each framework, guizero has a limited options to cater to beginner designers, Kyvi Python requires previous knowledge of layout design, Electron uses an extensive amount of Node.js dependencies, and Tkinter appears outdated [25]. Even if the design is outdated, the effectiveness of the overall user experience that we have in mind would not be impaired, therefore Tkinter would be ideal. Overall, since there are no inherent advantages to web design, and the Tkinter framework would fit the requirements for a GUI developed on the Raspberry Pi, that is the option we decided on.

Power System:

There were two ways to go about the power system: design and build our own or purchase a premade one. If we were to construct our own, we would have needed to consider how to make our output either AC or DC based on what the loads demanded, look at how to step up or down the voltage, and how to regulate and isolate the systems. The different components that would need to be considered in a DC power supply connected to a wall outlet can be seen in Figure 15 below.

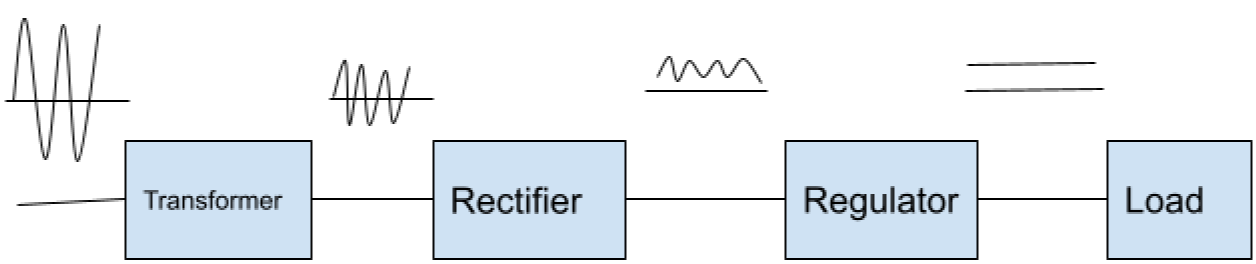


Figure 15: DC Power Supply Subsystem

The benefit to constructing such a system is that it could be more specifically designed to meet the needs of our intended system. The disadvantage of designing such a system would be that it would be very complex to build and it most certainly would not have all the built in safety features that a purchased one would include. The biggest disadvantage of purchasing a power supply would be its cost. In our situation, we expect to be dealing with a human patient and are expected to need roughly 40W, which could imply drawing many amps. Therefore, to ensure the safety of both the system and the user, the most reasonable choice is to spend the extra money on a pre-made power supply. The next difficulty from here is deciding which power supply to choose.

When it came to connecting various physical components together, there were various options available. There was screwing them together, clipping them together, or using some adhesive to stick them together. Since the system overall needs to be connected together such that thermal energy is transferred efficiently. At the same time, these would provide the most secure and firm connection between components. On the other hand, thermal adhesives would make it easy to apply, and depending on the adhesive, it may be strong enough to firmly hold the components together. In order to use screws and clips other pieces of equipment, such as a drip press, may be needed. Since this device will be resting on a stationary individual and not be used during vigorous motion, we decided to go with some form of thermal adhesive since it would be easier to apply, cheaper, lighter and could provide similar levels of thermal conductivity.

Preliminary Proposed Design

Our preliminary design involved the integration of three broad subsystems, whose structured block diagram can be seen in Figure 16 below.

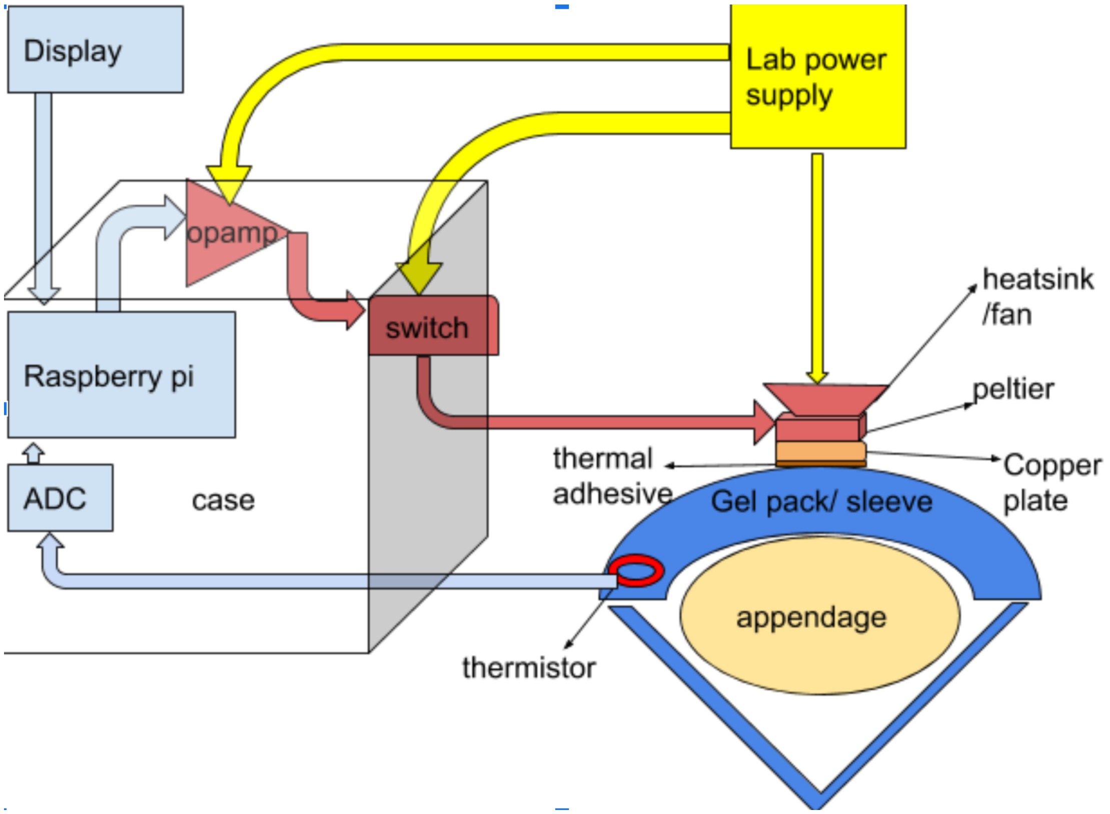


Figure 16: General Block Diagram

The three key subsystems for our preliminary design are arranged according to the following color scheme: red for components relating to thermal regulation, light blue for the user interface, and yellow for the power supply system. From here, we dove into deeper detail about how to design each system. Throughout the process we revisited our specifications to ensure we were meeting our requirements.

<u>Thermal Regulation</u>

The preliminary implementation of the thermal regulation system begins with the overall motivation for the system. Mainly, the system must induce at least 40W of continuous heat transfer away from the human body, maintaining a temperature of at most 10ºC at the contact point with the skin. We decided to do this using a modular system to make testing and building the device easier and the device more effective. The device was intended to ultimately consist of five identical modules each controlled by the single mini-computer and powered by the single power supply. The advantages of a modular system are manyfold.

One concern brought up early in our design process was the difficulty of ensuring the heat transfer away from the body occurs uniformly throughout the area being cooled. After all, Peltier units themselves are relatively small in area, and could never cover the entire surface being cooled in the design. This means that areas not directly underneath the Peltier units would not be cooled as effectively as those directly underneath them. Having many separate Peltier units mitigates this issue somewhat, compared to having one or few Peltier units. The modular system also simplifies building and testing tremendously. Instead of having to build the entire system in order to begin testing, we could build a single module and perform tests on it and be confident the results will be applicable to our final design. In order to further mitigate the uneven heat transfer, we planned to have a 2”x1/8”x4” copper plate between the Peltier and gel pack to help smooth out the heat transfer. A simplified diagram of the intended heat transfer is shown below (Fig. 17):

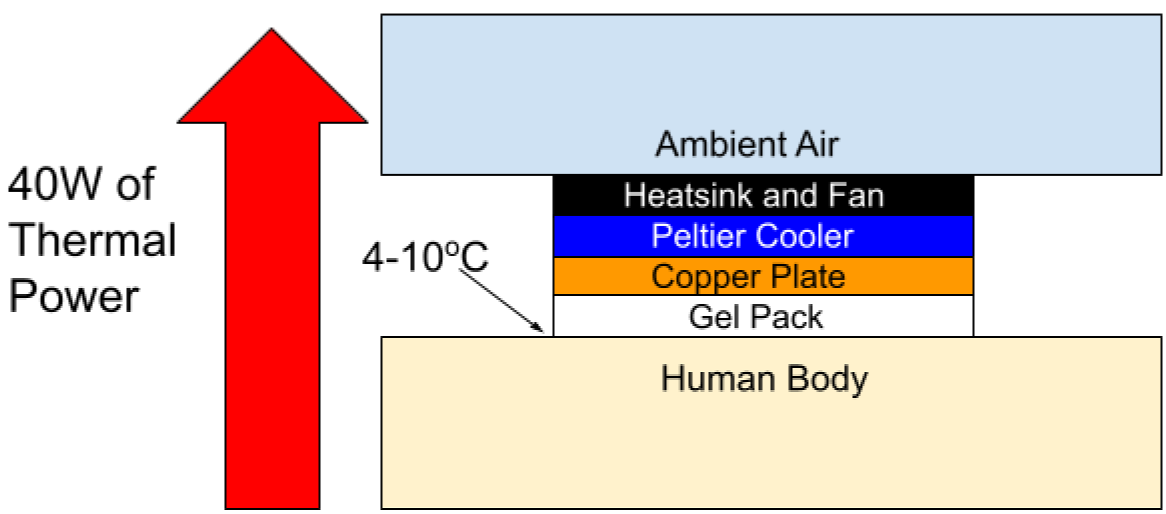


Figure 17: Thermal Transfer System

Peltier Coolers:

We can now look in some detail at the preliminary design of a single one of our modules. We will begin with the choice of the Peltier unit, as this largely determined the driving method used later on. As mentioned before, to ensure the heat transfer away from the body is as even as possible, there is a desire to employ a large number of Peltier coolers. The two main options we considered were the TEC-30-38-71 and the TEC1-04905. The main difference between the units is that the TEC-30-38-71 is a slightly more powerful unit, requiring higher voltages of up to 8V to operate, compared to the lower operating voltages of the TEC1-04905 of up to 6V [26][27]. Another difference between them is the smaller unit is slightly more efficient. Though the units were similar, we chose the TEC1-04905 because of the preference for more, less powerful Peltier units and the greater efficiency of the smaller units. We decided to use two Peltiers on each subsystem. Because the TEC requires voltages of above 5V and there will almost inevitably be some voltage drop across the driver, we also decided to go with a 12V power supply to make sure there will be enough voltage to run the Peltiers at their rated conditions. With that decided, we could finally turn to the design of the Peltier driver itself.

Peltier Driver:

One fundamental thing to understand is that Peltier coolers are current driven systems. That is, their effectiveness is directly related to the current passing through them. It then made a lot of sense to drive the Peltiers with an adjustable current source, to give the most flexible and direct control over the Peltier units. The preliminary circuit diagram for this adjustable current source is shown in the Figure 18 below:

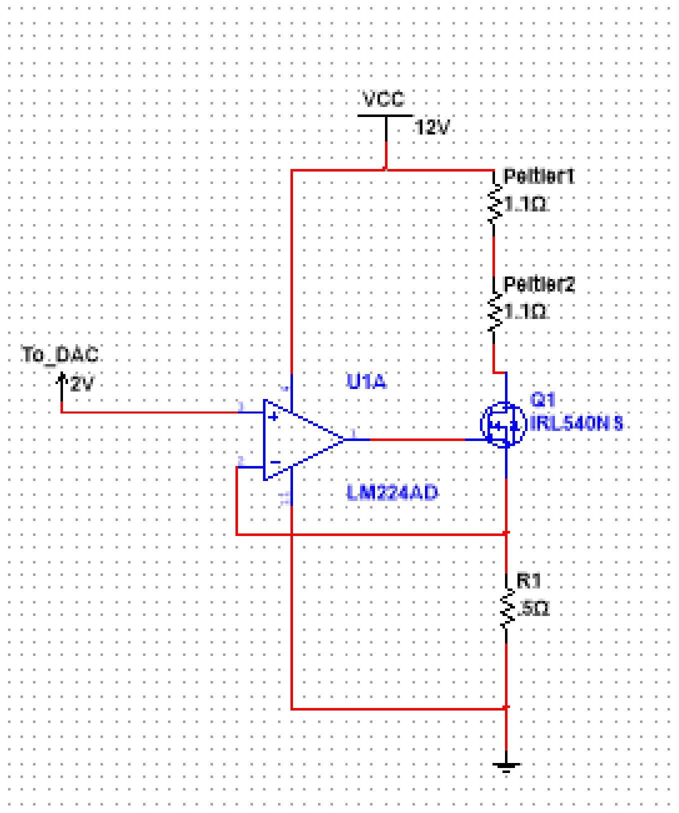


Figure 18: Adjustable Current Source

From this diagram, some implementational details become clear. Firstly, the control input would come from a digital to analog converter which would give an output voltage between 0V and 2.5V. This will set the current source to provide between 0A and 5A. Here, the Peltier units are modeled as resistors. This is a slight simplification, as the effective resistance of the Peltier units changes slightly both with ambient temperature and the temperature gradient across them. However, this change is small and the system still works reasonably well even in the worse case. We chose to use a power MOSFET for the design, the IRL520NPbF, which can handle up to 10A and has a low $R_{DS,ON}$ of below 0.3Ω even at high currents. Because we chose to use a MOSFET, the op-amp we chose had no current requirements and we were able to just use one from one of Professor Buma's labs. Finally, it is important to note that we chose to have two of our small Peltiers wired to each current source in series. This lets us further spread out the Peltiers without increasing the number of current sources needed, which would save money and building time.

Temperature Sensor:

The next important feature to design was the thermistor temperature sensor. Luckily, the preliminary design we settled on was very similar to one from one of Professor Buma's ECE 386 labs, so we were able to borrow some of his components. The design uses only an instrumentation amplifier, a thermistor, and various resistors. The preliminary circuit diagram is shown below (Fig. 19):

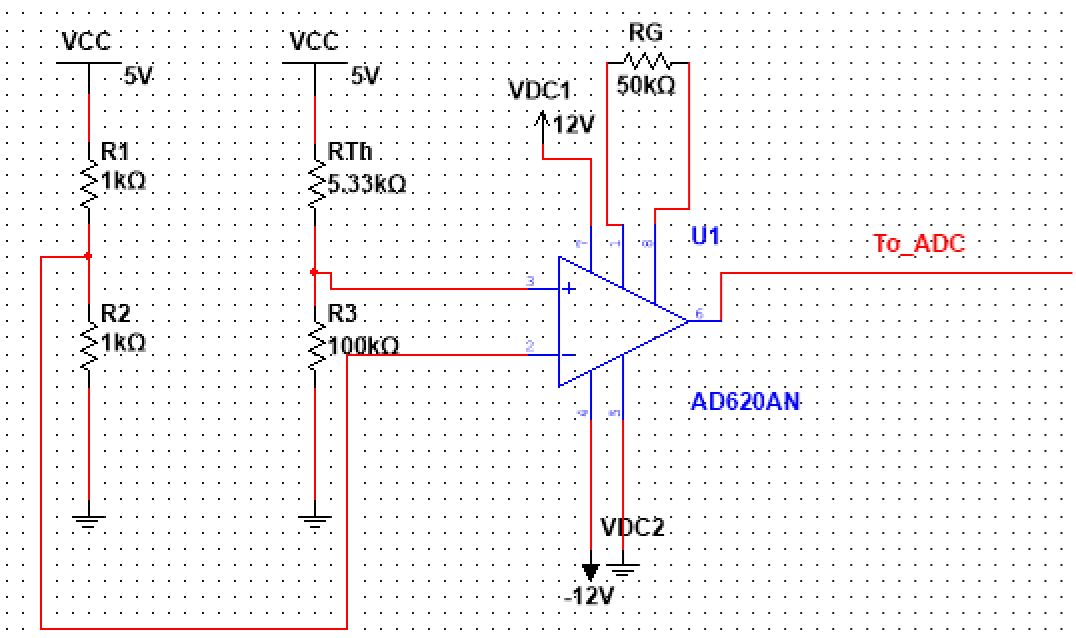


Figure 19: Temperature Measurement System

One important thing to mention is that the Raspberry Pi cannot handle analog voltages directly, so the output signal must be routed through an analog to digital converter before being read. Each module in our preliminary design would have one thermistor temperature sensor. We wanted each thermistor to be able to measure temperatures ranging from -20ºC to 40ºC.

According to the thermistor's datasheet, this corresponds to thermistor resistances(R_T) of 96.36kΩ and 5.33kΩ respectively [28]. Since we wanted the output of the temperature sensor to be a value between 0V and 5V, we designed the voltage divider for the non-inverting input to vary from around 2.5V to around 5V, set the inverting input to 2.5V with a voltage divider, and set the gain to 2 with a gain resistor of 51kΩ.

Heat Sinks:

One last consideration is which heat sink to use. It is critically important that the hot side of the Peltier coolers remain reasonably close to the ambient temperature. The greater the temperature difference across the Peltier coolers, the less power they can provide. Keeping in mind that the Peltiers will likely operate at around 40% efficiency, the heat sink of each Peltier would need to dissipate 10W of power [26]. Given this, we chose the HSF-55-33-B-F, which includes a fan and has a thermal conductivity of 0.6 ^{o}C/W [29]. This means that at full power, these units can keep the hot side of the Peltiers less than 6^{o}C above the ambient temperature. A diagram of the system surrounding a single Peltier unit is shown below (Fig. 20). Recall that the final product was intended to include 5 modules, each with 2 Peltier coolers, two heatsinks, and one temperature sensor.

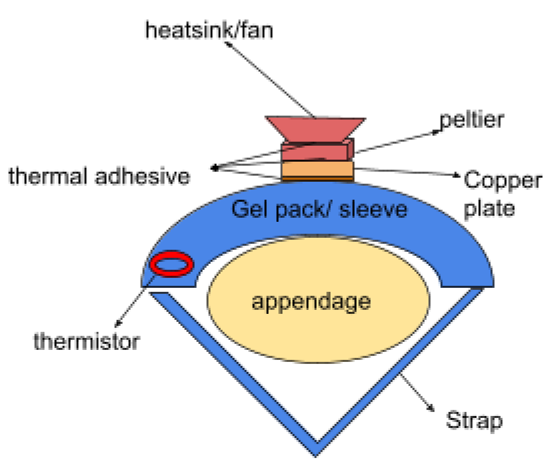


Figure 20: Thermal Regulation System

<u>User Interface:</u>

<u>Input Signal</u>

The user interface for this project involves the implementation and integration of both software and hardware components. The external inputs to the system include a signal based on the temperature on the surface of the sleeve, and the output of the user interface system converts the input signal to drive the thermoelectric cooling module. A more detailed preliminary block diagram of how the input from the thermistor is processed, which will be elaborated on in the coming passages, is shown in Figure 21 below.

Figure 21: Thermistor Signal Processing

In order to better understand how the signals of the thermistor would interact with the input to the Raspberry Pi, research was done to learn how these devices communicate. First, we learned about the specifications of the NTC Thermistor we were supplied with, which we will use to take measurements on the exterior surface of the gel pack. An NTC (Negative Temperature Coefficient) Thermistor will exhibit a resistance that decreases as temperature increases and will be known with an accuracy of 0.5 degrees C over temperatures ranging from 0 to 50 degrees Celsius. This output signal as a resistance value will be converted to a temperature using the Steinhart-Hart equation. As shown in the Raspberry Pi 4 documentation, the general-purpose input/output pins (GPIO) are digital (3.3V high and 0V low) [30]. Therefore, we will need to do analog to digital signal conversion. Since the Raspberry Pi does not have hardware to do this conversion built in, an external ADC will be used, specifically the chip

MCP3008 [31]. For the software needed to convert this signal, Python code for Serial Peripheral Interface (SPI) would need to be implemented on the Raspberry Pi to read external analog devices.

Physical Interface

Once the Raspberry Pi kit and the touchscreen were delivered in the latter half of the fall term, we connected the two so that the Raspberry Pi now had its own physical monitor in the form of a 7 inch touch screen. Next, we tried to begin to download software packages on the Raspberry Pi through the Linux terminal. First, we had to download a virtual keyboard, since we didn't have access to one. After difficulties with SSH-ing into the Raspberry Pi, we instead chose to copy the commands onto a txt file which we downloaded on a USB. Since Union's residential WiFi requires the device to be registered for use, we found the MAC (Media Access Control) Address to utilize the MyResnetLegacy connection. Using the commands, `sudo apt update; sudo apt upgrade; sudo apt install matchbox-keyboard`, we were able to successfully enable a virtual keyboard. Next, we enabled the SPI and I2C Interface setting in the Raspberry Pi Configuration menu. From here, we downloaded the software necessary to communicate with the MCP3008 so that we could obtain an analog signal from the thermistor.

Reading Analog Signal from ADC

As shown in the document "Analog Inputs for Raspberry Pi Using the MCP3008", the packages for pip3, adafruit-blinka, and mcp3008 must be installed [1]. Then, a Python script would be used to read the analog value, convert it to a range, and print the new output to the Pi. Using the wget command, followed by the url containing the python code, the script would be saved to the Raspberry Pi and can be accessed using the following command `sudo python3 ./Analog_Inputs_for_Raspberry_Pi_Using_the_MCP3008.py`. Next, we implemented the hardware component for reading an analog input from a trimpot using the MCP3008; this circuit is shown in Figure 22.

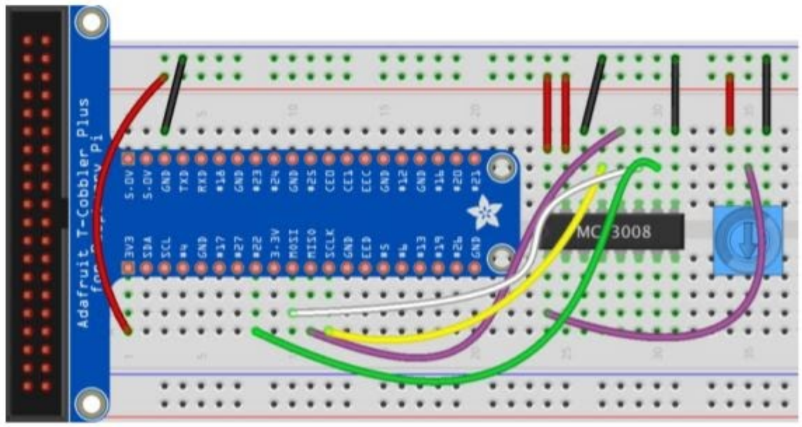


Figure 22: Circuit for Reading Analog Input from Trimpot using ADC

After running the Python code from the source above with the trimpot circuit, we were able to read the signal and print to the Raspberry Pi terminal (Figs. 23 and 24).

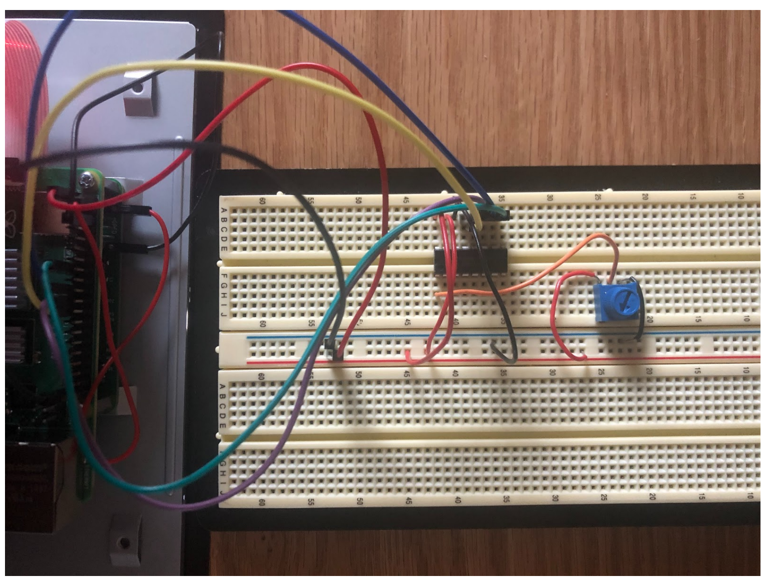


Figure 23 Implementation of ADC Circuit

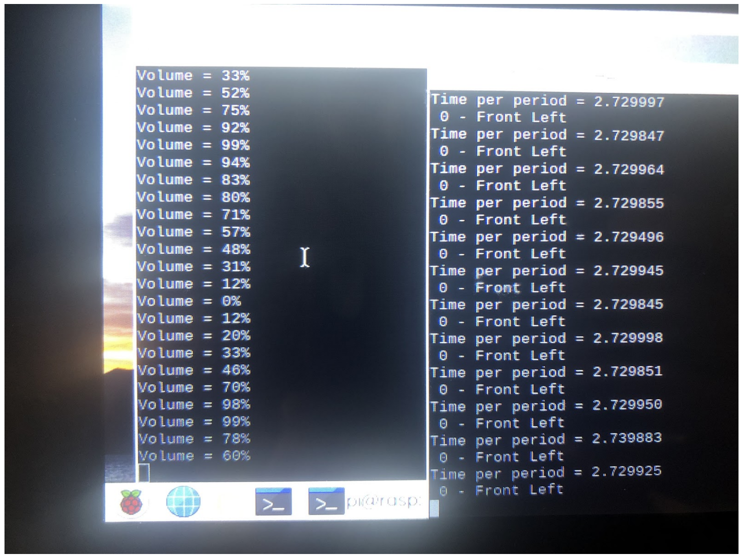


Figure 24: Terminal Output of Trimpot Signal

Obtaining Temperature Value

Next, we began to configure the signals to the Raspberry Pi from the value of the thermistor. Using the datasheet for the Vishay 103 NTC Thermistor, we found three temperature (°C) to resistance (kΩ) points to plug into a systems of equations solver for the three coefficients (x, y, and z) of the Steinhart-Hart equation: $T = [x + y\ln(1000R) + z\ln(1000R)^3]^{-1} - 273.15$ (Fig. 25).

$$\left\{x + 9.21034\,y + 781.31658\,z = \frac{1}{298.15},\; x + 11.47585\,y + 1511.3122\,z = \frac{1}{253.15},\; x + 7.82004\,y + 478.21874\,z = \frac{1}{333.15}\right\}$$

Result:

$$\{x + 9.21034\,y + 781.317\,z = 0.00335402,\; x + 11.4759\,y + 1511.31\,z = 0.00395023,\; x + 7.82004\,y + 478.219\,z = 0.00300165\}$$

Solution:

$$x \approx 0.00113414, \quad y \approx 0.000233106, \quad z \approx 9.32975 \times 10^{-8}$$

Figure 25: Solving Coefficients of Steinhart-Hart Equation

We then designed a circuit, shown above in Figure 19, that would let us read the thermistor resistance and output an amplified voltage to the analog to digital converter (ADC). Using this voltage, we would convert to the internal resistance of the thermistor, in order to find the temperature from the equation above. In order to know how to convert the output voltage from the temperature sensor into a resistance, R_T, we analyze the circuit shown in Figure 19. We know that the output voltage is the difference between the non-inverting and inverting inputs multiplied by the gain, which is two:

Loading…

Then, we can simply solve for R_T:

Loading…

Note that despite the output bit length of the signal, the ADC will convert the value to 16 bits, which yields a range of 0 to 65535. The ADC has eight output channels, and the output can be either a raw voltage (chan0.voltage) or a scaled 16 bit value (chan0.value). By obtaining a voltage from the amplifier circuit connected to the output of the thermistor and reading that from the ADC output to the Raspberry Pi, we were able to successfully interpret temperature values from the NTC Thermistor.

Output Signal

Depending on the value of the temperature from the input signal, code would be developed in the final implementation to determine the value of the output signal from the Raspberry Pi, which would be used to control the thermoelectric circuit. As explained earlier in the "Thermal Regulation" section, the method we would use to drive the Peltier system is a continuous current method, which would require an analog input, but provide greater flexibility and increase the efficiency of the Peltiers. Since we wish to use a MOSFET for a variable current source to the

Peltiers, we need a 0 to 5V analog signal. Since the output of the Raspberry Pi is digital and we need a 0 to 5V analog signal (which will be the input to Fig. 23), we would need a digital to analog converter (DAC) with I2C configuration [32][33]; this hardware would be implemented in the final design.

To clarify, there are different types of software that would allow for serial communication between the converters and the Raspberry Pi, predominantly SPI (Serial Peripheral Interface) and I2C (Inter-Integrated Circuit). Half-duplex is the term for one way communication at a time between devices, rather than full-duplex, which can go both ways simultaneously. For our device, the communication would only need to go in one direction, therefore half-duplex (I2C uses) will suffice (SPI uses full-duplex). When communicating with more than two devices, SPI can get complicated due to the need for individual chip select (CS) lines. I2C, on the other hand, can handle many devices at the same time without wiring difficulties. Given that the chip for the ADC supports SPI and the chip for DAC supports I2C, we would use SPI for the software for the input signal and I2C for the software for the output signal.

Graphical User Interface

For the interface, the user must be able to communicate with the device in an effective and safe manner; Figure 26 below shows a draft of the physical user interface for controlling the device.

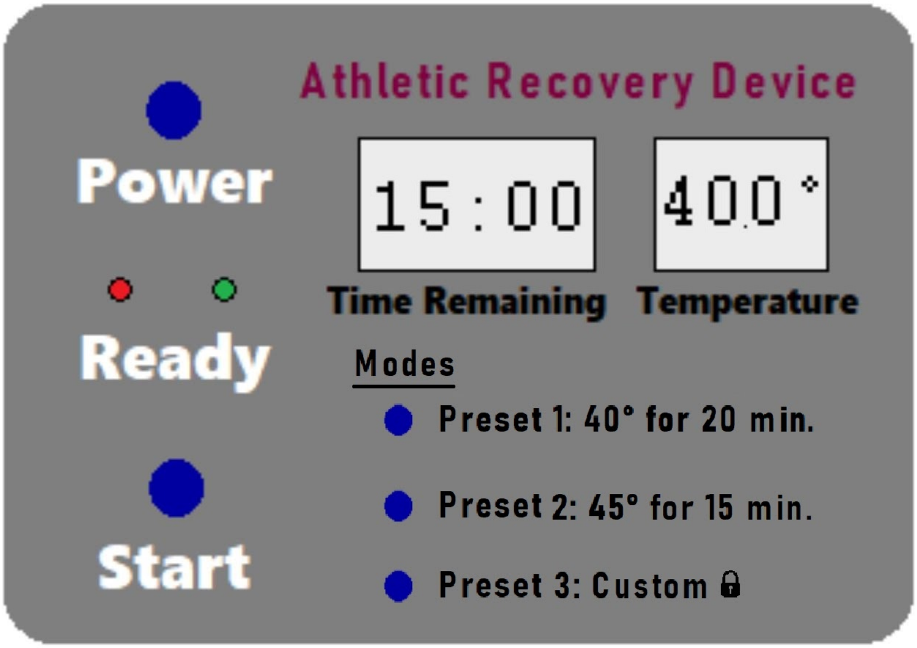


Figure 26: User Interface Appearance

For the physical command board shown above, which we planned to implement with a touchscreen display, there will be an on/off button, a red light to indicate that the cooling system is not ready, a green light that will turn on when the system is ready, a treatment start button, a display for the current temperature, and a display for the time left in the treatment period. Below each of the displays would be two preset treatment options. The third option would be a passcode locked customizable preset. For the user to be able to control the device through the touchscreen display, a python script using Tkinter framework would be developed.

<u>Power System</u>

<u>Power Supply</u>

In order to choose the appropriate power supply it was essential that the load requirements be clearly specified. For the thermal regulation, we were expecting to use 10 Peltier units. The Peltiers that have been selected, while current driven, require a DC voltage. At maximum they were expected to run at 5V and 5A. We intend to lay them out as shown in Figure 27, where there are 5 parallel sets of two Peltiers in series, with each set connected to a current source. The internal resistance of the current source and Peltiers in series is expected to consume roughly 2 volts.

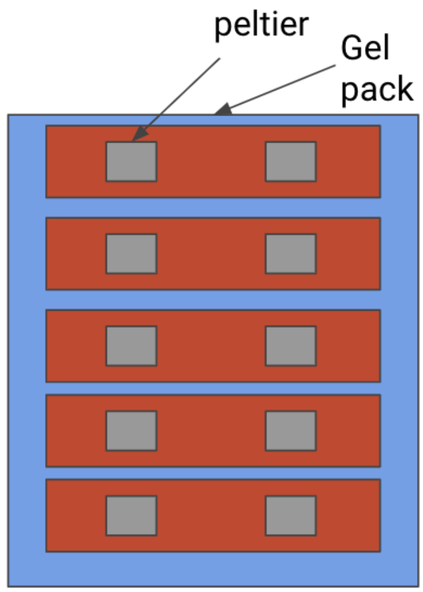


Figure 27: Peltier Layout

Since each set of Peltiers was expected to draw a max of 5A and there are 5 sets we were expecting a maximum current draw of 25A. At the same time, since each one consumes 5V and we expect 2V to be consumed by internal resistance, the overall voltage going into each set needs to be at least 12 volts. Furthermore, the heat sink fans that have been selected need 12VDC in order to work. Thus the overall thermal load requirements were 12V and 25Amps. This means that a power supply we selected needed to have at least a 50A current rating, and output 12VDC. As for the user interface, the Raspberry Pi 4 that has been selected needs a 5 volts DC and a minimum of 3amps in order to operate.

Power Distribution and Connection

Since both the thermal and user interface load requirements differ significantly, and deal with such large values, it is essential that both are properly isolated from one another. This means that the power supply we selected needs to have multiple channels in order to properly isolate the two systems. Furthermore, since the Raspberry Pi required a lower voltage, a voltage regulator that could step down the voltage and handle the required amount of current also needed to be selected. Our load requirements for our power supply include a 12VDC power supply that had multiple channels and had a current rating greater than or equal to 50A.

In the end we selected the SE-600-12 power supply, as seen in Figure 27, since it met all our load specifications. This is a 12VDC power supply, with three channels, and 50A current rating. Furthermore, this power supply has a built-in fan making it so we do not need to worry about cooling it [34]. Regarding a voltage regulator for the Raspberry Pi, we chose the Drok 5A USB voltage regulator. This regulator can step 12V down to 5V and can provide between 3 and 6A. In addition, it has a built in USB port which makes it capable of easily connecting to the Raspberry Pi [35].

Figure 28: SE-600-12 Power Supply

Since we are dealing with such high current draws, we needed to determine which wires we were going to use to connect the components. To do this we referenced this source [36], which lists out the current rating for different wire gauges. Unlike selecting a power supply that has a current rating double the expected maximum current draw, as the current rating of a wire gets bigger so does the actual wire itself. Larger wires get cumbersome when trying to wire a circuit and figure out if pin holes are too small for a given wire. This means that when selecting a wire, one should be just a little bit over the expected current. By looking at this site and knowing that we expect the maximum current to be 5A, we can determine that 20AWG is the ideal choice since it has a current rating of 6A and is still fairly small and able to fit into our breadboards [36]. This would also work for connecting the power supply to the voltage regulator. At the same time since the selected power supply only has three channels and we will need to power five sets of Peltiers we needed to get a cord that had five conductors so that one channel could neatly be able to power five sets. In addition, each wire for the power supply pelter connection needed to be of a higher rating. There are two reasons for this. Firstly, they need to have higher current ratings to account for the current drawn by the heat sink fans. Secondly, it is a good idea to have a higher current rating for the wire between the system and power supply, to deal with any large transients or inrushes that could occur. For this reason we chose to go with an 18AWG-5 conductor wire [37]. This is the next level above the 20AWG which is already going slightly above what is expected to be drawn.

In order to connect everything together, we settled on using some sort of adhesive. The two options we could have gone with were paste or double sided thermal tape. We decided to look into tape since it would be less messy and easier to apply. The most important feature we were looking for was the smallest thermal coefficient and ability to hold the system togher. As a result, we selected LLPT double sided Thermal Conductive tape, shown in Figure 29, since it had the smallest thermal coefficient we could find for tape [38].

Figure 29: LLPT Double Sided Thermal Tape [38]

While double sided copper tape may have had a smaller thermal coefficient, the price was too high to consider it. While there were initial concerns about whether the tape would be strong enough, preliminary tests quickly resolved these.

<u>Container and Sleeve:</u>

For the container we needed to choose a case that could hold all the expected electronics including the power supply, Raspberry Pi 4, and the Peltier drives. In addition, it needed to be made of a material that was insulating and durable. In the end, we selected the junction box seen in Figure 29 below because we believed it would be of the right size, and allow for easy wiring access [39]. That being said, we still needed to figure out how we will cut out space in the container to place the fans to cool other electronics in the case and the touchscreen to allow for convenient user access.

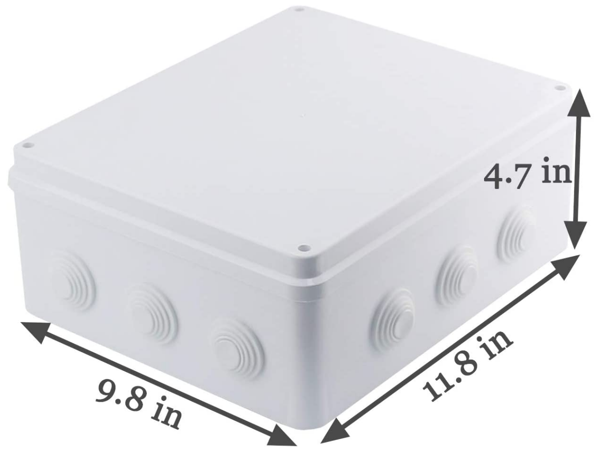


Figure 30: Case For Electronics

Regarding the pack, we decided it would be best to choose a multi-compartmental gel pack. The reason for this is that it could both work with the sets of Peltiers we are planning on making and could serve as a good way to prototype one system and then scale it up. In addition, based on the models we choose, we could get one that straps onto the user. The first gel pack we were using can be seen below in Figure 31 [40].

Figure 31: Gel Pack Sleeve [40]

Final Design and Implementation:

The final design of our system can be broken down into three categories: Thermal Regulation, Construction and Power, and Information Processing. Below is the final circuit diagram of our system, broken down into the Temperature Management System, the Information Processing Hardware, and the Peltier Driver System (Fig. 32). These are shown in more detail in the Thermal Regulation and Information Processing sections.

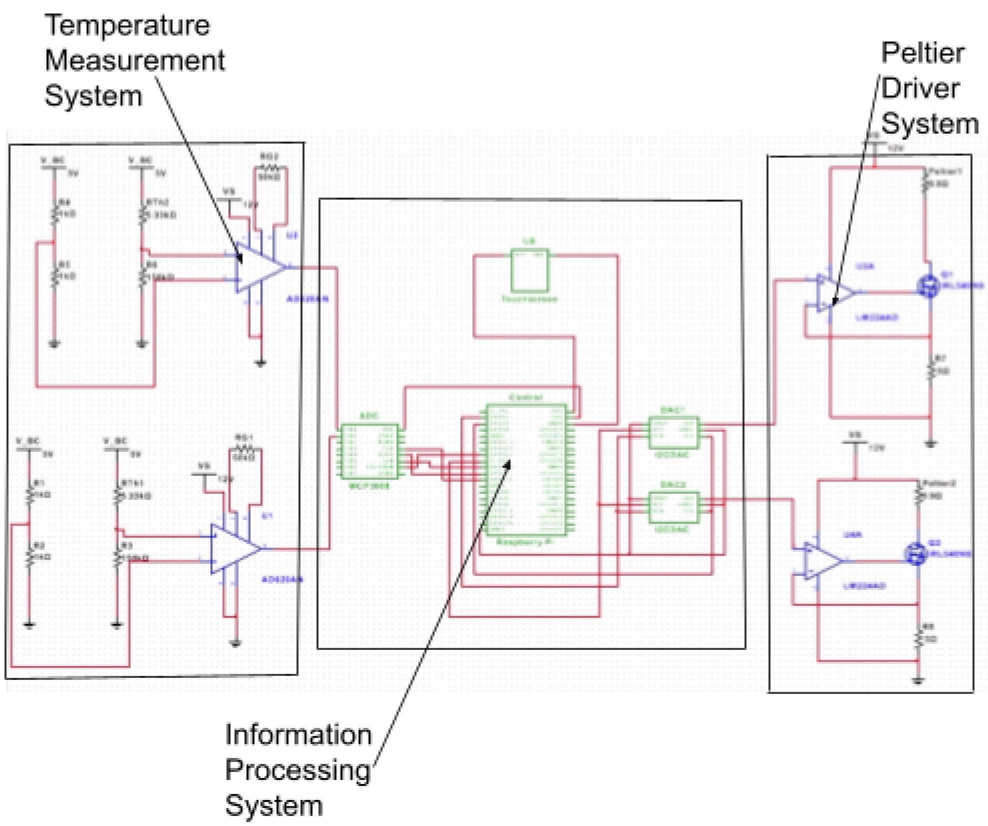


Figure 32: Complete Circuit Diagram

The final appearance of the project can be seen below in Figure 33, to be elaborated on further in the Information Processing and Construction and Power sections.

Figure 33: Final Design Appearance

Thermal Regulation:

<u>*Temperature Measurement System:*</u>

While the thermal regulation system was very similar to our preliminary design in the end, there were a number of changes made based on elements we later realized were not fully thought through in the preliminary design. One consideration that was not very well explored in our preliminary design was the headroom requirements of the instrumentation amplifier. Headroom is the requirement of op-amps and instrumentation amplifiers to be powered by a source of 1V above and below their maximum and minimum required output voltages. During initial testing, we had access to a -12V power supply as well as a 12V supply. Because of this, we were able to use the -12V supply to power the instrumentation amplifier along with the 12V supply. This meant we didn't have to worry about headroom, as we were trying to output a voltage between 0 and 5V. In the final design, we did not have access to a -12V supply, so we had to redesign the instrumentation amplifier circuit to manage without a -12V pin. This means the output of the instrumentation amplifier will only go between 1 and 5V instead of the 0 and

5V, which the ADC prefers, which will lead to a small loss in the accuracy of the temperature sensor. Since in the end we decided to only use two subsystems, the final design will contain two temperature sensors. Their circuit diagram is shown below in Figure 34.

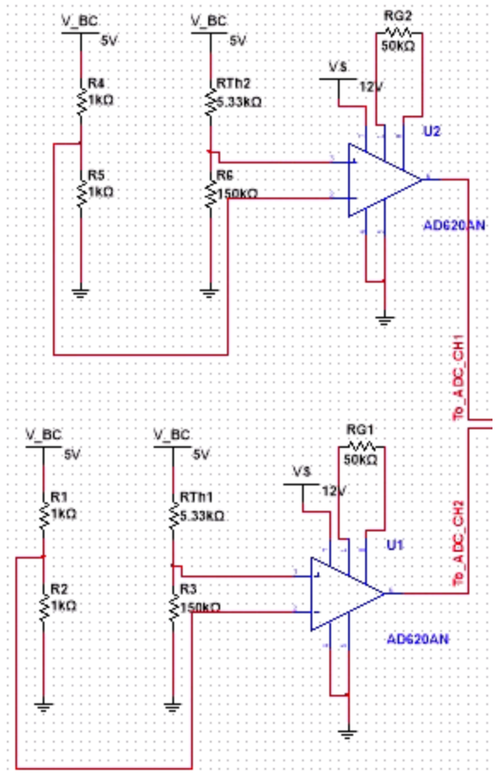


Figure 34: Temperature Sensors

The only difference between this circuit and the one in our preliminary design is that in this design we used a 150kΩ resistor instead of a 100kΩ resistor under the thermister. This reduced the minimum voltage the circuit would output to 1V, the minimum the instrumentation amplifier could supply, as shown in the equations below:

Loading…

Loading…

Loading…

With the design of the thermistor circuits complete, we were able to move on to the calibration of the system. We had a commercial temperature sensor that we used to perform the calibration. We took two temperature measurements, one at room temperature and one in a cup of ice water. During this, we noticed that the gain on the instrumentation varied slightly by around 0.1 on either side of 2 depending on the temperature. We found that this slight change of the gain in our equation had a significant impact on the temperature. We were able to compensate for this by implementing an approximation of the gain based on our measurements of it at two different output voltages. Even this very rough approximation helped to improve our temperature readings, and we found that we were within a couple degrees of our independent commercial temperature sensor. The measurements and calculations made in this process are summarized below in Table 2. We directly measured the positive and negative input voltages, the output voltage, and the supply voltage. From there, we calculated the gain and normalized ADC value.

Thermistor	**1**	**1**	**2**	**2**
Actual Temperature	71°F	32°F	71°F	32°F
$\mathbf{V_{OUT}}$	4.79V	3.77V	4.86V	3.82V
$\mathbf{V_{(+)}}$	5.47V	4.77V	5.48V	4.78V
$\mathbf{V_{(+)}}$	2.97V	2.97V	2.98V	2.98V
$\mathbf{V_{S}}$	5.95V	5.95V	5.95V	5.95V
G	1.916	2.094	1.944	2.12
$\mathbf{ADC_{NORM}}$	0.805	0.634	0.817	0.642

Table 2: Thermistor Calibration

The equations we used to calculate the gain in our table and for our linear approximation are shown below:

Loading…

Loading…

Loading…

Loading…

Loading…

From this equation, we obtained our final gain equation, for thermistor circuit 1 and 2 respectively:

Loading…

Loading…

At this point, our temperature measurements were still noisy, with noise amplitude getting up to 2°F peak, even with a capacitor across the buck converter output. To fix this, we soldered two 100 μF capacitors between the ADC inputs and ground. This reduced the noise to 1°F peak, which was acceptable, as shown below in Figure 35 and Figure 36.

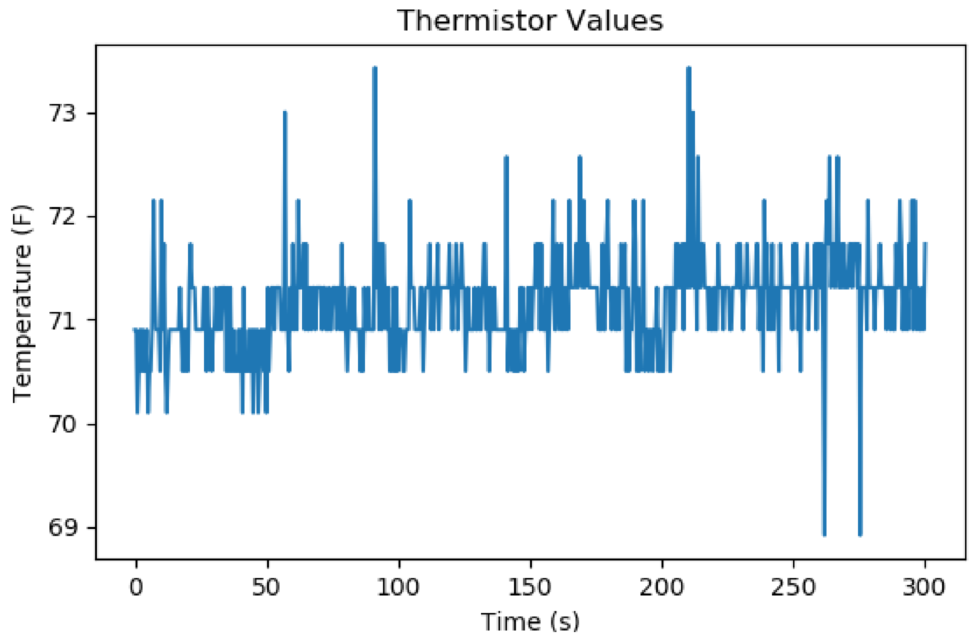


Figure 35: ADC with Only One Capacitor

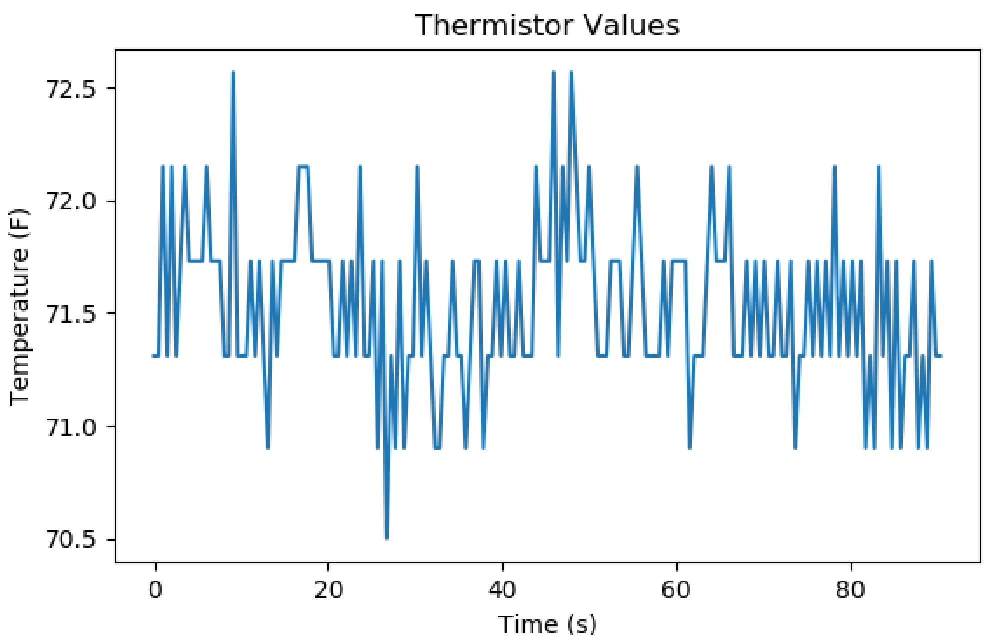


Figure 36: ADC with Three Capacitors

Peltier Driver System:

With the temperature measurement system complete, we could move on to the Peltier driver system; the complete circuit diagram of which is shown below in Figure 37. The main problem we encountered with this subsystem was a sudden exhaustion of the supply of the Peltier units we initially intended to use, TEC1-4095.

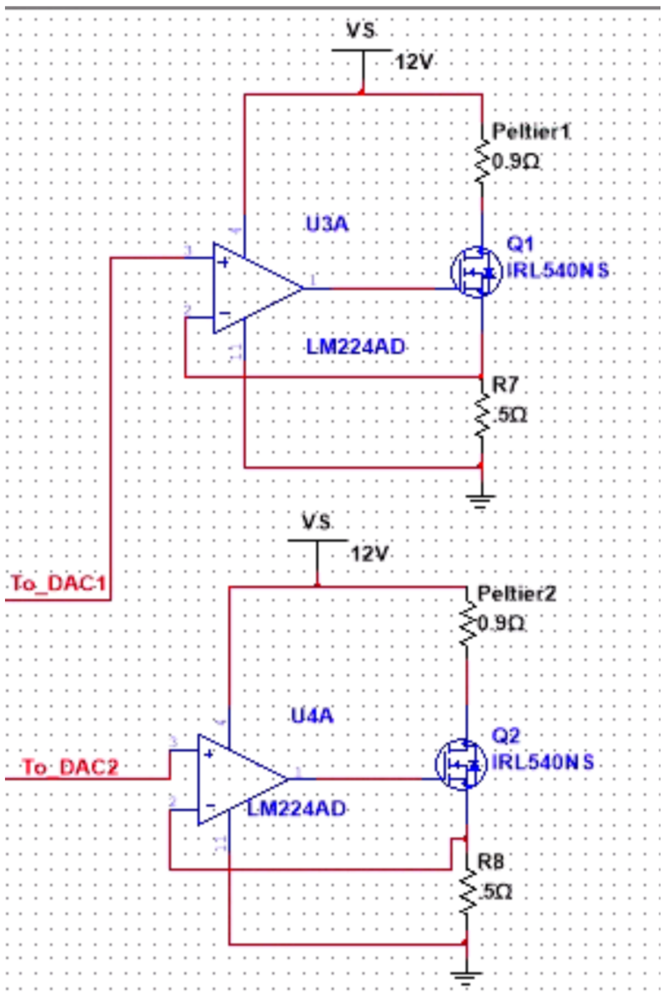


Figure 37: Peltier Driver System

As a result, we needed to select new Peltier units that would be compatible with our system; in the end, we settled on the CP85338 model. Since we decided to use the copper bars, it does not seem necessary to use two Peltiers units on each subsystem. We found that for a single

Peltier unit, the copper bar distributed the temperature well. If we had wanted to add a second, this would not have been very practical. Most reasonably priced Peltiers have higher typical resistance ratings, meaning that using two of them would cap out the Amps our current source could provide rather quickly. This meant we had to buy slightly larger heatsinks to compensate for the Peltiers being more concentrated; we decided to buy the HSF-55-45-B-F heatsinks and fans.

The CP85338 model has a number of advantages. For example, it has a maximum voltage rating of a little over 8V and a maximum current a little over 8A[42]. This means that if we need to, we can push our current source a little harder than we initially designed without damaging the Peltiers or running into voltage capping issues. It is also only slightly less effective than the Peltiers we designed for under likely operating conditions, and it is relatively inexpensive.

Power System and Overall Construction:

When it came to the power system, not much changed from the preliminary design. To clarify, in our final design, we still ended up using many of the same components that were discussed in the Preliminary Proposed Design section. This included the SE-600-12 power supply, the Drok 5A USB voltage regulator, and the 18AWG-5 conductor cable wire. At the same time, even though many of the initially proposed designs remained the same, the final load requirements differed quite significantly. The main reason for this is due to the final implementation only consisting of two subsystems, as seen below in Figure 38. In particular, in the preliminary design, when we had envisioned using five subsystems, we were expecting the max power requirements to be 12V and 25A. Now, with just two subsystems, each with Peltiers that were almost equivalent in voltage and current requirements to two of our previous Peltiers in series, our new max power requirements became 12V and 16A. In regards to the user interface, the load requirements remained unchanged from the original design.

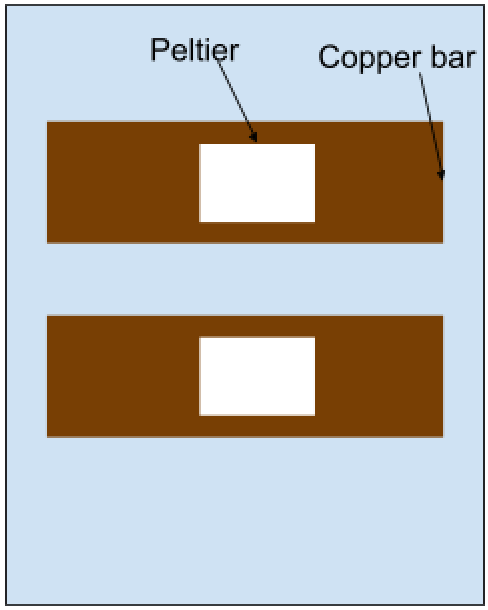


Figure 38: Final Peltier Layout

Power Distribution:

Looking at power distribution, the only detail that needed to be reconsidered was how the power distribution needed to be altered when switching from a breadboard circuit to a perfboard circuit. In particular, one of the inconveniences associated with the soldering onto a perfboard was the need to prevent high currents from flowing through the thin copper connections that usually serve as rails on perfboards. Because of this, we needed to make sure that only low current drawing systems were connected to the rails, while high current systems were directly connected to each other and the power supply.

Case Modification:

In order for our system to properly function we needed to make several modifications to the case. This included adding external fans, enlarging holes to enhance the fan's airflow, making space for the touch screen in the lid, and altering the rubber stoppers. We received nuts and bolts to secure our fans in place from Mr. Hooker, then we submitted a Solidworks drawing of our proposed modifications of the case to the machine shop (Fig. 39). These changes were necessary in order to properly utilize the airflow the fans could provide. Since most of the airflow that fans provide occurs along the edges of the fan blades, and our fans were significantly larger than the holes on the case, we had to enlarge the fan holes and add four holes to bolt the fans in place.

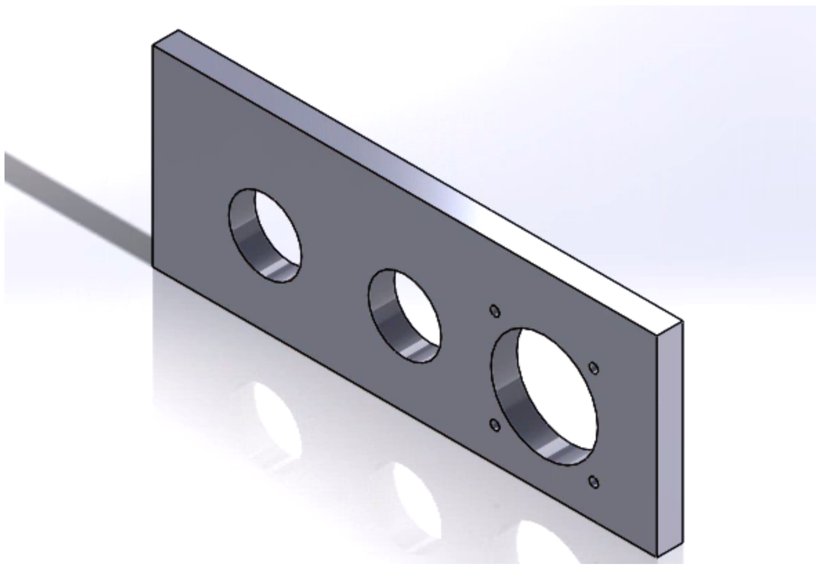


Figure 39: Solidworks Case Model

We also had to cut a hole in the top of the case for the touchscreen to fit inside of. The result of these alterations can be seen in Figure 40. Lastly, we made incisions in the rubber stoppers on the case to hold the cable more securely and prevent the wires being pulled out.

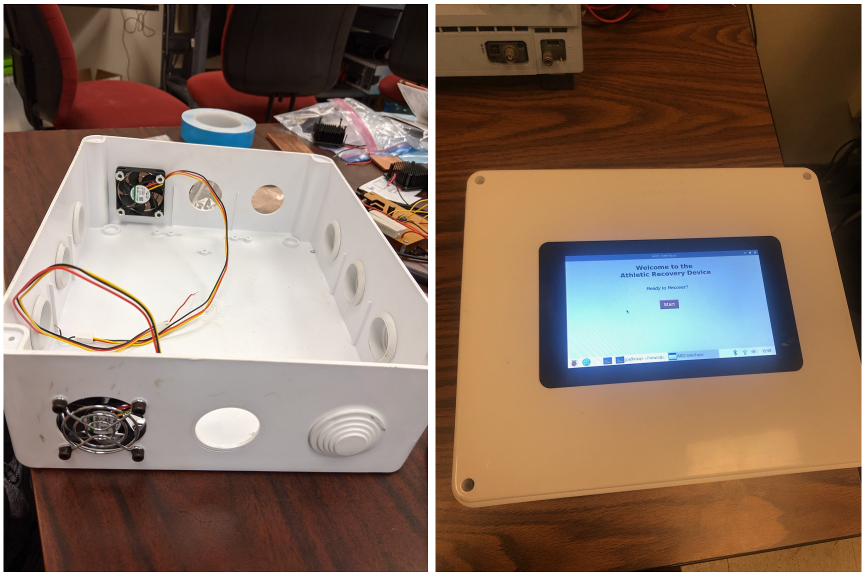


Figure 40: Case and Touch Screen

Sleeve Electronics:

One of the biggest challenges throughout this project was figuring out how to hold all the sleeve electronics together. In the preliminary design, we thought thermal tape would be sufficient. Unfortunately, we discovered through testing that it did not provide a strong enough thermal connection to be used on the Peltier units or the copper bars. As a result, we figured out an alternative to the thermal tape to securely hold the bar, Peltier unit, and heatsink together--rubber bands. Specifically, we removed the fans from the heat sink, threaded the rubber bands in the crevices of the heatsink, twisting them 3 times around the heatsink and copper bar, with the Peltier sandwiched in the middle (Fig. 41). This method has several benefits. Firstly, it allowed us to use thermal paste instead of thermal tape to ensure good thermal connection between the Peltier and copper bar and the Peltier and heatsink. Secondly, since rubber is a terrible thermal conductor, even if it is in contact with the bar and heatsink, it will have little effect on the thermal function of the system. Thirdly, and possibly most importantly, due to the elasticity of rubber, there is constant strong force being applied to the Peltier on both sides, further ensuring good thermal contact.

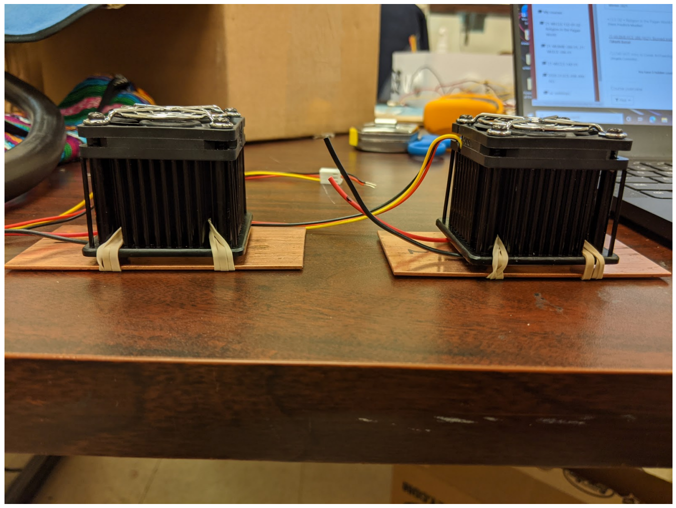


Figure 41: Rubber Band Securing Mechanism

The thermistors also needed good thermal contact with the top of the copper bar, where we placed them. In the final design, the thermistors were coated in thermal paste before being secured to the copper bar, adjacent to the Peltier unit, with several layers of electric and thermal tape. We decided to do this because placing them on the underside of the copper bar may have further reduced the thermal contact the copper bars had with the skin by making the copper bars even more uneven and requiring a layer of electric and thermal tape, which would have acted as insulators. This method of attachment also serves to secure the copper bars in place. While we did want to measure the temperature on the surface of the skin, we decided it would be too impractical to try and have the thermistors directly on the boundary between the skin and sleeve, as this would require wires outside the sleeve and some pouch for attachment, which we did not have time to construct.

<u>Layout of Electronics:</u>

Due to the very tight fit between the case wall and power supply outputs, we placed a terminal block on top of the power supply, then connected the outputs of the power supply to the terminal block (Fig. 42).

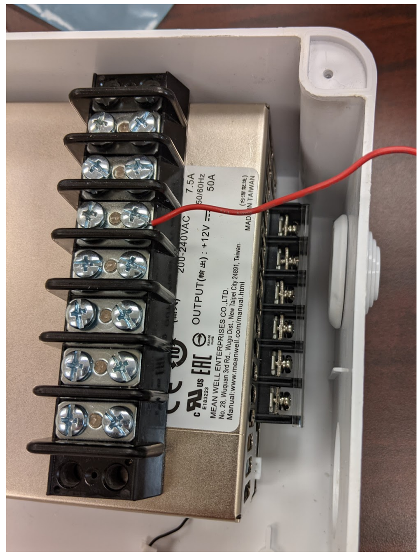


Figure 42: Power Supply and Terminal Block

This made it so that we did not have to make all the connections to the electronics and power supply before we put everything into the case, which facilitated construction. That being said, it needs to be noted that we used multiple wire connections between each power supply output and terminal. If we had used single wires to connect outputs of the power supply to the terminal block, when we attached multiple high current wires to the terminal block, we would have run the risk of damaging the wire.

We placed the power supply in the case and fed the wall outlet power supply cable through one of the openings, putting a rubber stopper in the opening of the case as a make-shift strain relief. Then, we fed the case-sleeve cable through one of the openings, while simultaneously putting all the connected electronics in the case, before attaching a rubber stopper to the sleeve side case hole. As far as the layout of the subsystems in the case, the Peltier drivers and thermistor amplifier circuit were put on the bottom of the case while the buck converter and the Raspberry Pi with its perfboard containing the ADC and DACs rested on top of the power supply (Fig. 43). In order to ensure isolation between electronics and the power supply, we put a layer of electric tape on top of the areas of the power supply in contact with these components.

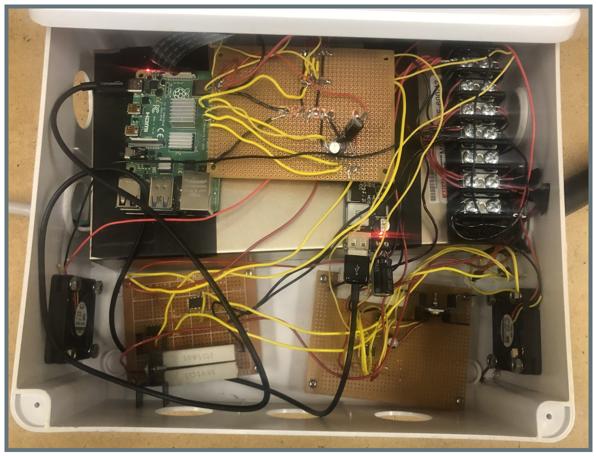


Figure 43: Layout of Electronics

Orientation and layout played a crucial role when cooling down the internal components of our system. Initially, we had the fans set up so they would use a combination of positive and negative pressure to cool the case. In other words, they were blowing in the same direction. However this resulted in one of MOSFETs that was more centered in the case burning out. The air being directed towards this MOSFET had already been blown past the other MOSFET, which was itself very hot, such that that air was significantly hotter and less effective. As a result, we made several changes. These included changing the orientation of the fans so they were both blowing inwards while opening the side holes to let out warm air, adding on a second heatsink to each MOSFET and placing each MOSFET closer to the fans.

<u>*Connections and cable:*</u>

For the case electronics, we began to shorten all the long wires and make the needed power and ground connections to the buck converter and power supply. Due to the numerous wires, we were finding it difficult to get all the individual wires into power and ground on the terminal block. There simply was not enough room underneath the terminal block's screws. To counter this, we twisted the various power wires together and soldered them together, with only one wire of the bunch getting screwed in, which made screwing them into the terminal block much easier. We repeated this with the ground wires.

For the sleeve, some other important steps we took included trimming and stripping the wires of the pelter, fans, and sleeve cables', as well as making a table of corresponding connection between the sleeve electronics and the various colored cable wires (Table 3). We then soldered all the connections between the cable and sleeve electronics and covered the exposed wires with electrical tape. In addition, since the cable wire was initially intended for five subsystems, there were numerous unused wires which we removed (Fig. 44).

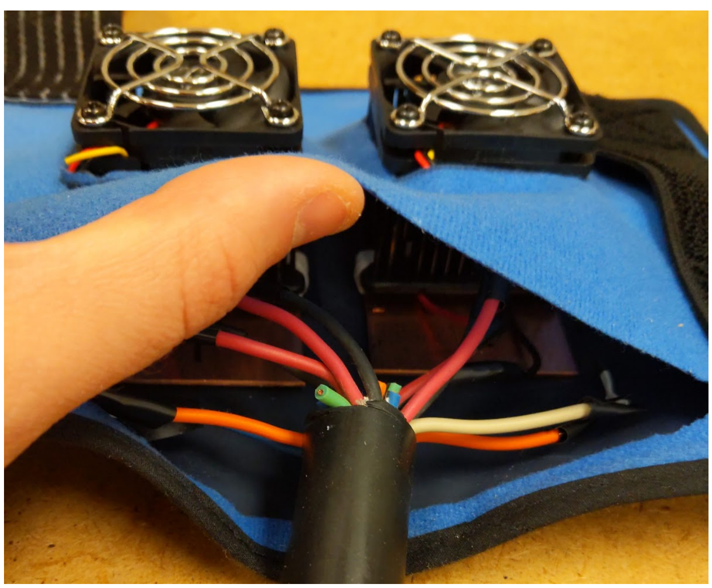


Figure 44: Sleeve Electronics and Cable Connections

Lastly we applied several layers of electric tape in order to secure the cable to the sleeve and to close the sleeve.

System wire	Cable wire color
Peltier 1 +	Red wire yellow stripe
Peltier 1 -	Black wire yellow stripe
Fan 1 +	Red wire 1
Fan 1 -	Black wire 1
Thermistor 1 +	Orange wire red Stripe
Thermistor 1 -	Blue wire red stripe
Pelter 2 +	Red wire black stripe
Peltier 2-	Black wire red stripe
Fan 2 +	Red wire 2
Fan 2 -	Black wire 2
Thermistor 2 +	Orange with green stripe
Thermistor 2 -	White wire v

Table 3: Sleeve Electronics and Cable Connections

Sleeve:

We were unable to find a pre-existing empty sleeve that would meet our criteria. As a result, we decided to settle on ordering a large reusable gel pack, which we would adjust to fit our needs. In the end we selected the Flexikold gel pack seen in Figure 45 below.

Figure 45: Flexikold Gel Pack [43]

We selected this one since it was large enough to hold our electronics and had a side made from nylon, which made it easy to clean, and thus satisfied the sanitation requirement. From here, we made a small incision and emptied all the gel into the garbage, which was nonhazardous material. Then we rinsed it out several times and dried it before we cut out spaces for the heatsinks/fans, as well as an entrance to fit all the electronics and cable wire. This resulted in producing the sleeve scene in Figure 46.

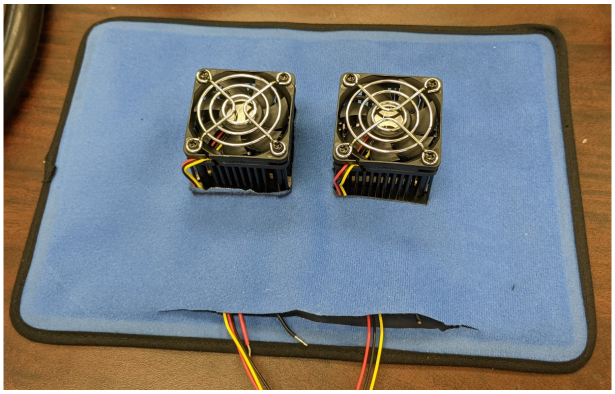


Figure 46: Completed Sleeve

Lastly, we developed a viable velcro strapping mechanism for the makeshift sleeve with the velcro straps we hand on hand. Specifically, we used one of the smaller velcro scraps to make a functioning buckle that we could feed the other velcro strap through. An image of this strap can be seen below in Figure 47.

Figure 47: Velcro Strapping System

Current sensor:

Initially, we thought that having a current sensor at the end of each Peltier unit, set to monitor the current and shut off the system should something go wrong, would be a useful function. However, we came to realize that if there was a short in the system, this method would be ineffective, as such a short by its very nature would bypass the MOSFET, our only method of controlling the current draw. As it stood, we thought that redesigning the system to incorporate the current sensor effectively into our system would be too time consuming and complicated, and thus cut it from the final design.

Information Processing:

Going off of the preliminary design, we had a good starting point to develop the software components of the final design. Figure 21 shows an initial outline of the signals being analyzed during the cooling treatment. Figure 48 gives a more in-depth look at the inputs and outputs at each stage of the process.

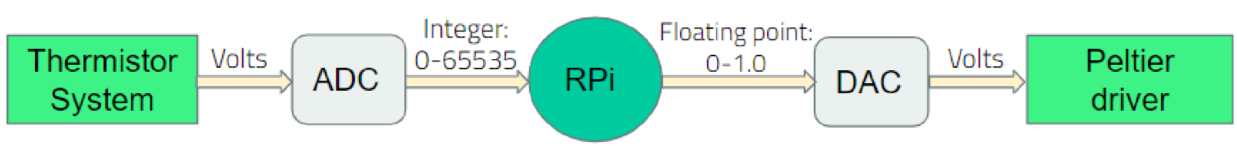


Figure 48: Final Signals Block Diagram

Not only did the Raspberry Pi have to communicate with other chips to facilitate the exchange of information between the thermistor and Peltier units, but it also needed to display a user-friendly interface and calculate the control algorithm to produce the correct output to the Peltier driver. The implementation for these sections of the interface are explained further in the coming sections.

Signal Processing

Given that the first half of the block diagram was accomplished in the preliminary design, the initial objective for signal processing was to be able to drive the Peltier units at a variable current, not only at its maximum and minimum value. As mentioned earlier, since the Raspberry Pi only uses digital inputs and outputs, an external chip, in this case the MCP4725, is used to give an analog signal to the Peltier driver system. This 12-bit digital-to-analog converter (DAC) uses I2C serial communication to interact with the Raspberry Pi. Unlike the ADC chip, the DAC has only one output per board. Therefore, another DAC object is needed in both hardware and software. The wiring diagram for the DACs is shown on the right side of the complete circuit diagram for the information processing system below (Fig. 49).

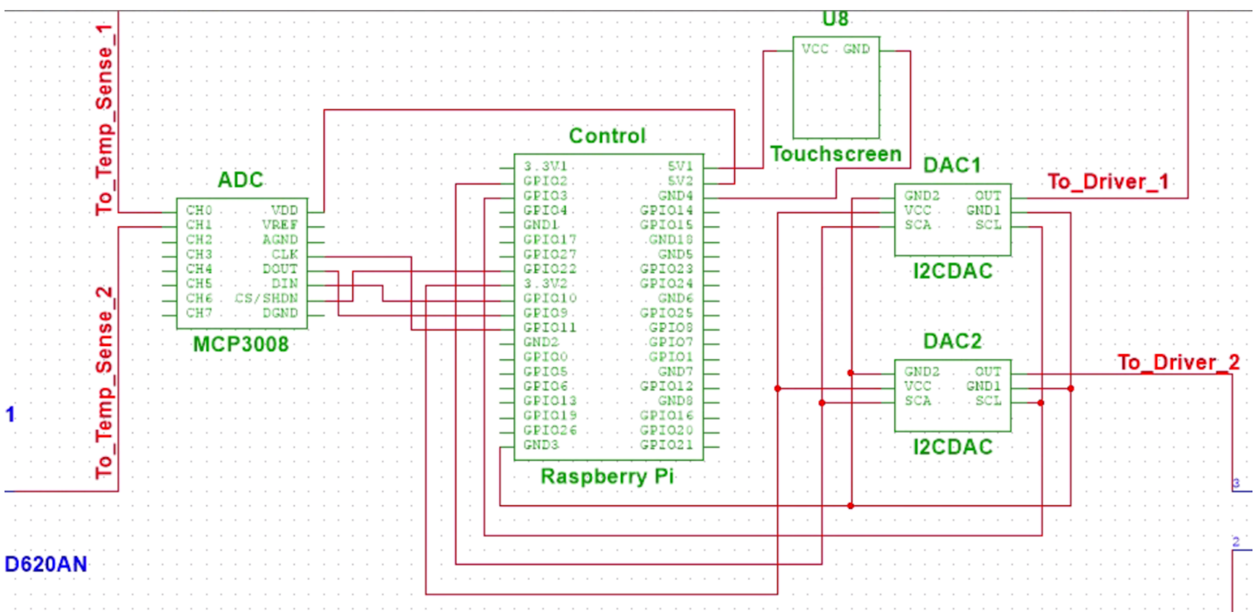


Figure 49: Circuit Diagram of Information Processing Subsystem

The physical wiring of the second DAC will be the same as the first, with the exception of two changes: i2c address and i2c internal pullup resistors. As explained in the MCP4725 documentation, there is a jumper pad with a GND (ground), ADDR (address), and VCC (voltage at common collector) [44]. By soldering the middle pad to either the top or the bottom pad, the i2c address can be changed (Fig. 50).

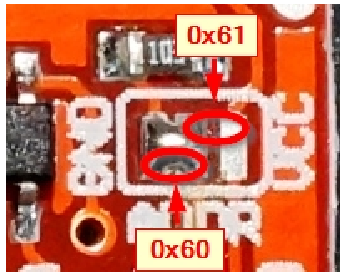


Figure 50: i2c Address Jumper on MCP4725 [45]

For more than one MCP4725 on a bus, it is necessary to disable the pull-up resistors on one of the boards. These are located on the back of the board, which we separated using an X-acto knife (Fig. 51).

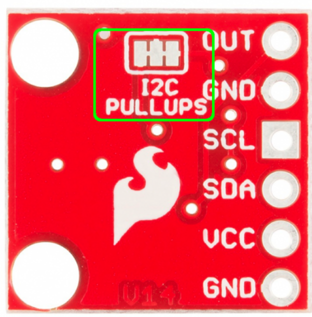


Figure 51: i2c Pullups Jumper on MCP4725 [45]

Regarding the software, the i2c address for each board is specified using: `dac = DAC.MCP4725(i2c, address=0x60)` or `dac2 = DAC.MCP4725(i2c, address=0x61)`. Running the command `i2cdetect -y 1` confirms the i2c addresses on the RPi terminal as 60 and 61. In order to get an output signal from this DAC chip, the function normalized_value() is called on each DAC object, which takes a floating point value that it will scale to an analog output voltage from 0V to 3.3V. From our measurements, the actual high value of the output pin (measured from the multimeter) from the RPi is 3.55V. Therefore, as documented in the pi_output.py, located in the development files folder in the GitHub repository for this project, the command for setting the output voltage is formatted as such `dac(2).normalized_value = VOLTAGE/3.55`, where the `VOLTAGE` variable represents the desired voltage output of the DAC as a floating point. This functionality will allow us to drive the Peltiers at a variable current by outputting a voltage ranging from 0 to 2.5V to the Peltier drivers circuit.

Graphical User Interface

As explained earlier, we chose to use the Tkinter package, which is the standard Python interface to the Tk GUI toolkit, to implement the user interface. On the application level, Tkinter works by layering frames on top of each other, which are raised to the top to be visible once that frame is called. Given that the athletic trainer communicated that pre-programmable treatment modes would be a convenient function, we divided the interface into six frames: a home page, a select mode page, a password locked page, a custom mode page, a confirm treatment page, and a treatment summary page. Another component of Tkinter that allows for the addition of objects to the frames are widgets. These can be “Labels”, which represent text, “Buttons” which execute commands once pressed, “Entry” boxes, which save the user input, etc.

Populating the frames with widgets and layering them to create a functioning application was simple: frame objects are created within the application, and widgets are defined within the frame to create the appearance of the interface. For example, a code snippet for the start page is shown below (Fig. 52).

```
class StartPage(Frame):

    def __init__(self, parent, controller):
        Frame.__init__(self, parent)
        title1 = Label(self, text="Welcome to the \nAthletic Recovery
Device", font= "Verdana 20 bold")
```

```
        title2 = Label(self, text="Ready to Recover?", fg="navy",
font='Helvetica 17 italic')
        title1.pack(pady=23, padx=10)
        title2.pack(pady = 20)

        startButton = Button(self, text="Start", bg='#8B0000', fg='#ffffff',
font='Helvetica 15 bold', borderwidth=3, command=lambda:
controller.show_frame(Modes))
        startButton.pack(padx = 0, pady = 20)
```

Figure 52: Sample Frame Implementation

A sample interaction between the user and the interface is shown in Appendix 4; the code used to create these frames are contained in the file tk_interface.py on the seniorcapstone GitHub repository [41].

To add some functionalities to the screens, we needed to learn how to carry parameters into different frames, such as the preset mode temperature and duration values into the display of the "Begin Treatment" screen (Frame #3: Confirmation). From the Python documentation about Tkinter, we learned that "in the current implementation of tkinter it is not possible to hand over an arbitrary Python variable to a widget through a "variable" or "textvariable" option. Since Tkinter windows are created by layering the frames on top of the other, the values in the Confirm frame are loaded with the empty initialized value, not with the temperature and duration values of the selected mode or custom entry box. The only kinds of variables for which this works are variables that are subclassed from a class called Variable, defined in tkinter" [46]. The solution we implemented was to create individual temperature and duration StringVar() objects in the ARDapp class, which we could call to set the value of (after the submit button press in the Custom frame) and set variables equal to (the desired temperature label in the Confirm frame). A key difference is that these built in variable objects have get() and set() methods that can be used to be tracked and updated by widgets. Therefore, the use of the built-in StringVar() object allowed for key variables to be dynamically updated across frames.

Another important function of the user interface is to inform the user of the current temperature of the gel pack during cold therapy treatment. We decided to display the temperature as an average between the two thermistors, since the temperature difference between the two isn't ever significant, and the average would be a simple value to understand, rather than two individual thermistor values. From the control_final.py file, where the thermistor reading and PID loop is contained, a function was created to calculate and return the average temperature reading of the two thermistors. In the interface file, similar to the way the desired temperature and duration is set, an attribute of the application, called self.average, is defined and assigned as a StringVar(). This allows the program to obtain a dynamically updated version of the variable when it is called in the frame. Finally, a function is initialized in the ARDapp application object to get the average value from the function in the control file and set the self.average value. This ability is showcased in the Confirmation frame once the "Begin" button is clicked.

PID Control Loop Theory

In order for the Raspberry Pi to adjust the Peltier units' temperature based on the desired temperature, a control signal algorithm is needed to calculate the appropriate voltage output of the DAC. A very popular form control loop mechanism is a PID (proportional, integral, derivative) controller. These three terms all act in different ways and have different effects on the system, dependent on the feedback signal given by the "plant." For a temperature control system, the difference is taken between the desired temperature and the measured temperature (plant output)--this produces an error value that drives the PI terms.

Proportional control is the most common and simplest implementation of a control loop, in which a constant gain is multiplied by the error value to drive the control signal [47]. A lower gain value will reach the target, slowly, while a large gain will reach the desired target sooner, at the cost of initially overshooting it (Fig. 53).

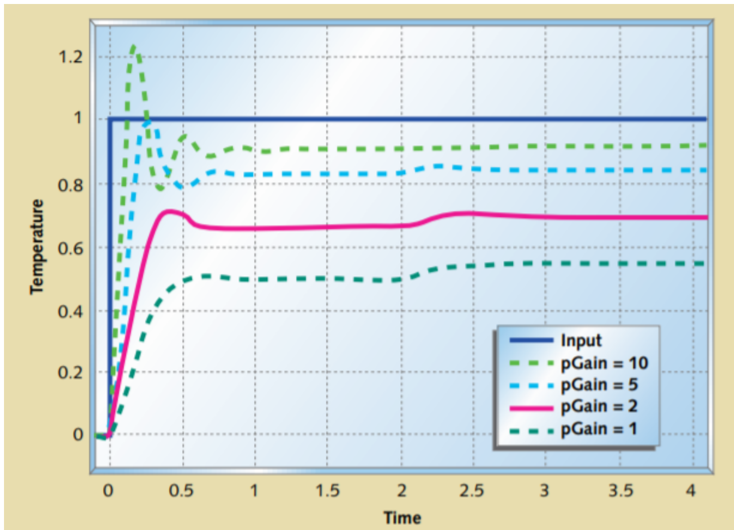


Figure 53: Proportional Control on a Thermal System [47]

For our situation, time it takes to cool is a metric we would like to keep low, so a large proportional gain is expected (note that increasing the gain by too much will not decrease the cooling time more, but will increase the oscillations around the target value--called ringing). There is also not much risk in having the sleeve be slightly colder than the desired value because the system is not capable of producing dangerously cold temperatures. It has been shown that solely proportional feedback does not ever reach the desired temperature in a thermal system; therefore, integral control is used, mostly in conjunction with proportional control, to add long term precision and decrease the steady state error [47].

An integral system remembers the previous errors and compensates--which also leads to instability (that can be compensated for using the “present” error term, or the proportional value). Adding the proportional term to the integral system will decrease the time it takes to reach the desired temperature (Fig. 54).

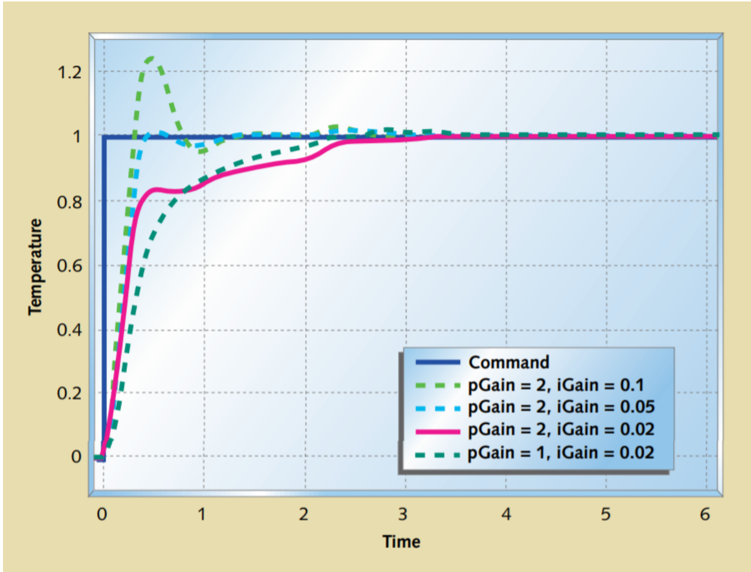


Figure 54: PI Control on a Thermal System [47]

Adding a differential term will stabilize the plant by predicting the future behavior of the measured feedback value [47]. As you would expect from a "differential" term, this looks at the velocity of the change given by the difference between the last value and the current value of the position. Along with differential control comes three potential problems: sampling irregularities, noise, and high frequency oscillations. Given these potential difficulties, we will stick to PI control unless it becomes necessary to stabilize the system, since using a non-zero differential gain will amplify the noise of our system.

PI Control Implementation

Once familiar with the theory of a control loop, we began to implement the algorithm in our code. In Python, we developed a control file to regulate the input voltage to the Peltier system depending on the thermistor output (Appendix 3). In this file, the thermistor value is read, converted to a temperature, and fed to the PI control. In the control loop, the difference is taken between the target and measured thermistor temperature in degrees Fahrenheit. An adjusted system output temperature, depending on the values of the proportional and integral gains, is produced from this system. This represents the voltage at which to feed to the Peltier driver circuit which at steady state will produce the temperature matching the PI control system output. To convert from the output in degrees Fahrenheit from the PI output to the voltage input to the Peltier drive, we decided to use a linear equation that maps the desired output temperature to corresponding voltage that would drive the Peltier to that temperature, as discussed earlier. The block diagram for this control loop mechanism for our given system is shown in Figure 55 below, where ∝u(t) represents the proportional constant at which the temperature is multiplied by to obtain the voltage value to output from the DAC.

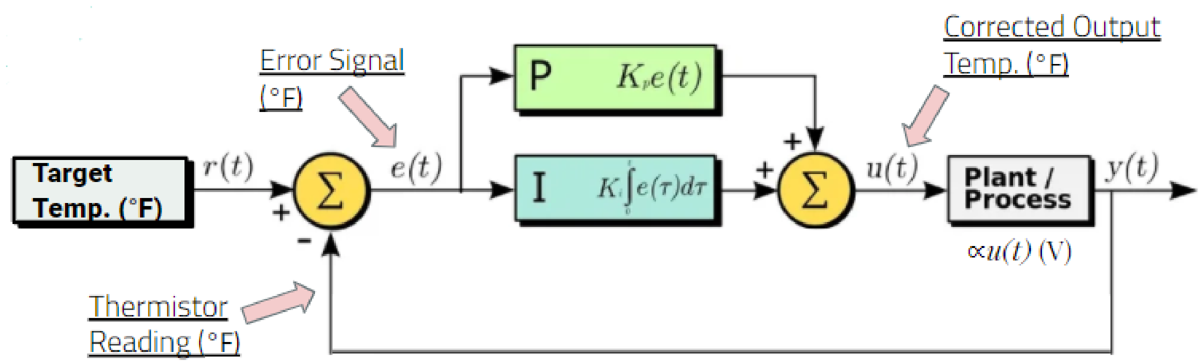


Figure 55: PI Controller Block Diagram

The control file also sets the target temperature of the PID controller objects and prints a graph of the thermistor values over time after resetting the DAC output voltages to 0V. The code for the PID was sourced from an existing file on Git [48]; all of these files for the PID system, control, and tests, are contained in our main GitHub repository [41].

To test this control system, we began to tune the PID gain values using the Ziegler-Nichols PID tuning method [49]. Initially, we set out to find the value that the system oscillates given a constant K_P, proportional gain, and values of 0 for the integral and derivative gains, K_I and K_D; this is called the ultimate gain, K_U. Given that the final system was not assembled at the initial tuning, we tested the system with a thermistor attached firmly to the cold side of a Peltier unit with thermal paste. At a low proportional gain, the system was unable to increase its temperature to reach the target after an overshoot. As seen in Figure 56, the system would initially decrease very sharply, then stay far below the target temperature of 50 degrees.

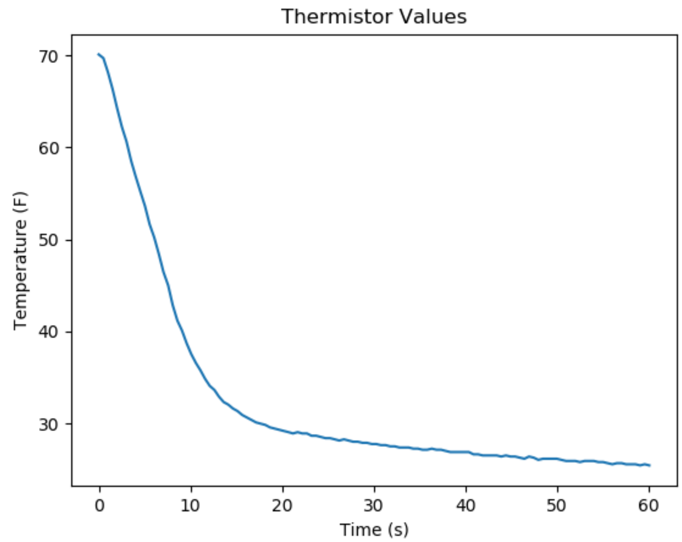


Figure 56: Tuning Single Thermistor System (Thermistor Temp.)

Looking at the plot of the PID values, we saw that the system was approaching an output voltage of about 1.6V (Fig. 57).

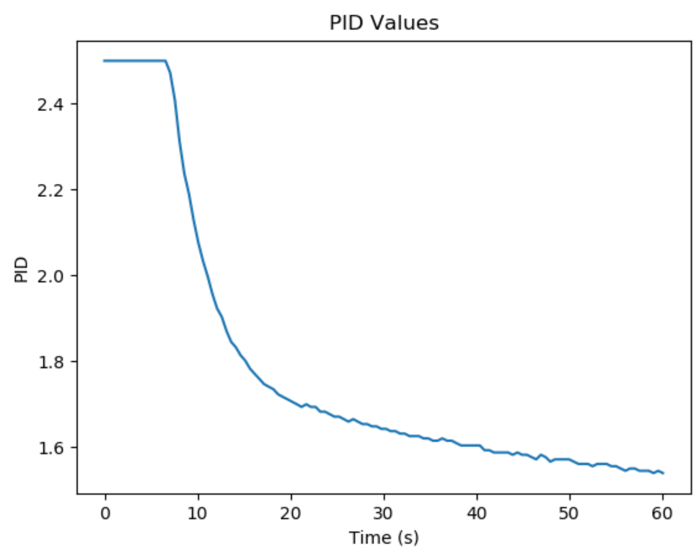


Figure 57: Tuning Single Thermistor System (PID output voltage)

Reversing the equation, we found that this target output temperature from the system was not above the target temperature. Feeding the current numbers through our PID system, we came to the conclusion that this was because our proportional gain, K_P, was too low. After a few tests of increasing the proportional gain by a factor of 2, we reached our goal of an oscillating, unstable system (Fig. 58).

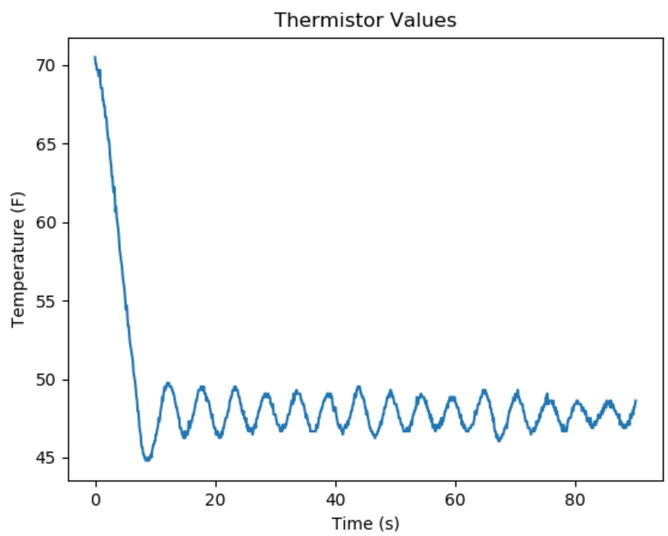


Figure 58: Tuning Single Thermistor System: Oscillating at $K_P = 24$

After our device was assembled, we began to tune the PID control system to find the optimal proportional and integral gain constants that would implement safe, controlled cold therapy treatment. Ideally, the system would reach the desired temperature quickly with a small overshoot, a low settling time, and maintain that temperature within a few degrees for the length of the treatment.

For our final set-up, each of the two thermistors had its own PI control loop, which had to be tuned according to the thermal behavior of the thermistors' final orientation. Once again, in order to determine the values for K_p and K_I, we initially needed to determine the ultimate gain,

K_U. This was found by increasing K_P, while K_I and K_D were kept at zero until we observed constant, steady oscillations (Fig. 59).

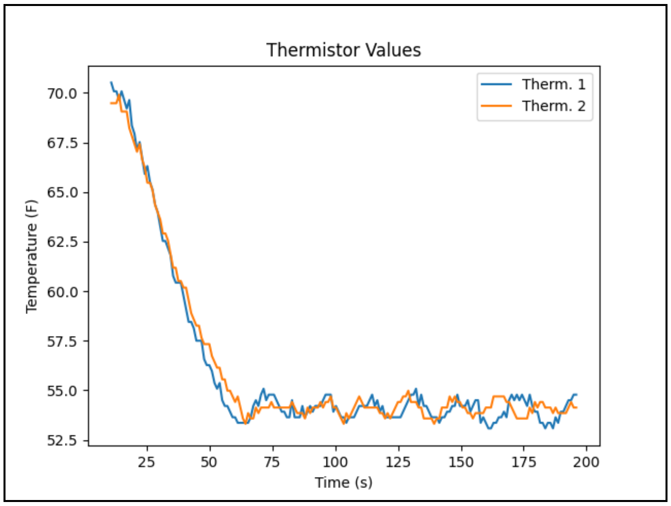


Figure 59: Finding Ultimate Gain K_U of Final Device

For the test above, the target temperature of the system was set to 55 degrees, and the treatment duration was set for three minutes. Though we observed slight performance differences resulting in out-of-sync oscillations, we settled on an ultimate gain, K_U, of 63. From here, we would also determine the period of the oscillation, before using the Ziegler-Nichols method to determine the exact values of K_P and K_I (Table 4). Our calculations, based on the peak-to-peak waveforms of the oscillations, yielded an oscillation period, T_U, of 23 seconds.

Ziegler–Nichols method

Control Type	K_p	K_i	K_d
P	$0.50K_u$	—	—
PI	$0.45K_u$	$0.54K_u/T_u$	—
PID	$0.60K_u$	$1.2K_u/T_u$	$3K_uT_u/40$

Table 4: Ziegler-Nichols PID Controller Tuning

Due to inconsistencies between tests, we guess that we only know K_U within 10%. This may have been even more inaccurate, given that we calibrated the system when it was just sitting on a table, not strapped onto a body part. This means the system was not calibrated for its intended loaded use. Furthermore, we could not seek a clear pattern between oscillations in various tests, so we ended up estimating and assuming that there is up to a 25% error in the oscillation period. After noting all this, we used the table above to settle on a proportional gain, $K_P = 28.35$, integral gain, $K_I = 1.48$, and derivative gain, $K_D = 0$; we tested these values for three minutes at a target temperature of 55 degrees (Fig. 60).

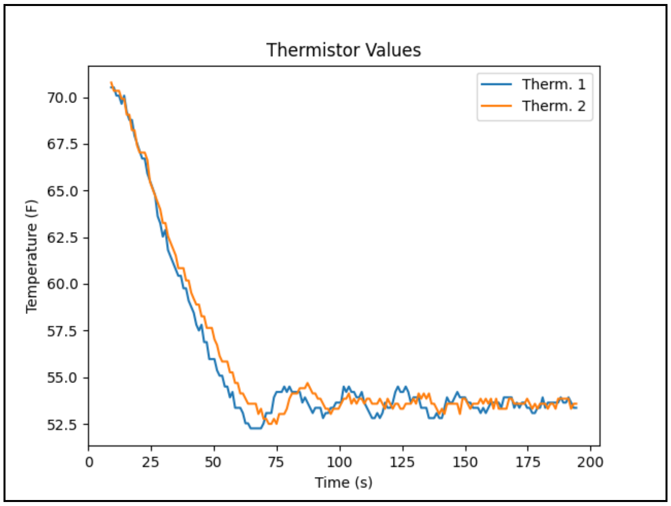


Figure 60: Final Tuning of Assembled System

This test demonstrated that the device could reach the desired temperature range rather quickly (about 60-75 seconds) and maintain that temperature with a decreasing amplitude of oscillations as time went on. But even after this calibration, we noticed that the actual temperature was staying rather reliably 1.5°F below where we expected. To compensate for this, we had the system actually aim 1.5°F higher than we wanted and this fixed the issue.

Performance Estimates and Results:

Estimates:

While we had a solid set of design constraints and a rough idea of how our device would behave, we struggled in knowing how our device would perform and if these goals were achievable. What follows is an exhaustive list of our estimations of how our project would function before we actually began the construction process. We estimated the project would cost about $1,200 in material components alone, a complete breakdown of which is available in the appendix. Our device was thought to be capable of wrapping between 180º and 360º around the body part undergoing therapy. We believed our project would be able to cover 500 cm^2 of area on the human body, as each copper bar, of which there would be 5, could cover 100cm^2. We also expected our system to be capable of delivering 52W of thermal transfer away from the body. At our expected conditions of 50ºC across the Peltier units, each of the 5 subsystems would contain 2 Peltier coolers, each pushing 7W through themselves. If we assume that a third of this is wasted to cool the air around the device, this gives us 5 systems, each drawing 9.3W of power away from the body. We determined that each Peltier would push 7W from the datasheet of the TEC1-4095, which indicated a maximum current and a ΔT of 50ºC, each Peltier would push around 7W. We arrived at 50ºC as follows:

Loading…

Loading…

Loading…

Where ΔT is the temperature gradient across the Peltiers in ºC, T_{hot} is the hot side of the Peltiers, T_{cold} is the cold side of the Peltiers, $T_{ambient}$ is the ambient temperature, P_{pelt} is the total power the

Peltiers are consuming in W, $R_{th,sink}$ is the thermal resistance of the heat sinks in °C/W, T_{body} is the temperature at the surface of the body, ΔT_{body} is the temperature difference between the body's normal temperature and the temperature we are inducing, $R_{th,hel}$ is the thermal resistance of the gel pack, and $R_{th,body}$ is the thermal resistance between the surface of the body and where the body returns to its normal temperature. This was evaluated as follows:

Loading…

Loading…

Loading…

We also expected our system to be able to reach its desired temperature in under 5 minutes and maintain its temperature within two degrees Celcius. We believed the copper bars would evenly distribute the temperature to the surface of the gel pack in contact with the injured appendage. We thought it would draw 30A, 5A for each of the 5 Peltier subsystems, 3A for the Raspberry Pi, and 2A for all the fans. At 12V, that would mean our system would consume 360W at maximum operating conditions. We also expected the user interface to be easy to use, the sleeve to fit easily on the body part being treated and be easy to clean.

Results:

Of course, a lot of these expectations proved difficult to reach once we started to construct our device. In the end, we were only able to construct 2 out of our 5 planned systems due to time constraints. On the upside, in the end, our total material cost was much lower than we had anticipated, totaling $618.23. While this was mostly because of the reduced scale of our final design, we were also able to find the heatsinks, which were some of the most expensive items on our parts list, on sale for less than a third of their listed price. This drastically reduced the cost as well.

The success of our device is most apparent in the ten minute treatment test shown below in Figure 61.

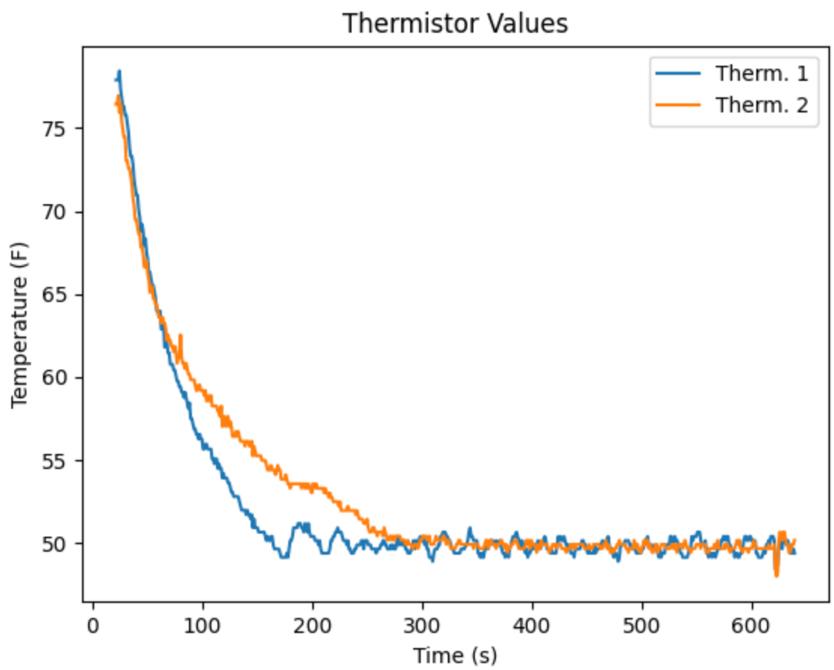


Figure 61: Ten Minute Test on Herschel at 50ºF

For this test, the device was attached to Herschel's leg and set to 50ºF for 10 minutes. As you can see, once the device got to desired temperature range, it was able to maintain the target temperature within a degree Fahrenheit. You may also notice that one Peltier cools down significantly faster than the other. This is likely because of imperfections in how much thermal paste we applied to the Peltiers. The slight drop in effectiveness could be due to the thermal contact simply not being good enough on the Peltier represented in orange. It could also be explained by the thermistor's contact with the copper bar not being as good. The fix for this would have been simple, reapplying the thermal paste and thermistor, but since this was inconvenient to access, we would have needed more time to actually implement it. This discrepancy is even more apparent in Figure nioa, where the device was on Ryan's arm and set to 45ºF.

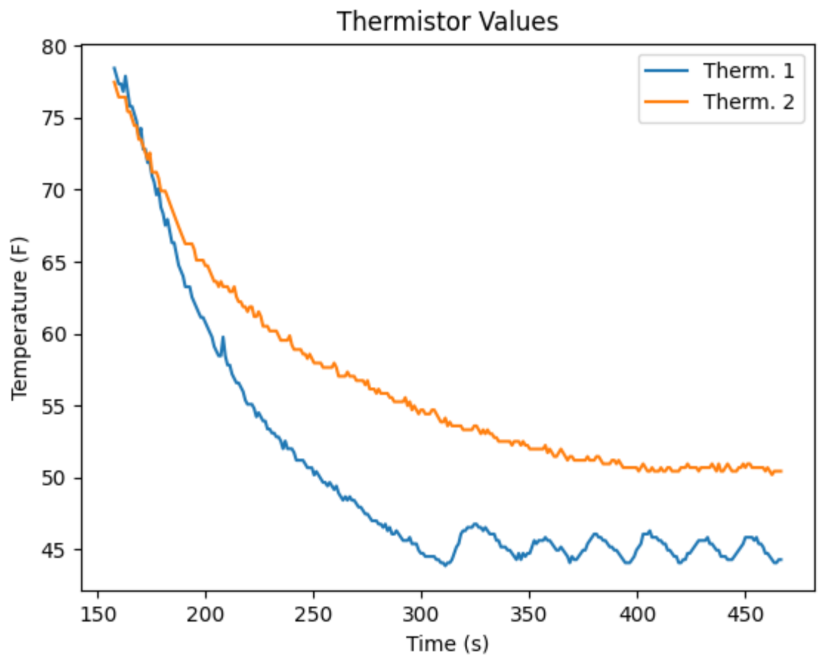


Figure 62: Five Minute Test at 45ºF on Ryan's Arm

The blue subsystem reaches the proper temperature and oscillates around it, but the orange subsystem is unable to reach the desired temperature. Again, we know how we would attempt to solve this, but considering our time constraints, this was not a prioritized adjustment.

Our device was also able to run continuously, or near continuously, for long periods of time. During our testing, the system would be on intermittently for multiple hours without noticeable degradation or overheating. We also tested the system continuously for thirty minutes, shown below in Figure 63.

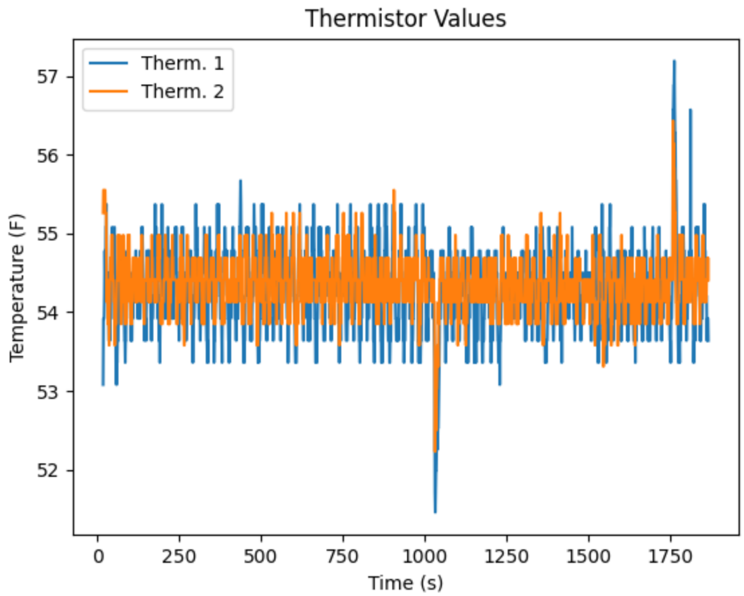


Figure 63: Thirty Minute Unloaded Test

This was before the temperature control mechanism worked as well as it did in its final implementation, so the center of the oscillations was slightly lower than the target temperature, but the temperature regulation was still very consistent. We also noticed that the copper bars are properly distributing the temperature across the area being cooled. We know this because the skin being cooled was red underneath the copper bar, to the extent that the slight shape of the bar was visible.

Ultimately, while the minimum temperatures of each of our subsystems were within the temperature range goal we had set for ourselves, neither could get all the way down to 4ºC (40ºF) as we had desired. This means the device was not delivering the amount of wattage per subsystem that we were expecting. There are a number of differences between our expected conditions and the conditions our device actually operated under. Firstly, the gel pack we mentioned in our design ended up having a much smaller thermal resistance than we were anticipating. In our estimation we assumed the gel pack would be full, but in the final design we had emptied it. Given this, we would expect our device to have overperformed our expectations, which it did not.

There are many possible reasons for this. The most likely is that there was simply much more cooling power lost to the air around the device than we were anticipating. Another possible culprit is the lack of a solid thermal connection between the Peltier units and their surroundings. This effect may or may not have been present in the subsystem that reached 45°F, but it was most certainly present in the device which could only make it to 50°F. There also could have been subpar thermal contact between the copper bar and the thermistors, which would have also caused our system to appear to underperform, as our measurements would have been warmer than the reality. Another possibility is that the human arm is not accurately modeled in the way we were trying to model it, ie. as a 2-D rectangle of skin and muscle with a reservoir of fixed temperature (30°C) below it. This is particularly possible considering that blood will be actively flowing through the body through these areas constantly. Overall, we consider the 4-10°C goal to be only partially fulfilled.

As for achieving a 180° to 360° coverage, we were also unable to meet this expectation, simply because of a lack of time to build the full planned system. For a similar reason, we were unable to meet the expectation of covering 500cm^2, although each module of the device was able to cover 100cm^2, as was expected. The device was able to reach the desired temperature quickly, always within 5 minutes, averaging about 1-2 minutes. Obviously, we implemented less subsystems, the device drew less current than we were expecting, drawing under 14A total. This was 5A for each subsystem, 3A for the Raspberry Pi, and less than 1A for all the fans. Because we actually ran our device at closer to 13.5V, this means the total power consumption of our device was less than 190W. Overall, the feedback we received from our customer indicated that the user interface was easy to use and the material of the device was easy to clean.

FDA and Standards:

If we intended to take our device to market, we would first have to present it to the Food and Drug Administration (FDA). In particular, we would need to get it through the Center for Device and Radiological Health (CDRH). First, we would need to determine the classification of our device. Next, we would have to determine whether to apply for FDA Clearance or Approval depending on the class of our device. Lastly, we would need to submit relevant standards with our application to CDRH.

Classification of a device is based on its associated risk; low risk devices are Class 1, medium risk devices are Class 2, and high risk devices are Class 3 [50]. We determined our device to be classified as a Class 2, although we would need to submit a classification request to the FDA to know for sure. Since we are dealing with a high amount of current, we did not think that we classify as a low risk device (Class 1), and at the same time, we were not supporting life functions, so we were not a high risk device (Class 3). Furthermore, when looking on the FDA website under medical devices, we found that the device most similar to ours was a powered heating pad, which was graded as a Class 2 device. Such a device is defined as: “A powered heating pad is an electrical device intended for medical purposes that provides dry heat therapy for body surfaces. It is capable of maintaining an elevated temperature during use”[51]. While not exact, our device is an electrical device intended for medical purposes, but instead it provides cold therapy to the body’s surface.

According to the FDA, powered heating devices are exempt from needing a 510(k)[52]. If the FDA decides that our device is similar enough to a powered heating pad, since it could essentially act as one if the current is run in reverse, we may not need to get any clearance before we bring our device to market.

At the same time, since we could not specifically find any device that has the same function as ours, it could very well be that our device is not substantially equivalent to any other device, therefore we would have to apply for a De Novo clearance. This is essentially applying for a tradition clearance with the added step of applying for a De Novo classification. Before applying for a De Novo Clearance, we would have to apply for a De Novo Classification, where the FDA would classify our device into Class 1, 2, or 3[53].

In order to have a better chance of getting the clearance, we would also submit some standards we thought were relevant and could support our application. Some examples may include the ISO59752 [54], which pertains to medical device quality management, and ISO9241 [55], which is on ergonomics of human-system interactions.

Production Schedule:

While it is important to try to arrange and schedule how a project will progress, oftentimes adjustments and setbacks derail careful planning. What is important is to reflect on what improvements could be made to the schedule to have the production go smoother. Below are the intended schedules for the fall and winter terms (Fig. 64 and Fig. 65).

Figure 64: ECE498 Fall Term Plan

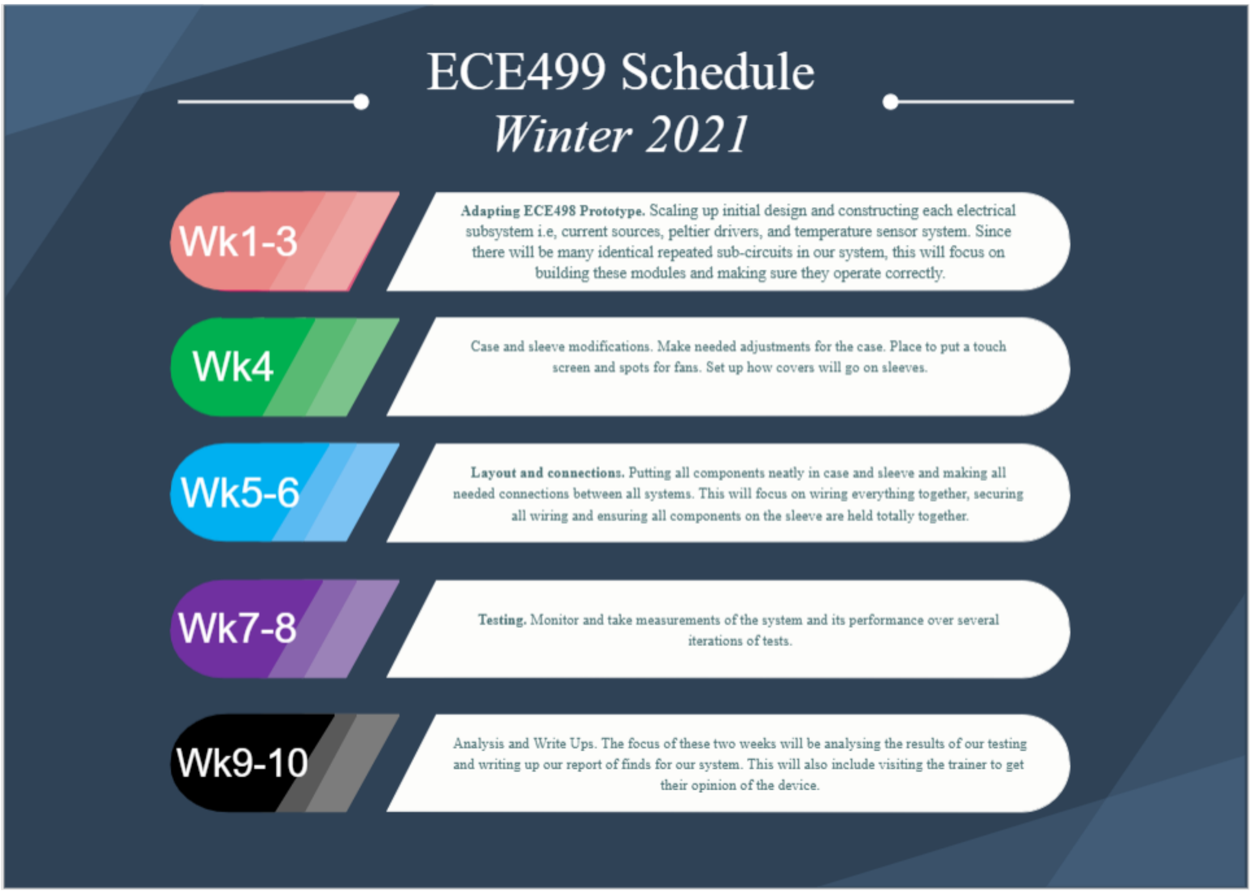


Figure 65: ECE499 Winter Term Plan

In actuality, our schedule is more accurately represented below in Figure 66.

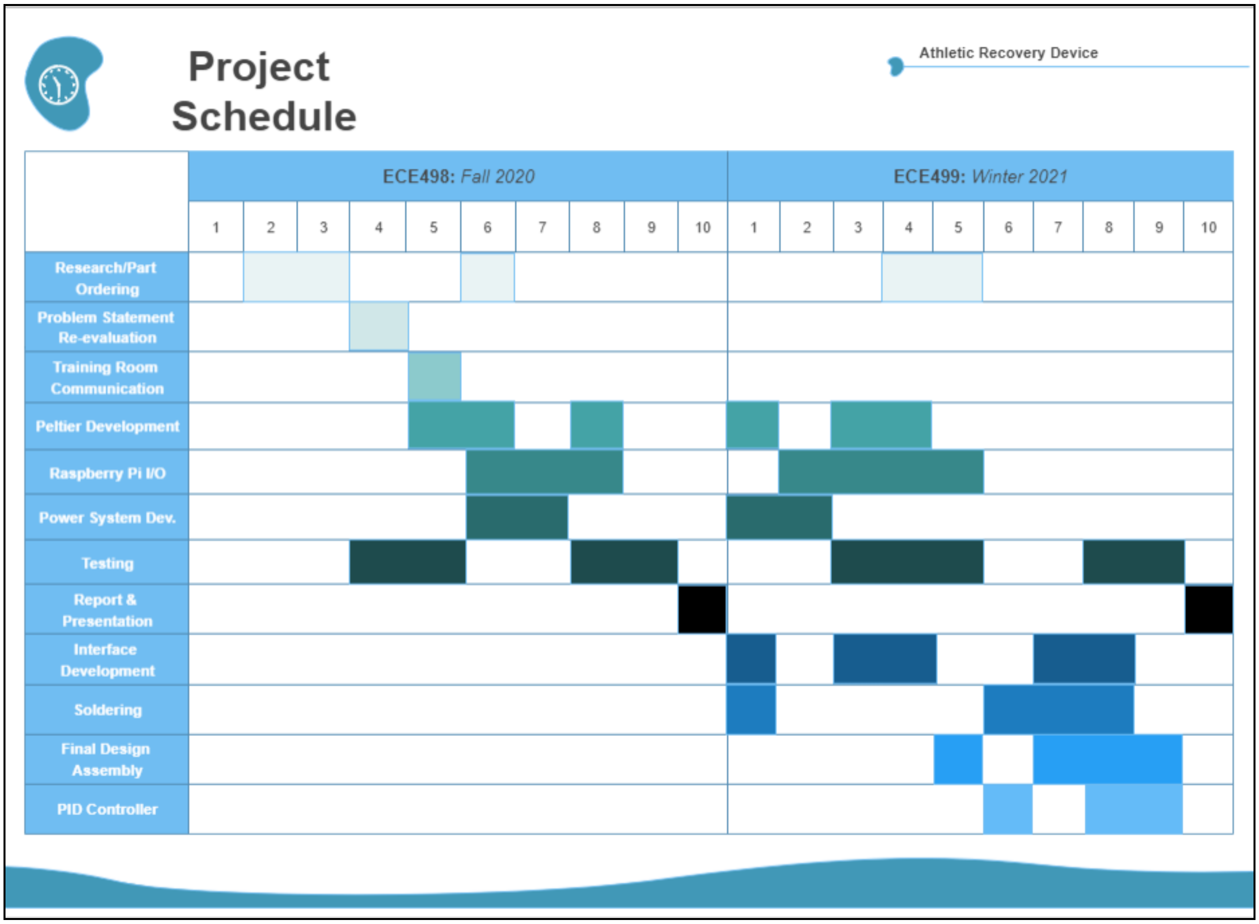


Figure 66: Actual Implementation Schedule

As shown in the actual implementation schedule, the initial phase of our project was allotted to research and planning. We needed to gain a solid understanding of the potential methods and solutions to our problem statement, so that we did not waste time or effort retracing our steps at a later date. Next, we began to access ways of implementing our respective subsystems: temperature control, power regulation, and information processing. The majority of our time was spent in this middle phase of our design. At the conclusion of the fall term, we were able to integrate these subsystems into a preliminary design. For ECE499, we tuned the initial design so that it could best fulfill the design requirements. Scaling the product for modularization and assembling the final device characterized the final design phase. Calibration and testing of the final product was completed shortly after.

Upon completion of the project, there are a few modifications to the implementation of the schedule that should have been made. For example, as seen in the fall term plan, we had a slow start to the term; it would have been better to shift our accomplishments back one week, so that we could have received our parts earlier. Overall, we were able to complete our preliminary design in a timely manner. For the winter term, we had a good start to the term, accomplishing our tasks that we had laid out in the plan. Week 4 through Week 6 is where we experienced some slowdown, as soldering and the integration of the subsystems into the final design took longer than expected. Designing comprehensive circuit schematics and being more organized when wiring components would have allowed for faster completion of these tasks. The final stage of design: assembly, testing, and calibration, were able to be completed in a timely manner considering their importance. We did not have enough time to effectively tune the PI control of the device to reach its maximum performance, since this had to be done after the final system was assembled.

Cost Analysis:

When we initially set out to make this device, our goal was that it would not only be viable in terms of implementation but in cost as well. Our target goal was for the device to be under $1500, calculated from the current cost of recyclable ice bags used by the training room. As we reach the end of our project, we can estimate the material cost as well as the labor costs to assemble our device. Regarding material cost, we made a table of all the components that were used in the final product, as shown below (Table 5). Note that any item under $10 was omitted from this estimate. Furthermore, it needs to be noted that these costs are with respect to our product implementing 2 subsystems. If we increased the number of subsystems, this cost would increase.

Item	**Cost**
Power supply	$93.50
Raspberry Pi 4 kit	$89.99
Peltiers x2	$47.56
Heat sinks x2	$61.94
Cable	$22.34
Electronics case	$25.99
Gel pack	$16.99
Touch screen display	$63.40
Wires: red, black, and yellow	$29.71
Buck converter	$9.99
Copper bar	$12.42
Terminal block	$11.85
Total	**$485.68**

Table 5: Tabulated Material Cost

Another factor that must be accounted for is labor. We were not entirely sure what an hourly pay for such a job would be, so we assumed $20/hour, which is $5 more than the NY state minimum wage. Then, we made a table of tasks and hours of labor to determine the labor cost for our systems. It is important to note that we only included activities that needed to be done for every system. One time labor costs, such as writing the code and designing the circuit, were not included. The results of this can be seen below in Table 6.

Task	**Time (man-hours)**	**Cost (dollars)**
Perf board soldering	12	240
Inter perf board connections	4	80
Machining case	1	20
Case interior assembly	9	180
Cable connections	2	40
Peltier connections	2	40
Sleeve assembly	4	80
Quality assurance/calibration	1	20
Total	**35**	**700**

Table 6: Labor Costs

Thus we find that for our system the material costs would be $485.68, while the labor costs would be $700. This means that the total cost for our system would be $1185.68. Since this falls under our goal cost of $1500, we are satisfied with our system's cost.

It should be noted that this cost analysis does not take into account other factors that could increase the overall price to manufacture. One factor is the needed capital to start up production. This could require taking out a substantial loan, repayments for which would increase the cost. If investment capital can be secured instead of a loan, the opportunity cost of that investment should also be considered. Another detail is initial costs of researching, designing, and prototyping. This device did not appear overnight, it took almost a year of thinking, designing and testing to make it a reality. It would undoubtedly need even more to actually make it to market. Lastly, profit must be considered. While most companies do not start making a profit until a few years into production, once a company begins making these devices, they would have to be sold at some small percentage above cost.

There are also a number of factors that could lead to cost savings if our device were to go to market; the first of these economies of scale. Once a single factory is set up to produce our product, a lot of costs begin to reduce as production is optimized. We would be able to buy all of our components in bulk, which decreases the cost of each product. We will also be able to ship components in bulk, further reducing their cost. This is especially applicable because many components in our table were ordered from the same supplier at different times, meaning that we paid shipping more times than we had to. It is also possible to reduce labor costs with assembly-line style work. This will lead to more efficient workflow and decrease the hours of work needed for each system. There is even the possibility of completely automating certain aspects of assembly, replacing labor costs with power and maintenance costs. Overall, there are a great many factors that make it difficult to determine the exact price of our product if we were to bring it to market.

User Manual

Operation:

1. Plug ARD grey cable into the wall outlet.
    a. All four fans should turn on and the touch screen should begin to power on.
2. Once at the home screen of the Raspberry Pi, an onscreen keyboard can be loaded if needed.
    a. Click the icon on the top left corner of the screen to load the start menu for the operating system.
    b. On this menu, select “Accessories” then “Keyboard” to launch the software.
3. To load the interface application, open the terminal and execute the interface file

   a. Click on the Terminal icon in the top menu bar (or choose Menu > Accessories > Terminal). A window opens with a black background and some green and blue text. You will see the command prompt: `pi@raspberrypi:~ $`.
   b. Type into the terminal the following commands, pressing enter after each one: `cd capstone/`, `cd seniorcapstone/`, `python3 tk_interface.py/`.
   c. You will be prompted in the terminal to enter a filename to save the thermistor data to. Do not use spaces. Press enter when finished.
4. A Start Page will appear, click the start button to begin navigating the interface.
   a. If the interface does not appear in full screen, exit the keyboard, click the fullscreen button in the top right corner of the interface, then reload the keyboard.
5. Choose one of the pre-programmed treatment modes or select Custom
   a. If selecting an existing preset, proceed to Step #6.
   b. If entering a custom treatment mode, enter the password: U
      i. Set the desired temperature and duration of treatment, then hit the enter button and proceed to Step #6.
6. Proceed to strap on device to appendage.
   a. Place the small velcro strap parallel to the cable on either end of the light blue side of the gel pack.
   b. Take the long velcro strap and place it on the opposite side and wrap it around the arm, feeding the other end under and around the smaller velcro strap before wrapping it around the arm again.
7. Once securely fastened to appendage, press Begin on the interface to start the cooling treatment.

8. While treatment is proceeding, remain still and observe the remaining time and current temperature.
9. Once the timer on the current treatment ends, a message will display informing the user that treatment is over.
    a. Users should notice that the temperature of the device quickly returns to room temperature after about 30 seconds.
10. Proceed to remove sleeve from appendage.
    a. To remove, perform Step #6 in reverse order.
11. Once the sleeve is removed, wipe the side of the gel pack that was in contact with the skin with a sanitary wipe.
12. Place the sleeve back in the position it was found for the next user.
13. If you are the last user, please unplug the ARD (Fig. 67).

Figure 67: System in Operation

Troubleshooting:

1. Ensure that the system is fully plugged into the wall.
2. If there is a problem with the user interface, try restarting the touchscreen and repeating the set-up process.
3. If there is another problem, while the system is unplugged, remove the lid of the ARD (Fig. 68).

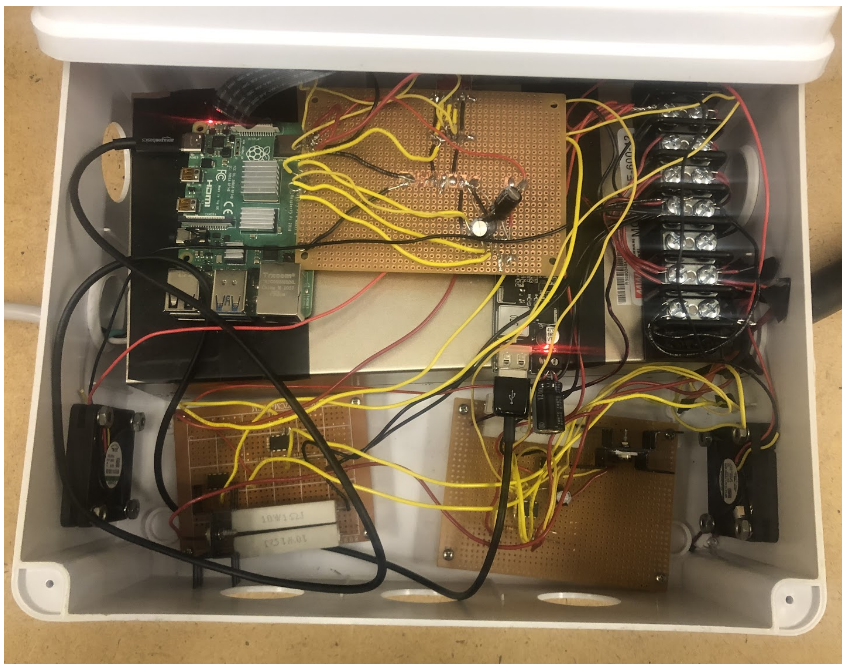


Figure 68: Case Interior Wiring and Layout

4. Look for any signs of scorching or melting in any of the components. If there is a burning smell, it is likely that a component has overheated, this part will need to be replaced. An example of a broken component is shown below (Fig. 69).

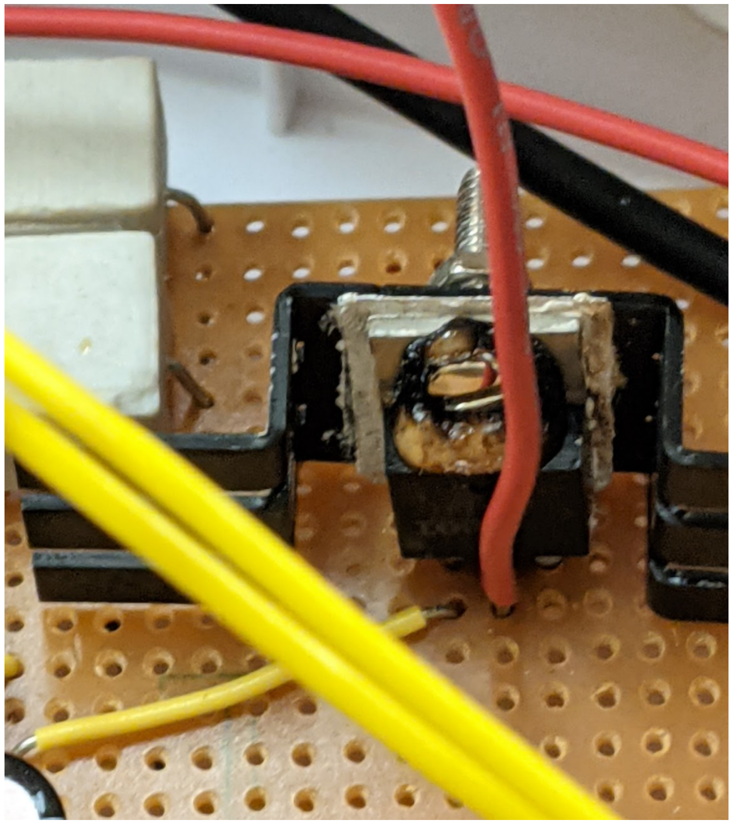


Figure 69: Burned out MOSFET

Storage and Maintenance:

1. Wipe down the area of the gel-pack exposed to skin after every use.
2. Do not store in wet or damp space.
3. Keep liquids away from the system.

Discussion, Recommendations, and Conclusion

Discussion:

Over the course of three academic terms, what started out as little more than an idea has been born into reality. We have successfully researched, designed and constructed an medical device that is able to treat an existing issue in a unique and sustainable way. While not all the design requirements were fully met, we successfully demonstrated the proof of the concept, as well as gathered and learned enough information about the system so that it could be improved to exceed all of its design requirements.

Problem:

Injuries are an unavoidable part of sports. No matter how many regulations, safety protocols, or prevention programs, some forms of injury are bound to occur. Along with these unexpected, acute injuries comes treatment. By far the most prevalent and widely known form of treatment in sports is icing. The most common and practical way of applying icing in athletic environments is through the use of single use disposable ice packs. While sanitary, efficient and practical when dealing with large numbers of injuries, such a method also comes with several drawbacks, including not providing high quality treatment, consistent temperature regulation, and being generally unsustainable in material use.

Solution and Implementation:

The main objective of this capstone project was to design a viable device for a trainer that could provide better treatment than standard ice packs while being more sustainable. The end system that was designed used electronics to cool copper bars inside a makeshift nylon sleeve, made of a lobotomized gel pack, down to a desired temperature by using negative feedback from temperature sensors on the copper bars. This system was divided between the sleeve and the electronics case. The sleeve contained the Peltier units that performed the cooling, the thermistors sensors to monitor the temperature, and the copper bar to distribute the cooling. The case contained the drivers that control the Peltiers, the control loop mechanism to adjust the drives for the Peltiers, and the interface to interact with users. As a result of this system, individuals that use this device will receive consistent high quality cold therapy treatment with accurate temperature regulation.

Results:

Our end system managed to achieve most of its design requirements, with only a few caveats. When it came to cooling performance, the system succeeded in being able to maintain and regulate the temperatures uniformly over an area fairly well, while reaching set temperatures in less than 5 min and being able to both do 10 to 15 min treatments, as well as continuously operate for a 3 hour duration. While both subsystems were able to reach 50°F, only one was able to go down to 45°F. The primary cause for this issue was improper manufacturing practices that resulted in one subsystem being better thermally connected than the other, though with more time this could be rectified. As for the software, the final system provided an easy to use interface with both preset and custom treatment modes, as well as a fairly accurate and calibrated thermometer system and acceptable control loop mechanism. As for the physical requirements, the system easily fit in a 2 cubic foot area. While the system was not able to achieve even 180° coverage, the trainer did say that the area it did cover was equivalent to the area covered by an ice pack. Lastly, according to our cost analysis, the system achieved its requirement of costing less than $1500. As a result of meeting the vast majority of the design requirements and possessing the knowledge and ability to fix the requirements that fall short, there is a high probability that the design of our proposed system could be a viable device to provide cold therapy treatment and reduce the abundance of single use plastic ice packs.

Recommendation:

While the system satisfies a majority of the aspects that we set out to accomplish, there are several ways in which it could be improved. Some forms of improvement were recommended by the trainer, while other areas of improvements were recognized by ourselves.

Trainer Feedback:

Future improvements suggested by the head athletic trainer ranged from addressing issues in the system, to pointing out ways to improve and how to market it. Potential issues pointed out by the trainer included: reducing the size of the protruding heatsinks and improving the robustness of the overall system, since it might not fare well in a rough athletic environment, She also suggested we reduce the size and weight of the of the cable, as well as secure it more firmly to the sleeve. Another suggestion she had for improving the system was to see if it was possible to make the system provide contrast therapy, or heating as well as cooling. Lastly, she said this device would be great in situations when a team is traveling and does not have access to ice. She advised us to highlight this quality if we were to market it.

Future Works:

While overall, we believe this is a satisfactory system, we identified 3 areas that could be improved: performance, construction, and software.

1. Performance:

While the system could roughly reach the desired temperatures, we think it could perform at even lower temperatures. There are few ways which we think we could implement changes to accomplish this. Since we suspect that the reason that one of our subsystems was not performing as well as the other was due to insufficient thermal connection, one way would be to apply an optimal amount of thermal paste to optimize performance. Another is to drive the Peltiers at a higher current, which would make them produce a lower temperature on the cold side. While the Peltiers we were using could manage higher currents, their efficiency may decrease. Alternatively, if we apply two Peltiers to one copper bar we are certain that the system will be able to reach much lower temperatures. However, given that it is not desirable to produce temperatures lower than 32 degrees Fahrenheit, this may be overkill.

Another area that we would improve is coverage. Our goal was for the system to provide 360 degree coverage, but unfortunately it does not even reach 180 degrees of coverage. Such an issue could easily be reminded by implementing more subsystems, adjacent to the existing modules, such that they could wrap around the injured appendage.

When it came to temperature measurements, we saw several areas for improvement. First, due to the tight space in the sleeve, it was difficult to place the thermistors on the copper bar. As a result, the positioning and thermal contact of the thermistors may be slightly different and could impede the performance of the system. Secondly, the thermistors measure the temperature of the copper bar, when the real measurement of interest is the temperature of the surface of the sleeve in contact with the skin. Thus, by changing the placement of the thermistors, we may be able to improve the quality of the treatment.

2. Construction:

There is an extensive list that can be improved for construction of the system. One improvement would be making the wiring neater. Another would be having a more viable means of securing the sleeve electronics. Using rubber bands to hold the sleeve electronics together is not a viable long term solution, since there is a greater potential for the rubber bands to break than bolted heatsinks. When it comes to safety and isolation there is much that needs to be improved. Using electric tape is not a viable solution in the long term to provide isolation because if the electric tape were to loosen or begin to unravel, wires could become exposed, potentially causing a short circuit. Instead, we should have more space in the case so that electronics do not need to rest on top of the power supply, and we should use shrink tubing for wire connections.

Looking at the sleeve and the cable, there is also much that could be improved. Regarding the sleeve, it would be better if we could find or design our own that was easier to work with rather than draining and cleaning a commercial gel pack. Furthermore, while our velcro strapping method works, a different strapping method could be designed that would be more secure and easy to use. As for the cable the two biggest improvements that could be made are using one with the precise number of needed wires and having it secured better to the case and the sleeve. Using a cable with the exact number of wires we needed we would cut down on weight and make it more flexible. Instead of securing the cable to the sleeve using electric tape, it would be better to sew the sleeve shut to hold the cable more securely and thus relieve strain on the internal sleeve wires.

3. Software:

In order to further improve the information processing component of the device, two main areas which we recognize could be adjusted are the tuning of the PI control mechanism and the structure of the code.

Unfortunately, we were not able to perfectly calibrate our PID system. This practically means that while the system approaches the right temperature consistently, there are small oscillations of less than 1°F pk at various temperatures. We noted that this is a negligible temperature change and should not affect the quality of the treatment. But, to fix this issue, we would increase the K_P values for the oscillating thermistor while keeping the other thermistor's value of K_P constant. Another component of the PI controller that we would have adjusted if time had allowed is the integral constant. Increasing this term decreases the steady state error.

Multiple iterations of adjusting this gain value could have diminished the difference between the target temperature and the temperature the device settles on after a few minutes, which is consistently about 1.5 degrees Fahrenheit less.

In addition, the format of the code could be improved upon so that in future models, more subsystems could be implemented. While the interface and the control file interact with each other well for our current final product, it would be difficult to adjust the code for more than two thermistors. As it is, the code for each thermistor is duplicated in the control file to read the value, calculate the temperature, and adjust the PI output. In a modularized system, a separate thermistor Python script would be created so that Thermistor Objects could be created in the control file. Though the current code functions well, implementing object oriented code for the thermistor objects would greatly improve the readability and scalability of the software.

Lessons Learned:

Throughout this project we have come to learn several vital engineering lesions. One is communication and collaboration. Such a large scale project has shown us how engineers will often have to work ingroups in order to solve a problem. In order to work properly together there needs to not only be a lot of communication but good communication. It is essential that everyone is kept inside the loop and knows what has been done, what needs to be done and where the group is at. As for good communication, it is essential that everyone is on the same page and in agreement about what is being said. This means more than just telling other members you make sure that they understand. Another lesson we learned is the importance of organization, both in our work and in our schedule. Our weekly memos have shown use just how valuable it is to keep track of what you have done and what it means going forward. Not only do they help when putting everything together in the end it helps show the flow of the project and keep us on course by setting weekly milestones and goals. Lastly, this project has shown us and given us first hand experience with the Design process. This included: coming up with a problem to address, setting the requirements needed to address the problem, seeing what existing solutions already existed, brainstorming alternative solutions, researching the best one, designing the system, testing the system, and then analyzing the performance of the system. Some other skills and lessons we learned along the way included communicating with and seeing what a client wanted, how to create and propose budgets, and how to acquire parts and interact with specialists in other fields in order to make up for the skill that we ourselves did not possess.

Conclusion:

To conclude, this project addressed the issues brought on by single-use recyclable plastic ice packs, which provided sub par cold therapy treatment due to lack of temperature regulation, by designing an sustainable alternative electrical cooling system. This system was intended to consistently provide high quality, temperature controlled, cold therapy treatment that was sanitary, able to be used by multiple individuals, and safe. Clearly, this was a very tall order to meet, but in the end the overall systems achieved this objective well. Though it fell short in some areas, enough testing and data was gathered that in the future, it would not only be possible to meet these requirements, but to exceed them. The means by which we achieved this both did not exist in existing commercial products and proved to provide the satisfactory results. What we noted was interesting was how much of the overall preliminary design remained the same in the final design. As for the project schedule, while not always on track, we can confidently say that even though intermediate personal deadlines were sometimes missed, the practical deadlines for the project were always achieved on or ahead of schedule. Though our design for the project is now concluded, there is still much room for improvement in the system. All this being said, the most important part of this project was the immeasurably valuable lessons learned, that will prove useful in our engineering experiences for years to come. Having come up with this project idea, seeing it bloom into reality, and witnessing it successfully perform what we envisioned is immeasurably rewarding, and it fills us with more joy than we can express in words.

References

[1] Rockwood, Cheryl; Crary, Jill, personal communication, Apr.-Nov 2020.

[2] Murray, A., Cardinale, M. Cold applications for recovery in adolescent athletes: a systematic review and meta analysis. *Extrem Physiol Med* 4, 17 (2015). https://doi.org/10.1186/s13728-015-0035-8

[3] Stöckle, U et al. "Fastest reduction of posttraumatic edema: continuous cryotherapy or intermittent impulse compression?" *Foot & ankle international* vol. 18, no. 7, pp. 432-8, Jul. 1997. [Online]. Available doi: doi:10.1177/107110079701800711

[4] *PowerPlay*, PowerPlay, 2019. [Online]. Available: https://powerplay.us/

[5] *Game Ready,* Avanos Medial, Inc, 2020. [Online]. Available: https://gameready.com/

[6] Source Ortho. (2020). *Breg Polar Care Kodiak Cold Therapy Unit*. retrieved from https://www.sourceortho.net/breg-polar-care-kodiak-cold-therapy-unit/

[7] Source Fitness (2020). *ThermaZone Temperature Therapy System.* Retrieved from https://sourcefitness.net/thermazone-thermal-therapy-system/?sku=003-99&gclid=Cj0KCQjw5eX7BRDQARIsAMhYLP8IOMaAxEn4RUb5t6uZJbQ541Pj74HGMQ-APQyfNO0Mc9hdtgukpGYaAkWXEALw_wcB

[8] *The Science Behind Refrigeration*, Berg Chilling Systems Inc., 2017. [Online]. Available: https://berg-group.com/engineered-solutions/the-science-behind-refrigeration/

[9] T.M.Tritt, "Thermoelectric Materials: Principles, Structure, Properties, and Applications", in *Encyclopedia of Materials: Science and Technology,* South Carolina: 2002, pp. 1-11, Available: https://www.sciencedirect.com/topics/chemistry/peltier-effect

[10] Gurevich, Yuri & Velázquez-Pérez, Jesús. (2014). Peltier Effect in Semiconductors. 10.1002/047134608X.W8206. Available: https://www.researchgate.net/publication/262179901_Peltier_Effect_in_Semiconductors

[11] H. Lu, "Peltier Temperature Controller," MEng. report, Cornell University, Ithaca, NY, 2013.

[12] Kordani, N., Rahmani, A., & Hasanzadeh, R. P. (2018). Smart Portable Cryotherapy System Involving Controlled Thermoelectric Cooling Modules for Medical Applications. *IIUM Engineering Journal*, *19*(1), 117-128.

[13] Mejia, N., Dedow, K., Nguy, L., Sullivan, P., Khoshnevis, S., & Diller, K. R. (2015). An On-Site Thermoelectric Cooling Device for Cryotherapy and Control of Skin Blood Flow. Journal of medical devices, 9(4).

[14] Okabe, T., Fujimura, T., Okajima, J., Aiba, S., & Maruyama, S. (2018). Non-invasive measurement of effective thermal conductivity of human skin with a guard-heated thermistor probe. *International Journal of Heat and Mass Transfer*, *126*, 625-635.

[15] Nicholas, J. J., Taylor, F. H., Buckingham, R. B., & Ottonello, D. (1976). Measurement of circumference of the knee with ordinary tape measure. *Annals of the rheumatic diseases*, *35*(3), 282-284.

[16] Zimmer MedizinSystems. (2020). *Cryo 6 Air Chiller Device*. Retrieved from https://zimmerusa.com/products/cryo-therapy/cryo-6/

[17] RecoverX. (2020). *Electric Cold and Heat Therapy in 60 Seconds.* Retrieved from https://www.recoverx.io/

[18] Melin, O. (2019) *Driving a Peltier Element(TEC): Efficiency and Aging* (Application Report No. SLUA979A) Retrieved from https://www.ti.com/lit/an/slua979a/slua979a.pdf?ts=1601492867207&ref_url=https%253A%252F%252Fwww.google.com%252F

[19] Texas Instruments, "TPS54200, TPS54201 4.5-V to 28-V Input Voltage, 1.5-A Output Current, Synchronous Buck Mono-Color or IR LED Driver",TSP54200, June 2018. Retrieved from: https://www.ti.com/lit/ds/symlink/tps54201.pdf?ts=1601495369446&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS54201

[20] Analog Technologies, "High Voltage High Current TEC Controller", MAX1969, August 11, 2020. Retrieved from:http://www.analogtechnologies.com/document/TEC28V15A.pdf

[21] Maxim Integrated Products, "Power Drivers for Peltier TEC Modules", TEC28V15A, May 2015. Retrieved from:https://datasheets.maximintegrated.com/en/ds/MAX1968-MAX1969.pdf

[22] Pounder, Les. "Raspberry Pi vs Arduino: Which Board Is Best?" *Tom's Hardware*, Tom's Hardware, 10 July 2020, Retrieved From: www.tomshardware.com/features/raspberry-pi-vs-arduino .

[23] Raspberry Pi vs Arduino: Top 10 Useful Differences You Need To Learn. (2020, August 10). Retrieved November 21, 2020, from https://www.educba.com/raspberry-pi-vs-arduino/

[24] Experience, W. L. in R.-B. U. (n.d.). *The Difference Between Web Design and GUI Design*. Nielsen Norman Group. Retrieved November 21, 2020, from https://www.nngroup.com/articles/the-difference-between-web-design-and-gui-design/

[25] Shturma, S. (2020, May 26). *Pyxis by Renetec, Inc.* Renetec, Inc. https://www.renetec.io/blogs/makers/7-tools-for-gui-development-on-raspberry-pi

[26] Thermonamic electronics, "Specifications of Thermoelectric Module", TEC1-04905. Retrieved from:http://www.thermonamic.com/TEC1-04905-English.pdf

[27] Wakefield-Vette. "TEC Series", TEC-30-38-71. Retrieved from http://www.wakefield-vette.com/Portals/0/Peltier%20Thermoelectric%20Coolers/Wakefield%20Vette%20Thermoelectric%20Cooler%20Full%20Data%20Sheet%20Final.pdf http://www.wakefield-vette.com/Portals/0/Peltier%20Thermoelectric%20Coolers/Wakefield%20Vette%20Thermoelectric%20Cooler%20Full%20Data%20Sheet%20Final.pdf

[28] Vishay. (January 1, 2019). *NTC Thermistors, 2-Point Mini Chip Sensor, Flexible Leads.* Retrieved from https://nexus.union.edu/pluginfile.php/563941/mod_folder/content/0/Thermistor10k_NTCLESB.pdf?forcedownload=1

[29] Wakefield-Vette. "HSF Series", HSF-55-33-B-F. Retrieved from http://www.wakefield-vette.com/Portals/0/Heat%20Sink%20Fan%20Combinations/Heat%20Sink%20Fan%20Combinations%20Data%20Sheet%20HSF%20Series%20Wakefield%20Vette.pdf

[30] The Raspberry Pi Foundation(2020) *GPIO* Retrieved From https://www.raspberrypi.org/documentation/usage/gpio/README.md

[31] Adafruit. (2020). *MCP3008 - 8-Channel 10-Bit ADC with SPI Interface.* Retrieved from https://www.adafruit.com/product/856

[32] Adafruit. (2020). *MCP4725 Breakout Board - 12-Bit DAC w/ I2C Interface.* Retrieved from https://www.adafruit.com/product/935

[33] Adafruit. (2020). *Configuring I2C.* Retrieved from https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c

[34] Digi Key Electronics. (2020). *SE-600-12.* Retrieved from https://www.digikey.com/en/products/detail/mean-well-usa-inc/SE-600-12/7706613?s=N4IgTCBcDaIMoFEC0A2ADGpBGCBdAvkA

[35] Amazon. (2020). *Buck Converter 12v to 5v, DROK 5A USB Voltage Regulator DC 9V-36V Step Down to DC 5V-5.3V 5.2V 3.5-6A Volt Transformer Power Supply Module for Phone Fast Charging.* Retrieved from

https://www.amazon.com/Converter-DROK-Regulator-Inverter-Transformer/dp/B01NALDSJ0/ref=sr_1_8?dchild=1&keywords=12v+to+5v+step+down+converter&qid=1604166099&sr=8-8

[36] Engineering Tool Box. (2020). *AWG Wire Gauges Current Ratings*. Retrieved from https://www.engineeringtoolbox.com/wire-gauges-d_419.html

[37] Show Me Cables. (2020). *18 AWG - 5 Conductor - 600V - Stranded Conductor - Unshielded - VNTC Tray Cable - Per Foot.* Retrieved from https://www.showmecables.com/18-awg-5-conductor-600v-stranded-conductor-unshielded-vntc-tray-cable-per-foot

[38] Amazon. (2020). *LLPT Double Sided Thermal Conductive Strong Adhesive Tape Multiple 1.2 Inches x 108 Feet for Electronic Components LED Strips Heat Sinks(TC128).* Retrieved from https://www.amazon.com/LLPT-Conductive-Electronic-Components-TC128/dp/B07P9LV4V3/ref=asc_df_B07P9LV4V3/?tag=hyprod-20&linkCode=df0&hvadid=459454121958&hvpos=&hvnetw=g&hvrand=4935736664993830224&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9004745&hvtargid=pla-945945765541&th=1

[39] Amazon. (2020). *Sunnyglade Plastic Waterproof Dustproof Junction Box DIY Case Enclosure (11.8"x9.8"x4.7").* Retrieved from https://www.amazon.co.uk/Sunnyglade-Waterproof-Dustproof-Junction-Enclosure/dp/B07KR4PCD8

[40] Amazon. (2020). *Gel Ice Packs for Injuries- Adjustable Wrap for Pain Relief Ice Pack+Heat Pack， Non-Toxic Smooth Reusable, Support Rehabilitation, Flexible Therapy - Shoulder, Back, Knee, Neck, Ankle, Jaw & More.* Retrieved from https://www.amazon.com/Injuries-Adjustable-Non-Toxic-Reusable-Rehabilitation/dp/B07KWVFMLC/ref=sr_1_13?dchild=1&keywords=thin+ice+pack&qid=1603053550&sr=8-13

[41] Ashmen, Amanda, seniorcapstone, (2020), Github repository, https://github.com/amandaashmen/seniorcapstone

[42] CUI Devices, “CP85 Series Datasheet - Single-Staged Peltier Modules”, CP85 datasheet, October 28, 2019. Retrieved from https://www.cuidevices.com/product/resource/cp85.pdf

[43] Amazon. (2021). *FlexiKold Gel Ice Pack w/Straps (Half Size: 7.5" x 11.5") - Reusable Cold Pack Compress (Therapy for Pain and Injuries of Knee, Shoulder, Foot, Back, Ankle, Hip) - 6303-COLD-STRAPPED.* Retrieved from

https://www.amazon.com/FlexiKold-Pack-Straps-Standard-Large/dp/B07J1P77W5/ref=sr_1_2_sspa?dchild=1&keywords=flexi%2Bcold%2Bgel%2Bpack%2B7x11&qid=1612999424&sr=8-2-spons&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzT1FSUjhROUJaRzE2JmVuY3J5cHRlZElkPUEwMTU0MzU2MjFFS09HVkxHN0JaVSZlbmNyeXB0ZWRBZElkPUEwODM4MDQxRzAzNUlCWENWM1dOJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ&th=1

[44] Ada, Lady. “Overview: MCP4725 12-Bit DAC Tutorial” Adafruit. https://learn.adafruit.com/mcp4725-12-bit-dac-tutorial/overview.

[45] “MCP4725 12-Bit DAC Module” Proto Supplies, https://protosupplies.com/product/mcp4725-12-bit-dac-module/.

[46] TKinter - Python interface to Tcl/Tk. “The Python Standard Library,” Python Software Foundation. https://docs.python.org/3/library/tkinter.html

[47] Wescott, Tim. “PID Without a PhD,” in *Embedded Systems Programming,* Elsevier, Burlington, Ma, US; 2008, pp. 86-108. Available: https://cms.edn.com/contenteetimes/documents/embedded.com/2000/f-wescot.pdf

[48] Durmusoglu, Caner, ivPID (2015), GitHub repository, https://github.com/ivmech/ivPID/blob/master/PID.py

[49] George Ellis, “Chapter 6 - Four Types of Controllers” in *Control System Design Guide*, 4th ed., (Fourth Edition), Butterworth-Heinemann, 2012, pp. 97-119. Available: https://www.sciencedirect.com/topics/computer-science/ziegler-nichols-method

[50] Buma PhD., Takashi, W 2021, *Lecture 15: Medical Device Regulation*, Lecture notes, Biomedical instrumentation with lab BME/ECE 386, Union College, delivered 4 March 2021.

[51] “CFR - Code of Federal Regulations Title 21.” *Accessdata.fda.gov*, FDA, 1 Apr. 2020, www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?fr=890.5740.

[52] “Product Classification.” *Accessdata.fda.gov*, FDA, www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfpcd/classification.cfm?id=5320.

[53] Center for Devices and Radiological Health. “De Novo Classification Request.” *U.S. Food and Drug Administration*, FDA, 20 Nov. 2019, www.fda.gov/medical-devices/premarket-submissions/de-novo-classification-request.

[54] “ISO 13485:2016.” *ISO*, ISO, 21 Jan. 2020, www.iso.org/standard/59752.html.

[55] “Ergonomics of Human-System Interaction — Part 161: Guidance on Visual User-Interface Elements.” *ISO*, ISO, 2016, www.iso.org/obp/ui/#iso:std:iso:9241:-161:ed-1:v1:en.

Appendices

1. Budget breakdown

Budget Breakdown (Total = $1,226.06) (Consumable = $926.37) (non-consumable=$299.69)
We are requesting SRG funds of $1136.15. The ECBE dept will provide 30% matching funds of $89.9 for non-consumables.

Note: * = Item is consumable and not reusable equipment

Stage	Part	Purpose	Supplier	Price
User Interface	CanaKit Raspberry Pi 4 4GB Starter Kit	To provide control over the device to the user and determine the signal to power Peltiers	Amazon	$89.99
	7' Touchscreen Display	Physical interactive display of user interface	Amazon	$63.40
	2 Adafruit MCP3008 8-Channel 10-Bit ADC With SPI Interface *	Converts analog output of thermistor to digital input to Raspberry Pi	Amazon	$14.18
	5 Mini MCP4725 Module I2C DAC *	Converts digital signal from Pi to analog signal for MOSFET	Amazon	$10.99
Peltier and Drivers	3 20AWG wire (red, yellow, black)*	Wire that can handle up to 6A of current and will be used for wiring up the Peltiers and other components	Digi-key	$29.71
	1 Terminal blocks	Serve a a high current junction that can distribute current to multiple pelters for testing	Digi-key	$11.85
	10 10W Resistors*	Part of Current Source to Drive Peltiers	Amazon	$17.26
	7 IRL520NPbF MOSFETs*	Part of Current Source to Drive Peltiers	Digikey	$12.06
	12 Peltier Coolers *	Required to maintain cold temperature	Amazon	$138.45
	13 HSF-55-33-B-F heatsinks with fans *	Prevent the Peltiers from overheating	Newark	$486.81
	5 Thermistors*	Measure external temperature of system for optimal healing	Digi-key	$10.75
Power	1 DC-power supply SE-600-12	Convert wall outlet down to 12V DC output and up to 50A.	Digi-key	$93.50
	1 Power cable 211018-06	Power cable connecting wall outlet to DC-power supply	Digi-key	$13.50
	1 18 AWG - 5 Conductor - 600V - Stranded Conductor*	Wire to handle and distribute power to Peltiers and fans	Show me cable	$22.34
Regulation of Pi	2 Buck Converter 12v to 5v, DROK 5A USB Voltage Regulator DC 9V-36V Step	Step down the voltage from 12 to 5 volts and is able to provide the 3 amps needed by the pi	Amazon	$19.98
	1 AmazonBasics USB Type-C to USB-A 2.0 Male Charger Cable, 3 Feet (0.9	Used to connect the regulator output to the pi	Amazon	$7.47
Safety features	4 Gikfun 20A Range Current Sensor ACS712 Module for Arduino (Pack of 2pcs) *	Measure the current through the Peltier	Amazon	$43.92
Other	Sunnyglade Plastic Box *	Case to hold user interface components and power system.	Amazon	$25.99
	1 Roll of Thermal Adhesive Tape*	Attach multiple components on the sleeve together	Amazon	$17.27

	1 Copper plate*	Evenly distribute temperature across cold pack	McMaster Carr	$76.75
	2 Gel pack*	Provide medium for heat transfer, give design flexibility	Amazon	$19.89
Total				**$1226.06**

2. Code for Reading Thermistor and Producing Data Output - seniorcapstone/input_csv.py

```
import os
import time
import busio
import digitalio
import board
import numpy as np
import csv
import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn
from matplotlib import pyplot as plt

# steinhart-hart coefficients
K0 = 0.00113414
K1 = 0.000233106
K2 = 9.32975E-8

# create the spi bus
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)

# create the cs (chip select)
cs = digitalio.DigitalInOut(board.D22)

# create the mcp object
mcp = MCP.MCP3008(spi, cs)

# create an analog input channel on pin 0
chan0 = AnalogIn(mcp, MCP.P0)

# begin reading
start_time = time.time()

# length of time program will run
DURATION = 360

last_read = 0       # this keeps track of the last value
tolerance = 250     # to keep from being jittery we'll only change
                    # volume when the pot has moved a significant amount
                    # on a 16-bit ADC


tempList = []       # creates an empty list of temperature values read
timeList = []       # creates an empty list of time values per temperature
```

```
def convert_V_to_T(V):
    # takes a voltage value from amplifier to ADC
    # maps to internal resistance of thermistor
    # calculates temperature in Celcius and Fahrenheit from Steinhart-Hart
Equation
    # returns temperature value in Fahrenheit

    # voltage to resistance
    R = (5.2*100)/(.5*V + 2.6) - 100

    # resistance to temperature
    term1 = K0
    term2 = K1*(np.log(1000*R))
    term3 = K2*(np.log(1000*R))**3
    c = (term1 + term2 + term3)**(-1) - 273.15  # degrees celcius
    f = c*(9/5) + 32                            # degrees fahrenheit

    #print formatting
    print('{:.3f}'.format(c) + " C")
    print('{:.3f}'.format(f) + " F")

    return(f)

def remap_range(value, left_min, left_max, right_min, right_max):
    # this remaps a value from original (left) range to new (right) range
    # Figure out how 'wide' each range is
    left_span = left_max - left_min
    right_span = right_max - right_min

    # Convert the left range into a 0-1 range (int)
    valueScaled = int(value - left_min) / int(left_span)

    # Convert the 0-1 range into a value in the right range.
    return int(right_min + (valueScaled * right_span))

def graphData(dataList, timeList):
    # creates csv file to write data to
    filename = 'test11_4.csv'
    with open(filename, mode='w', newline= '') as data:
            tempData = csv.writer(data, quoting=csv.QUOTE_MINIMAL)
            tempData.writerow(["Time    Temperature"])

            xList = []
            yList = []
            for point in range(len(dataList)):
                temp = dataList[point]
                time = timeList[point]

                xList.append(time)
                yList.append(temp)
                tempData.writerow([str(time)+'\t'+str(temp)])

    plt.ylabel('Temperature (F)')
    plt.xlabel('Time (s)')
    plt.title('Thermistor Values')
    plt.plot(xList, yList)
    plt.show()
```

```
while True:
    # we'll assume that the thermistor didn't move
    therm_changed = False

    # read the analog pin
    therm = chan0.value

    # how much has it changed since the last read?
    therm_adjust = abs(therm - last_read)

    if therm_adjust > tolerance:
        therm_changed = True

    # convert 16bit adc0 (0-65535) thermistor read into 0-5.2V voltage value
    set_voltage = remap_range(therm, 0, 65535, 0, 5.2)

    volts = round((chan0.value*5.22)/65535, 2) # DO i need both of these

    degrees_f = round(convert_V_to_T(volts), 2)
    elapsed_time = round(time.time() - start_time, 2)

    tempList.append(degrees_f)
    timeList.append(elapsed_time)

    # print statements to console
    print('Raw ADC Value: ', chan0.value)
    print('Raw Converted Voltage: ', str(volts) + ' Volts')
    print('Time: ', str(elapsed_time) + ' seconds')
    print()

    # save the thermistor reading for the next loop
    last_read = therm

    # hang out and do nothing for a half second
    time.sleep(0.5)

    # end program after specified time in seconds
    if elapsed_time > DURATION:
        break

graphData(tempList, timeList)
```

3. Code for Temperature Control Mechanism - seniorcapstone/control_final.py

```
############################################################################
##
# Main control algorithm to read thermistor values, calculate output signal
# needed to approach target value, and drive a voltage signal to peltiers.
#
############################################################################
#
```

```
import os
import sys
import csv
import time
import busio
import board
import digitalio
import numpy as np
from pid_git import PID
import adafruit_mcp4725 as DAC
from matplotlib import pyplot as plt
import adafruit_mcp3xxx.mcp3008 as MCP
from adafruit_mcp3xxx.analog_in import AnalogIn

## SAVING TO FILENAME
try:
    FILENAME = sys.argv[1]
except IndexError:
    FILENAME = input("Enter filename.\n")

## GLOBAL VARIABLES
OUTPUT = 0                             # Volts
MAX_DAC = 3.78                         # Volts
MAX_PELT = 2.5                         # Volts
VS = 5.9                               # Volts
R1 = 151.2                             # kOhms
# steinhart-hart coefficients
K0 = 0.00113414
K1 = 0.000233106
K2 = 9.32975E-8

## DAC SET-UP
i2c = busio.I2C(3, 2)                                      #
Initialize I2C bus: 3 = scl pin, 2 = sda pin
dac = DAC.MCP4725(i2c, address=0x60)                       #
Initialize MCP4725 - DAC objects
dac2 = DAC.MCP4725(i2c, address=0x61)
dac.normalized_value = 0                                   #
Initialize DAC outputs to 0 Volts
dac2.normalized_value = 0

## ADC SET-UP
spi = busio.SPI(clock=board.SCK, MISO=board.MISO, MOSI=board.MOSI)   # create
the spi bus
cs = digitalio.DigitalInOut(board.D22)                     # create
the chip select
mcp = MCP.MCP3008(spi, cs)                                 # create
the mcp object MCP3008 - ADC
```

```
chan0 = AnalogIn(mcp, MCP.P0)                                        # create
an analog input channel on pin 0
chan1 = AnalogIn(mcp, MCP.P1)                                        # create
an analog input channel on pin 1
ADC_MAX = 65535

## PID SET-UP
Kp = 28.35                              # proportional gain
Ki =  1.48                              # integral gain
Kd =  0.0                               # derivative gain
SAMPLE_TIME = .5                        # seconds
TARGET = 70                             # Fahrenheit
pelt_pid = PID(Kp, Ki, Kd, TARGET)      # create PID object for therm. 1
pelt_pid2 = PID(Kp, Ki, Kd, TARGET)     # create PID object for therm. 2

## GRAPH SET-UP
DIFF = 0                                # seconds
start_time = time.time()                # begin reading
t1_tempList = []                        # creates an empty list of
temperature values read
t2_tempList = []
timeList = []                           # creates an empty list of time
values per temperature


def adc_voltage(adc_counts):
    """Converts 16bit adc0 (0-65535) thermistor reading to 0-VS voltage
value."""
    adc_16bit = 65535
    volts = (adc_counts*VS)/(adc_16bit)
    return volts

def convert_V_to_T(adc_value, therm):
    """
    Takes a voltage value from amplifier to ADC, maps to internal resistance
of
    thermistor, calculates temperature in Celcius and Fahrenheit from
    Steinhart-Hart Equation. Prints Resistance (kOhms) and Temperature (F).
    Returns temperature value in Fahrenheit.
    """
    if therm == 1:
        GAIN = 2.754-1.041*(adc_value/ADC_MAX)
    else:
        GAIN = 2.766-1.006*(adc_value/ADC_MAX)
    Vout = adc_voltage(adc_value)
    R = (VS*R1)/((1/GAIN)*Vout + (VS/2)) - R1

    # resistance to temperature: Steinhart-Hart Equation
    term1 = K0
```

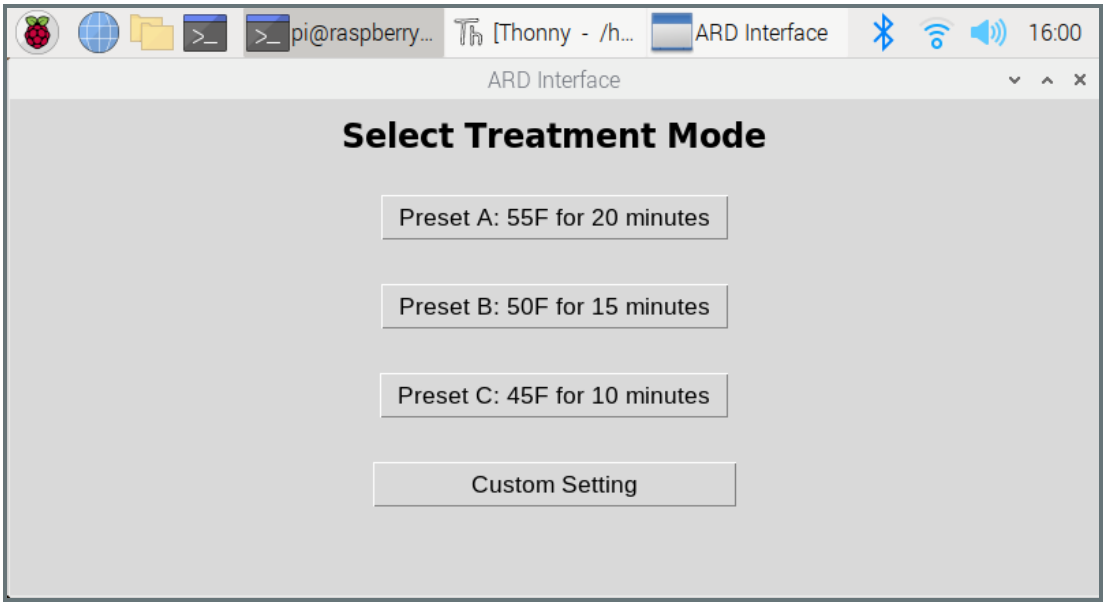
```
    term2 = K1*(np.log(1000*R))
    term3 = K2*(np.log(1000*R))**3
    c = (term1 + term2 + term3)**(-1) - 273.15  # degrees celcius
    f = c*(9/5) + 32                            # degrees fahrenheit

    return(f)

def adc_to_degrees(value, therm):
    """
    Given a value from an ADC, converts value to voltage then
    converts that voltage to temperature in Fahrenheit.
    Returns rounded value in degrees F.
    """
    return round(convert_V_to_T(value, therm), 2)

def convert_T_to_V(temp):
    """
    Takes a target temperature value in Fahrenheit and maps to a linear
equation
    representing the steady-state voltage required to drive peltier
    to reach that desired temperature.
    Returns target voltage (Volts)
    """
    return (71-temp)/10.4

def graphData(dataList1, dataList2, timeList):
    """plots graph of data for two sets of inputs."""
    plt.ylabel('Temperature (F)')
    plt.xlabel('Time (s)')
    plt.title('Thermistor Values')
    plt.plot(timeList, dataList1, label='Therm. 1')
    plt.plot(timeList, dataList2, label='Therm. 2')
    plt.legend()
    plt.savefig(FILENAME+'.png')
    plt.show()

def getAverage():
    """
    Returns the average temperature of the two thermistors
    """
    degrees_f1 = adc_to_degrees(chan0.value, 1)
    degrees_f2 = adc_to_degrees(chan1.value, 2)
    return (degrees_f1+degrees_f2)/2

def setTarget(temp):
    """
    Sets the target temperature for the system to approach.
    """
    global TARGET
```

```
    TARGET = int(temp) +1.5    # Fudge Factor remove if necessary

def updatePID(current_temp, pelt, dac_no, therm):
    """
    Given degrees of a thermistor in Fahrenheit, sets the PID system to
    compute a new output target temperature, which is converted to a
    voltage to drive the peltier unit.
    """
    pelt.update(current_temp)# update pid system with current thermistor
temp.
    target_out_temp = pelt.output
    dac_out = max(min(convert_T_to_V(target_out_temp), MAX_PELT), 0)    #
scales output to maximum voltage peltier can handle
    dac_no.normalized_value = dac_out/MAX_DAC                           # set
pin output to desired voltage value

def ctrlfunc(starttime, counter):
    pelt_pid.setSampleTime(SAMPLE_TIME)
    pelt_pid.setSetPoint(int(TARGET))
    pelt_pid2.setSampleTime(SAMPLE_TIME)
    pelt_pid2.setSetPoint(int(TARGET))

    therm = chan0.value         # read the analog pin of the first thermistor
    therm2 = chan1.value        # read the analog pin of the first thermistor

    global DIFF
    if DIFF == 0:
       global ORIGINAL_DIFF
       DIFF = time.time() - start_time

    DIFF = 0
    elapsed_time = round(time.time() - start_time - DIFF, 2)
    minutes, seconds = divmod(elapsed_time-start_time, 60)
    print("Time elapsed")
    print(elapsed_time)
    timeList.append(elapsed_time)

    # Thermistor 1
    degrees_f = adc_to_degrees(therm, 1)
    print("Therm. 1")
    print(degrees_f)
    t1_tempList.append(degrees_f)

    # Thermistor 2
    degrees_f2 = adc_to_degrees(therm2, 2)
    print("Therm. 2")
    print(degrees_f2)
    t2_tempList.append(degrees_f2)
```

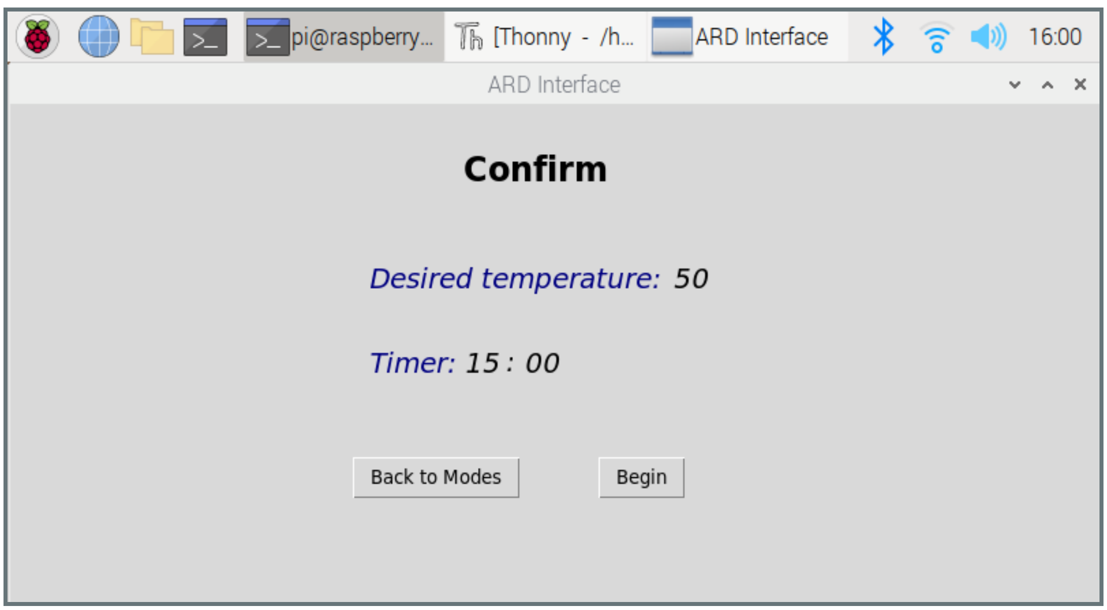
```
    # Thermistor 1
    updatePID(degrees_f, pelt_pid, dac, 1)

    # Thermistor 2
    updatePID(degrees_f2, pelt_pid2, dac2, 2)

def endProgram():
    """When called, sets DAC output to 0 to turn off peltiers and produces
graph of temperatures."""
    dac.normalized_value = 0.0
    dac2.normalized_value = 0.0
    graphData(t1_tempList, t2_tempList, timeList)
```

4. User Interface TKinter Frames

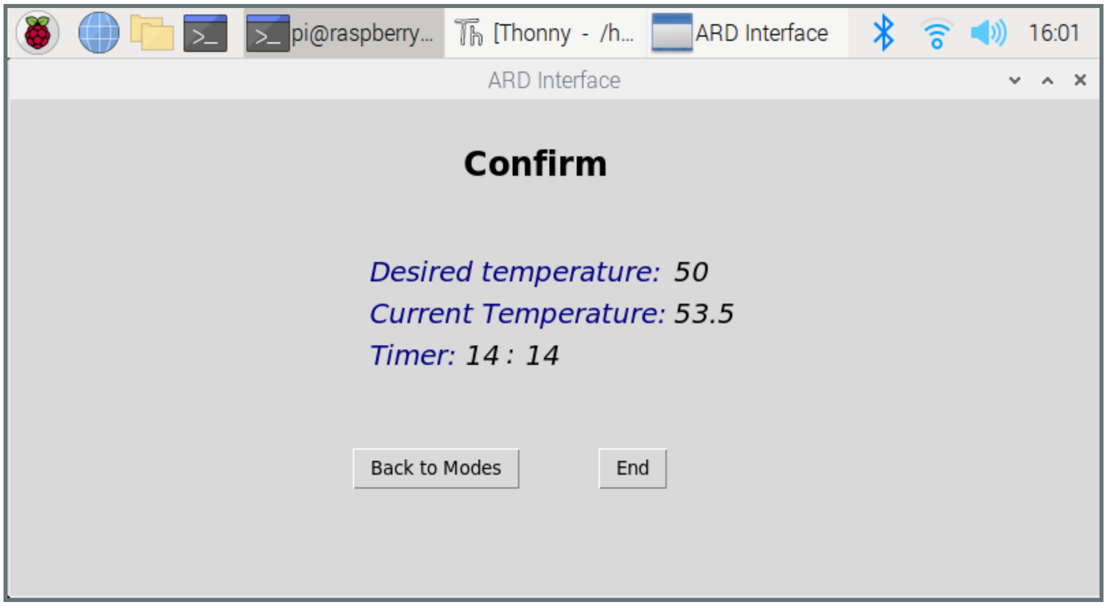
Frame 1: Start Page



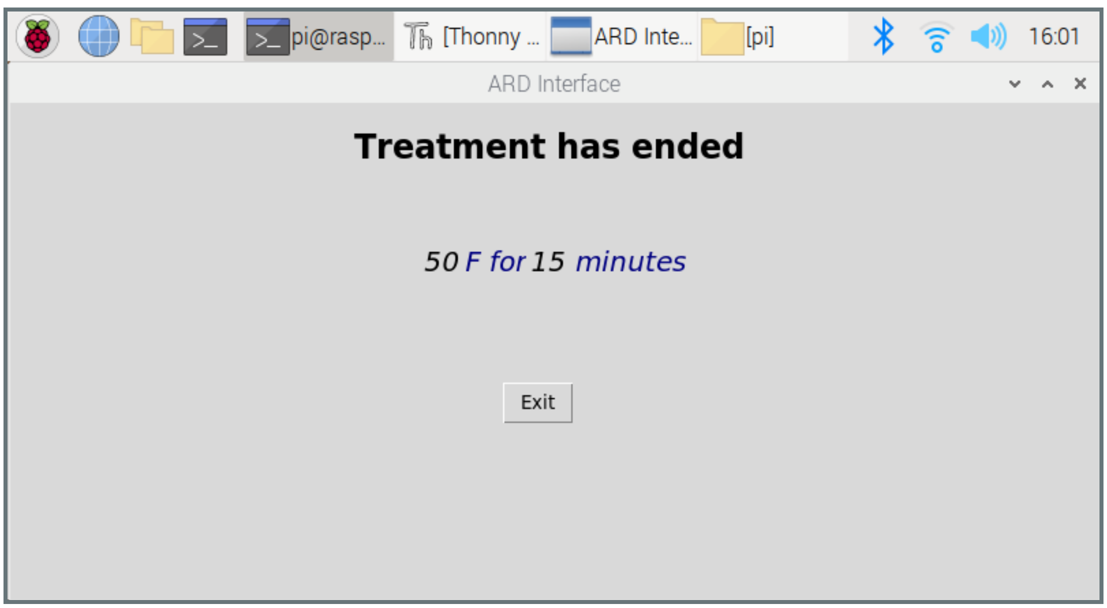
Frame 2: Treatment Mode Selection



Frame 3: Confirmation (Before Treatment)



Frame 4: Confirmation (During Treatment)



Frame 5: Treatment Summary

5. Final Costs (Actual Spending)

Item	**Supplier**	**Cost**
Pressure cuff	Amazon	$16.55
Peltiers*(8A)	Digikey	$47.56
Heatsinks	Newark	$61.94

Ribbon Cable	Adafruit	$8.00
FlexiKold Gel Ice Pack	Amazon	$16.99
Copper bar, (1/2" x 2" x 6")	McMasters	$16.75
5x Perf Board	Digi-key	$34.28
2x 20AWG wire (1 yellow, 1 black)	Digi-key	$20.16
3 HSF-55-33-B-F heatsinks with fans *	Newark	$37.00
4 Fan guards	Digi-key	$2.50
1 DC-power supply SE-600-12	Digi-key	$92.94
1 Power cable 211018-06	Digi-key	$13.60
1 18 AWG - 24 Conductor - 600V - Stranded Conductor (quantity: 5 feet of 24 conduction wire)*	Show me cable	$13.50
2 Buck Converter 12v to 5v, DROK 5A USB Voltage Regulator DC 9V-36V Step	Amazon	$9.99
1 AmazonBasics USB Type-C to USB-A 2.0 Male Charger Cable, 3 Feet (0.9	Amazon	$8.10
Sunnyglade Plastic Box (11.8"x9.8"x4.7")*	Amazon	$27.99
1 Copper Bar(⅛”X2”, 3feet long)*	Mcmasters	$16.75
Gel pack	Amazon	$6.99
Analog to Digital Converter for Thermistor output to raspberry pi input (ADC)	Amazon	$7.49
10W Resistors(10 pack)	Amazon	$9.99
Peltier Coolers	Amazon	$31.95
LLPT Double Sided Thermal Conductive Strong Adhesive Tape	Amazon	$15.99
HSF-55-33-B-F (heatsink)	Digi-key	$79.41
IRL520NPbF (MOSFET)	Digikey	$4.85
7 circuit terminal block	Digikey	$5.98
20 AWG wire	Digikey	$10.98
Total		$618.23