Western University Faculty of Engineering
Studio Section 17 – Chris Urbaniak
Team Identifier: 17Phone

Team Members: Hamza Elkababji, Saad Al-Bayaty, Zachary Ward, Muneeb Asif, William Osborn, Damanpreet Basra

Phone Warmer

A compact and lightweight solution to maintain ideal phone temperature whilst being easy to use and accessible.

Figure 1 - Final Product

1 Need / Challenge

People with movement disabilities such that they have limited reach and limited fine motor skills need a way to reliably and inexpensively keep their phone warm enough to maintain an optimal temperature while away from their body outdoors because their disability prohibits them from keeping their phone on their body to warm it to its optimal temperature range, so its battery life and performance decrease from the cold.

2 Final Design Documentation

Physical Device:

Our final design combined a variety of different elements available separately to the user into one compact device. Through a thorough examination of potential clients' needs and wants we were able to invent this physical device (Figure 1) to meet all crucial criteria.

Key Features:

As shown in Figure 2 the main design that stands out is the sloped slot for the phone. This grants the client the ability to use the phone much easier while they are in a wheelchair (with a tray in front of them) since they will not feel the need to adjust their seating to reach for or lean over the phone. At the same time, the model is kept at a conservative 36° (Figure 3) to drastically lower the possibility of the phone slipping out. Finally, once covered with the heating element, it allows for a protected space for the electric circuit as shown inside the back of the device (Figure 5).

A smooth base was designed to place non-slip grips that are evenly spread out to all edges (Figure 4). This will ensure that the entire phone warmer does not slip off any tray it is placed on no matter the material. The grips can also be easily removed if the client does not like the extra elevation.

Furthermore, a larger slot (16.2 cm as shown in Figure 3) for the phone allows for compatibility with different phone sizes including some iPhones, and many Samsungs, such as the client's S9, or the S9+. Although the slot is bigger than some phones (such as the S9), the phone still sturdily stays in place, as seen in Figure 1. Moreover, we included an edge on the front as a stopper for the phone, as well as a smaller edge at the back to stop anything from falling out of the back, (Figure 3) however, a consumer-ready version of the prototype would have the back compartment fully closed off with a removable cover.

Also shown in Figure 5, there is a temperature sensor that sends a signal to the Arduino which regulates the heating elements based on the outdoor temperature through Arduino code.

The heating element shown in all final product photos is a thin fabric that takes a minimal amount of space and generates enough heat (up to 50° C) to warm up the phone from the back, letting the heat soak through to the battery.

Lastly, whilst staying close to the recommended budget of the client, we were able to purchase an external power bank that can be used for various reasons. As shown in the upper right corner of Figure

1, you can detach the power bank when you are not using the phone warmer to charge any mobile device or recharge the power bank itself. Furthermore, you can also use it for charging your phone while you are using the phone warmer and still have an extra port available if need be.

For a more detailed view of the OnShape Model that was 3D printed along with a full assembly of the prototype with realistic dimensions of the Arduino, Relay, and extra space for wiring, click here.

Physical Assembly:

Using the OnShape model that has been provided within the "Physical Device" section, the first step that needs to be taken is to 3D print the body of the phone warmer. This can be done using a PLA-plus material. Once done, you will have a strong structure for the phone warmer.

Next, you must check the logic of the circuit that you will build using software such as Tinker CAD to make sure the following is possible. You can trial and error the Tinker CAD software to test out different voltages/currents from a power bank with different power receivers that are equivalent to the heating element and the Arduino. A basic diagram that can be experimented with is shown in Figure 6.

Then you must gather all materials listed in the "System" part of the documentation to start building the electrical circuit. Following the circuit provided in Figure 7 very precisely will give you the correct outcome of a working circuit for the phone warmer. There are three types of wires in this circuit: Positive, Negative, and Data. The positive wire will start from the power bank providing enough voltage and current (Power = Voltage * Current) to every element in the circuit since they are connected in parallel. Then, there are negative/ground wires coming out of every element, essentially giving all excess charges a safe place to go so they do not end up harming the client or creating a safety hazard. Finally, there are two data wires included in the circuit. The first date wire is the signal output from the Arduino to the Relay to tell the Relay to open and close the loop to the heating element at any time based on a certain code (mentioned later). This wire will be connected from pin 8 on the Arduino to the signal (S) on the Relay. Then, the second wire that must be connected is the temperature sensor's output pin to the A1 connection on the Arduino so that it can send voltage readings to the Arduino and perform operations based on it.

Once you understand what the circuitry does, you can start cutting/stripping up the wires and connecting them together on a breadboard. All of the nodes (3+ connections on one point) that you see in Figure 7 are going to be on the same line on a breadboard. Now that you have the circuitry in place and it has been tested on a breadboard, the next step is to solder all the wires together to make them permanent and ready to be placed in the body. Soldering the wires is as simple as taking all the wires on the same lines (negatives with negatives and positives with positives) and connecting them together by heating up the solder on top of the copper wire ends with a soldering iron.

Next, it is time to utilize the data wires that are in place to code the Arduino to execute certain programs on repeat. This can be done by installing the Arduino IDE launcher and connecting the Arduino using the printer cable provided in the Arduino kit. The goal of this Arduino is to use an if-else-statement to open and close the relay loop based on the exterior temperature. The second you finish coding the Arduino and upload the code, it will always run the code on a loop (when connected to a power source) if it is placed under the void loop() part of the code, and anything placed under the void setup() part will only run once. Through a lot of research on similar projects and Arduino tutorials listed under the "References" section of this report, along with plenty of trial and error in the IDE, you will be able to come up with a similar code to Figure 9 if you decide to include the temperature sensor Figure 9 if you do not. Furthermore, in these figures, there are comments on every line explaining what each one does.

Moreover, the reason you can choose to go with either version of the code while not cutting on any safety precautions is because of the delay placed on the void loop. There is a small delay in Figure 9 as we want it to check more frequently to make sure any inconsistencies from the readings of the temperature sensor do not affect the functionality of the circuit over a long period of time, and therefore can go back to consistent temperature readings; thus, executing the code accurately. There is a small delay in Figure 9 as we want it to check more frequently to make sure any inconsistencies from the readings of the temperature sensor do not affect the functionality of the circuit over a long period of time, and therefore can go back to consistent temperature readings; thus, executing the code more accurately. However, there is a larger delay in Figure 8, because there are two parts to it, turning on the relay and turning it off. Often, turning off the relay will cause the heating element to lose heat much quicker than the time it would take to heat up when turned on. To counter this, we placed a larger delay on the High switch and a lower delay on the Low switch. For an overall diagram of how the code interacts with the power source see Figure 10 - Flow Chart 1.

Lastly, you can now take everything (the circuit and the body) and assemble them together. The entire circuit will be able to fit into the hole designed for its placement at the rear bottom of the 3D-printed holder. Then, the heating elements can be taped onto the front and their wires passed through the hole in the slope (which will lead to the Arduino). Then connect everything together (The heating element female connector to the Arduino circuit's male connecter, and the Arduino's USB to the power bank). Follow Figure 1 for a general idea of what it should look like.

System

Part	Specifications	Source
 JAYO PLA 3D Print Filament	• 1.7mm PLA Filament • Dimensional Accuracy 0.02mm • Black in colour • 1.1kg roll (about 24% used in the prototype)	Click Here

 Arduino	• Input Voltage Between 7-12V • 5V I/O Voltage • 20 mA DC Current • 14 Digital Input/Output Pins • 6 Analog Inputs • 8 x 5.99 x 0.99cm • 27.22 Grams	Click Here
 5V Relay	• Operating Voltage of 5V • Trigger Current of 5mA • Maximum Load of 250V/10A (AC) and 30V/10A (DC) • 50 x 26 x 18.5mm Dimensions	Click Here
 Heating Element/Heating Pad	• 35℃-50℃ Temperature Range • 5V Minimum Voltage • 7.72 x 2.36 Inch Dimensions	Click Here
 Power Source / Power Bank	• 13800 mAh Battery Capacity • Lithium Polymer Battery • 3A Current Output • 14.99 x 7.37 x 1.5cm Dimensions	Click Here
 Rubber Feet	• 1 Inch Diameter, 0.25 Inch Thickness • 1.13g Weight per Pad • Rubber Exterior Material	Click Here

3 Testing and Validation

3.1 Compliance Matrix: Constraints

#	Constraint	Compliance Assessment	Rationale/Evidence
1	Must be between 300-400 Grams	Met	The final weight of the prototype was 347 grams which meets the criteria.
2	Must fit the size of the Samsung S9 dimensions (147.7 x 68.7 x 8.5 mm)	Exceeded	Not only did the dimensions correctly fit the Samsung S9, but was also able to fit phones with even larger dimensions such as the Samsung S9+, and even various iPhones, up to a maximum dimension size of 16.2 x 9 x 1.5 cm as shown in Figure 3, after an extrusion of 9 cm.
3	Must be less than $75	Not Met	Final Prototype Cost (Post-Tax): $7.05 for the 3D print filament used $3.47 for the relay $27.11 for the Arduino $11.37 for the heating element $3.10 for the anti-slip feet $4.40 for the temperature sensor $31.58 for the external battery **Total: $88.08**
4	Must be able to warm phone for travel periods that span 1.5 hours long	Exceeded	The phone warmer was connected to the power bank and left to see how long it would stay on, and it lasted 5.5 hours before it shut off due to running out of charge.
5	Phone must stay between 0° C and 35° C	Met	Testing was conducted, where the phone was placed on the phone warmer (in a freezer), and throughout the entire test, the temperature never went below 0° C nor over 35° C. The Arduino's code makes sure to keep the heating element on while the temperature is less than 30° C to avoid any unsafe temperatures that could cause a hazard.

#	Constraint	Compliance Assessment	Rationale/Evidence
6	Must not slide off wheelchair's tray when moving at 8km/hr (average wheelchair move speed)	Met	The average human jogging speed is about 8 km/h. We tested by putting the phone warmer on a tray (with a phone in heating position on the warmer) and sprinting, to simulate a wheelchair moving on a bumpy road. Throughout the test, the phone warmer stayed in place due to the non-slip feet implemented in our design (except in one of our trial runs, where the phone slipped).
7	The client's phone must be able to slide into place with a low friction coefficient to ensure ease of placement	Met	The heating element we have covering the slope of the phone warmer is encased with a cloth-like material, which allows for the smooth plastic of a case/smooth metal on the back of a phone to easily slide onto the phone warmer. This is due to the material-based friction coefficient between the heating element's cloth and the phone case, which is extremely low.

3.2 Objectives Evaluation

#	Objective	Evaluation	Evidence
1	Keep the prototype approximately 175g	Not Met	The phone warmer was put onto the scale, and it was not able to get to the value point of 175g, but rather a weight of 347 grams. The bulk of the weight came from the 3D print filament (which was 218 grams). This could be reduced by decreasing the infill density, however, that will decrease the structural integrity of the phone warmer.
2	Try to keep the cost of the final prototype to less than $60	Not Met	After taxes, the total cost came out to $88.08 making it $28.08 over the objective. The external battery was $31.58 making it the bulk of the expense. The battery can be replaced with a battery source that is of lower quality/has less mAh. This would have a lower cost, but the phone warmer would not be able to stay on for as long, depending on which cheaper alternative was purchased. However, the lower cost objective could potentially be met this way. We can also start to offer different phone warmers based on customizability of the client's wants.
3	Keep the phone warm (maintaining 22° Celsius)	Partially Met	Our Arduino code regulated the temperature of the phone warmer to the ideal temperature range for the Samsung S9 (between 0 to 35 Celsius)

#	Objective	Evaluation	Evidence
4			however, the temperature often fluctuated within the desired range, but did not stay at an exact temperature. Due to external factors (i.e., wind, fluctuating temperatures, etc.) it is very hard to maintain an exact temperature, but if the ideal temperature is maintained then the overall required outcome is met.
4	Keep the phone warmer functioning for any length of time the user desires	Mostly Met	After testing the battery source, it can provide ~5.5 hours worth of power to the system before it dies and requires charging. Though this supplies enough power for the client's intended usage, in the rare scenario that the client wishes to use the phone warmer for a longer duration of time (such as 7 hours), the battery will fail to do so. The duration of time that the phone warmer stays on directly relies on the capacity of the power bank that is used, so purchasing a power bank with larger capacity can extend the duration that the phone warmer lasts.
5	Keeping components inside dry in case of rain	Partially Met	Besides the battery that was on the outside, the group simulated rainwater falling over the prototype by pouring droplets over it, and after reconnecting the battery to the system to turn it back on, everything worked fine indicating that the electronics inside remained dry.
6	Having a non-slip material on the bottom to prevent prototype from moving	Exceeded	Tests were conducted in the exact same way as in the constraint version of this objective. However, it was at faster speeds than 8 km/h.
7	Keep heat insulated by applying cover to the front of the phone warmer (active and insulated heating combined)	Not met	Although we created a working prototype to keep the phone warm, we did not manage to implement a cover to the front of the phone warmer to keep the heat insulated. If we implemented this, we would have gone way over budget, which would not be ideal as the budget constraint was very important to our client.
8	Building upon the last objective, it should also be a see through cover to still be able to access or view the phone	Not met	This adds on to the last objective – to add a cover that will insulate heat and have a transparent screen that allows a user to see/be able to use their phone. As with the above objective, including this would go way over our client's budget, which was a very important constraint that we had to watch out for.

4 Comparison

4.1 Comparison 1

Winter Thermal Phone Case / Insulating Phone Case / Store Page

Practicality Comparison	- Competitor's product costs less at $45.09 shipped compared to our current $88.08 design. - Our design sits directly on the user's tray in front of them while the competitor's design must attach to a bar which may be out of reach. - Our design can be easily placed onto or picked up from the wheelchair (without sliding due to nonslip pads) while the competitor's product must be fastened and unfastened from the wheelchair. - Competitor's product has limited access to the accessories (headphone jack, volume buttons, etc.) of the phone. - Meanwhile our product allows for ease of access to most accessories even with limited hand movement. 
Comparison of Strengths	- Our design allows the phone to be easily placed on or taken off the holder for our user (who has cerebral palsy) while the competitor's product must be unzipped, which would be extremely hard. - Our design offers portable charging from the portable power bank.
Comparison of Weaknesses	- Competitor offers a much lighter weight object (67 grams compared to our 347 grams) . - Competitor's product requires a low maintenance from client as there is no battery to charge and no concern regarding if water will damage the product. - Competitor's product last for a longer time because of less mechanical components.

4.2 Comparison 2

PHOOZY XP3 Series Ultra Rugged Thermal Phone Case / Insulating Phone Case / [Store page]

Practicality Comparison	- Competitor’s product is cheaper at $50.84 compared to our final product at $88.08. - Competitor’s product has nothing to prevent it from sliding on our user’s wheelchair while our design has nonslip pads. - Competitor’s product is much simpler to use, not requiring charging or potential electronics maintenance. - Competitor’s product has an internal stash product to store other goods, such as wallet, cash, etc. 
Comparison of Strengths	- Our design is usable while being heated while the competitor’s product is not since the phone disappears into the phone case. - Our design keeps the phone easy to grab while the competitor’s design may cause the user (who has cerebral palsy) to struggle to remove the phone from the case. - Our design heats the phone while the competitor’s product only acts as an insulator.
Comparison of Weaknesses	- Competitor’s product has space to store other goods, while our design caters only to a user’s phone. - Competitor’s product has no mechanical parts and has a simple insulating design, which could potentially allow it to last longer. - Due to the competitor’s simple insulating design, no charging of any components is required, thus allowing it to be ready for use at any time, rather than needing to charge a battery.
Other Comparisons	- Our design and the competitor’s product are larger/ smaller in different ways, at 13.1 x 9.5 x 9 cm for our design and 17.1 x 1 x 9.1 cm for their product.

5 Potential Improvements

If our team were to work on this project further, we would use a different microcontroller, choose a different battery, put a plastic covering over the entire project, and make other small changes.

To begin, we would use an Arduino Nano Every rather than the Arduino Uno Rev 3 we used. Doing this would mitigate the issue of our high budget by lowering the microcontroller cost by $15.10 (Arduino store), which would help us meet our $75 budget. It would also lower the weight of the board by 20g (Arduino store) in turn allowing the user to easily transport it. The Nano can operate our circuit as well as the Uno can, so there would be no decrease in performance.

Next, we would use a smaller battery. Choosing a smaller battery would allow us to fit the battery into the holder so it is protected from the rain which is important for safety. It would also lower our budget to push us further below the $75 limit since our user expressed that they would prefer it to be cheaper, and this helps us stay competitive in the market. A smaller battery also reduces our weight further below 400 grams which is important because the lighter our product is, the easier our user can handle it.

Additionally, we would cover the phone and phone holder edges with a thin plastic which has a hole at the bottom to take out the phone after our user's trip. This would address the safety concern of our current design where rain could fall onto the phone or get into the electronics, causing short circuits. This would also act as an insulator for the phone, trapping heat beneath the plastic, and compensating for the smaller battery we would use. The heating element does not get hot enough for melting the plastic to be a safety concern. The plastic would ideally be thin enough to allow for the phone to still be used through it, but not thin enough to tear.

Smaller improvements with low cost and low weight include adding a Velcro (which our user has verified being able to use) to hold down the phone in place in case of bumps, changing filaments to potentially decrease costs, and putting the temperature sensor in the back of the holder, rather than outside, to keep it safe.

6 References

[1] *Arduino Forum*. [Online]. Available: https://forum.arduino.cc/. [Accessed: 07-Apr-2023].

[2] C. Team, "Everything you need to know about arduino code," *circuito.io blog*, 01-Jan-2019. [Online]. Available: https://www.circuito.io/blog/arduino-code/. [Accessed: 07-Apr-2023].

[3] "If," *if - Arduino Reference*. [Online]. Available: https://www.arduino.cc/reference/en/language/structure/control-structure/if/. [Accessed: 07-Apr-2023].

[4] M. James, "Arduino course for absolute beginners - tutorial 03," *Programming Electronics Academy*, 28-Jul-2020. [Online]. Available: https://www.programmingelectronics.com/tutorial-3-arduino-ide-and-sketch-overview/. [Accessed: 07-Apr-2023].

[5] M. James, "What is serial.begin(9600)?," *Programming Electronics Academy*, 17-Apr-2021. [Online]. Available: https://www.programmingelectronics.com/serial-begin-9600/#:~:text=Serial%20begin%20is%20used%20to,require%20the%20serial%20print%20function. [Accessed: 07-Apr-2023].

7 Appendix A – Examples of Design Documentation

7.1 Flowchart

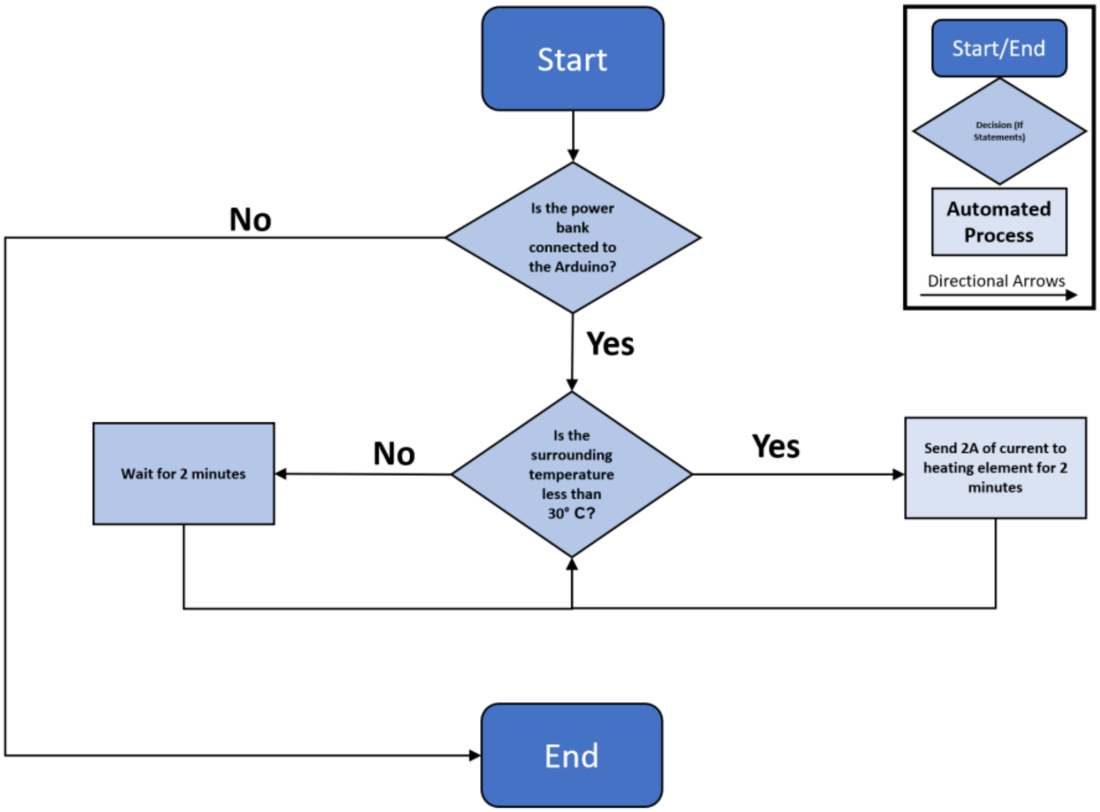

Figure 10 - Flow Chart 1

7.2 CAD Models or Drawings

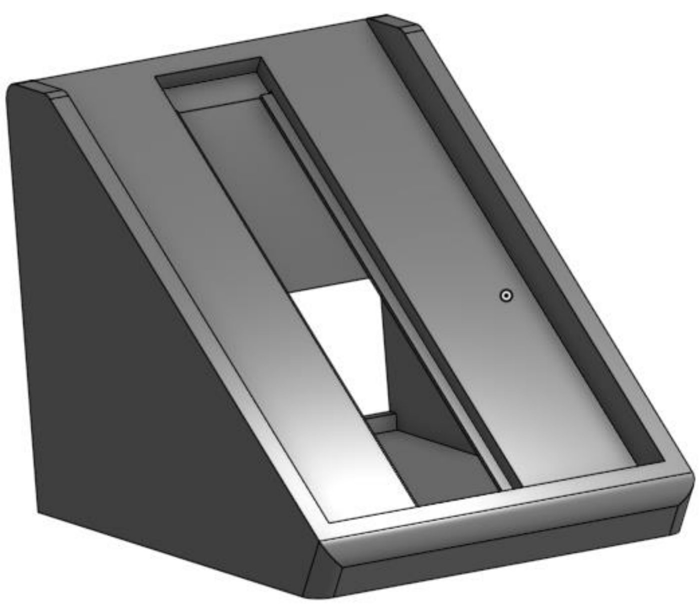

Figure 2 - 3D Printed Part

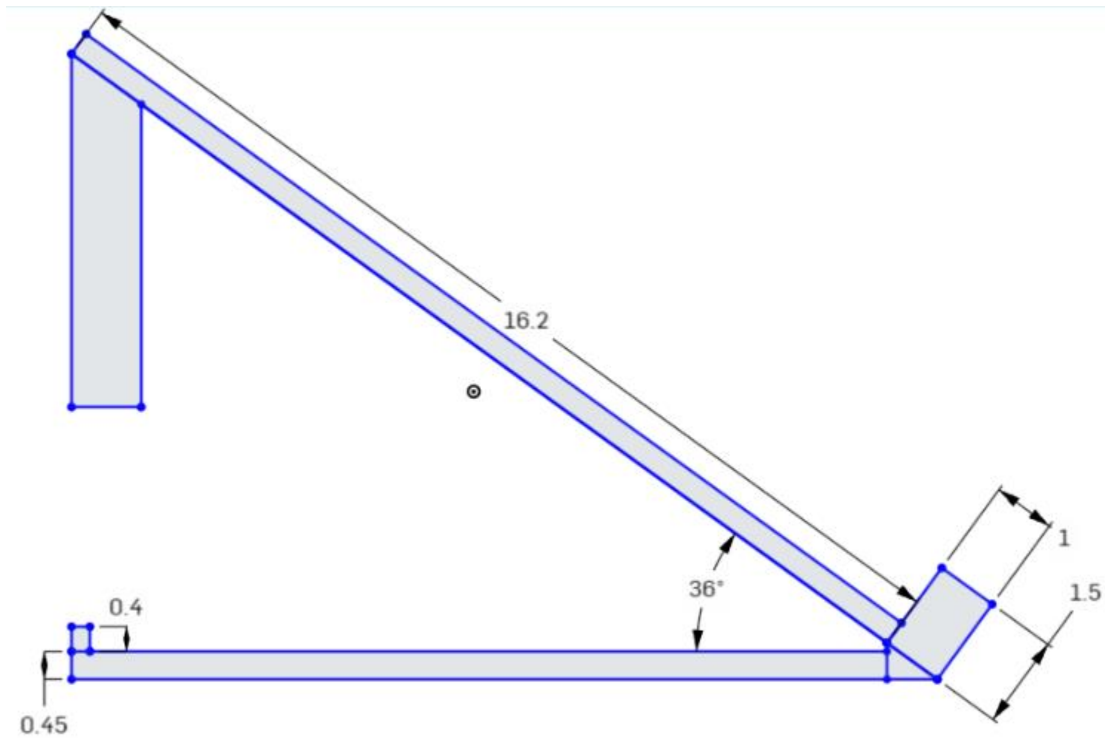

Figure 3 – OnShape Sketch

Figure 5 - Back View with No Cover

Figure 4 - Bottom View

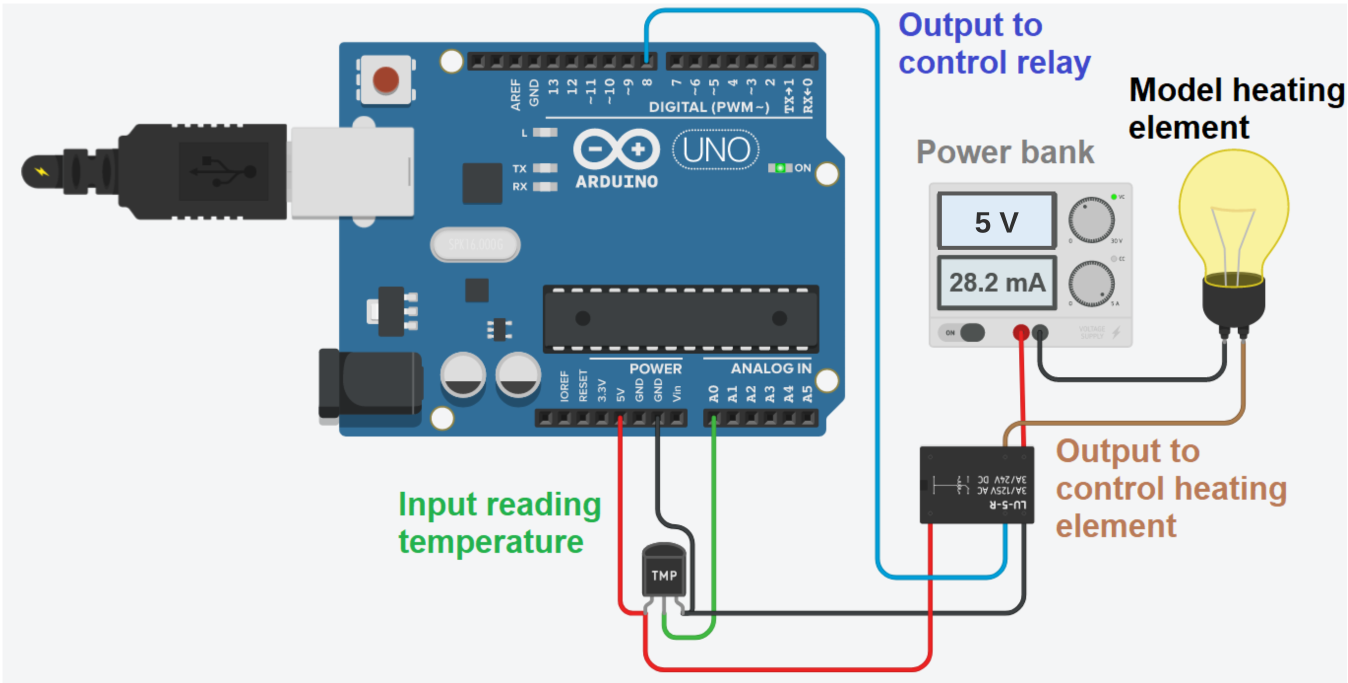

Figure 6 - Tinker CAD Model

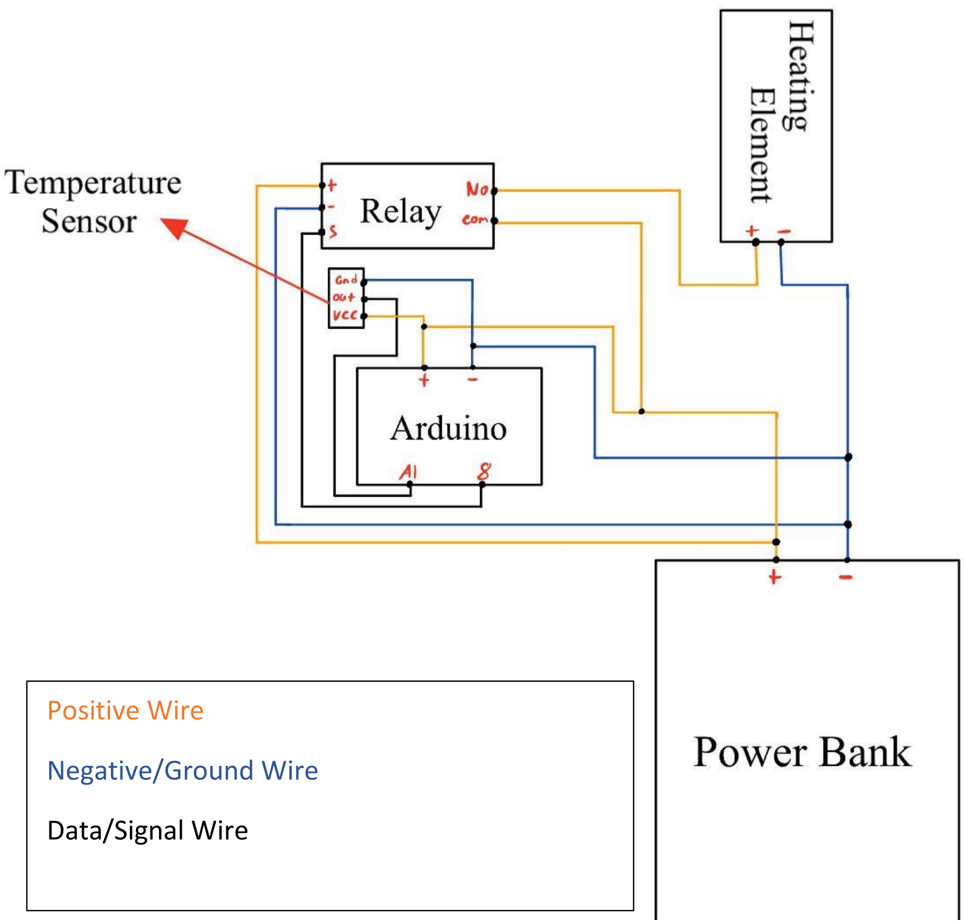

Figure 7 - Electric Circuit

```
const int LM35 = A1; // Declaring which pin we are using on the Arduino for the serial sensor input.
const int relayPin = 8; // Declaring which pin we are using on the Arduino for the relay output.

void setup () { // The code in this scope will only run one time.
  Serial.begin(9600); // Starting the serial input on the Arduino at a rate of 9600
  pinMode (relayPin, OUTPUT);
}

void loop () { // The code in this scope will run infinitely since it is in a loop.
  float lmvalue = analogRead(LM35);
  // Setting the value that the Arduino reads from the temperature sensor to a variable.

  float tempr = lmvalue*(500/1023); // Converting the voltage it reads to degrees celsius.
  Serial.println(tempr); // Printing out the temperature value calculated to the screen .

  if (tempr < 30){ // Executes the next line if the value we calculated is less than 15 degrees C.
    digitalWrite (relayPin, HIGH); // This will close the loop in the relay, thus turning everything on.
  }
  else { //Executes the next line if the value we calculated is greater than 15 degrees C.
    digitalWrite (relayPin, LOW); // This will open the loop in the relay, thus turning everything off.
  }
  delay(3000);
  // Whatever part of the if-else statment is executed above will be run for 120000 ms or 2 mins.
}
```

Figure 9 - Temperature Sensor Code

```
const int relayPin = 8; // Declaring which pin we are using on the Arduino for the relay output.

void setup() { // The code in this scope will only run one time.
  pinMode(relayPin, OUTPUT); // Declaring the relay connection to the Arduino as an output.
}

void loop() { // The code in this scope will run infinitely since it is in a loop.
  digitalWrite(relayPin, HIGH); // This will close the loop in the relay, thus turning everything on.
  delay(120000); // The above line will execute for 120000 ms or 2 mins.

  digitalWrite(relayPin, LOW); // This will open the loop in the relay, thus turning everything off.
  delay(30000); // The above line will execute for 30000 ms or 30 seconds.
}
```

Figure 8 - No Temperature Sensor Code

7.3 No App/Website or Interface For Our Project.