Electric Thermos Box

Team: Zerui An (zeruian2), Tingfeng Yan (ty7), Celine Chung (mwchung2)
ECE445 Project Proposal - Spring 2020
TA: Roumu Hao

1 Introduction

1.1 Objective

Most drinks taste differently at different temperatures, and it is certainly nice to enjoy drinks at the temperatures they taste the best[1]. However, most drinks are only available at their storage temperature, which is either room or refrigerator temperature. There are a few ways to change the temperature of drinks, like using a microwave or refrigerator, but none of these methods are precise. Besides, not everyone has access to these devices. For example, a dormitory resident probably won't buy a refrigerator if food storage is not needed, even if it's affordable.

To address the inconvenience in controlling the temperature of drinks, we plan to develop an Electric Thermos Box. This product would offer users a simple way to bring any drinks to any temperature they want (within some range, of course). Our product would be especially beneficial to (1) people like dormitory residents who do not have access to other heating and cooling devices; and (2) people who want precise control of the temperatures of their drinks.

1.2 Background

Although some companies (Ember, etc.) have been developing a temperature controllable mug, most of the existing thermos cups only have functions of heating the liquid, and they could take as long as an hour to get the drinks ready according to user reviews[2]. Also, there is a possible problem with burning users' hands since the heating modules are usually exposed to air.

The device we are developing would not only be able to heat up drinks but also to cool them down. Furthermore, we expect our device to be much more efficient than the ones currently on the market.

1.3 High-level Requirements

- Our product should be able to bring 250g of water to any desired temperature (set by the user) within the range 0-50℃ starting from room temperature (20-30℃).
- The entire process should take no more than 20 minutes.
- The final temperature should have an error of at most ±5℃ compared to the desired temperature.

2 Design

2.1 Block Diagram

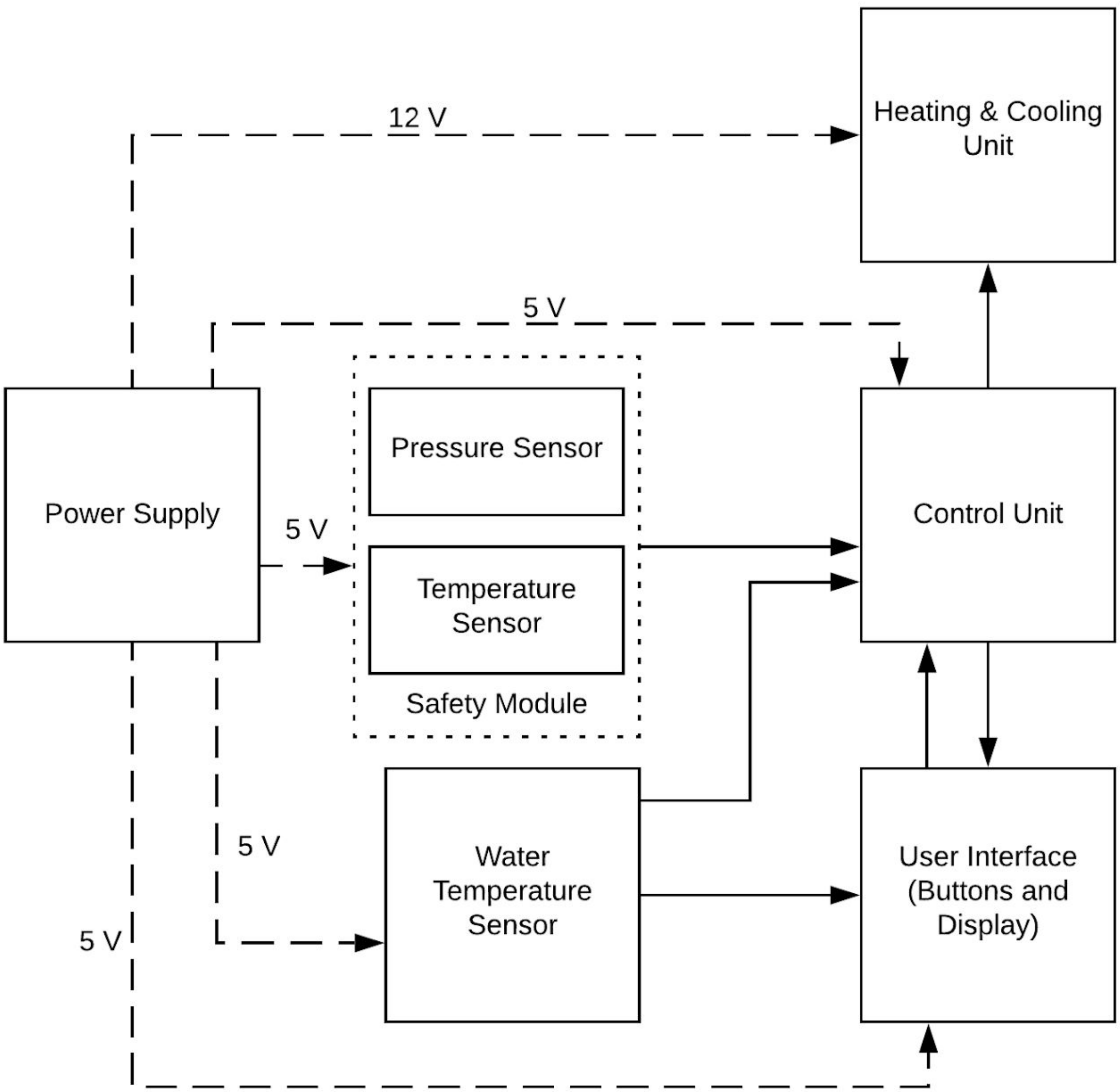

Figure 1. High-Level Block Diagram

As shown in figure 1, once the user sets the desired temperature by pressing the buttons, the control unit would compare it with the current temperature, and send control signals to the temperature control unit to perform either heating or cooling.

2.2 Physical Design

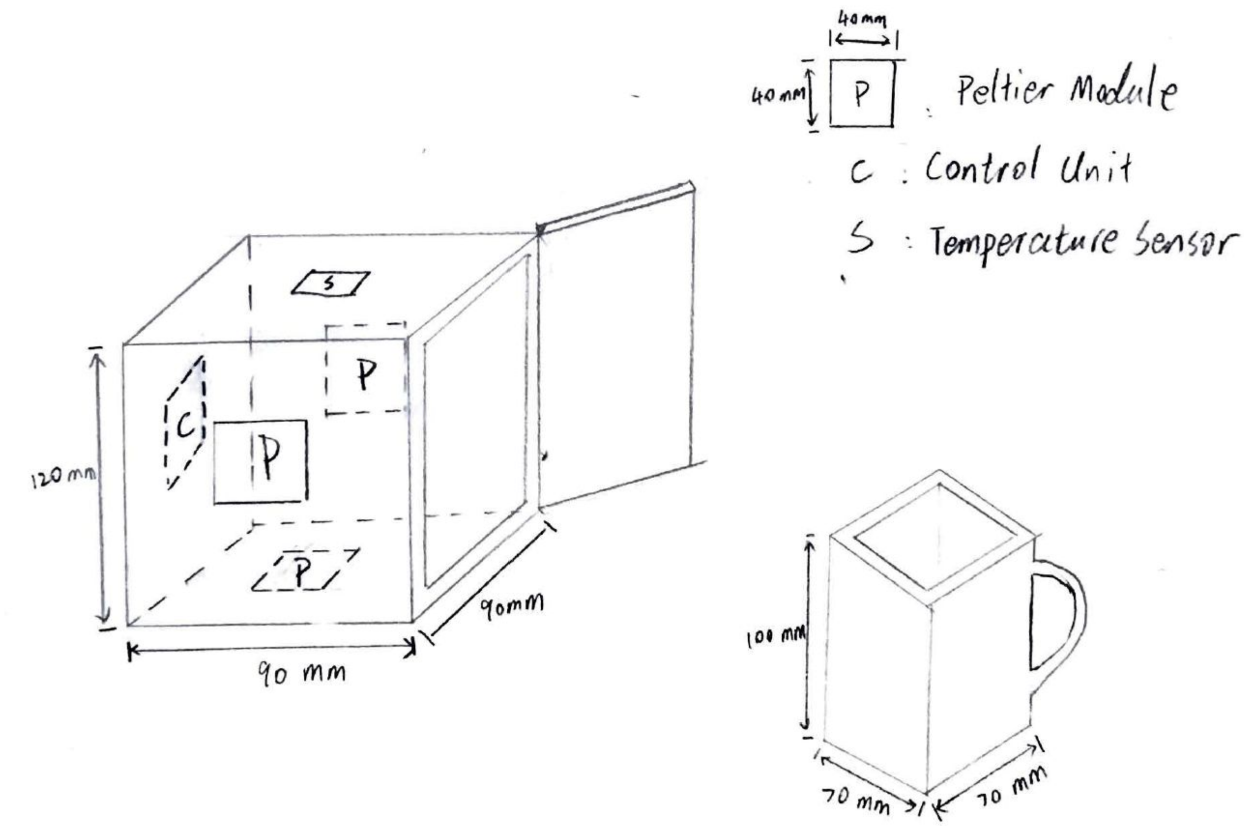

Figure 2. Physical Design

We plan to design a box-shaped heater/cooler and a cup of the same shape. To heat or cool the drink, the user must first place the cup inside the box, then use the buttons to set the desired temperature. We plan to put all the electronics on the box. Therefore, this box should ideally work with any container that fits in the box, but with lower efficiency compared to our original cup.

We plan to install the user interface (including buttons, number pad, and display) on the top of the box. The current design assumes that we use 3 Peltier modules. If we end up needing a 4th one, we will put it where the control unit originally was, and move the control unit to the top of the box.

2.3 Block Descriptions

2.3.1 Heating & Cooling Unit

Power: 12V
Input: control signals from the control unit
Outputs: none

The heating & cooling unit has 3 modes: “heat”, “cool”, “idle”, corresponding to when it’s heating/cooling the liquid or doing nothing. The mode is controlled by the control unit. We plan to use Peltier modules from the TEC1 family. We will use the same set of modules to perform both heating and cooling. We will use H-bridges to achieve this.

The heating & cooling unit is supposed to generate enough heat in heating mode and be cold enough to absorb heat fast in cooling mode.

Requirement	Verification
1. Able to raise or lower the temperature of 250g of water by 10℃ within 7 minutes.	1.a Add 250g of room temperature water to a container, then attach our module to it. 1.b Use switches to simulate the 2-bit logic input from the control unit. Send “heat” or “cool” signal to the module. 1.c Monitor the temperature of the water using a digital aquarium thermometer. Record the time it takes for the water to rise or drop by 10℃ and compare it to the requirement.

2.3.2 Control Unit

Power: 5V
Inputs: Temperature data, User requests, and is_safe
Outputs: Control signals to the heating & cooling unit and user interface

The control unit takes 3 inputs, 2 of which are used in temperature control: “desired temperature” from user inputs, and “current temperature” from the sensor. After comparing these two temperatures, the control unit then sends a signal to the temperature control unit to perform the corresponding task.

The control unit also takes an input (“is_safe”) from the safety module. This signal indicates if the system is operating normally. If not, the control unit would stop any ongoing process until the system is back to a normal state.

The control unit should be able to stop all activities of the heating & cooling unit when receiving “0” from the safety module.
When receiving “1” from the safety module, the control unit should correctly compare the current temperature and the desired temperature, and then send a proper signal to the heating & cooling unit and let it switch into the right mode.

The Control Unit outputs 2 bits, where “1x” means “idle”, “00” means “heating”, and “01” means “cooling”.

Requirement	Verification
1. (Top priority) If safety module output (‘is_safe’ signal) is 0, the output should be the ‘idle’ signal. 2. When the current temperature is at least 2℃ below the desired temperature, the output should be the ‘heating’ signal. 3. When the current temperature is at least 2℃ above the desired temperature, the output should be the ‘cooling’ signal.	1. Use a switch to simulate the input (logical 0 or logical 1) from the Safety Module. See if the Control Unit always outputs the “idle” signal when input is 0. Set the input to 1 when verifying requirements 2 and 3. 2. Set the desired temperature to 2℃ above the current temperature. Check if the Control Unit outputs the “heating” signal. 3. Set the desired temperature to 2℃ below the current temperature. Check if the Control Unit outputs the “cooling” signal. ● Notes: ○ the output of the Control Unit is 2-bit logic and can be monitored using 2 LEDs ○ “Setting the desired temperature” can either be done using the user interface if it is implemented and tested or simulated using the I/O pins on Arduino.

2.3.3 Water Temperature Sensor

Input: 5V from the power supply
Outputs: water temperature reading for the control unit and user interface

This module measures the temperature of water (or any other liquid inside the container), and sends the measurement to the control unit. We are planning to use a contactless sensor for this module.

Requirement	Verification
1. The error should be within 3℃ (compared to a thermostat measuring the water directly). 2. The precision should be 0.5℃ or higher.	1. 2. Use an Arduino board and the provided code in the sensor library to get the sensor readings. Place the sensor at least 5cm above the surface of a cup of water, also place a digital aquarium thermometer inside the water. Compare their readings. Repeat this process using cold, room temperature, and hot water. ● Note: cold water should be at least 10℃ below room temperature, and hot water should be at least 10℃ above.

2.3.4 User Interface

Power: 5V
Inputs: temperature data from the water temperature sensor, and control signals from the control unit
Output: User requests to control unit

We plan to have the following components in our user interface:
An on/off switch that controls the power supply.
A 3*4 keypad for entering the desired temperature.
An LCD display to show both the desired and current temperature at least 2 digits.
A 3-color LED indicating the current state(red when heating, blue when cooling, and green when drink is ready)

Users use the 3*4 keypad to set the 2-digits of desired temperature freely (within the range 0-50℃).
Then, press the # key to confirm the temperature on the display.
If users want to change the desired temperature, press and hold the * key for more than three seconds to enter the new desired temperature.

Requirement	Verification
1. Current temperature display should be 2(or more)-digit decimal integer, and equal to the measurement rounded down. 2. The desired temperature should be displayed as a 2(or more)-digit decimal integer between 0-50. 3. 3-color LED should be red when desired temperature is at least 2℃ above the current temperature, blue when at least 2℃ below, and green when the temperatures differ by no more than 2℃.	1. Use an Arduino board to simulate the sensor readings, and send the 8-b logic type data to the user interface. Check if the displayed data is equal to the integer part of the data sent. 2. a. Use the number pad to set various temperatures between 0-50. Check if the display is as expected. b. Use the number pad to set some temperature above 50. This action is expected to fail (one should not see a number greater than 50 being displayed). 3. Use methods described in 1 and 2 to set desired and current temperature. Try different combinations and make sure to cover all 3 cases mentioned in the requirement. Check if the 3-color LED always shows the right color.

2.3.5 Power Supply

Outputs (power): 5V for the control unit, thermostat for water, user interface, pressure sensor and temperature sensor in safety module, and 12V for temperature control unit

Supplies 12V for each Peltier module; 5V for other modules.

Requirement	Verification
1. To ensure an operating voltage around 5V, the power supply must provide an open-circuit voltage in the range of 6.5-7.5V.	1. Use an oscilloscope to monitor the output. Check if the voltage always stays in the range 6.5-7.5V.

2.3.6 Safety Module

Power: 5V, low current
Input: none
Output: is_safe signal (0 or 1)

The safety module will include two parts:

(1) A temperature sensor that monitors the temperature of the circuit;
(2) A pressure sensor that measures the weight of items in the box.

When the circuit temperature is too high, or weight is too small (when trying to heat an empty container), the safety module would signal the control unit to enter the stop (idle) state.

Requirement	Verification
1. The Safety Module must output a “0” when the circuit temperature is above 60℃, or a “1” when temperature is below 50℃. 2. The Safety Module must output a “1” when the weight inside the box is at least the weight of container plus 120g, or a “0” when the weight inside is less than the weight of container plus 80g.	1.a Load enough weights first to make sure this verification is not affected by requirement. 1.b Place the module near a cup of boiled water, along with a digital indoor thermometer. 1.c Monitor the output using an LED until the thermometer reading drops below 50℃. 2. Attach an LED to the output. Load enough weights to see “1” being output by the Safety Module. Remove the weight and check if the output turns to “0”.

2.4 Schematics

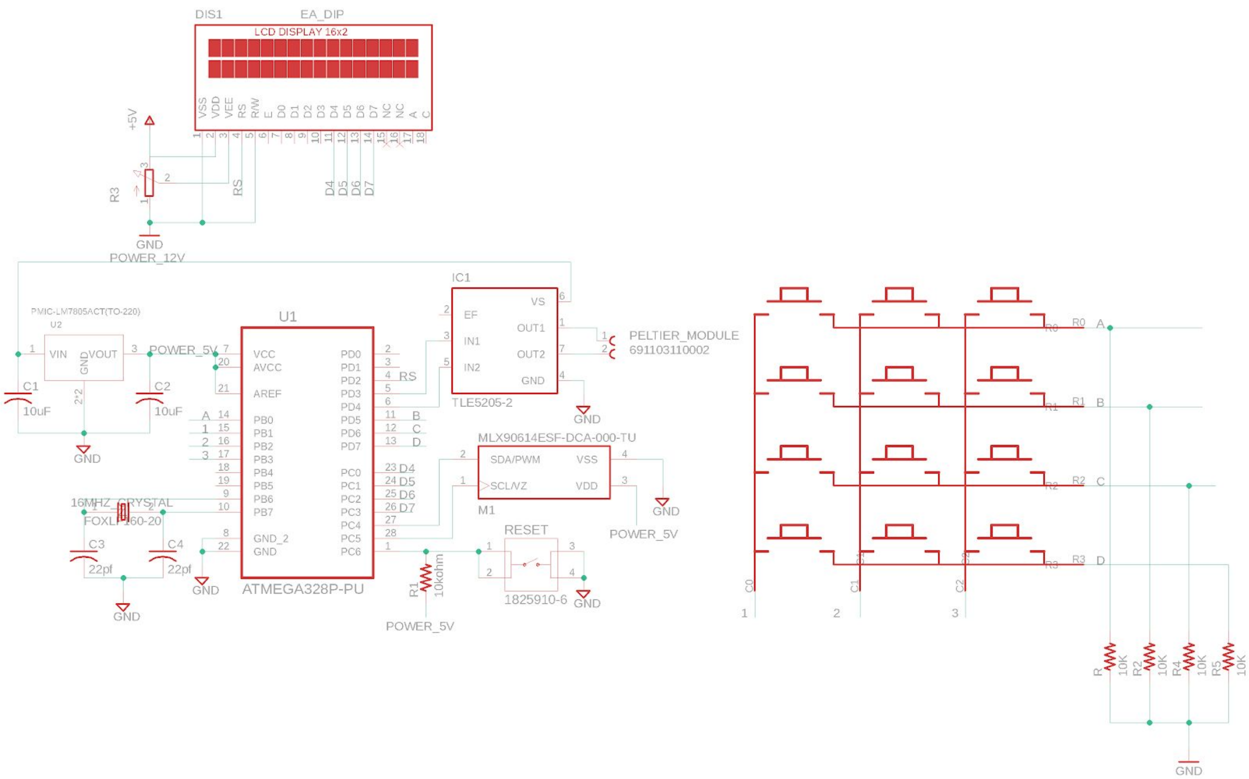

Figure 3. Schematic

2.5 State Diagram

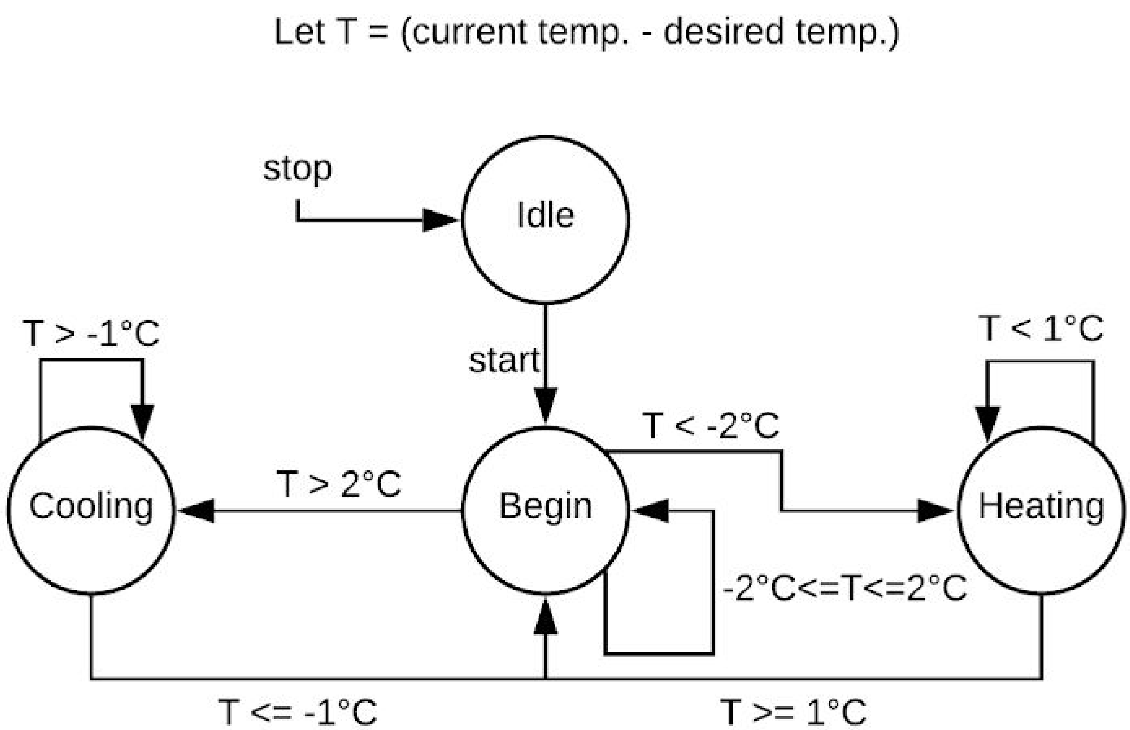

Figure 4. State Diagram for Control Unit

2.6 Calculation and Simulation

Given that water has a specific heat of 4.2kJ/(kg*℃), it takes about 1kJ to heat 250g of water up by 1℃. Our temperature control module is expected to heat up 250g of water by 10℃ within 7 minutes, so the required power is 10kJ/420s = 23.8W. The actual value would be higher since we also need to take efficiency into consideration. According to the datasheets, each piece of TEC1-12706 has 50W power. The overall efficiency should be 15.9% if we use 3 pieces, or 11.9% if we use 4.

3 Tolerance Analysis

According to our high level requirement, the final temperature should have an error of at most ±5℃ compared to the desired temperature. This error range is directly related to our control logic and the accuracy of the sensor used to measure water temperature. We plan to have the control logic try to keep the liquid temperature within desired temperature ±2℃, therefore we can tolerate a ±3℃ error in the sensor measurement.

The power supply includes a DC-to-DC converter which generates 5V for all modules except the Temperature Control Unit, so we plan to design one generating about 7V without load, so the resulting voltage level is expected to be 5V. However, 4V is also enough to support those modules so we can tolerate a ±1V error in the DC-to-DC converter.

As an important part of the safety module, the pressure sensor only wants to tell whether the container is empty or not. In case of accidents, we want the safety module to output "1" if the weight measured is larger than the weight of the container plus 100g. 100 is a rough number, 80 or 120 do not make a large difference. Thus we can tolerate a ±0.2N error in the measurement of the pressure sensor.

Temperature sensor uses 55℃ to discriminate between high and low temperatures. A range between 50℃ and 60℃ is pretty reasonable so we will tolerate a ±5℃ error in the measurement of the temperature sensor.

4 Cost Analysis

4.1 Cost of labor

We assume that all people working on this project are compensated $50/hour. We will each be working on this project for about 10 hours a week for 11 weeks. The estimated machine shop work time is about 10 man-hours.

$$3 \ * \ \$50/hr \ * \ 10hr/week \ * \ 11 \ weeks \ * \ 2.5 \ + \ 10 * \$25/hr \ = \ \$41,500$$

4.2 Cost of parts

Part	Cost (prototype)
4x H-bridge (Mouser; TLE5205-2G)	$17.40
4x 2 Position Terminal Block	$6.24
DC Barrel Jack	$1.09
12V 20A Power Supply (LEDwholesalers)	$41.99
16 AWG wire (GS Power;16AWG25RB)	$6.95

3x4 Matrix Keypad (Adafruit;100486)	$6.99
Infrared Temperature Sensor (MLX90614)	$16.99
Microcontroller (Mouser; ATMega328P-PU)	$2.08
Various Logic ICs	$10 (estimated)
Peltier Modules (TEC1-12706)	$15.99
Total	$124.72

Grand Total = $41,624.72

5 Schedule

Week	Zerui An	Tingfeng Yan	Celine Chung
02/24/20	Write design document; Implement water temperature sensing module	Write design document; Implement 5V power supply	Write design document, Implement User Interface module
03/02/20	Finalize physical design	Implement safety module	Implement control unit logic
03/09/20	PCB design; Test User Interface; Implement control unit circuit	PCB design; Test water temperature sensor; Continue working on safety module	PCB design; Test power supply; Continue working on control logic
03/16/20	Continue implementing control unit circuit	Implement heating & cooling unit	Test safety module
03/23/20	Connect all parts together; Test the entire design.	Test control unit logic	Test heating & cooling unit
03/30/20	PCB testing and troubleshooting	PCB testing and troubleshooting	PCB testing and troubleshooting
04/06/20	Finish the final PCB design	Finish the final PCB design	Finish the final PCB design

04/13/20	Reserved for unexpected delay		
04/20/20	Mock demo	Mock demo	Mock demo
04/27/20	Prepare for demo	Prepare for demo	Prepare for demo
05/04/20	Final demo, Final report	Final demo, Final report	Final demo, Final report

6 Ethics and Safety

There are several safety concerns about our project. The Peltier modules we are planning to use require high current (reference datasheet). Currents as large as a few amps could easily burn the circuit if wires are not chosen properly. To avoid such accidents, we would choose our wires based on the AWG standard: use AWG20 or lower wires for 5A paths, and AWG16 or lower wires for 20A paths[3].

Since our product works with liquids, we need to consider the case when the liquid spills out and causes a short circuit. To prevent a short circuit from causing damage, we plan to use fuse wires in the power supply module. When the total current exceeds the max working current, the power supply will be cut off.

The Peltier modules are essentially heat pumps, so overheating would be a big concern in cooling mode. The heat removed from the liquid doesn't just "disappear". Instead, it is simply transferred to the other side of the module. We worry that the accumulated heat might be a threat to our circuit. Therefore, we have a temperature sensor in our safety module, which would prevent the circuit from generating or transferring more heat at high temperatures. We also plan to use extra heat dissipation modules if necessary, like heatsinks and fans.

The high heat capacity of water (4.2kJ/℃) defines the high power nature of our product. Despite that, we still want to avoid wasting energy as much as possible. We do not want the user to heat up or cool down an empty container, so we also plan to add a pressure sensor or some equivalent device to the safety module. The system would not operate unless there's enough liquid inside the container.

References

[1]J. Michale, “The Perfect Serving Temperatures for Your Favorite Drinks” *DRINKFRIDGE,* 31-Oct-2017. [Online]. Available: https://drinkfridge.com/serving-temperatures-favorite-drinks/ [Accessed: 27-Feb-2020]

[2]M. Simon, “Ember Ceramic Mug and Ember Travel Mug reviews: Smart at home, less so on the road,” *TechHive*, 09-Jul-2019. [Online]. Available: https://www.techhive.com/article/3405525/ember-ceramic-mug-review-ember-travel-mug-review.html. [Accessed: 11-Feb-2020].

[3]“Wire Gauge and Current Limits Including Skin Depth and Strength,” *American Wire Gauge Chart and AWG Electrical Current Load Limits table with ampacities, wire sizes, skin depth frequencies and wire breaking strength*. [Online]. Available: https://www.powerstream.com/Wire_Size.htm. [Accessed: 11-Feb-2020].