A CHEAPER ALTERNATIVE FOR TEMPERATURE CONTROLLED SLEEP

By

Alex Dicheva

Patrick Wang

Wyatt Sass

TA: Luoyan Li

29 March 2024

Project No. 12

Contents

1. Introduction

1.1 Problem and Solution

Extensive research has been done on the impact temperature has on your sleep. Due to a decrease in the body's ability to thermoregulate, body temperatures decline throughout the sleep cycle. To allow for a faster decline in body temperature and to ensure relaxation, bodies prefer warmth before trying to fall asleep. Around 10 min before falling asleep, the body starts to decline in temperature; therefore, cooler temperatures allow the body to stay in sleeping mode, which drives bedroom temperature recommendations to be around 66-70 degrees [1]. Various efforts have been made toward temperature regulation while we sleep to improve the quality and length of sleep. One solution is a Smart Thermostat; however, this requires heating the entire room or the entire home. There are also temperature-controlled bed sheets or duvets, such as the BedJet ($1329), Smartduvet ($1555), and EightSleep ($2195) [2, 3, 4], but all of these are priced at over $1000, making them unaffordable for the average consumer to even consider trying out.

To achieve cheap and attainable temperature-controlled sleep, we will make modifications to an everyday heated blanket. Initially, we will add temperature sensors to the outer layers of the heated blanket to track the temperature of the blanket within its environment. Using a small IMU, we can determine when the user's movements slow down or stop, indicating that they may have fallen asleep. Using this data, we will create an app that allows the user to set a desired temperature for each part of their sleep routine. We will provide temperature recommendations, such as warm temperatures before sleep, a gradient decline in temperature based on the user's desired time of sleep and their actual sleep time, and lower temperatures throughout the night. Finally, the user will have the option to gradually increase the temperature in their preferred morning hours to make waking up easier and serve as an aid to an alarm clock. Ideal sleeping temperatures vary between individuals, so users will be able to adjust each of these parameters. To receive feedback on the quality and length of their sleep, users will be able to track sleep time and movements in their sleep provided by the accelerometer data. This data will be stored over time to allow users to see whether their temperature adjustments are in their favor and edit accordingly. Ultimately, this serves as a much cheaper alternative for users to optimize their sleep by improving upon existing products that are already commonplace.

1.2 Visual Aid

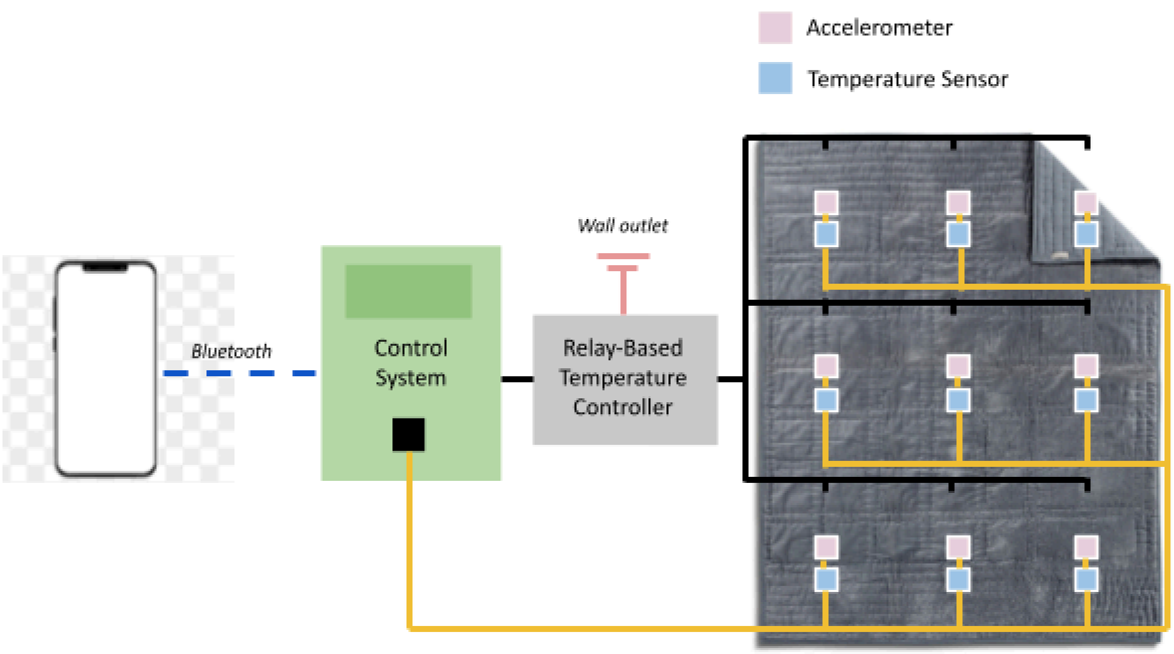


Fig. 1. Diagram components adapted from [5, 6]

1.3 High-Level Requirements

Our project will aim to complete the following three high-level requirements:

1. The user will be able to set the temperature remotely through an app, and the blanket will be able to control the physical temperature in response. Furthermore, they will be able to schedule a temperature, and the blanket will reach that temperature within 45 minutes. While the user lies under the blanket, the heating coils will reach the preset temperature (+/- 3 degrees Celsius), and that temperature will be presented on the app. The blanket will be able to maintain that temperature for over 10 minutes and adjust to the next target temperature under the same requirements.
2. When a user gets under the blanket, the blanket can detect the user's movements spanning greater than 3 inches (+/- 1.5 inches) or with acceleration greater than 10 m/s^2 (+/- 1 m/s^2). This data will be relayed to our app, and the controller can use this data to detect a "sleeping" state after a period of non-movement.
3. The temperature reading of the blanket should be broken down into 9 distinct zones. The blanket should be able to determine the absolute temperature within each zone (+/- 1.5 degrees Celsius), and it should be accurate within a 2-inch radius. This temperature will be displayed on the app.

2 Design

Our design will be broken down into three distinct systems, each with multiple subsystems of its own. The three main systems will be the Power, Control, and Heating systems. They will communicate with each other to achieve the high-level requirements. A block diagram of each system, its inputs, and subsystems, the way it will interact with the other systems, and descriptions of each system are shown in the following subsections:

2.1 Block Diagram

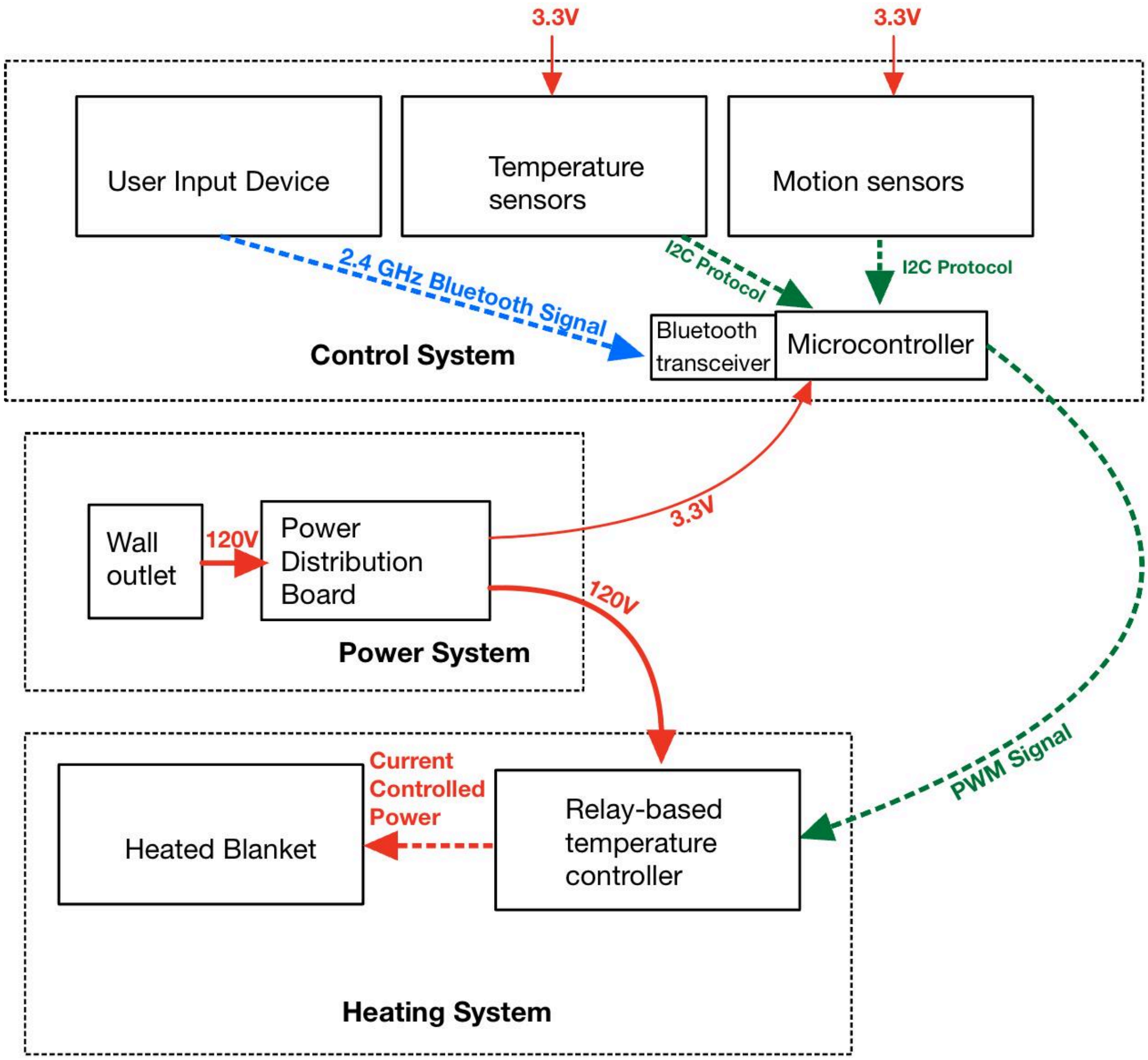


Fig. 2. Block diagram of our overall design.

2.2 Power Subsystem

The power subsystem will be responsible for supplying the other two subsystems with the power they need to function. The subsystem will need to plug into a wall outlet. The subsystem will supply the heating subsystem with around 120V AC and supply the control subsystem with around 3.3V DC.

2.2.1 Requirements and Verification

- The heating system will be in parallel and take on the full 120V of AC power. So, no voltage conversion will be needed for the heating system. A more specific discussion of the heating system circuit will be discussed in that subsystem description.
    - This subsystem will be important when implementing the heating system, especially if our goal is to keep our entire blanket at a steady temperature. We need to be able to provide a relatively steady 120V AC to each of the coils for each of them to heat up properly. If the magnitude of the voltage varies across coils, this will lead to uneven heating.

Table 1 Power System Verification

Requirement	Verification
Cord connecting a wall outlet, supplying 115-125V AC to a power distribution circuit for an expected current load of no more than 1.5A.	Measure and track the voltage w/ oscilloscope (30-60s), and see that it is in the desired range (115-125V AC). Determines magnitude validity and consistency of sine wave signal.
Utilize HLK-PM03 buck converter to produce 3.0-3.6V DC voltage for control subcircuit	Use a voltmeter to track voltage values of 3.0-3.6V (30-60s). Use oscilloscope to check that signal is DC.
The power distribution circuit will be used to run the other two subsystems in parallel at the aforementioned desired voltages.	Use configuration of resistors and capacitors to model the expected resistance and capacitance values of our subcircuit designs. Use oscilloscopes/voltmeters to determine each component is running at desired voltage (30-60s).

2.3 Control Subsystem

The control subsystem will use input from three different sources and use the input to send PWM control signals to the heating subsystem. First, the microcontroller will receive Bluetooth signal inputs from the user input device, which will most likely be an iOS or web application, in the form of a temperature/heating setting and optional inputs like whether the user is ready for sleep, how long they would like to sleep, how they would like the temperature to change throughout the night, and perhaps other options. The microcontroller will also be receiving inputs from temperature and motion sensors through packet-based protocols and will need to determine whether to raise the temperature or drop the temperature according to all three input sources. The microcontroller will then send signals to the heating subsystem to adjust the heat setting.

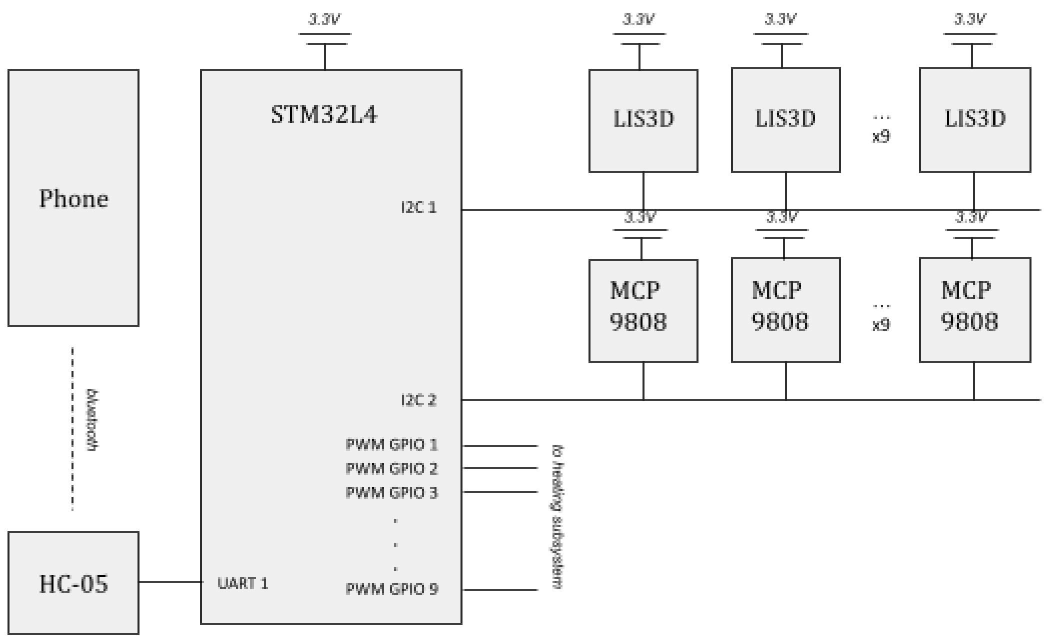


Fig. 3. Control circuit design around our microcontroller.

2.3.1 Requirements and Verification

- It will include a PCB with a microcontroller of the STM32F series, with at least 18 GPIO pins, and at least 9 PWM outputs, and it will also have an HC-05 Bluetooth chip attached to the serial IO.
- The blanket will be divided into 9 rectangular sections in a 3x3 pattern, each with a temperature sensor and an accelerometer in the center. These temperature sensors and accelerometers will be used to give feedback to the microcontroller.
    - Each accelerometer (LIS3DHTR) will contribute to checking whether the user is asleep based on whether the accelerometers record movement and this will contribute to the calculation of the optimal temperature to improve the users' sleep based on how far along they are in their sleep.
    - The temperature sensors (MCP9808T-E/MS) will feed input to the microcontroller to ensure that the temperature that is being achieved by the blanket is consistent with the desired temperature, and it will tell the microcontroller to adjust the temperature accordingly. Table 2 Control System Verification

Requirement	Verification
Powered by a 3.0-3.6V DC parallel network.	Use a voltmeter to track voltage values of 3.0-3.6V (30-60s), oscilloscope to check that signal is DC, and an ammeter to check that the provided current is >= 0.5A.
Output 9 distinct PWM signals, one for each block of the blanket, that feed into the heating system. Set each signal to a distinct duty cycle.	Check that our microcontroller can output 9 distinct PWM signals. Try to set each signal to a specific duty cycle at intervals of 10% (10%, 20%, 30%, … 90%). Use an oscilloscope to estimate the duty cycle of each wave. We want each wave to be within +/- 1% of the desired duty cycle (i.e. 19-21% for a 20% duty signal).
Each accelerometer will contribute to checking whether the user is asleep. Our goal is to report the user's movements spanning greater than 3 inches (+/- 1.5 inches). This should be visible in the app/	Test the instantaneous magnitude (and direction) of the accelerometer. Integrate to determine the distance traveled within a set time. Output the distance serially to a computer to check validity. Test accelerometer: flip and check magnitude; physically move it by set intervals (e.g. 1 in, 2 in, 3 in, 4 in). Check that measurements fall within the tolerance intervals specified by the requirement. Repeat for 9 accelerometers simultaneously.
Temperature sensors will feed input to the microcontroller to ensure that the temperature that is being achieved by the blanket is consistent with the desired temperature. Our goal is to determine the absolute temperature within each zone (+/- 1.5 degrees Celsius).	With just one temperature sensor, test that we can output a physical temperature using our sensors and microcontroller. Output serially to the computer. Next, use an external thermometer to determine the temperature (goal: sensor data within 1.5 degrees Celsius). Repeat for 9 temperature sensors at the same time.
The phone/web app communicates with the microcontroller through Bluetooth. The app sends temperature settings to the blanket that adjust PWM waves. The app receives temperature and movement readings from sensors.	Establish Bluetooth communication. Display the message to a computer upon successful connection. Test output: check that movement/temp. readings sent to the app are equal to computer readings during sensor testing. Test input from app to microcontroller: 1. Specify a heating temperature in our app 2. Track the temperature of the blanket with a thermometer. 3. Once the temperature stops changing significantly (+/-0.5 C/min), check the mean temperature for 2 min (goal: within 3 C of our desired temperature) 4. Increase the temperature of the blanket in the app by 5 degrees Celsius and repeat step 3.

2.4 Heating Subsystem

The heating subsystem will be directly responsible for the majority of the functionality of our heating blanket. The heating subsystem will receive signals from the control subsystem and change the heat emitted by the blanket accordingly. The relay-based temperature controller will receive PWM signals, and it will react to these signals by modifying the voltage being supplied to the heated blanket's heating supply. When coming up with the initial design, we took a look at MKDas's circuit from *LabProjectsBD,* but we eventually created our own version. An example of the circuit we plan to use is shown below.

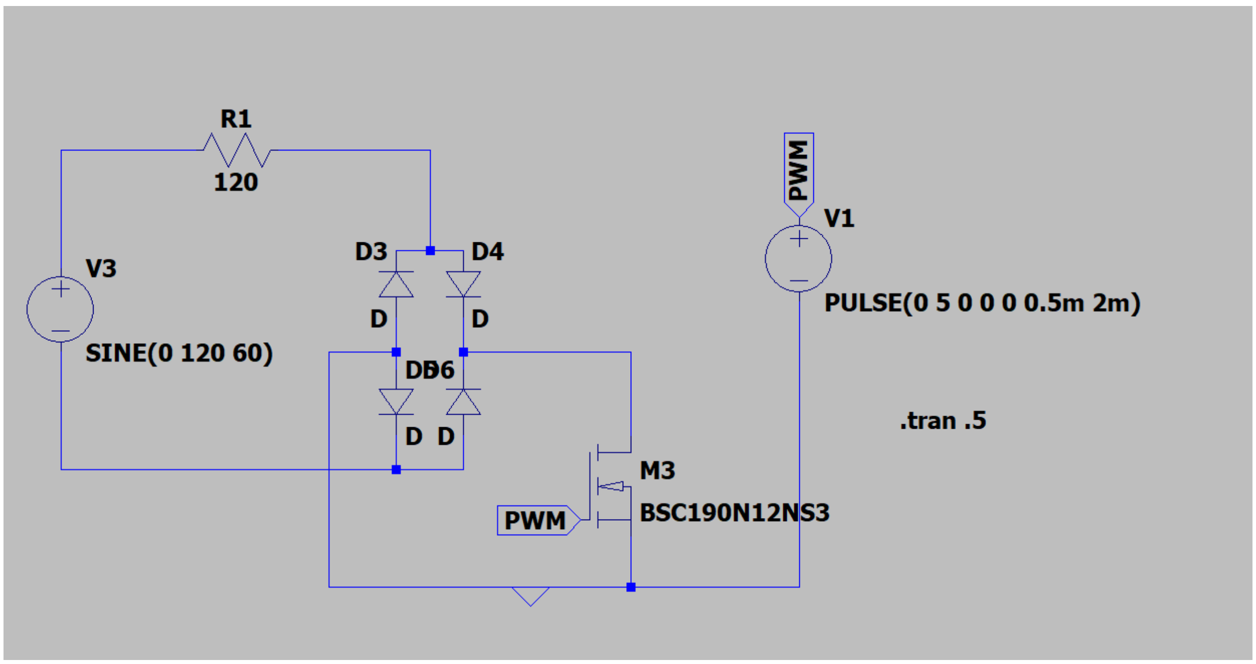


Fig. 4a. AC Pulse Width Modulation circuit using MOSFET.

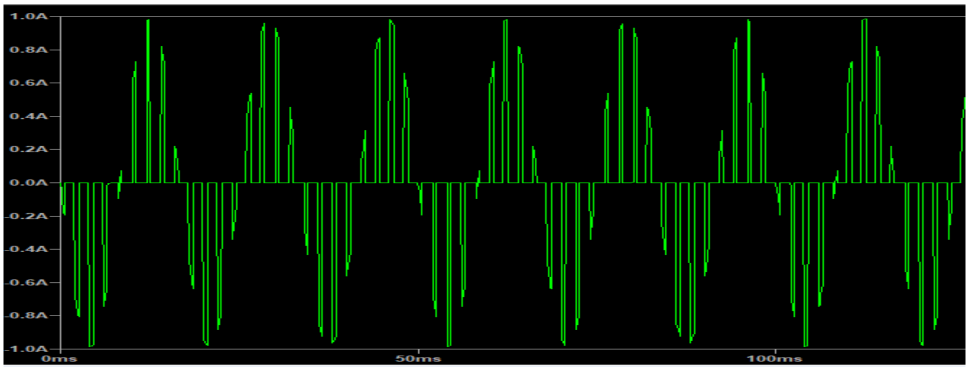
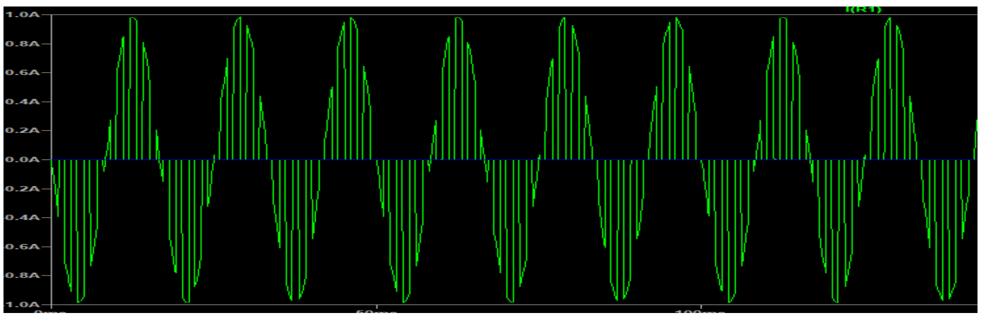
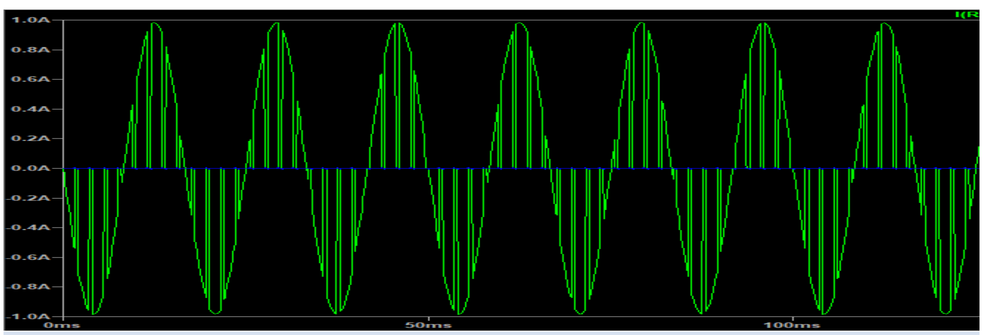
Duty Cycle (%)	Current Across Heating Coil
25%	
50%	
75%	

Fig. 4b. Voltage readings at various points in the circuit from Figure 4a.
Data is from LTSpice Simulations.

Here, we are using a Bridge Rectifier to modify our AC current into a purely positive signal, which can then be fed through our MOSFET. The label ‘PWM’ is the place where we plan to input the control signals coming in from our Control Subsystem. Here, the ideal behavior will be that a 50% duty cycle input signal to the gate of our MOSFET will cause our AC wave across our heating coil to be close to 50% as well. The specifics will be discussed in the Requirements and Verification section, but a visualization can be seen in Figure 4b.

In our actual, physical design, the heating coils will take the place of the resistor that is labeled R1 in the circuit above. We believe that a resistor is a good approximation for the heating coil, since the intended functionality is simply dissipating the absorbed power as heat into the environment, much like a standard resistor.

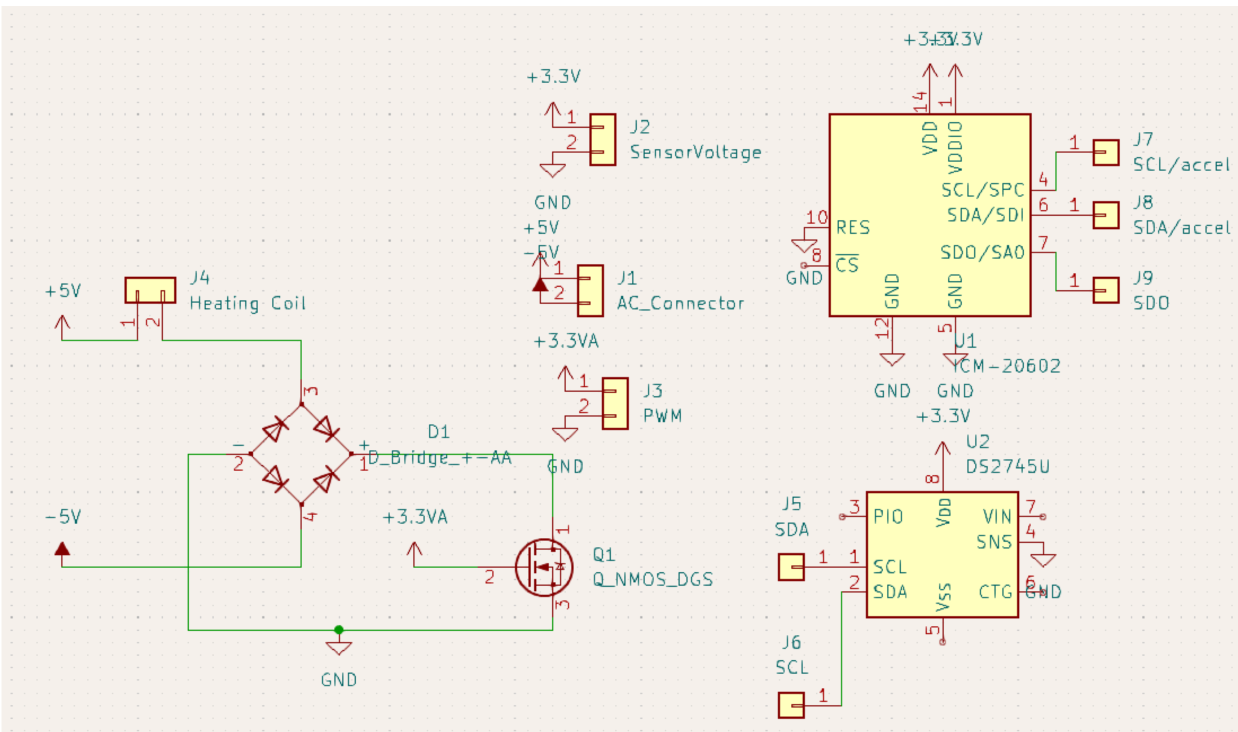


Fig. 5. Schematic for Heating Module in KiCAD.

We were able to create a schematic for the heating module that we plan on using for each heating zone on our blanket. To put this into perspective, in each heating zone, we would have a small PCB to house the heating circuit as shown above, as well as locations for our sensors which will interact with the control subsystem. The relevant heating module is on the left hand side of Fig. 5., and connectors for the sensors can be found on the right hand side of Fig. 5.

2.4.1 Requirements and Verification

Table 3 Heating System Verification

Requirement	Verification
We want our heating subsystem to be able to interpret the incoming PWM waves and to change its behavior depending on the signal. More specifically, we want the duty cycle of the AC current powering each of our coils to be within 5% of the duty cycle of the incoming wave. If our incoming duty cycle is 50%, we want our AC current to be modified to 45-55% duty cycle as well.	1. For just a single coil, input a 50% duty cycle PWM wave into the gate of our MOSFET in the power circuit as described in Figure 4 above. At this point, we want to use an oscilloscope to measure and track the voltage that is output to our load: the heating coils. The goal is to see our modified AC wave to have between a 45-55% duty cycle as well. 2. Change the input duty cycle to 25% and then 75% in turn. Repeat the same test as in part 1, and check that our duty cycles are in the correct tolerance range: 20-30% and 70-80% for 25% and 75% respectively.
Our blanket will ideally be able to modify all 9 PWM waves - for each of our 9 heating coils - all simultaneously. We recognize that the signals in the coils might interfere with each other, but ideally our end result will be within +/-10% of our ideal duty cycle for all coils at the same time.	1. For all the coils at the same time, input a 50% duty cycle PWM wave into the gates of all the MOSFETs in the power circuits for each heating coil. Then, use the oscilloscope to measure and track the voltage across all of our heating coils. The goal is to have each AC wave have between 40-60% duty cycle. 2. Repeat step 1 for input duty cycles at 25%, looking for our AC voltage to be within 15-35% across our heating coils. Do the same thing with our input at 75%, looking for 65-85% duty cycle across our heating coils.

2.5 Tolerance Analysis

Sensor Tolerance: In order for our blanket to achieve proper heating and motion detection, we need to make sure that our sensors are working properly.

Firstly, we need to check if our temperature sensors are sensitive enough to satisfy our high level requirements. We intend to use the MCP9808T-E/MS, which is small enough to fit multiple sensors on our blanket without being too noticeable. Realistically, we do not expect to heat our blanket outside of the 15°C - 50°C range. The most accurate operation range for our sensors is -20°C - 100°C, which means our expected temperatures should not pose a problem for our temperature sensors. Let us assume that due to random error, or due to uneven heating, our coils end up being 5% hotter than we intend them to be. This means that our coils could reach up to 52.5°C, which is still well within the standard operation range. At lower temperatures, this effect will be even less. As such, operating temperature will likely not be an issue for our temperature sensors.

For the sensitivity of the MCP9808T-E/MS, it should give us measurements within +/-0.5°C of the real temperature, and often closer to +/-0.25°C. This will be more than sensitive enough for us to achieve an ideal output tolerance of +/- 1.5°C.

Secondly, we need to check if our accelerometers (LIS3DHTR) will work for our use case. Firstly, checking the operating temperature range, -40°C - 85°C, we see that these sensors will have no problems at the temperatures we intend to work at. Let's say we set our accelerometers at their most sensitive setting. In this case, they can measure up to +/- 2g of acceleration, or around +/- 20 m/s^2. This will cover our high level requirement of movements with acceleration exceeding 10 m/s^2 in magnitude. Our ideal output tolerance for our accelerometers will be +/- 1 m/s^2, or around +/- 0.1g. At 25°C, the datasheet tells us to expect +/- 40 mg tolerance in our measurement, or +/- 0.04 g. At room temperature, our accelerometers will be able to easily achieve our desired accuracy. However, the datasheet also tells us to expect this tolerance interval to change with temperature at a rate of 0.5 mg/°C.

At our highest operating temperature of 50°C, this should mean our tolerance interval +/- 52.5 mg or +/- 0.0525g (40 + 25*0.5) for our maximum linear acceleration measurements of +/- 2g. As such, these sensors will be more than sensitive enough for us to achieve an output tolerance of +/- 1 m/s^2 for our acceleration measurements at all times.

3 Costs and Schedule

The following is a summary of our predicted costs as well as the schedule we will undergo to complete this design:

3.1 Cost Analysis

First we will perform a labor analysis. We estimate a total of 20 hours of work per team member, paid an hourly wage of $40/hr:

15 hr/person per week * 8 weeks * 2.5(overhead factor) * $40 /hr * 3 people = $36000

Next, we will perform a parts cost analysis:

Table 4 Parts Costs

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Quantity	Actual Cost ($)
Accelerometer - LIS3DHTR	DigiKey	$2.29	$1.76	12	$21.12
Temperature Sensor - MCP9808T-E/MS	DigiKey	$1.40	$1.40	12	$16.80
Carbon Fiber Heating Wire - 49 Feet	Amazon	$11.59	$11.59	1	$11.59
Blanket	Amazon	$8.99	$8.99	1	$8.99
Microcontroller - STM32L412	DigiKey	$4.62	$4.62	4	$18.48
Thermal Reset Cutoff - AD55ABB	DigiKey	$2.43	$2.20	10	$22.02
HLK-PM03 Buck x3	Amazon	$9.90	$9.90	1	$9.90
Bluetooth HC-05	Ebay	$7.49	$7.49	1	$7.49
Bridge Rectifier- ABS10A-13	DigiKey	$0.41	$0.41	15	$6.15
MOSFET - TK42A12N1	DigiKey	$1.43	$1.43	15	$21.45
Total					**$143.99**

Based on our labor and parts cost analysis, the total cost to make our temperature controlled blanket is **$36143.99**. While this is a significant cost, it is important to note that the significant labor cost is due to the trial and error process, and upon successful completion of the design and with the assistance of automated production processes, these costs will significantly go down, which will allow us to make the blanket cost effective.

3.3 Schedule

Table 5 Schedule

Week	**Milestones to be completed**
02/18-02/24	Design Document and Project Proposal regrade should both be finished (all members work together synchronously).
02/25-03/02	We will have our first design review and make any necessary changes to our design document. We will also do our first draft of our PCB, headed by Patrick, and bring it to the PCB review, and begin to order parts, headed by Alex.
03/03-03/09	We will revise our PCB design and order it if ready. Otherwise, refine the design and finish ordering all other parts we need (all members work together synchronously).
03/10-03/16	Because this is spring break, all members will work on the project asynchronous on their own time. All three members will contribute to starting the application portion of our project, as well as getting ready to begin manufacturing.
03/17-03/23	Alex and Wyatt will work together on code for the application, and Patrick will lead the code for the microprocessor but all three will work on it. We will begin unit testing sensors and other parts as they arrive as a team.
03/24-03/30	We will begin creating the circuit and testing all together as a team. We will also be polishing the application at this point. Each member will individually complete their individual progress report.
03/31-04/06	We will finish at least our first versions of each of our subsystems, and rigorously test them as a team. By now, each member will have a good idea of what portion of the project they can help the most with; Patrick will lead the electrical work and Alex and Wyatt will work together on the application.
04/07-04/13	The group as a whole will combine each subsystem and begin to test the project as a whole.
04/14-04/20	We will all work together on finishing the project testing and refinements, and attend our mock demo. We will also work together on the team contract fulfillment assignment and the first draft of our presentation.
04/21-04/27	We will work as a group to make the necessary changes suggested at our mock demo, and perform our final demo. We will also give our mock presentation and get the first draft of our paper done. Each member will individually finish their lab notebooks.

04/28-05/04	We will give our final presentation, submit our final paper, and submit our lab notebooks. We will also do the lab checkout and award ceremony.

4. Ethics and Safety

We will move forward with this project with safety as our number one priority. As outlined in both the IEEE and ACM Code of Ethics, we need to prioritize the health and safety of our potential users and make sure to disclose any possible dangers to anybody who could reasonably be affected by the project.

Before starting any manufacturing step, we will ensure we have fully researched any potential dangers and minimized the risk. This will include testing the range of possible temperatures that are achievable by our prototypes before we have anyone personally test them. This will also include testing the quality of our circuits and circuit components. Further, this will include incorporating safety limits, such as overcurrent and overtemperature protection, directly into our circuits. Fuses will be added so when our blanket gets too hot, it will forcefully shut the heating circuit down. When working with electricity, we will make sure that we are following proper safety procedures to avoid accidental discharge into the environment, users, or engineers. When working with the heating elements of our project, we will take caution in avoiding incidental contact between skin and heat by wearing protective clothing during development and ensuring that any user would avoid exposure to high levels of heat. We have also intentionally chosen heating elements that already come with a layer of insulation, so as to minimize this risk even further. Our ideal final product will not have heating elements exposed to the outside, where a user might accidentally touch them. We will also indicate high risk areas of the blanket to the users, such as where the power distribution circuit is located.

In terms of specific steps to take when working on our project, we will ensure that our circuits are disconnected from electricity when not in use to minimize the risk of something like a fire. We will only test our project in the presence of a fire extinguisher. We will never physically work on or test the project without at least two group members or other supervisors present. If at any point we feel that the amount of electricity or heat being produced by our project is uncontrollable or unsafe, we will immediately disconnect the power, step away from the project, and grab the fire extinguisher if necessary.

If we suspect there may be any risk of danger to the engineers or users, or if we suspect there may be any sort of violation of procedures and regulations, we will openly communicate concerns within the team, with the course staff, and with appropriate authorities. We will make sure to thoroughly inform anyone involved with the project of any potential concerns or health risks that they might be reasonably exposed to while working on the project.

References

[1] McCoy Abby, "Thermoregulation During Sleep: How Room and Body Temperature Affects Your Rest Quality," *Sleepopolis*, para. 3, "Long Story Short", January 25, 2024. [Online], Available: https://sleepopolis.com/education/thermoregulation-sleep/#:~:text=Once%20we%27ve%20fallen%20asleep,nearly%20stop%20regulating%20temperature%20altogether. [Accessed Feb. 7, 2024].

[2] BedJet, "BedJet 3 Dual Zone Climate Comfort Sleep System for Couples", *BedJet.com*. [Online], Available: https://bedjet.com/products/bedjet-3-d ual-zone-climate-comfort-system-for-couples. [Accessed Feb. 7, 2024].

[3] Smartduvet, "Smartduvet", *Smartduvet.com*. [Online], Available: https://www.smartduvet.com/products/smartduvet. [Accessed Feb. 7, 2024].

[4] EightSleep, "Pod Cover", *EightSleep.com*. [Online], Available: https://www.eightsleep.com/product/pod-cover/. [Accessed Feb. 7, 2024].

[5] Pngtree, "Mobile Phone Png Smartphone Camera Mockup", *pngtree.com*. [Online], Available: https://pngtree.com/freepng/mobile-phone-png-smartphone-camera-mockup _6067590.html. [Accessed Feb. 7, 2024].

[6] Demdaco, "Velvet Throw Blanket - Gray", *demdaco.com*. [Online], Available: https://www.demdaco.com/velvet-throw-blanket-gray/. [Accessed Feb. 7, 2024].

[7] MKDas, "Learn to regulate an AC load using PWM Signal", *labprojectsbd.com*, July 7, 2020. [Online], Available: https://labprojectsbd.com/2020/07/21/ac-load-control-using-pwm-technique/. [Accessed Feb. 22, 2024].