Retro Fit for Steam Heating

Coover Hall

Design Document

Team Members

Jevay Aggarwal	Technical Lead
Sarah Coffey	Weekly Report Lead
Thomas Devens	Project Plan Lead
Joseph Filbert	Client Contact Lead
Ken Wendt	Website Master
Liz Wickham Kolstad	Design Document Lead

Client: Leland Harker

Contents

1. Introduction

1.1 Acknowledgement

Leland Harker is our client, providing funding and guidance for this project. We would like to thank him for the opportunity to work on this project.

1.2 Problem Statement

Coover was built in the 1950s to be the Electrical Engineering Building. After over 60 years, it has only been renovated significantly in 1999 to include the addition of the Active Learning Complex. With that being said, there has not been a significant renovation of the steam heating system since it was built (“Coover”). While the steam heating system allows the heat to be generated off-site in a power plant across campus and transferred through pipes into the various offices, labs, and classrooms in Coover, the system lacks the ease of use that modern day technology offers. The steam is controlled through a number of valves distributed throughout the building. The system cannot react to changes in outside temperature or adjust to faculty’s choice in temperature. According to the Room Temperature Policy, “University guidelines for space temperatures are 70 to 76 degrees during occupied hours and 63 to 83 degrees during unoccupied hours” (“Building”). Coover is in need of an upgraded system to control the steam and keep the temperature in the rooms stable in fluctuating outside temperatures. This will lead to more comfortable rooms to for occupants and use steam more efficiently. There are two main problems associated with the current steam heat system which we are faced with: the steam radiator valves are often hidden behind miscellaneous furniture or equipment making it very hard to adjust the temperature in the rooms. Additionally, there is no reliable way to control the temperature of the room. Currently, the only way to adjust the temperature is to open or close the radiator valve manually and wait to see if the room gets too hot or cold.

1.3 Operational Environment

Our product will be used in classrooms around Coover. Extreme temperatures will not be an issue for our product, nor will adverse weather conditions. However, dust may accumulate on our components and unintentional abuse may occur due to human interaction. Some preliminary testing has shown that the valve mount will need to withstand temperatures less than or equal to 150 degrees Fahrenheit. Additionally, the mount and motor should be made of corrosion resistance material because some moisture is expected on or near the radiator valves because of small amounts of steam leakage.

1.4 Intended Users and Uses

Our product will be used to control the temperature in classrooms and offices. People in those rooms as well as administrators to the system will be able to control the temperatures.

1.5 Assumptions and Limitations

Assumptions:
- The maximum number of people using the thermostat is one at a time
- The user will not move the valve by hand
- The user will have permission to change the temperature if they have access to the room

Limitations

- The valve controller will fit on the valve and have access to a wall outlet
- The thermostat will be battery controlled
- The user will be able to change the temperature on the thermostat in the room
- An administrator of the system can access and change the temperature for any room from anywhere with access to campus wifi

1.6 Expected End Product and Deliverables

First deadline: end of April 2018
- ability to turn the motor
- basic wifi communication between the valve controller and thermostat
- information on the temperature patterns in the rooms (for a baseline)
- information on battery usage of the thermostat

Final deadline: end of December 2018
- Thermostat and Valve controller prototypes with user interface and
- Instructions on how to create the project from its components in order to duplicate for any room in Coover Hall
- Parts list of materials needed
- Working prototype and data to show improvement in temperature consistency in rooms where it’s implemented
- Sends error reports if the valve is stuck or communication is lost

2. Specification and Analysis

2.1 Proposed Design

The design for this project is constrained by our client's requirements. Specifically, these requirements define two systems: a thermostat and valve controller. The thermostat will be able to control the temperature of the room. While our group is only prototyping with a single valve controller and thermostat, the client would like to extend the solution to all rooms containing steam valves in Coover. This means that we have to design for numerous systems, while only prototyping one. We are developing our solution with these considerations in mind.

The valve controller uses a motor from the last group's attempt of the project. We are designing a mount for the motor using a CAD program. The design will be versatile and will fit on any of the steam valves in Coover. The motor will then be operated by a microcontroller through a motor drive board. Because the microcontroller will communicate with the server with WiFi and power is not a concern for this part, we have decided to use a Raspberry Pi. The valve controller can connect and report to this server the information about temperature and status. This will also enable a web-user to change the temperature remotely. Finally, we are planning to develop a feedback system with the valve controller that will keep the temperature consistent. We have designed the motor mount in our CAD program based on our observations and data of the valve controller and the specifications of the motor. We are gaining experience with SSH (secure shell) with the raspberry pi that is logging data from the temperature system. Using SSH, we can establish SSH keys, experiment with crontab (software used to time execution of programs) and pm2 (a program manager that can control continuously running programs). All of this experience will carry over to design of the software for the valve controller.

The thermostat has to be mounted on the wall and operate with power from batteries for a semester, so we have opted to use a low-energy display and a Feather microcontroller board with a WiFi chip built in. This board can be put into a deep sleep mode that will increase the battery life considerably. As we order and assemble these parts of the thermostat, we will measure and make corrections to the software to reduce the energy consumption as needed. We have connected a Raspberry Pi to an I2C temperature sensor and created a program that logs the indoor temperature from the sensor, the outdoor temperature from an independent data source, and the date and time of the query and stores it into a file on the Pi.

In order to facilitate building-side communication, we have a dedicated Raspberry Pi that will be set up as a server. We have been experimenting with ssh to control the pi and designing what the network should look like.

2.2 Design Analysis

Our team has been divided into software and hardware teams. The software team is working on communication with the server, the pis, and the feathers for their respective rooms. The hardware team is developing the valve controller.

Software: We have set up a raspberry pi in 1316, the room our client has designated for us to use as a test space, recording temperatures every 15 minutes. We started out with thinking of using an analog temperature sensor and then sending the value over UART to the microcontroller. After prototyping the initial design, we found it to be too complicated to use a pi and microcontroller. We found a sensor built with a I2C bus capable of communicating with the pi. We also have our design for the thermostat ready to prototype. We initially thought of using a 16x2 character LCD. Interfacing with the display would require hardware that another display would have built in. Using an LED with built in I2C would simplify our designs, use fewer pins on our feather (there are only 9 available GPIO pins), and enlarge our display. We decided to use alpha numeric character displays. After considering our alternatives, the alpha-numeric display seemed our best option. We are using I2C to retrieve temperature data from our digital sensor, so communicating with the I2C display would not use any additional pins. While the LEDs do use more power, we believe that the display will sleep a majority of the time, so the display will not be our main power concern. The buttons are momentary switches. They are easy to interact with, use little power, and are inexpensive. We are testing each component individually with the board that it will communicate with in our design.

Hardware: We have been working on the physical mount that will hold the motor that turns the valve. We are using the CAD software OnShape to model several ideals we have discussed at our meetings. We took measurements of the steam valves in Coover to use as a reference in our design. While measuring the valves, we made an important observation about the valve. Some of them are next to right angle turns in the pipes. Our original design encases the entire valve in a metal mount, holding onto the pipes. This could be problematic if the pipe has a curve, as the box will not fit. We switched to a mount that supported itself by holding onto the valve. This actually made the mount easier to design. Our design now includes a ring that could be tightened around the valve to provide adequate support. We drew up this design in our CAD program and presented it to our client. The assembly drawing can be found in Figure 1 in the appendix. After discussing with our client, we agreed that a four sided steel case was overkill and only one side could be mounted to the motor, which we believed was inadequate. After making changes to improve our original design, we have passed off our drawing to Leland Harker to have them manufactured. A view of our new design is in Figure 2 in the appendix. Once our prototyped design comes back, we can assemble and test functionality of our design. Currently, we are looking at power and control of the motor. We plan to test our motor, mounting bracket, and programming interface as soon as we have all the parts.

3. Testing and Implementation

3.1 Interface Specifications

We realize that testing will be a major part of the project. So we started with logging temperatures inside room 1316 and kept track of the temperatures outside as a comparison. Data collection occurs automatically every 15 minutes. When we want to look at the data, we graph the acquired data. We will use data from the last few months as reference point and compare temperature patterns before and after we implement our system. For the thermostat, external buttons will change the desired temperature. and because that would be displayed on the display we would know if those buttons are working based on whether or not the temperature value is changing. To make sure the valve and thermostat pi's are communicating with each other we plan on having LED's on either pi's that go off on sending and receiving data. We do not have the valve control equation yet, but we also will calculate a relation between the temperature changes based on how much the valve opens and closes. For that we will start with turning the valve and seeing what changes we observe in temperature and then coming up with a function which will probably not be linear and will be loaded on the valve pi, this would take the temperature difference as input and turn the valve according. For remote access to adjust the temperature, we plan on creating a webpage that can only be accessed by the staff within Coover. Logging in and authentication of users will be handled by the ISU Shibboleth system, and the webpage itself will start as two buttons similar to the physical thermostat, and work in the same way. The testing for this process will also be similar to the thermostat, with LED's to determine if the commands are being sent.

3.2 Hardware and Software

Our project is using several tools and platforms to validate our ideas as we produce our prototype. The testing of our project is taking place throughout the design and build, so the tools we are using to create the project are also being used to test. These tools include Python, oscilloscopes, OnShape, calipers and torque wrenches.

Our software team is using Python to program the Raspberry Pi for the thermostat and valve controller. The Raspberry Pi will have several physical inputs and outputs that are handled by our code, so we will be simulating these inputs and evaluating the output products as needed. The main tool we are using to evaluate outputs and debug is an oscilloscope. We have used two different usb oscilloscopes to monitor our Pi's signals, including an N-scope and a Picoscope. Both tools work similarly and enable us to monitor the outputs of the pins on the Raspberry Pi.

Our hardware team is using an online CAD software called OnShape. OnShape is preferred over other CAD tools for our team because all our files are stored online so the team can work on the same model from their personal computers and not have to worry about how to share the drawings with each other. We are using OnShape to model our motor mount design to test its dimensions before fabrication. We also modeled one of the steam valves to help with modeling our mount. We used a caliper to gather the valve dimensions because the valve manufacture, Ohio Brass Co., is no longer in business and therefore do not have their own CAD drawings online for us to use.

One test we conducted when we choose a motor was the torque required to turn the valve. Using a torque wrench, we found it takes around 1 ft-lb to turn the valve and around 8 ft-lb to close the valve. We had some brushed DC motors from a previous project that we are going to use. We took those dimensions and modeled it in OnShape with our mount design and valve to model the system as a whole.

The next step for the hardware team is creating circuits that will run the motor and communicate with the thermostat. This circuit, and all upcoming circuits, will follow the same design and testing procedure. First, the circuit will be conceptually designed using our math and circuit analysis technical we learned in our previous circuits classes. They will then be breadboarded and analyzed using our oscilloscope and multimeter. At the same time as the prototyping, the circuit will also be modeled in Modelsim, a SPICE program that will easily allow us to transfer the circuit to the program Ultiboard, program that allows us to create our PCB layout and generally the Gerber files to be fabricated.

3.3 Functional Testing

As each part of the project is worked on, we will test expected functionality within the system. Once we have verified its functionality, we will build up our system incrementally, testing as we add components and functionality. This will allow us to debug the system in smaller parts, and catch edge-cases where our system may not respond according to our design specifications.

The valve controller will need to be placed on a variety of steam valves to determine if it can operate with various amounts of torques. We will also include evaluations of our software as it responds to different environments The thermostat will need to be tested extensively as it is a user interface; this will include measuring the microcontroller’s response to the button inputs and the screen’s display. Each valve controller and thermostat are going to have wireless connections to each other and to the web server, so the communication protocols will also need to be verified.

3.4 Non-Functional Testing

This testing will be completed as the project is completed; we will establish the criteria to validate these test results at a later date.

3.5 Process

Hardware: We currently do not have anything to report for test results at this time from a hardware side. We are currently are waiting on our mount to be manufactured so we can begin testing it in a live environment.

Software: We have tested our temperature sensor by logging the temperature in 1316. We can then compare it to the outside temperature to establish a baseline for the current temperature patterns in the room. See figure 3 below for a graph of the data we have obtained so far.

3.6 Results

While we are still developing the system, our tests for the temperature sensor are promising, offering consistent temperature readings over our 2 week test (see figure 3). We are also testing a pi, verifying that the operation is consistent over long periods of time. We have concluded that with updates and reboots every day, our pi is operating with nominal response times. As of the writing of this paper, results for the initial phases of this project look promising. We will continue to test and modify our project as we develop the system and fix problems.

4. Closing Material

4.1 Conclusion

Software: Our main goal for the software side of the project is to set up and test the communication between the thermostat, motor controller, and server such that we can adjust the temperature on the thermostat via a button interface and it will maintain the selected temperature in the room. Our secondary goal is to set up and test a simple webpage to change the desired temperature of any room from any connection to the campus WiFi from an authorized user. Our methods of testing this communication will be to create a command by pressing a button on the thermostat, and then send that command through the server to the motor controller which will light an LED to confirm that it has received the command. Once we have basic communication set up, we plan to implement the motor control based on the command sent by the thermostat. To test this, we will again send a command to the controller and measure the change in position of the valve to see if it moves to the correct position. For the webpage portion, we plan to implement ISUs Shibboleth authentication system to limit access to the staff inside Coover. We also plan to design the remote access system such that an administrator has access to and can change temperature in any room, individually or simultaneously.

Hardware: So far, we have gone through several iterations of the physical mount that holds the motor that turns the valve. We have passed these drawing to Leland Harker to have them manufactured for use so we can begin assembling and testing these mounts in a physical environment. Motor control is in development. We are discussing different H-bridges found online for controlling the motor, as well as building our own. We would like to get this done as soon as possible since our current time chart has us finalizing this part by March 3rd. We will not meet this deadline. This delay occurred because we did not originally account for our mount deign to go through several iterations as it did. Our current goal is to get our motor up and running as soon as possible to try to get us back on schedule. We also may have someone from our software team help the hardware team to get them caught up.

4.2 References

"Building Energy Plans: Coover Hall." *Iowa State University Utility Services*. https://www.fpm.iastate.edu/utilities/bldg_energy_plans/building.asp?bldg=COOVER Accessed 6 February 2018.

"Coover Hall." *Iowa State University: College of Engineering.* https://www.engineering.iastate.edu/the-college/engineering-buildings/coover-hall/ Accessed 6 February 2018.

4.3 Appendices

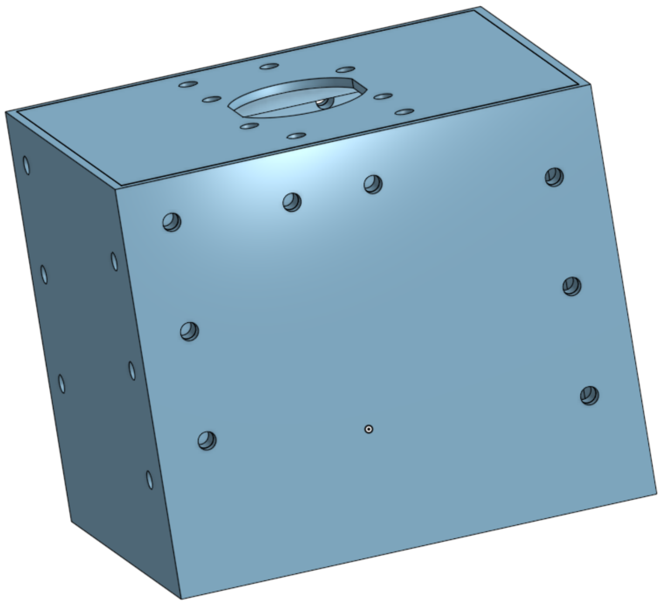

Figure 1. Scrapped motor mount assembly drawing

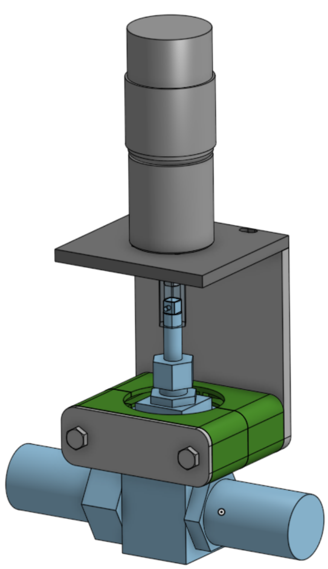

Figure 2. Current motor mount assembly drawing

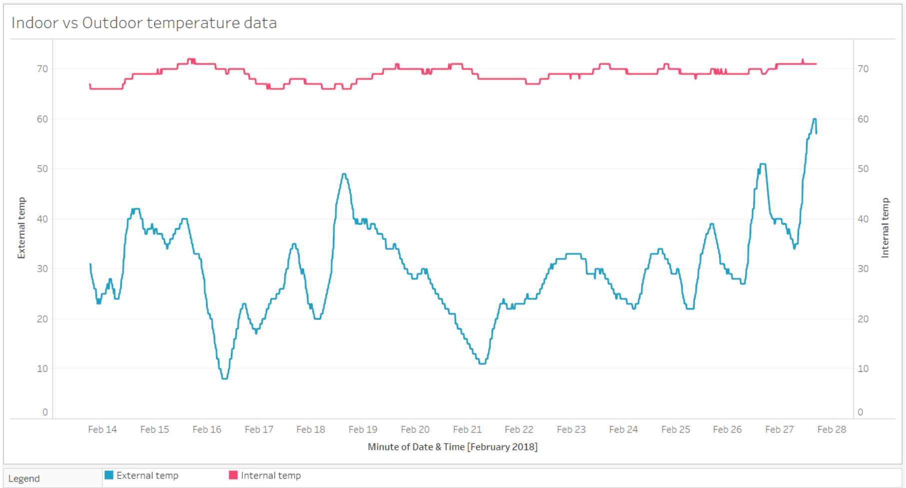

Figure 3. This is the data we logged in the first few months of this project in room 1316. The temperature is quite stable for the most part, and only drops after a drastic change in outside temperature. This does include changes in the valves control