Steam Heat Controller Retrofit

Project Plan - Final

April 2018

Team # 2

Advisor/Client: Lee Harker

Team Members:

Jevay Aggarwal - Technical Lead

Sarah Coffey - Weekly Report Lead

Thomas Devens - Project Plan Lead

Joseph Filbert - Client Contact Lead

Ken Wendt - Webmaster

Liz Wickham Kolstad - Design Document Lead

Team Email: sddec18-02@iastate.edu

Team Website: http://sddec18-02.sd.ece.iastate.edu/

Contents

1. Introductory Material

1.1. Acknowledgement

Our client and faculty advisor, Leland Harker, will be our point of contact for any equipment we need as well as any fabrication of specific parts. This is especially convenient as he runs ETG who normally orders parts for ECpE students that are working on various projects. Additionally, he will provide technical advice on various parts of the project, and guidance on the project as a whole.

1.2. Problem Statement and Proposed Design

As with many buildings on ISU's campus, Coover relies on steam to heat the building. The issue at hand is that the valves that control the steam 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 exact degree of temperature in the room. This means that the only way to adjust the temperature is to open/close the valve manually and wait to see if the temperature changes desirably.

Our solution is to retrofit the steam valves with a networked motor controller that is connected to a thermostat in the room (Diagram 1). Because each device will be networked, we are also going to create a web interface. This will allow users to select a desired room temperature with the web interface or the thermostat, which will in turn set the valve to a new position based on the requested temperature. The valve controller has a motor that opens and closes the valve and eliminates the need for manual interaction. This will be controlled by a Raspberry Pi that will communicate back and forth with the server. The thermostat will be mounted on a wall in the same room as the valve, and wirelessly communicate with the controller Pi. It will have a display to show the current temperature and error messages, as well as buttons to change the desired temperature.

Based on the requirements and constraints (1.1-1.6) we have been provided/come up with, we believe that our solution is the simplest and most cost effective option. Any additional components or interfaces would overcomplicate the method of adjusting the temperature, create more error potential, and/or push the creation/operation cost to an infeasible amount.

1.3. Operating Environment

Our product will be used in a standard classroom or lab. The thermostat will only be exposed to standard room temperatures i.e. 60-80 °F. There may also be a bit of dust in the air from being in the labs with drills and saws. Therefore, the end product will have a protective casing covering the circuitry. The motor and circuits controlling the motor will be above the valve on the steam pipe, so the design for the mount will be designed to protect the inside from the heat coming from the pipe.

1.4. Intended users and intended use

Our intended end users for our steam valve retrofit project are any students or faculty in Coover. The thermostat will be placed in an easily accessible location so whoever is using the room or lab should be able to change the temperature. This is our base requirement that needs to be met. Ideally, we would like to create a network interface that ETG could use to monitor and manipulate each individual thermostat in Coover. This could be used to automatically decrease the temperature during breaks where fewer people will be using the building to save money.

The design plan is to solely rely on the temperature sensor located on the valve controller Pi for collection of data. This being the actual temperature in the room and the desired temperature given by the user. Based on the difference between the current temperature and the desired temperature it will determine which way and how much to change the position of the valve. It will also monitor and detect any issues, sending error messages to be displayed on the thermostat, as well as sending an email to any staff that have remote access.

1.5. Assumptions and limitations

Assumptions

- The product will only be used in Coover.
- It will be installed in one room with the capability to replicated in an unlimited amount of rooms.
- There will be a single thermostat per valve.
- The temperature in Coover will be within 60 °F to 80 °F.
- There is a wall outlet near the steam valve to power the motor control circuit.

Limitations

- Our product has a $500 budget for building and testing our whole project.
- The thermostat must be battery operated and be able to last for at least one semester.
- The valve controller must be fitted to the valve with little to no change to existing setup.
- The remote access must be secure i.e. only accessible to the staff of Coover

1.6. Expected end product and other deliverables

The thermostat will be delivered at the end of the second semester. It will handle the local input of temperature adjustments, as display any error messages or temperature updates received from the controller.

The motor control circuit will be delivered at the end of the second semester. It will take the variables of current temperature and desired temperature. It will then do computation to determine which way and how much to turn the steam valve using a motor connected to a gearbox to increase its torque to the required amount to turn the valve. It will also detect if the valve is stuck and various other possible errors and report them back to the thermostat to display to the user.

The webpage interface will be delivered at the end of the second semester. The webpage will be able to monitor the thermostat temperature for all thermostats in the building. It will display the errors back to the user as well as have options to control temperatures remotely.

2. Proposed approach and statement of work

2.1. Functional Requirements

- Maintain room temperature within ± 1 degree from setpoint
- Thermostat must be powered by a battery that lasts 1 full semester
- Thermostat must communicate with valve controller wirelessly
- Valve controller must detect stuck or broken valve and report error
- System can receive administrative commands to control one or all valves remotely
- A method to send administrative commands and monitor status of valve controllers

2.2. Constraints considerations

Our approach to this project will be influenced by several constraints, including the client specified set-up, budget, testing period, and time. Our client, Lee Harker, has given us the directive to implement our system with a single thermostat per valve; further, he would like the thermostat to be battery operated, and for the battery charge to last an entire semester. The steam valves in Coover are responsible for heating the rooms they are in, as well as the adjoining rooms, so changing the valve position will have a varied temperature effect on the rooms. Since we are only able to measure the temperature in one room (the one with the steam valve), this means we have to collect data and try to account for the temperature variance in the feedback system. The majority of the prototyping and testing phases of the project will be done when heat is not necessary, so it is vital that we collect as much data as possible.

The budget will need to be managed well; we are only able to spend $500, so we anticipate needing to make decisions for part sourcing and creation based on this factor. The final constraint we have identified is the length of this project. We must complete the project within the 2 semesters of Senior Design, while we take other classes. It is necessary then that we manage our time and set hard deadlines and goals in order to complete the project.

2.3. Technology considerations

Keeping in mind our constraints, we have identified what kinds of technologies are in and out of scope for this project. The valve controller is going to be networked, so we need to use a microcontroller that has networking capabilities and correct pinouts for driving a motor. Since this system will be enclosed in a box that uses power from the wall (no energy consumption limitations), we have decided to use a Raspberry Pi; it allows for both WiFi and Bluetooth communication, is easy to program, and supports PWM output for a motor. There are also a variety of motor drivers that can assist in the connection from the Pi to the motor.

For the thermostat, low energy consumption is our number one priority and design constraint. We have decided to use an Arduino-based microcontroller called a Feather board that supports a deep sleep mode, which will essentially cut the microcontrollers power draw to zero when not in use; to interact with the microcontroller, we can program interrupts that will take it out of this deep sleep mode. The board also supports the I2C standard, enabling us to use a digital thermostat and a variety of screen types with ease. The Feather board comes in two networking flavors: WiFi and Bluetooth. Depending on the type of display we can utilize either a WiFi or Bluetooth connection to the valve controller. This decision will be made through testing procedures on the chips.

Our project will have to control steam valves that could stick or break. We will have to include checks in the software while the motor turns the valve. If the valve is not working properly, we can send error reports to someone who can fix it. For simplicity, we will have to make our user interface as simple and straightforward as possible. This includes system setup when a new valve controller is fitted to a room, or when an existing system is changed. Additionally, our client informed us that we would not be able to fit a different valve into the system, creating the need for it to be retrofit onto the existing setup.

2.4. Testing requirements considerations

Several of our design and technology options are going to be determined through intermediate testing procedures. Decisions will rely on such conundrums as the power consumption of the thermostat, protocols on the Pi, and ease of programming and control of the motor from the microcontroller. Each of these situations will involve setting up hardware and software environments, as well as measuring and observing the effects on the systems being tested. For example, the communication between the thermostat and valve controller could be done via Wifi or Bluetooth. Since we want to consume minimal power, we tested the consumption rates between the two, and found that Bluetooth uses less energy. This will change the method of creating/sending/reading the data sent between the two devices.

The valve controller and thermostat are expected to maintain a set temperature without the need for maintenance or manual adjustment. With this in mind, we will need to benchmark each subsystem for reliability and usability. Additionally, our project is composed of several subsystems, so we will need to evaluate how the components interact with each other. In terms of reliability, the system must operate without user intervention indefinitely; in preparation for inevitable failures, the device will be programmed to report its status and alert the appropriate overseer in the case of errors. Every possible condition and error will need to be tested thoroughly. For usability, we will rely on our client’s feedback of the interfaces as we create them to gauge their efficacy. Finally, extensive testing must be completed for each subsystem at every stage of integration into the project in order to guarantee a unified final product. We must also confirm that any relevant IEEE or other industry standards are adhered to, as well as not violating any rules or laws that may apply to our project.

2.5. Security considerations

We will be working on the wireless network which is open to all students. In creating a remote interface, we will have to consider the networking protocols we use, and make sure that people that should not have access, do not have access. To combat this, we plan to integrate Shibboleth, ISU’s existing authentication system to limit access to the staff of Coover.

2.6. Safety considerations

We will be abiding by all safety precautions throughout this project as we will be working with steam valves, electric drive motors, circuitry, and heavy equipment. This means that the motor/controller will be constructed such that they will not endanger any person or equipment. Our group has also been trained on safety practices for the room we will be testing the system in, as it contains many machines that could be easily damaged.

2.7. Previous work/literature review

Danfoss has several temperature sensors that work for many different types of heating including radiators. They also offer a variety of radiator valves as well (Danfoss). However, Coover already has radiator valves in place and these will not be changed. Additionally, we want to connect our thermostats to a network and have a webpage that can monitor and control all of them.

There has been a prior attempt to solve this project from another senior design group in 2011. After an examination of their materials, we were able to identify that groups reasoning, approach, and methods, and have decided what parts of their project would be useful for ours going forward. The previous group attempted to create a circuit from scratch to support their microcontroller, display, and communication devices; we decided that their work on this circuit would not be useful to us because it did not have sufficient documentation. What we could learn from it however, is that we did not want to build our own microcontroller from scratch. Instead, we determined that using premade boards like the Raspberry Pi for our implementation would be much simpler to program, document, and setup. The other significant part of their project that was of use to us was their motors. We found documentation online for both of them, and have been able to repurpose one of them for our project.

2.8. Possible risks and risk management

The risks we have identified so far include falling behind schedule, running out of budget, equipment failures, and software errors. By outlining clear goals and deadlines at the beginning of our project, we are actively engaged in staying on schedule. We have also broken our project up into several goals, such that we are able to complete basic functionality before working on extra features. This gives us some extra time to complete the project in case we do fall behind.

We are mindful of our budget and have a member of our group in charge of managing it and our resource requests. By setting up a process of requesting and delivering through a single group member, we will be able to stay on budget and keep unnecessary expenses at a minimum. Another risk is that our equipment or software fails, either in development or once its deployed. We will mitigate these risks by researching the datasheets for each component before working with them, and accounting for these constraints in testing. For software, we will ensure exceptions and error reporting are apart of the deliverables of the project. As we encounter risks through the development of the project, we will document them as part of our testing procedures.

2.9. Project proposed milestones and evaluation criteria/testing plan

- Design and construct a motor mount to hold the motor to turn the valve
  - The secureness of the mount will be evaluated to make sure the motor is does not move around
  - The size of the mount will be tested to make sure it fits in the area around the valve
- Control the motor using the H-bridge controlled by the Raspberry Pi
  - Test that the motor turns the valve in the desired directions and angle based upon software calculations and data acquisition
- Create the thermostat
  - Make sure the thermostat can communicate with the Pi for the motor controller from across the room (Find maximum communication distance)
  - Make sure the battery will last one semester, even after inquiry
  - Make sure the Pi can communicate with our thermostat microcontroller via bluetooth
  - Make sure the thermostat microcontroller wakes on button press
- Program the logic of the Raspberry Pi
  - Make sure the Pi can compare the desired temperature and the accurate temperature and adjust the motor accordingly
  - Make sure the Pi can get accurate information from the temperature sensor and the reading is accurate even when positioned on the mount
  - Make sure the Pi is getting the correct amount of power to avoid damage
- Test the Server Interaction
  - Input different desired temperatures through the server's interface and test that the valve is moved according to the difference between the desired temperature and the current temperature
  - Test that the interface cannot set an invalid desired temperature
  - Test that only authorized personnel can change the temperature in their respective rooms

2.10. Project tracking procedures

In order to complete our project on time, we must be aware of the deadlines and goals before we miss them. We have established detailed goals for each section of development as a group; each member is cognizant of their responsibilities for reaching the goals by the deadline. We have set meeting dates at least twice a week that are documented by a member of the group; this information, as well as each member's update information, will be included in the weekly status report, and provide us with a benchmark for comparison with the original project plan. These procedures will enable us to stay up to date with our deadlines.

2.11. Expected results and validation

We expect to complete the project within the timeline set out in our plan. This includes a working valve controller and thermostat system. We will consider our project a success when the following functionalities are complete:

- Pressing the temperature control buttons on the thermostat change the temperature on the display. This change will also be seen on the web interface.
- The temperature change is relayed from the thermostat to the valve controller successfully.
- The valve controller changes the valve position accordingly.
- The temperature is adjusted in the room to maintain set temperature through the feedback system.

We will test each of these sections as we develop them according to our testing considerations. As they pass or fail, we will adapt the project and validate results.

3. Estimated Resources and project timeline

3.1. Project Timeline

The timeline for our project is layed out in the form of a Gantt chart (Diagram 2); we have scheduled each section of the project by week for both semesters and set goals and due dates by subsection. Since our team is broken into hardware and software teams, we have several concurrent goals that each group is responsible for. The Spring semester begins the last week of January and spans until the week of finals. Since our group was not able to meet with our advisor and client until the end of January, that gave us one less week than the Fall. The Fall semester begins the first week of classes and spans until the first week of December. We did not budget the second week of December into this schedule in preparation for dedicating that time to finish documentation for the class.

Due to our group's prior knowledge and experience (3.2), we decided that almost all of the subsections dealing with development could be accomplished within a week or two. Each section also has two weeks dedicated to testing and documentation; this is to ensure that there is extra time for the completion of the section, and to allow us the time to sufficiently document the features before moving forward. The final weeks are dedicated to final integration testing and completion of stretch goals.

3.2. Feasibility Assessment

Our previous experiences and knowledge will be of great use in completing this project. Sarah and Jevay have both been TAs for an introductory embedded systems class; this has given them an advantage at reading datasheets, selecting microcontrollers, and understanding how the different parts of the project will be integrated. Liz has extensive experience with Raspberry Pi projects that include embedded systems implementations. Her knowledge has been invaluable as we select parts and begin prototyping. Thomas has the most experience with software development and creating user interfaces in the group, which is going to be useful for developing the web interface. Ken has previously designed and printed integrated circuit boards; since we are expecting to have several boards in our project, we will be relying on his knowledge to this have them built correctly. Joe has a background in working with mechanical systems which has enabled him to create and work with the motor mount in an intelligent fashion. We have laid out the various challenges we might encounter throughout this project, and have come up with plans to overcome them in a time and cost effective manner. At this time, we believe this project to be feasible in terms of the length of the time given, budget, and requirements laid out in Section 2.

3.3. Personnel Effort Requirements

Due to the diversity of majors in our group, and since the project consists of a major hardware and software portion, we can divide our group into two groups. The hardware group which consists of Joe and Ken will be tasked with creating the assembly that will fit onto the valve as well as the circuitry contained within. The software group which consists of Jevay, Liz, Sarah, and Thomas will be handling the communication between the two devices, programming for the thermostat, and the remote access.

3.4. Other Resource Requirements

For the valve controller, we will need to acquire the following materials:

- Motor
- Motor Gearbox
- Power Supply
- Raspberry Pi

The thermostat will require the following materials:

- Raspberry Pi with display
- Buttons/Switches

Both components will also require a custom casing to protect them from any tampering as well as the high heat conditions next to the steam pipe. In addition to the materials listed, we will require access to the room that the product will be created/installed in.

3.5. Financial Requirements

As previously stated in Section 1.5, our budget for the entirety of this project will be $500. This amount cannot be exceeded, as it would make the cost for the potential expansion of the project not feasible. We have designated one member of our group to maintain this budget and control all purchases necessary for the completion of the project.

4. Closure materials

4.1. Conclusion

The steam heating system in Coover does not have a reliable method to adjust the temperature in each room. Currently, blocks of 2-3 rooms are controlled by a valve in one of the rooms, and the only adjustments that can be made are done by manually turning it and waiting to see if the temperature is adjusted accordingly. In addition, the valves in many rooms are located in inconvenient or hard to reach places. To solve this, our plan is to implement a system that consists of a thermostat that sends data to a Raspberry Pi that determines the course of action to adjust the temperature by using a motor to control the steam valve. In addition, we plan to implement a server interface such that the staff in Coover may adjust their respective rooms remotely. This will make adjusting the temperature easier and more accurate, as well as allowing a staff member to control the heat of all rooms over breaks.

Given each of our group's individual skills, and the plan that we have created, we are confident that this project can be completed by the end of the Fall semester. We are looking forward to using what we have learned for something that will help improve the well-being of everyone that works/learns in Coover.

4.2. References

Danfoss. Danfoss http://products.danfoss.com/home/#/. Accessed 24 March 2018

4.3. Appendices

Diagram 1 - System Overview

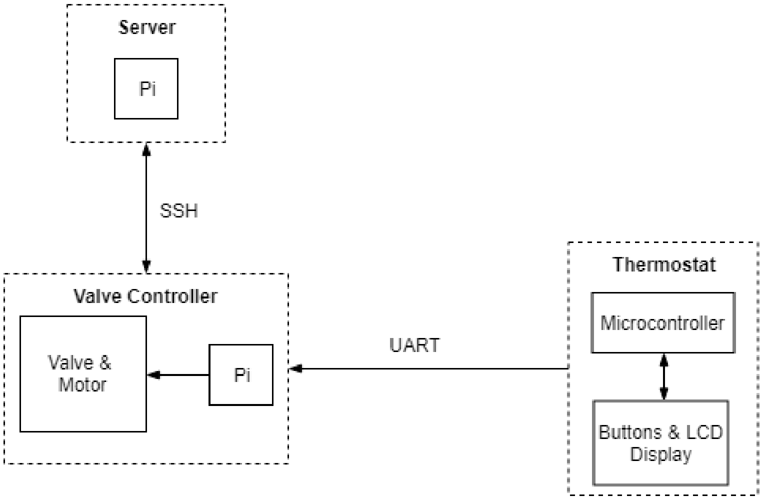

Diagram 2 - Project Timeline

	Semester 1													Semester 2													
Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Deliverable date	3-Feb	10-Feb	17-Feb	24-Feb	3-Mar	10-Mar	24-Mar	31-Mar	7-Apr	14-Apr	21-Apr	28-Apr	5-May	25-Aug	1-Sep	8-Sep	15-Sep	22-Sep	29-Sep	6-Oct	13-Oct	20-Oct	27-Oct	3-Nov	10-Nov	17-Nov	1-Dec
Research	■	■																									
Data Collection			■	■	■	■	■	■	■	■																	
Thermostat																											
- Temperature sensor - Microcontroller communication			■																								
- Logging temperature				■																							
- Fetching log					■																						
- Battery and Case						■																					
- Testing and documentation							■	■																			
Valve Controller																											
- Mount			■																								
- Transmission				■																							
- Motor controller					■																						
- Microcontroller						■																					
- Logging torque in software							■																				
- Error reporting								■																			
- Case									■																		
- Testing and documentation										■	■																
Thermostat - Valve controller communication												■	■														
- Testing and documentation														■	■												
User Interface																											
- Buttons and LCD								■																			
- Remote control									■	■																	
- Error communication											■																
- Status Reporting												■															
- Testing and documentation													■														
Feedback Control																											
- Analyze data														■	■												
- Code																■	■	■	■								
- Testing and documentation																				■	■	■					
Final testing and documentation																							■	■	■	■	■