Temperature Sensor Network for Thermostat Control

Team 33

ECE 445 Design Document
TA: Dongwei Shi
Haige Chen, Ryan Finley, Heming Wang

Objective

Traditional thermostats collect temperature from one location. This may be insufficient in a place such as a multi-room apartment where different rooms, or different corners of the same room do not get heated/cooled evenly. While some modern HVAC systems can check for these imbalances, it's not practical for older buildings to replace existing systems. Also, replacing the whole HVAC system could be very costly. Regardless, incidents such as forgetting to close a door or window may cause dramatic disparities in temperature - hiking heating/cooling bills if not warned early.

We seek to build a scalable temperature aggregation system as a cheaper add-on (than replacing with newest zoning HVAC) to older HVAC systems to collect and interpret temperature data across multiple rooms in any internal environment. The design would require temperature sensors, wifi chips, and MCUs integrated on PCBs, and a central hub that gathers all the sensor data. The user can monitor temperature across rooms and receive alerts through a phone app in real time. The system will also attempt to regulate the internal temperature through two methods: adjusting the thermostats temperature and opening/closing air-vents.

Background

There exists zoning products [1][2] that can convert traditional single zone HVAC into multizone. However, most of these products require complicated installations and modifications to the wiring in an apartment. Besides, the number of zones that can be divided is fixed to the zone controller installed, i.e. if the user originally installed a two zone system but wants to expand it to four zones, he/she has to purchase and reinstall a new controller. Our system has two key differences to address these problems. Firstly, all of the temperature sensors are powered by batteries and communicate wirelessly, which eliminates the need to lay new wires. The only component powered externally will be the central hub, but it can be plugged into any wall outlets. Secondly, because all the communications are carried via wifi, our system is easily scalable: we can slice/combine zones by removing/adding new sensors and set them up using the software for the central hub.

High-level requirements

- Room temperature sensor modules must collect and communicate temperature data to a central hub within the range of 10˚C to 30˚C.
- The central hub should coordinate hardware decisions based on live-analysis with the goal of maintaining a consistent home temperature.
- Analysis and alerts should be viewable through a web application hosted on the central hub.

Block Diagram

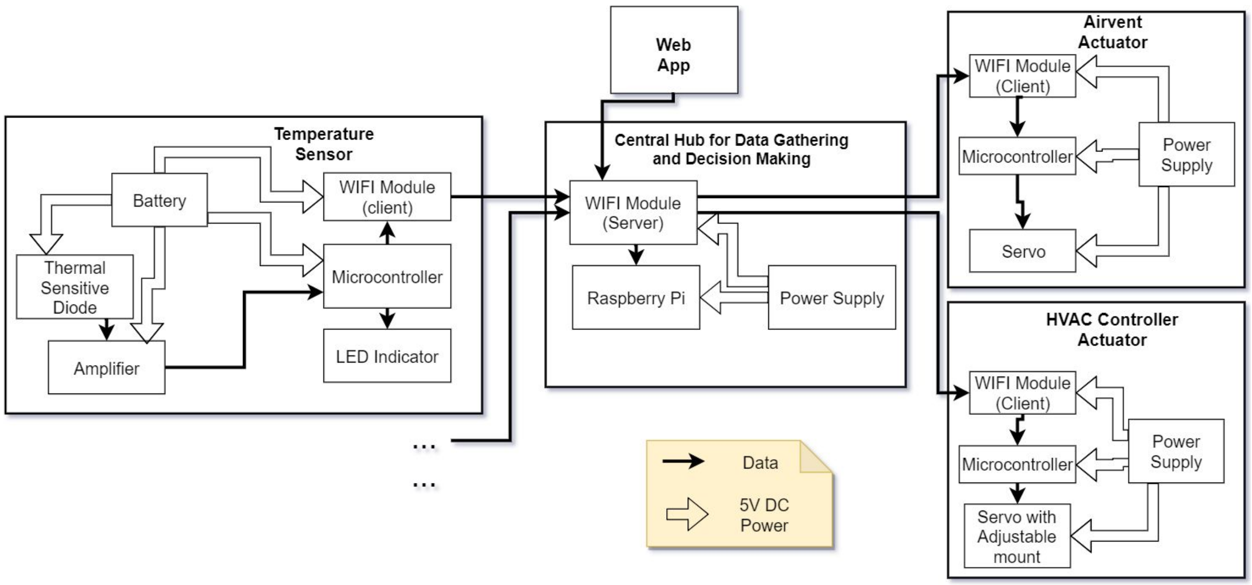

Figure 1: High Level Block Diagram

Physical Design

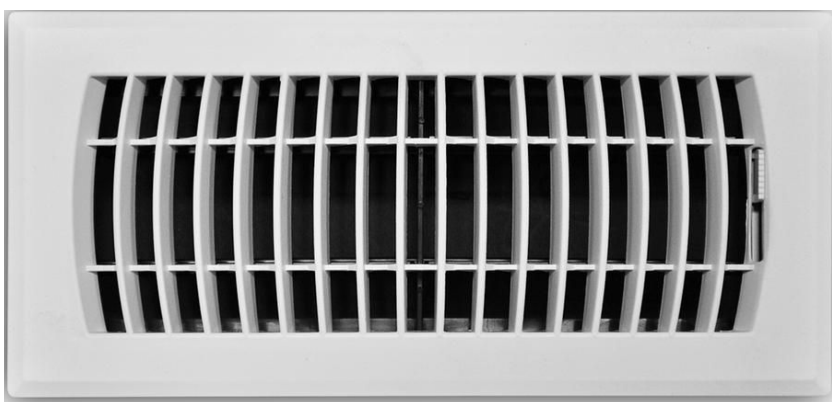

Figure 2: Sample Air Vent with Open/Close Slider

The automated floor-register air-vent will involve modifying an existing commercial floor register which includes a manual open/close slider, such as the one displayed above which contains a slider located at the far-edge of its vents. The modification will require us to attach and configure a stepper motor to drag the slider up and down, depending on the central hub's requested action.

Potential Reach Goal: One-fits-all clamp-on HVAC controller actuator case

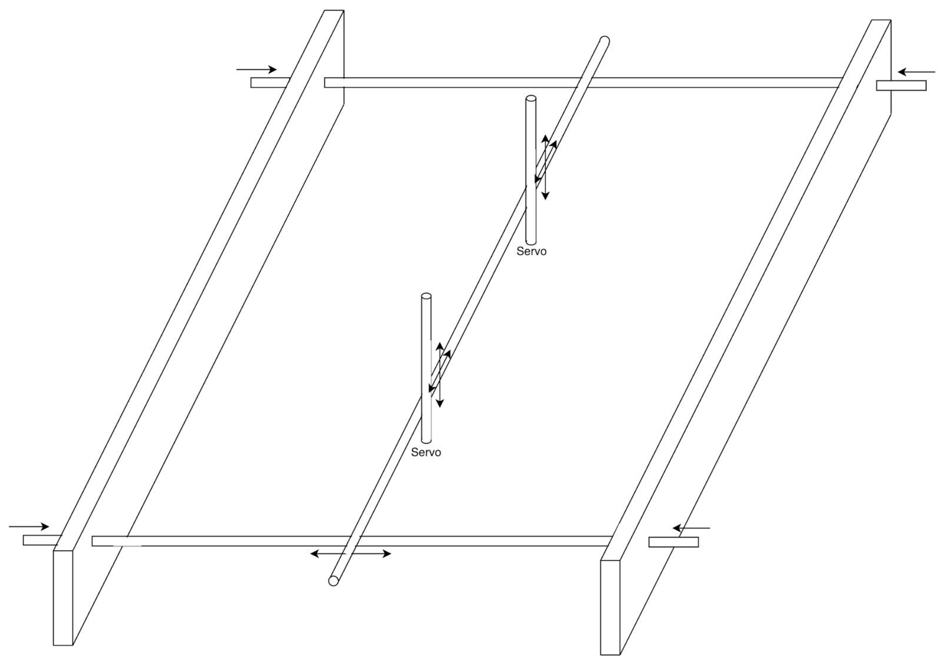

Figure 3: Idea for Mechanical Design for One-Fits-All HVAC controller actuator

We also plan to create a mechanical thermostat clamp out of several adjustable rods such that we can orient the up/down actuators into their correct positions. The thermostat clamp must be simple, but as adjustable as possible to accommodate a wide range of thermostats. Since the height of a thermostat is unknown, it's best to be conservatively large in the ridgid dimension. A simple design would consist of 2 narrow boards, each covered on on side with a strip of rubber to better grip the thermostat. These board will then be held on either side of the thermostat using threaded rods with wing-nuts. Two shorter rods will be fixed on the threaded rods between the two boards. These rods are adjustable up and down the rods, and hold a third adjustable rod which harnesses the actuators. These actuators can be adjusted up and down the rod into their necessary location. Components used to adjust and fix perpendicular rods together will require 3D printing, as commercial alternatives aren't easily acquired.

Functional Overview

Temperature Sensor Module

The temperature sensor modules measure the temperature in different rooms, then send data to the central hub. It will be implemented on a PCB that contains:

1. A pair of AA batteries connected to the component with a battery holder
2. An LED indicator light (HLMP-3507) to signify when a temperature reading is being sent out

3. A microprocessor (Atmega328) to coordinate logic between components
4. A low power wifi module [3] (ESP8266 in client mode) which is used to communicate with the Central Hub (power rating shown in Table 1).
5. A thermal sensitive diode (DS18b20) for temperature reading.

Parameter	Typical	Unit
Tx 802.11b, CCK 11Mbps, P_{OUT}=+17dBm	170	mA
Tx 802.11g, OFDM 54Mbps, P_{OUT} =+15dBm	140	mA
Tx 802.11n, MCS7, P_{OUT} =+13dBm	120	mA
Rx 802.11b, 1024 bytes packet length , -80dBm	50	mA
Rx 802.11g, 1024 bytes packet length, -70dBm	56	mA
Rx 802.11n, 1024 bytes packet length, -65dBm	56	mA
Modem-Sleep	15	mA
Light-Sleep	0.5	mA
Power save mode DTIM 1	1.2	mA
Power save mode DTIM 3	0.9	mA
Deep-Sleep	10	uA
Power OFF	0.5	uA

Table 1: Power Consumption of ESP8266 Wifi Module[9]

Central Hub for Data Gathering and Decision Making

The central hub gathers temperature data sent from sensor modules distributed in different rooms, process the data, present the data to users and send commands to actuators to manipulate temperature. It contains the following notable hardware components:

1. A wifi module (ESP8266 in server mode) that communicates with the sensors and actuators
2. A Raspberry Pi that processes the data and hosts a website so that users can check temperature on their phones.

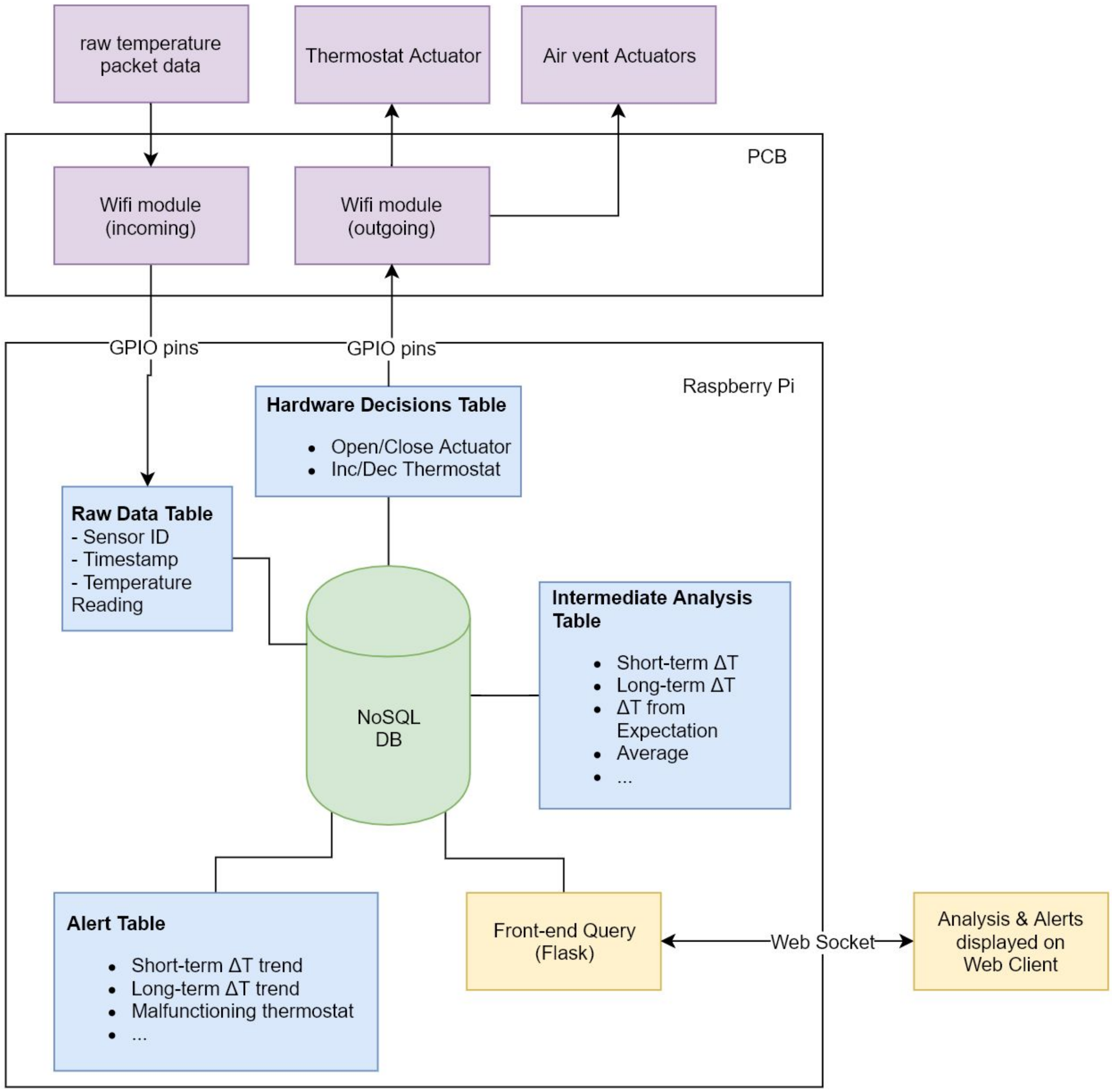

Figure 4: Raspberry Pi software block diagram and relevant peripherals

The raspberry pi's software usage can be split into the following dockerized components:

1. A NoSQL data storage instance (MongoDB) to store and retrieve temperature data from wifi module
2. A web-server instance (Flask) to analyze temperature data, display alerts, and display historical temperature data across rooms

The Database will consist of four tables: raw data, intermediate analysis, alerts, and hardware decision (as illustrated in Figure 4). The purpose for this separation is to separate Read/Write permissions into a more logical fashion.

Regarding external communication, the web client will request for the latest data upon an HTTP request whereas the hardware decision logic will kick off whenever new raw data is received in the DB.

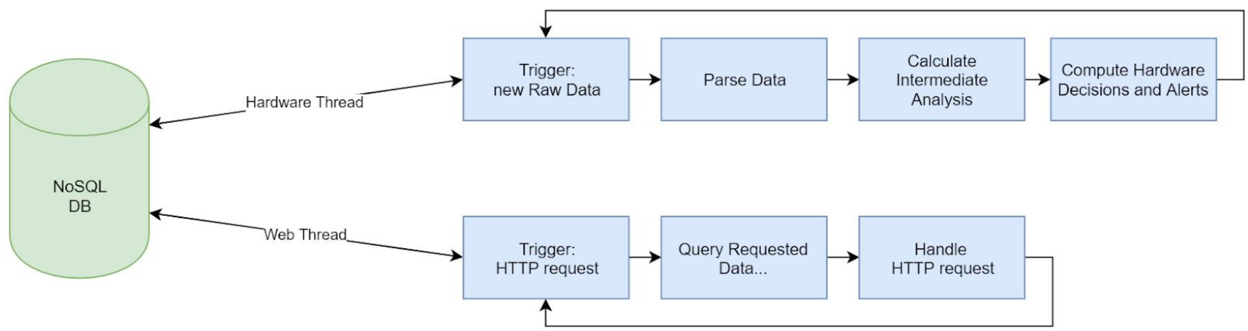

Figure 5: Raspberry Pi Database access threading

Hardware decision logic will be determined experimentally and will be derived from the intermediate analysis.

Air Vent Actuator Module

The air vent actuator is built into standard air vents with rotating sliders and controls the temperature by opening and closing the air vent. It contains:

1. A wifi module (ESP8266 in client mode) that receives commands from the central hub
2. Servo motors that rotates sliders based on received commands
3. Power supply that converts wall voltage (110V - 220V) to 3V-5V DC

HVAC Controller Actuator Module

The HVAC controller actuator is meant to be clamped onto an existing thermostat to manipulate its set temperature. It contains:

1. A wifi module (ESP8266 in client mode) that receives commands from the central bug
2. Two servo motors which are set-up to activate the "increase" and "decrease" temperature buttons on the thermostat.
3. Power supply that converts wall voltage (110V - 220V) to 3V-5V DC

Block Requirement

Module analysis

Module	Requirement	Verification
Temperature Sensor Module	• This module should read temperature with accuracy up to 0.5 ˚C within the range of temperatures 10˚C to 30˚C.	• Set up a heating coil and place the sensor inside the coil; compare the readings against the coil temperature settings, and ensure sensor reading is in the

	 • This module should be able to send correct temperature data to WIFI server reliably within 5-25 meters of range indoors. • This module should have low power and battery should last for at least one week on a single full charge. • This module should have an LED indicator that shows the status (power, connection, etc.) of the module.	range 10˚C to 30˚C is within 0.5 ˚C of the actual temperature over an extended period of time. • Place the sensor 5-25 meters away from the WIFI server with typical obstacles (doors, furnitures, walls) in between and send known test sequence to server and read the correct values at server. • Fully charge the battery and check whether the module can run for a week. • Test LED indicator is displaying correct information under the following condition: looking for connection, connected, and battery low.
Central Hub Module for Data Gathering and Web Service	• This module should be capable of hosting a web service and sending alerts to the user through email or text. • The wifi module should listen and store new temperature readings from all clients. • This module should be powered with standard wall AC outlet (110-220V 50-60Hz).	• Periodically send signals to the central hub and confirm it received and stored them. • Access the web service on another computer in the same network and read the displayed sensor values. • Measure the DC output of power supply and check whether it is within (5V±2.5%, 2A±2.5%)
Air-vent Actuator	• receives signals from Central Hub Wifi Module with instructions. • Air-vent opens/closes vent upon order and maintains that state until a new signal is received.	• Send mock signals through the central hub, and confirm they were received by the air-vent actuator • Simulate signal response for both opening and closing by sending mock signals to the actuator and confirming the correct action was taken
Thermostat Actuators	• receives signals from Central Hub Wifi Module with instructions.	• Send mock signals through the central hub, and confirm they

	● HVAC Controller increases/decreases upon order and maintains that state until a new signal is received.	were received by the HVAC Controller Actuator ● Simulate signal response for both increasing and decreasing temperature by sending mock signals to the actuator and confirming the correct action was taken.

Table 2: Requirement and verification of all modules

Component Analysis

Component	Requirement	Verification
Temperature Sensor	● The components temperature read is accurate up to 0.5 ˚C within the range of temperatures 10˚C to 30˚C.	● Set up a heating coil and place the sensor inside the coil; compare the readings against the coil temperature settings, and ensure sensor reading is in the range 10˚C to 30˚C is within 0.5 ˚C of the actual temperature over an extended period of time.
Temperature Sensor Module: Battery	● The temperature sensor's battery should last up to a week on a single charge. [4] ● Voltage 4V-5V, Current >= 200mA	● Monitor a battery with simulated usage and ensure it stays charged over the course of 7 days. ● Use digital multimeter to test voltage and current output.
Temperature Sensor Module: LED Indicator	● The LED indicator should signify the power state and signal activity of the system.	● The indicator should blink periodically whenever a reading is sent to the central hub
Temperature Sensor Module: Wifi Module Client	● The temperature sensor's wifi module should communicate temperature readings no less frequent than once every 5 minutes.	● We will kick off the system and monitor signal frequency over a period of 30 minutes, ensuring about 60 signals were sent at a period of about 5 minutes.
Central Hub: Wifi Module Server	● The wifi module should listen and cache new temperature readings from all clients. ● The central hub should communicate cached messages	● We will periodically send signals to the central hub and confirm it received and cached them. ● Confirm that cached messages are received by the Raspberry Pi

	with the Raspberry Pi for data storage	
Central Hub: Power Supply	● Less than 10W±5% power consumption (5V±2.5%, 2A±2.5%)	● Run the central hub at full load, measure the power output of the DC power supply using multimeter, and ensure the peak power consumption is below 10.5W
Raspberry Pi: Data Storage	● The central hub should accurately receive communicated messages from the Wifi Module Server for data storage in a dockerized database	● Confirm data sent by wifi module cache arrives to the RBP and is accurately stored by sending mock signals through the raspberry pi, and query internally to confirm the data was received.
Raspberry Pi: Web app	● Raspberry pi capable of hosting a web service and sending alerts to the user through email or text ● Raspberry pi should only send alerts when certain data thresholds are met (e.g. temperature difference between rooms is > 5˚F ● The website displays temperature visualization and alerts should update within 60 seconds of temperature reading delivery from wifi module.	● Set-up a mock alerts that should be viewable on the web app ● Simulate each threshold (and each with mock data, and confirm corresponding alerts were sent. ● Periodically send signals through wifi component to ensure the web app displays information within the 60 second time window.
Actuator Power Supply	● Converts wall power (110-220V, 50-60Hz) to 3V-5V DC Voltage	● Use digital multimeter to measure the output voltage

Table 3: Requirement and verification of all individual components

Schematics

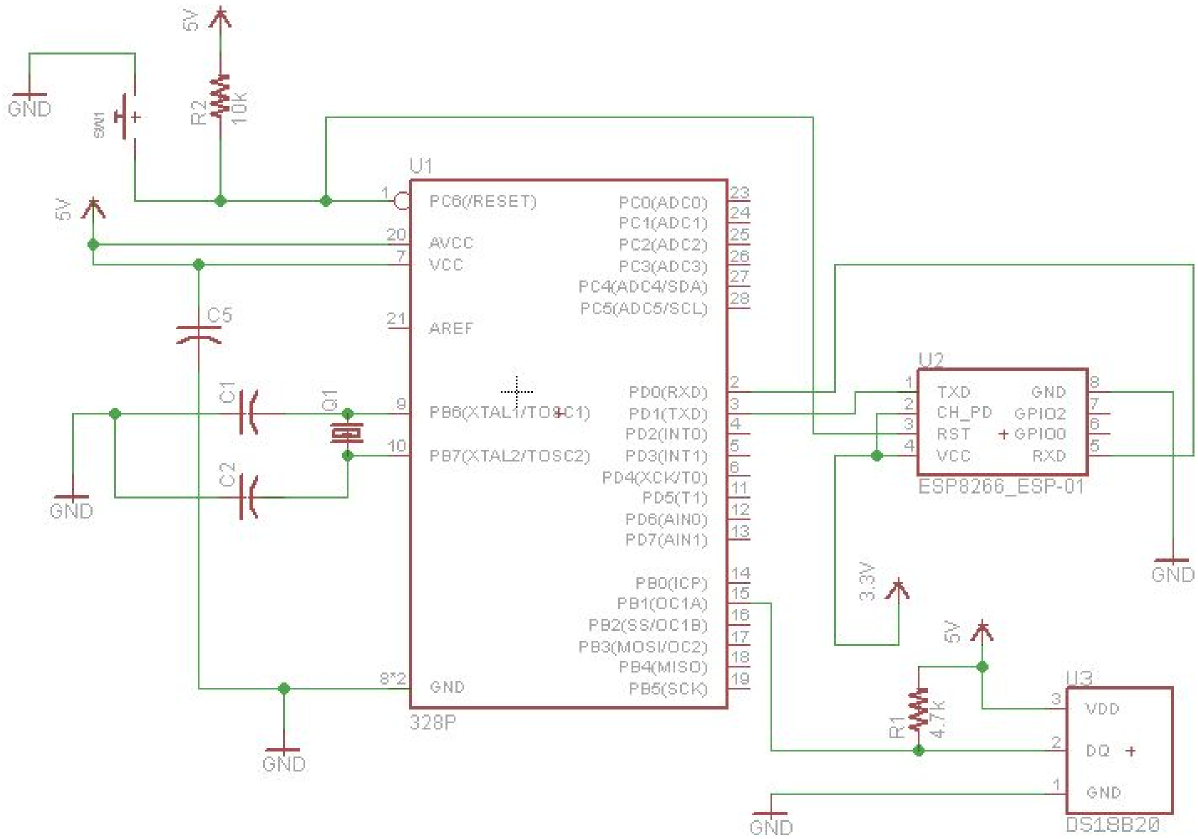

Figure 6: Schematic of Temperature Sensor Module [10][11]

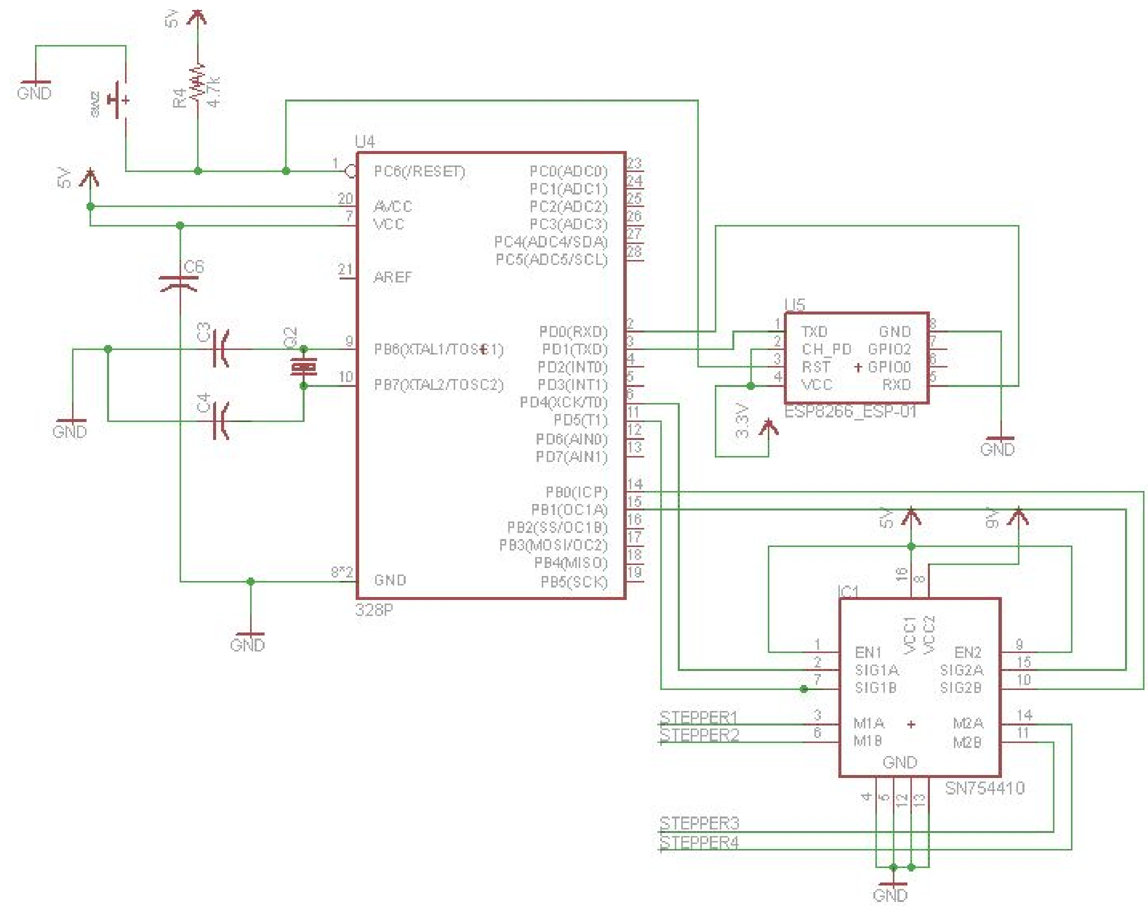

Figure 7: Schematic of Air vent Actuator Module[10][11]

Tolerance Analysis: Battery consumption of Temperature module

A typical Li-ion battery used for powering the microcontroller and the Wifi module is 3.7V, 2000 mA h. It is important that our temperature sensor modules can operate for at least one week under a single charge of the batteries. Therefore, it is crucial to keep the power consumption per day of our temperature sensor module within a certain tolerance range:

$$P_{max} = 2000 \text{ mA h} / 7 \text{ days} = 285.7 \text{ mA h/day}$$

Power Consumption

While Idle

- 20.0 uA - Wifi module in deep sleep mode [7]
- 0.75 uA - ATMega328p microcontroller [6]

While Transmitting (frequency of once every 5 minutes)

- 14 mA - ATMega328p microcontroller [6]
- 1mA - Temperature sensor [8]
- 140 mA - Wifi module in 802.11g active mode [7]

We ignored the momentary blinking of the HLMP-3507 LED, since its current consumption is negligible. If we make measurements and transmit for 10 seconds in every 5 minutes, the power rating per day is:

24 h x (10 s / 300 s x (14+1+140 mA) + 290 s / 300 s x (20 uA + 0.75 uA)) = 124.48 mA h/day

which is well below the power capacity of the battery.

Because this power rating is slightly less than half of the battery capacity and we did the calculation based on the assumption that we transmit for 10 seconds each time in every 5-minute slot, the maximum number of times of transmission in 5 minutes is 2. Therefore, The closest we can sample temperature readings under power constraint is one sample every 2.5 minutes.

Cost

1. Labor

Rate: $40/hour

3 members * 16 weeks x 10 hours = 480 man hours

Total: 480 * 40 = $19,200

2. Parts

Parts (Quantity)	Subtotal
Raspberry Pi 3 Model B (1)	$34.49
SandDisk Ultra 32GB microSDHC (1)	$7.41
Temperature Sensor DS18b20 (3)	$3.95 x 3 = $11.85
WIFI Module ESP8266 (3)	$6.95 x 3 = $20.85
HLMP-3507 (3)	$0.41 x 3 = $1.23
ATMega328p (3)	$4.30 x 3 = $12.90
Stepper Motor (3)	$18.00 x 3 = $54.00
Servo Motor (2)	$12.00 x 2 = $24.00
SN754410 H-bridge IC (3)	$2.52 x 3 = $7.56
Miscellaneous	$15.00
Total	**$189.29**
Grand total	**$19389.29**

Table 4: List of costs

3. Schedule

Week	Haige Chen	Ryan Finley	Heming Wang
2/18	Design review; prototype wifi capability using Arduino and breadboard	Design review; configure Raspberry Pi; set up web server	Design review; test sensor readings using Arduino
2/25	Transfer Wifi submodule design to standalone ATMEGA328p in place of Arduino	Develop Python script for interfacing between Wifi module and Raspberry Pi	Transfer temperature sensor submodule design to standalone ATMEGA328p in place of Arduino
3/2	Design, test and CAD ATMEGA328p peripheral circuit for temperature sensor module; make sure Raspberry Pi receives correct data	Design, test and CAD ATMEGA328p peripheral circuit for temperature sensor module; make sure Raspberry Pi receives correct data	Design, test and CAD ATMEGA328p peripheral circuit for temperature sensor module; make sure Raspberry Pi receives correct data
3/9	Test power consumption and potentially modify algorithm to save power	Design web front end; set up and test data display and alerts system	Test battery power supply circuit and duration of continuous operation
3/16	Spring break; wrap up any unfinished work from previous weeks	Spring break; wrap up any unfinished work from previous weeks	Spring break; wrap up any unfinished work from previous weeks
3/23	Prototype air vent actuator circuit with Arduino	Test actuator can correctly receive commands from the central hub through Wifi	Build air vent actuator (both circuit and mechanical design; make sure the sliders can rotate correctly
3/30	Transfer actuator design to standalone ATMEGA328p	Design decision algorithm that determines how to change the air vent based on temperature readings	Test actuator can turn according to the commands sent from central hub; test power circuit of air vent actuator module
4/6	Wrap up any unfinished work from previous weeks; if there is time, start on the reach goal, which involves a button	Wrap up any unfinished work from previous weeks; if there is time, start on the reach goal, which involves a button pushing	Wrap up any unfinished work from previous weeks; if there is time, start on the reach goal, which involves a button pushing

	pushing mechanism that can change the HVAC controller settings	mechanism that can change the HVAC controller settings	mechanism that can change the HVAC controller settings
4/13	Put all the modules together; tweek parameters in decision algorithm if needed; if there is time, keep working on the reach goal, which involves a button pushing mechanism that can change the HVAC controller settings	Put all the modules together; tweek parameters in decision algorithm if needed; if there is time, keep working on the reach goal, which involves a button pushing mechanism that can change the HVAC controller settings	Put all the modules together; tweek parameters in decision algorithm if needed; if there is time, keep working on the reach goal, which involves a button pushing mechanism that can change the HVAC controller settings
4/20	Final debugging and testing	Final debugging and testing	Final debugging and testing
4/27	Final Presentation	Final Presentation	Final Presentation

Table 5: Work schedule

Risk Analysis

The main selling point of our system is that all communications are carried via wifi, but bookkeeping all of that data can be challenging, especially when there are large number of sensors connected to the network. Because the central hub cannot request data from the sensors, we need to make sure our communication protocol can handle the situation where multiple sensors send their data at once. The central hub must be capable of caching and queuing the data properly so it does not lose import information.

Additionally, while we will use an aftermarket processor as the brain for our temperature sensor, we are designing the supporting circuit ourselves. Obviously the circuit has to function properly, but also be simple to lower the cost and minimize build time, while at the same time offer enough protection mechanisms so any component failure won't damage the rest of the circuit.

The third concern is that our sensors are powered by batteries, therefore we have to make sure the sensor assemblies meet the power requirement such that they do not drain batteries too rapidly to render the system impractical.

Ethics and Safety

Since there already exist zoned HVAC product in the market, it is important that our final product be unique in certain aspects. Careful analysis of competitors, and identifying components of their fundamental technology are required to ensure that our product does not accidentally infringe the intellectual property rights of other companies and individuals. While researching the competition, we can be tempted to ‘borrow’ ideas from other product without the consent of the owners of that idea. ACM code of ethics 1.5 states that we are to respect the work required to produce new ideas, and entry 2.3 requires us to respect existing rules pertaining to professional work [5]. Should it become clear that we have to incorporate some ideas and/or technologies of other parties for our product to be functional, we must obtain explicit consent from the owner of said ideas/technologies before using them in our design.

Data security is of major concern in the days of massive security breaches. Since our product collects room temperature data, which can be considered private information, and sends them on wireless network to transmit data, steps need to be taken to ensure that those data are securely encrypted before our product can transmit them, thus preventing compromising users’ privacy should such data been intercepted by a third party. It is also important that our product offers protecting against hijacking should third parties attempt to take control of our product in order to collect sensitive personal data without the consent of the owners. All of the precautions are in accordance of ACM code of ethics 1.6 which states computing professionals should respect privacy [5]. Our responsibility to user privacy also applies to ourselves such that we cannot deliberately engineer vulnerability and/or back doors to our product so we can collect our users’ data for our own benefits.

We do not believe our project poses any danger to our group members or the end users. All of the components, the temperature sensor, the central hub, and the actuators, are low power devices, thus there is very little risk of electric shock. Additionally, we use non-rechargeable, off the shelf alkaline batteries to power all the devices, therefore we do not have to be concerned with overcharging and battery thermal runaways are extremely unlikely. Finally, the central hub is powered by an aftermarket, UL certified DC power supply so we are not under the risk of fire and/or electrocution.

Reference

[1] "Zoning", *Lennox*, 2019. [Online]. Available: https://www.lennox.com/products/comfort-controls/zoning. [Accessed: 08- Feb- 2019].

[2] "Multi Zone Home Heating and Cooling - Modernize", *Modernize*, 2019. [Online]. Available: https://modernize.com/home-ideas/12694/pros-cons-multi-zone-heating-cooling. [Accessed: 08- Feb- 2019].

[3] "ESP8266 - NURDspace", *Nurdspace.nl*, 2019. [Online]. Available: https://nurdspace.nl/ESP8266. [Accessed: 08- Feb- 2019].

[4] "ESP8266 Voltage Regulator For LiPo and Li-ion Batteries", *Random Nerd Tutorials*, 2019. [Online]. Available: https://randomnerdtutorials.com/esp8266-voltage-regulator-lipo-and-li-ion-batteries/. [Accessed: 08- Feb- 2019].

[5] “ACM Code of Ethics and Professional Conduct,” Association of Computing Machinery. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 08- Feb- 2019].

[6] Ww1.microchip.com. (2019). [online] Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf [Accessed 22 Feb. 2019].

[7] Ww1.microchip.com. (2019). *Home | Microchip Technology*. [online] Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdfhttps://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf [Accessed 22 Feb. 2019].

[8] Ww1.microchip.com. (2019). *Home | Microchip Technology*. [online] Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdfhttps://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdfhttps://datasheets.maximintegrated.com/en/ds/DS18B20.pdf [Accessed 22 Feb. 2019].

[9] "ESP8266 Power Consumption - ESP8266 Developer Zone", *Bbs.espressif.com*, 2019. [Online]. Available: https://bbs.espressif.com/viewtopic.php?t=133. [Accessed: 22- Feb- 2019].

[10] "How to connect an ESP8266 to an Arduino UNO — Teo's blog", *Teo's blog*, 2019. [Online]. Available: http://www.teomaragakis.com/hardware/electronics/how-to-connect-an-esp8266-to-an-arduino-uno/. [Accessed: 22- Feb- 2019].

[11] "Build Your Own Arduino Board", 2019. [Online]. Available: https://www.electroschematics.com/8064/diy-arduino-making-your-own-arduino/. [Accessed: 22- Feb- 2019].