Smart Vent

Charles Durfee / John DeLapp

Abstract:

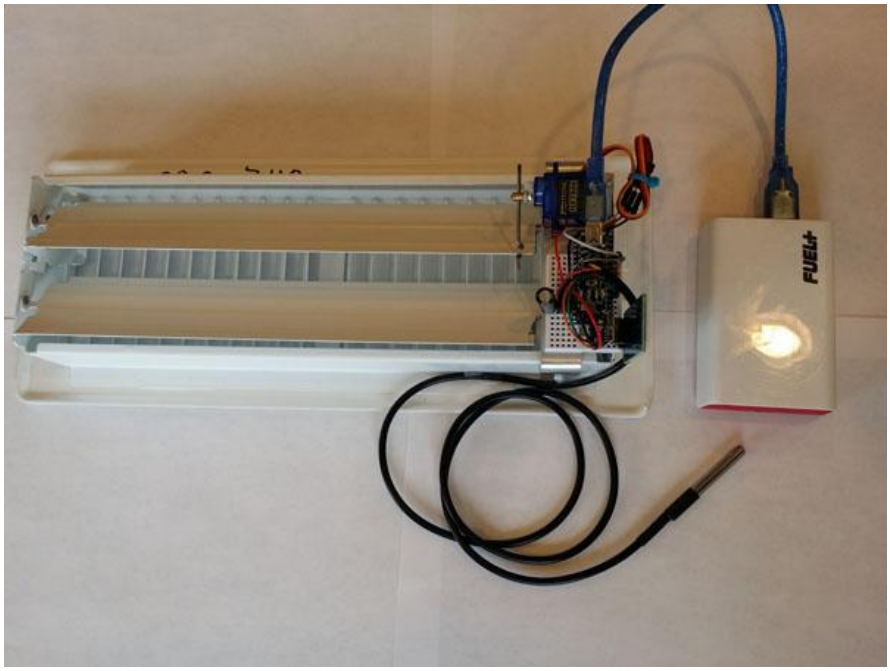
Temperatures in some homes can vary greatly from room to room; wouldn't it be great to control each room's temperature independently? With our Smart Vent System, it will be quick and easy to monitor and automate the temperature in each room. Whether you prefer a consistent temperature across the house, maximum energy efficiency by turning off heat to unused rooms, or even setting different temperatures in each separate room; you can have the precise control you want with the Smart Vent.

Our system begins with standard heat vents (or registers) similar to those found in your local hardware store, but they are better... Our vents have motorized movement that can precisely control the airflow from your home's HVAC system into the room. Each room temperature is monitored and each vent is actively adjusted to maintain the desired temperature for that room. An easy web interface or a convenient app for your phone or tablet can be used to set the desired temperature for each room.

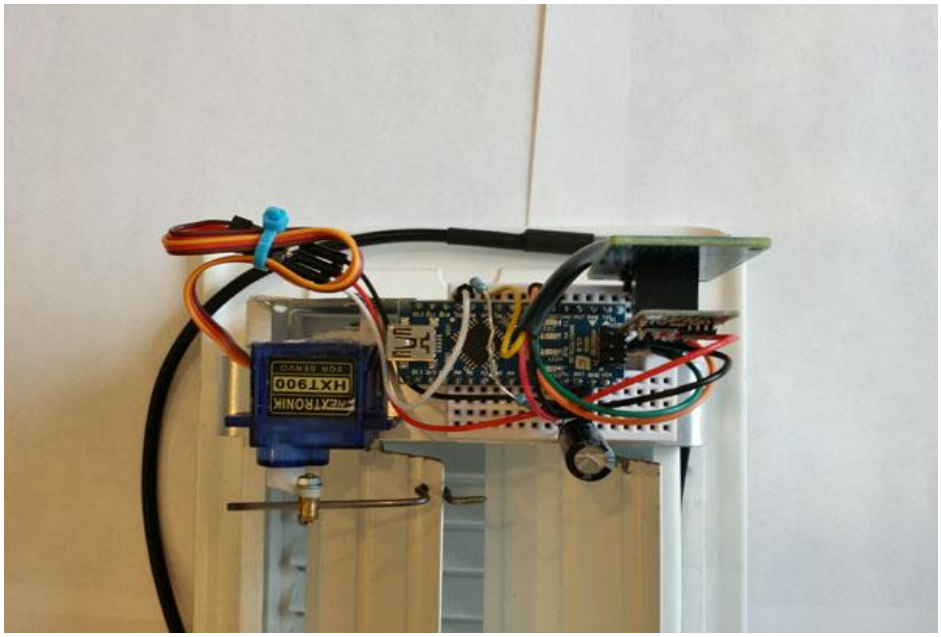
How it works / technical details:

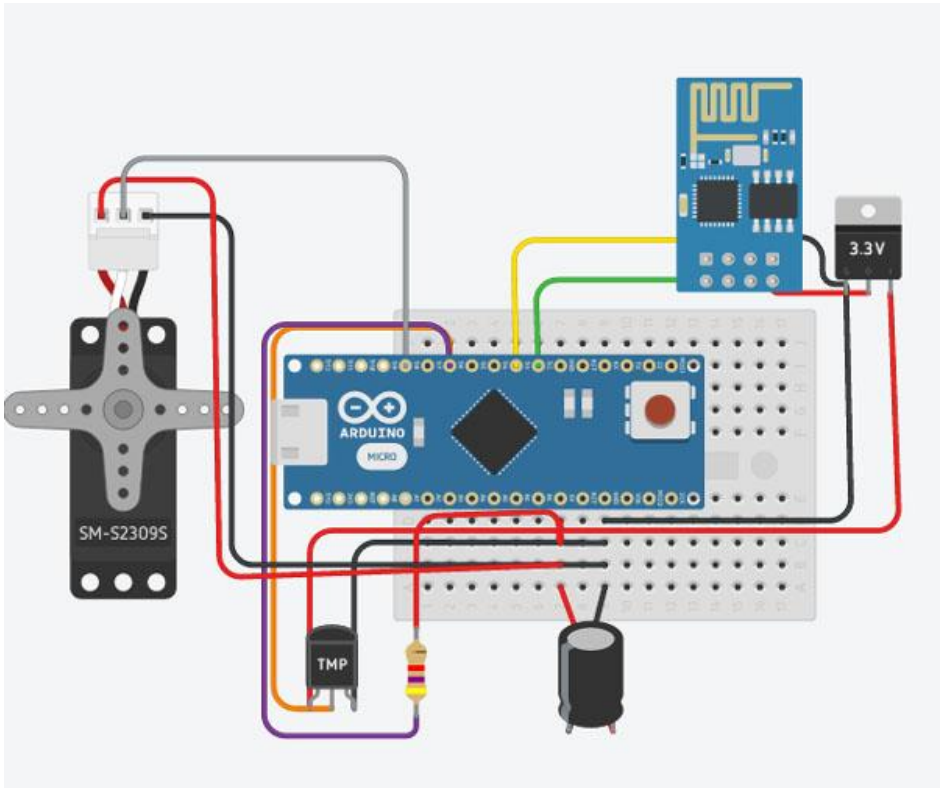
We made two different prototypes which both seem to work, but have their advantages and disadvantages. The first uses a servo motor to control the vent position. The second uses a stepper motor for control. Each version has some differences in electrical circuitry as well as some differences in the code.

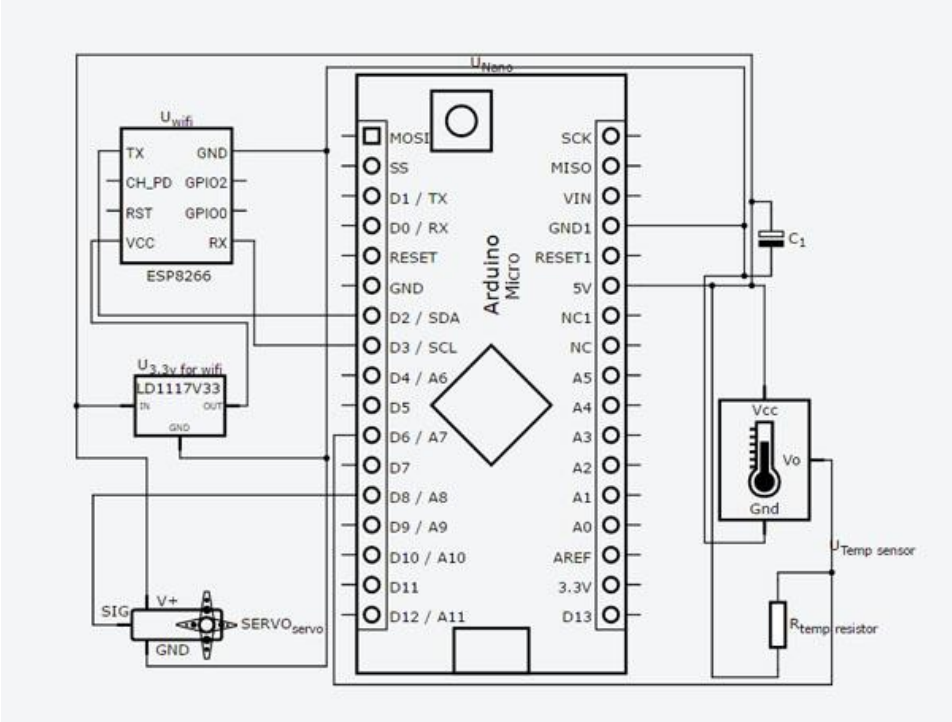
Prototype #1:



This prototype uses a servo to control the movement of the dampers, which can be opened and closed to restrict airflow from the HVAC system into a room. The servo is controlled by an Arduino nano which has wifi connectivity (ESP8266). Room temperature is sent from the ds18b20 temperature sensor, via the wifi to the control server, which replies with a number that represents the needed vent position. The vent position is then set and will stay in that position until it receives a command to change the vent position due to changing temperatures, or a request by the user to change it.







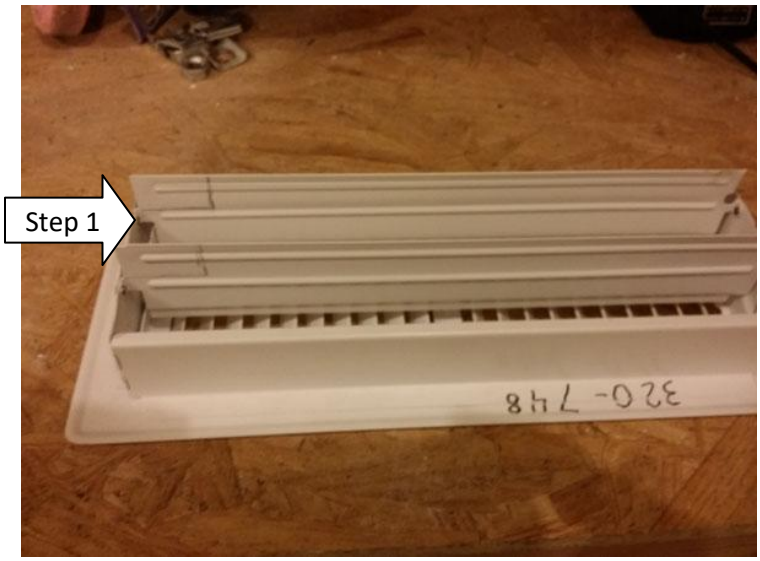
Construction of vent1:

Parts:

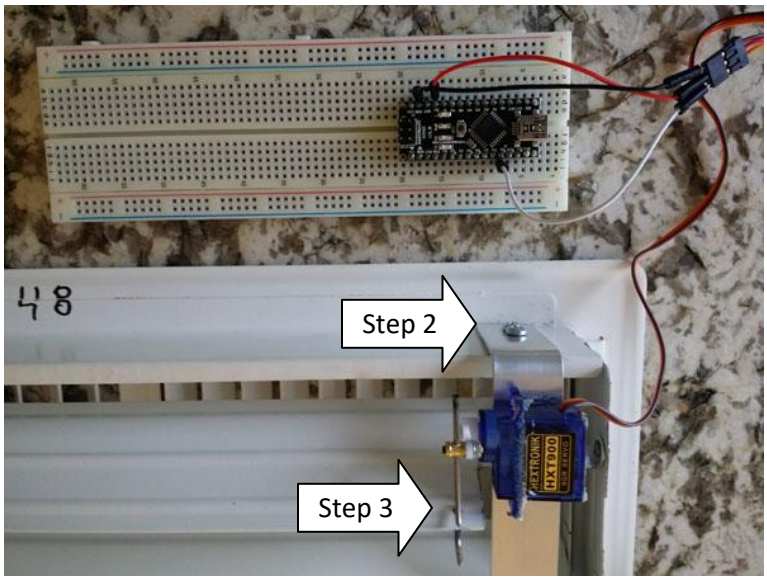
1. 12 inch vent
2. Breadboard
3. Breadboard wires
4. Arduino Nano
5. ESP8266 wifi
6. ESP-01 adapter 3.3v regulator
7. ds18b20 temperature sensor
8. 4.7k ohm resistor
9. 1500uf Capacitor
10. HXT900 Servo
11. USB power source/battery and cord

Construction:

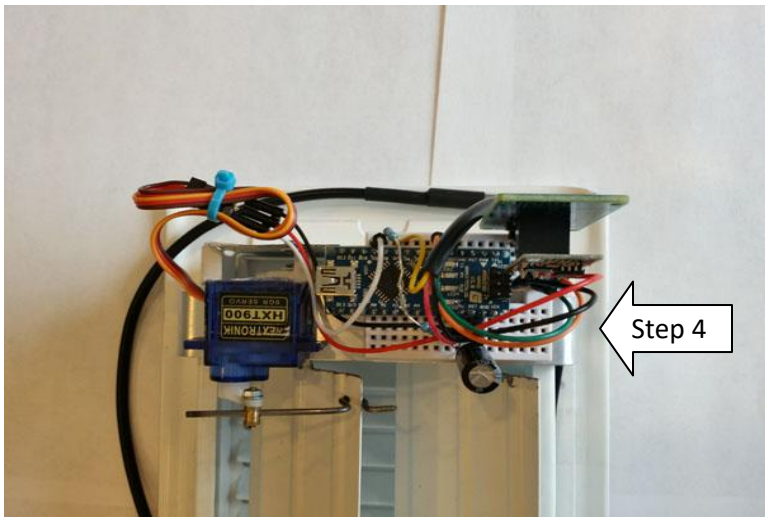
1. The first step is to cut a small portion of the vent flaps to make room for the servo and Arduino:



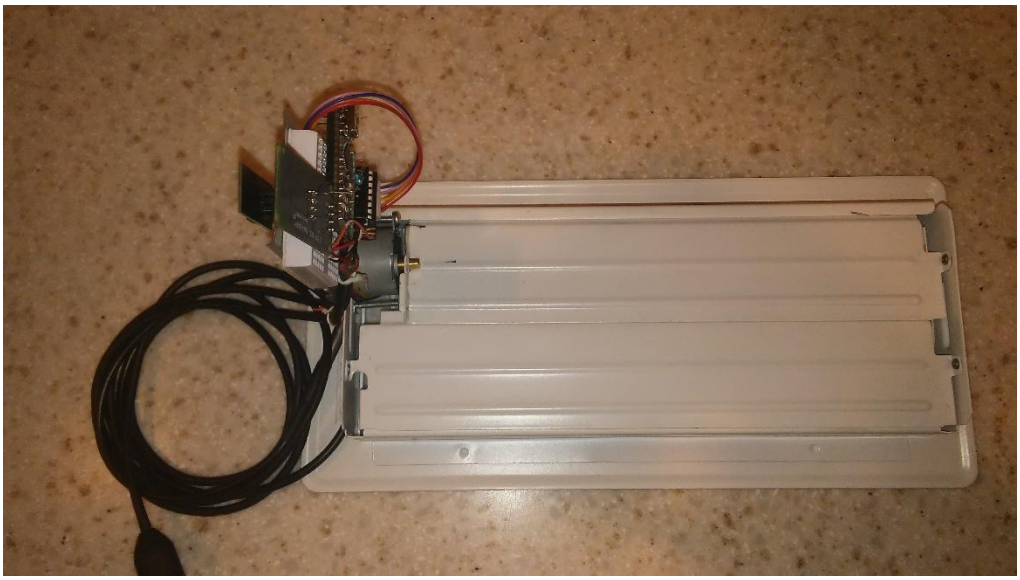
2. Next cut, bend, and mount a thin piece of 1in aluminum for the servo and Arduino to rest on:



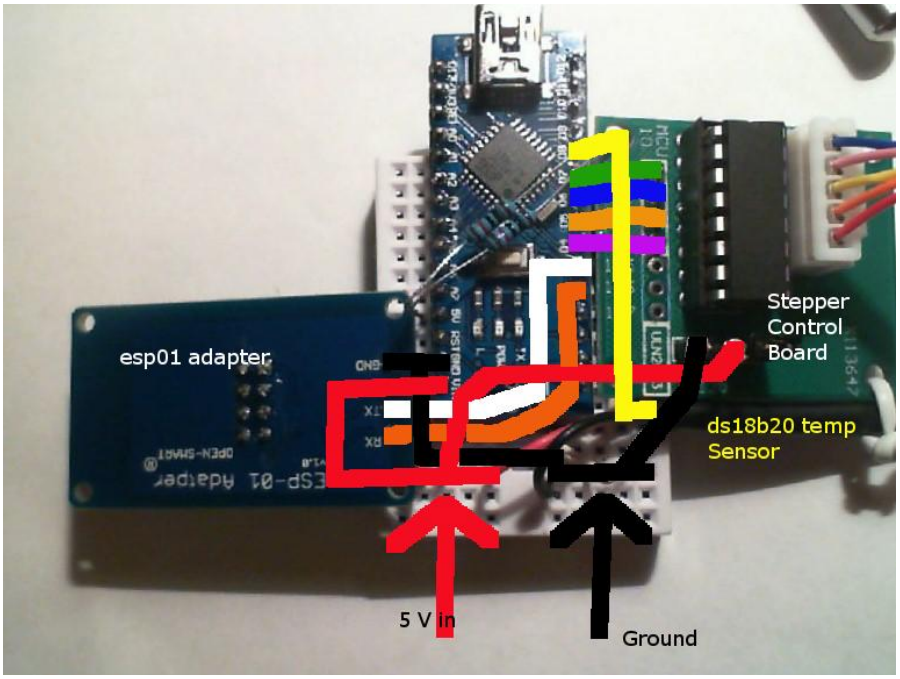
3. Drill a hole to connect a thick wire to the vent flap and the servo arm
4. Wire the components and mount the components next to the servo, be cautious that the wires do not get caught in the vent movement



Prototype #2



This prototype vent uses a stepper motor to control the movement of the dampers. Its function is the same as the servo type vent, using Arduino, ESP8266 (wifi), and ds18b20 temperature sensor, but it also includes a small control board for the stepper motor.



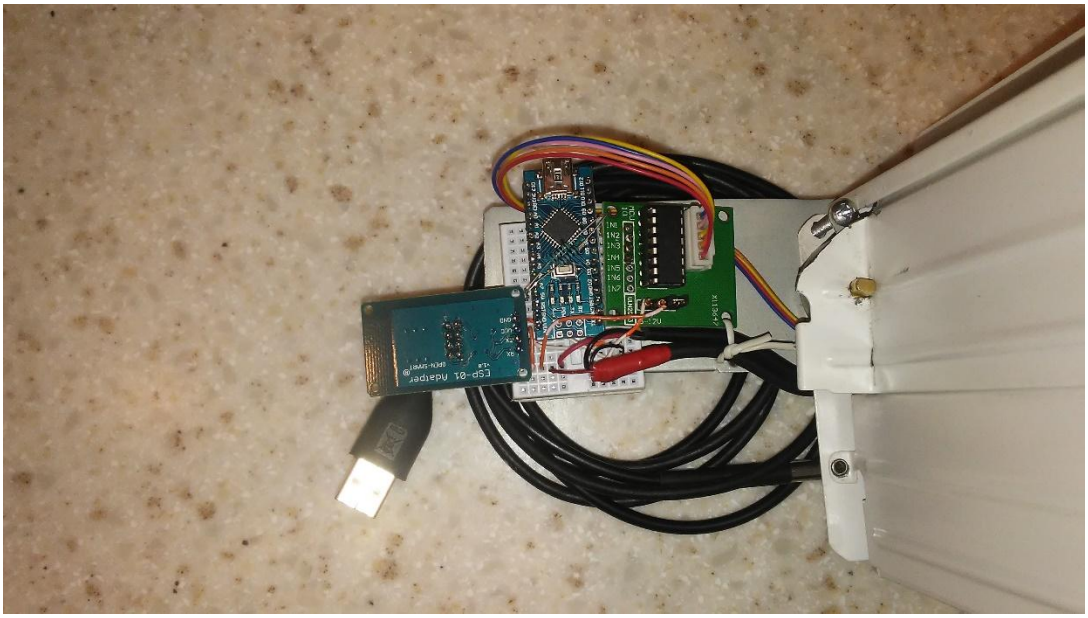
Parts:

1. 12 inch vent
2. Breadboard
3. Breadboard wires
4. Arduino Nano
5. ESP8266 wifi
6. ESP-01 adapter 3.3v regulator
7. ds18b20 temperature sensor
8. 4.7k ohm resistor
9. 28BYJ-48 5V Stepper Motor
10. ULN2003 Driver Board
11. USB power source/battery and cord

Construction:

1. This vent was constructed by trimming the first register and bending a flap upward. This flap allows room for the motor to be mounted to the frame and a hole is drilled to fit the motor shaft in the upturned flap. Take care to trim any protruding edge that restricts the movement. Make sure that alignment and location of pivot remain similar to original.

2. Next mount motor and bracket for electronics.



Advantages/Disadvantages of Servo Motor versus Stepper Motor- What we learned:

The advantages/disadvantages of the servo motor are:

1. Advantage: Slightly reduced complexity in electrical circuitry.
2. Advantage: Servo position is easily known
3. Advantage: Slightly less code required.
4. Disadvantage: Low torque- the 9 gram servo was approaching its limit with a 12 in vent
5. Disadvantage: Noise- the servo used was loud
6. Disadvantage: Durability- this type of servo is easily stripped if something blocked the vent

7. Disadvantage: Complicated mechanical setup- servo arm and wire attaching to vent flap

The advantages/disadvantages of the step motor are:

1. Advantage: More torque from the motor, if larger vent is required this would be the better choice
2. Advantage: Durability- motor can be stalled without stripping gears
3. Advantage: Noise- very quit operation
4. Disadvantage: stepper motor position must be re-zeroed after power off or reset.
5. Disadvantage: additional code for resetting the stepper to a known point after power off as well as additional code to produce the function to move the stepper motor.

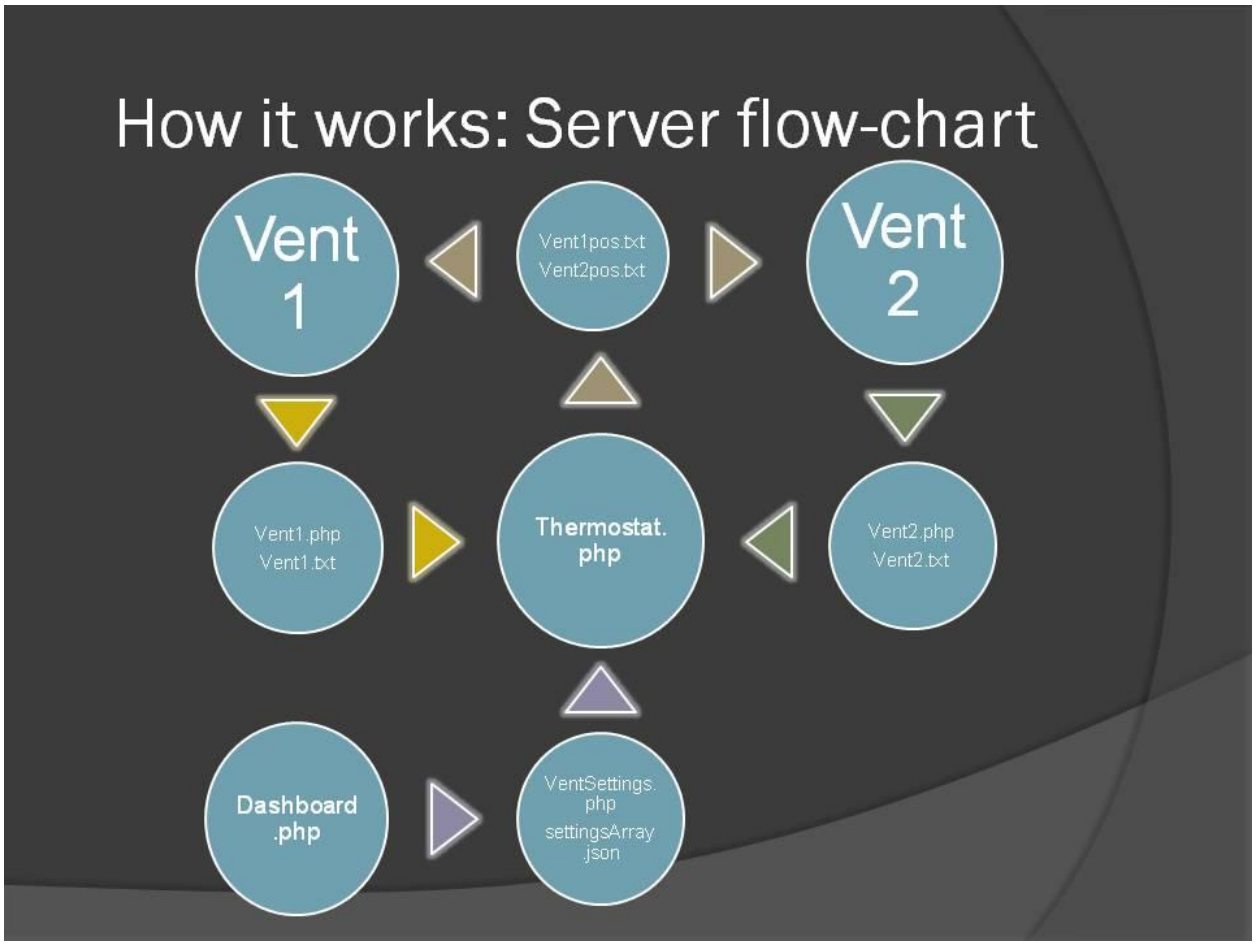
Web Server / User interface:

We used an Amazon web server. The Web server / user interface can be accessed at the following link:

http://ec2-52-41-237-167.us-west-2.compute.amazonaws.com/dashboard.php

Data movement:

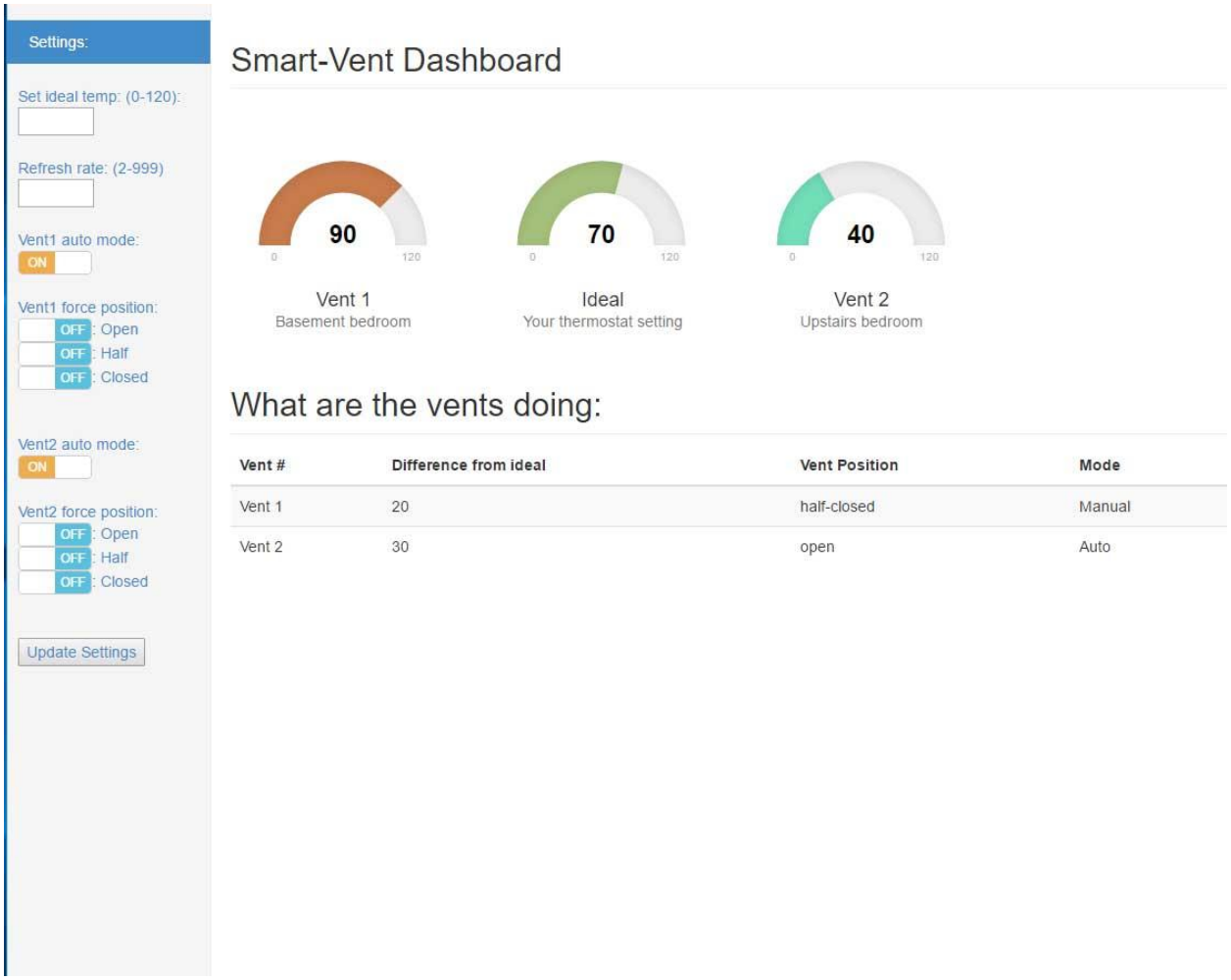
The smart vents send a php GET request to the Web Server via a TCP connection on port 80. The Web Server captures the temperatures for each vent and writes the temperature to file. Each vent temperature is captured separately to avoid read/write conflicts. The temperature is written to file to ensure data is not lost as the user interface page updates with the data on user selectable intervals. When the GET request is received by the server, it responds with the position the vent should be in. The smart vent position is then parsed by the vent and the position changes if needed.



Interface:

The interface allows the user to select an ideal temperature, refresh rate, and auto or manual modes. The interface has three color adjusting gauges reflecting the each vent and the ideal temperature. It also has information on what each vent is doing (difference, position, mode).

Manual mode allows the user to manually open, half-close, or close a vent as needed. Automatic mode will adjust the vent that is furthest from ideal in relation to the other vent. If both vents are the same distance from ideal we would likely want them both open as it is evenly cooling/heating. We also do not want a situation to occur where both vents are closed as it may impact the HVAC system. Therefore, if the vent difference is between 5 and 10 degrees further from the other vent difference it will half-close the vent. If the vent difference is greater than 10 degrees from the other vent difference it will fully close the vent.



The interface is based on the bootstrap dashboard example linked below:

http://getbootstrap.com/getting-started/#examples

The gauges use the Just Gauge plug-in linked below:

http://justgage.com/

The toggle buttons use the bootstrap switch plug-in linked below:

http://www.bootstrap-switch.org/

Goals (accomplished)

Precision temperature readings sent to interface/server.

Vent control dependent on individual room temperature readings.

Connectivity to enable main interface communication and smart control.

Basic interface and smart control of vents.

Improvements (incomplete)

Perfect the algorithm for vent control. This will need some real world testing and logging to ensure that each vent opens or closes as needed to maintain temperatures as requested.

Separate temperature modules to move temperature readings to a location in the room that is further away from the heat vent. To ensure more accurate and stable temperature readings.

Make the vent device cordless, which requires some gains in electrical efficiency and some way to keep a battery charged, like solar.

Make a main control unit which could coordinate smart vent interaction with the HVAC system as well as control the furnace and A/C.

Make a mobile app to control temperature.

Conclusions: what we learned

-Servo is easier to implement for a DIY project.

-Step motor is more durable/quiet, would be better for a production unit.

-Using the ESP8266 wifi module was difficult (steep learning curve), a built in unit such as the photon may have been easier to implement.

-The server files/flow became complicated due to all the separate files. Ajax may be a solution to reduce files and complexity.

Citations:

Here is a list of tutorials and resources that were used in our development:

1. General info on using esp8266: https://github.com/esp8266/Arduino
2. ESP8266 wifi server tutorial: https://www.youtube.com/watch?v=AGn6Ksf3OEw
3. Temp sensor tutorial: https://www.youtube.com/watch?v=qxEclOy6jpI
4. Send data to web site tutorial: https://youtu.be/q02f4sPghSo
5. Bootstrap dashboard example: http://getbootstrap.com/getting-started/#examples
6. Just gauge: http://justgage.com/
7. Bootstrap switch: http://www.bootstrap-switch.org/

8. Stepper motor example: https://twit.tv/shows/know-how/episodes/209