Smart Home Project Synopsis

GroupList

Levi Balling - leviballing@gmail.com

Dario Bosnjak -bdario1@comcast.net

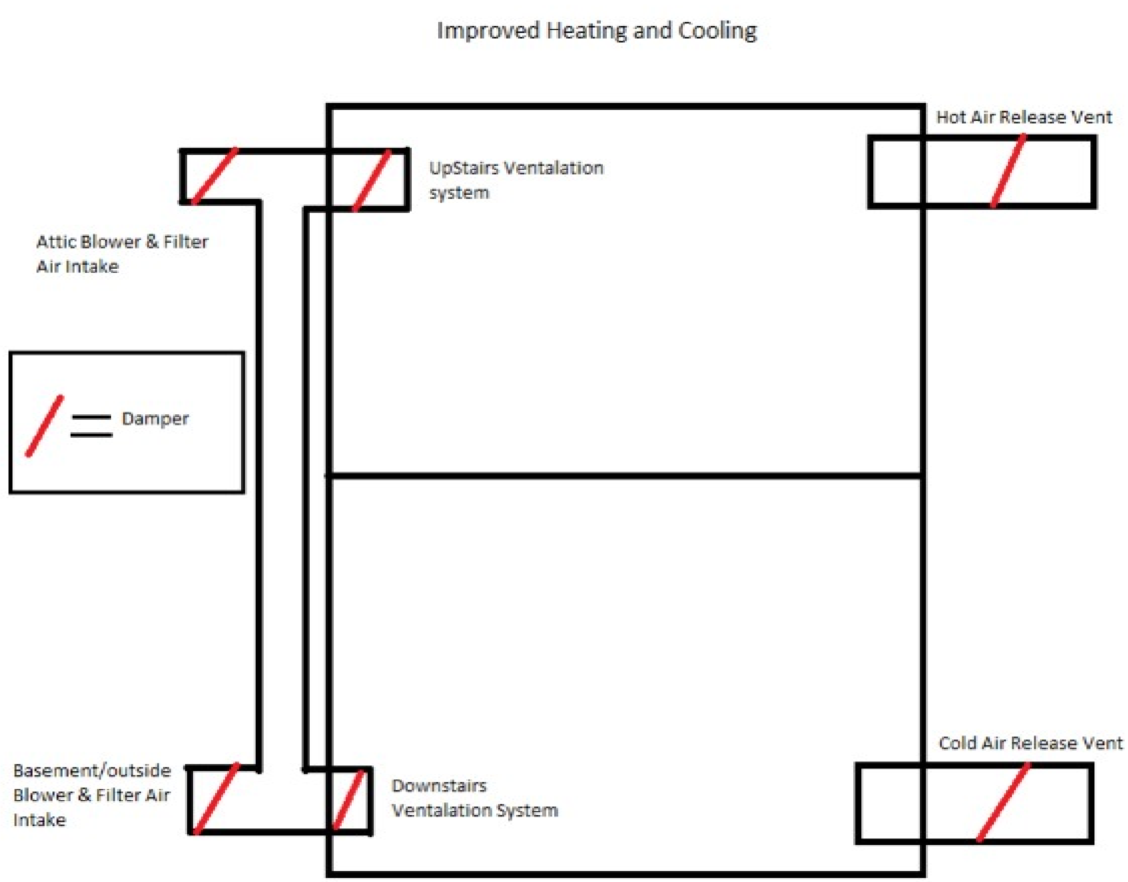
Christopher S. Johnson - christopher.s.johnson@utah.edu

Todd Rogers - u0634986@utah.edu

Functionality

Temperature Controller

The goal of temperature control is to make the home more efficient, save money and allow the temperature to be evenly displaced throughout the home. This will be controlled using temperature sensors displaced in each room, in the attic and outside. Heating and cooling systems in new homes are designed to function without taking advantage of any outside conditions. This makes most homes inefficient when whether conditions are better outside than in the home. An example of this occurs during summer



nights when it is 63 degrees outside (noaa.gov), and 75 degrees inside with the A/C on to lower it to 72 degrees. This turns on the A/C unit, and costs a lot more than necessary.

This system is useful during the winter too. When the temperature is warmer in the attic, the system will pull air from the attic into the basement. It will also open a cool air release damper in the basement to outside and allow the cold air to flow out of the house. If the temperature is warmer outside, and not in the attic (unlikely in winter but common in spring and fall), it will draw from outside, and displace it into the basement, and release the cold air. When it is colder in the attic than it is outside then it will maintain regular temperature control.

During the summer if the temperature is cooler outside, then it will pull air from the basement or outside into the upstairs ventilation system. It will also open a hot air release vent, allowing better air flow. If it is warmer outside, and cooler in the attic (unlikely) it will distribute like before.

Room Sensor

Tracking the occupied or vacant status of each room allows the house to turn lights and appliances on and off automatically as well as manage the temperate in individual rooms without human intervention.

Determining this room occupancy status has been attempted in the past using a motion sensor and a countdown timer. Whenever there was motion detected, the room would be deemed occupied, and the countdown timer would be reset to begin counting down toward zero. When the timer reached zero without being reset by further motion, the room would be deemed unoccupied.

The problem with this is that if the duration of the countdown timer is short, then if motion is not detected on a frequently basis, the room may be inaccurately assumed vacant when it is still occupied; if, on the other hand, the duration of the countdown timer is long, then the room will be deemed occupied long after all persons have vacated it.

The solution to this problem is to add a third component to supplement the motion sensor and countdown timer: a light beam interruption detector. The light beam interruption detector is positioned between the two posts of each doorway or between the sides of each archway dividing spaces into separate rooms. The occupied state of each room can only change immediately following the interruption of the light beam as people pass from one room or area to another (assuming people do not enter and leave through the windows). Furthermore, motion will quickly follow an interruption of the light beam if the room becomes occupied.

Due to these facts, once the light beam is interrupted, the house can determine in a very short interval—in as little as 5-10 seconds—whether a room bordering that light beam has become occupied or vacant. If the room is found to be occupied during that time period due to motion having been detected, then no further motion is required to maintain that status—just so long as the light beam is not interrupted. If the room is mistakenly assumed to be empty after that time interval, because of a lack of motion during that interval, all it takes is one—just one—wave of an occupant to the motion sensor and the house will change the status of the room to occupied.

WIFI

Implementing an Arduino Microcontroller will allow us to gather the data from different sensors throughout the house (heat sensors, light sensors, laser trip wire) and broadcast the data to a server that will be set up using Wifi Interface. A wifi shield for Arduino will be connected to the server using the TCP/IP protocol with an assigned IP address that the server will use to listen to the information on that IP. We are planing on running an Apache server where we could store the data that comes in from the micro controllers. This data will be formatted and displayed on a secure website which is accessible from any internet connections. Using the wifi we will be able to send and receive data. Once we get the information from a temperature sensor in one of the rooms, and we decide to turn on the fans in the room to make it cooler, we would send a command to that micro controller that will be in charge of running the fans, to start to pump the cold air inside that room. Below is a diagram to illustrate wifi operations in the house.

Room Controller box

Each room will have a single controller box, that will contain a MCU that will connect a variety of devices that provide different features of the smart home. The box will connect to the room monitoring sensors and room fan vents. It will also be connected to an outlet 12V 1.5 Amp or some greater power supply.

Server

The central server is the hub of all activity and performs two important functions. First, it takes input events created by sensors and other talking devices, and it executes scripting commands in response to these input events to generate output events in the form of commands that it sends to devices like switched outlets. Second it creates a web-based interface for interacting with the home's various systems as well as for editing the scripting commands that the home executes in respond to various stimuli.

We have chosen Python as the language in which we will implement the core server software that tracks the state of input devices and sends command events to output devices. User plug-ins will be written in Python to hook into the core server software and make it do things in response to input or other events such as things occurring on a schedule. The reason for choosing Python is that it is an easy-to-learn interpreted general purpose language which can add new code like plug-ins on the fly without recompiling, relinking, and restarting the whole program.

Outlet Controller

The outlet controller will be in the form of a power strip, it will be mobile and reconfigurable. This power strip will plug into any standard wall socket, and be unwired except for power. The sockets will be switched on and off via the controller box, or the server computer using the internet. Power strips will enable persons to control power to any device which would normally run from a 120v 60 Hz outlet. The power strip will monitor power for devices which are plugged in. This data will be available at the controller boxes and server computer.

Implementation

Temperature Control

The temperature control system will use a variety of components. Temperature sensors will be placed in every room, outside, and the attic; we are using LM335AZ temperature sensor. They will be calibrated together to allow for relatively accurate temperature readings. We are going to be using dampers to control the air flow. These dampers will be built using stepper motors, gears, and tin. This allows our project to meet Utah building code. For the blowers we will have 2 different type of blowers, a furnace blower, and small fans for the individual room blowers. These will be controlled by using the same relay as our outlet controller. The individual room blowers will be Delta AFB1212SHE-4F1A, They are variable speed fans that are controlled by a frequency's duty cycle.

This will allow us to pull more air out of certain specific vents. Each room MCU will connect one PWM signal line to the fan and control the speed of the fans.

For our demo we will be using smaller fans that will show the effectiveness of our system. Our demo large fan will be a Delta AFB1212SHE-4F1A. This will pull air from outside, and from the attic. We will access smaller fan by removing them from old broken computers. Both of these fans will be powered with 12V, the relay that will power these fans run on 5VDC.

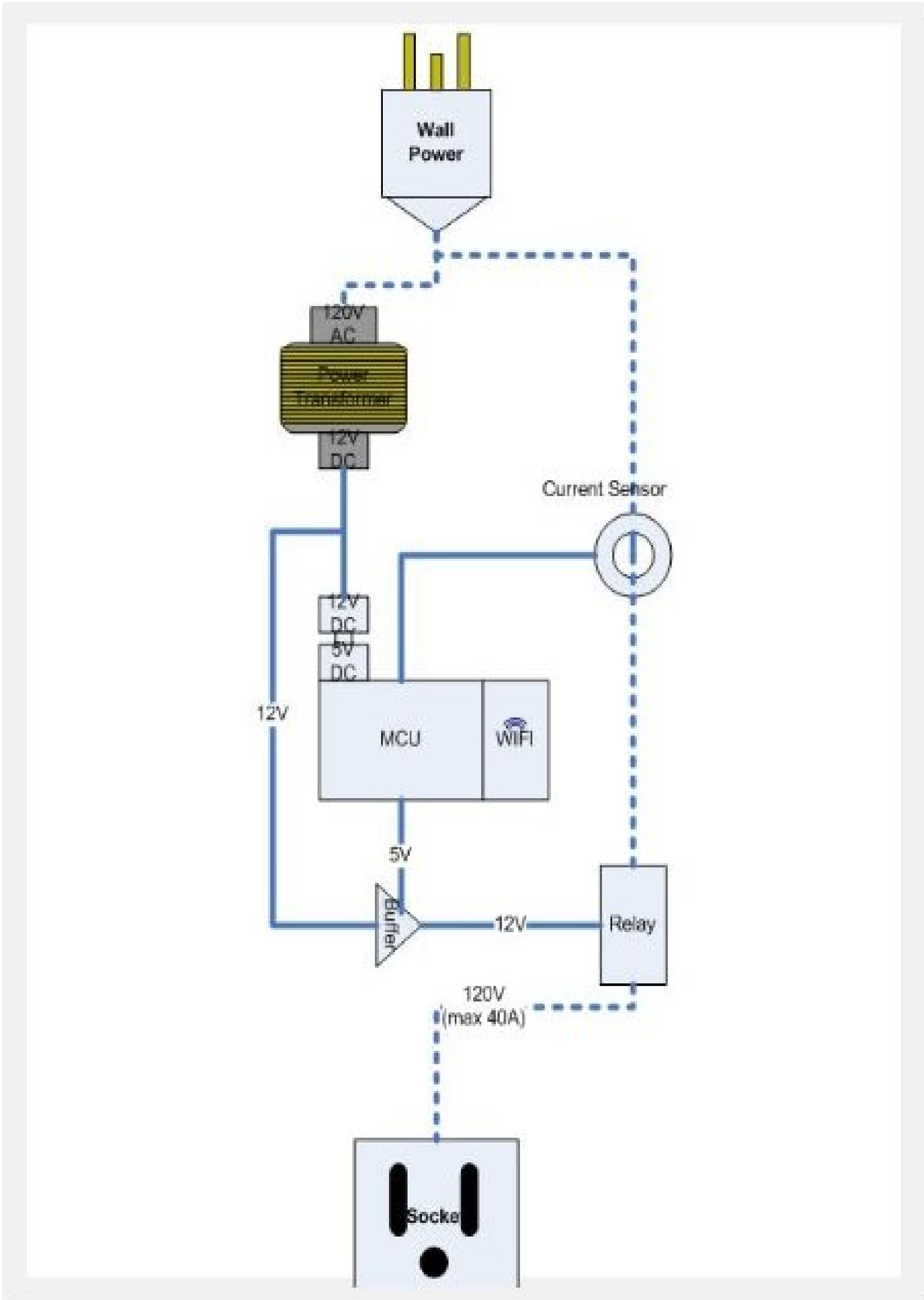
WIFI

Wifi Shield with Arduino will consume about 5V-7V. The voltage regulator will be able to provide enough power for it to run. The Wi-Fi will be 802.11b at 1Mbps and 2Mbps throughput speeds. It will give us the ability to establish secure and insecure networks such as WEP (64-bit and 128 bit) and WPA/WPA(TKIP and AES) PSK security. This chip has low power usage.

- Sleep mode: 250μA
- Transmit: 230mA
- Receive: 85mA

Outlet Controller

The module will provide power to the MCU and WIFI chip, power sensors, LED's, and the outlet. A 120VAC to 12VDC power supply will be used to provide power to the relay. The 12VDC will also power the on-board electronics via a 3-5V voltage regulator. Monitoring these power sockets will be done using a hall effect current sensor. These sensors will possibly need to be combined with a low amperage sensor for the full range of use. Relays will be used to switch power on and off. An automotive relay will be used to switch power from the wall. It can source 40 amps through contact while the coil runs on 12VDC.



Wall
Power
120V
AC
Power
Transformer
12V
DC
Current Sensor
12V
DC
5V
DC
12V
MCU
WIFI
5V
Buffer
12V
Relay
120V
(max 40A)
Socke

Room Controller Box

The Controller Box in each room will be connected to the room monitoring system. It will also supply power to it, and communicate across a cable. The cable could be installed in the walls of the home due to low cable power.

The controller will also interface with the smaller room blower using a single PWM signal line to control the speed of the fan. The fans speed is controlled by the duty cycle. It will provide power to the fans with 12VDC with a LM7812 voltage regulator. This will supply the fans with 1.00 amp of current. The voltage regulator may not be necessary if we have a 12VDC power supply.

The controller will also have a LM335AZ Temperature sensor that provides an output of 10mV/K. We can use this to control the temperature of the room. This will send a command to the server, and the server will control the dampers, along with the individual fans in each room.

Interfacing together

The different rooms will be connected to a server over a router. The server will allow for an easier programming interface. We will also be able to setup the server to be connected to from anywhere. It will have a simple security setup that will prevent outsiders from taking control of the home. The server will also allow real time information on the status of the home. The information will indicate whether the room is being occupied, what the temperature is in that room, and in the home. The server will calculate the average temperature by combining all the different temperatures in the home. It will have control of the furnace and A/C allowing occupants to turn it off while away from home, and turn it on before they arrive home to a warm/cool house.

Along with control to the different rooms one will be able to configure which outlets go to which room. One will be able to turn on any desired device while away from home. This will also allow occupants to turn on appliances like a coffee maker from their bed in the morning, or on their way home from work.