Smart Home Project Synopsis
GroupList

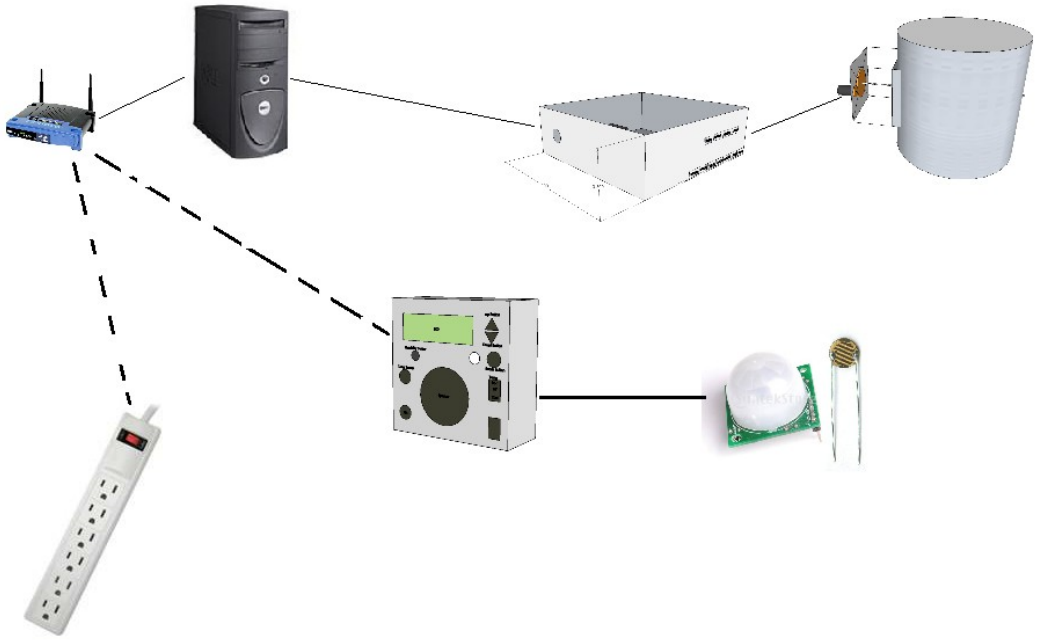
Levi Balling - leviballing@gmail.com

Dario Bosnjak - bdario1@comcast.net

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

Todd Rogers - u0634986@utah.edu

www.eng.utah.edu/~lballing/SmartHome



Introduction

A smart home will take advantage of its environment and allow seamless control whether you are present or away. With a home that has this advantage, you can know that you home is performing at its best in energy performance.

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 throughout the house with one 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 weather conditions are better outside than in the home. An example of this occurs during summer nights when it is 63 degrees outside [1], 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 (Figure 1). 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.

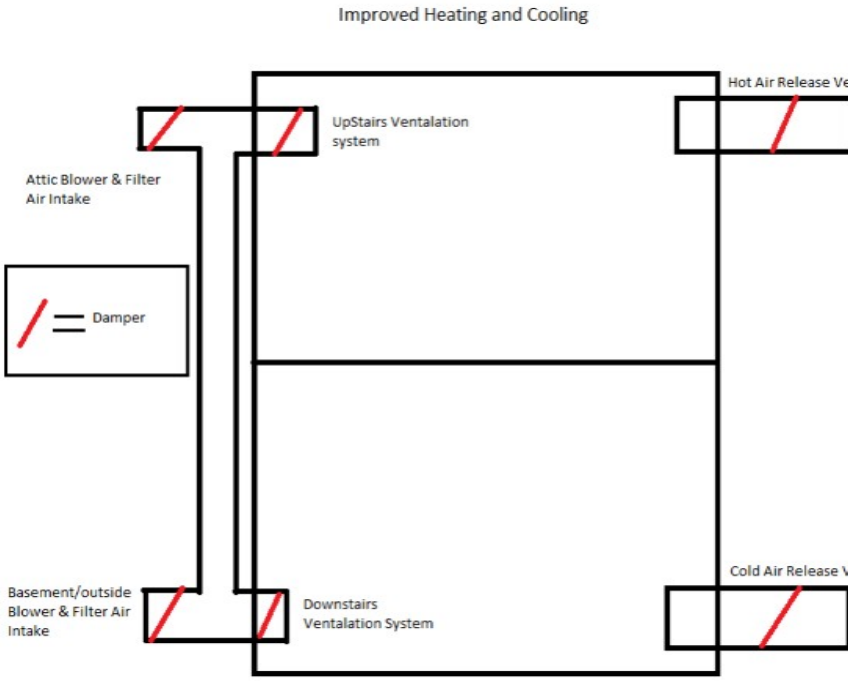


Figure 1

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.

Occupancy Detection

This system tracks the occupied or vacant status of each room in order to turn lights and appliances on and off automatically as well as manage the temperature in individual rooms without human intervention.

In the past, occupied status was determined using a motion sensor and a countdown timer. Whenever motion is detected, the room would be deemed occupied, and the countdown timer would begin counting down. If the timer reached zero without being reset by further motion, the room would be deemed unoccupied.

The problem with this strategy is that establishing the length of the countdown is problematic. If the duration of the countdown is too short and motion is not detected, the system may inaccurately adjust the room to vacancy status when it is still occupied.. If, however, the countdown is too long, the system may deem a room occupied long after the occupants have exited the room. Both of these issues undercut the efficiency of the system and could prove inconvenient.

The solution to this problem is to add a third component. This is a light beam interruption detector. The light beam interruption detector is positioned between the two posts of each doorway. The occupied status of the room will change when the light beam is interrupted. The light beam will detect, in 5-10 seconds, whether a room is occupied. When the light beam is interrupted and a room is determined by the system to be occupied, only detection by the motion sensor is required to maintain the room status until the light beam is interrupted again.

Materials

The light beam interruption detector can be created easily with a CdS photo-sensitive resistor and a laser pointer or other concentrated light source. This requires appropriate analog circuitry to produce a digital output. The motion sensor will be the DYP-ME003DD-H low voltage PIR motion detector module (Figure 8). It may need some circuitry also to create a digital signal with the correct voltage range.

Figure 8

WIFI

The WiFi interface for this system will be implemented using an Arduino microcontroller. This microcontroller will allow us to gather the data from different sensors throughout the house and broadcast the data to a server. This server will use a SiGi interface. A Wi-Fi shield for Arduino will be connected to the server using the TCP/IP protocol with an assigned IP address which the server will use to process to the information on the IP address. We are using the Wi-Fi Shield, WiShield V2.0, which will

have direct plug and play with the Arduino board. This will allow us to use the existing pins on the board. Because the Arduino Wi-Fi shield comes without an external antenna, we will add this for improved range.

We will run an Apache server where we can store the data that comes in from the microcontrollers. This data will be formatted and displayed on a secure website which is accessible from any internet connection (Figure 2). An extra function of the Arudino Wi-Fi shield is its capacity to host a less-capable web server directly on the chip. However, this is less user-friendly as its display content is limited. Using the Wi-Fi shield we will send and receive data. Once the microcontroller receives the input from the temperature sensor, it will send the information to the server. The server calculates and initiates the required system changes based on the information received. Below is a diagram to illustrate the Wi-Fi operations in the house.

Figure 2[1]

Room Controller box

Each room will have a single controller box. This controller box will contain a MCU that will connect to and control the devices of the system. It will powered by a 12V, 1.5 amp power supply. The box will have multiple temperature sensors. One of the sensors will be on the box, one will be on the motion sensor board and one will be on the fan. This gives temperature readings from the air near the ceiling, four feet from the ground and in the vent.

Server

The central server is the hub of all activity. It performs two important functions. First, it takes input information from the sensors, and executes scripting commands in response to this information. These scripting commands control the various system devises. Second, it creates a web-based interface for interacting with the home's various devises.

We have chosen Python as the language with which we will implement the core server software. Python is the core language of this program. It will be used in conjunction with other programming languages.

Outlet Controller

The outlet controller will be in the form of a power strip (Figure 3). It will be mobile and reconfigurable. This power strip will plug into any standard wall socket, and be wireless 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 the home owner to control power to any device via the internet. The power strip will also monitor power for devices which are plugged in. This data will be available at the controller boxes and server computer.

Figure 3

Implementation

Temperature Control

The temperature control system will use a variety of components (Figure 4). Temperature sensors will be placed in every room, outside and the attic. We are using the LM335AZ temperature sensor [2]. The sensors will be calibrated together to allow for accurate temperature readings. The system will use dampers to control the air flow. These dampers will be built using stepper motors, gears, and tin. The dampers are designed to meet Utah building code. The system will have a furnace blower and small fans for the

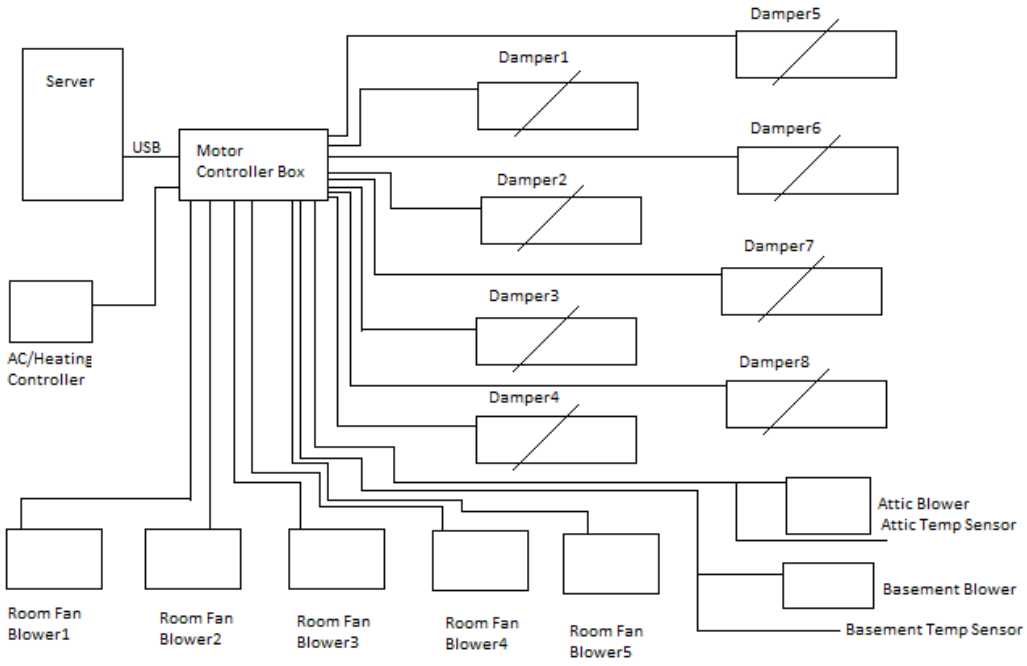


Figure 4

individual room blowers. These will be controlled 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 vents. The vents will be controlled by the motor controller box. The purpose of the individual room fans is to reduce the air pressure in the ventilation system. Because of the air pressure reduction, more air from the furnace blower will reach the rooms furthest from the furnace.

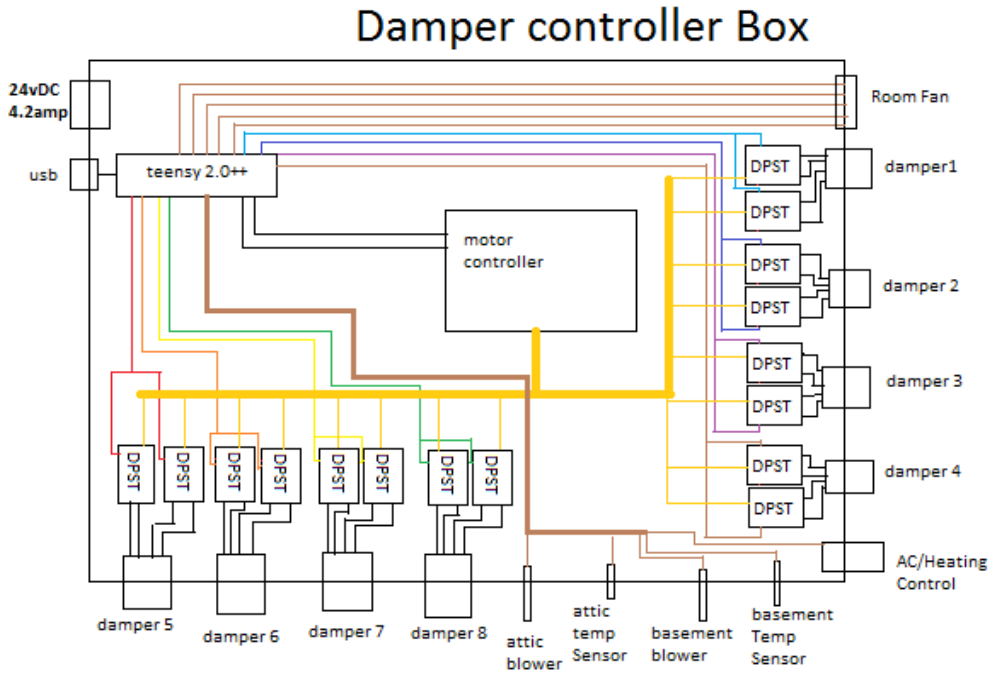


Figure 5

The motor controller box will interface with five fans, two blowers, eight dampers, the existing AC and heating, two temperature sensors, and a server (Figure 5). It will communicate over wire, which will also provide lines for ample power to each device. The motor controller box will require 24 V and 2 amps to drive the motors and other circuits. It will be powered by a 24V DC, 4.2 amp power supply. This will drive all of the fans, relays and other components. It will communicate over a teensy 2.0++ USB. This microcontroller will control all of the signals to all of the devices.

The motor controller is a bipolar, directional, stepper motor driver. It will step the motors in either direction. It will supply up to 1 amp on each coil of the 4 wire stepper

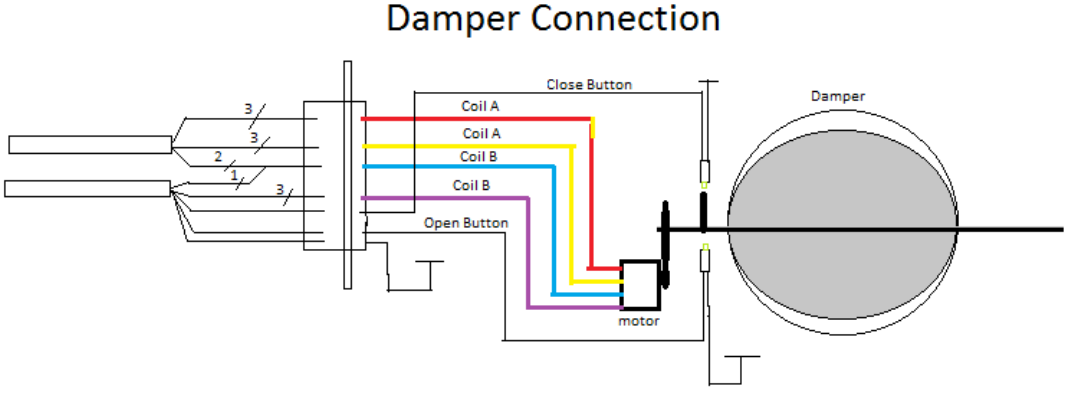


Figure 6

of 0.1(best case) when the dampers are open. This provides a range of 980 CFM - 1585 CFM.

The room blowers will help the air through the ducts if it is not flowing well. (Figure 7) Each room blower has 148 CFM without resistance. These blowers will be especially useful in rooms located far from the furnace. The dampers will control how much air goes into specific rooms.

In a home with 11 heat registers there is ~100 CFM going into each room. With a 980 CFM blower in the attic and the basement there will be enough air flow to circulate through the whole house.

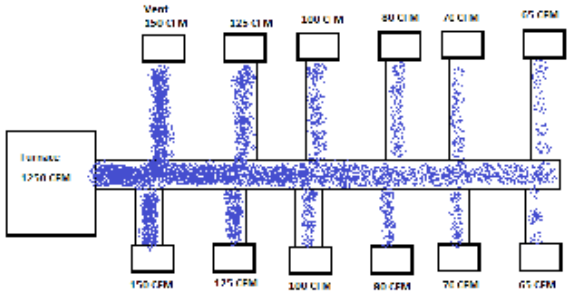


Figure 7

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 obtain the smaller fan from bold computers. Both types of fans will be powered with 12V. The relay that will power these fans runs on 5VDC.

Interface specifications for motion detector

Supply current	DC 0.8V--9V
Current drain	<150uA (Voltage different, current different）
Voltage Output	High/Low level signal:3.3V TTL output
Detection distance	3--7M
Detection range	<140°
Delay time	(Default 10S +-3%) Can make other time
Blockade time	2.5 S(default) Can make other time
Trigger	L:Non-repeatable trigger H:Repeat Trigger (default)
Turn off during daylight	Can be added as customer requirement
Compact size	28mm*28mm

*CdS photo-sensitive resistor specifications to be determined experimentally.

WIFI

The power usage of the Wi-Fi Shield with Arduino is low. It requires 5-7 volts. The voltage regulator will provide enough power for it to run. The Wi-Fi will be 802.11b at 1Mbps and 2Mbps throughput speeds.

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

The Wi-Fi shield has many capabilities. It can be used to establish secure and insecure networks such as WEP (64-bit and 128 bit) and WPA/WEP (TKIP and AES) PSK security. The chip uses SPI for host communication. (Max speed 25MHz) It has a feature for an on-board PCB antenna. The CS pin is switchable for serial flash between digital pin 10 and digital pin 7. It will communicate on 802.11b with the Netgear router at about 1 Mbps. The Wi-Fi shield is shown in Figure 8 below.

Figure 9 [5]

Pin Usage

- SPI
  - Slave select (SS) : Arduino pin 10 (port B, pin 2)
  - Clock (SCK) : Arduino pin 13 (port B, pin 5)
  - Master in, slave out (MISO) : Arduino pin 12 (port B, pin 4)
  - Master out, slave in (MOSI) : Arduino pin 11 (port B, pin 3)
- Interrupt (Uses only one of the following, depending on jumper setting)
  - INT0 : Arduino pin 2 (port D, pin 2)
  - DIG8 : Arduino pin 8 (port B, pin 0)
- LED : Arduino pin 9 (port B, pin 1)
  - To regain use of this pin, remove the LED jumper cap
- 5V power
- GND

The implementation strategy involves establishing communication between the Arduino board and the Wi-Fi router. The web page will run on an Apache server with .jsp commands ready to send necessary instructions to the board. The data from the sensors will have to be captured by the pins on the Arduino and sent over Wi-Fi for the server to display the information for the user. The server will organize the information using HTML, CSS and Java Script (.jsp) languages.

Two rooms will be controlled via the Arduino with the Wi-Fi shield attached to it. Each shield will have a static IP address assigned to it for convenience. Each Arduino board will collect the necessary information from the sensors in the room and send the information back to the server. Each IP address will correspond to one room. The server

will match the information from the IP address to the room from which it came. The Arduino board will collect the data from the sensors and send it over the SPI interface. The Wi-Fi Shield will, then, transmit the date to the server. The following is an example of the input data which will be received by the server:

192.168.1.101/=132

That data represents the following:: The IP address,"192.168.1.101" is from room #1. The "=" corresponds to data being sent to the server. "1" is the sensor number. "32" is the temperature in Fahrenheit.

The follwing is an example data received by the devise:

http://192.168.1.101/?3.21

The IP address "192.168.1.101" indicates the device in room 1. The Arduino board receives the data that follows the '?". The "3", following the "?", corresponds to the power strip. The "2" corresponds to a specific outlet on the power strip. The "1" shows the status of the outlet-- 1 if it's on, 0 if it's off.

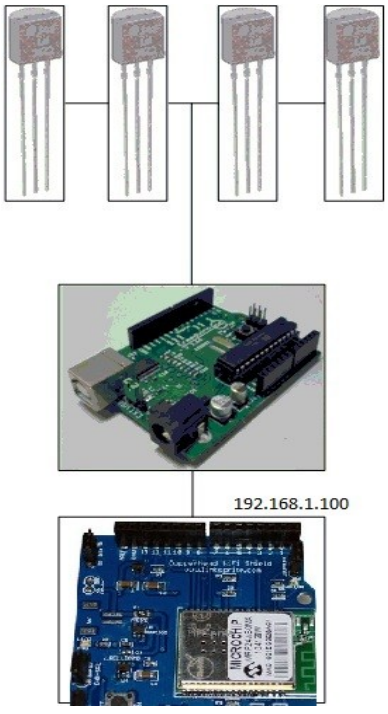


Figure 10

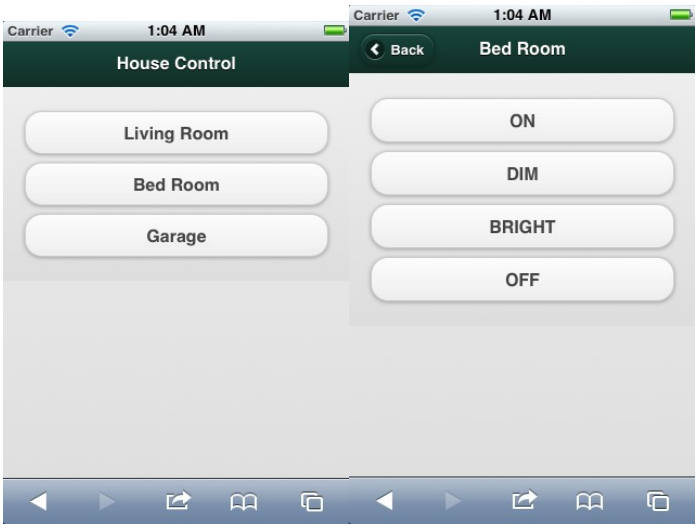


Figure 11: The user communication interface

Outlet Controller

The module will provide power to the MCU, WIFI chip, power sensors, LEDs, and the outlet. A 120VAC to 12VDC power supply will be used to provide power to the relays. The power supply 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 may need to be combined with a low amperage sensor to provide increased accuracy with lower currents. These sensors will interface with the on-board ADCs on the Arduino board. The information from the current sensor will be displayed using a system of on-board LEDs. 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. These 12V coils will be switched on and off via a transistor controlled by the 5V logic on the Arduino. An optional function of the power strip is a surge protector. This

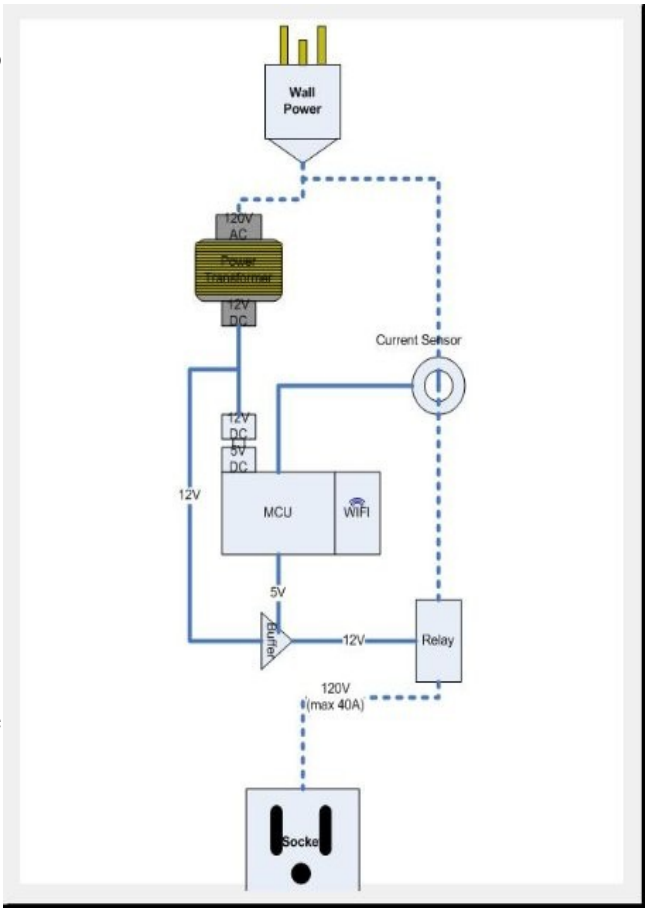


Figure 12

surge protector will offer a set/reset function both manually from the power strip and on the server control system.

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 can be installed in any wall, and meet building code, due to its 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 fan speed is controlled by the duty cycle. The fans require 12VDC power. This will supply the fans with 1.00 amp of current. The controller will also have a LM335AZ temperature sensor that provides an output of 10mV/K.

Web Interface

The web interface will control all Smart House functions, statistics, and controls. This interface will be accessible to the Smart House users on PCs, smart phones, and any other web-displaying devices. The main control systems will be grouped into outlet, heating, and occupancy sections. Statistics will be available for all the different systems. Figure 13 is a simple mockup of the web interface with displayed outlet control.

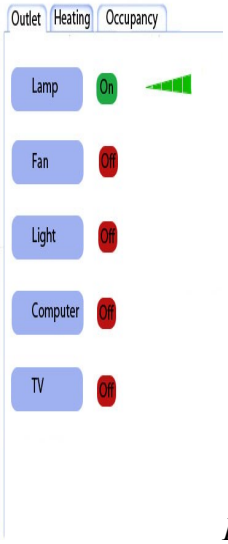


Figure 13

Interfacing

The information from each room will be sent to the server over a router. Using a PC as the server will facilitate programming. The server will be accessible from any internet connection but will have safeguards to prevent hacking. The server will also allow real time information on the status of the home. This information includes temperature readings and averages, room occupancy, and power usage in the home. The server will control the furnace, the AC and the dampers of each room. The server can also control the power supply to any device that is plugged in to the power strip.

Time Line

Development of this project will begin with occupancy, heating, outlet, and WIFI systems. These will be developed in parallel. After all the separate systems function, we will integrate them into the server control. Finally, the web interface will be created after the server developments are underway.

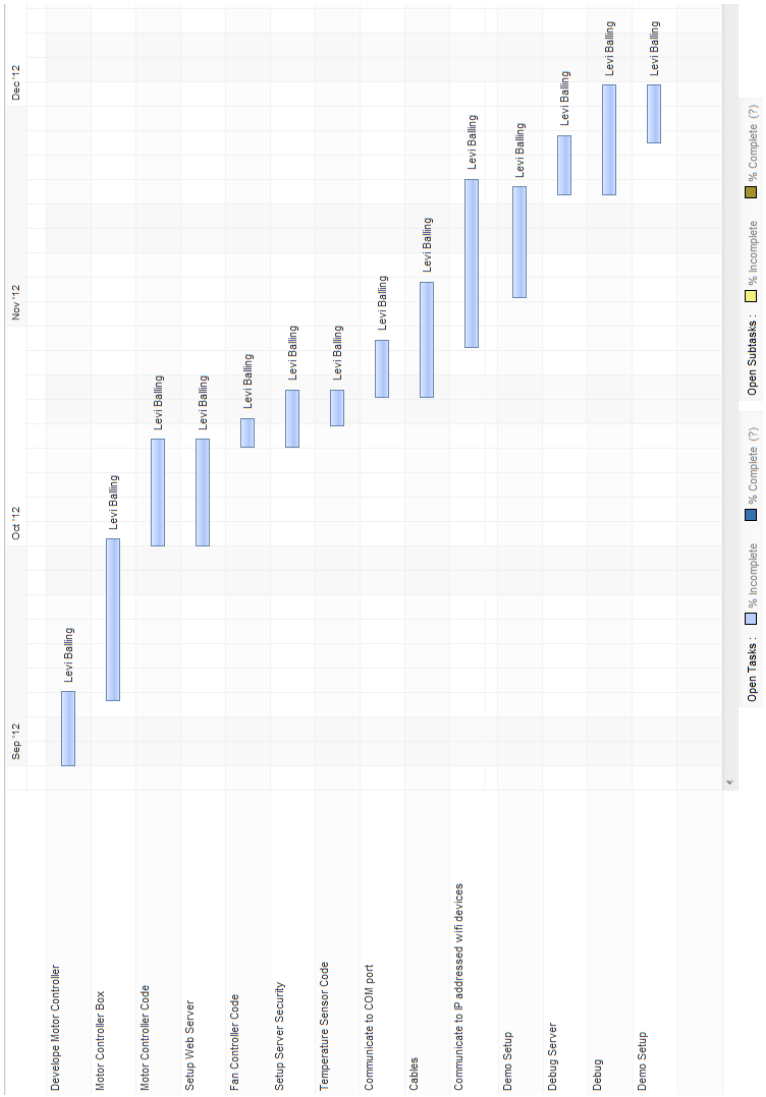
Demo Setup

We will demonstrate the heating, occupancy, and outlet control in the following ways. First, the heating system will be demonstrated using a small-scale setup. We will

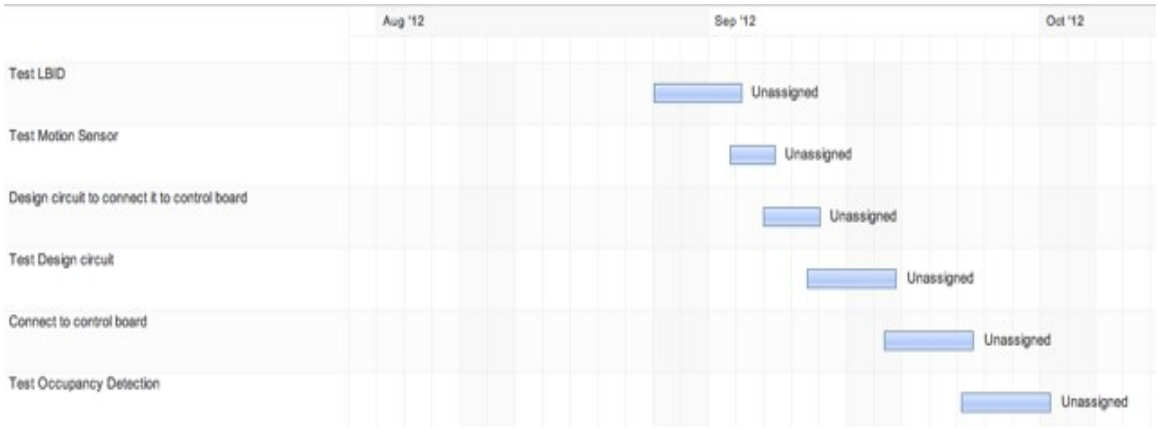
use a greenhouse with small-scale versions of the fans and dampers to operate the heating system. Users will be able to change the temperature settings, and also provide cool air or heat to the system, and watch how the system adjusts properly. Second, we will be using the small VLSI testing room in the senior hardware lab for the occupancy detection system. As users enter and leave the room the occupancy sensors detect their movement, control the lights and record occupancy statistics. These statistics are available on the web interface. Third, we will use the outlet to control the power going to the other parts of the demo. Users will control several devices plugged into the power strip and see the devices' power statistics via the web interface.

Tasks Schedule

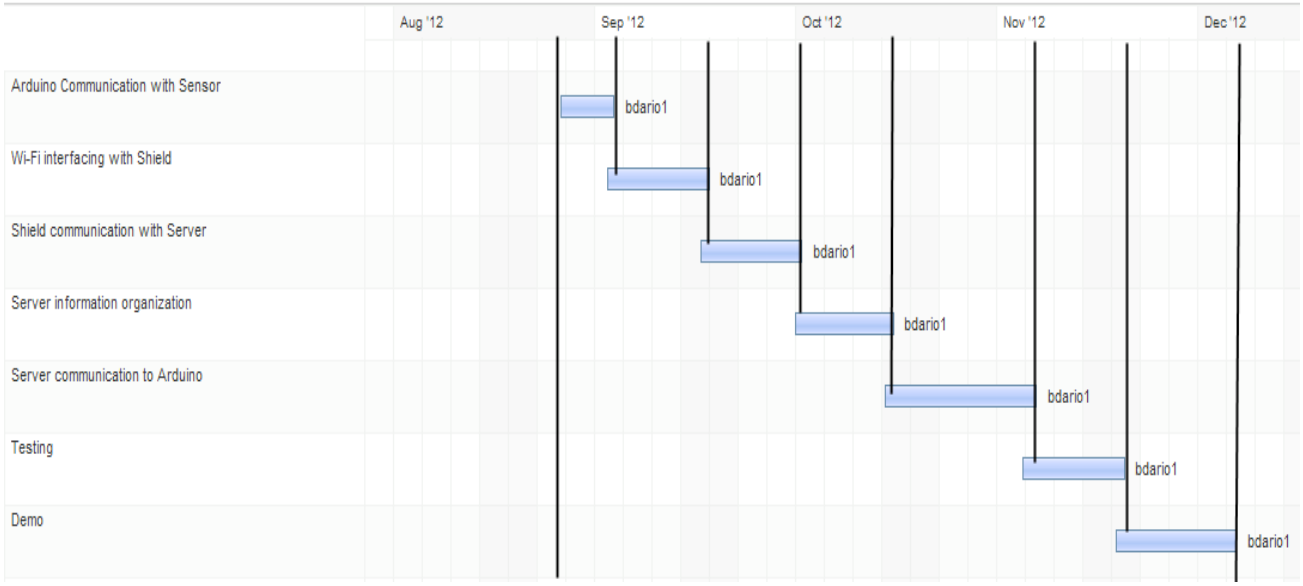
Temperature Controller – Levi



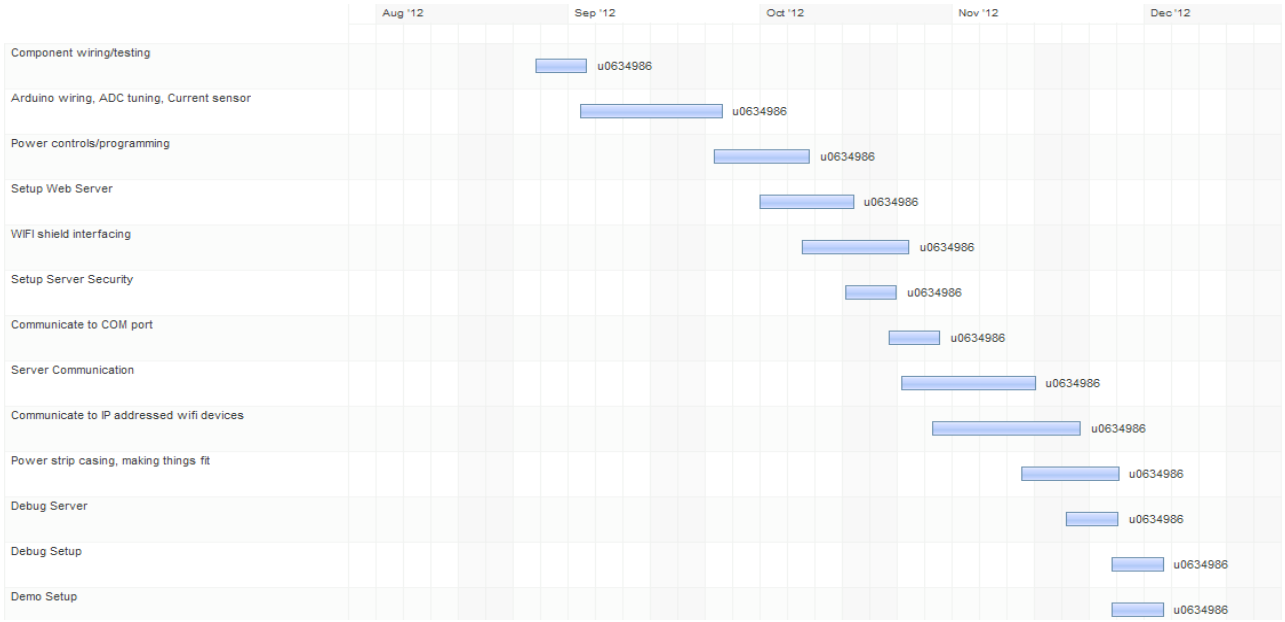
Occupancy detection - Christopher
Schedule



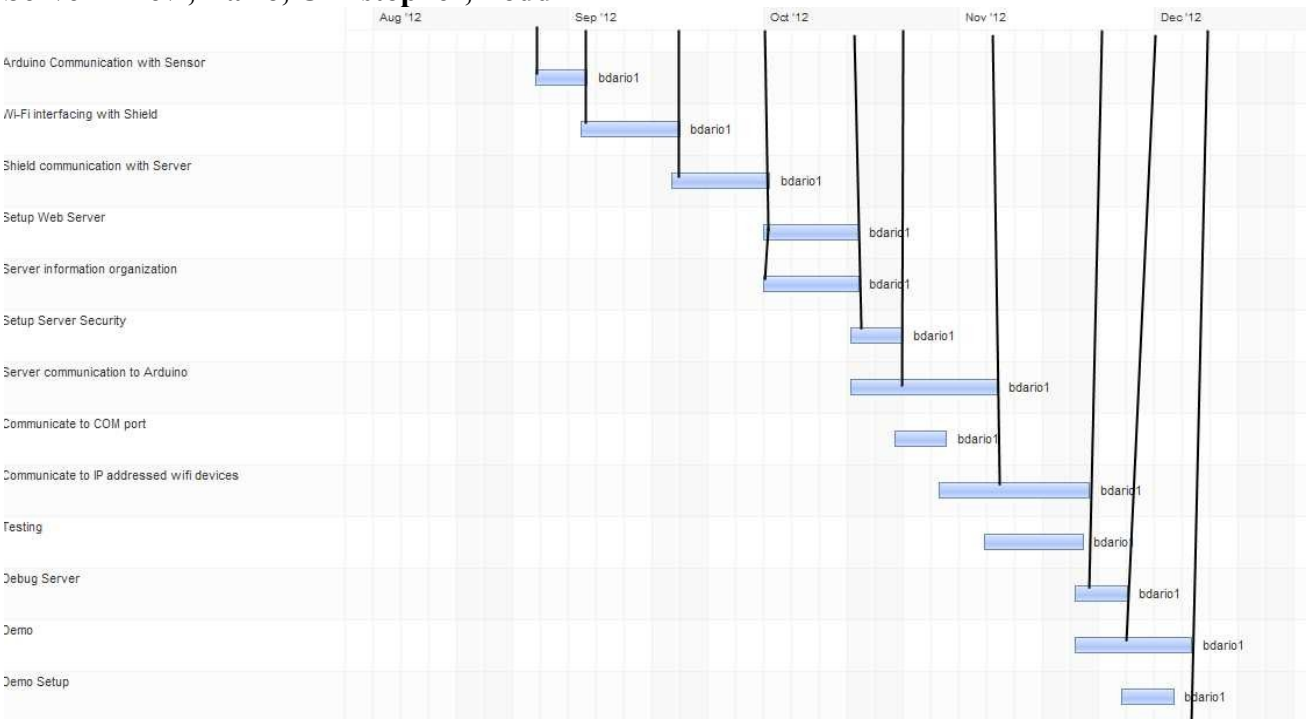
WIFI – Dario Bosnjak



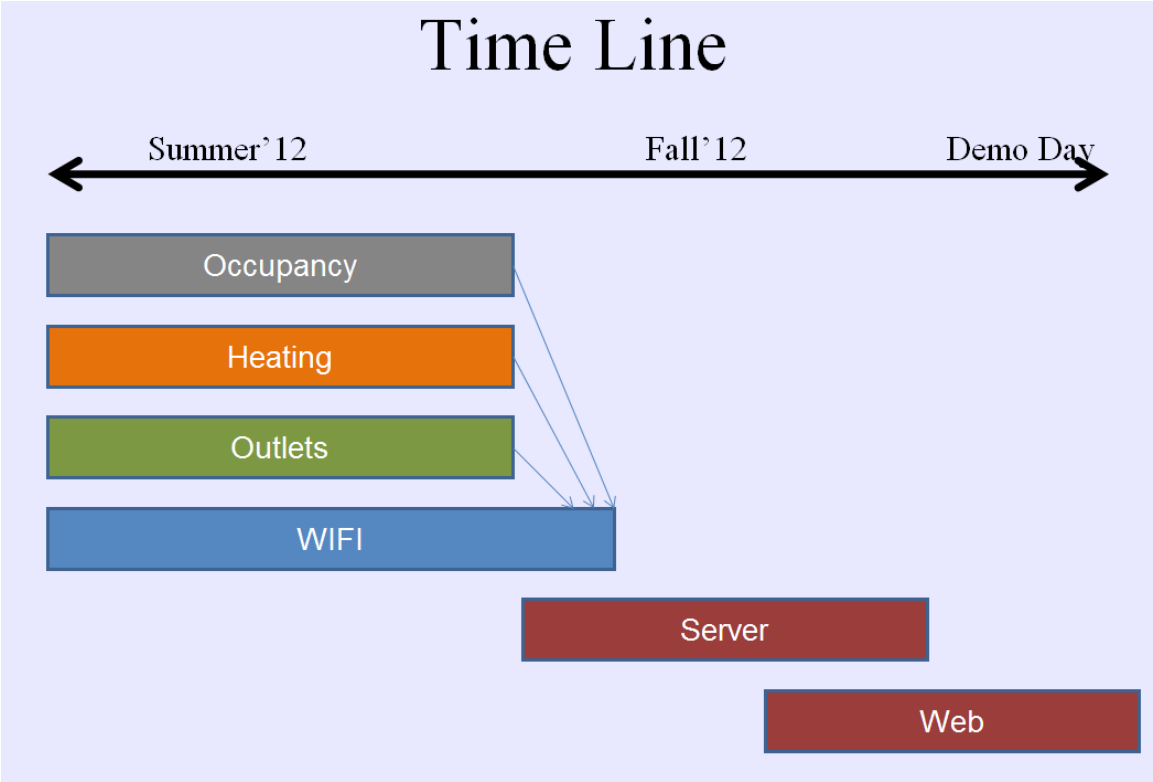
Outlet Controller - Todd



Server – Levi, Dario, Christopher, Todd



Overall Time Line



Bill of Materials

Temperature Controller

Motor Controller

Qty	Vendor	Part number
1	National Semiconductor	555 timer
1	OctoPart	MC74HC194N(4 bit bi-directional Shift Reg)
1	National Semiconductor	SN754410NE(Quad Half-H Bridge)
1	National Semiconductor	LM7805(5V voltage regulator)
6	National Semiconductor	2N3904 or 2N4400
1	Radio Shack	LED
1	N/A	1N4148 Diode
1	N/A	1N4001 Diode
5	N/A	3.3Kohm
2	N/A	470ohm
7	N/A	10Kohm
1	N/A	1uF
2	N/A	4.7uF
1	N/A	470uF

Damper

Qty	Vendor	Part number
8	All Electronics	Mineba NMB-MAT # PM42L-048-UAJ9 Stepper Motor
1	Homedepot	Thick sheet of plastic -Custom laser cut gears
8	Homedepot	Vent Dampers
8	Home depot	5/16x1” Coupler
1	Homedepot	5/16x4' Screw pole
8	All Electronics	9 Dsub connectors
8	All Electronics	9 Dsub Backshells
16	All Electronics	SPST button Switches
2	Homedepot	6'x Heating vent pipe

Motor Controller Box

Qty	Vendor	Part number
18	All Electronics	DB9 Female board mount
16	Digikey	5VDC Relay
16	Senior hardware lab	2N3904
1	PJRC	Teensy 2.0++
2	Digikey	100 mil sockets
16	Ece parts room	Resistors

Room Fan Controller

Qty	Vendor	Part number
5	Jameco	AFB1212SHE brushless fan
5	All Electronics	DB9 female

Basement/Attic Blower Controller

Qty	Vendor	Part number
2	Google	Furnace Blowers
2	All Electronics	A/C 120V Socket
2	A/C	A/C 120V Plug
2	DSPT	120VAC Relay

Room Sensor

Item	Vendor	Contact person	Phone
CdS photo-sensitive resister	already in possession	N/A	N/A

laser	already in possession	N/A	N/A
DYP-ME003DD-H low voltage PIR motion detector module	http://www.suntekstore.com/	could not be reached	+86-13725596257 dial 011 first (number appears bad)

WIFI

Qty	Part number
1 WiFi shield for Arduino	WiFi Shield WiShield V2.0
1 Arduino Duemilanove	Digikey,Arduino and Jameco
1 WIFI Router	Netgear wifi router

Outlet Controller

Qty	Part number
5 Relays	CB1AHF-12VRELAY AUTOMOTIVE SPST 70A 12V, Panasonic
5 Current Sensors	ACS709LLFTR-35BB-TSENSOR CURRENT 75A 5V BI 24QSOP, Allegro Microsystems Inc
5 Transistors	2N3904TFTRANSISTOR NPN 40V 200MA TO-92-Fairchild Semiconductor
1 Power Suppl	VOF-25-12PWR SUPPLY 24W OPEN 12V 2.0AV-Infinity VOF-25CUI Inc
2 Voltage Regulators	LM78L05ACZXAIC REGULATOR 5V 0.1A 5% TO-92-Fairchild Semiconductor
1 Arduino Demilanove Board	
1 Wifi Shield	
1 Power supply	

Server

Qty	Part number
1	Computer
1	Wireless Router

References

[1]Average Temperatures www.noaa.gov
[2] National Semiconductor November 2000

[1] "Diagram Wi-Fi communication setup" *http://www.jfkreuter.com/?p=9, http://asynclabs.com/store?page=shop.product_details&product_id=26&vmcchk=1, http://httpd.apache.org/,http://home.cisco.com/en-us/home*

*[2]"*Arduino image" *http://asynclabs.com/store?page=shop.product_details&product_id=26&vmcchk=1*

[3] "Arduino Server configuration" http://Arduino.cc/en/Tutorial/WebServer
[5] "Wi-Fi shield setup"- http://www.jfkreuter.com/?p=9
[4] "Website images" http://thenewtech.tv/community/Arduinox10-home-automation-over-wifi

Appendix

Motor Controller Box Commands

Text	Cmd/Sts	Action
Damper###	Cmd	Turn on and off Damper ex: Damper051 = Damper 5 close. Damper020 = Damper 2 open
Damper?##	Sts#	Replies whether damper is open or closed, ex: Damper?04 => returns 1 if closed 0 if open
Attic#	Cmd	Turns on/off the Attic blower, ex: Attic1 = Attic blower on
Attic?	Sts#	Returns whether on or off, ex: Attic? = 1 if on or 0 if off
AtticTemp$?	Sts###	Returns temperature in the attic, ex: AtticTempc => returns 023
Base#	Cmd	Turns on/off the Basement Blower, ex: Base0 = Basement blower off
Base?	Sts#	Returns whether on or off, ex: Base? = 1 if on or 0 if off
BaseTemp$?	Sts###	Returns temperature in basement near blower, ex: BaseTempk => returns 290
Fan####	Cmd	Turns on/off fan from 0 to 100 percent, ex: Fan4085 = fan 4 to 85 percent.
HVAC?	Sts#	Returns HVAC status: 1: blower on 2: blower on & heating 3: blower on & cooling

		4: server controlled blower on 5: Server controlled blower on & heating 6: Server controlled blower on & cooling 7: blower off 8: Server controlled blower off
HVAC#	Cmd	Sets the HVAC into on of the following states 1: thermostat controlled 2: Server controlled blower on 3: Server controlled blower on & heating 4: Server controlled blower on & cooling 5: Server controlled off
Possible features		
Garage	Cmd	Garage open/closed command, toggles the garage door to open or close
Garage?	Sts#	Returns whether garage is open(1) or closed(0) ex: Garage? => returns 1 if open
Sprinkler###	Cmd	Turns on/off section of sprinklers, ex: Sprinkler102 = turns on section 2 of sprinklers
Sprinker?	Sts##	
Symbol Table ? = Status request # = number; 0-9 $ = Temperature Format; c, f, k;		