HVAC Smart Control

Steven Jones, Jerthwin Prospere, Elroy Ashtian, and Matthew Arcuri

School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The HVAC Smart Control is a feedback and control system that makes economically efficient control decisions for heating and cooling of buildings. It does so by processing temperature and humidity data from indoor and outdoor sensing units, in addition to CO2 readings indoors, to intelligently and economically maintain desired comfort and air quality within buildings the system is implemented. The system contains three units: an outdoor sensing unit, a main control sensing unit, and the display unit. It is also designed such that additional units and features can be added easily in the future. The outdoor sensing unit takes outside temperature and humidity readings and transmits that data either wirelessly over 802.15 RF or - should interference prove to be problematic - wired over RJ-45 cable to the main control unit. The main control unit receives both inside temperature, humidity and CO2 readings, as well as the outside temperature and humidity readings, and sends that data to the display unit. The low power display unit gathers the parameter data and determines the most economical way to run the HVAC system. In addition to the readings, the main control unit takes into account the types of HVAC equipment installed, such as a dehumidifier or an additional AC unit, customizing the efficiency computation for the particular hardware installed at the premises. It then sends control signals back to the main control unit that then turns on and off relays that control HVAC system components. Furthermore the display unit acts as a gateway to the internet for secure remote access should the user realize the air conditioning was left set at a less economical setting with no occupants, or if they simply wish to view the current readings in the building. The display unit also features a seven inch touch screen to make interactions and programming easier and more intuitive, eliminating a common reason why many households do not use the programming features of their thermostat.**

***Index Terms* — HVAC, control and feedback, LCD touch screen, CO2 sensor, temperature sensor, humidity sensor, microcontroller, RJ-45 line driver, RS-232 line driver.**

I. INTRODUCTION

HVAC, short for Heating, Ventilation, and Air Conditioning System are designed to maintain desirable air temperature and quality levels inside of a building. Throughout the extremities of the year heating and cooling is responsible for the majority of power consumption in many buildings. With the recent surge in energy cost and increased environmental concerns, automating and improving the controlling of HVAC systems in buildings and homes takes the switch from luxury to necessity.

Our project is set to design a more energy efficient HVAC system that consumes less energy while providing better user interfacing to make it easier to program. In its current design it is responsible for the monitoring and control of temperature and humidity and monitoring air quality via CO2 readings in a building. This system should be able to do all responsibilities of a traditional HVAC system, as well as being installed within the traditional frameworks of legacy systems. The additions to the unit will allow the user to select from a list of power settings, ranging from maximum comfort to maximum savings. These settings will allow the user to control the tolerance levels within the HVAC system, such as level of time the unit stay on, how far beyond the limits the system will let the variances become before taking action, as well as which unit to choose. To further save money we have designed the system to be able to control a dehumidifier, outside ventilation, additional smaller AC units, air filtering, dampers for zones, disperse a scent into the air, and determine when such systems should be turned on based on inside and outside conditions, scheduling, or on demand manually via the user interface by the user. In total currently seven such devices can be controlled by the system via relays. The flexibility of additional devices being introduced to the system allows the user to start off with increased savings via automation and programming with existing systems however also allows them in the future to add such devices for even more savings. With added devices such as the dehumidifier, this should encourage the usage of a higher temperature setting as the temperature indoors will “feel” cooler due to a decrease in humidity rather than an actual decrease in temperature. In addition the system will provide the ability to use more desirable outside air if it is applicable.

The majority of the power saving functionality of the HVAC system - besides making the system more intuitive for users, especially those in non-commercial buildings, to program scheduling - is designed around splitting the traditional HVAC cooling system from one AC unit into two or more units and use less energy intensive systems before ever turning on the AC unit. Depending on power settings the system will use a smaller unit than traditionally larger unit used to maintain temperature levels within the tolerance acceptable by the user. When in maximum saving mode, for instance, if the temperature deviates beyond the tolerance level, it will first check the outdoor temperatures. If the temperature outside is less than inside then it will vent the outside air indoors. The economic benefit in this situation to the user with this system is huge.

It avoids using the high energy consumption of the AC unit compressor, while allowing much greater cooling then simply opening one's windows. This is especially if the temperature difference between outside and inside temperatures is minimal. The "laziness" of the user is also taken out of the equation as one would then have to go opening and shutting numerous windows in the house at the right time. Finally it is also is advantageous at night when one might be concerned about security issues when opening windows on ground levels. The system will be able to determine if the outside temperature isn't sufficient to adequately cool the building. When that threshold is met, it will turn on the smaller of the AC units and will wait a time determined by the max savings tolerance and check to see if the temperature settings are adjusting in the proper manner. Energy savings can be clearly quantified as in comparison to the larger, more traditional, sizes that would be used to adequately control temperature. If there aren't adequate changes made to the indoor environment levels in comparison to user settings and tolerance then the first choice in Air Conditioner was too small, and the system will adjust by turning off the first Air Conditioner and on the larger of the two and running through the logic again. This will continue until the maximum amount of units is used which would match that of the traditional system; meaning at worse the system will spend equally with a traditional system.

The comfort setting tolerances are adjustable on each level in administration controls. The adjustable tolerances are temperature quantities exceeding user desires and the times that these levels should be adjusted. These levels can be set within the bounds of the power saving level adjacent to it. These levels are a key feature in the amount of money that can be saved per comfort setting.

It should be noted, that while the purpose of this project is to save energy consumption, there will be a max comfort setting. This setting spares no expense in the attempt to maintain desirable user little environmental conditions. This assures the user the ability to keep comfort settings as on par with traditional comfort known in traditional HVAC systems.

In addition to common HVAC system controls of temperature the addition of humidity sensor will allow the user to evaluate and provide a parameter of controlling humidity through an optionally of a dehumidifier. Also the addition of the CO2 sensor will allow the system to help the user control air quality levels. Measured in parts per million (PPM), a level over 1,000 PPM CO2 is considered to be unhealthy. The user will be able to manually configure the way the system vents. The unit will vent in some outside air or simply circulate the air to alleviate levels higher than healthy CO2 level in the building. The user will be also able to control manually the ability to vent the building for a specified period of time in situations such as the burning of food, good ventilation for safety reasons when using strong chemicals for cleaning, or for various other reasons.

Though energy savings is the primary goal of the unit, increased user interface capabilities will be a stronger goal. While increasing the ease of use of the unit isn't explicitly an energy saving feature, we hope the improvement to its ease of use will in turn increase the use of energy saving features currently underused on thermostats due to the complexity of programming, such as scheduling times in which the unit should be at a certain temperature threshold. The foundation of the user interface is the seven inch touch screen. Touchscreen phones and tablets are becoming much more common for use and having such a device to use for the user interface advances the ability of the user to interact with the device and have a positive experience when doing so. The display is large enough to display enough information to help the user make decisions easily, however not too large to increase the cost of the overall unit too much to the point where less people will be able to use it. The system will be run on the Linux operating system that will be on an ARM TI-OMAP microcontroller. The use of the ARM processor helps provide much less energy consumption than most traditional x86 computers and while not as computationally powerful as its x86 counterparts, for our purposes fits our design needs in terms of cost and computational needs better at this time. Having a separate unit, different than the microcontroller allows us to have a device dedicated to running the program on a modern operating system will allow using a high level object oriented programming language like JAVA instead of traditional microcontroller language C, giving a much greater flexibility to additional features, the appearance, and providing a more stream lined and visually appeasing user experience. Another additional benefit of the advanced operating system is the ease of internet connectivity and the ability to run various webservers, in this case a webserver for remote access.

Internet connectivity provides numerous possibilities however we have decided on three essential features to implement as the baseline for our design. First, the ability for the user to remotely access the data and control the HVAC system will be added. This will be done through a web page. The temperature and humidity data will be saved on file of the display system, and this data be displayed on the website, with frequent periodic updates of this data. Secondly, the user desired data will be streamed and they will have the ability to remotely access and change desired temperature and humidity controls.

Finally, the internet connectivity will provide administrator or technician controls. If there is an issue with the HVAC system instead of a technician having to schedule an appointment and physically access the unit, the ability to remotely control and evaluate the HVAC system would be possible. This will also allow technicians the ability to remotely upgrade the software of the display and control system.

An additional portion of this project will be the introduction of mood scents. The user will be able to control the scent of selected rooms through the user interface. There will be relays in the main control unit in order to control a scent dispersion unit. This will be based upon user defined times.

II. Components

A. Sensors

This part of our project is vital to the functionality of our device and is the reason why it is discussed first. There are three points of data that will be monitored in two separate places. Temperature, humidity, and CO2 data will need to be monitored inside the building, and temperature and humidity will need to be monitored remotely.

Having temperature and humidity measured on one sensor unit provided us with a more economical and easier interfacing option. We chose the SHT21 temperature and humidity sensor. This sensor as seen below has a built in A/D convertor and provides humidity levels from 0-100% and temperature levels from -40 to 257 °F. This is well within the parameters of real world situations. The sensor will interface with the microcontrollers through I2C protocol.

Fig 1 Temp/ Humidity Sensor

The CO2 sensor chosen was the SenseAir K-30. It provides multiple ways of interfacing with microcontrollers. There are possible analog outputs to microcontroller that can be evaluated and converted digitally through the microcontroller’s analog to digital convertor, but this requires additional connections to the microcontroller to turn on the A/D ports. There is also UART connectivity. Unfortunately all the UART ports on the microcontroller were accounted for, so unless there was a connection through a MUX to select devices to connect with UART, this wasn’t an optimal possible connection. There are also ports for I2C connectivity to this sensor which ended up being the protocol we used to interface it to our MCU. The CO2 sensor in fig 2 provides tolerances of 0 to 5000 ppm with an accuracy of +/- 20 ppm.

Fig 2 CO2 Sensor

B. Microcontrollers

This project will require the use of two separate microcontrollers, one for the outside unit and one for the main unit. For a more streamlined coding process, we’ve decided that the use of similar microcontrollers would provide a chose the PIC24F04KA201, but this proved to have several issues with the I2C interfacing by some silicon errata on chip. We decided soon to change our chip to the PIC24FJ128GA006. Already having the development board with this chip proved to be advantageous in our design. This chip has 64 pins, which could provide increased options of expansion later and has sister chip PIC24FJ128GA010 which has 100 pins which is more than adequate for our main control unit.

The microcontroller has 2 UART ports, 1 for RS-232 communication with the ARM display unit, and one for the connectivity to the XBee chip (802.15) which will allow us to have wireless communication between the main and remote unit. The microcontroller also has 2 I2C ports, providing enough for the CO2 and temperature/ humidity sensors. It also supported the SPI protocol but there isn’t any use for it.

The PIC24F microcontrollers are low power consuming units with lots of information and support available. Due to the features of the PIC24F, we were able to massage the component to our desire such as power saving modes, interrupts, watchdog timer and resets. For example, since we are powering the remote unit by batteries, it would waste power to have the microcontroller continuously run even when it is not gathering data. Therefore, we have added the feature of having the watchdog timeout into sleep mode for a certain amount of time that there isn't any measurement in order to save on our power consumption. This is important because it will add to the convenience of the user to not have to change the batteries very often. This of course is not very much of an issue on the main unit because it is being powered through the building's power supply and also because there is continual activity.

C. Communications

The 3 separate devices all have to communicate with each other in order to make this a unified system.

The outdoor system has to send the data of the outdoor humidity and temperature to the main control system. The preferred method will be through wireless communication. We chose the low power XBee. It transmits using 802.15.4 IEEE standard. It's a low wattage communication with sleep enable. Both microcontrollers communicate with the XBee through their UARTS ports. Additional connectivity from the secondary unit to the main unit needed to be provided in the chance that wireless connectivity was unavailable.

If wireless connectivity wasn't available, the ability for wired connectivity had to be available for a certain distance. A twisted pair RJ-45 is an adequate means of wired digital data communications. The TI SN65176 was chosen to drive the twisted pair communications. This driver is a two way driver, both sending and receiving data, so it could be used on both the main and outside control units. The driver automatically inverts the transmit and receive lines across 4 wires to provide the twisted pair with its increased data transmission length.

For communication of the main control unit with the ARM unit, RS-232 was chosen. This is a commonly used communication protocol that was pre-installed on our ARM unit and allowed us to use known drivers with JAVA on the display unit to communicate with the main control unit. The RS-232 driver chosen was the MAX232SOIC16.

Internet communications will be done with the on-board 802.11 wireless module of our ARM display unit.

D. Relays

This unit will also contain the control relays used for an HVAC system. Traditional HVAC relays use a 24VAC signal to turn on the fan, ac, and heater units. In order to keep compatibility with legacy systems, the unit was designed to use the same 24VAC signal. To expand the functionality of the system, additional control relays were used, totaling 14 in this system: AC1, AC2, Fan, Heat1, Heat2, Natural Air, Natural Air Dilute, Filter, Dehumidifier, Mood Scent 1, Mood Scent 2, Zone, Other 1 and Other 2.

These relays will provide the 24VAC signal used to power all the devices. The control of the relay will be tied to the main microcontroller, with the switch powered by a 3VDC signal. A PNP transistor will provide protection from switching bounce back current between the relay and the microcontrollers.

E. Power System

The entire main control unit and display system is powered using the 24VAC input. Our power supply is designed to take a 24V RMS, or 34V input as its power in. This is because HVAC components are designed around a 24V RMS supply line. The components of the system use 12VDC, 5VDC, and 3.3VDC. The conversion was done using a full wave rectifier with 50V, 3A tolerance diodes. The display ARM unit requires power of 5V from 3 to 5 Amps. We used National Semiconductors LM2679 simple switcher regulator in order to not waste power through the step down process. There are dedicated regulators each for the 5VDC display unit, 12VDC LCD screen, 5VDC main controller units, 3.3VDC main controller units, and 3.3VDC Relays. The power was split up into multiple regulators to help limit the overall current going through .

III. Design and Implementation

A. Software

The HVAC Smart Control software framework was built utilizing the Java programming language to allow for object oriented processes and portability. The software application allows user to view relevant data of the environmental conditions for increased awareness to provide the highest quality of product interaction with the system. The critical part of the Java application was the ability to perform serial communication between the main control unit and high level control unit using RXTX to control the various components used by the system. RXTX is a Java library using a native implementation via Java Native Interface(JNI) to provide serial and parallel communication for the Java Development Toolkit.

The two system user groups using for the software application are customers and technicians. Both customers and technicians are allowed to view indoor/outdoor temperature, humidity, carbon dioxide(CO_2) levels and selected city weather forecast along with the available installed devices such as the Air Conditioning(AC) units, Dehumidifier(DH), external Fan unit (FAN), and Natural Air System(NA) Both users can also set the heating/cooling points at which the system will identify to create the most optimal environment to nearest to the user's desires. The user case diagram is pictured below.

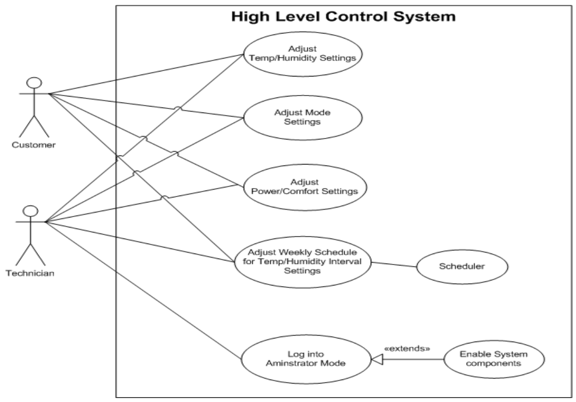


Fig 3 User Diagram

.

In addition to the previous stated features, users can also select between five levels of comfort settings ranging from Maximum Savings(Level 1) to Maximum Comfort(Level 5). Each of the levels retains its own temperature range and AC elapsed time for triggering additional components to turn ON or OFF. Additional selection options are available to the users for immediate system actions instead of allowing the system to determine the ideal settings. The Selection Menu buttons such as the Fresh Air, Air Quality, Times/Zones, Power Costs, and Mood Scents allows this to be possible.

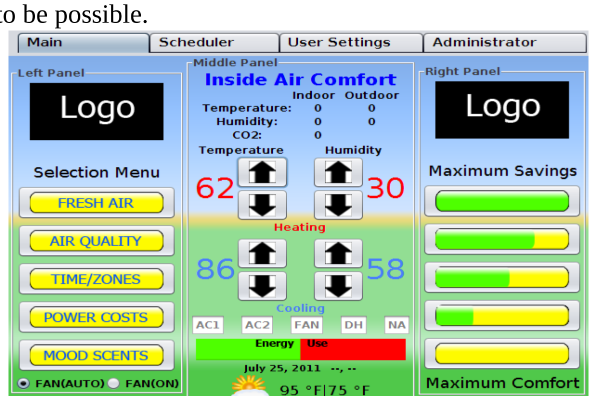


Fig 4 Main Screen

Besides the above stated controls, users can also utilize the built-in weekly planner for selecting desired settings for a future day or select set of days. The scheduler is seen in figure 5.

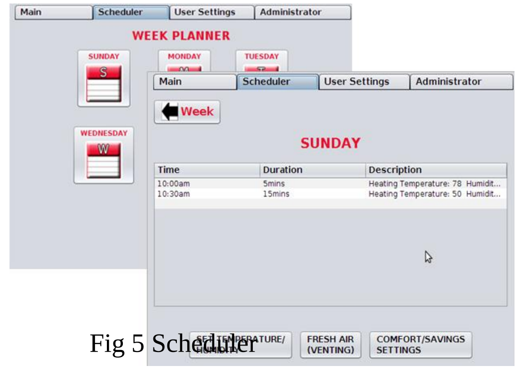


Fig 5 Scheduler

Technicians can also access the password restricted segment of software to select the installed system components configuration which will allow the system to adjust and achieved the optimal environment setting based on the configuration as seen below in fig 6.

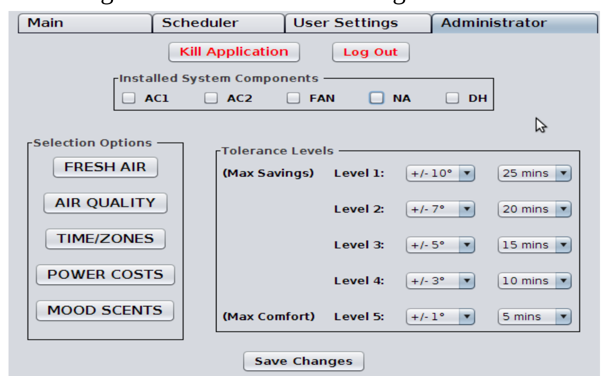


Fig 6 Administration

B. Explorer 16 Developer Board

The testing and code development on our main control unit and outside control unit was done using Microchips Explorer 16 Developer Board. Originally we tested a different PIC24F and a dsPIC33 before settling on the PIC24FJ128GA006 and PIC24FJ128GA010 for the outside and main control units respectively. By obtaining these chips for the board, we were able to program one the exact chips used and see how they interacted with its ports.

The development board was used with MPLAB C compiler from Microchip. The microcontrollers on the main and outside units were both programmed in C. Using this development, we were able to step through the code and find errors and debug while in development stages

C. Main controller

The main controller consists of a wide varying set of components. This controller will be providing the power for everything from the 24VAC power input. This input will flow through a full wave rectifier and smoothing capacitor and into the various buck convertors to distribute out the various DC voltages. In order to not focus all the current on one converter, there will be three 5VDC and two 3.3VDC. The 5 volt regulators will be split off, 1 for the display unit, one for the relays, and 1 for the rest of the components on the main board. The 3.3VDC regulators will be for powering the main board components and the relays.

The main controller will contain a PIC24FJ128GA010 100 pin micro controller to control all the devices. This will use one of its UART ports to connect to an RS-232 driver, which will obviously connect to an RS-232 port that is designed to connect to the RS-232 on the display unit, and this will be their mode of communication. The other UART port will be used for communication to the secondary outside control unit. This port will be connected to both the wireless XBee chip and an RJ-45 line driver. The installer will have the option of choosing which device to use via jumpers, as the device will be connected through one or the other, never both at once.
There will also be two separate sensors connected to this micro-controller on the main board. The temperature and humidity sensor and the CO2 sensor are both take their readings digitally. They will both connect to the microcontroller using I2C protocol.

The relays for the HVAC system will also be controlled by the main board microcontroller. These will be controlled by the I/O lines on the microcontroller, using reverse logic. The relays will be tied to 3VDC on one end and due to the reverse logic, will have 3VDC on the other end when the switch is to be open. Upon a 0 being sent to the relay, the switch will close, allowing the 24VAC through to power the device. There will be a PNP transistor between the relay and micro controller to prevent feedback current bounce when the relay is switched. A pull up resistor will also be used.

D. Outside controller

The secondary controller is on the outside of the building containing the main control unit. The outside unit will take temperature and humidity readings using the same temperature and humidity sensor that the main control unit uses. The micro-controller will be a PIC24FJ128GA060, which is the same as the main unit but less I/O pins. It will be powered outside by 2 AA batteries. The outside controller will have the ability to send the information of the outside temperature and humidity either wirelessly (preferred), or wired (if wireless is no available or capable). The wireless will be controlled by an XBee chip through 802.15 protocol. The data will be sent from the micro-controller to the XBee using one of the UART ports. This will also be the method of sending the data wired. An RJ-45 line driver will be hooked up to the same UART port and then to an RJ-45 port. The driver will automatically invert the signal for the twisted pair. There are jumpers connecting the wires from both the XBee and the RJ-45 drivers to the UART port that the installer will have to decide on which method of communication will be used.

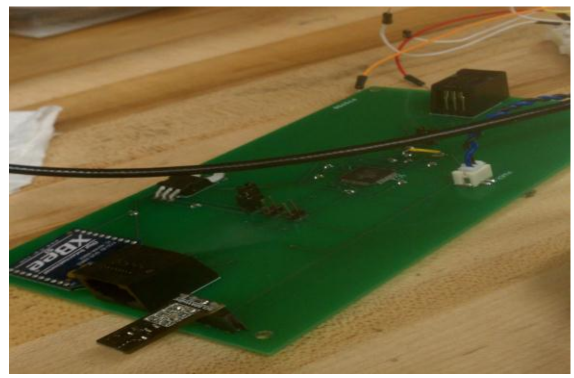


Fig 7 Outside controller

E. PCB

PCB design was a large segment of our project. We had to make two printed circuit boards, one for each control unit. We designed our circuit boards using Cadsoft's Eagle 5.11 software.

The outside controller was the first board designed as it had fewer components and allowed us to learn how to use the Eagle PCB design software. We opted for the two layer board and spaced out traces on our first revision for both boards as should we make a mistake that needed to be corrected on the PCB we could cut the traces and hopefully modify as necessary.

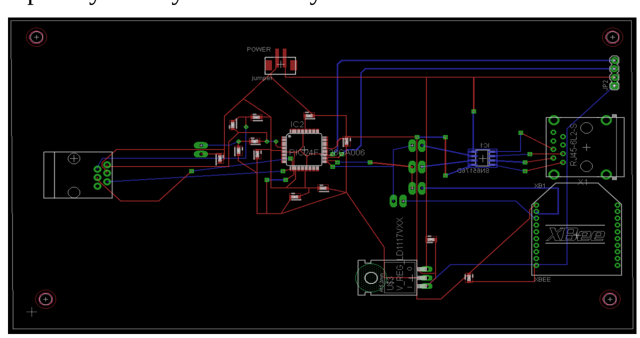


Fig 8 Outside Controller PCB

First, the schematic of the parts were placed from the library in Eagle. This was then converted to the board where the parts were placed and wires routed. Eagle's built in auto-routing was very helpful for setting up the initial routes. We then continued where it left of by moving the position of the components and traces. The board, as seen below, was then converted to the proper gerber files which are used by PCB fabrication companies. These files were sent to 4pcb.com, who we decided to make our board.

The main board was made after the second board. It contained many more parts and took a much longer time to construct. Most of the parts were not in the Eagle library for the main board so several had to be constructed by hand or imported via computer program which then created the footprint and schematic in the program, which was the case for our Microchip processors. After the parts were properly constructed with their proper footprints, the board was then routed. Due to our design requiring a power system and higher currents, we calculated correct wire width for the board as well. Since no one in our group has dealt with somewhat complicated PCB design previously, several tedious electrical connection and design error checks were done both automatically by the software and via double checking the work of our colleges and ourselves through simulations and the data sheets, the board file was completed as seen below and converted to its proper gerber format to send to 4pcb.com.

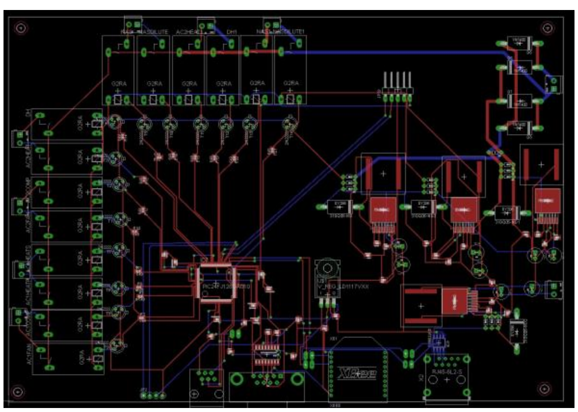


Fig 9 Main Controller Schematic/Layout

III. Testing and Integration

As is in our industry, testing is a key part of ensuring that a product is legitimate and is worthy of being of being placed in the market. We have been extensive amount of time testing and ensuring that the units are fully operable. Once all the different parts of the project got to a threshold, we were able to begin testing and integrating the units towards its final product.

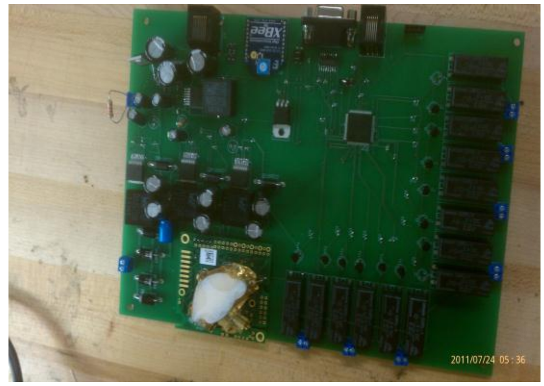


Fig 10 Main Controller PCB

A. Microcontroller with Software(GUI)

We had to ensure the communication between the microcontroller and our software was optimal. One issue that we looked at extensively were the correct display of data such as the inside and outside temperature/humidity, as well as the CO2 reading from inside the building. This is essentially a key part of the system because the user is going to have to know this data in order to make a proper decision as to which components to turn on and it can also be the difference between high energy consumption and low energy consumption. For example, having the correct data will enable the system to turn on Natural Air System if the outside temperature is colder than the inside temperature. Having any tremendous variance in the true data will cause our system to be faulty.

Another issue that we spent time on testing and integration was the responsiveness of the system. Based on the previous version, there was a lot of lag and delays in the communication system such as turning on devices. We ensured that our system was very responsive as to turn on immediately upon activation. We were able to accomplish this by connecting LEDs to indicate every component in our system and going through every single possibility such as AC1, FAN and Dehumidifier coming on. Of course, with this condition, we expected that those 3 LEDs would immediately come on once the software sent that command to the microcontroller. Although we desired a fast responsive, we had to account for situations where this may cause the components to turn on and off because of jitter so we designed our algorithm with hysteresis. Therefore, if AC1, FAN and dehumidifier came on, then it would turn off only after the condition for it turn off plus some delta for it turn off. This was implemented once again to prevent our components from turning on and off within close periods.

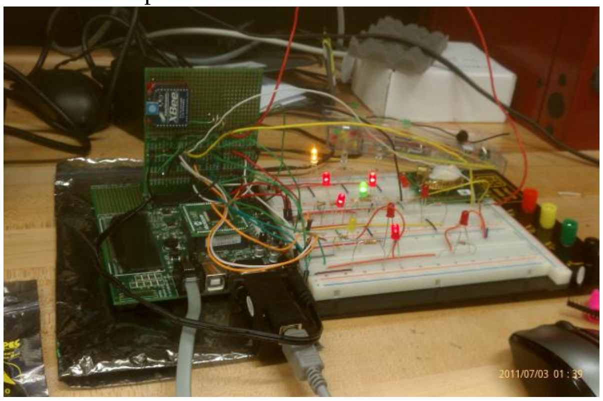


Fig 11 Development

B. Power Supply

To test the power supply we designed around the specifications of the data sheet using our design requirements. We then simulated the circuits using National Instruments design simulator on their website.

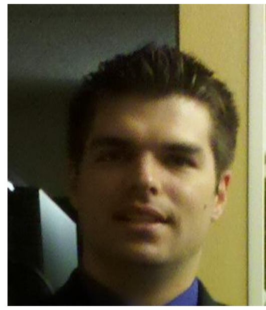
We then verified the circuit would work as expected measuring the output on the oscilloscope of the same parts simulated on a breadboard.

Finally upon receiving the PCB we soldered and tested the output of the power supply components individually as they were being added to the board, without any of the devices they were powering connected to prevent damaging equipment in the event of either a flaw in our design, a flaw on the PCB, flaw in the soldering of the components, or some combination of the above.

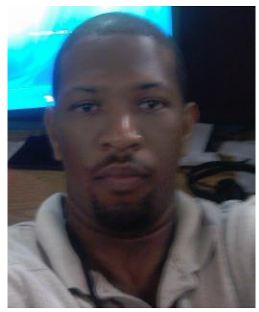
Acknowledgement

We would like to thank Mr. Carson, of HVAC Development Group for their sponsorship of this project. We would also like to acknowledge the assistance and support of Dr. Samuel Richie for his coordination of the Senior Design courses and guidance during this process.

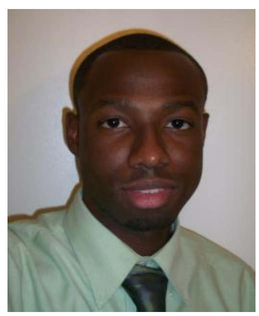
Project Engineers



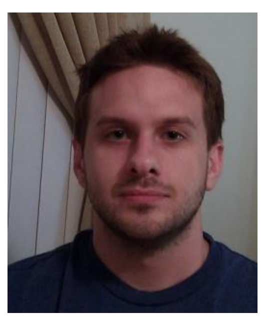
Steven Jones is a senior at the University of Central Florida and will graduate in August 2011 with a B.S. in Electrical Engineering. His interests include salt water fish tanks and technology. Steven plan on getting a job specializing in power systems and continue his education after he has relocated to pursue masters in engineering.



Jerthwin Prospere is a senior at the University of Central Florida and will graduate in December 2011 with a B.S. in Electrical Engineering. His interests include classical and jazz music. Jerthwin plans on pursuing masters after graduation or take a job with Intel or Lockheed.



Elroy Ashtian is a senior at the University of Central Florida and will graduate in August 2011 with a B.S. in Computer Engineering. His interests include the Nintendo Wii and Marvell compics. Elroy is moving to Oregon upon graduation to work at Intel.



Matthew Arcuri is a senior at the University of Central Florida and will graduate in December 2011 with a B.S. in Electrical Engineering and a minor in Computer Science. His interests include rebuilding old cars, technology and documentaries. Matthew plans on pursing masters after graduation.