Automatic Vent Controller

Wendy Dominguez, Gisela Griesheimer, Angelica Longo, David Munyon

Dept. of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This project aims to take our sponsor Chris Neiger's analog vent controller system and update it to a digitally controlled system. A controller and five sensor units have been built to read the inside temperature and humidity of the warehouse, send that data to the controller via BLE, compare our results to outside temperature and humidity collected via web scraping, and then change the position of the vents to cool the warehouse more efficiently. To make the system easier to use, we have installed a touch screen so that our project can easily be controlled in the warehouse and have a website where users can make profiles and remotely control the vents.**

***Index Terms* — Temperature Sensors, Bluetooth, Web Services, Relays, Multithreading, Analog-Digital Integrated Circuits, Databases, API**

I. Introduction

This project consists of updating sponsor Chris Neiger's warehouse ventilation system from its current analog state to a digital automated system. This will make it able to be accessed via an installed touchscreen and cross platform web application with .We have been allocated $2000 to complete this project.

The warehouse is 60' x 100' with the ceiling height ranging from 18' by the walls to 24' in the lofted center, and is located in Niceville, Florida. It has four in-ceiling vents running along the center of the roof that are currently controlled by four double pole double throw switches used to operate the ventilation motors with forward polarity to make the roof vents open and with reverse polarity to make the roof vents shut.

This project aims to replace the analog switch system with a controller and five sensors, which will be able to monitor the temperature and the humidity levels inside the warehouse. These measurements will then be compared to local temperature and humidity data in the controller, where relays will control the opening and closing of vents based on the logic given to the controller. The vent can be automated, or the automation can be overruled to allow the user to choose vent status. The user can do this from either a touchscreen inside the warehouse or the mobile-friendly website built for the project

Fig. 1. The sponsor's warehouse located in Niceville, Fl

II. Sensor Units

A.Parts

Each of the five sensor units built for this project needs to be equipped with a way to collect data on the temperature and humidity and transmit it to the control unit, where it can then be interpreted and acted upon.

Each sensor unit is controlled by an Arduino Nano Every. While prototyping, the Nano Every was used in all designs due to its simplicity, power requirements, and price point. The nano was kept in the design due to COVID supply shortages preventing Measuring both temperature and humidity are important to this project so that incoming rain as well as shifts in temperature can be detected. When looking at sensors for our model, using a 2-in-1 temperature and humidity sensor was the best choice due to space and price constraints. The DHT22 was selected due to it's high accuracy (within 0.5℃) and wider humility spectrum which will be needed to detect rain.

To transmit sensor data, a BLE module was chosen over wifi and standard bluetooth due to its ability to use less power, which was a priority when the sensors were designed to run on battery power. The wireless connection device also needed to be able to communicate across the warehouse, which the HM10 models chosen are easily able to do.

B. Power Source

The Arduino nano every used on each of our sensor units requires between 7-21V of power to operate and supply power to the other components on the board. The Nano Every was, in turn, powered by battery units until installation, at which point it was clear that using a 12V AC to DC wall converter would grant more power stability and lower maintenance for our sponsor.

C. The PCB

The sensor unit PCB was designed in Altium, which was suggested by the sponsor and is compatible with EAGLE files. Its multi-layered design has tracings on both the front and back, making it a 2 layer board, and it measures approximately 5cm x 5cm.

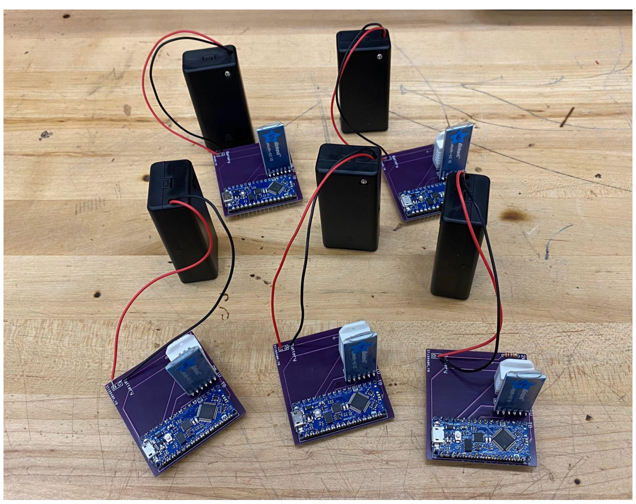


Fig. 2 The five sensor units on their PCBs

D. Casing

To protect the sensor unit PCB once assembled, a casing needed to be created to secure the PCB and its components from any harm. This casing also needed to be breathable so that the temperature and humidity modules can take accurate measurements of the space. The best solution was to create a 3D printed design to the needed size requirements that would also be breathable to the DHT22. The design was created in Solidworks and printed in a black and white plastic filament. The sensor housing is printed in four pieces as shown in the diagram below. Once printed the casings are fixed together with an adhesive to hold everything together. The cases will be affixed to a wall with a nail or wire via a small hole in the back of the casing. Prior to being sealed, the power supply wire will be run through one of the grate openings on the side of the case.

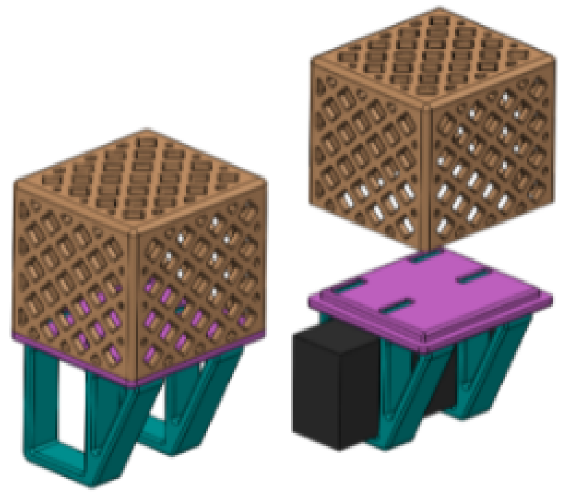


Fig. 3. Sensor Casing designed in Solidworks

III. MAIN CONTROL UNIT

A. Parts

The Main Control Unit is composed of six main parts. These components are the microcontroller, LCD touch display, buck converter, relay module, Bluetooth module, and wifi module. All of these parts serve a specific purpose and are used in combination with one another to make a cohesive unit to achieve the design goals of this project. The first component to discuss is the microcontroller. The microcontroller selected for this project was Atmel's ATMEGA 2560. This microcontroller has 100 pins of which 54 are digital input/output pins, 16 analog pins, and 4 dedicated UART ports. This microcontroller was selected due to its large number of pins that are needed to drive the large touch screen display and 16 relays. This microcontroller also has a lot of documentation to support its use because of its popularity, and has proven itself in prototype testing. The purpose of the microcontroller in this project is to send and receive data from the sensors and website, process, and utilize the data to control the vents.

The next component to talk about is the LCD Display. The display that was chosen for this project is Adafruit's 3.5" TFT Touch Screen Display. This display was chosen first due to its size. The 3.5" display allows for a clear user interface that can be seen from a distance away and allows for bigger touch screen buttons which make the user interface easy to interact with. The second reason this display was chosen is due to the drivers that Adafruit supplies with the display. The drivers make programming easier and come with documentation to support its use. The purpose of the LCD Display in this project is to display the most recent data from the sensors and to display the current status of the vents in the warehouse. The LCD display also provides an intuitive way to vents locally and see other data such as wifi connection and individual sensor temperature.

The third component used for this project is the power supply. The power supply chosen for this project is the LM2576 Switching voltage regulator. This voltage regulator has a set output of 5V and is capable of supplying up to 3A. This voltage regulator was chosen due to its availability, as many of the other voltage regulators are back-ordered due to the supply shortage. This voltage regulator is mid-range with only 75% efficiency but is still more than suitable for the needs of this project. The supply voltage for this switching regulator is a 12V DC power supply that is used to drive the vent motors. The LM2576 will be supplying the power required for all processes on the PCB. It is estimated that during normal operation the PCB will draw 150mA of current with the largest current draw being the wifi module. However, when the vents need to be opened or closed the relay modules will draw around 80 mA per relay. This transient operation may require up to 750 mA for a thirty-second period. During testing the LM2576 was capable of supplying this with ease.

The fourth component of the main control unit is the relay modules. These modules will be connected to control the polarity applied to the 12V DC vent motors that are already installed inside the warehouse. The relay chosen for this is the Elegoo 4-channel relay unit. These relays were chosen due to their popularity and price point. The 4 channels will be run in pairs and each pair will control a direction of travel for the DC motor. The current warehouse contains four vents each with its own motors. A four-channel relay module will be required for each motor, making a total of 16 relays. Each four-channel relay module requires six wires, one for 5V VCC, one for ground, and four wires to be connected to digital input/output pins which will control the operation of the relays.

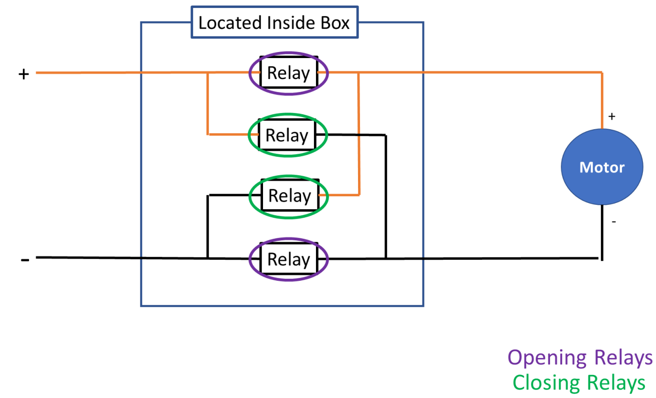


Fig. 4. Relays Wiring Diagram

The next component to discuss is the Bluetooth module. The Bluetooth module selected for this project is the Inland KS0455 HM-10 Bluetooth 4.0 Module. This module was selected after testing several failed modules because of its reliability and comprehensive datasheet. This module has the additional benefit of utilizing a low power draw with the Bluetooth low energy feature of Bluetooth 4.0 technology.

The purpose of the Bluetooth module for this project is to provide a reliable method of communication between the temperature sensors and the main control unit. The main control unit controls the connection between the five temperature sensors by utilizing a loop that connects, receives data, and disconnects before moving on to the next sensors. These connection and disconnection commands are done by utilizing the built-in AT+ commands that are native to the Bluetooth module. From rigorous testing, the Bluetooth modules have a reliable range of 75 feet which will be more than enough to provide adequate coverage in the warehouse.

The last component to discuss is the wifi module. The wifi module selected for this project is the Adafruit HUZZAH ESP8266 Breakout. This module was more expensive than several others that were tested for this project, but it provided the most reliable connection to the internet. This module also comes with robust documentation to support its use with this project. The downside of using this module is its programming requires a special FTDI cable instead of utilizing the microcontroller to implement its program.

The purpose of the wifi module in this project is to establish a reliable method of communication with the internet. This connection is used to send data collected from the sensors to the database to be stored and to receive data from the website. The data received from the website will be used for remote, manual control of the vents and also web scraped data of the outside air temperature.

C. Main Unit's Display

The touch screen on our Main Unit displays two different screens, one is the "Home Screen" and the other is the "Information Screen". The Home Screen displays outdoor temperature and humidity gathered by local weather API, the indoor temperature and humidity gathered by taking the average readings of the 5 Sensor Units placed within the warehouse, 8 ventilation control buttons that show the current state of each vent, and lastly a button at the top right to navigate to the Information Screen.

Fig. 5. Controller Main Screen

The Information Screen displays additional device information and controls such as the ideal temperature set by the user, the temperature and humidity readings from each of our Sensor Units, a reset button to restart Bluetooth communications, a button at the top left to navigate back to the Home Screen, and the automation override feature for when the user wants to turn off automation and manually take control of the vents with the buttons on the Home Screen. This override is meant for temporary use and thus automation begins again after 3 hours pass.

Fig. 6. Controller Information Screen

The graphics on our project's LCD touch display are made using the Adafruit_GFX library for Arduino. The Adafruit GFX graphics library is the 'core' class that all other graphics libraries derive from. It is a generic graphics superclass that can handle all sorts of drawing for all Adafruit displays and is compatible to use on all Arduino boards. It provides a common set of primitive graphics which allows us to do things like filling the screen with colors, setting texts, and drawing shapes such as lines, dots, rectangles, and circles. Adafruit_GFX always works together with an additional library unique to each specific display type. This allows Arduino sketches to easily be adapted between many different display types with minimal fuss. Thus, we paired our graphics library with the hardware-specific Adafruit_TFTLCD display library for our Adafruit 3.5" LCD touch display in order to handle other functions such as touch capabilities.

IV. SOFTWARE

A. Web Application Software

We wanted to be able to give our users the capability of not only having control of the vent controller through their mobile devices but also being able to utilize any computer connected to the internet, making it as cross-platform as possible. There was a lot of styling that had to be completed to make sure that the interface was coherent and consistent at any size of the screen, ranging from small mobile screen displays to laptops and monitors. On the website itself, all our information is stored in a

MySQL database, this includes a user's information, their full name, email, and desired password; As well as the information collected from the Arduino module, all 5 temperatures and humidity values collected from each sensor, as well as the average temperature and humidity computed from the 5 values. These can be seen in the following ERD diagram,

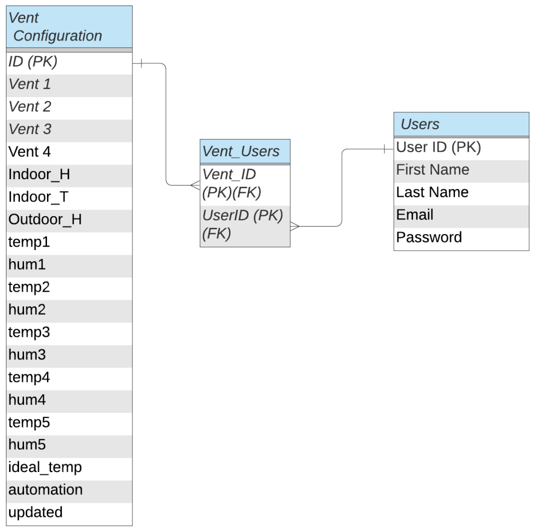


Fig. 7. Entity Relationship Diagram

There are 4 main sections on our website, the login page, register page, home page, and account information page. The first page that pops up when clicking on our link is the login page, if a user has already signed it though, they will automatically be redirected to our home page. If a user does not have an account there is a link to be redirected to our register page, this page is where any user is asked for the four aforementioned information. Once this form is completed, there is an API that checks that the email address entered does in fact exist, it does not send an email to them but checks that an email can be sent. If the email does exist then the account gets successfully created, they are redirected back to the login page. Clearing the login page, any new user will only be able to see an *Add Controller* section, this was implemented to add even more security, this way the vents aren't messed around with by a random individual. The user is forced to insert a five-digit controller ID to see any information for a particular controller. At any time before a controller is connected to an account, there can always be changes made to a user's information. After a controller ID is submitted and verified to exist, using our PHP API, only then is a user able to see all of the information divided into three main sections. The first of these is called ventilation, and it is populated by the configuration of each of the four existing vents, each of which has two possible modes, open and closed. To show which mode is currently in use, we have an underline under the word.

Following this section, we have the climate section, which shows the data collected from each of the sensors placed around the warehouse as well as the average of these values. It also contains the data collected from scrapping the OpenWeatherMap API. This site comes with a free-to-use API that gives any free user the ability to have 60 calls per minute without restriction, it also provides other weather data like 7-day forecasts and national weather alerts. Though we only collected the temperature and humidity data as they were part of the deciding factors on the automation of each vent. We added a button to call the API and change the values in the database as well as in the website, this change only occurs when there is a change from the OpenWeatherMap API, however, since temperature and humidity don't usually change drastically within seconds or even minutes there is only a slight change if any. Our last main section contains the device information and the override button, this override functionality allows a user to open or close a vent as they see fit, deviating from the automated plans.

Our last page is the account page, which contains the information that gets entered upon the creation of the account. This information can be changed if the user wants to change it by selecting the field from a drop-down menu and filling out the text box to replace it. These pages were all PHP files, containing mixtures of JavaScript and HTML. We decided to keep the CSS out of the main pages as they ended up being over 300 lines of code. With our home page taking the crown at about 1900 lines.

All these functionalities can be seen in our use case diagram, and we can visually see how a user will be interacting with our application. We decided to use a use case diagram instead of other UML models available because it was the clearest form to show what our goals were.

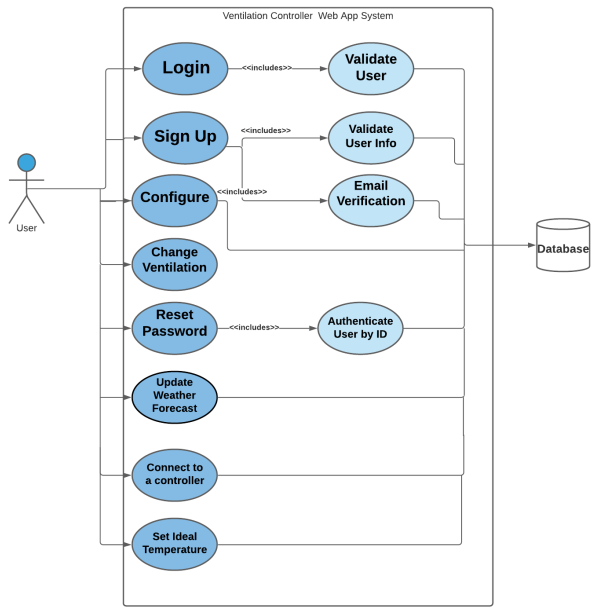


Fig. 8. Use Case Diagram

B. Web Application Design

Based on the use case diagram, we wanted a simple yet sleek and straightforward way to display the information to the user. We prototyped in Figma as it had the best free resources as well as the most open-sourced we could find. This also helped when it got down to style each and every element in the page. Below is the final design of the website.

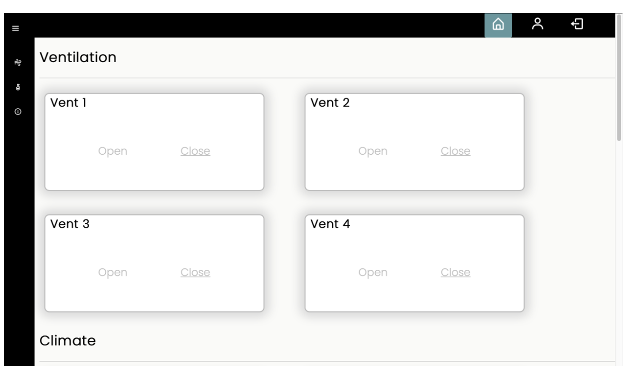


Fig. 9. Website ventilation status display

C. Wireless Connections

Our Main Control Unit has an HM-10 Bluetooth module set as <u>master</u>, and all the Sensor Units placed around the warehouse have HM-10 Bluetooth modules set as <u>slaves</u>. Mainly, the master coordinates communication throughout the network since slaves cannot talk to other slaves. Since we have multiple slave modules, they will all be waiting for our one module operating as a master to initiate a connection. While setting up the master module, we establish a connection (also known as "*paging"*) by providing the IP address of a slave module. Once this is done, the two Bluetooth modules will be paired together and are ready to begin transferring data to each other. Thus, we use Point-To-Point protocols so that each slave module is the one with gathered data from sensor readings and the master module is the one that connects to the slave to gather that data.

D. The Communication Loop

Essentially, we instigate a bi-directional communication with each slave's IP address in a loop. Since we know all the IP addresses for each of our slave modules, we can connect to them one by one. Before beginning our communication loop, the Control Unit's Bluetooth module must first be configured to master at startup. The loop then officially starts when the master module sends a connection command to Sensor Unit 1 using its IP address and is able to establish a connection. Once a connection is made, temperature and humidity data is read from the Sensor Unit's DHT22. After grabbing data, the master immediately breaks communication from the connected slave module by cutting off its power, restoring the power, and then disconnecting. Subsequently, the master will then begin paging the next Sensor Unit's slave module and repeat the process all over again in an infinite loop.

The multiple indoor sensors will use this "communication loop" in order to gather a well-estimated amount of climate data within the warehouse. After reaching the 5th Sensor Unit, there is enough data to calculate the average temperature and humidity, and the loop repeats. Every time the loop restarts, a long data string is generated which holds all the data that was taken from each module, including the calculated average temperature and humidity, the state of each vent, and the automation status. This long string of values is how we manage to send over all our Arduino data to our database. The data string is generated in a way that each value in the string is separated by a comma delimiter so that it can be parsed and stored in our database once received by our website's PHP code. Since the time it takes for all our Sensor Units to complete a full loop is usually about 40 seconds, the climate and information on the touch screen can be expected to be updated about every minute to additionally account for the time it takes to send and parse the data string over Wi-Fi.

V. INSTALLATION

The first part of the installation process was preparation. The objective of this preparation was to make the final installation in Niceville, Florida go as smoothly as possible. For this to occur an installation plan had to be made. The installation plan consisted of four parts: Pre-trip, Hardware Installation, Logic Programming, and Final Testing. This section will go into detail about these four sections and how each aided in the final installation of this project.

The objective of the pre-trip was to make the Hardware installation, Logic Programming, and Testing goes as smoothly as possible. For this to be successful everything that could be taken care of prior to the trip should be taken care of. This meant testing out the project fully in the lab to ensure that all components worked together. Additionally, the PCB and all components needed to be mounted in a case. This would ensure safety during transportation and ease of hardware installation while on site. For the case, we decided to use a standard electrical box made by Cantex. The box proved to be large enough to house all components, Rigid enough to protect them, and easy to work. The PCB was mounted to the back of the box using Velcro tape which enables it to be removed and replaced if needed. The Wifi Module was mounted on the left side of the box using female pin headers. This allows the Wifi module to be connected and disconnected with ease for additional programming if needed. A USB port was also added to the left side of the project housing to facilitate uploading code to the PCB. This allows for making quick adjustments to the automation algorithm while at the warehouse without the need to open the box every time. The Bluetooth Module was mounted at the end of a 10-foot wire which allowed it to be placed above machinery that may have potentially blocked it from connecting to all of the indoor sensors. The screen was mounted to the front cover of the box to allow for easy viewing and control of the indoor data and vent positions. A large hole was drilled on the right side of the box to give the relay module connection a path to be installed in the existing box at the warehouse.

After everything was completed for the Pre-trip phase. The project was ready for installation in the Sponsor's Warehouse. The next phase of the project was Hardware installation. The first step of hardware installation was mounting the project housing to the wall and connecting it to the existing box located to its right. The box on the right houses the vent wires, 12V DC power supply, and relay modules.

Fig. 10. Controller Final Installation

The next step to hardware installation was to mount the Bluetooth sensor units. The sensor units need to be placed around the warehouse in a way that displays any pockets of heat while also giving a strong picture of the average temperature throughout the space. Once in the warehouse, it was clear that placement positions would be more limited than previously anticipated. We chose to mount three of our sensors on the longest wall, one on the protrusion of the control room, and one on the back wall. This allowed for testing a large area of space, and would also get reading from where people will be working most.

After all the sensors were mounted in their place the next step was to connect all of the relay modules to their respective vents. Wiring all of the relay modules took much longer than expected during the installation process due to the limited space inside of the box that housed the relay modules. Once all of the hardware was installed in its permanent location. The next step of the hardware installation was to create the automation logic.

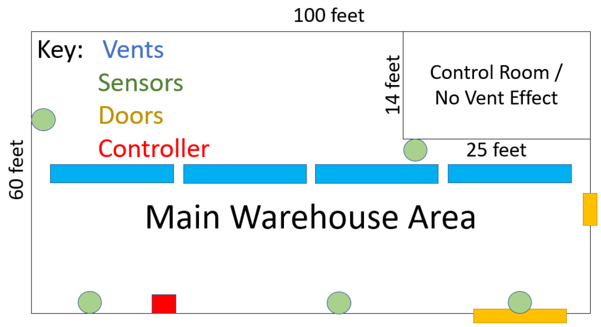


Fig. 11. Sensor Placement Diagram

The automation logic was designed to cool the warehouse during the summer. To accomplish this an ideal temperature was used. The ideal temperature can be set by the user. If the indoor air temperature reaches the ideal temperature the vents will close to trap the heat inside of the warehouse. If the indoor temperature is greater than the ideal temperature, the controller compares the indoor temperature to the outdoor temperature. If the indoor temperature is greater than the outdoor temperature the vents will open to release heat from the warehouse. The automated logic was also designed to help keep humidity out of the warehouse. To accomplish this, When the controller reads that the outdoor humidity is greater than 90% it is assumed a storm is passing through and all of the vents will close. Lastly, a hysteresis was implemented to prevent the vents from continually cycling open and closed if the temperature bounces around the setpoint.

The final step of the installation was to test all features of the automated ventilation controller. The first item to test was the manual control feature. To accomplish this the touchscreen was used to place the controller in manual mode and press the buttons on the touch display to open and close all of the vents. The next test was manual control of the vents via the website. To accomplish this, the website was accessed via a mobile phone. The control unit was placed in manual mode remotely and all of the vents were cycled open and closed via the buttons on the website. The next test was to ensure the manual mode timer would place the controller back in automatic mode after 3 hours. To accomplish this The vents were placed in manual mode and left for 3 hours. After the 3 hours had expired the controller checked to ensure that the mode was changed to automatic by the timer. The final test is to ensure the automation logic works properly. To accomplish this the vents were placed in automatic mode. The outdoor data and indoor data were monitored for several hours to ensure the vents cycled properly throughout the day.

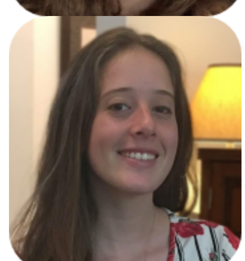
ACKNOWLEDGMENTS

The authors would firstly like to thank Chris Neiger, who is sponsoring the project and whose guidance we have relied on in our solutions. We would also like to thank Dr. Lei Wei and Dr. Samual Richie, who have advised us throughout our senior design journey. Additionally, we would like to thank David Douglas for allowing us to use of the senior design lab and Quality Manufacturing Services for aiding in our surface mounting on the controller PCB. We would finally like to thank all of the members of our review committee for volunteering to give feedback on our work over the last two semesters.

BIOGRAPHY

Wendy Dominguez is graduating with a degree in Computer Engineering and plans to work as a Software Engineer.

Gisela Griesheimer is graduating with her degree in Electrical Engineering and plans to do work in the areas of project management or technical sales. She is excited to graduate and begin her career.



Angelica Longo is graduating with her degree in Computer Engineering and plans to work in Software Development while continuing education in the field of Interactive Entertainment and Video Game Design.

David Munyon is graduating with a degree in EE and plans to work for FPL designing distribution systems in North Florida.