ECE 445

SENIOR DESIGN LABORATORY

INDIVIDUAL PROGRESS REPORT

Automatic Humidity Sensing and Water Refilling Cool-Mist Humidifier

---

Team No. 11

ANDREW SHERWIN
(zyxie2@illinois.edu)

TA: Surya Vasanth
Professor: Dr. Jonathon Schuh

March 27, 2024

Abstract

This paper is an individual progress report for the work done by Andrew Sherwin as an individual in ECE445 – Senior Design Laboratory. Andrew Sherwin is the active member and the team manager in Team 11. The team consists of members: Andrew Sherwin, Woojin Kim, and Jalen Chen. In Spring 2024, Team 11 is designing an Automatic Humidity Sensing and Water Refilling Cool-Mist Humidifier. The humidifier can turn on automatically in relation to the detected humidity and temperature values, as well as refill its water tank depending on the water level value. Andrew Sherwin is responsible for the team's project progression, as well as maintaining a checklist all goals are reached at an appropriate time. He is also responsible for the selection of components and design of the PCB. Aside from that, he also helps with the software design for the control of the microcontroller. Andrew Sherwin is also the representative for the team during communication with the course staff, Surya Vasanth, and the machine shop.

Table of Contents

Introduction

Problem

In September 203, the United States Environmental Protection Agency (EPA) released an article regarding indoor humidity recommendations. The EPA suggests indoor humidity to range between 30 and 50 percent. Though a lower humidity decreases the presence of pests, such as mites, it allows for airborne particles and pollutants to freely travel around. A higher humidity above 50 percent leads to a higher percentage of mold growth. As a result, it is critical to keep indoor humidity between 30 to 50 percent, at room temperature.

The majority of cost effect humidifiers in the market are manually controlled. They do not feature an automatic on-off feature. When the humidifier user is not at home, the indoor humidity can pass 50% humidity when the humidifier is left on. Similarly, the indoor humidity can drop below 30% when the humidifier is left off. In addition, all humidifiers in the market require manual water filling by the user. Our humidifier strives to solve the issues above to guarantee safe and healthy indoor humidity for the user.

Solution

To resolve the issue of a manual humidifier, our design proposes a solution that can be incorporated into the majority of manual consumer humidifiers on the market. We will have a system that has three remote humidity sensors that relay humidity data to the base system at the humidifier. The humidifier will plug into the base system, and the base system will plug into the wall. Based on the humidity sensor readings, the base system will use an in-house developed algorithm to turn on or off the humidifier. Furthermore, the humidifier's water tank will be connected to a DC water valve. Depending on the water level sensor's readings, the base system will activate or deactivate the water valve to refill the humidifier's water tank. The laissez-faire approach to our humidifier's function generates a luxurious and revolutionary product for the market with high autonomy.

Team Responsibilities (non-design)

Andrew Sherwin is responsible for managing the team and giving the team a direction in design philosophy and written reports. Andrew is also responsible for the communication between the team and the course staff, as well as the machine shop. Andrew has set out specific goals to be accomplished each week in the 16-week semester. Every week, Andrew actively communicates with the team members in-person and on the private Discord server about goals and requirements to be met at the end of the week. While Andrew participates in the written reports, he also proofreads and formats the written reports for the course. Andrew leads the team on the weekly meetings with the TA, Surya Vasanth, and updates the Machine Shop bi-monthly about the team's project progression.

Sensor Subsystem Responsibility

In *figure 1*, the sensor subsystem, Andrew Sherwin was responsible for choosing the components and designing the PCB for the subsystem. Andrew has been cooperating with Woojin in the software design of the humidity sensor subsystem. The ESP32 has to read the data of the humidity sensor and relay it to the ESP32 at the base system in the humidifier subsystem. In the hardware design, Andrew has been picking components suitable for this subsystem, while communicating with Jalen regarding the power subsystem in *figure 1*. With his prior experience in PCB design, Andrew is also responsible for drawing the schematic and PCB layout in KiCAD, whilst Jalen proofreads the system design.

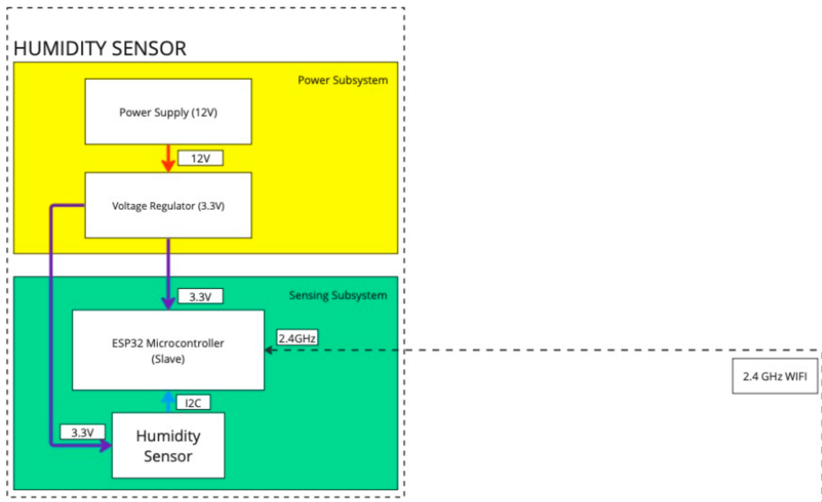

Figure 1 – Sensor Subsystem Block Diagram

Humidifier Subsystem Responsibility

Similar to the sensor subsystem's responsibilities, Andrew has been working with Woojin on the software of the ESP32. Andrew developed a software design for an access point that Woojin uses for the ESP32 communication. Andrew has also developed a program where LED lights can be controlled using a phone on the ESP32's MAC address. The webserver also shows a live readout of the LED values. On the hardware side, Andrew has been working on picking components suitable for the humidifier subsystem in *figure 2*. For example, Andrew selected the water level sensors, electrical relays, and numerous other parts for this subsystem. Similar to the sensor subsystem, Andrew also designed the PCB schematic and layout of the humidifier subsystem. Jalen has worked with Andrew by discussing on the choice of parts and proofreading the KiCAD schematic.

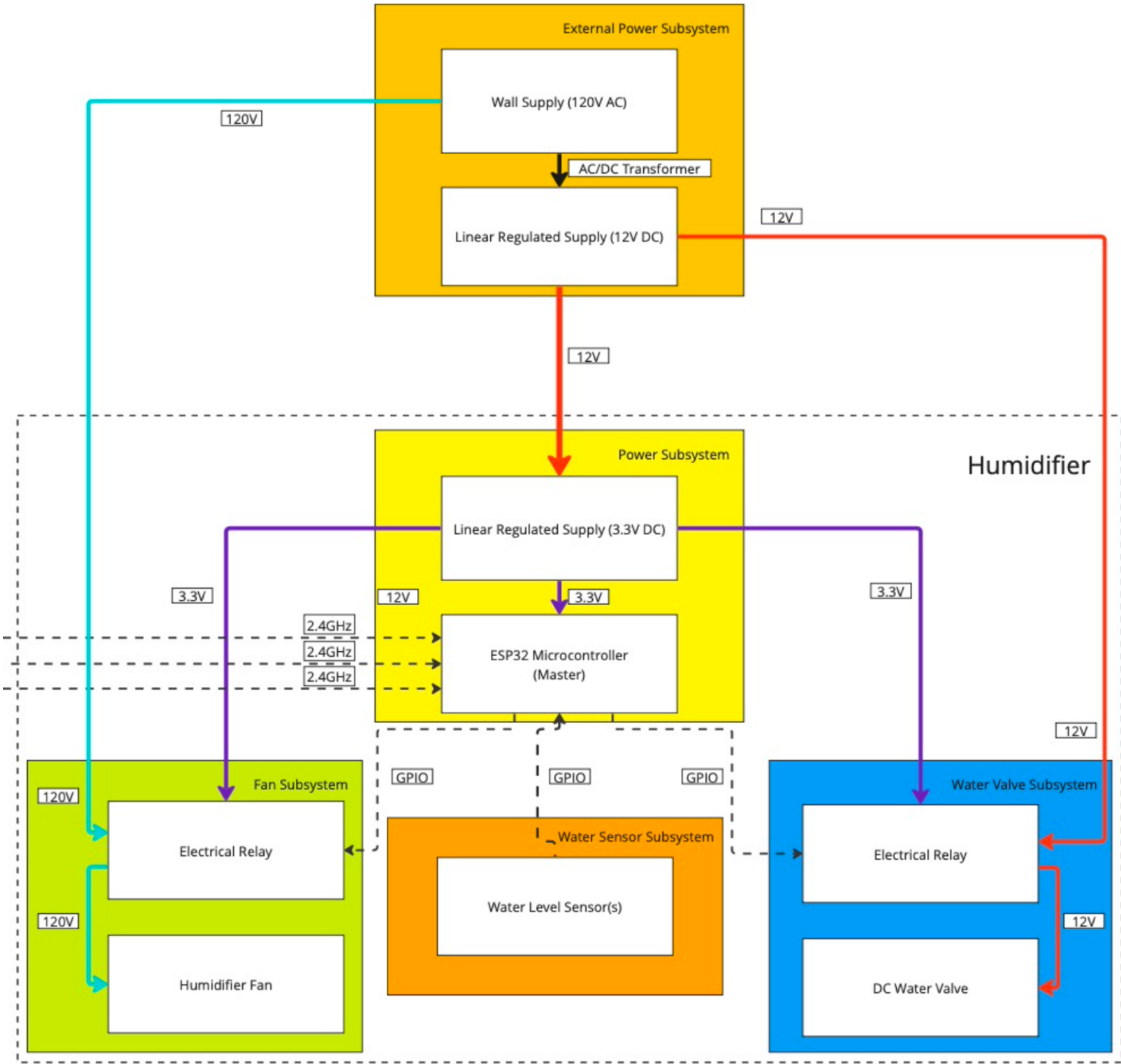

Figure 2 – Humidifier Subsystem Block Diagram

Individual Design

ESP32 Design

In the development phase, we used an ESP32-S3 WROOM1 development board. This board uses Micro-USB Type-B ports to upload the programs to the microcontroller. The Micro-USB sends UART data to a UART communication chip, U4 in *figure 3*, to communicate with the ESP32. When programming, the BOOT button had to have a logic of 0 to load the button, hence the importance of the BOOT button. The RESET button loads and starts the program in the microcontroller. It is important to replicate these parts of the development board's design in our PCB design, as they are critical for the functionality of the microcontroller. It is also important to have Zener diodes to act as reference voltages and to prevent static discharges from affecting other components. We also have some 0-ohm resistors for testing. The design also includes an alternative way to program the board, using just the Micro-USB. This is a secondary way just in case if the USB to UART bridge does not work. The completed schematic, with the humidity sensor can be seen in *figure 3*.

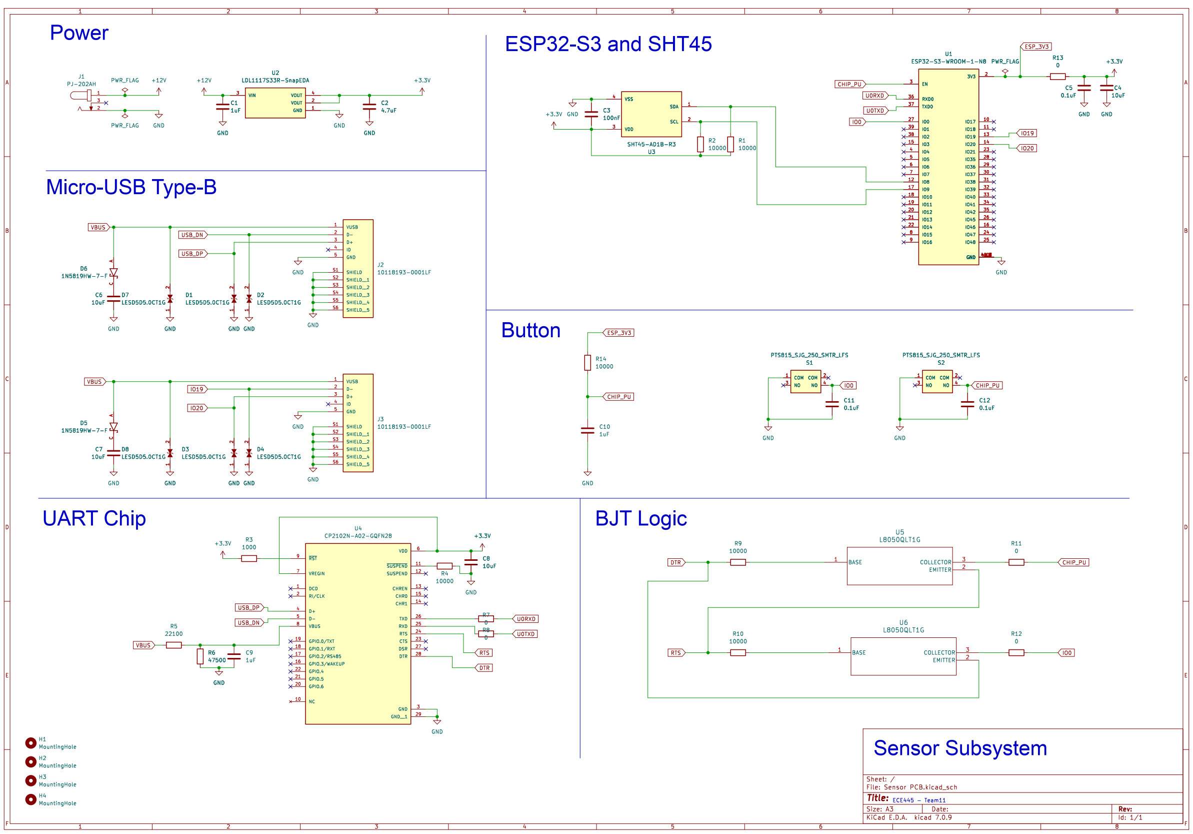

Figure 3 – Sensor Subsystem Schematic

Sensor Subsystem Progress

Figure 3 shows the KiCAD design for the sensor subsystem designed by Andrew Sherwin. Most of the diagram consists of the ESP32 circuit, with pins 12 and 17 being the input of the SHT45 humidity sensor (U3). The SHT45 communicates in I2C, which means it needs SDA and SCL pins. The SHT45 has a 100nF decoupling capacitor, as specified by the Sensirion documentation, as well as one 10k ohm resistor for SDA and SCL respectively. The SHT45 sensor will be mounted on the rear of the PCB, with the ESP32 and all other components being on the front. This will isolate the humidity sensor, thus, making the readings more accurate.

The SHT45 was determined to be used due to its accurate temperature and humidity measurements. Whereas most other sensing devices has an error range or 2%, the SHT45 has only an error range of 1% in the indoor temperature range around 23ºC, as seen in *figure 4*.

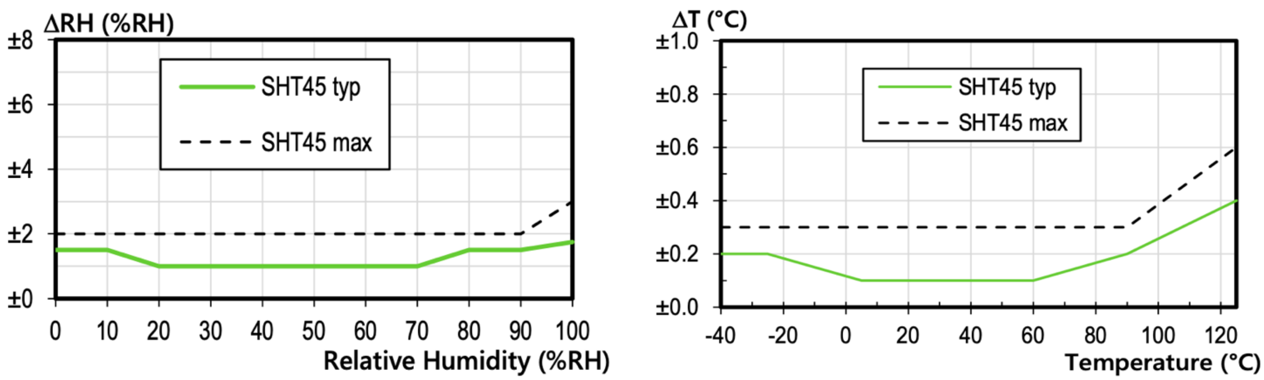

Figure 4 – SHT45 Reading Accuracy Table

Humidifier Subsystem Progress

Originally, we planned the ESP32 microcontroller in the humidifier subsystem to be communicating with two motor controllers. The first which controls the DC water valve, and the other that controls the humidifier fan.

As the machine shop is not creating a ground-up humidifier for the team, taking apart and controlling the humidifier's fan proved difficult. As a result, the alternative solution was to control the humidifier with a relay. The relay will work like a light switch that is controlled by the ESP32. The switch will allow 120V AC to reach the humidifier. A sample diagram is shown in *figure 5* below.

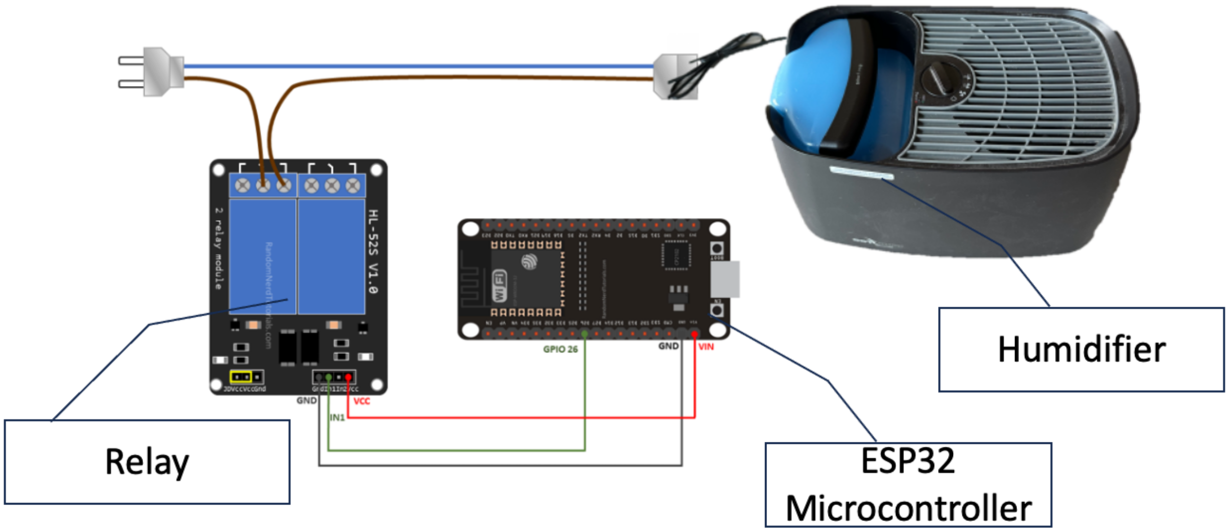

Figure 5 – Relay Diagram

Similarly, it was discovered that the original water valve that the team intended to use did not work unless there was active pressure on it. This will normally work when connected to a building's water supply. However, for our demonstration, there will be no water pressure. As a result, we had to change parts. The pervious water valve featured a stepper motor. This required a motor controller. However, the new one features only a simple DC motor, which did not require a DC motor. As a result, we can simply use another relay to control the water valve, with a setup like that of *figure 5*.

When designing the schematic in *figure 6*, most of the schematic was a carry-over from the ESP32 schematic. The main difference in *figure 6* was the inclusion of the two relays. A design choice was made during the design of the relay schematic to include a kick-back diode and an optocoupler. The kick-back diode helps prevent voltage spikes from going to the ESP32 when the circuit is turned off. The optocoupler helps isolate the ESP32 from the higher voltage components in the relay. This further prevents noise and voltage spikes from destroying the ESP32.

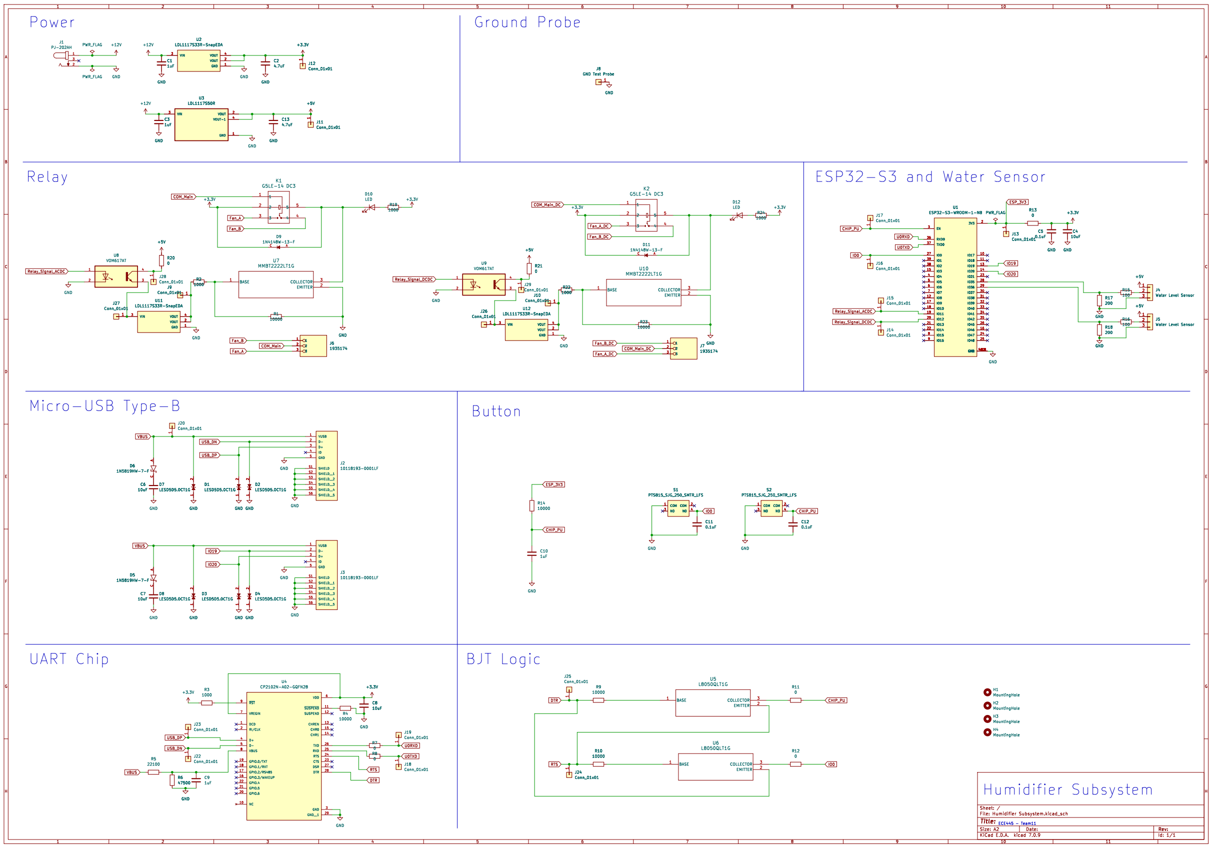

Figure 6 – Sensor Subsystem Schematic

Individual Verification

ESP32 Progress

Andrew was the first team member to work on the ESP32. He set up the Arduino IDE and installed the necessary libraries for programming the ESP32. The first basic test that Andrew did was turning LED lights on and off. Through a button input to a GPIO pin, depending on the logical input, the ESP32 will match the input signal to another output GPIO with an LED. For example, then the button is pressed, the LED will turn on, and the opposite vice-versa.

After confirming the functionality of the setup, Andrew worked on setting up an access point for the ESP32. He planned to connect to it with his personal device. On this access point, after accessing the MAC address on the personal device, he created a HTML code that shows the live output of the LED mentioned in the previous paragraph. When the LED is on, the webpage will display “ON”, and “OFF” when the LED is off. This will be the starting point of being able to read the humidity sensor data on a personal device, which the team plans to accomplish later as a bonus.

Sensor Subsystem Progress

Coupled with the ESP32 progress, Andrew has aided Woojin in the verification of the software development in the sensor subsystem. Andrew helped troubleshoot the code in the Arduino IDE for the SHT45 humidity sensor using modular and system testing methods. Originally, the ESP32 was not getting an input from the SHT45. The troubleshooting method was to first probe the ESP32’s SDA output and the ESP32’s SCL clock speed. Using an oscilloscope, Andrew verified that the output message was correct for the ESP32’s master instructions. Following, the oscilloscope was used again to probe the clock speed. The error was simple, the clock speed was incorrect. The ESP32 was using I2C at the “Fast-Mode”, transferring data at 400k bits/s, while the SHT45 was supposed to be reading and sending data at 100k bits/s. The SHT45 was not able to receive the ESP32’s instructions, due to the ESP32 sending data at a 4 times higher frequency than the SHT45 was reading.

Verification of the PCB that was placed on the first round will be done when the part for the PCB arrives. The PCB has test-point pads that provides easy access for probing signals with an oscilloscope. This was done during the design process. On the final PCB, the test-points will be removed so that the final product will be cleaned up.

Humidifier Subsystem Progress

In the humidifier subsystem schematic and PCB design, 1x1 connectors are included for trouble shooting. The 1x1 connectors will make it easy to probe the oscilloscope on the section. Furthermore, there are numerous 0-ohm resistors throughout the schematic and PCB. The reason for these resistors is so that jumper wires can be soldered for testing if the original traces were not correct. This is highly suggested for the UART TX and RX connections. The use of the 0-ohm resistor pads can be seen in *figure 6*.

The new 12V DC water valve operates as a plunger. When it receives 12V power, there will be 0.48A of current going through the water valve solenoid. This results in a magnetic field, which pulls the plunger up, allowing water to pass through the valve. Andrew did some modular tests on the percentage chance the valve will activate at the lower voltage. Based on 100 trials for each 0.2V increment from 10V, it can be concluded that there is 0% chance that the valve will operate in the range 0V to 11.4V. There is a 39% chance of the valve activating at 11.6V, 94% chance at 11.8V, and 100% chance at voltages 11.9V and greater. The with 1100 trials performed, and 1100 data collected, Andrew is confident that as long as the relay supplies 12V, with a 0.2V error range, the valve will function for our design. Through this verification, and by disassembling the valve to understand the inner workings, Andrew brought up the suggestion to use a valve rated for a higher pressure should be considered for the project, as this valve does not rely on input pressure. This will guarantee a better water seal within the valve.

The PCB testing will be similar to the process mentioned in the Sensor Subsystem Progress Verification section. Currently, the PCB has not arrived yet, but there are test-points for probing the PCB, if needed, with an oscilloscope to decipher the signals. The testing will be done within 3 days of the completion of the component soldering onto the PCB. This will allow the team to place a new PCB order if needed for revisions.

Conclusion

Self-Assessment

The workload for Andrew Sherwin in ECE445 is a bit higher than his peers in his team. This is expected as he is managing the direction of the project and is the representative for the team.

Andrew communicates in-person and on Discord with the team to get updates and progress on the project. He makes sure that all goals are met on time, or ahead of time. If there are any issues with parts, he quickly organizes meetings and finds solutions for troubleshooting. As the representative of the team, Andrew communicates with the TA, Surya Vasanth, and the machine shop regarding project updates and parts to order. Each week, Andrew sets out new goals and plans for the team to meet, in accordance with the set schedule. Andrew makes sure that each milestone is met at, or ahead of a certain date.

As the bridge between the software engineer, Woojin, and the hardware-power engineer, Jalen, Andrew is involved in both parts of the project. With his prior experience, he also is the designer for the PCB schematic and layout. He helps both members troubleshoot potential issues and helps clarify unclear project goals.

With Andrew's planning, the team is currently ahead of schedule. During the first round of PCB orders, Andrew placed an order for the Sensor Subsystem's PCB. During the second round of PCB orders, Andrew placed an order for the Humidifier Subsystem's PCB. On the week of the third round of PCB orders, with the help of Andrew's original ESP32 research, Woojin has updated the team that he is able to send and receive humidity data from one ESP32 to another ESP32 using 2.4GHz Wi-Fi. With the above finished, the team is 2 weeks ahead of the original conservative time schedule.

Future Plans

Currently, the team is waiting on parts from Digikey, so that the PCBs can receive their components. Jalen will be soldering the PCB with the later arriving components, and Andrew will be working with Woojin on programming and troubleshooting any potential errors from the PCBs.

Andrew will also be working with Jalen on the wiring of the humidity subsystem. The team will need to purchase a 120V AC to 12V DC transformer assembly. Currently, Andrew has his sights set on the Mean Well USA EPS-15-12 AC/DC converter. This will output a 12V DC line that will go to power the PCB via a barrel jack connector.

If the PCBs do not need revisions, the team will work on a webpage to view the live humidity sensor readings from the SHT45 sensors. This will be a bonus to our project if everything is completed early. However, this plan will only commence if the original design is indeed functional.

As a team, Team 11 will continue to follow all IEEE Code of Ethics outlined in section 7.8. We will uphold the highest integrity and responsible behavior when designing our product. Working as a team, we will treat one another with respect, and will not engage in any form of discrimination. We will strive to uphold the IEEE Code of Ethics throughout the period working on the Automatic Humidity Sensing and Water Refilling Cool-Mist Humidifier.

References

[1] Environmental Protection Agency, “Care for your air: A guide to indoor air quality,” *US EPA*, Aug. 07, 2023. https://www.epa.gov/indoor-air-quality-iaq/care-your-air-guide-indoor-air-quality (accessed Mar. 25, 2024).

[2] Sensirion, “Datasheet -SHT4x,” *Sensirion*, Aug. 2023. https://sensirion.com/media/documents/33FD6951/6555C40E/Sensirion_Datasheet_SHT4x.pdf (accessed Mar. 26, 2024).

[3] A. Sherwin, *ECE445 Team 11 - Sensor Subsystem Schematic*. 2024.

[4] A. Sherwin, *ECE445 Team 11 - Humidifier Subsystem Schematic*. 2024.

[5] Espressif, “ESP32-S3-WROOM-1 ESP32-S3-WROOM-1U Datasheet,” *Espressif*, 2023. https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf (accessed Mar. 26, 2024).

[6] IEEE, “IEEE Code of Ethics,” *IEEE Code of Ethics*, Jun. 2020. https://www.ieee.org/about/corporate/governance/p7-8.html (accessed Mar. 26, 2024).

[7] Espressif, “sch_esp32-s3-devkitc-1_v1_20210,” *Espressif*, Apr. 13, 2022. https://dl.espressif.com/dl/schematics/SCH_ESP32-S3-DevKitC-1_V1.1_20220413.pdf (accessed Mar. 26, 2024).