Your Smart Automatic Humidifier

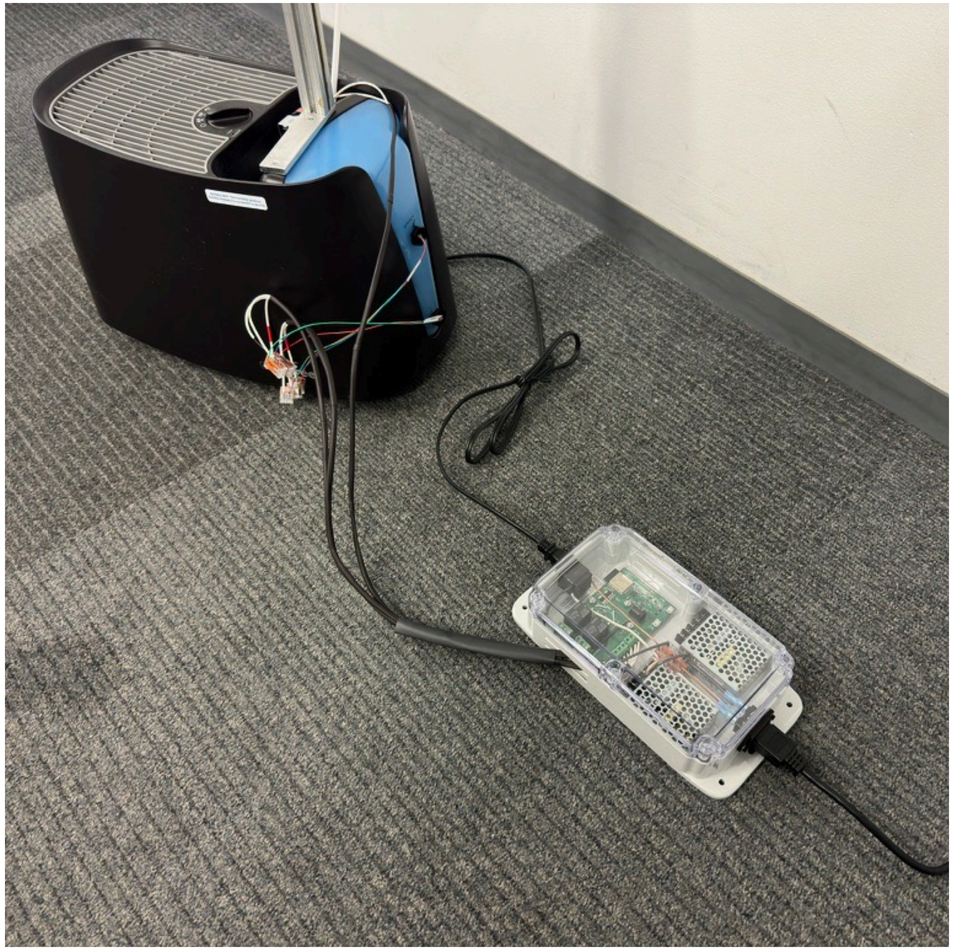
Andrew Sherwin - Woojin Kim - Jalen Chen

TA: Surya Vasanth

Professor: Dr. Jonathon Schuh

Agenda

- Problem & Solution
- Visual Aid
- High Level Requirements
- Block Diagram
- Webserver and Algorithm
- RV Table
- Project Achievements
- Future Plans



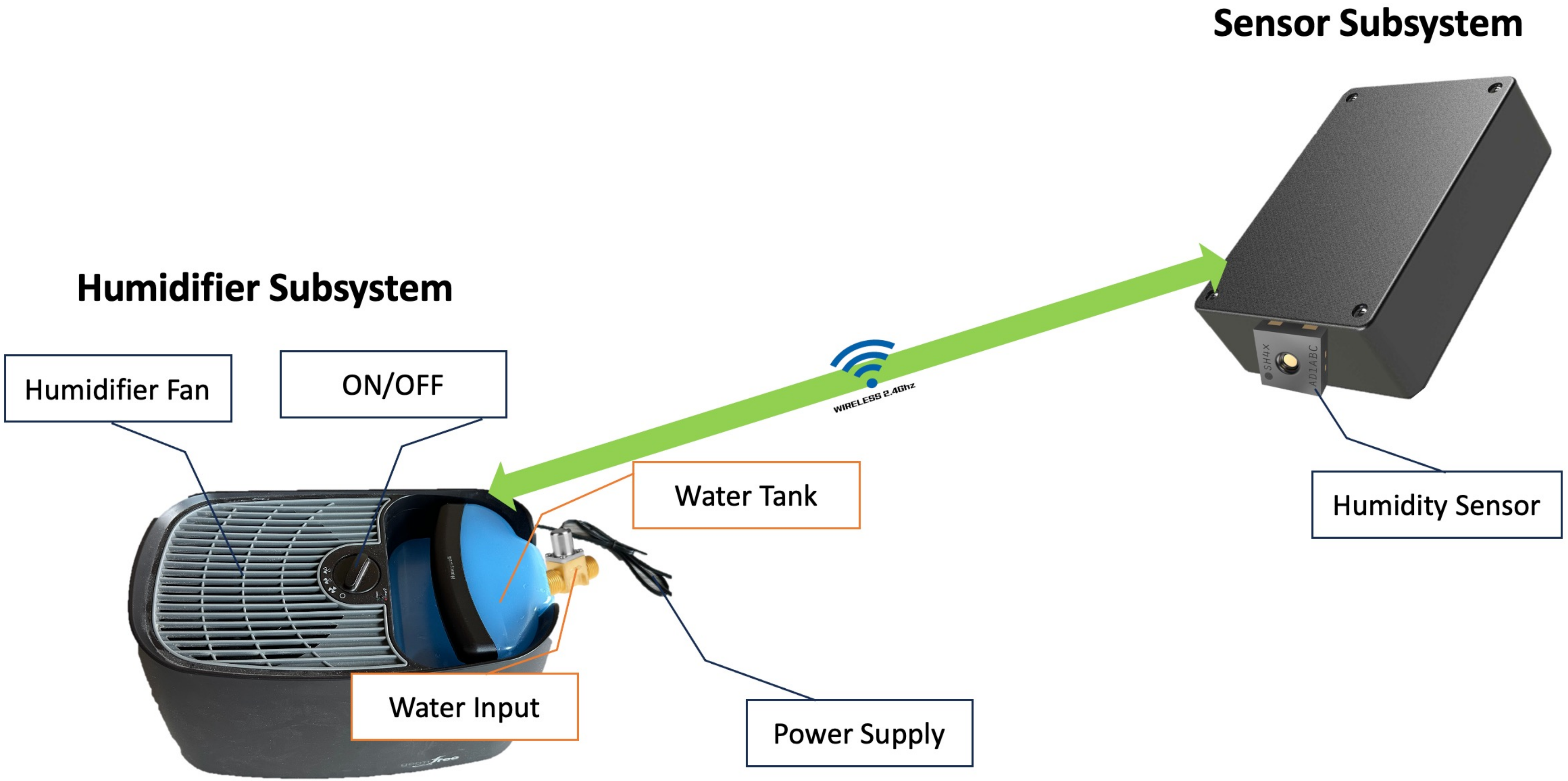
Problem

- Environmental Protection Agency highly suggests 30-50% humidity indoors
- Chemicals, gasses, pollen, mold, and other particles are reduced
- Most cheap humidifiers are bad for the user's health
    - Warm mist – cause nasal passages to swell
    - Ultrasonic – displaces minerals and bacteria
- Manual humidifiers can over/under humidity
- Humidifiers on market need manual water refilling

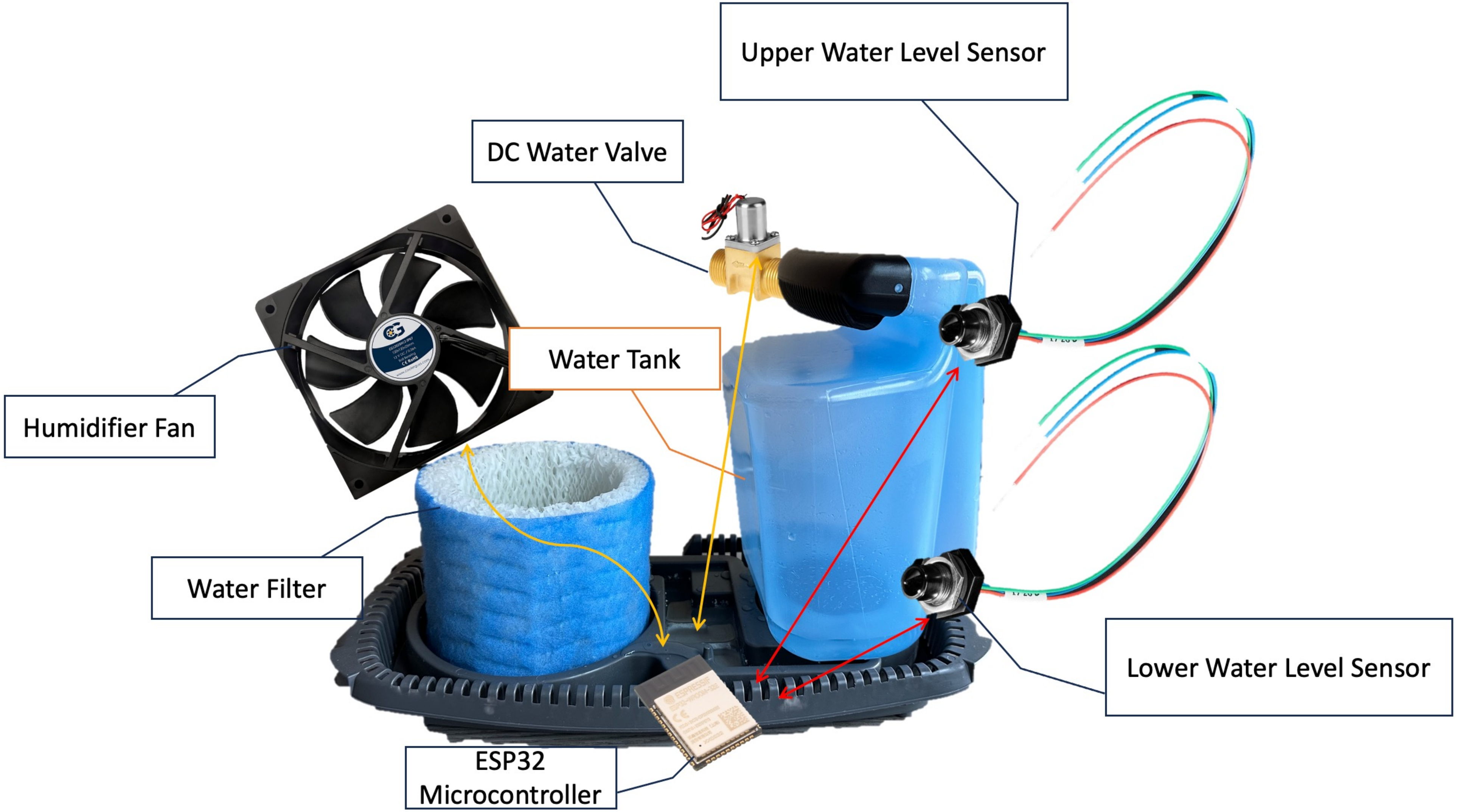
Solution

- Cool mist humidifier
  - Good for health
  - Evaporates water off filter
- Automatic on/off humidifier based on humidity
- Remote humidity sensing via 3 sensors
- Automatic water refill via water valve with water level sensing

Visual Aid


Sensor Subsystem
Humidifier Subsystem
Humidifier Fan
ON/OFF
WIRELESS 2.4Ghz
Water Tank
Humidity Sensor
Water Input
Power Supply

Visual Aid - Component



High Level 1

- Humidifier's ESP32 communication with three different sensors via 2.4GHz Wi-Fi
- Average humidity data from three sensors
- Create temperature dependent humidity calculation algorithm

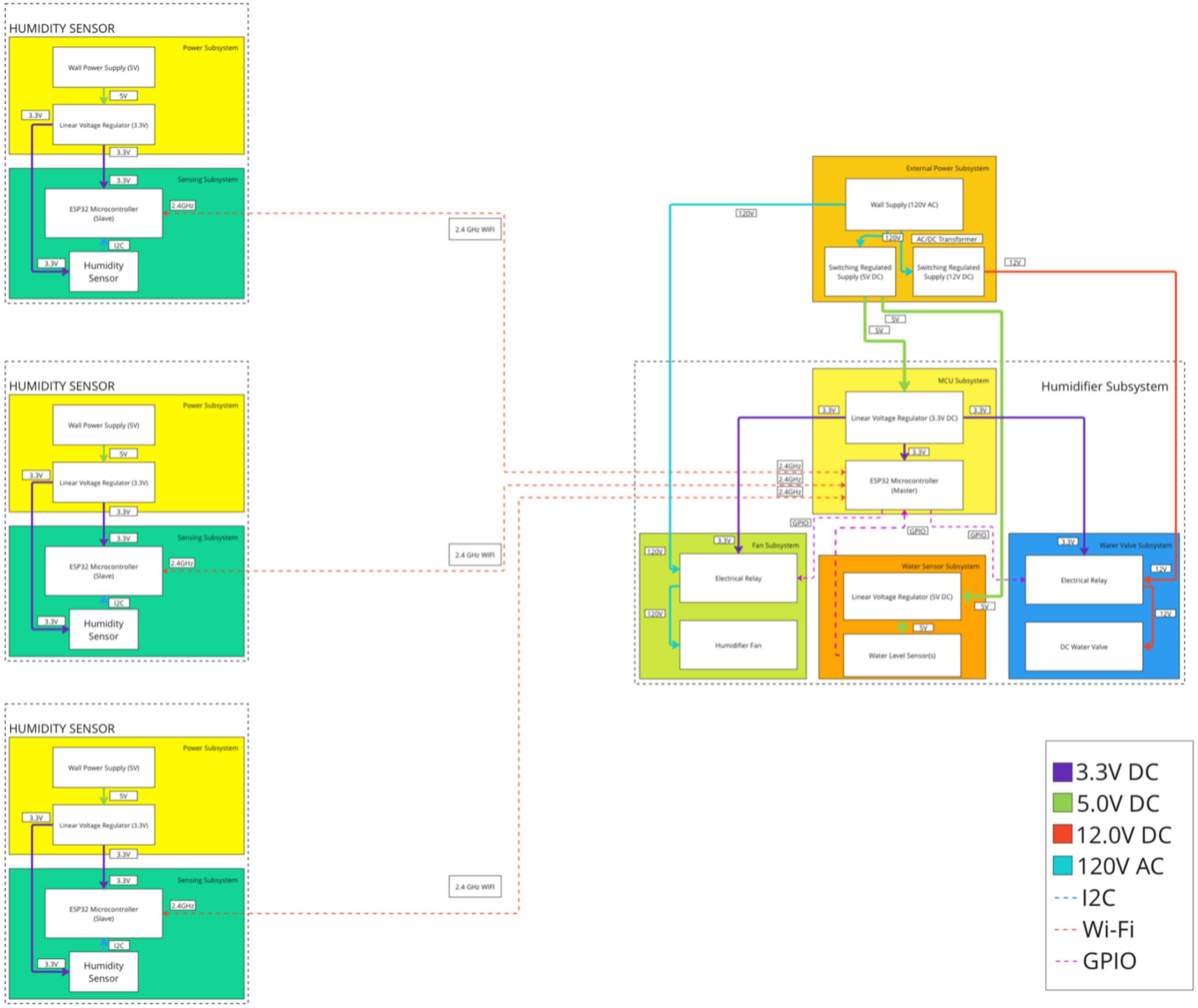
High Level 2

- Open/close valve to refill the water tank
- Two water level sensors GPIO input to ESP32
- Max capacity of the water tank: 4.16 liters
- Filter absorbs around 150 milliliters ± 5 milliliters
- Evaporation rate approximately 173 milliliters per hour (in 40% humidity room)

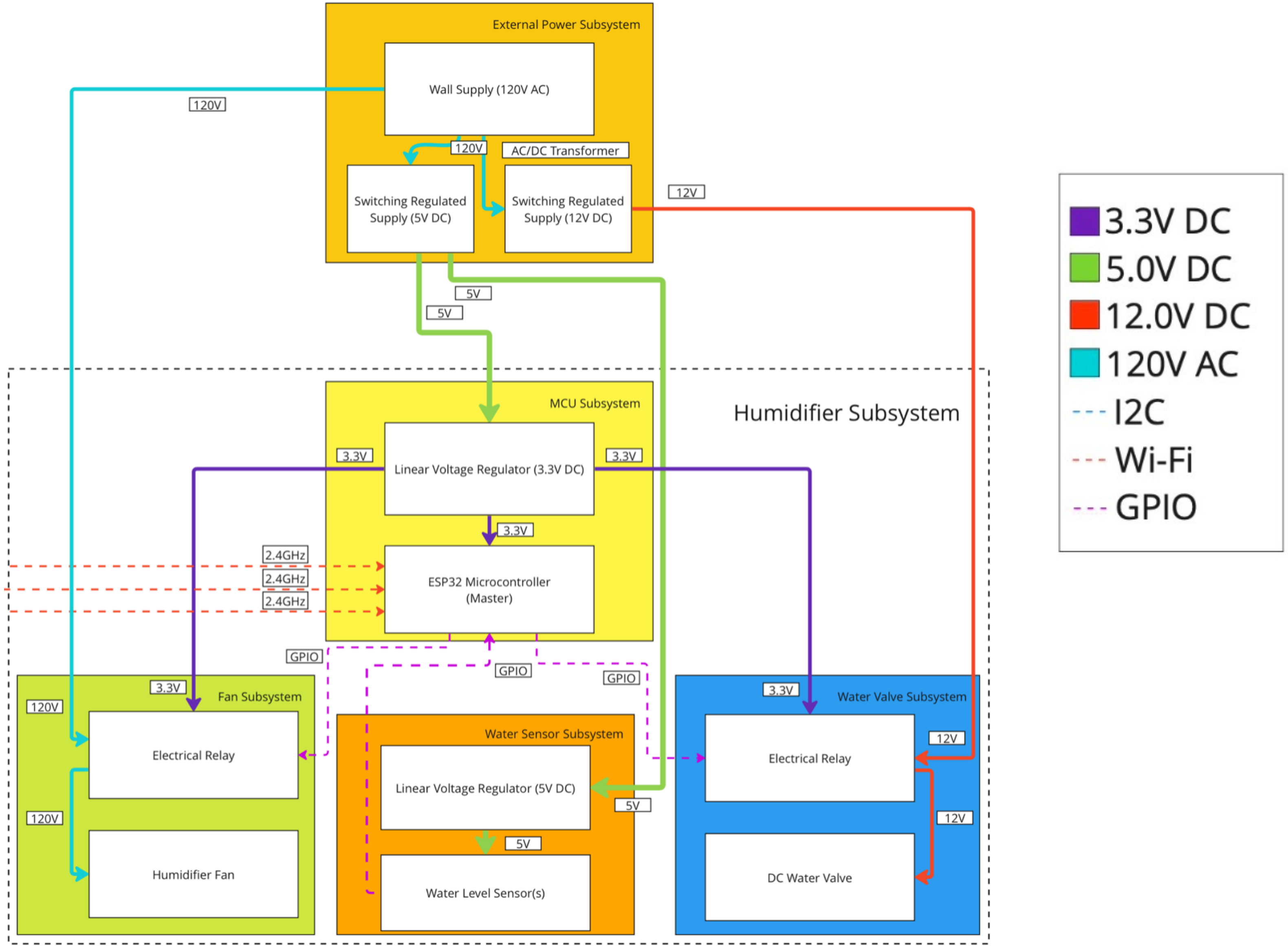
High Level 3

- On/off fan based on pre-configured ideal humidity range, 35% and 50% humidity, based on an algorithm averaging the readings from 3 sensors
- Actuate relay with LED output for ON signal
- Theoretical estimation of 44ns response time to on/off the humidifier’s fan (w/o runtime)
- High voltage (120V AC) components does not interfere with low voltage components (< 12V DC)

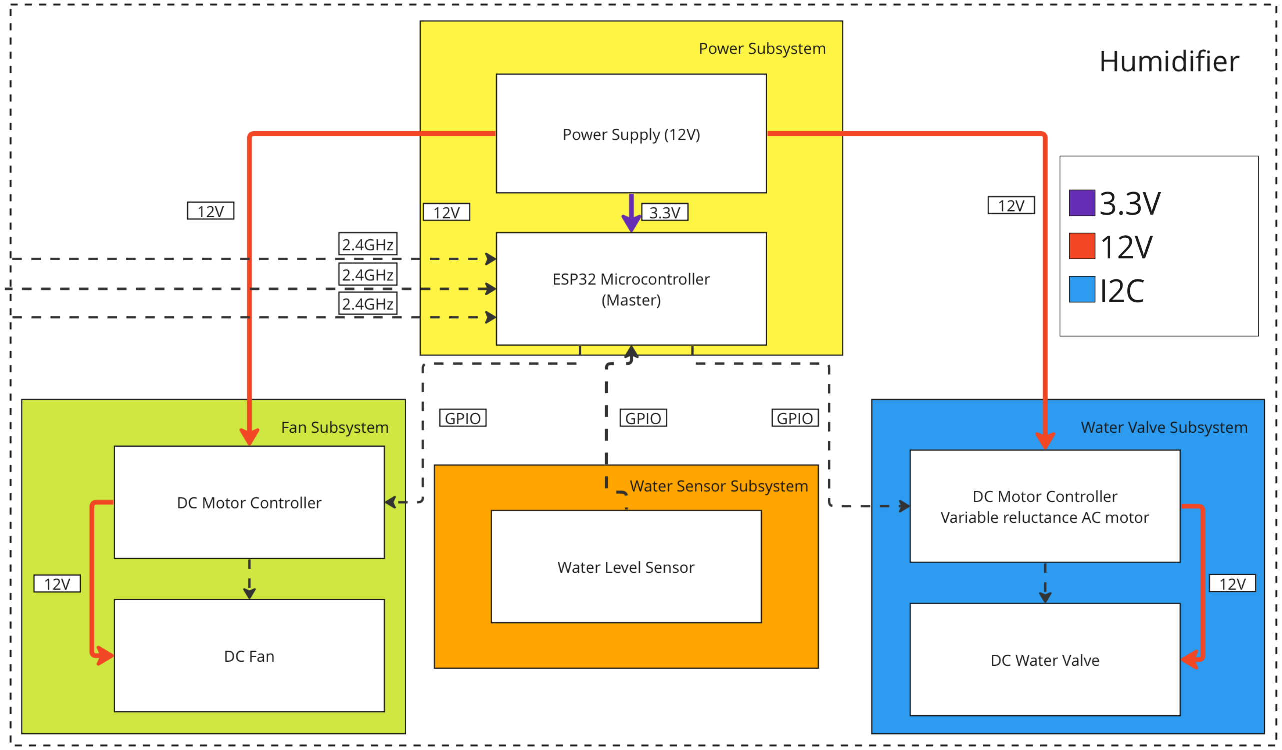
Block Diagram



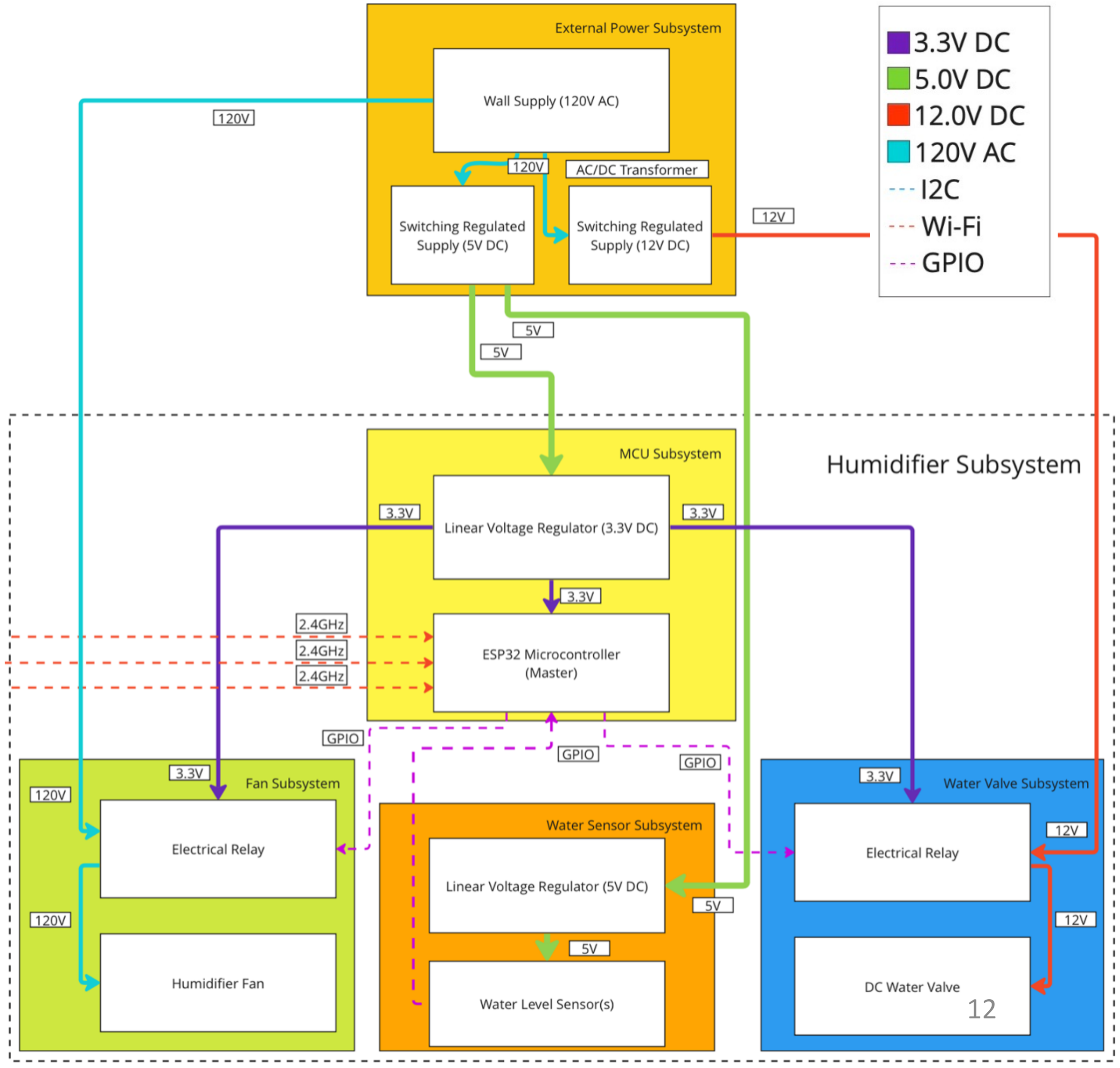
Block Diagram - Humidifier



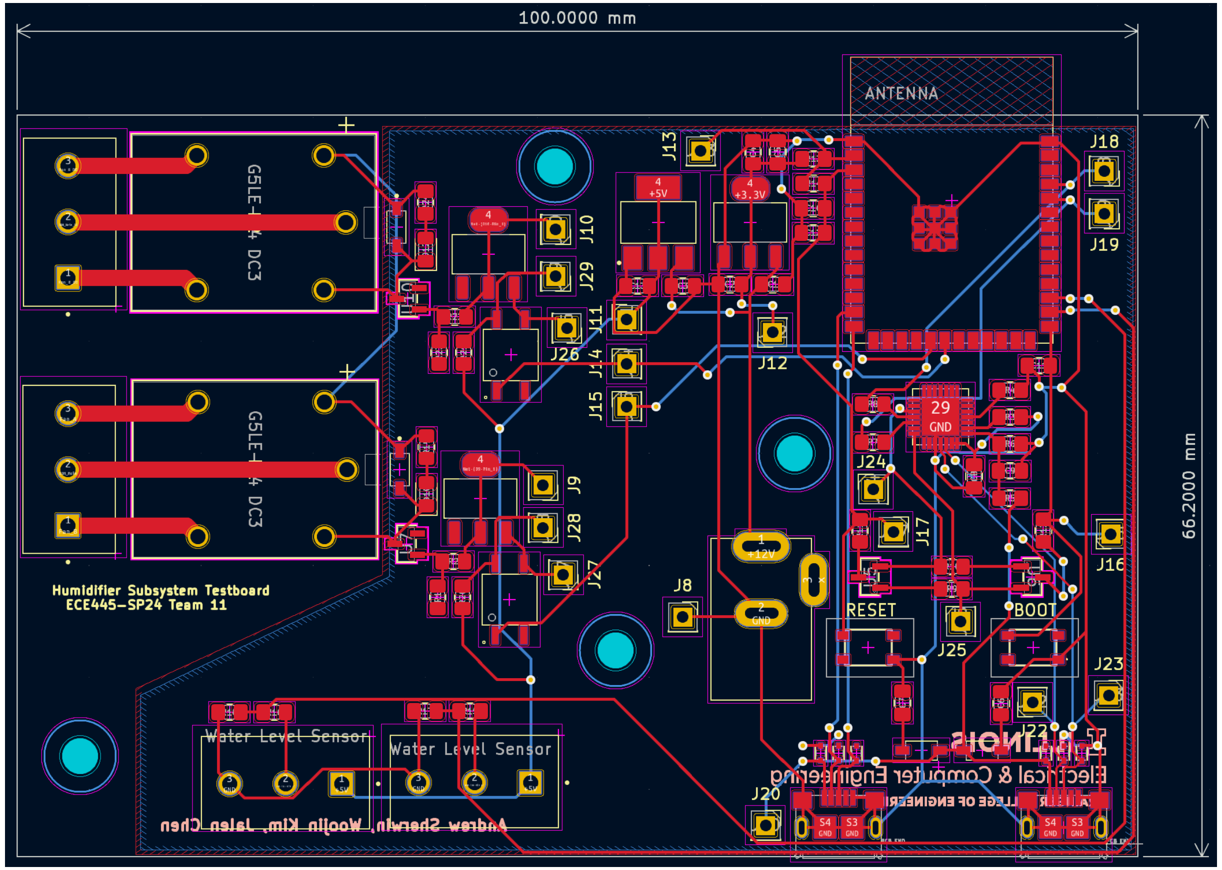
Current vs Original Humidifier Subsystem



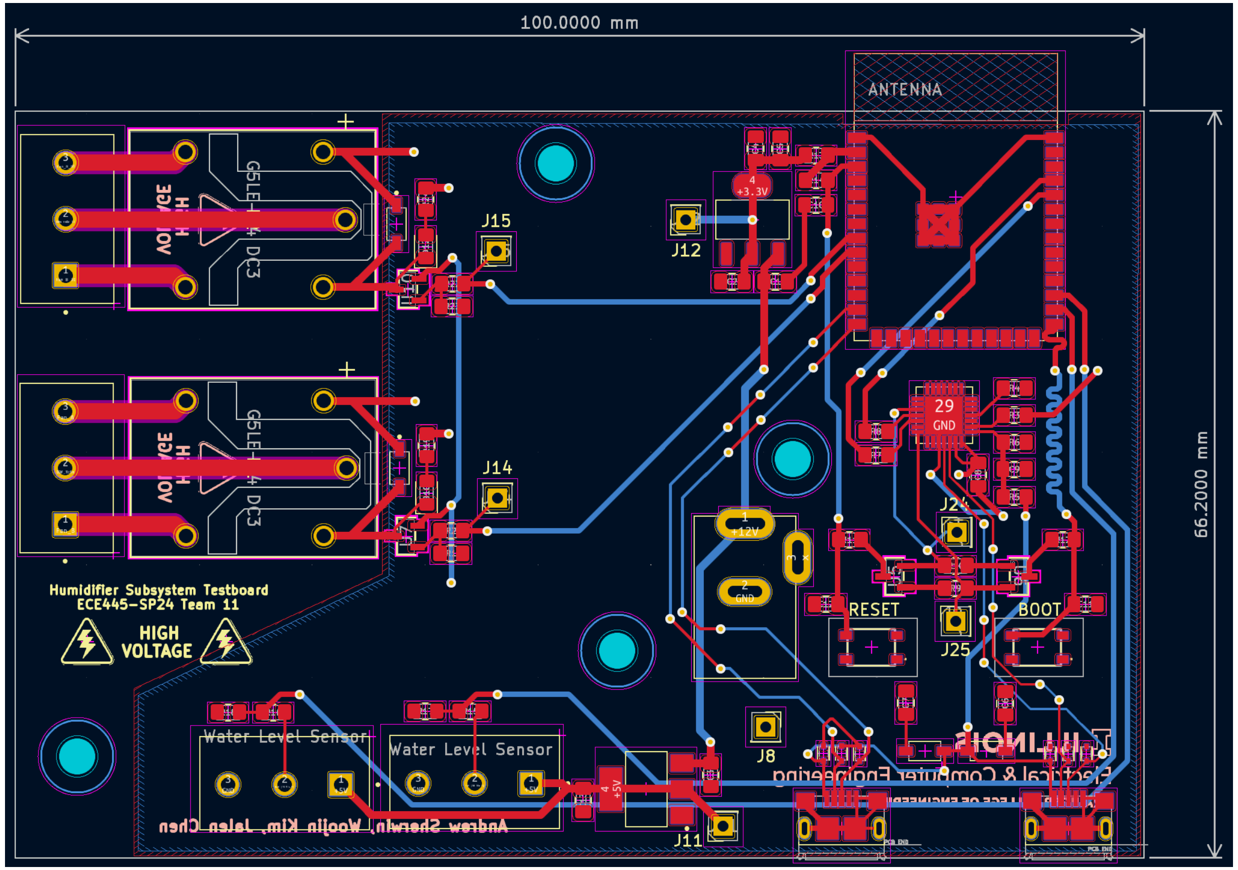
Original Block Diagram – Project Proposal



Humidifier Subsystem PCBs

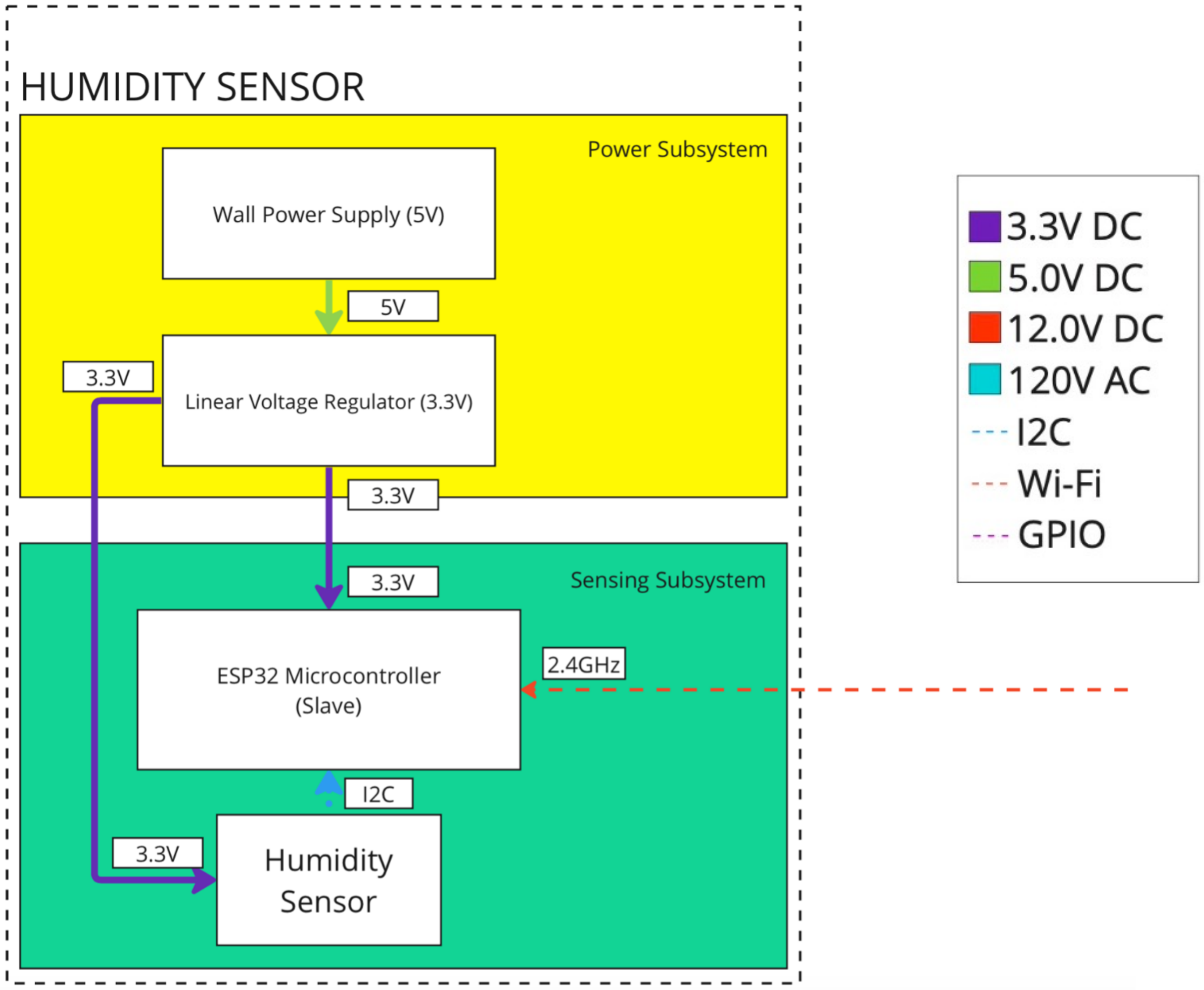


Humidifier PCB Rev0

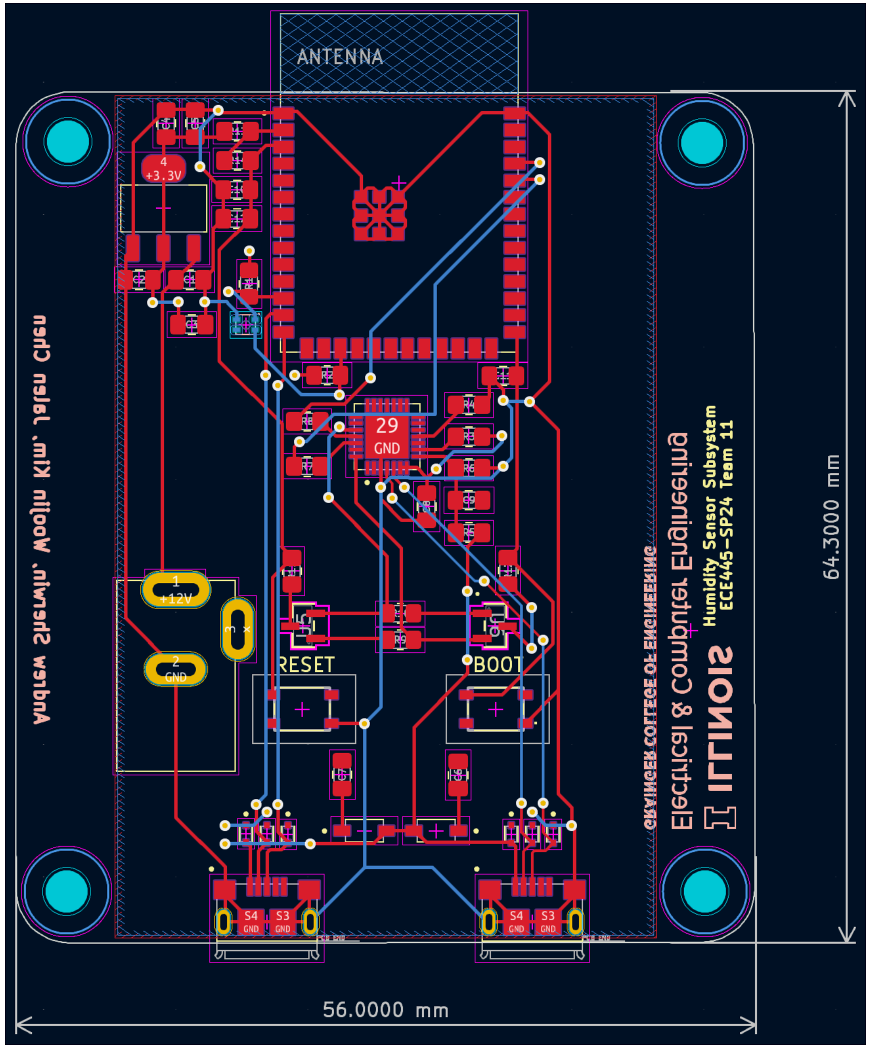


Humidifier PCB Rev4

Block Diagram - Sensor

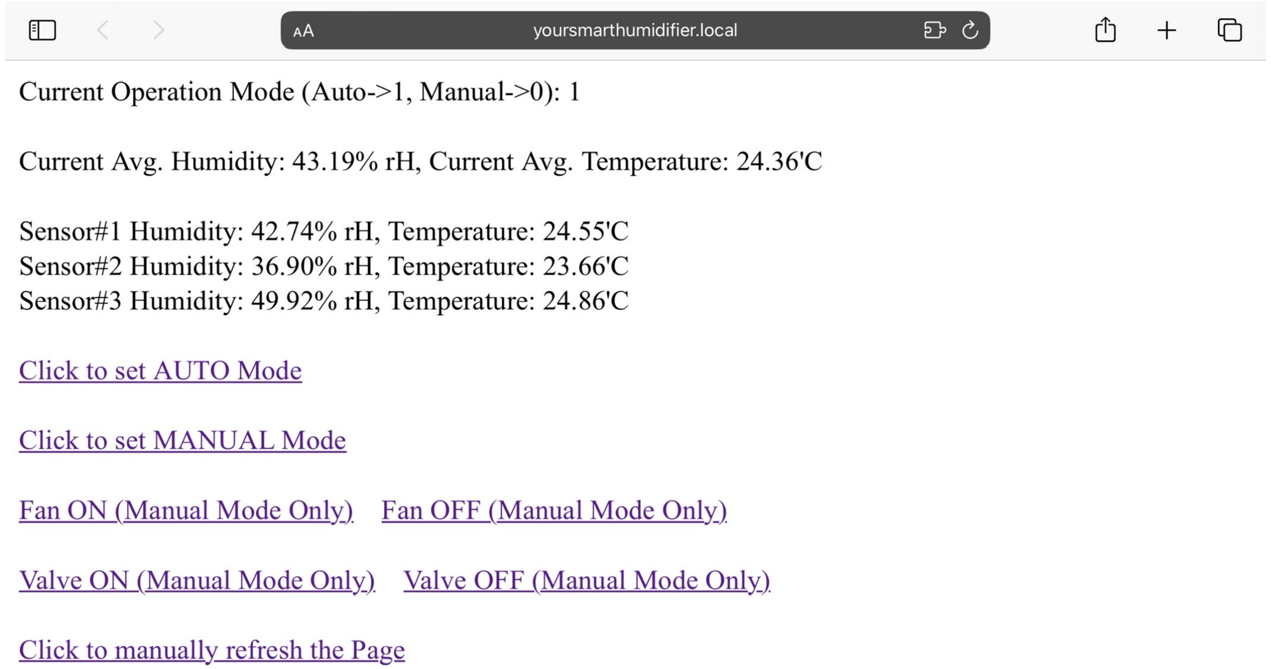


Sensor Subsystem PCBs



Humidifier PCB Rev0

Web Server & Algorithm



- WiFi access with mDNS

- Auto/Manual Mode

(Fan/Valve off when Auto->Manual)

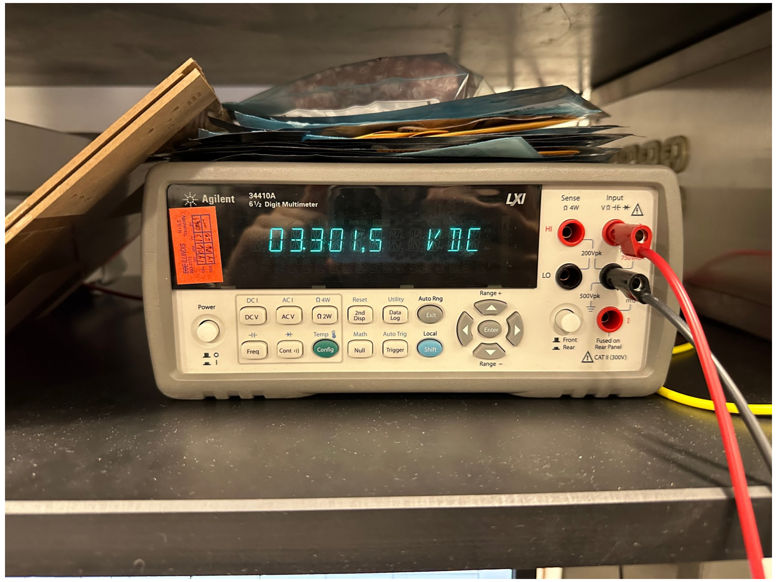
- Signal Loss Detection
- Refreshes every 2s

- Warmer air can handle more moisture
- -> new Humidity target based on average temperature

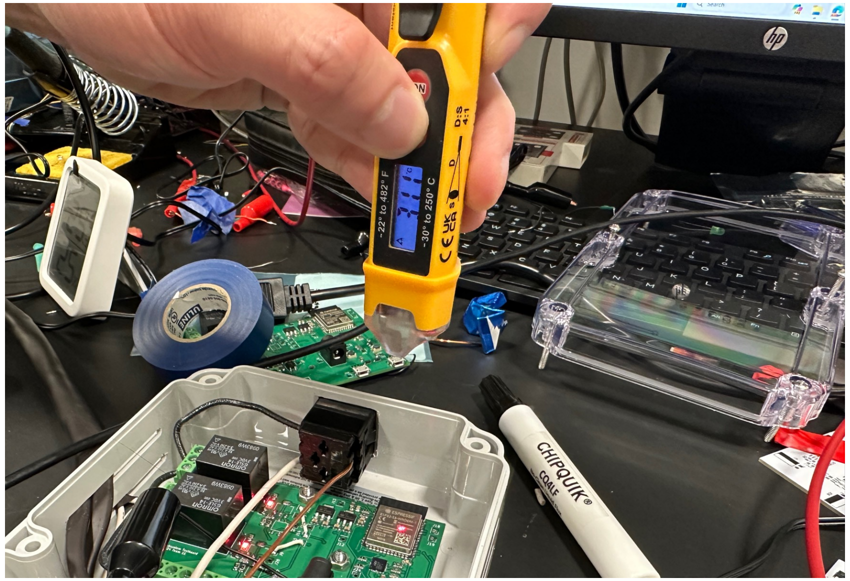
(if avg. Temp < 20'C -> 35~42.5% Rh)
(if avg. Temp > 22'C -> 42.5~50% Rh)
(else, 35~50% Rh)

RV Table – Central Humidifier Subsystem (1)

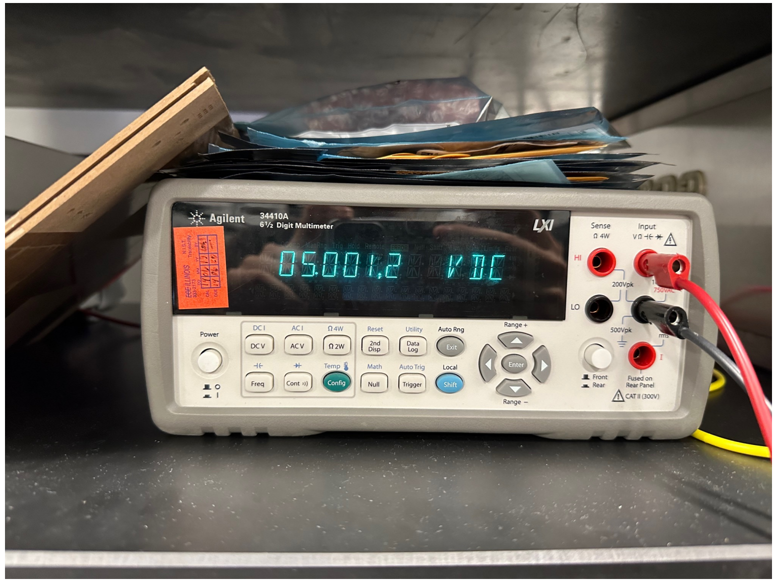
Requirements	Verification
• **Provide 3.3V +/- 0.5% from an input voltage of 5V +/- 0.1% DC source** • **Thermal stability maintains below 120'C**	• Use multimeter to test and take average voltage output over 1-hour to test stability • Monitor heat dissipation with thermal laser gun over an average of 1-hour use time, expected +/- 5'C under load



3.3V linear regulator output



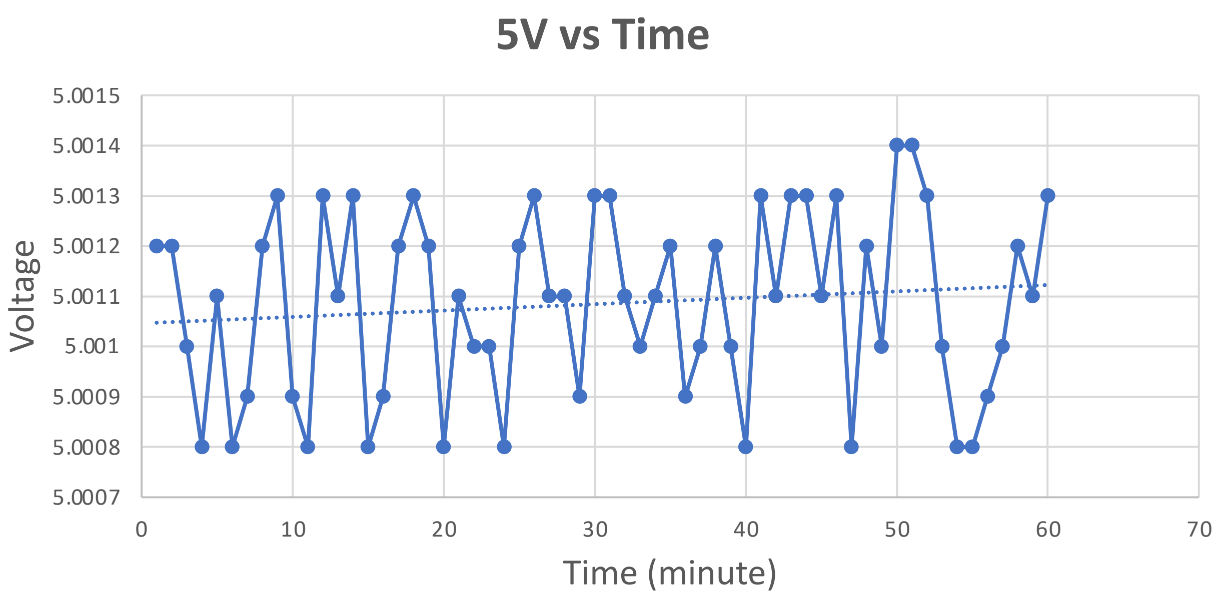
Temperature reading with laser

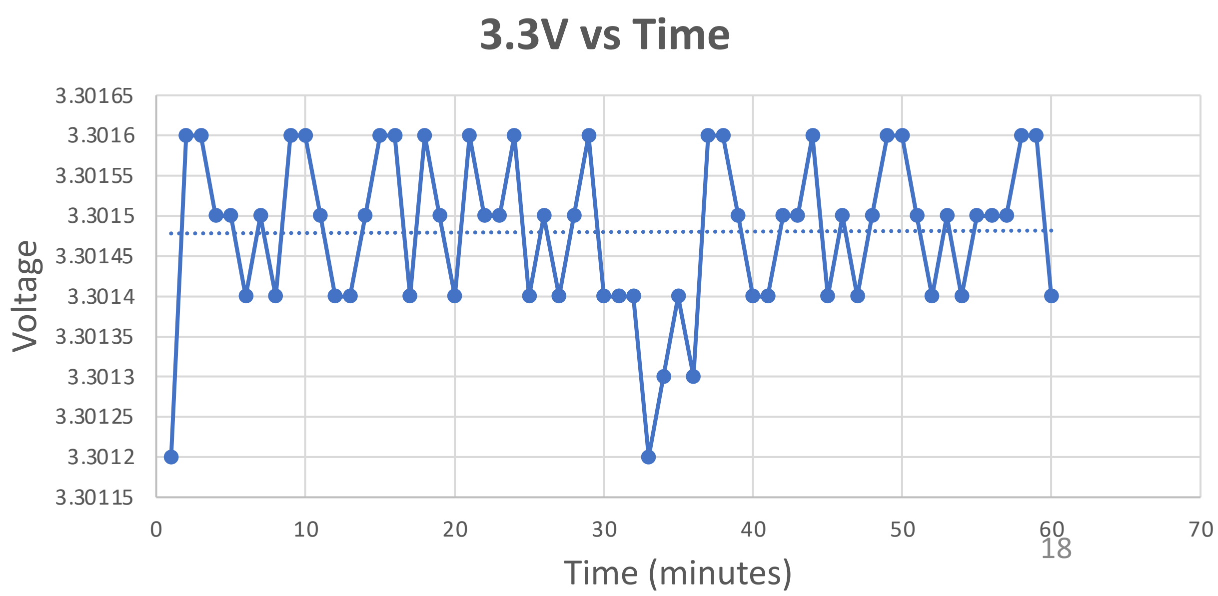


5V linear regulator output

RV Table – Central Humidifier Subsystem (1)

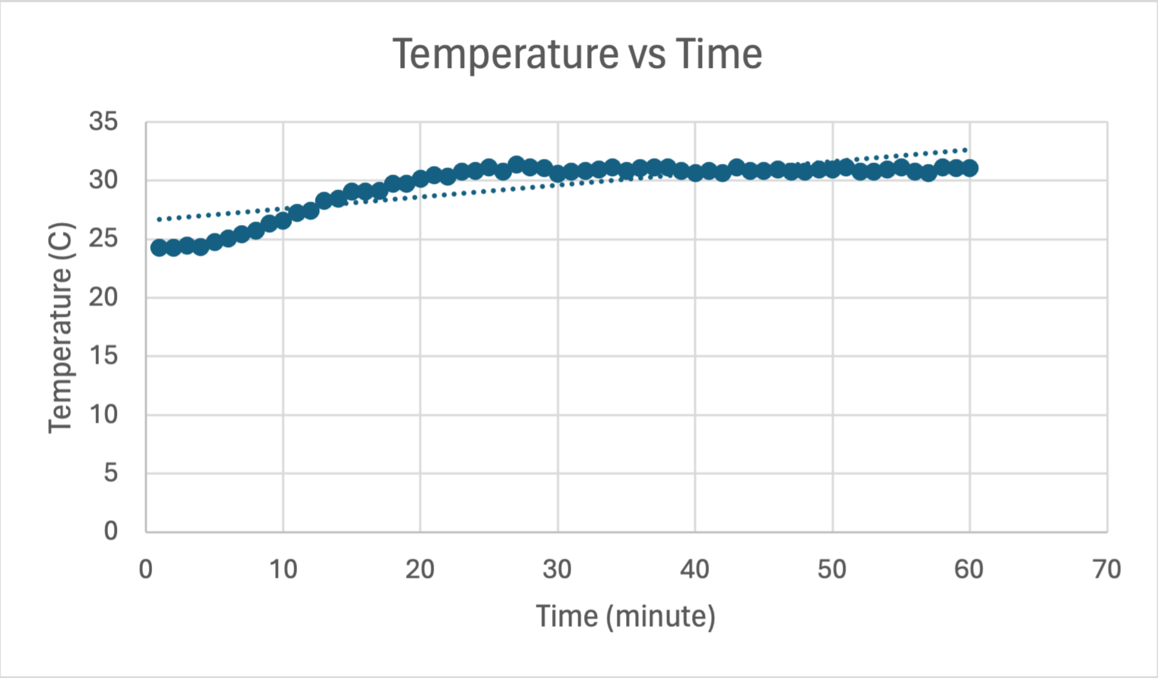
Requirements	Verification
• **Provide 3.3V +/- 0.5% from an input voltage of 5V +/- 0.1% DC source** • **Thermal stability maintains below 120'C**	• Use multimeter to test and take average voltage output over 1-hour to test stability • Monitor heat dissipation with thermal laser gun over an average of 1-hour use time, expected +/- 5'C under load





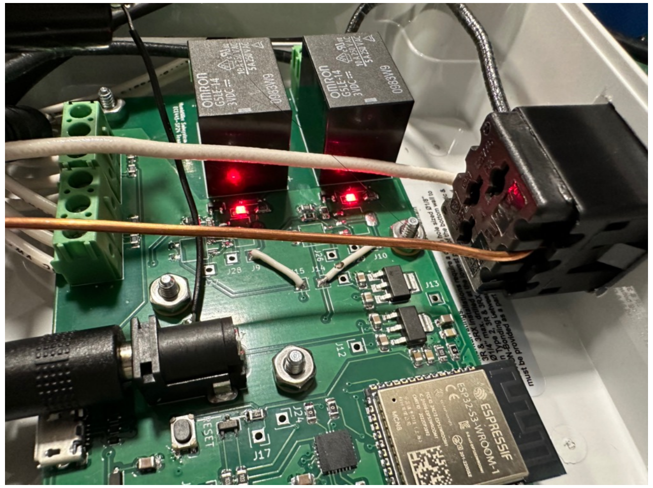
RV Table – Central Humidifier Subsystem (1)

Requirements	Verification
• **Provide 3.3V +/- 0.5% from an input voltage of 5V +/- 0.1% DC source** • **Thermal stability maintains below 120'C**	• Use multimeter to test and take average voltage output over 1-hour to test stability • Monitor heat dissipation with thermal laser gun over an average of 1-hour use time, expected +/- 5'C under load

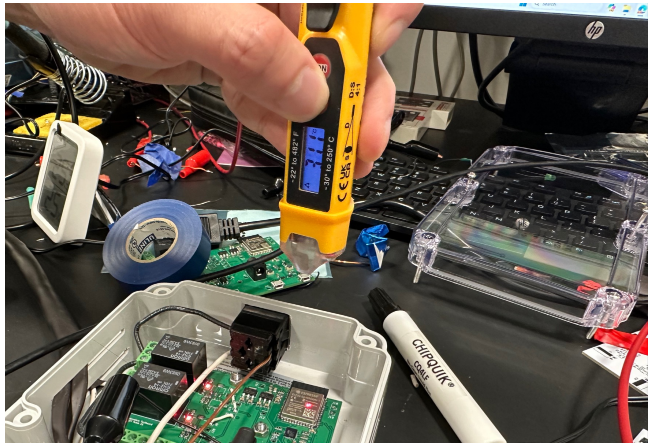


RV Table – Central Humidifier Subsystem (2)

Requirements	Verification
• **ESP32 communication with remote sensor ESP32 utilizing 2.4GHz Wi-Fi**	• Verify functionality of ESP32 Wi-Fi component by creating an access point, and accessing access point to remotely turn on LED lights • Verify communication between two ESP32 boards by sending command to turn on LED light from one ESP32 board • Verify thermal performance of ESP32 chip during operation by probing with a laser thermal gun



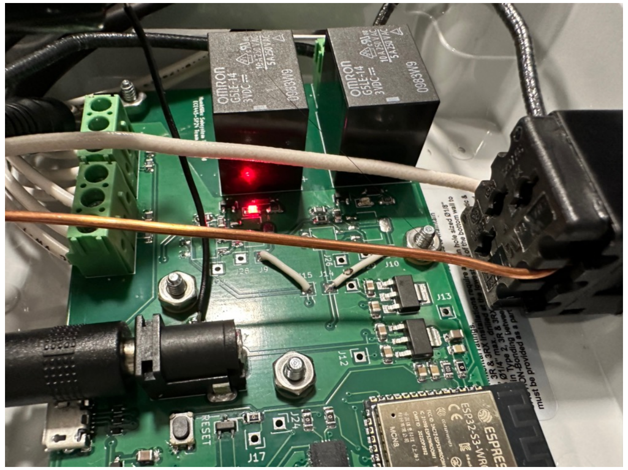
LED light control example



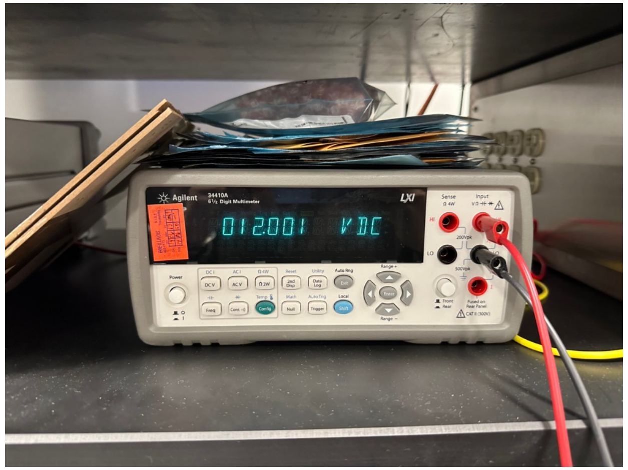
Temperature reading example

RV Table – Central Humidifier Subsystem (3)

Requirements	Verification
• **ESP32 control of activation and deactivation of water valve** • **ESP32 control of activation and deactivation of fan**	• Verify functionality of ESP32 communication with DC relay by sending activation signal to water valve/fan DC relay controller, and when signal is received, DC relay controllers output a lit LED light • Probe the output voltage of the DC relay controllers to make sure it is 12V +/- 0.5%



LED light when valve is on

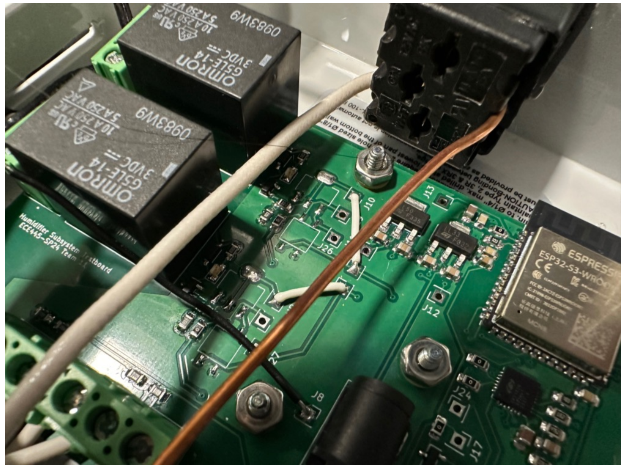


12V output to water valve

RV Table – Central Humidifier Subsystem (4)

Requirements	Verification
• **ESP32 detects signals from two water-level sensors**	• Contact/No Contact water to the water-level sensors and check through Arduino IDE to see if ESP32 receives contact/no contact signal from the two water-level sensors • Water level sensor low signal when in contact with water, and high signal when not in contact with water

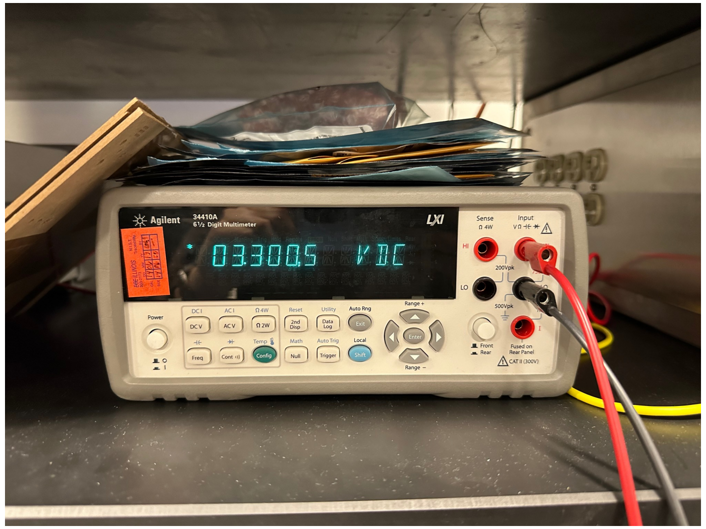
Water level sensor contact water



Water-valve off when water detected

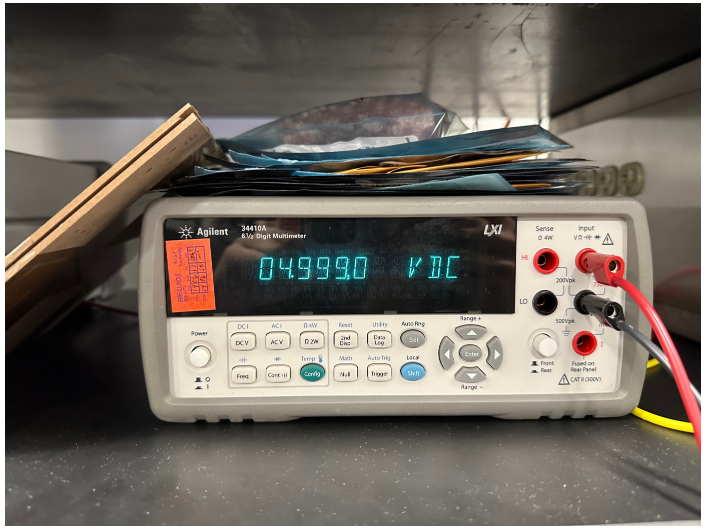
RV Table – Humidity Sensor Subsystem (1)

Requirements	Verification
• **Provide 3.3V +/- 0.5% from an input voltage of 5V +/- 0.1% DC source** • **Thermal stability maintains below 120’C**	• Use multimeter to test and take average voltage output over 1-hour to test stability • Monitor heat dissipation with thermal laser gun over an average of 1-hour use time, expected +/- 5’C under load



3.3V linear regulator output

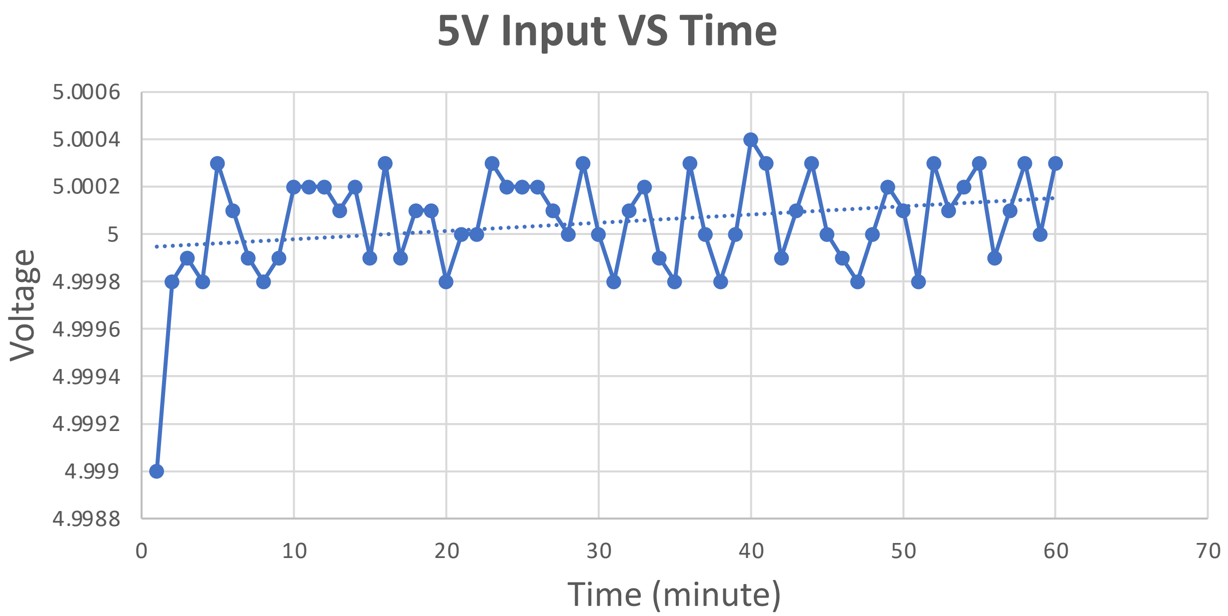

Temperature reading with laser

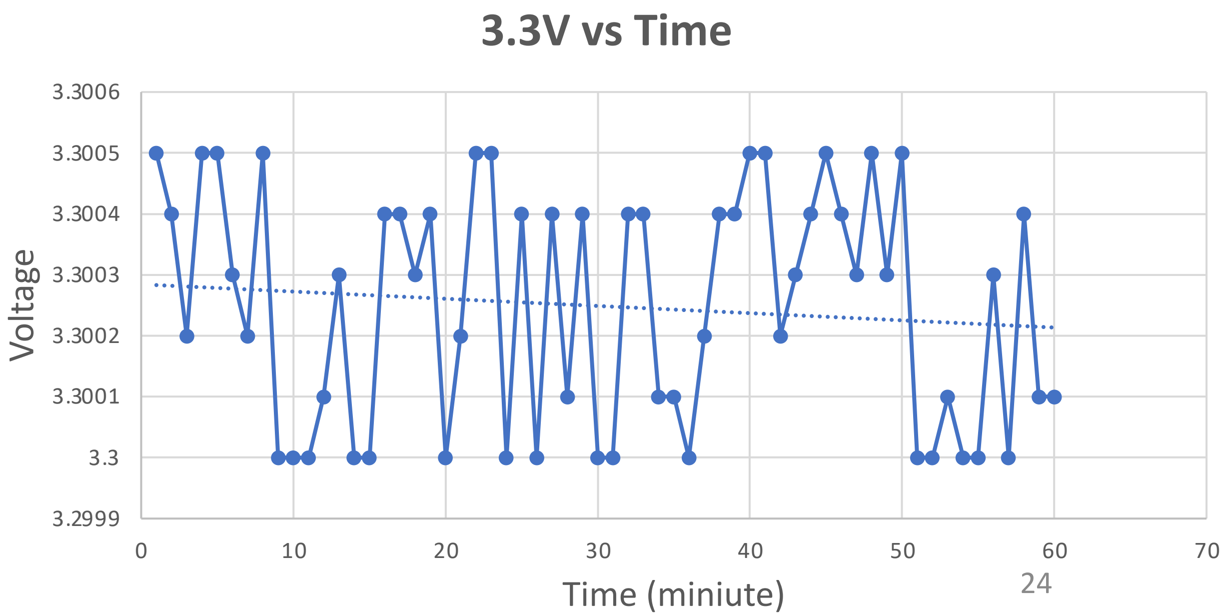


5V linear regulator output

RV Table – Humidity Sensor Subsystem (1)

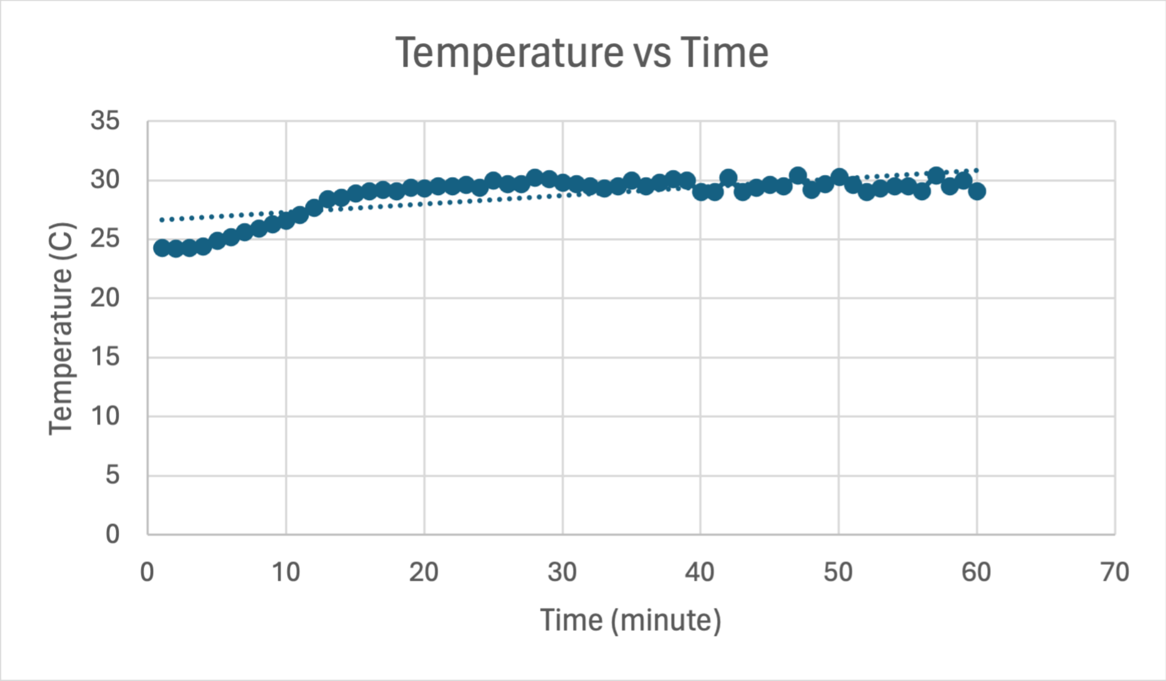
Requirements	Verification
• **Provide 3.3V +/- 0.5% from an input voltage of 5V +/- 0.1% DC source** • **Thermal stability maintains below 120'C**	• Use multimeter to test and take average voltage output over 1-hour to test stability • Monitor heat dissipation with thermal laser gun over an average of 1-hour use time, expected +/- 5'C under load





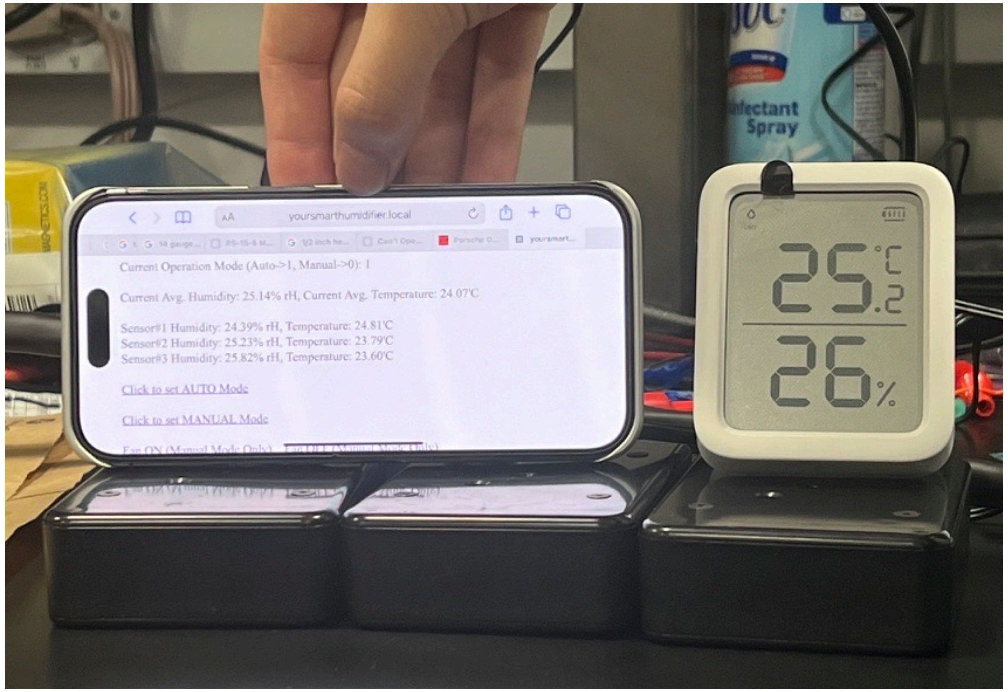
RV Table – Humidity Sensor Subsystem (1)

Requirements	Verification
• **Provide 3.3V +/- 0.5% from an input voltage of 5V +/- 0.1% DC source** • **Thermal stability maintains below 120'C**	• Use multimeter to test and take average voltage output over 1-hour to test stability • Monitor heat dissipation with thermal laser gun over an average of 1-hour use time, expected +/- 5'C under load



RV Table – Humidity Sensor Subsystem (2)

Requirements	Verification
• **ESP32 communication with humidity sensor receiving humidity data at least once in a minute** • **Humidity sensor measuring humidity data at least between 25% - 55%**	• Verify communication between ESP and humidity sensor Arduino IDE by checking humidity data ESP32 receives • Compare the humidity data received by ESP32 to the commercial humidity sensor and check if the humidity is within +/- 3% • Make the air dry/moist to see if the sensor can measure humidity between 25% - 55% • Exhale on sensor to see humidity levels are raised • Compare humidity reading with reference humidity reader



Commercial humidity and temperature sensor verification

Project Achievements and Successes

- Sensor Subsystem reads temperature and humidity data from SHT45
- Sensor Subsystem communicates with Humidifier Subsystem with Wi-Fi
- Humidifier Subsystem successfully activates humidifier
- Humidifier Subsystem successfully reads from water level sensors
- Humidifier Subsystem successfully activates water valve

- Bonus:
  - User Interface through access point to control Humidifier Subsystem
  - Reliability tested for 4 weeks, running 24/7

Challenges

- No failures on current design
- Hardware
  - Initial 12V input to PCB was too high for Linear Voltage Regulator
  - Relay design
    - Isolation, creepage, clearance
  - Optocoupler
  - SHT45 location and isolation
  - Modular overall design
- Software
  - Concurrent multi-node communication

Conclusion and Future Plans

- Integrate with Amazon Alexa, Apple HomeKit, and Google Assistant
- Change linear voltage regulator to buck converter
- Further compact PCB
  - Remove ESP32 programming components
- Create original AC/DC Converter if budget allows
- Improve PCB enclosure design
- Improve UI design
- Potentially remove water level sensor and water valve
  - Making product modular for all manual humidifiers

"Students learn the most then they struggle with challenges" - Minh Do

Your Oasis Awaits: Refresh, Rejuvenate, Breathe

References

[1] F. Hecht, “Wifi Propagation,” *freedom*, 2017. https://doc.freefem.org/tutorials/wifiPropagation.html (accessed Mar. 29, 2024).
[2] M. Marwell, “Issues with the I^2C (Inter-IC) Bus and How to Solve Them,” *DigiKey*, Aug. 09, 2018. Accessed: Mar. 29, 2024. [Online]. Available: https://www.digikey.com/en/articles/issues-with-the-i2c-bus-and-how-to-solve-them
[3] MetaGeek, “Wi-Fi and Non Wi-Fi Interference,” *MetaGeek*, 2024. https://www.metageek.com/training/resources/wifi-and-non-wifi-interference/ (accessed Mar. 29, 2024).
[4] A. V. Arundel, E. M. Sterling, J. H. Biggin, and T. D. Sterling, “Indirect health effects of relative humidity in indoor environments.,” *Environmental Health Perspectives*, vol. 65, no. 65, pp. 351–361, Mar. 1986, doi: 10.1289/ehp.8665351.
[5] V. Perrone *et al.*, “The Epidemiology, Treatment Patterns and Economic Burden of Different Phenotypes of Multiple Sclerosis in Italy: Relapsing-Remitting Multiple Sclerosis and Secondary Progressive Multiple Sclerosis,” *Clinical Epidemiology*, vol. Volume 14, no. 14, pp. 1327–1337, Nov. 2022, doi: 10.2147/clep.s376005.
[6] IEEE, “IEEE Code of Ethics,” *IEEE Code of Ethics*, 2020. https://www.ieee.org/about/corporate/governance/p7-8.html (accessed Mar. 29, 2024).
[7] University of Illinois, “Salary Averages,” *UIUC*, 2022. https://ece.illinois.edu/admissions/why-ece/salary-averages (accessed Mar. 29, 2024).
[8] 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).
[9] Sensirion, “Datasheet -SHT4x,” *Sensirion*, Aug. 2023. https://sensirion.com/media/documents/33FD6951/6555C40E/Sensirion_Datasheet_SHT4x.pdf
[10] A. Sherwin, *ECE445 Team 11 - Sensor Subsystem Schematic*. 2024.
[11] A. Sherwin, *ECE445 Team 11 - Humidifier Subsystem Schematic*. 2024.
[12] 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
[13] 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).
[14] 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).
[15] A. Sherwin, *Sensor Subsystem Block Diagram*. 2024.
[16] A. Sherwin, Humidity Subsystem Block Diagram. 2024.
[17] D. Workshop, “ESP NOW - Peer to Peer ESP32 Communications,” DroneBot Workshop, Apr. 03, 2022. https://dronebotworkshop.com/esp-now/
[18] K. Rembor, “Adafruit Sensirion SHT40, SHT41 & SHT45 Temperature & Humidity Sensors,” Adafruit Learning System, Feb. 04, 2021. https://learn.adafruit.com/adafruit-sht40-temperature-humidity-sensor/arduino

References (continued)

[19] cplusplus, “std::chrono::high_resolution_clock::now,” cplusplus. https://cplusplus.com/reference/chrono/high_resolution_clock/now/ (accessed Apr. 20, 2024).

[20] cplusplus, “std::chrono::nanoseconds,” cplusplus. https://cplusplus.com/reference/chrono/nanoseconds/ (accessed Apr. 20, 2024).

[21] me-no-dev, “ESPAsyncWebServer/README.md at master · me-no-dev/ESPAsyncWebServer,” GitHub. https://github.com/me-no-dev/ESPAsyncWebServer/blob/master/README.md

[22] Tea, “ESP32 Web Server periodic updating problem,” Stack Overflow. https://stackoverflow.com/questions/64610221/esp32-web-server-periodic-updating-problem

[23] antepher, “ESP32 Arduino: HTTP server over soft AP,” techtutorialsx, Jan. 07, 2018. https://techtutorialsx.com/2018/01/07/esp32-arduino-http-server-over-soft-ap/

[24] ESPRESSIF, “ESP-NOW - ESP32 - — ESP-IDF Programming Guide v5.2.1 documentation,” espressif.com. https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_now.html