ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

Scrubbing CO2 Operational Prototype

Team #32

SEUNGHWAN HONG
(sh34@illinois.edu)
ALAN CARDIEL
(acard6@illinois.edu)
KINJAL DEY (kinjald2@illinois.edu)

Prof: Arne Fliflet
TA: David Null

September 20, 2023

1 Introduction

1.1 Problem Overview

The Sahara desert is the only region in the world where water insecurity amongst the local communities is steadily increasing. This region houses 15 of the world's 20 most water insecure countries. In addition, the Sahara desert has grown 10% in size since the 1920s, and continues to grow at an unprecedented rate due to global warming [1]. Access to clean drinking water is a basic human right, and this is a problem that must address as soon as possible.

Professor Jont Allen, of the ECE department at UIUC, has proposed a system to harness the power of the sun to desalinate saltwater and turn it into clean, drinking water. The system Professor Allen proposes involves pumping saltwater from the ocean into a tank in the desert, where the heat from the sun evaporates water, separating the salt. Then the resulting water vapor is lead to a colder tank, where it condenses and the resulting clean water is collected [2]. Currently Professor Allen's system is just a prototype that's still in the works. The issue is that Allen has a variety of ideas for modifications be made since his significant breakthrough of over 50% yield from salt water. However, since this is a prototype components are all separate and if further modifications were made without a more user friendly approach to run, his entire project could suffer from inefficiency.

Amongst the current list of issues that professor Allen is plagued by in his current prototype is that all of the components in the system are separated which can cause problems when setting up experiments by forgetting about components. Secondly there are some components that have to be set manually either by turning them on or by setting a value to them, this can also cause issues when trying to do experiments. Additionally there isn't an easy method for streamlining the data collected from experiments and storing it properly. Lastly is the efficiency of the system can be improved upon, but the lack of easy variable control limits this factor to be more difficult than it currently is.

1.2 Solution

We will work along side Professor Jont Allen in his research to develop and improve upon current issues in his prototype. Helping him be able to run experiments more easily and concise along with data collection will allow him to further drive his research for potential upscale. This includes creating a control system to access and control the lab setup more easily for use in testing and experimenting. We will also be measuring the status of the system with the help of sensors to be able to understand, predict, and act on that status to ensure safety and efficiency in the system.

To address the current issues in his current prototype of organization we plan to consolidate and house all the components into a single place. This will allow for a more centralized way to interact with each component, from sensors to heater, and radiators. As for data collection and testing we plan to set up an algorithm that will take time and a flow rate as inputs to be able to autonomously run the lab setup with redundancies for faults, failure, safety, and termination so that each individual component does not need to be set or initialized by the user. As for the visualization we will process and col let the data on a UI to be able to make the collected data readable, presentable, and usable for the user to digest and understand.

1.3 Visual Aid

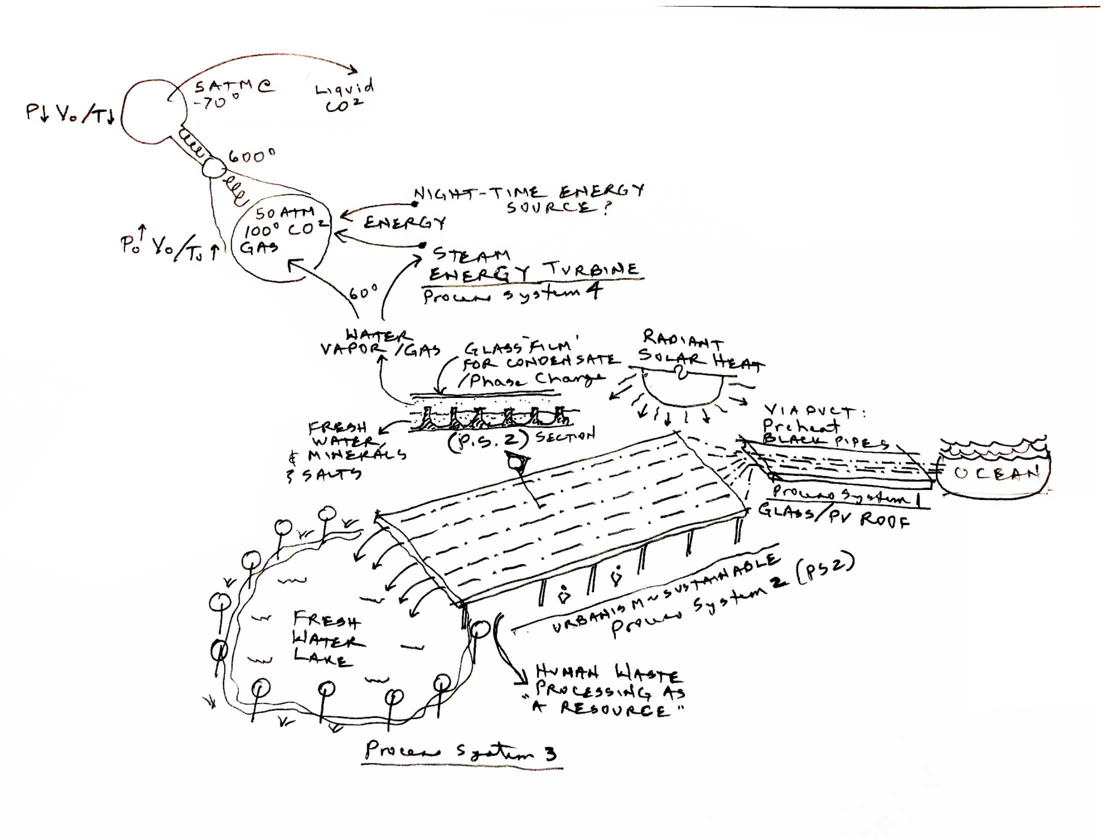
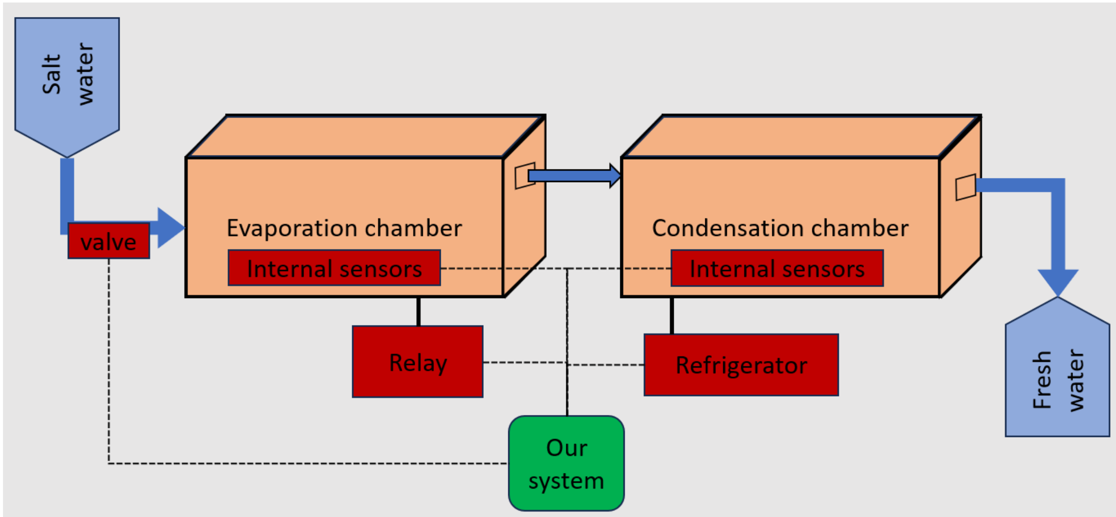


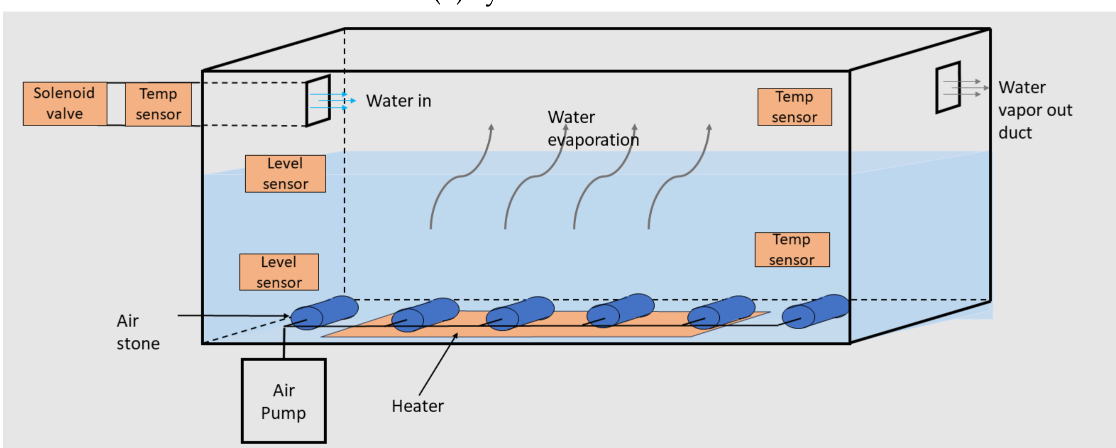
Figure 1: Desalination System: Large-scale[2]

The current prototype works by first manually powering on all sensors, valve, radiators, air pump, heater, and refrigerator. After powering on the flow rate is manually set by a flow meter. The process for conversion is the salt water intake flows into the evaporation chamber, as seen in figure 2b, where the water is evaporated by a heater to around 80°C (the air stones in the tank help to expedite the evaporation). Then vapor flows into the condensation chamber, figure 2c, through a duct between the two tanks to cool down the water vapor that is now desalinated. This happens by having a two radiators for proper cooling. The first radiator bring down the vapor from around 60°C down to about 40°C, and the second radiator with the help of a refrigerator cools the vapor down to sub 40°C where it begins to rain. The newly desalinated water is then collected in another tank for storage.

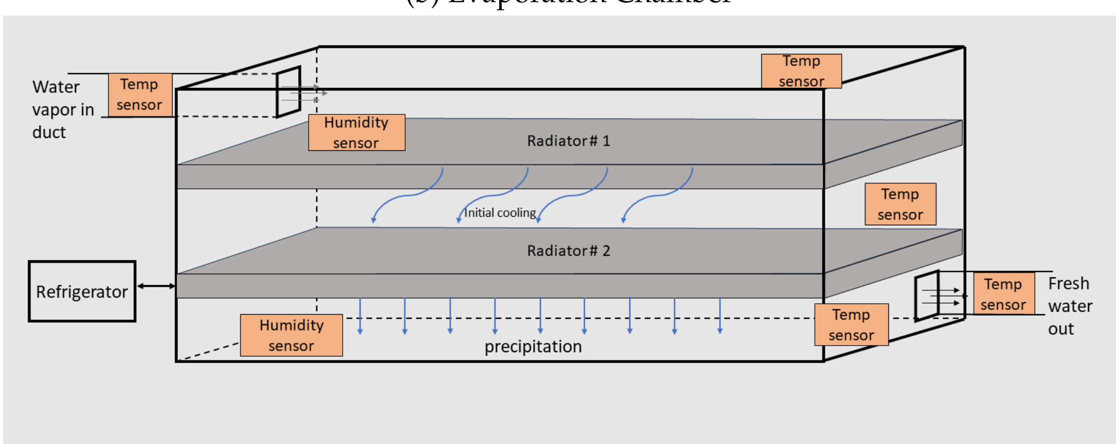
For our project we plan to take all the sensors (labeled in orange boxes) along with all external equipment to have a more user friendly and effective way to run experiments. This will allow for better efficiency of the system by determining yield percentage and better control of the various variables.



(a) system overview



(b) Evaporation Chamber



(c) condensation chamber

Figure 2: Prototype set up

1.4 High-level requirements list

The goals of our project which will determine if we are successful are as follow:

- Our system must take all current components in Professor Jont Allen's design, both electrical and mechanical, and consolidate them all under one roof for easier access and use when running experiments. These components include 8 temperature sensors, 2 binary level sensors, 2 humidity sensors, 1 solenoid valve, relay for the heater, 110v air pump, 2 radiators, and a refrigerator.
- With everything consolidated to one place, we will run our algorithm to power and start the system components of the 2 radiators, heater, air pump, and solenoid valve with the parameters from the sensors along with the input flow rate and time of the experiment in hours. Our algorithm will automatically update the system every 4 minutes as it gathers new data from the 8 temperature sensors, 2 binary level sensors, and 2 humidity sensors.
- The system will collect data every 4 minutes($\frac{1}{240}$ Hz) from the sensors in CSV file format which we will present the data graphically to express status in the system. The graph will be plotted for viewing after the experiment has completed to be able to properly display all information.
- If time allotted then we plan to run experiments and configure the systems setting to increase the efficiency yield of the system.

2 Design

2.1 Block Diagram

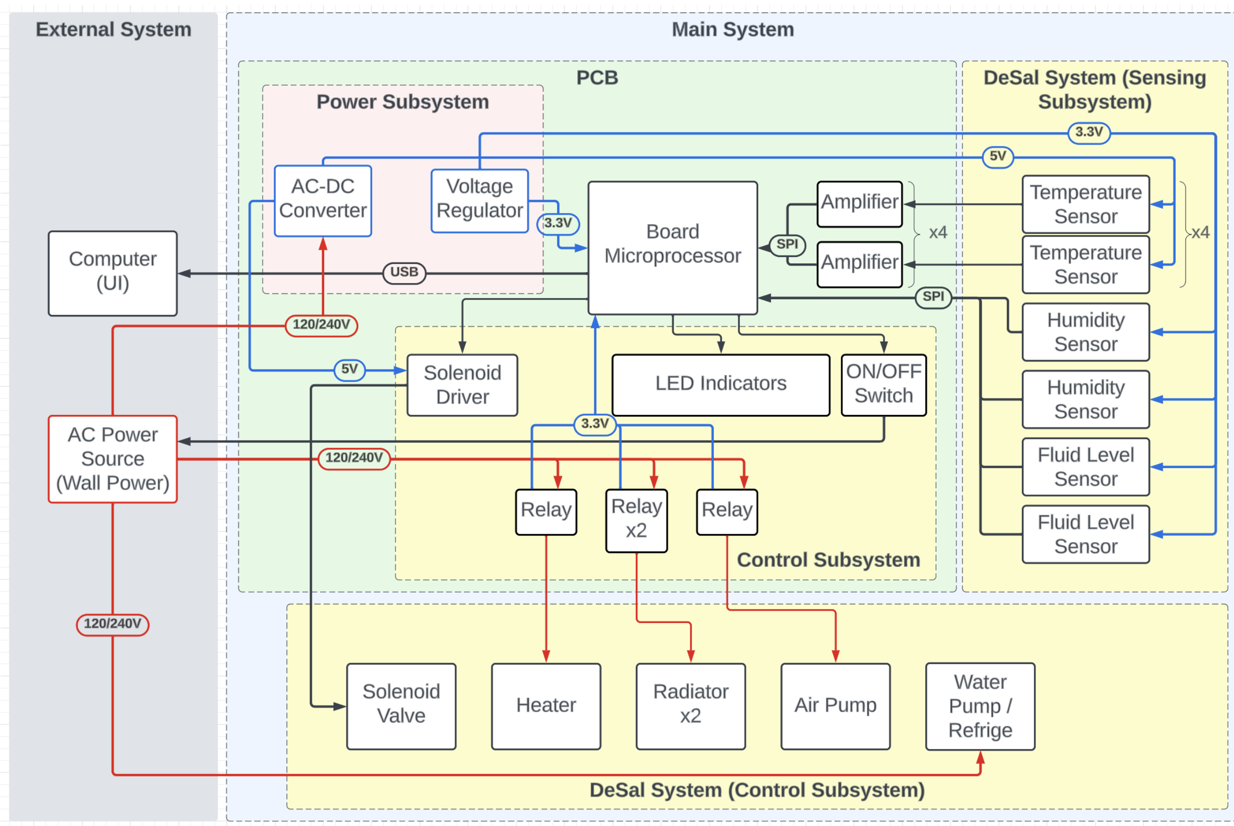


Figure 3: Block Diagram of DeSal system

Our desalination system comprises three main subsystems. Firstly, there is a power system that can provide electrical power to both AC and DC sources through converter and regulator. Secondly, a sensing subsystem is in place to gauge system control variables like temperature, humidity, and fluid levels. Lastly, there is an automated control system that manages the entire process by regulating the on/off switches of heater/radiators and a water valve responsible for introducing freshwater into our system.

2.2 Block Diagram Subsystems

2.2.1 [Subsystem: Power]

The AD-DC Converter, identified by part number 1939-WR9HADE3000USBFR6B-ND, plays the role of transforming power from a 120/240V AC source into 5V DC power. The rated input of this converter is 100-240V, output voltage is 5/9/12V and current out is 3/2/1.5A as specified in the datasheet. [3]. This 5V DC output is suitable for powering 8 temperature sensors, each of which needs 5V input voltage and draws a maximum of 1500uA of input current. Additionally, the converter can provide a 12V output, which is adequate for driving a solenoid driver with a 12V input requirement and a current draw of 540mA.

Table 1: AC-DC Converter - Requirement & Verification

Requirements	Verification
• Must be able to lower the high voltage output from an outlet: 120/240VAC that can be used to safely power up components with lower voltage intake each at different voltages: 5/12VDC.	• Connect input of the converter to wall outlet of 120V for voltage supply. Connect the output of the converter to a temporary load that equivalent to the circuit's load. • Check that the AC input voltage and output voltage using an oscilloscope to ensure power was properly converted from AC to DC at a 5/12VDC $\pm$5%.

The LM1117IMPX-3.3/NOPBTR-ND voltage regulator, with a specific purpose to maintain a stable 3.3VDC output, is crucial for taking the 5VDC output from the AC-DC Converter and converting it to the required 3.3VDC supply. This 3.3VDC is essential for powering 2 humidity sensors and an ESP32 microprocessor. The humidity sensors need a 3.3VDC input voltage and consume a minimum current of 1mA each, while the microprocessor requires a 3.3VDC input voltage with a minimum current requirement of 500mA. The datasheet for this voltage regulator indicates that it is capable of taking the 5VDC input and providing a constant 3.3VDC output with a current output capacity of 800mA. This output is more than adequate to meet the voltage and current requirements of both the 2 humidity sensors and the ESP32 microprocessor. [4]

Table 2: Voltage Regulator - Requirement & Verification

Requirements	Verification
• Use the voltage input of 5VDC and set it to a constant level of 3.3VDC to be utilized by humidity sensors and microprocessor.	• Connect input of the regulator to 5VDC and connect the output of the converter to a temporary load that equivalent to the circuit's load. • Check that the input voltage and output voltage using an oscilloscope to ensure voltage was properly converted from 5VDC to 3.3VDC $\pm$5%.
• The output current from the voltage regulator must meet the minimum current requirement of 500mA for humidity sensors and microprocessor.	• Connect input of the regulator to 5VDC and connect the output of the converter to a temporary load that equivalent to the circuit's load. • Check the output current using ammeter to ensure output current is 800mA $\pm$5%.

2.2.2 [Subsystem: Sensing]

The sensing subsystem has already been built by Professor Jont Allen, and we will be using it to collect raw data to manipulate in the Data Subsystem.

This subsystem consists of 8 temperature sensors, 2 fluid level sensors, and 2 humidity sensors. There are also 8 thermocouple amplifiers in the PCB which amplifies the low voltage signal generated by the temperature sensors to a high voltage that can be decoded by the microprocessor. Each sensor transmits data in SPI format to the microprocessor, allowing the microprocessor to both log and display the gathered data. The level sensors are binary level sensors each is position such that we know if water is above or below certain points in the tank.

Our microprocessor will read these sensors to know the state of the system and control the power on components in the control subsystem. This is so that the system maintains a steady state to ensure that experiments can run safely on their preventing any accident.

The DHT22 humidity sensor is utilized to measure the relative humidity as a percentage of water vapor in two different sections of a condensation chamber: one before the vapor passes through the first fan and the other after it passes through both fans. This sensor offers an accuracy level of approximately $\pm 2\%$. To ensure the proper operation of this sensor, it's essential to provide it with a power supply in the range of 3.3VDC to 5.5VDC. Additionally, the power source should be capable of delivering a current supply that meets the specified range of a minimum of 1mA and a maximum of 1.5mA. It communicates its data through a 40-bit digital signal transmitted via a 1-wire bus. However, the critical data consists of the initial 16-bit since 40-bit digital signal is composed of 16-bit RH data + 16-bit temperature data + 8-bit checksum. To obtain the relative humidity data, the 16-bit RH data is converted from binary system to decimal system, and then divided by 10 as specified in the datasheet. [5]

Table 3: Humidity Sensor - Requirement & Verification

Requirements	Verification
• The humidity sensor must operate with a minimum voltage of 3.3VDC	• Connect the VCC of the input voltage to a 3.3VDC power supply that is used to power our board along with other components • Using an multi-meter read the voltage of the IC by connecting the leads to the VCC and GND pins. if we read and absolute value of 3.3-6V the device is powered on.
• Read the 16-bit humidity data from the humidity sensor properly	• Ensure that power is connected to the input of the sensor. • using the known humidity of the current space the sensor is in, call the read function for the DHT. reading only the humidity portion of the data compare to the known value of the space occupied by the sensor. If it matches up withing a tolerance of $2\pm\%$ data is read correctly.

The MAX31855 temperature sensor amplifiers serve the purpose of measuring the temperature at nine distinct positions within both the evaporation and condensation chambers. These specific locations include the following:

1. The tube where the input water enters.

2. The upper section of the evaporation chamber situated above the water level.

3. The lower section of the evaporation chamber submerged in the water.

4. The tube where the water vapor exits the evaporation chamber.

5. The tube through which water vapor enters the condensation chamber.
6. The upper part of the condensation chamber, above the two fans.
7. The middle portion of the condensation chamber, positioned between the two fans.
8. The lower part of the condensation chamber, below the two fans.
9. The tube where the output water exits.

These sensors collectively provide temperature data from these crucial locations within the chambers. To ensure the proper operation of this sensor, it's essential to provide it with a power supply in the range of 3.0VDC to 3.6VDC. Additionally, the power source should be capable of delivering a current supply of 900uA. The data is provided in a 14-bit format that's compatible with SPI and is read-only. The temperature resolution of this converter is 0.25°C, allowing precise temperature readings across a wide range from as high as +1800°C to as low as -270°C as specified in the datasheet. [6]

Table 4: Temperature Sensor - Requirement & Verification

Requirements	Verification
• The temperature sensor must operate at a minimum of 3VDC	• Connect the VCC of the IC to a 3.3VDC power supply that is used to power our board along with other components • Using an multi-meter read the voltage of the IC by connecting the leads to the VCC and GND pins. if we read and absolute value of 3-5V the device is powered on.
• Be able to properly read the 14-bit temperature data output of the sensor	• Ensure that power is connected to input of IC to be able to read data. • Test with a known ambient temp, set CS to low and clock through 14 cycles. Using 2's complement with the 2 LSB being decimal point translate bits to compare to known value.

2.2.3 [Subsystem: Control]

The Control Subsystem is responsible for taking data from the sensing subsystem and commanding the valve and relays for the more power heavy components of the setup at the push of a button. With these given inputs, the control subsystem will have control over the relays and control variables of the setup. The Control Subsystem determines whether the setup will have power to run the experiment from the data it has gathered from the sensors with the implementation of our experiment running algorithm. Our Control Subsystem is made up of an ESP32microprocessor that interfaces with the sensors and amplifiers via SPI. These modules will be driven by the Power Subsystem via 3.3 volt outputs from the voltage regulator. To ensure that the Control Subsystem is fulfilling its responsibilities for receiving transmissions from the remote, taking data from the sensing subsystem, and commanding the relays, a requirements & verification table can be found below.

Solenoid Driver that Professor John Allen currently has is a "Plum Garden 1/4inch DC 12V 2 Way Normally Closed Electric Solenoid Air Valve". This device functions with a 12VDC voltage input and consumes approximately 540mA of current, resulting in a power usage of around 6.5W. It's installed at the input water tank and securely positioned, requiring a connection to our AC/DC converter by linking the red wire to VCC and the black wire to GND on our PCB.

Table 5: Solenoid Driver - Requirement & Verification

Requirements	Verification
• The Solenoid Driver must operate with a minimum voltage of 12VDC.	• Connect input of the drivers power to output of our AC-DC Converter. Using an oscilloscope we ensure that the driver receives 12VDC $\pm$ 5%. • Connect the input of the driver to a button to so when a high signal is received it power on. We will test for the signals using an oscilloscope to measure that the output is read from the input high.

The LED of choice for our project is the XLFBB01W, characterized by its blue source color and made with InGaN. This LED comes in a sizeable 10mm diameter, making it more than adequate for the task of signaling whether a device is powered on or off. To simplify our setup, we needed an LED that operates at 3.3VDC, which aligns perfectly with the voltage supplied by our voltage regulator. According to the datasheet [7], this LED has a forward voltage requirement of 3.3VDC, maximum of 4VDC, and draws a maximum current of 30mA. This LED is distinguished by its brightness, generous size, and a voltage rating that matches our requirements, making it an ideal choice for the purpose of indicating device status.

Table 6: LED Indicators - Requirement & Verification

Requirements	Verification
• We require a total of 5 LEDs, one for each of the following components: a Solenoid Driver, a Heater, two Radiators, and an Air pump. These LEDs should turn ON when their respective components are activated (or open in the case of the Solenoid Driver). The LED's need to operate at a maximum of 3.3VDC	• Connect the input of the LED to a signal equivalent to the corresponding relay that would come from the microprocessor.
• LEDs should run at 3.3 to 4VDC.	• Using oscilloscope, we probe the output of the microprocessor to that corresponds with the LED's to ensure that the output is at 3.3VDC $\pm$ 25%

We are intending to use MFS201N-Z slide switch as our ON/OFF switch for the system. This slide switch is designed for a minimum voltage rating of 5VDC and has a current rating of 10mA. It comes with six terminals, [8] and our plan is to configure it such that connecting pin 2 and pin 5 to pin 1 and 4, respectively, will turn the system ON. Conversely, when pin 2 and pin 5 are connected to pin 3 and pin 6, the system will be turned OFF.

Table 7: ON/OFF Switch - Requirement & Verification

Requirements	Verification
• A single ON/OFF switch is needed to activate all system components (a Heater, two Radiators, and an Air pump) and enable its operation.	• Connect the input of the switch to the output of the converter and using a voltmeter ensure that the input signal passes through when the switch is on.

We are incorporating the J107F1CS1212VDC.36 relay into our system, and its primary role is to enhance safety when dealing with high-voltage circuits. Given that we will be utilizing high-voltage wall power to operate our heater and fans, this relay is crucial for safeguarding our system. This relay is designed to operate within a specific voltage range. When the coil voltage (VDC) is set within the range of 3V to 3.9V, the relay's pick-up voltage (the voltage at which it activates) is specified as a maximum of 2.25V. This represents 75% of the rated voltage. On the other hand, the release voltage (the voltage at which it deactivates) is set at a minimum of 0.3V, which corresponds to 10% of the rated voltage [9]. This precise control over pick-up and release voltages ensures that the relay performs reliably and consistently within our system to manage high-voltage components safely.

Table 8: Relay - Requirement & Verification

Requirements	Verification
• Isolate the components to protect other subsystems in the system from the high voltage or currents that they carry.	• Connect the relay to a power supply of 120V and monitor the rest of the system with an oscilloscope. Ensure that when a short or surge occurs to the relay no other aspect of the system is affected.
• Control the high voltage and current feeding into each component for the safety of the components and the people who are working on the experiment.	• Using an oscilloscope we measure the input and output voltage of the relay to ensure that power of 120V is being properly fed to each component.

The microcontroller we've selected is the ESP32-S3-WROOM-1-N16, and it serves a important role in our system. Its primary functions include gathering data from 9 temperature sensors and 2 humidity sensors, which is then visualized for user control through graphical interfaces. Additionally, it manages system components by enabling or disabling the heater, two fans, and the solenoid driver.

One of the key factors in choosing this microcontroller was its voltage compatibility with our system, simplifying the power setup. All of our sensors operate at 3.3VDC, and the ESP32 conveniently accepts a 3.3VDC input. Furthermore, the ESP32 was favored due to its abundant GPIO pins, a necessity for accommodating various sensors and devices in our setup.

The device has four strapping pins, namely GPIO 0, GPIO 3, GPIO 45, and GPIO 46, each serving a distinct purpose. GPIO 0 allows us to toggle the device between "Download Mode," facilitating program memory flashing when set to 0, and regular boot mode when set to 1. We plan to connect this pin to a button for user control. GPIO 3 is designated for switching the source for JTAG signals, and we intend to link it to a voltage source through a resistor. This arrangement will allow us to set it to a high state when the resistor is soldered and to a low state when the resistor is detached. GPIO 45 is involved in establishing the voltage for the internal SPI memory IC, and we've decided to leave it floating, meaning it won't be actively connected to any specific value. Lastly, GPIO 46 plays a role in configuring the ESP32-S3 ROM to transmit messages, similar to GPIO 3, we plan to manipulate its state by attaching and detaching a resistor as needed. As stated in the datasheet, the power supply voltage for this device should be 3.3VDC, and it requires a minimum current of 0.5A from the external power supply [10]. Our voltage regulator, with an output voltage of 3.3VDC and a current capacity of 0.8A, fulfills this power supply requirement.

Table 9: Microprocessor - Requirement & Verification

Requirements	Verification
• Be able to decode data from the sensors to use in its calculations of our algorithm for self running experiments.	• Setting the values of the sensors inputs to set value and ensuring that the processor is properly reading the data correctly from the sensors.
• Utilize our software and algorithm using the data it receives from the sensors and computer to be able to run experiments autonomously by powering relays.	• continuously feeding the processor data on possible different sensor reading connect an oscilloscope to the output pins connected to the relays to see if the change when necessary.
• Process and send data it gathers about the sensors to the computer to be used for visualization for user interface.	• Save the data gathered by the microprocessor in a csv file format by hard setting the sensors to known values to ensure proper data transmission.

Our software should allow for a user to be able to autonomously be able to run an experiment without worrying about controlling or initializing any component. This will be achieved by having all the sensors and components connected to our board such that they can be initialized and controlled by our micro controller when the user opens our interface to start and run. When an experiment is run we will continuously check on certain aspects of the system, such as water level to ensure that the water isn't overflowing or evaporating too quickly such that it may cause burning. Our algorithm will check for this and will turn off the heater if the water is too low or turn off the water if it is too high. We will also calculate how much water we assume there to be left in the initial water tank so that if need be we can stop the experiment to cause no safety concerns. If no flags are raised then the experiment should run for as long as stated by the user at start up.

Table 10: Software - Requirement & Verification

Requirements	Verification
• The software should gather data from the sensors via a microprocessor.	• Save the data gathered by the microprocessor in a csv file format by hard setting the sensors to known values to ensure proper data transmission.
• The software should display the variables on a single graph, allowing for the visualization of changes in control variables: temperature, humidity, and water level. Each variable should be distinctly labeled with different colors to enable the visualization of sensor at different times.	• Graph the data from the saved csv file and compare to the expected previous known experiment graphs.

2.3 Circuit Schematic & Description

The following Describe each aspects aspects of our PCB to house the different components to be able to control and regulate Professor Jont Allen's lab set up.

2.3.1 Voltage Regulator

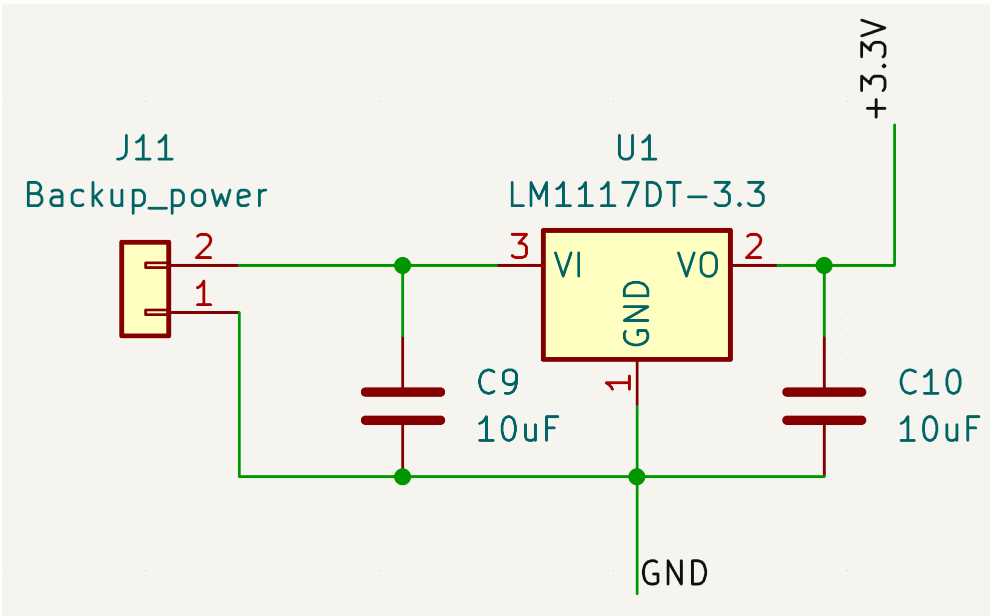


Figure 4: LM1117DT-3.3 voltage regulator

<**Backup_power**>

1x2 pin connector for external power supply of 5VDC.

<**LM1117DT-3.3**>

VI: 5VDC voltage input from external power supply.

VO: 3.3VDC voltage output for various sensors and microprocessor.

GND: Ground.

<**Capacitor**>

10uF capacitor for decoupling

2.3.2 Temperature Sensors

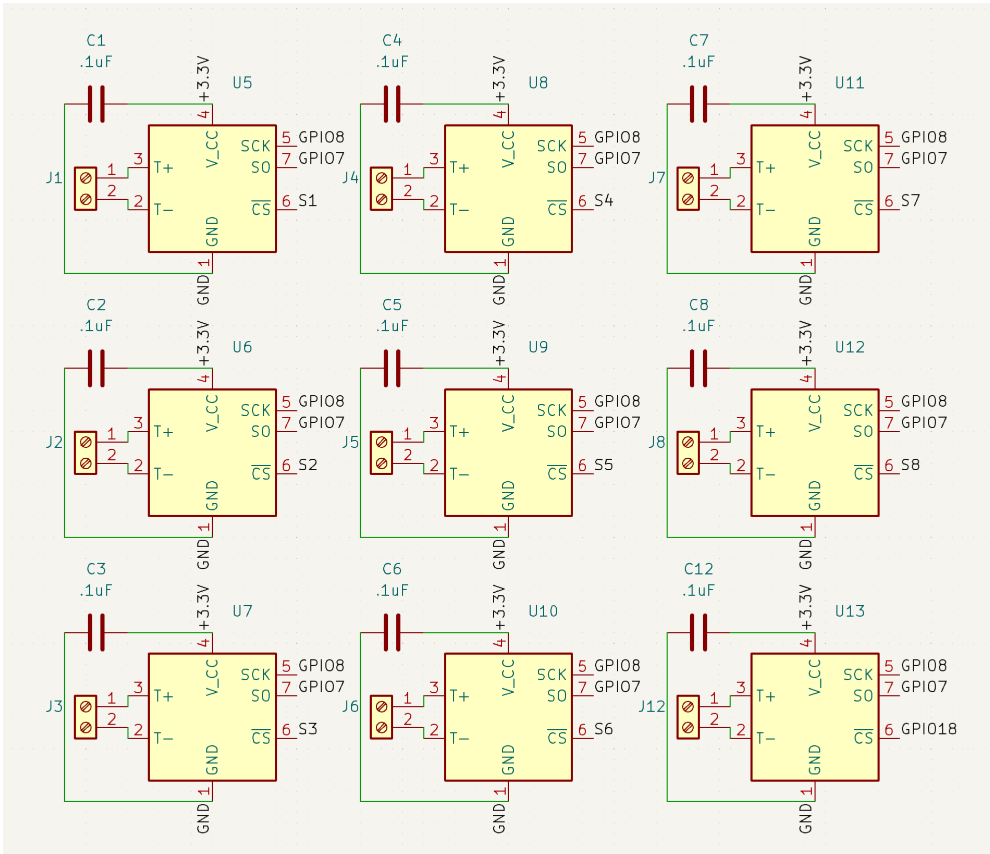


Figure 5: MAX31855JASA+T

<**MAX31855JASA+T**>

V_CC: 3.3VDC voltage input.

SCK: Source clock connected to GPIO 8 of ESP32.

SO: Data output connected to GPIO 7 of ESP32.

CS: Chip select connected to 3:8 Mux and GPIO 8.

T+: Connected to screw terminal for thermocouple.

T-: Connected to screw terminal for thermocouple.

GND: Ground.

<**Capacitor**>

0.1uF capacitor for decoupling

2.3.3 3:8 MUX

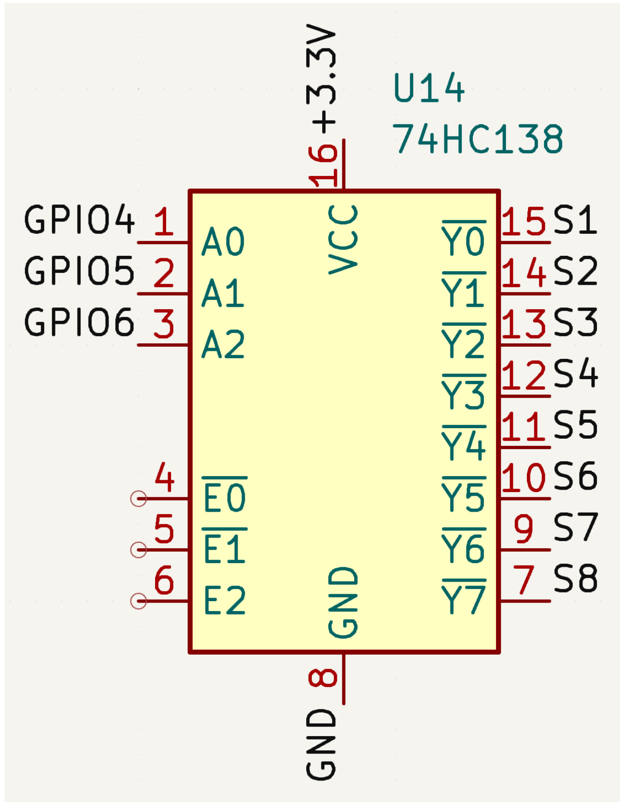


Figure 6: SN74HC138DR

<**74HC138**>

VCC: 3.3VDC voltage input.

A0∼A2: Connected to GPIO4 GPIO06 for chip select.

Y0∼Y7: Connected to CS of 8 MAX31855 to select which MAX31855 data to receive.

GND: Ground.

2.3.4 Humidity Sensors

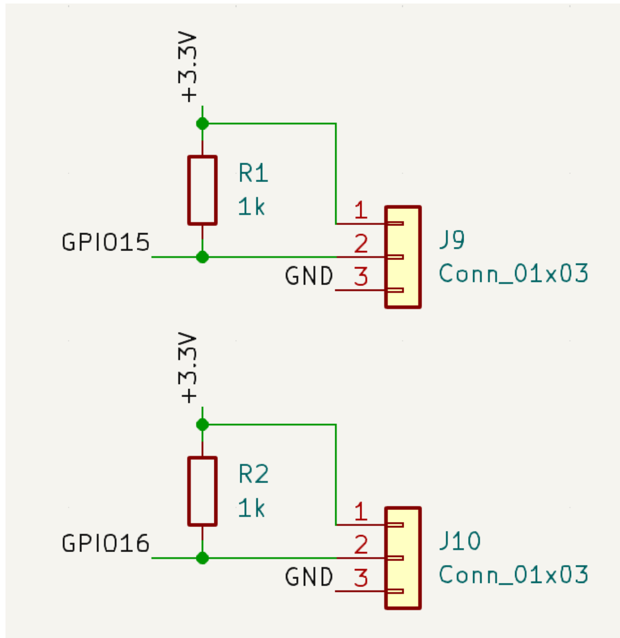


Figure 7: Connection for DHT22

<Conn_01x03>

Pin connector for DHT22 humidity sensors.

1: 3.3VDC voltage input for supplying power to DHT22.

2: Data out from DHT22 connected to GPIO15 and GPIO16.

3: GND.

2.3.5 USB Port

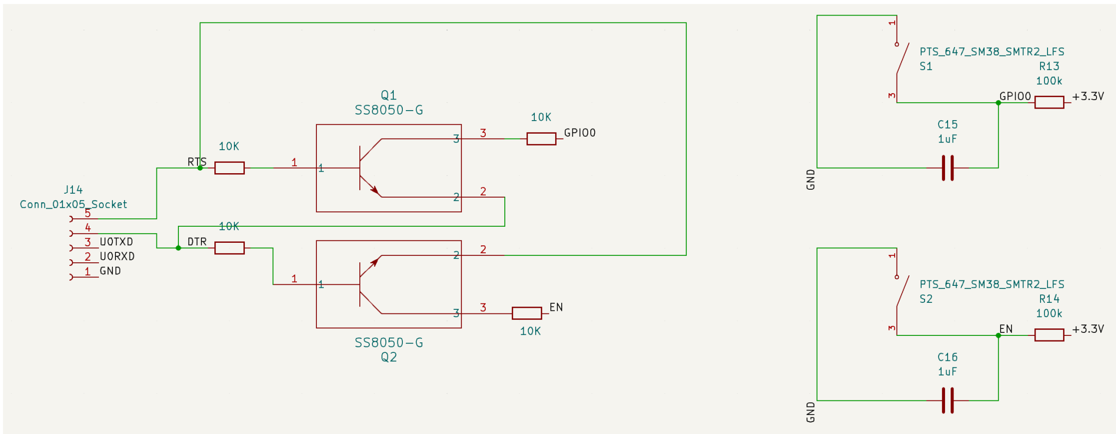


Figure 8: Connection for USB bridge/port

<**SS8050-G**>

NPN BJT to automate the use of GPIO0 to place the device into "Download Boot Mode".

<**PTS_647_SM38_SMTR2_LFS**>

Buttons for enabling strapping GPIO and EN of ESP32.

<**Conn_01x05**>

Pin connector for USB port module.

RTS: Reset To Send, an input used for flow control.

DTR: Data Terminal Ready, an output used for flow control.

UOTXD: Connected to TX, serial data transmit pin.

UORXD: Connected to RX, serial data receive pin.

GND: Ground.

2.3.6 Microprocessor

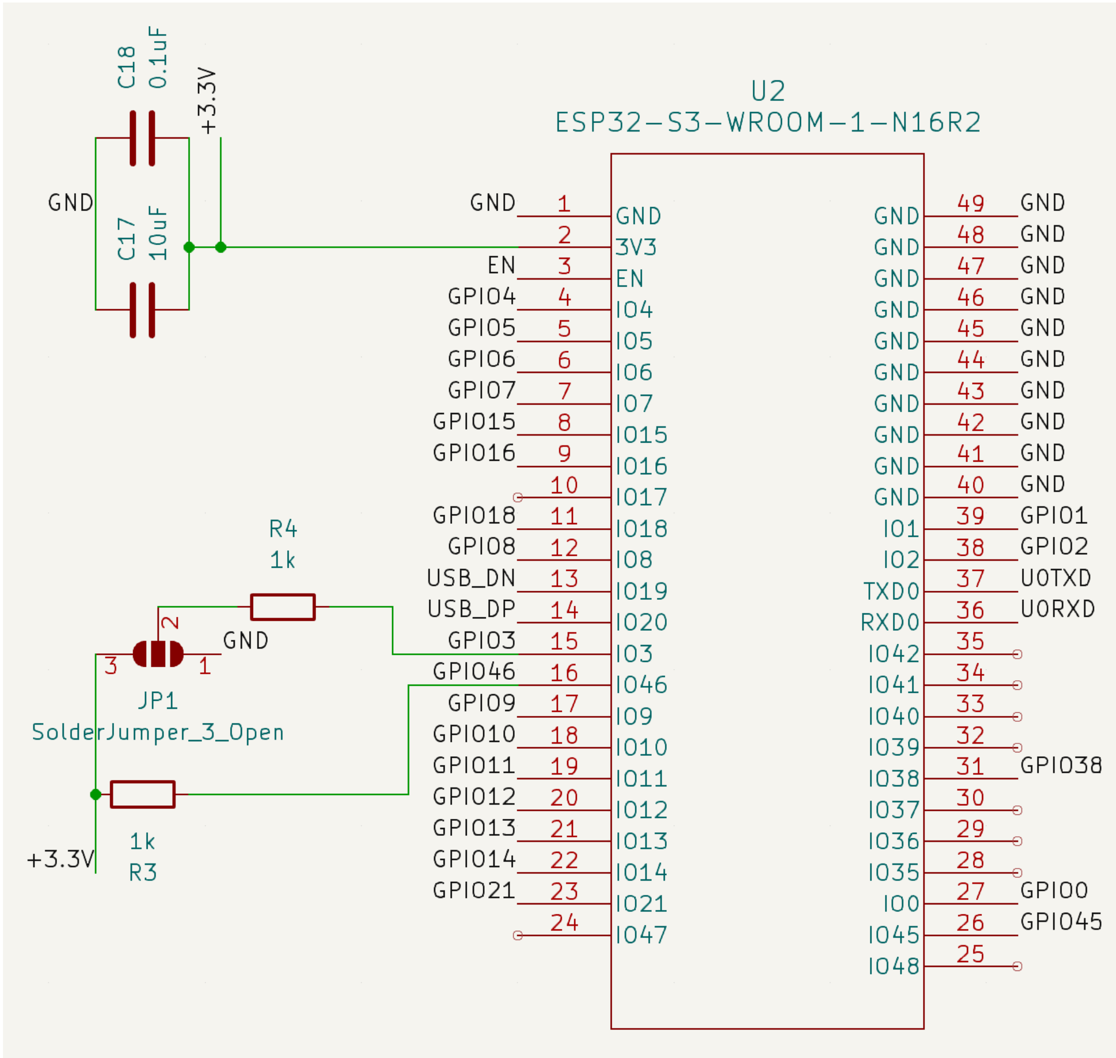


Figure 9: ESP32-S3-WROOM-1-N16

<ESP32-S3-WROOM-1-N16R2>

3V3: 3.3VDC voltage input.

GPIO4~GPIO6: Connected to 3:8 MUX for chip select.

GPIO7: Connected to SO of MAX31855 for receiving data.

GPIO8: Connected to CK of MAX31855 for generating source clock.

GPIO3: Strapping pin for switching the source for JTAG signals.

GPIO46: Strapping pin connected/disconnected to register for toggling high/low.

GPIO15, GPIO16: Connected to DO of DHT22 for receiving data.

UOTXD: Connected to USB port.

UORXD: Connected to USB port.

GPIO0: Connected to button for controlling download.

EN: Connected to button for controlling download.

GND: Ground.

2.3.7 Pin header

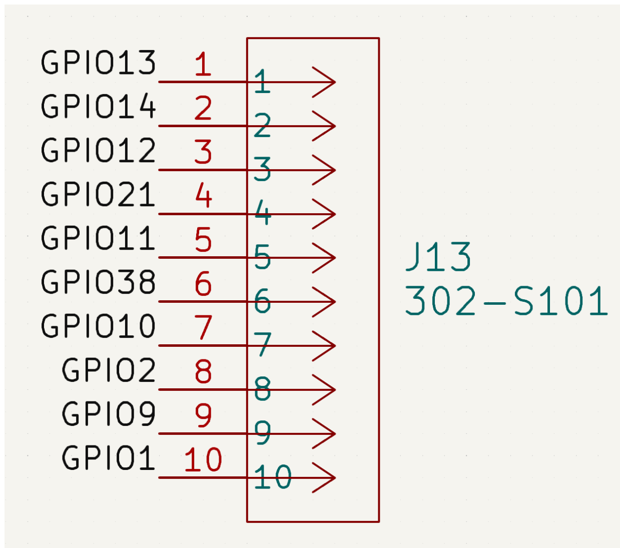


Figure 10: Pin header for GPIOs

<**302-S101**>

Connected to unused GPIOs of ESP32 for later use on the second PCB.

2.4 PCB Design

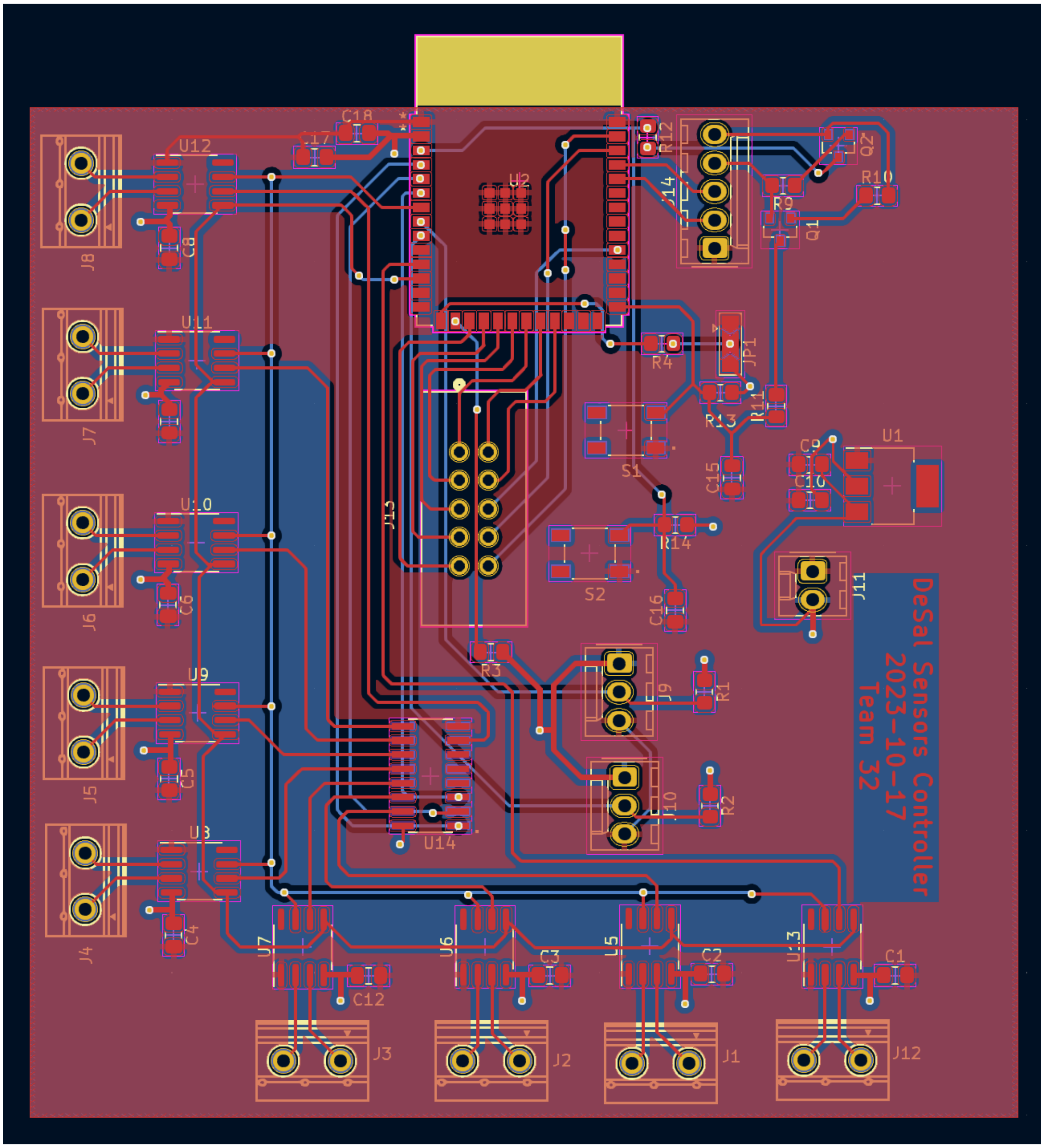


Figure 11: PCB Design

2.5 Procedure for Setup

Since we want to be able to run experiments autonomously from out PCB we need to define how we plan to set things up for this type of operation. If time allotted we plan to implement a UI that allows the user to input the experiment run time, flow rate, and location to where to store the graph after the experiment. If this cannot be implemented then we will simply use the command prompt to have the user execute the experiment as a program with the inputs of time and flow rate. Once the experiment had started it will run until the time it was specifies has terminated or there is no more water in the input tank to be able to use for the experiment. During the course of the experiment our board will collect data and pass it to the Raspberry Pi every four minutes. Regardless of if the UI is implemented the graph for the data output will be plotted using MatLab, as this is how Professor Jont Allen has the lab set up as. The graph plots the 8 temperature sensors and 2 humidity sensors of the system for the user to view.

2.6 Tolerance Analysis

To ensure that the system performs optimally, the water level must be kept to a certain level. In order to achieve this, we must predict the evaporation rate. The temperature inside the evaporation chamber is going to be held to 40°C as it is meant to imitate hot desert days. However the accuracy of the predictions will be limited by sensors' capabilities. The thermocouple amplifiers (Adafruit MAX31856) used to read the temperatures up to 1800°C with an accuracy of ± 0.1°C.

Evaporation rates can be modeled by the following equation[11]:

$$E_0 = \frac{700T_m/(100 - A) + 15(T - T_d)}{80 - T}$$

where:

E_0 = Evaporation rate (mm day^{-1})

T = Temperature (°C)

T_m = $T + 0.006h$ where h in the elevation in meters

A = Latitude (°)

T_d = Mean dew-point (°C)

This equation is a simplified version of the Penman equation for modeling evaporation.

For our purposes variable T will be set to 39.9°C and 40.1°C as that is the range of accuracy for the thermocouple amplifiers if we set it to 40°C. Variable A will be set to 40.12°, T_m will be kept as $T + 0.006(233)$ with the value of T changing with respect to the calculation and h being set to 233m as that is the elevation of Champaign IL, and T_d will be calculated using the following formula:

$$T_d = T - \frac{100 - RH}{5}$$

where:

RH = Relative Humidity (%)

The relative humidity in the evaporation chamber will be provided by the humidity sensor inside.

From the above information, we can bound the evaporation with the following equation:

$$\frac{700(39.9 + 0.006(233))/(100 - 40.12) + 15(\frac{100-RH}{5})}{80 - 39.9} \leq$$
$$\frac{700(40.1 + 0.006(233))/(100 - 40.12) + 15(\frac{100-RH}{5})}{80 - 40.1}$$

This can be simplified to:

$$0.074813RH + 19.483181 \leq 0.075188RH + 19.714626$$

The above equation models the range of the of the actual evaporation in terms of the mm day^{-1}. The RH variable will be the average humidity that is measured throughout the day. Let us assume that the relative humidity is 20%. Therefore:

$$20.979 \leq 21.218$$

This means that between 20.979 mm/day and 21.218 mm/day of evaporation is expected per day at a relative humidity of 20

Once a relative humidity level is decided, we will be able to predict when water the water is getting too low, or too high. The water level in the evaporation chamber must be between the two water level sensors set up in the chamber. The upper water level sensor is there to ensure that the water does not leak into the tube that transports water vapor. The lower sensor is there to make sure that the heater shuts off when the water gets too low as it can potentially overheat. This will allow the system to keep running optimally, and reduce the need for manual adjustment to make the experiments easier to run.

3 Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor

The average ECE graduate from UIUC makes about $51 per hour. working at 40hrs/week for 3 team members working for 9 weeks, $\$51/hrx40hrs/weekx3x9 = \$55,080/week$ in labor cost.

3.1.2 Parts and Materials

Table 11: Software - Requirement & Verification

Description	Manufacturer	Quantity	Price	Link
ESP32-S3-WROOM-1-N16	Espressif Systems	1	$3.48	link
amplifier Adafruit MAX31856	adafruit	8	$17.50	link
regulator TCR2EF33,LM(CT	Toshiba	1	$0.33	link
Driver 5648	adafruit	1	$3.95	link
Switch JS202011SCQN	C&K	1	$0.64	link
LED XLM2MR12D	SunLED	5	$0.73	link
AC/DC converter PBO-3c-5	CUI Inc	5	$2.67	link
Relay 3191	adafruit Inc	4	$9.95	link
100 ohm resistor RC0402FR-071KL	YAGEO	10	$0.14	link
1k ohm resistor RC0402FR-07100RL	YAGEO	10	$0.14	link
10k ohm resistor RK73B1JTTD103J	KAO Speer Electronics	10	$0.15	link
1000pF cap KGM05AR71H102KH	KYOCERA AVX	10	$0.14	link
10000pF cap KGM05AR71H102KH	KYOCERA AVX	10	$0.14	link
0.1uF cap CL05A104KA5NNNC	Samsung	10	$0.10	link

3.1.3 Net Cost

Accounting for all components necessary in table 11, this amounts to a net price of $217.20, accounting for tax of 10% and shipping of 5% we get the total parts cost to be $249.78. The cost of the labor plus the cost of the parts rounds the net cost to be $55329.78.

3.2 Schedule

Table 12: Schedule for Project Progression

Week	Task	Person
Oct. 2nd - Oct. 9th	Collect data regarding the sensors that are currently in use	Everyone
	Build circuit schematic	Seunghwan
	Start PCB design	Alan & Kinjal
Oct. 9th - Oct. 16th	Order parts for prototyping	Everyone
	Revise PCB design	Everyone
	Order PCB	Everyone
Oct. 16th - Oct. 23rd	Test the requirement & verification	Everyone
	Build prototype PCB	Seunghwan
Oct. 23rd - Oct. 30th	Revise PCB design	Kinjal
	Program Microprocessor	Alan
Oct. 30th - Nov. 6th	Order PCB	Everyone
	Build software	Alan & Kinjal
	Test software with professor Allen's DeSal system	Everyone
Nov. 6th - Nov. 13rd	Revise PCB design	Seunghwan
	Bug fixing for software	Everyone
	Continue testing with professor Allen's DeSal system	Everyone
Nov. 13rd - Nov. 20th	Order PCB	Everyone
	Continue testing with professor Allen's DeSal system	Everyone
Nov. 20th - Nov. 27th	Minor bug fixing	Everyone
Nov. 27th	Final Demo	Everyone

4 Ethics and Safety

Our project involves electronics in close proximity with water, so there are a few measures to take to ensure that no component that should not get wet comes in contact with water. The testing will take place in a room in the ECEB that Professor Allen has stored his prototype. In order to ensure that no water comes into contact any of the electronics, we will only connect the PCB to the prototype when the lid of the compartments are closed.

In addition, the PCB will also be stored in a closed compartment away from the water enclosures to prevent any possibility of contact if there is spillage.

The heater used to raise the temperature of incoming water to 40 °C within the evaporation chamber has the potential to result in system damage by overheating if not properly controlled. To address this concern, we are implementing a control system where the heater will automatically switch off without any manual interference once the lower water level is triggered..

On the other hand, the full scale solution can potentially have a few ethics and safety issues. One such potential issue would be siphoning too much salt water from one specific place, thus disrupting the environmental balance of that ecosystem. This would be in direct violation of ACM 1.2 [12]. Any real-life instance of this project must consider rotating between multiple sources and saltwater ecosystems in order to keep disruption as minimal as possible. Another potential concern could be securing the perimeter of the aquifer and all other water deposits to minimize injury to any on-site engineers and any others.

References

[1] R. A. Dargham. “Water doesn’t come from a tap.” (), [Online]. Available: https://unicef.org/mena/water-doesnt-come-tap#:~:text=The%20Middle%20East%20and%20North%20Africa%20is%20the%20world’s%20most,world’s%20most%20water-scarce%20countries..

[2] J. Allen. “Scrubbing co2.” (), [Online]. Available: http://auditorymodels.org/index.php?n=Site.IROSE (visited on 08/22/2022).

[3] “D2329960.” (), [Online]. Available: https://globtek.my.salesforce-sites.com/spec/pdf/1323330/d2329960.

[4] “Lm1117 800-ma, low-dropout linear regulator datasheet (rev. q).” (), [Online]. Available: https://www.ti.com/lit/ds/symlink/lm1117.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1697693812014.

[5] “Digital+humidity+and+temperature+sensor+am2302.” (), [Online]. Available: https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf.

[6] “Max31855.” (), [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/max31855.pdf.

[7] “Xlfbb01w.” (), [Online]. Available: https://www.digikey.com/en/products/detail/sunled/XLFBB01W/4901599?s=N4IgjCBcoMxgnFUBjKAzAhgGwM4FMAaEAeygG1wB2
2BBi7q1AGwWwdyrvNFdUyvRkX39KYyJKK11EIltqZwhpeHp4KwspXTMGBPdZKJAPMBcpZj
2BYTFJEABae0VlfgBXdVJIClymcTLcppAAGQBRABE8kEZFVigwNlHIfzagA.

[8] “Mfs201n-z.” (), [Online]. Available: https://www.digikey.com/en/products/detail/nidec-copal-electronics/MFS201N-Z/5086556.

[9] “J107f1cs1212vdc.36.” (), [Online]. Available: https://www.digikey.com/en/products/detail/cit-relay-and-switch/J107F1CS1212VDC-36/12502789.

[10] “Esp32-s3-wroom-1$_{w}room - 1u_{d}atasheet_{e}n$.” (), [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf.

[11] E. T. Linacre, “A simple formula for estimating evaporation rates in various climates, using temperature data alone,” *Agricultural Meteorology*, vol. 18, no. 6, pp. 409–424, Jan. 1977. DOI: 10.1016/0002-1571(77)90007-3.

[12] IEEE. “”IEEE Code of Ethics”.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).