CPE 301 – Embedded Systems Design

Final Project Report

Rajat Sharma, DJ DeLucca, Brysen Ross, Logan Wehr

May 9, 2025

Table of Contents

1) **Project Description**
    a. Description of Project
        i. Description of Components and Functionalities
        ii. States and State Transition Components and Functionalities
2) **System Overview and Circuit Image**
    a. Overview of Design
3) **Schematic Diagram**
    a. Specification Sheets
4) **System Demonstration**
    a. Demonstration Video
5) **Submission Links**
    a. Video of Operation
    b. GitHub Link

Project Description

This final report consists of implementing a swamp cooler utilizing the Arduino ATMega 2560 as well as sensors provided with the Arduino kit. These coolers are meant to be energy efficient in that they may be used as possible alternatives to traditional air conditioners, hence an emphasis is placed on the energy efficiency as well as register-level manipulation to allow lower-level control to systems.

The system allows for lower water levels to be analyzed and detected in the reservoir, triggering an alert which displays readings on the LCD screen installed. Based off the temperature threshold, the fan automatically starts or stops to adjust the temperature quickly as necessary. Moreover, the implementation of an interrupt system to enable or disable the system allows for proper usage of the interrupt system, and events are logged and transmitted via a USB connection. The system operates in four primary states that are outlined by the state diagram and the schematic created:

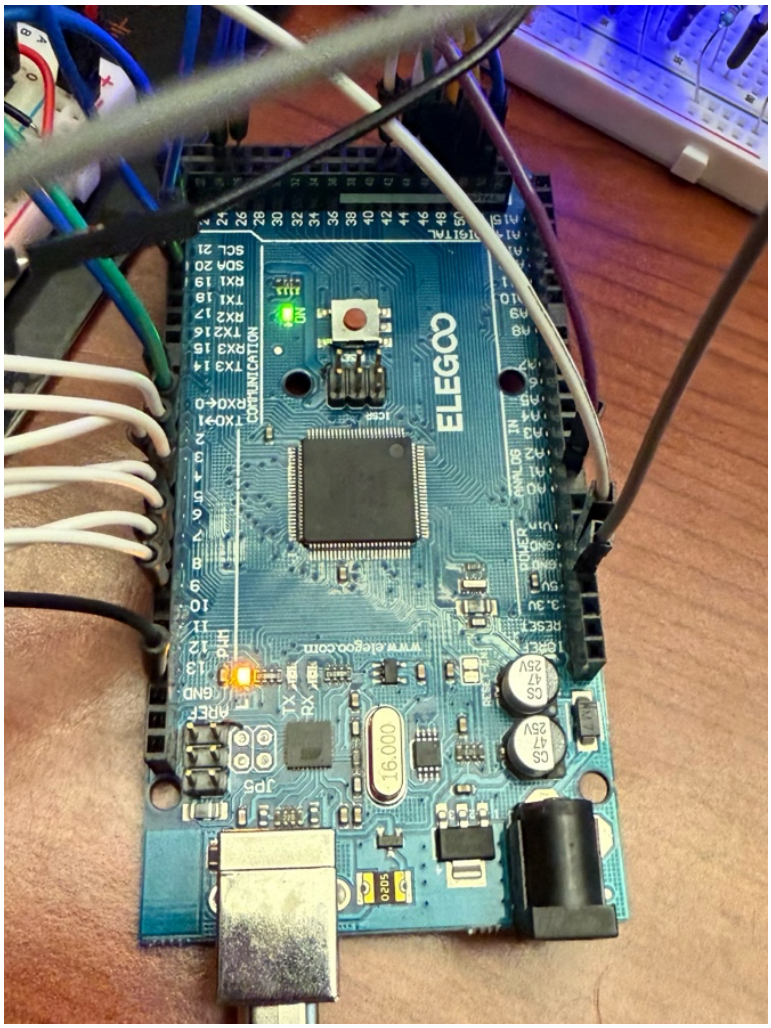
- **Disabled**
    - The system is off, and the yellow LED light is on.
- **Idle**
    - The system is monitoring the temperature and the water level, and the green LED on.
- **Running**
    - The system's fan is running if the temperature exceeds the threshold, with the blue LED on.

- **Error**
    - The system does not have enough water, and the motor is disabled, indicating that the red LED is on.

Components and Functionalities

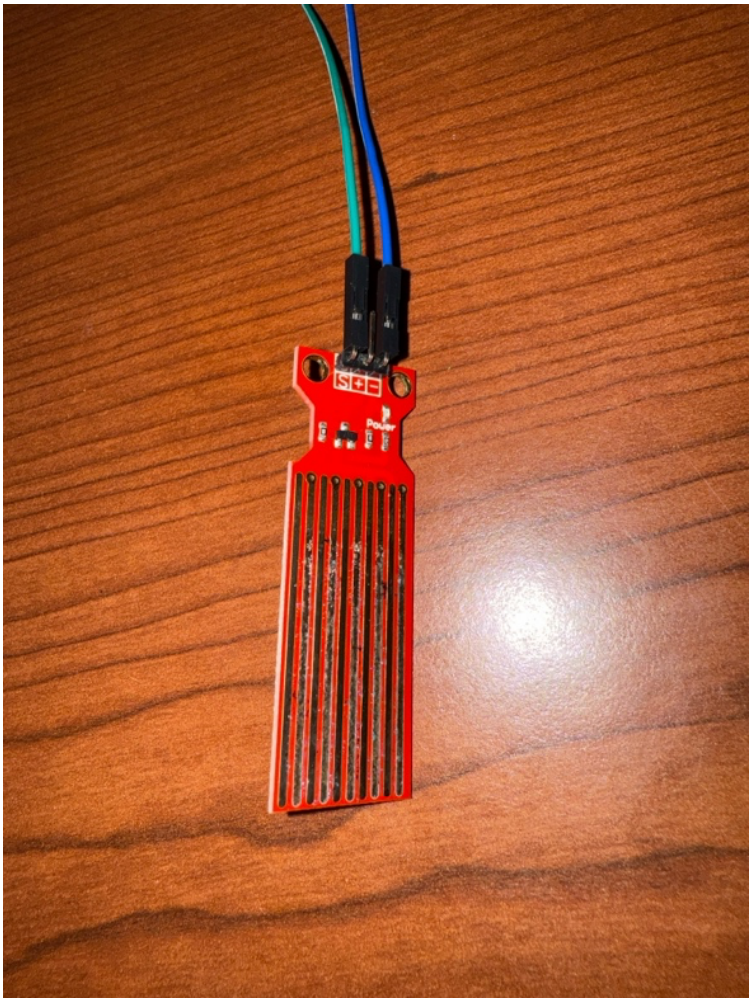
Arduino ATMega2560

- Serves as the main microcontroller for executing the logic, dealing with the sensors, and controlling all actuation throughout the system. Additionally, it is the central processing unit for sensor data, state logic, as well as peripherals.
- It provides 54 digital pins and 16 analog pins to be utilized as an interface with peripherals.
- Allows for direct register-level manipulation and avoids using built-in functions and commands in the Arduino.



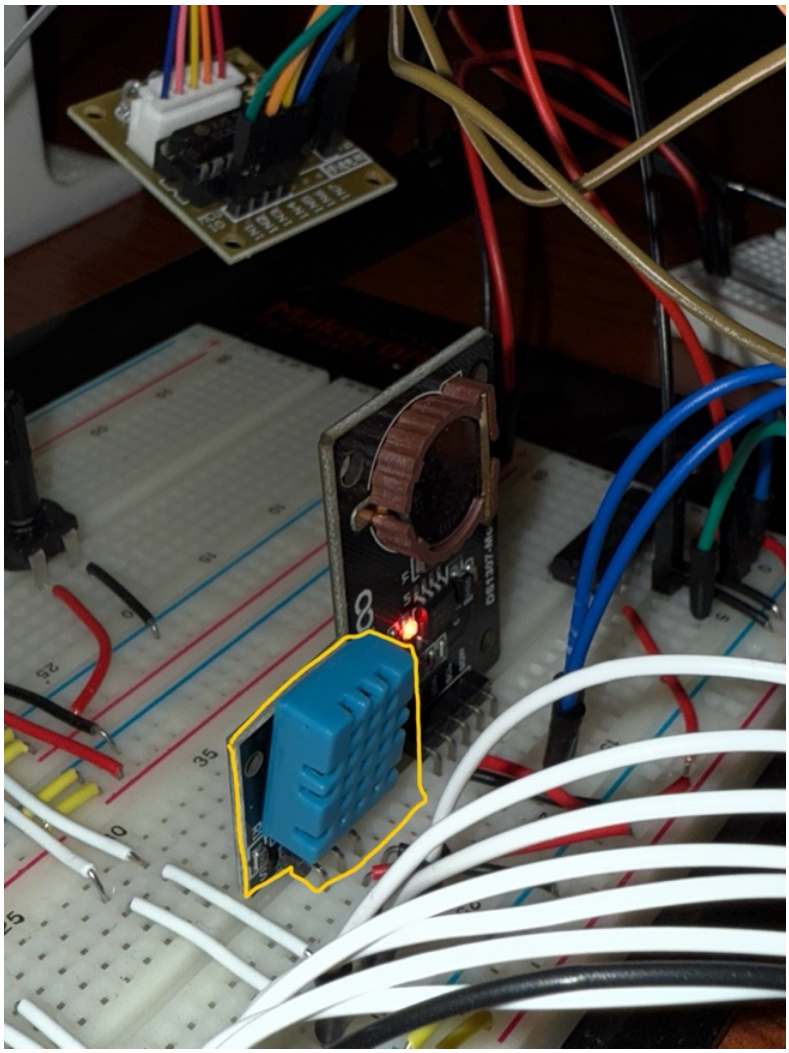
Water Level Sensor

- Detects water levels in the reservoir directly via ADC sampling and the analog input. It will directly trigger the error state if the water level is too low or if another issue arises.
- Utilization of an analog interface and poll-based sampling. An error state is triggered if the level is below a certain threshold.
- The sensor is connected to analog pin 0, with direct connections to power (5V) and ground from the Arduino.



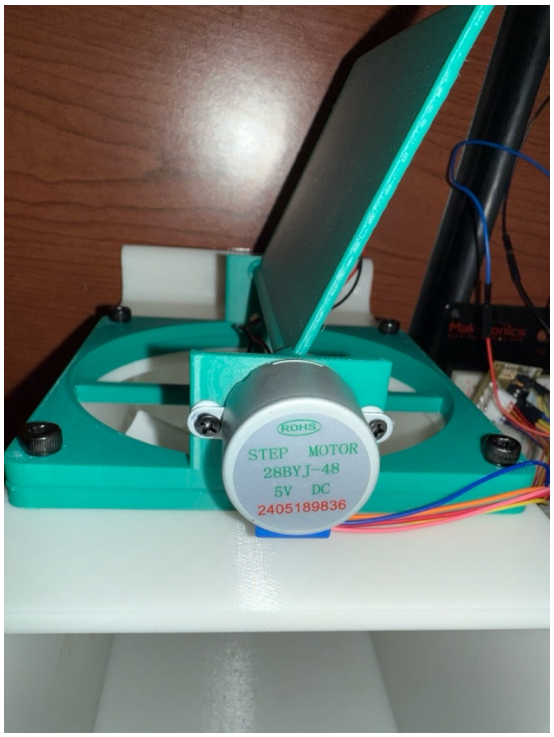
DHT11 (Temperature / Humidity Sensing Device)

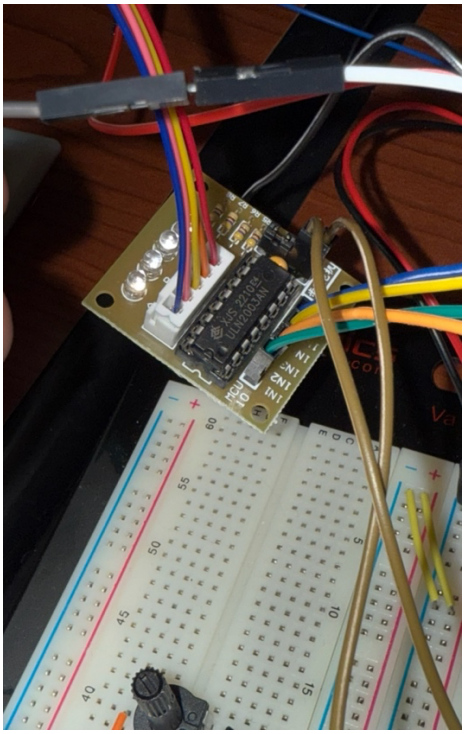
- Allows for the continuous monitoring of the temperature and humidity and remains active in all the states except for the disabled state, connected to digital pin 4 directly to the Arduino and direct connections to power (5V) and ground from the Arduino. It reports the temperature and humidity back and displayed to the LCD screen.



Stepper Motor / ULN2003 Driver

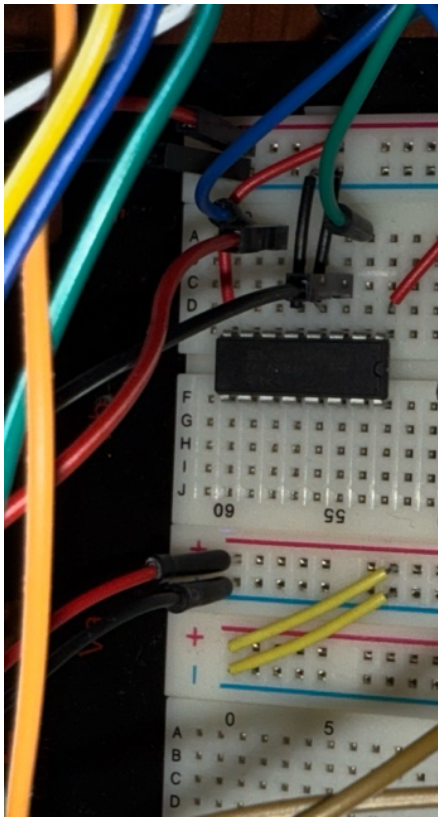
- Allows for the adjustment of the vent angle via user input (determined by the buttons) and will work in all stated except the disabled states. It will rotate clockwise/counterclockwise based off the button press.
- Direct connection to the ULN2003 driver, which powers the stepper motor, connected to Arduino digital pin 46, 48, 50, 52, ground and power.
- The button automatically determines whether the vent needs to be adjusted, and the motor needs to go clockwise or counterclockwise.





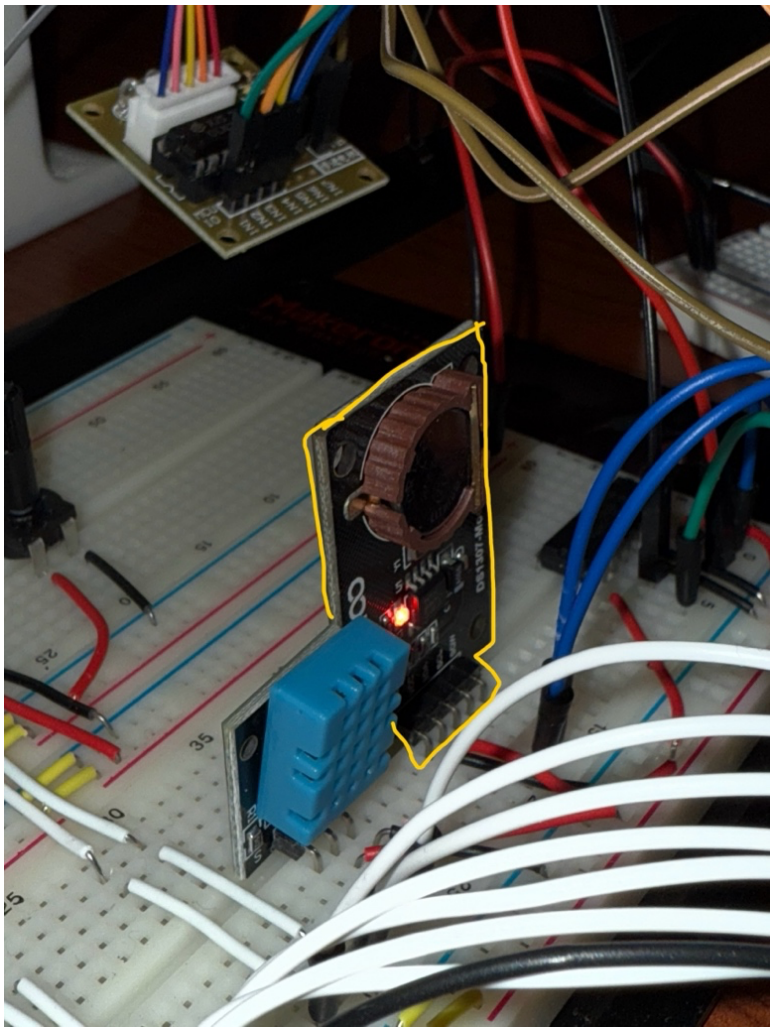
DC Motor / L293-DC Motor Driver

- Mainly used to provide cool air by spinning the attached fan and when the temperature exceeds the threshold during the running state. It is used harmoniously with the L293-DC Motor Driver, connected to the Arduino digital pins 2 and 3 as inputs and the DC motor ground / power pins.
- The usage of an external supply also helps avoid current overflow.
- The fan turns on when the temperature exceeds a certain threshold.



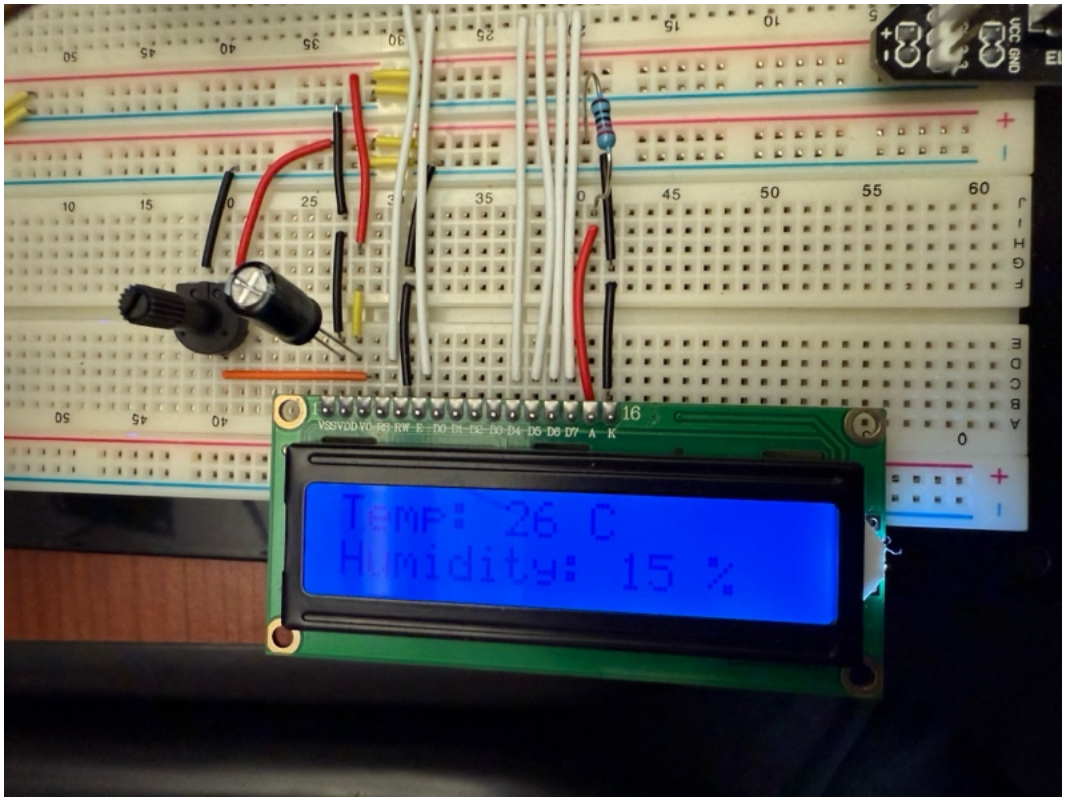
DS1307 Real Time Clock

- Provides exact timestamps for certain state transitions, required to be active in all states as timestamps are needed to print each state change that occurs.
- Utilizes communication pins via Arduino (20, 21).
- All timestamps are recorded on the serial monitor and the events are logged for proper debugging/recording purposes.



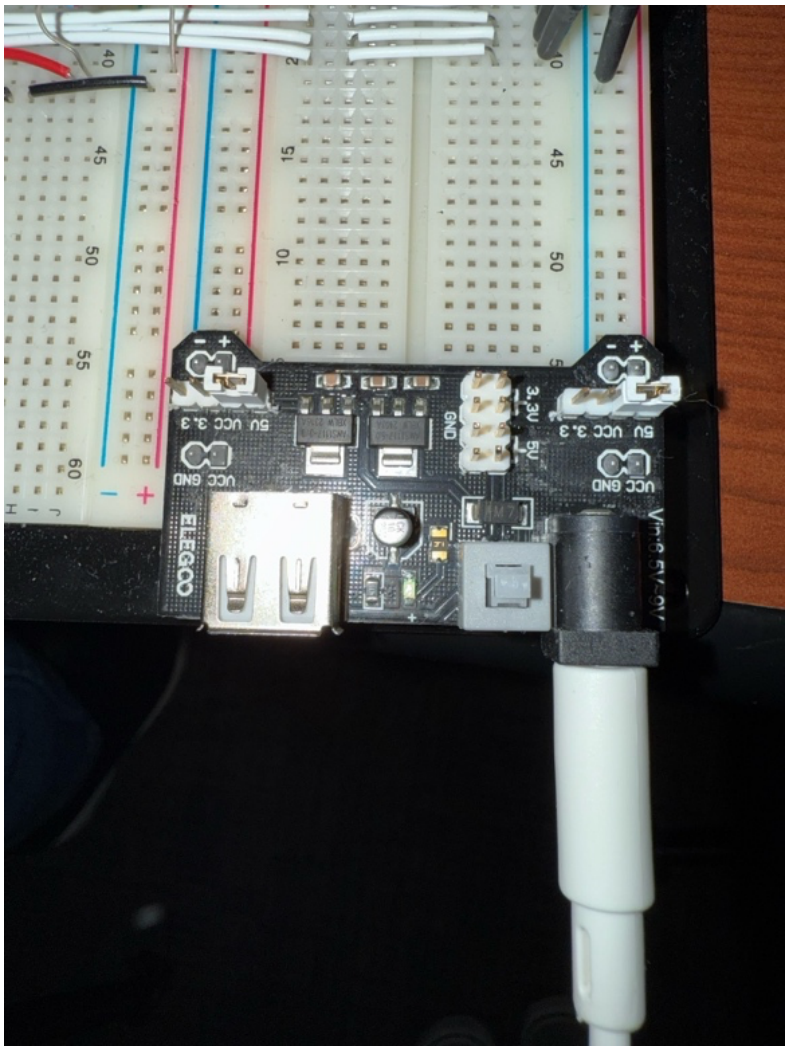
16x2 LCD w/ Potentiometer/Capacitor

- Allows us to provide the temperature, humidity as well as any error messages onto the screen for the user. It is active in all states but when the state is set to ‘disabled’, the LCD will also be disabled.
- Connected to a 10 kΩ potentiometer via V0 and a 100 µF connected to VSS and VDD to prevent voltage spikes.
- Connections to Arduino digital pins 5, 6, 7, 8, 9, 10 for data control and outputs.



PSU Module

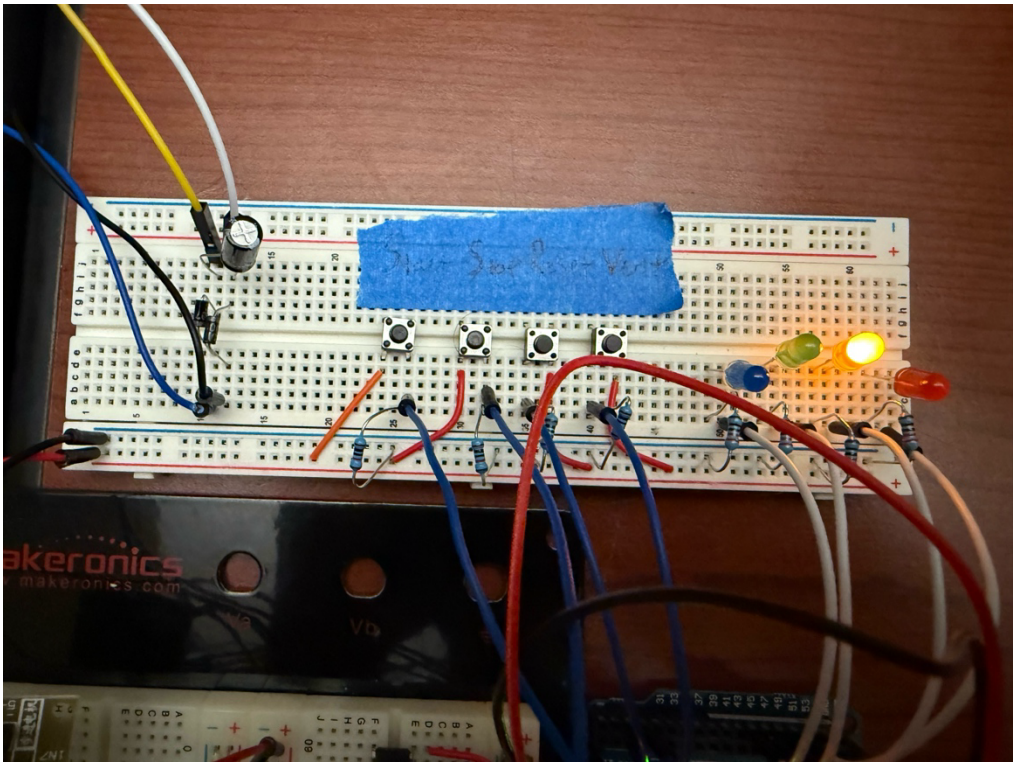
- Utilized to supply a higher current and a stable 9V/1A power to motor, which is safer than Arduino power.
    - Connected to power (5V) and ground via the Arduino.



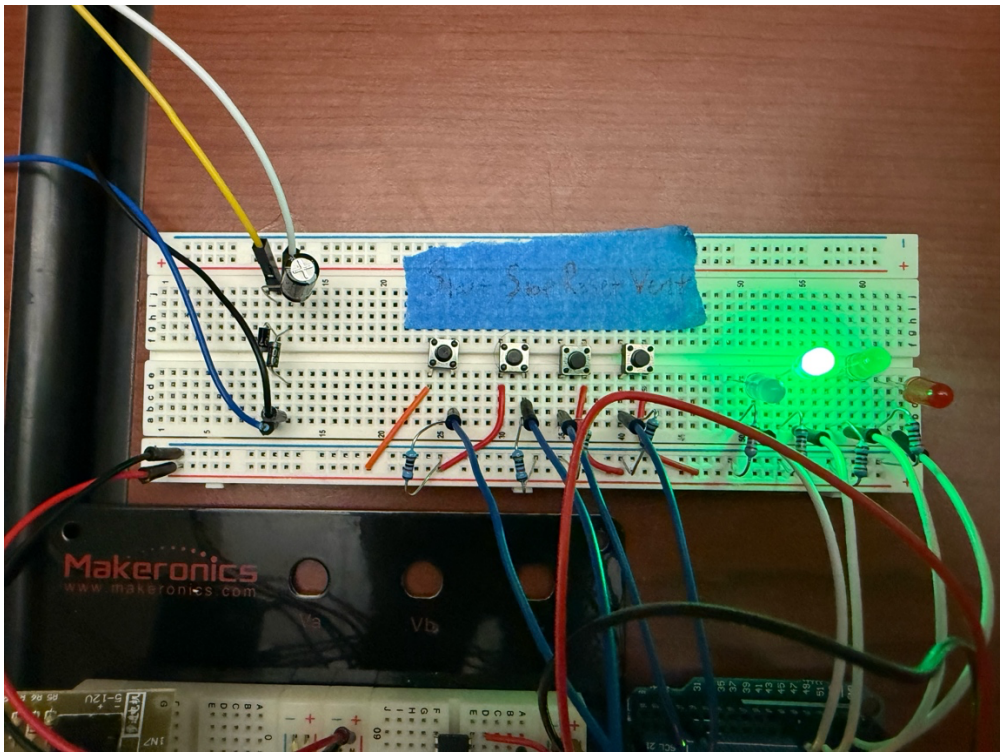
States & State Transition Components

LED / State Indication

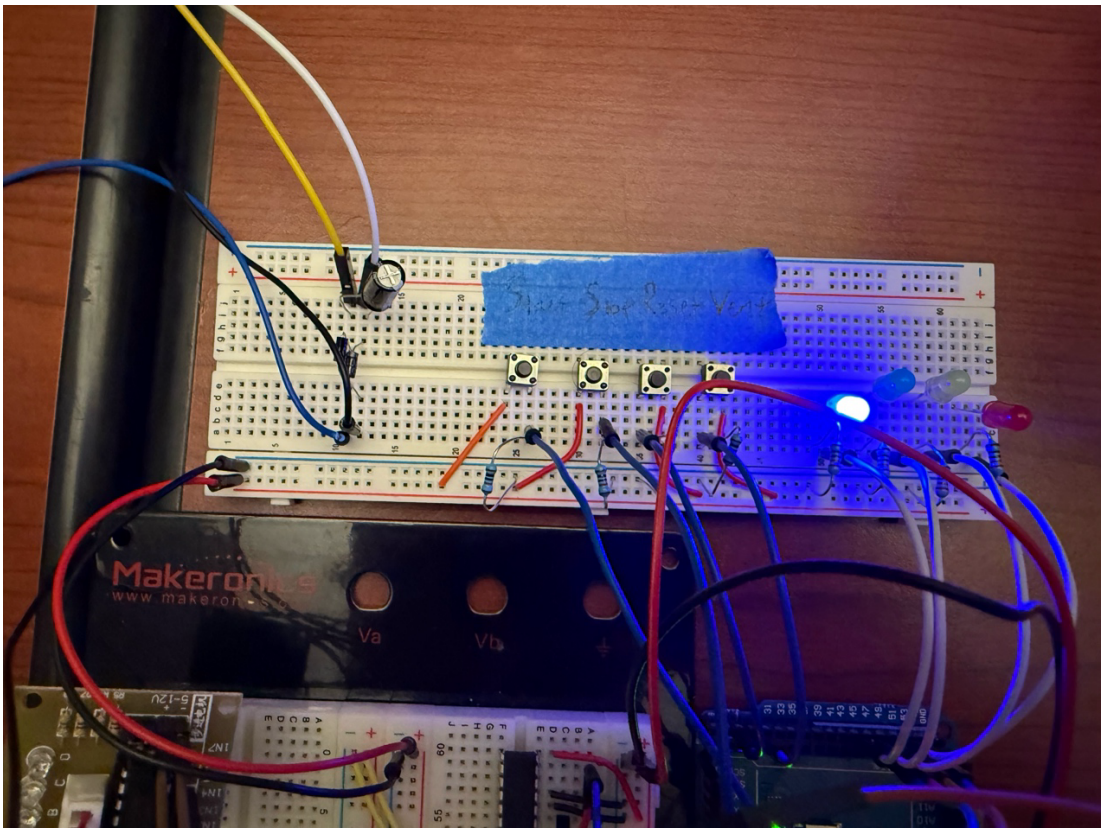
- Yellow LED
    - Connected to digital pin 43.
    - Represents disabled state.



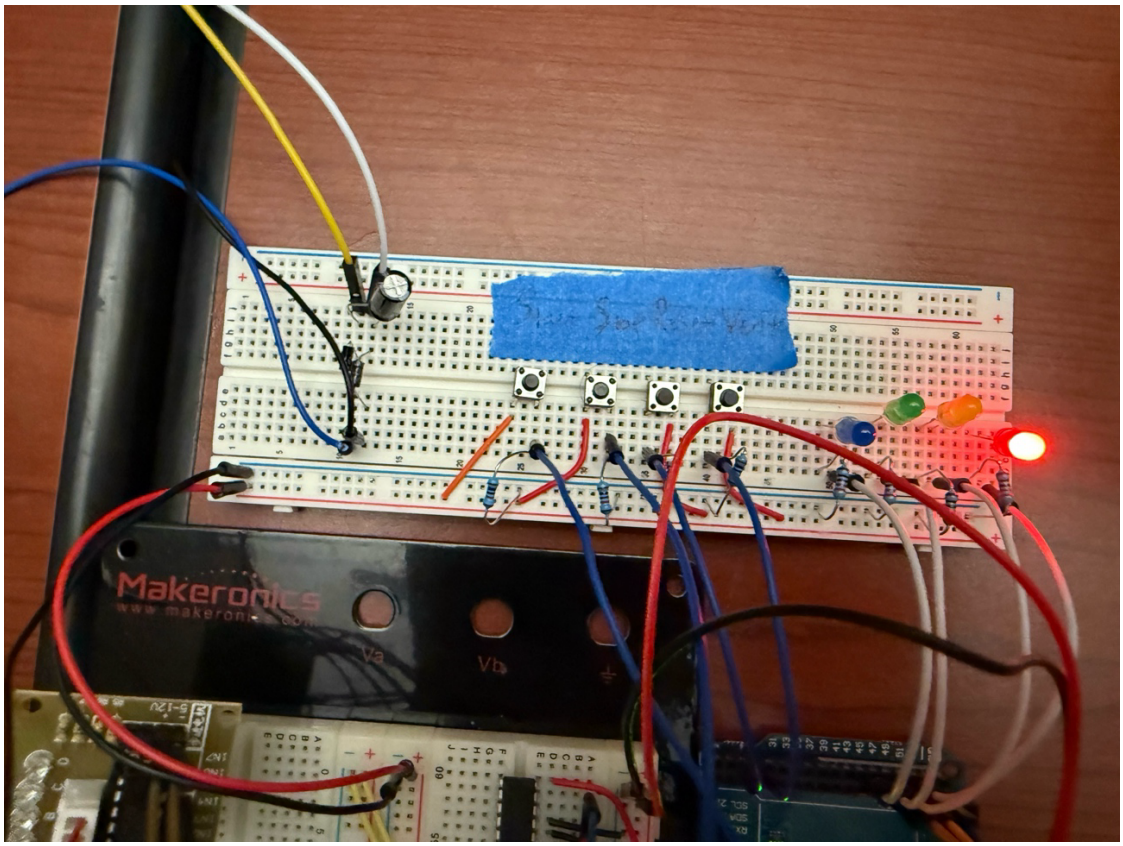
- Green LED
    - Connected to digital pin 45.
    - Represents idle state.



- Blue LED
    - Connected to digital pin 43.
    - Represents running state.



- Red LED
    - Connected to digital pin 49.
    - Represents error state.

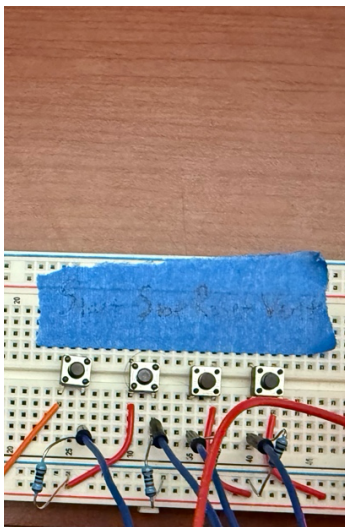


Recall the system operates in four primary states that are outlined by the state diagram and the schematic created:

- **Disabled**
    - The system is off, and the yellow LED light is on.
- **Idle**
    - The system is monitoring the temperature and the water level, and the green LED on.
- **Running**
    - The system's fan is running if the temperature exceeds the threshold, with the blue LED on.
- **Error**
    - The system does not have enough water, and the motor is disabled, indicating that the red LED is on.

Buttons

- **Start Button**
    - Directly changes state from disabled to idle, only when the state is in disabled.
        - Connected to communication pin 19 and ISR-triggered.
- **Stop Button**
    - Directly changes state to disabled only when the state is not disabled.
        - Connected to communication pin 18 and ISR-triggered.
- **Reset Button**
    - Directly changes states from error to idle assuming all requirements and conditions are met to reset only when the state is in error.
        - Connected to digital pin 25 and monitor checked.
- **Stepper Control Button**
    - Allows for the adjustment of the vent angle only when the state is not disabled.
        - Connected to digital pin 27 and monitor checked.



System Overview and Circuit Image

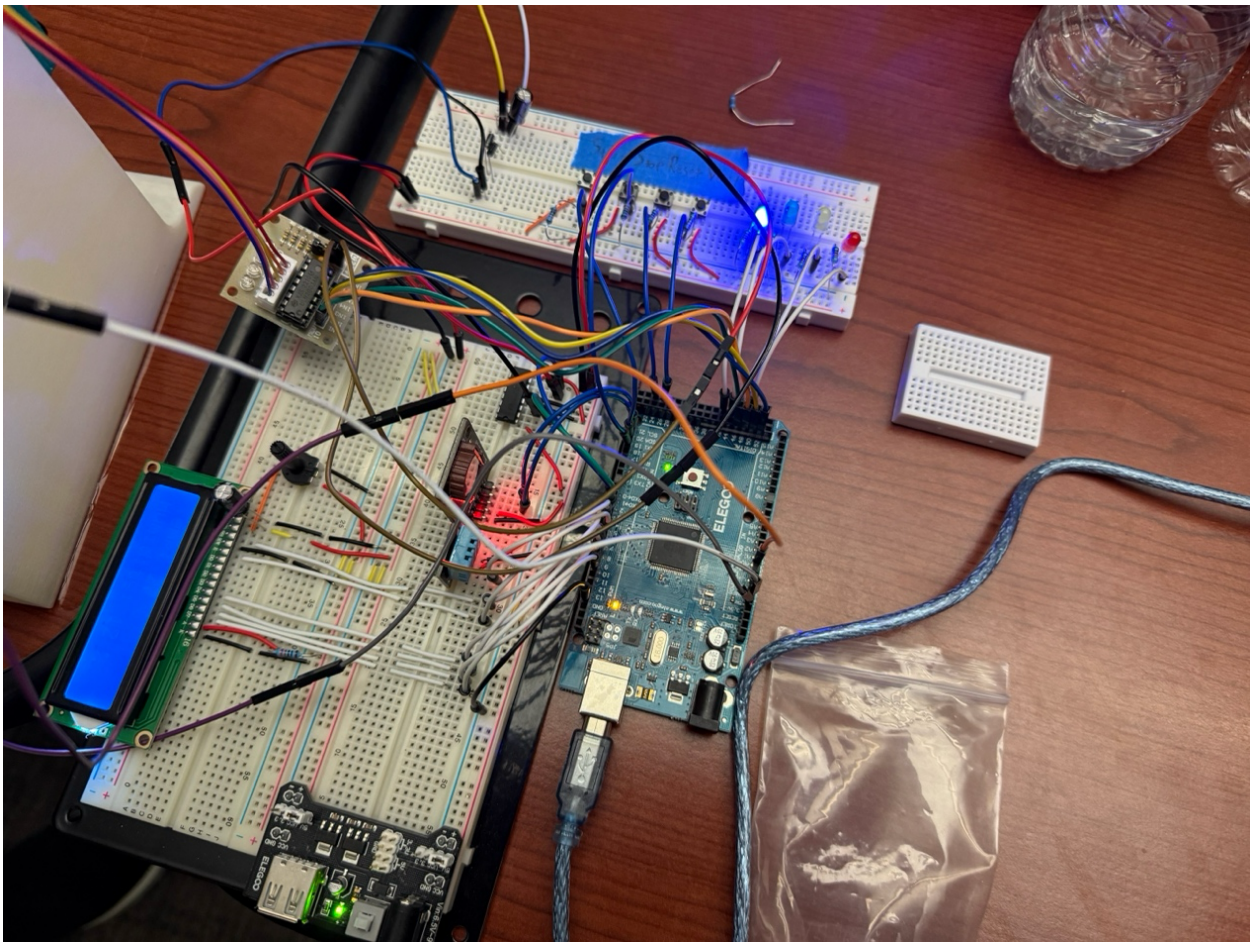


Figure 1 - Clear, Full-Image View of Cooler Circuit

Schematic Diagram

To view the schematic diagram, please click on this link: Schematic Diagram

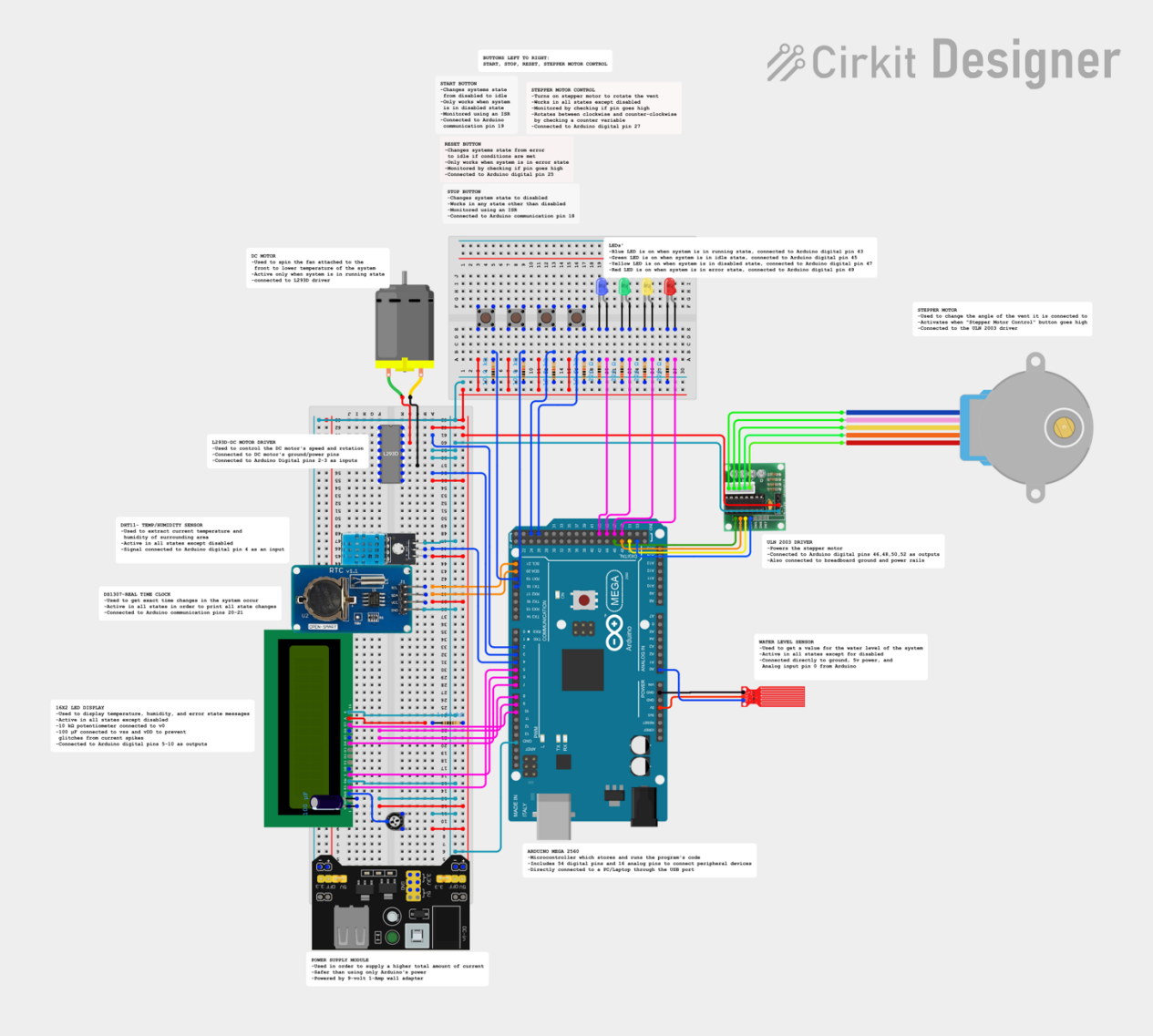


Figure 2 - CirkitDesigner Circuit

If the link does not work, copy and paste this link:
https://app.cirkitdesigner.com/project/41bea0f8-c0e1-45a1-a040-6ed1dc02367c

Specification Sheets

The following sheets were beneficial for the construction of this project:

https://docs.arduino.cc/retired/hacking/hardware/PinMapping2560/

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf

https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf

https://docs.arduino.cc/built-in-examples/

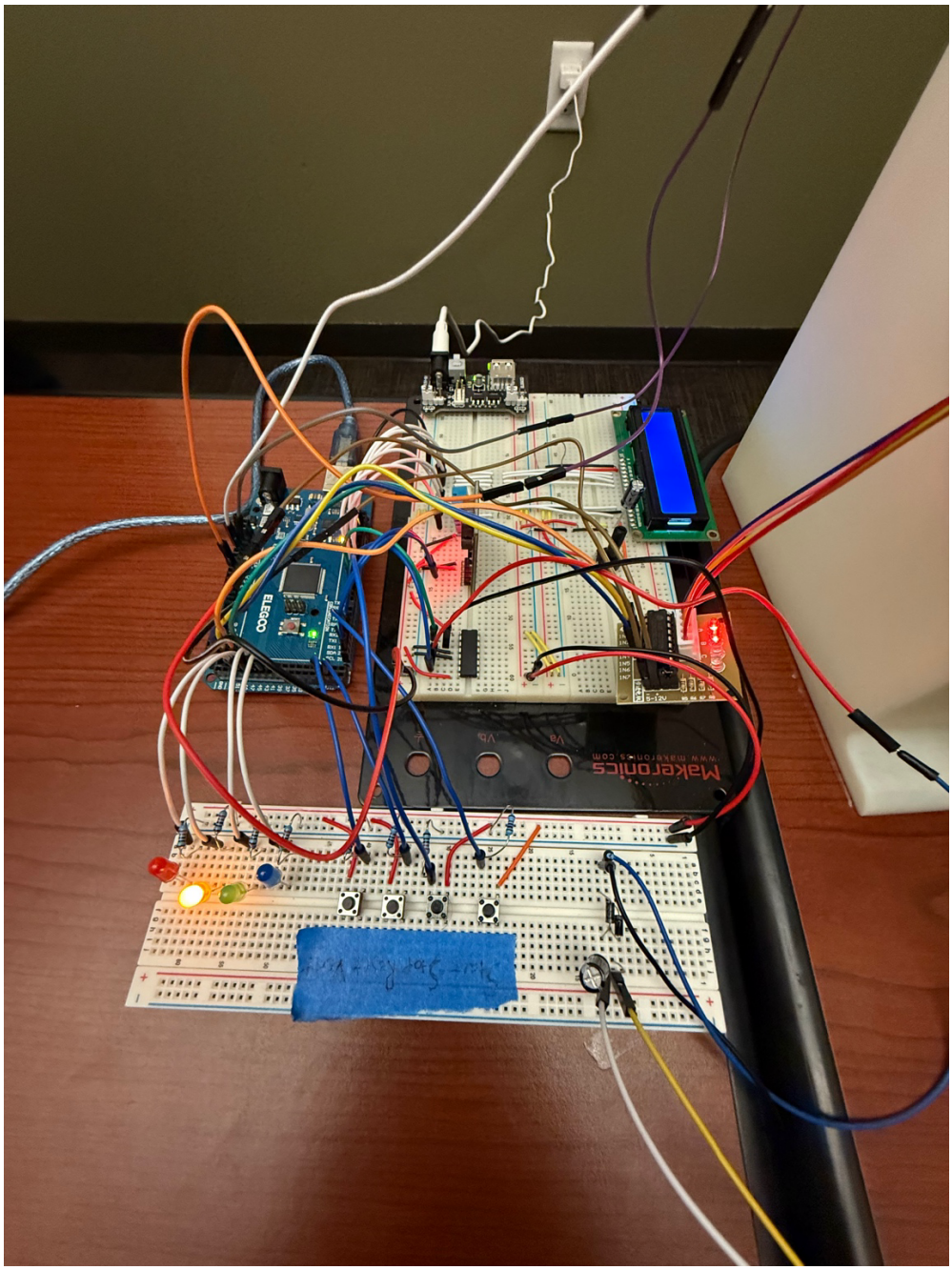
https://docs.arduino.cc/built-in-examples/arduino-isp/ArduinoToBreadboard/

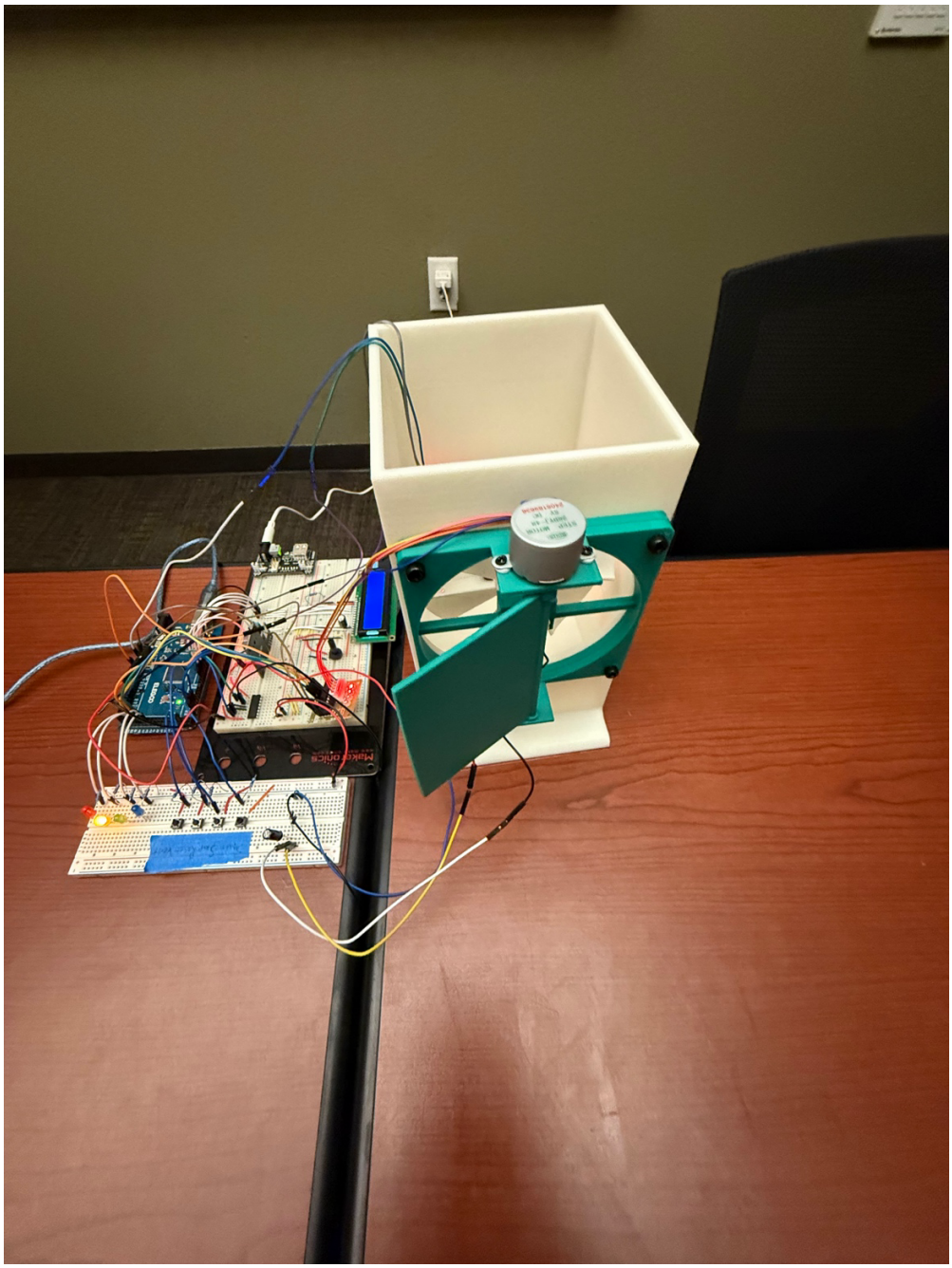
http://4.bp.blogspot.com/-Sj4AEX-ndNo/UeqPDlJo3DI/AAAAAAAAJN4/uN5tE84jGPI/s1600/MEGA2560.jpg

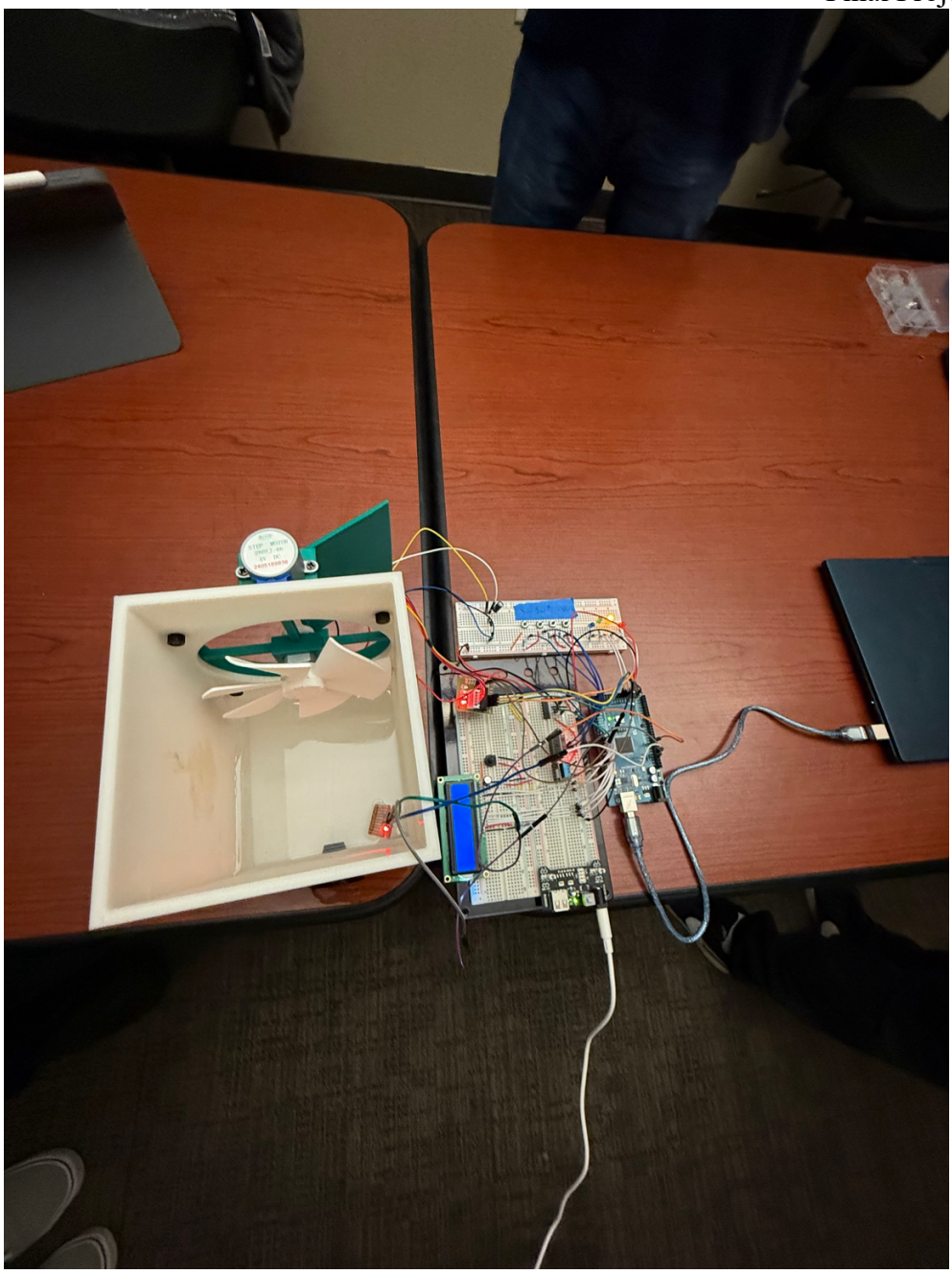
https://www.arnabkumardas.com/arduino-tutorial/adc-register-description/

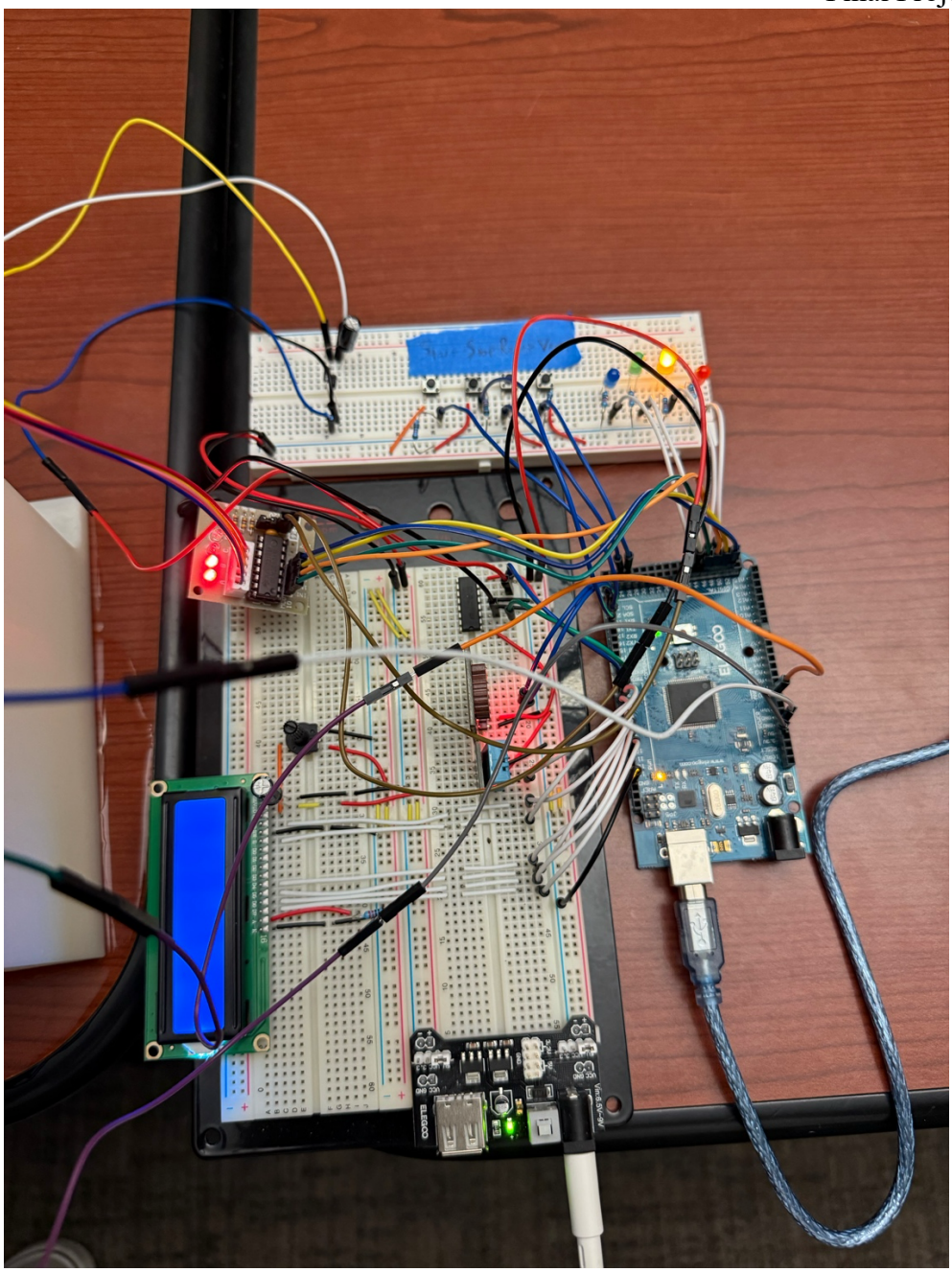
https://lastminuteengineers.com/water-level-sensor-arduino-tutorial/

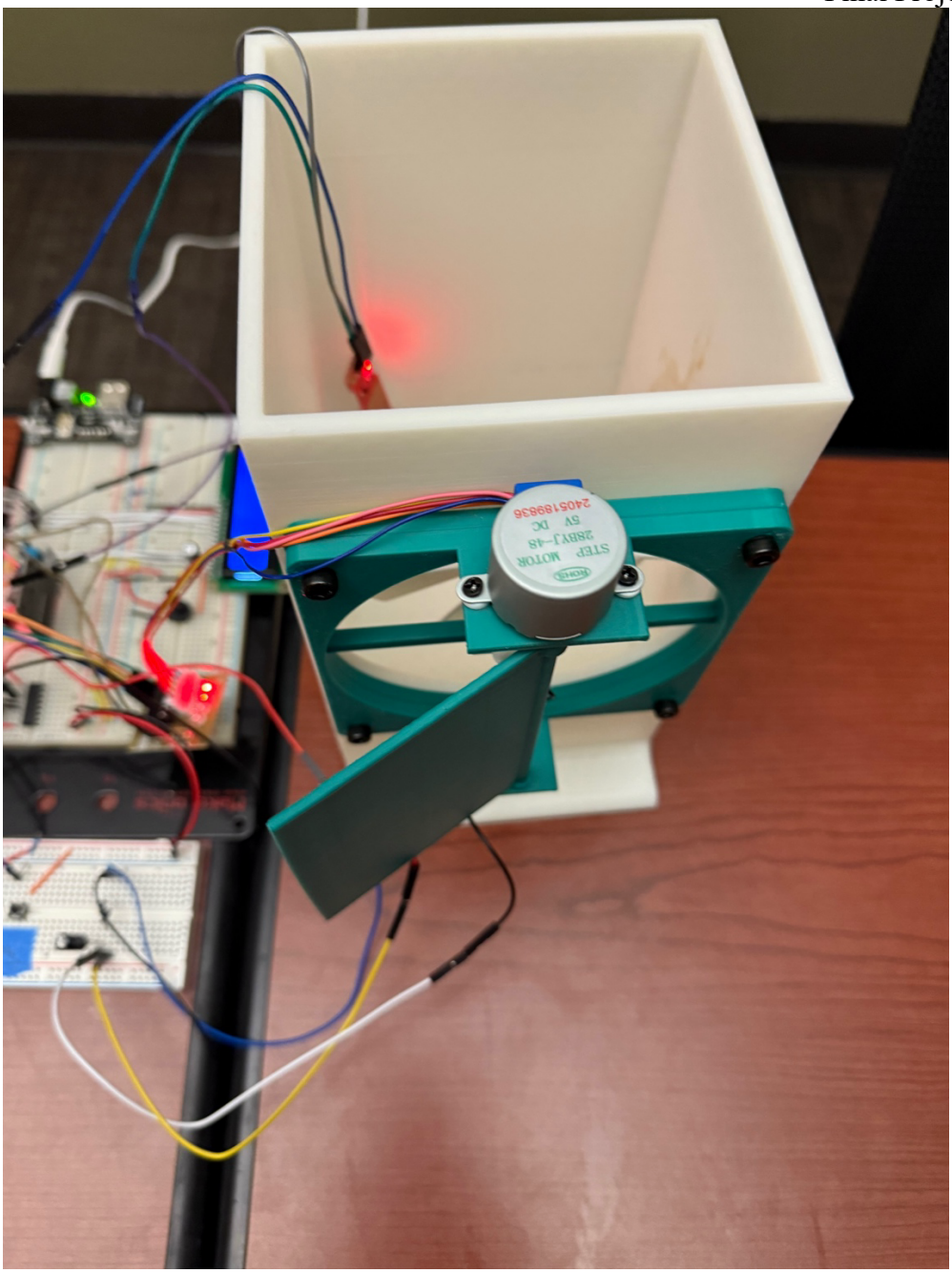
System Demonstration










STEP MOTOR
28BYJ-48
5V DC

System Demonstration Video

To view the video, please click here: System Demonstration Video

If the link does not work, copy and paste the link:

https://nevada.box.com/s/sombovxdms02x6y8vtccuijsdb9f6vfj

Submission Links

Video: System Demonstration Video

GitHub: GitHub

Video: https://nevada.box.com/s/sombovxdms02x6y8vtccuijsdb9f6vfj

GitHub: https://github.com/Brysen-Ross/CPE301_FinalProject_Group21