# CPE 301.1003 Final Project Overview

By Jeremy Buhain, Jonathan Chi, Cody Long, Jessica Nam

## Objective

The goal of this project was to produce a Swamp cooler cooling system that operates by monitoring temperature and humidity levels; adheres to the project requirements; and utilizes the Arduino Mega 2560 and the components inside the lab kit.

## Design Overview

The team implemented a functional swamp cooler utilizing the Arduino ATMega 2560 microcontroller in an interactive system containing the following components built from the Arduino kit: a water level sensor, a vent direction controller using a potentiometer and stepper motor, a LCD display for outputting messages, a temperature/humidity sensor, a fan motor, LED, and a real time clock for event reporting. The clock was used to report the times of the state transitions and changes made to the stepper motor.

Our swamp cooler measures the temperature and humidity levels of its surroundings and output the values on a LCD display. The LCD continuously refreshes every second with new temperature/humidity values. Only the devices being actively worked with will be reflected on the LCD display and temperature and humidity sensor. These values are grabbed by using the DHT library.

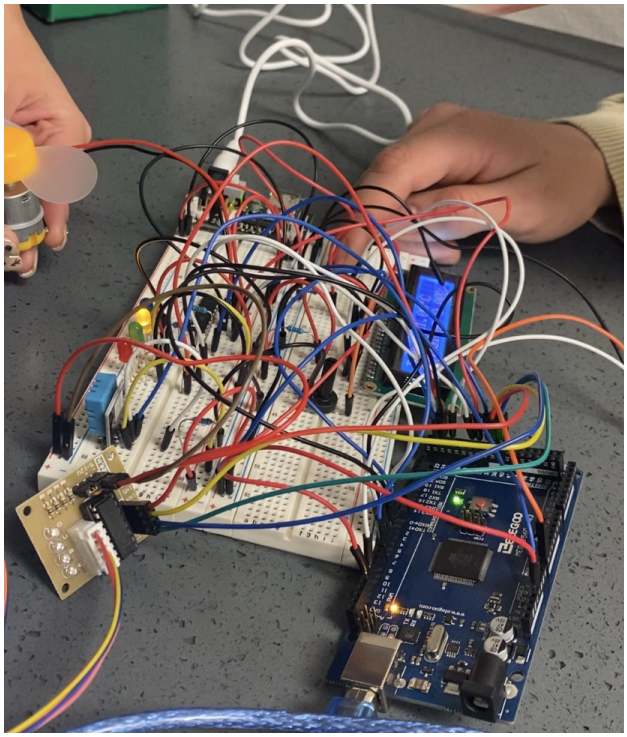
There are five potential states of the Swamp cooler; DISABLED, IDLE, ERROR, and RUNNING. Based on the values extracted from the water level sensor, the system will switch through these states. The water level sensor component was implemented so that the power of the water level sensor is set to digital pin 6 and set to high or low as needed. In the program, we defined pins and declared a variable to store the water level. The water level determines whether or not the fan is on or off. There is also a custom function that reads new water levels every 10 ms.

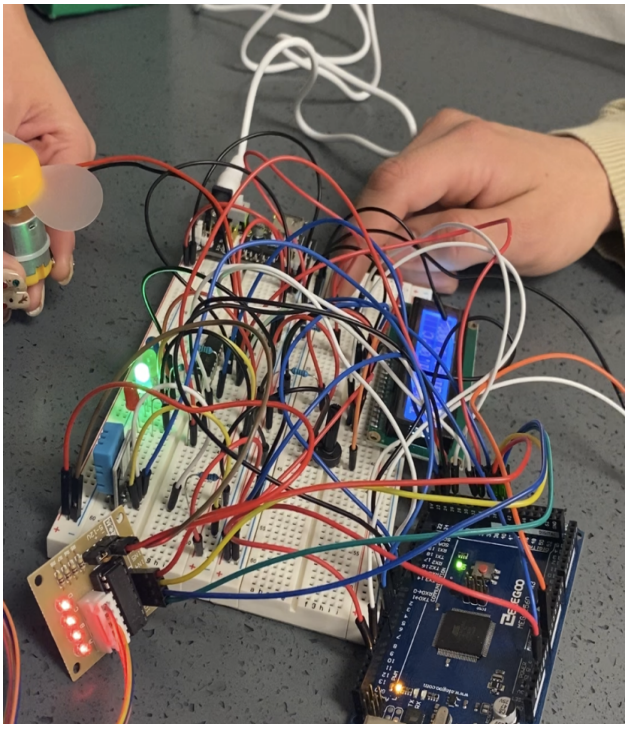
We then implemented the stepper motor alongside the potentiometer to control the vent direction. We did this by making each state transition a 90 degrees rotation with disabled state being zero degrees, idle being 90 degrees, running being 180 degrees, and error state being 270.

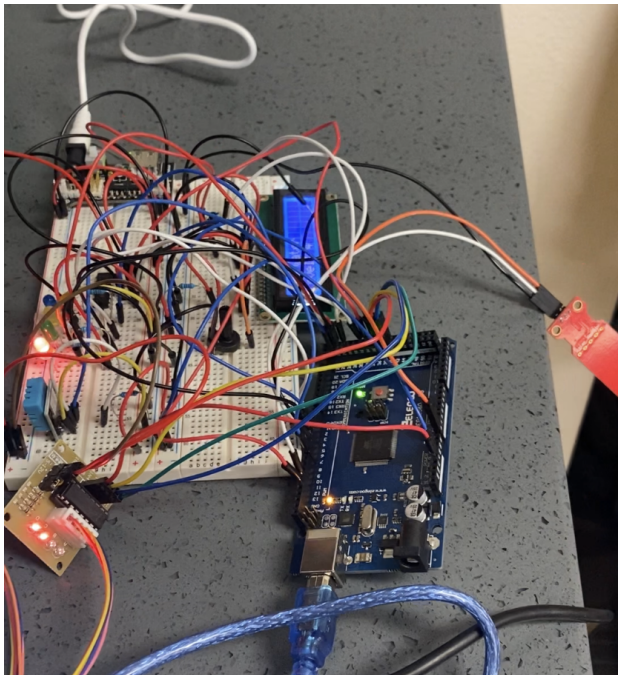
## Constraints

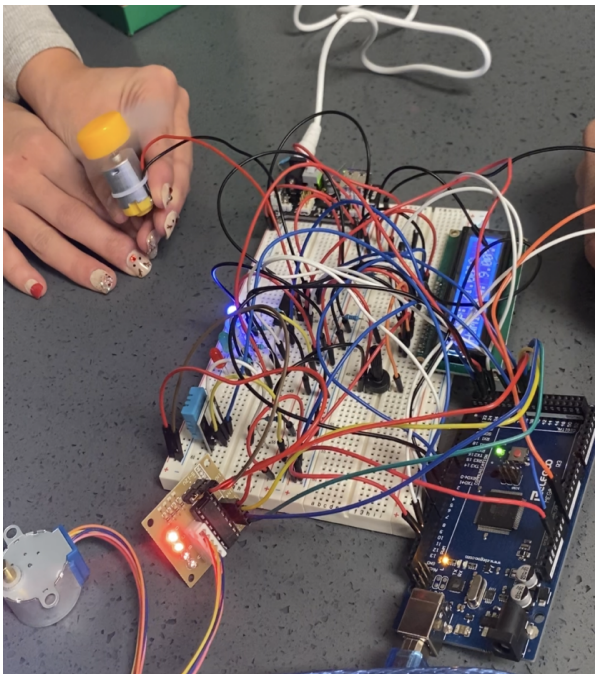
There were several constraints that limited the system to certain capabilities or affected our methods of implementation. First, we could only work with the Arduino Mega 2560 and the components inside the lab kit. Second, we were unable to use library functions such as pinMode and digitalWrite. Therefore, we resorted to the Arduino Mega 2560 Pinout and the specification sheet to set our pins.

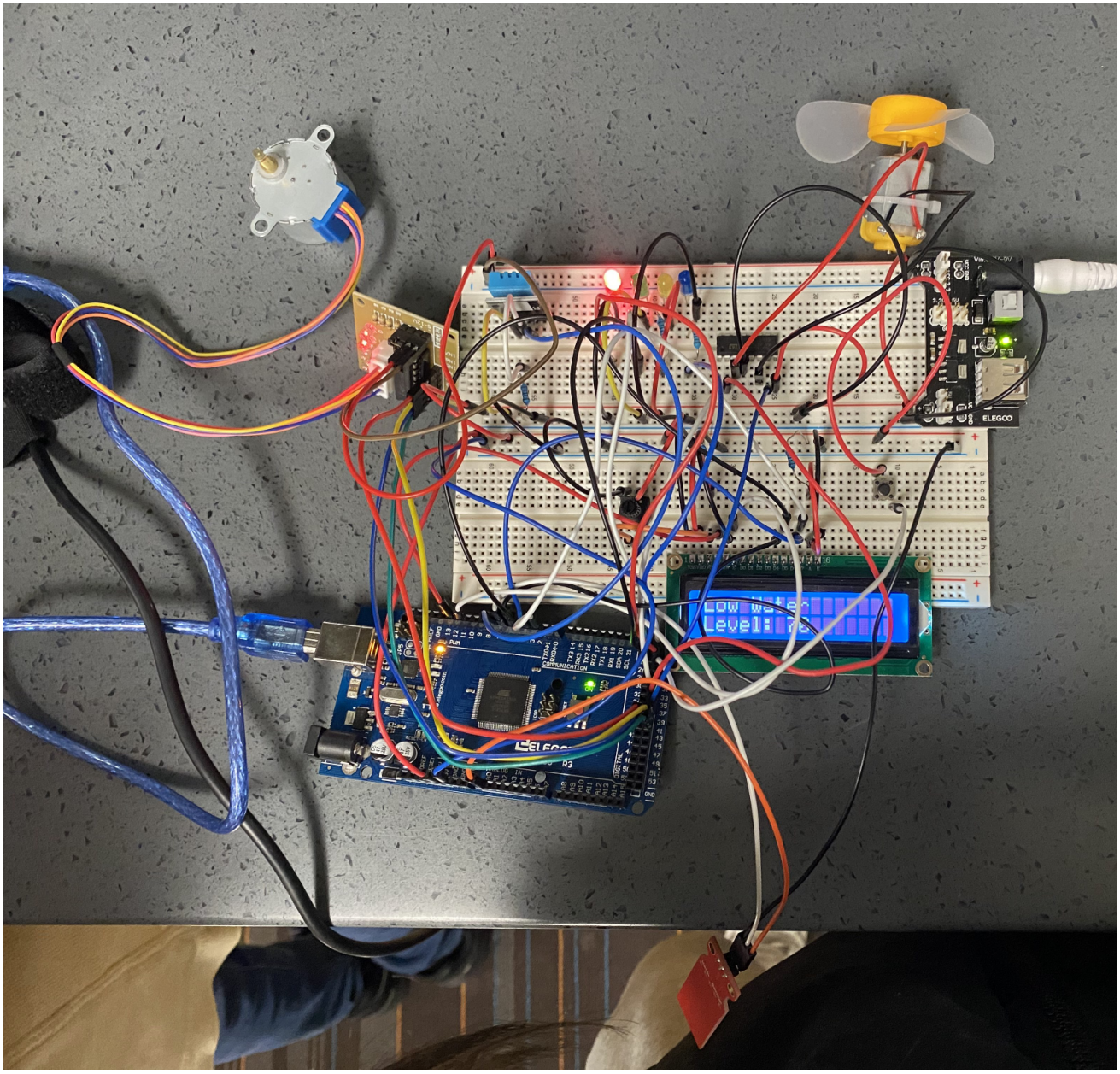
Pictures

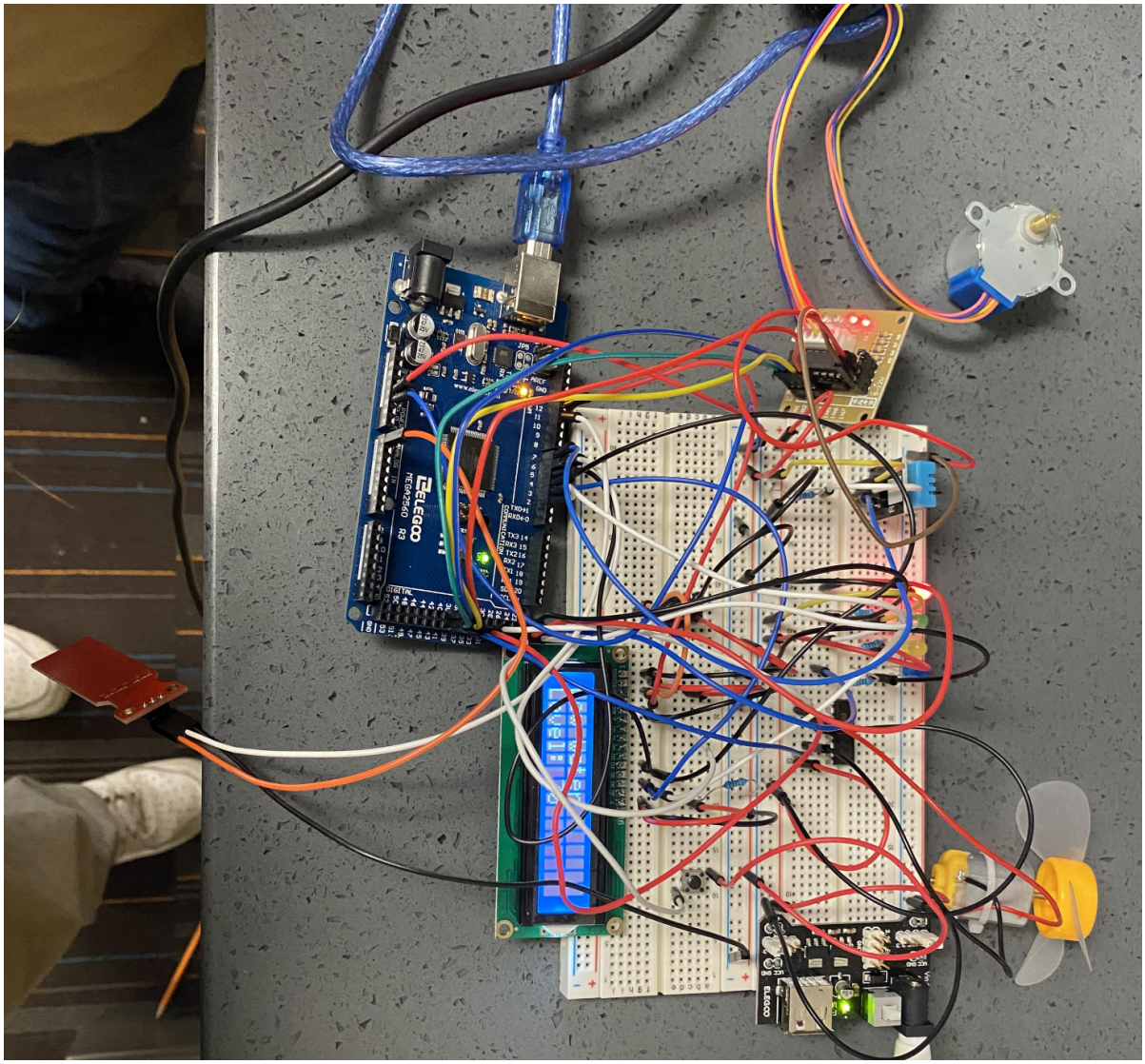








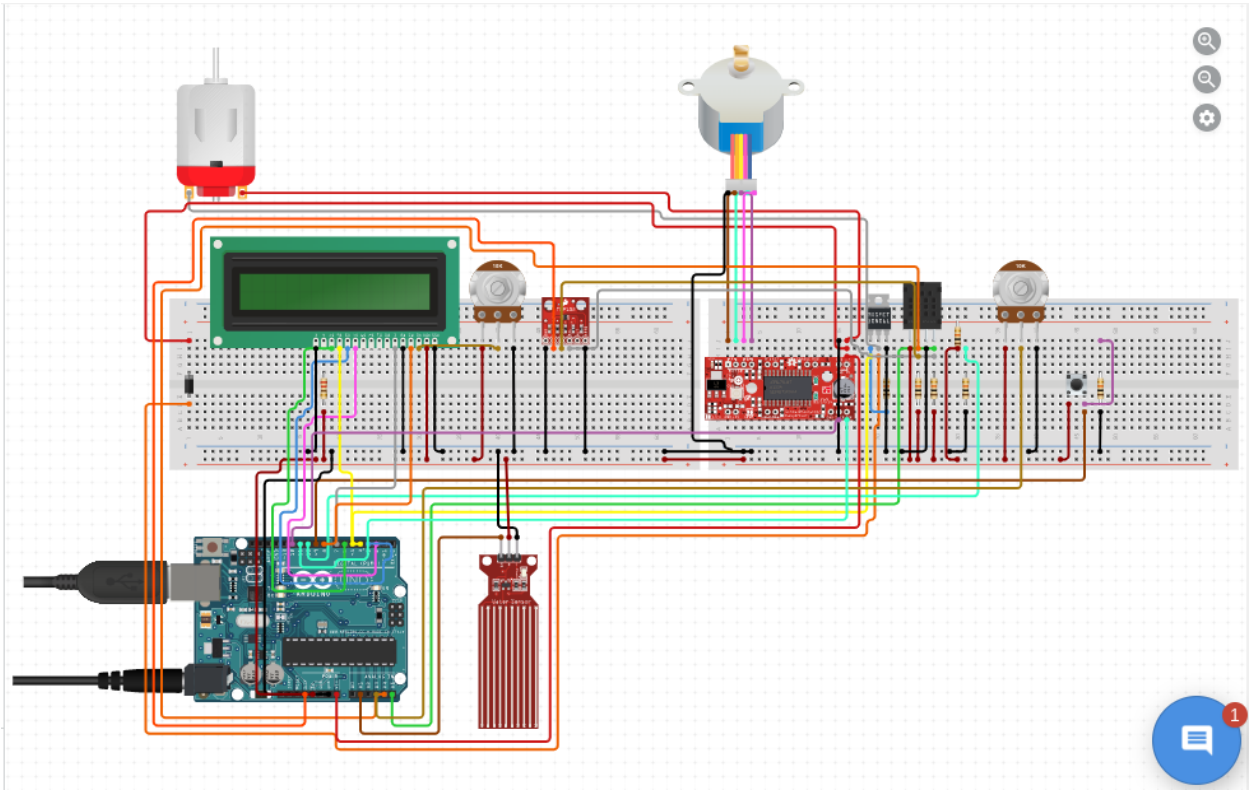

ELEGOO



## Video link

https://youtube.com/shorts/Gr-83FAhf5E?feature=share

## Complete Schematic (made in circuito.io)



## Specification sheets

- Arduino Mega 2560: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
- Arduino Mega Pinout: https://www.electronicshub.org/wp-content/uploads/2021/01/Arduino-Mega-Pinout.jpg

## Github Repository

https://github.com/LongCody/FinalProject