Intro

The purpose of this project is to create parts of a swamp cooler and code the circuit to work as a swamp cooler would. To achieve this I used a motor with a fan attached, a step motor for vent controls, a water sensor, a real time clock module, a potentiometer, buttons, LCD, and humidity and temperature sensor.

Experiment

Fan Motor - This component was meant to represent the fan of a swamp cooler. It would only turn on in the Running State of the machine. This component works by attaching it to ground and giving it power based on the colored wires. Even so you can reverse these wires to reverse the direction of the air. I needed to do this to make the air blow away from the motor.

Step Motor - This component was used to represent closing and opening vents. At any State of the machine you could either press the open or close button depending on if the fan was already opened or closed. This component needs to be connected to ground and power, then after it needs to be connected to 4 pins for code functions. Through the use of a library function I can connect the four pins to a stepper object. From there I can use the .step function of the object to set the amount of steps for the motor to turn which the step motor by default uses 2038. Then I can set the speed of step motor through .setSpeed which I chose to use 10 and make the motor turn ¼ a revolution to kinda represent how much it would need to turn if it was an actual vent.

Water Sensor - This component is used to sense the level of water in the swamp cooler. This component can tell how much of its designated sensor space is covered in water which it then sends a voltage value representing its % of coverage. This sensor would be active on each state except Disabled. Using this I set a value of 200 in digital as the threshold value of when the water is too low. If the water level is ever too low in the Idle or Running State the code will transition to the Error State until more water is added and the reset button is pressed where it transitions back to the Idle State or until the stop button is pressed where it transitions to the Disabled state.

Real Time Clock - This component was used to track the time stamps of each state transition. This module piece is connected to power, ground, and the SDA and SCL pins for it to connect with the Arduino for it to be used with code. After this I can utilize this module through library functions to create an internal clock in the system using seconds, minutes, days, weeks, months, and years. I specifically only used seconds, minutes, and days and started the time at 0:0:0. The code would then take whatever the current time is and display it to the monitor whenever a state transition occurred.

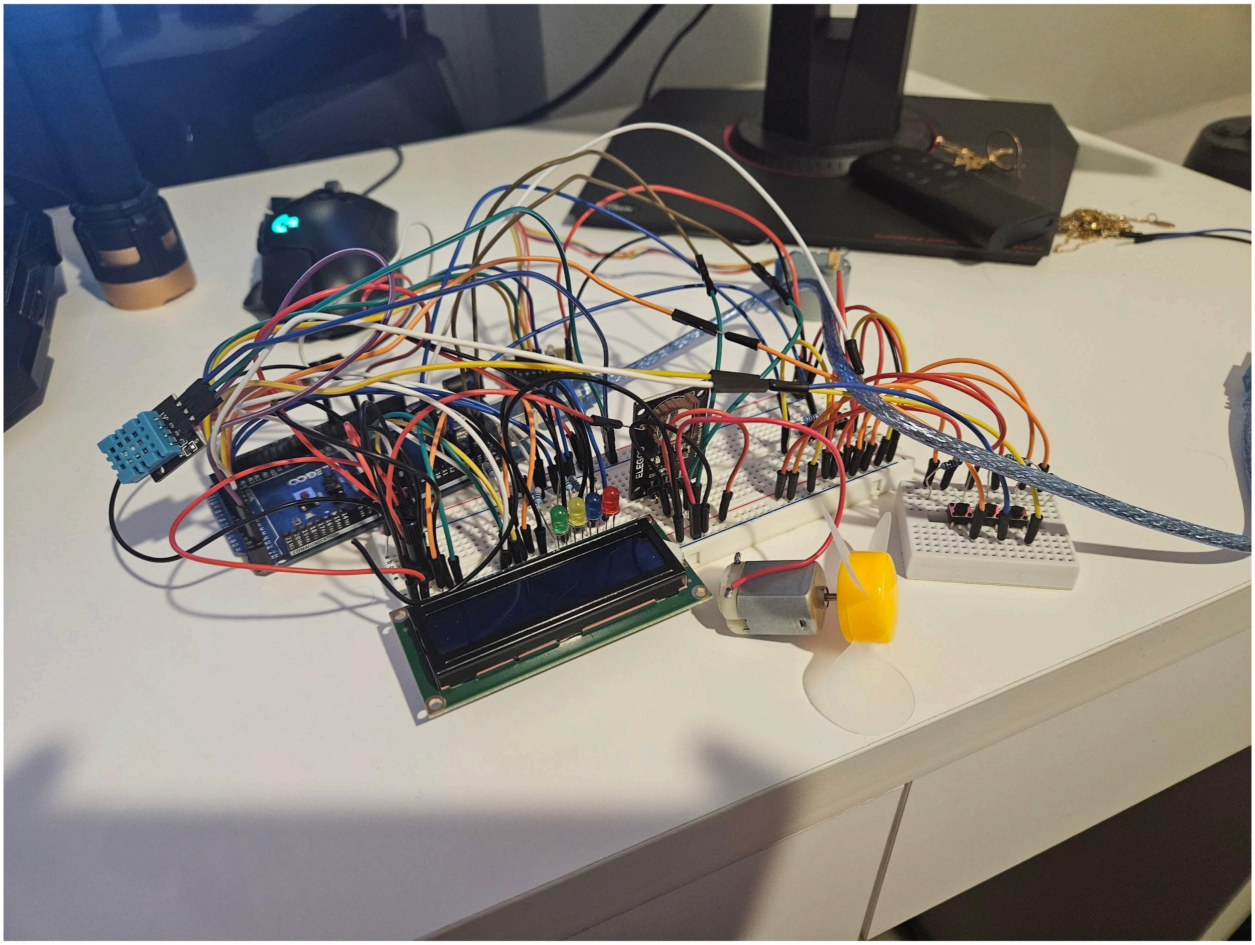
Potentiometer - This component can be altered to provide a different amount of resistance. Using this I can connect the output of it to the display of the LCD, the VO pin, to then alter the brightness of the display easily by turning the potentiometer.

Buttons - Buttons were used 5 times, 2 for controlling the step motor and 3 times for the overall systems Start, Stop, and Reset button. When in the Disabled State the user can press the Start button for it to transition to Idle. When not in the Disabled State the user can press the Stop button to transition to the Disabled State. Lastly the Reset button can be pressed during the Error State to attempt to transition back to the Idle State.

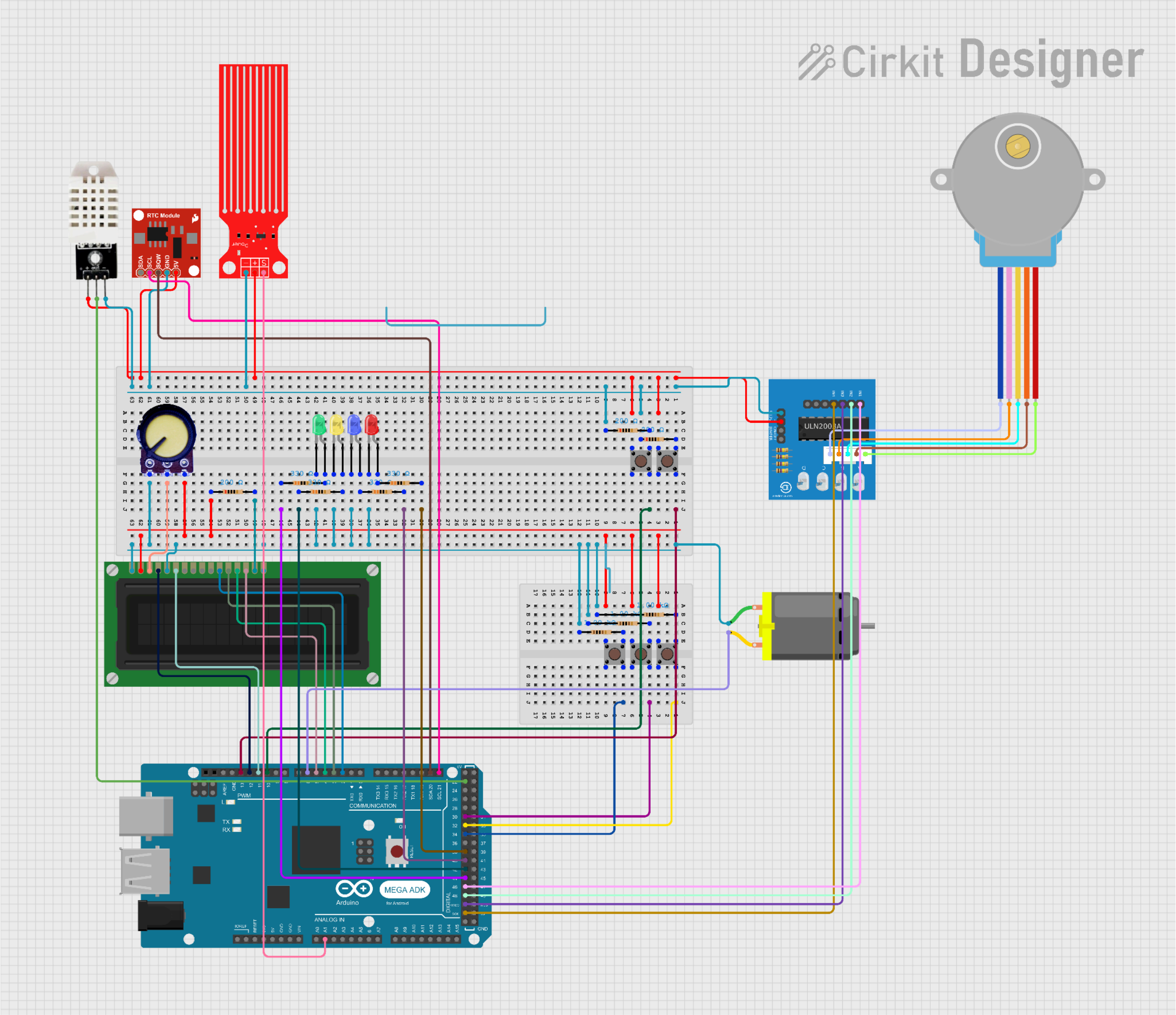
LCD - This display has various connections to pins, power lines, and ground lines. After connecting them all properly I can use the display, RS, and EN pins with a library function to set up the display in the code. At this point I can use the code to both display the error message in the Error State or display the temperature and humidity levels during the Idle and Running States.

Humidity and Temperature Sensor - This component is used to track the temp and humidity in the room and connect it back to the arduino. By connecting this to power, ground, and an out pin, I can then make a DHT object using this components library to then see what temp and humidity it senses and then display it to the LCD. This was also used to transition from the Idle State to the Running State when the temperature is too hot and back when the temperature is too low.

Finished Circuit



Circuit Diagram


Cirkit Designer
RTC Module
SDA
SCL
SQW
GND
5V
ULN2003A
Arduino
MEGA ADK
PWM
COMMUNICATION
ANALOG IN
DIGITAL
TX
RX