Sean Masterson

CPE 301

Instructor Bashira Anima

Final Project Overview

Link to Github Repo: https://github.com/smasterson64/Sean-Masterson-CPE-301-Final

Link to Demonstration Video: https://youtu.be/BPJbFkFSoxA

The goal of this project was to create a system that emulates a swamp cooler using various components of our Arduino starter kit. This project combines many of the skills and learned from previous labs. It requires the use of a power supply module, and an AC adapter that outputs 9v DC. The system consists of four main states:

1. **Disabled:**
   This is the state that the system enters from. It is considered the OFF state of the system. The Yellow LED indicates the Disabled state. No temperature or water level readings will be made in this state. It can be changed to the Idle state with the use of the START button. The vent motor can be controlled with the VLEFT and VRIGHT buttons.
2. **Idle:**
   This is the neutral state of the system. The Green LED indicates the Idle State. Water level readings and temperature readings are made in this state. If the system measures a temperature above 25 degrees Celsius, it will change its state to Running. If it detects that the water level is below 150 on the sensor, it will change the state to Error. If the STOP button is pressed, it will change the state to Disabled. The vent motor can be controlled with the VLEFT and VRIGHT buttons. The ambient temperature and humidity are displayed on the LCD.

3. **Running**
   This is the ON state of the system. The Blue LED indicates the Running state. The fan motor runs in order to cool down the area. The fan motor is turned off in every other state. Water level readings and temperature readings are made in this state. If the system measures a temperature above 25 degrees Celsius, it will change its state to Running. If it detects that the water level is below 150 on the sensor, it will change the state to Error. If the STOP button is pressed, it will change the state to Disabled. The vent motor can be controlled with the VLEFT and VRIGHT buttons. The ambient temperature and humidity are displayed on the LCD.

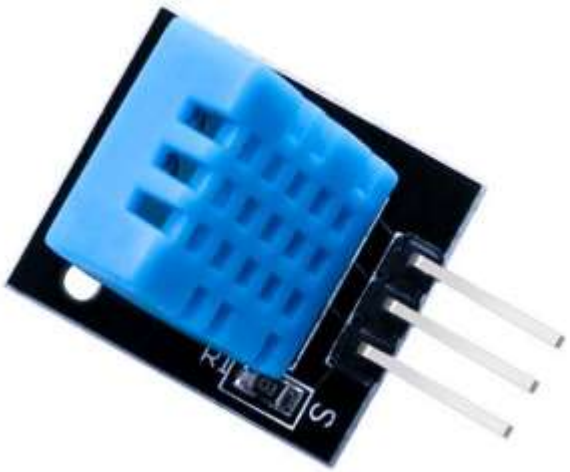
4. **Error**
   This is a state that occurs when the system can no longer run due to the lack of water. The Red LED Indicates the Error state. The vent motors do not function in this state. An error message is displayed to the LCD. If the STOP button is pressed, it will change the

state to Disabled. If the water is refilled and the RESET button is pressed, it will change to the Idle state.

This project utilizes the following components:

1. **The DHT11 Temperature and Humidity Sensor**

   This component will measure and output the relative ambient temperature and humidity. This component makes use of the DHT library.



2. **The Water Level Detection Sensor Module**

   This component will detect how much of its surface area is in contact with water and will output a voltage based on how much area is covered. The more water touching the sensor, the higher the voltage that is output. This component makes use of the ADC concepts we learned in class.

3. **The Real Time Clock Module**

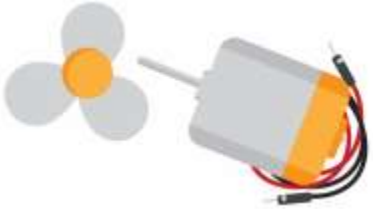
   This component, after being calibrated, will keep track of real time. Since it is battery powered, it can keep track of time even when it is not connected to the Arduino. I used the uRTCLib library for this component.

4. The LCD Display

This component is similar to the Seven-Segment display module we previously used in class, however this one can display a much larger set of characters, and can be formatted for two rows of text. The LiquidCrystal library was used for this component.

5. The DC Motor

This component is a simple motor that can be configured to spin either clockwise or counterclockwise. It made use of the GPIO concepts learned in class.



To operate this component, two additional components were needed.

5a. The Power Supply Module

This module will supply a stable source of either 5v or 3.3v, depending on where the jumper is placed on each track. This was used in order to avoid damaging the Arduino.

5b. The L293D Chip

This component allows for fine control of the motor. With it, you are able to control the motor's speed and direction.

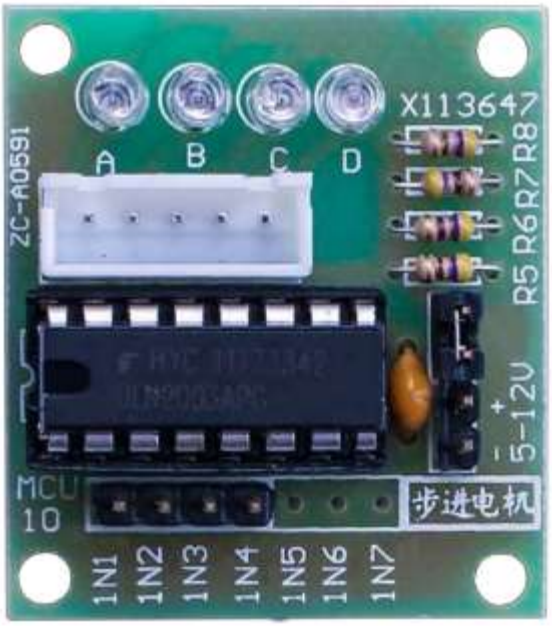
6. The Stepper Motor Module

This component is a special type of motor that allows for fine control. You can directly set how many "steps" the motor takes for each turn. This module makes use of the Stepper library.

This component is operated by the use of the following component:

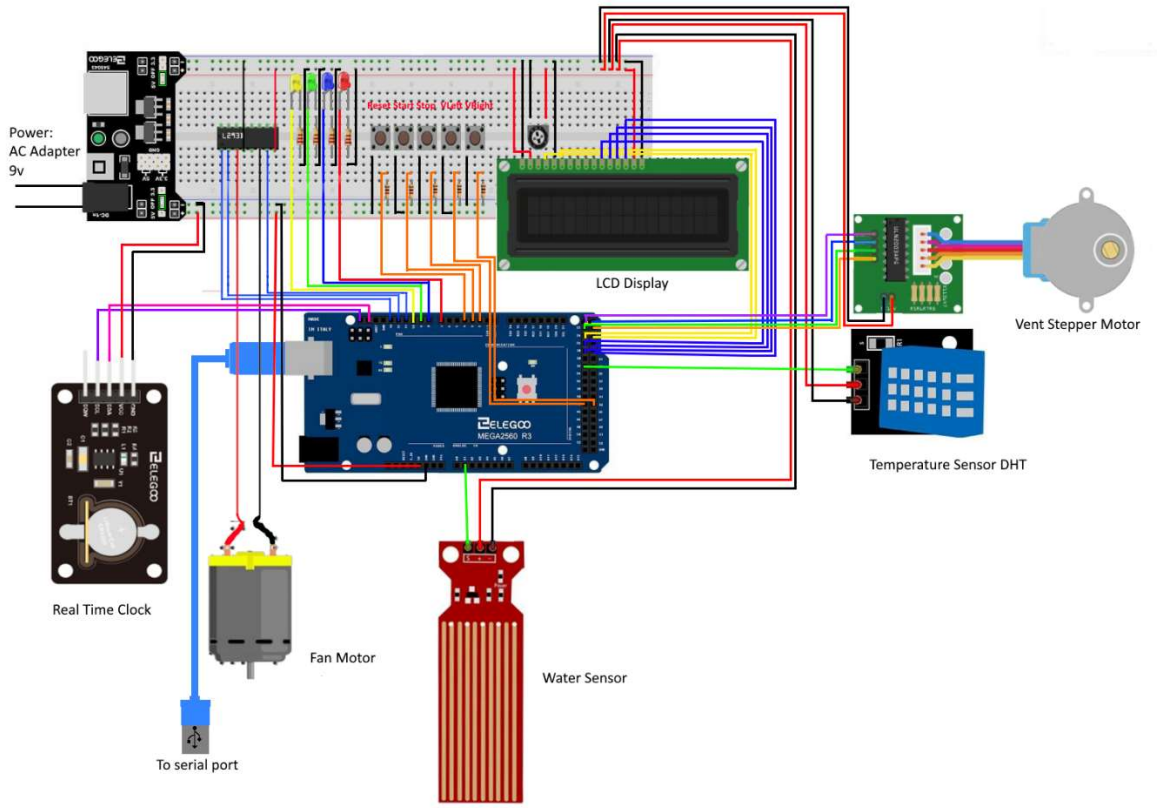
6a. The ULN2003 Driver Board.

This is the driver that actually allows the Arduino to interface with the motor.



In addition to all of these components, standard LEDs and buttons are used to utilize the GPIO concepts learned in class.

SCHEMATIC FOR PROJECT:



See “diagram.png” in the Github Repo for a closer look.

Photo 1: picture of swamp cooler system before connecting to computer for testing.

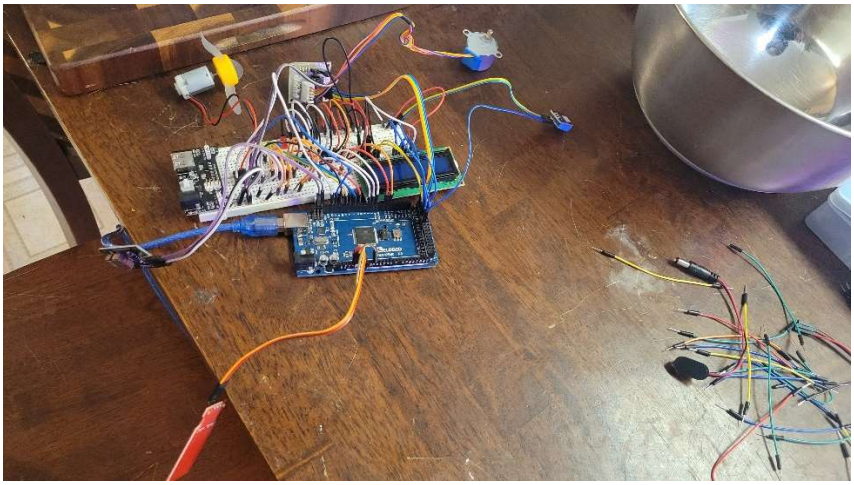


Photo 2: picture of swamp cooler system in DISABLED state

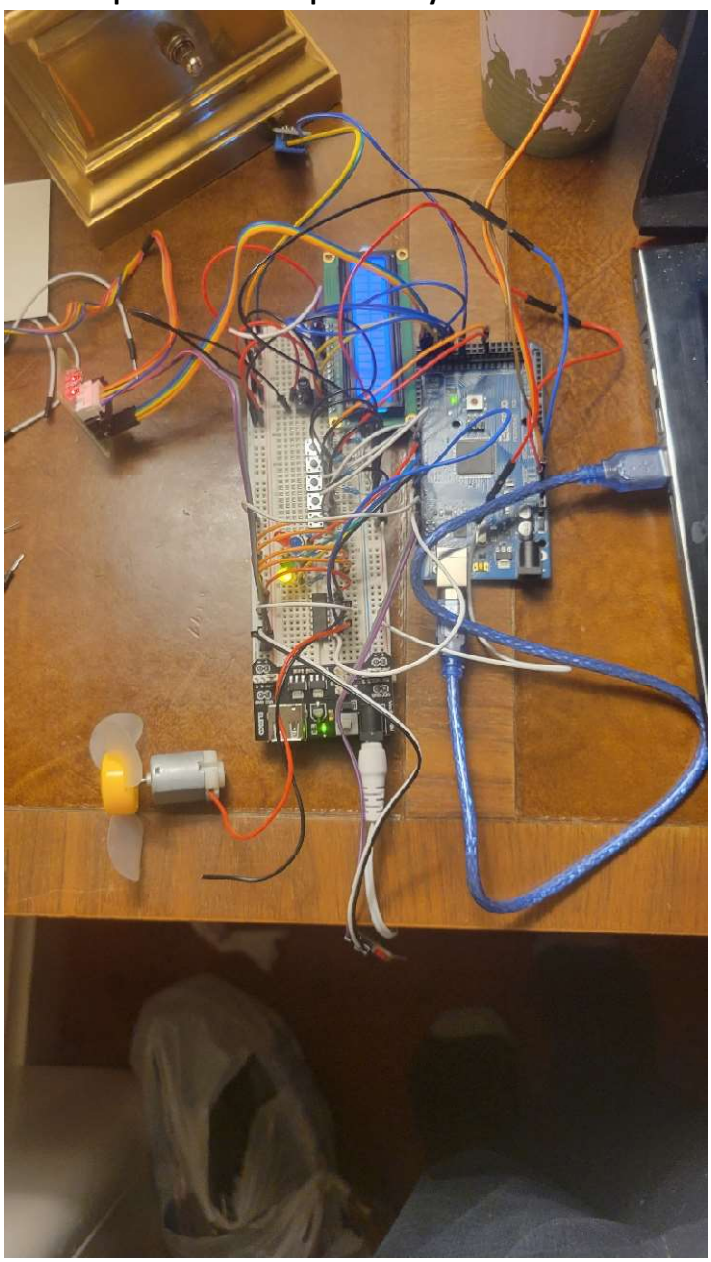


Photo 3: Picture of swamp cooler system in IDLE State. Note the temperature and humidity are displayed on the LCD.

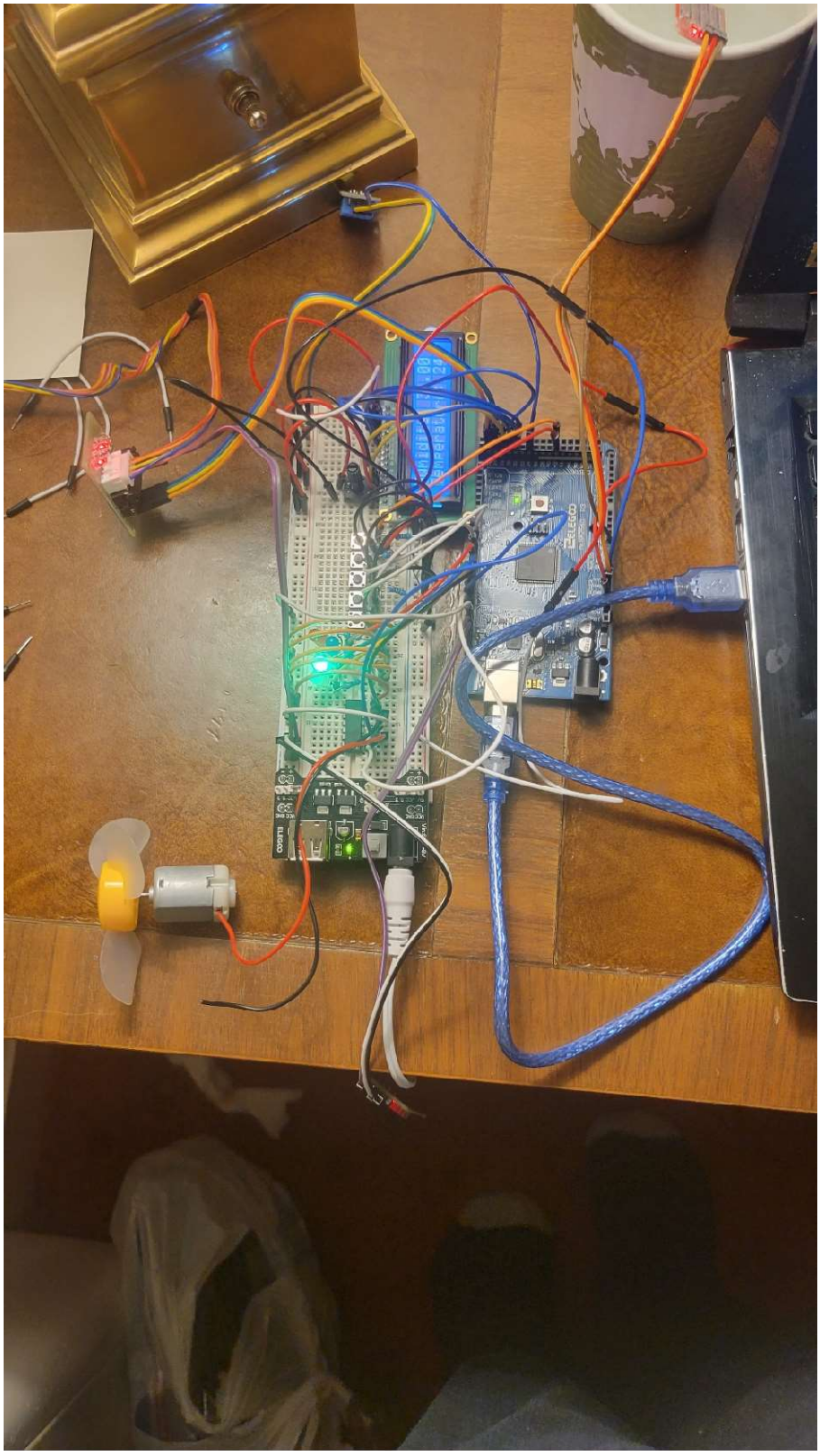


Photo 4: picture of system in RUNNING state. Temperature and humidity are displayed. Fan should be on, but is not due to a broken wire. See the video link to watch fan operate properly.

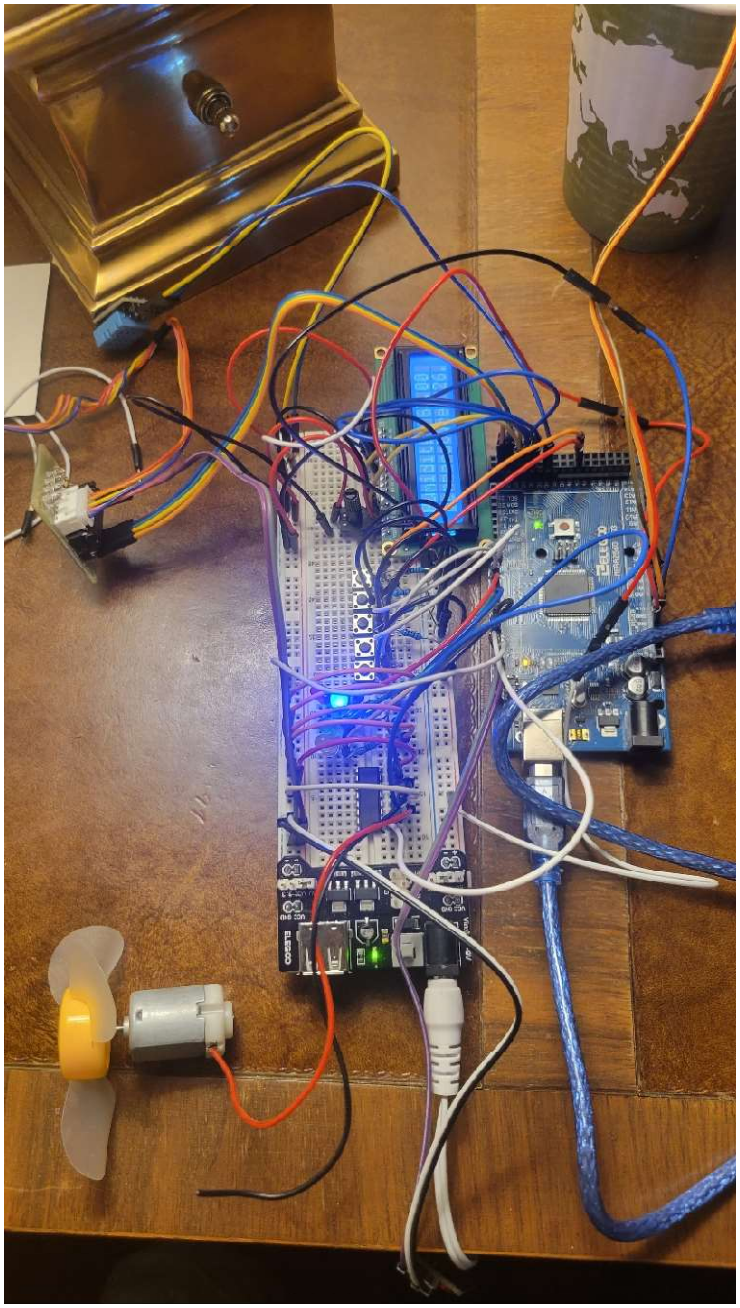


Photo 5: Picture of system in ERROR state. LCD is displaying the message “ERROR: Water Low”