CPE 301 Final Project Report

Ellison Domingo

Project Description:

This project was to create a water cooler using Arduino. The system begins in disabled mode, with the yellow LED on and the fan off. Pressing the start/stop button moves into idle mode, while the same button toggles back to disabled. Idle mode is represented by the green LED on, the fan still off, but temperature and humidity are shown on the LCD. If the temperature goes above a predetermined threshold, the system moves to running mode. The fan motor and the blue LED turn on to represent the running stage, and the temperature and humidity are still shown on the LCD. If the running stage is left for any reason, the fan turns off. The temperature going below the threshold returns to idle stage. In both the idle and running stage, if the water sensor detects the water level going beneath another predetermined threshold, then the system moves to error mode. Error mode displays an error message on the LCD and the red LED turns on. Pressing the reset button from error mode returns to idle.

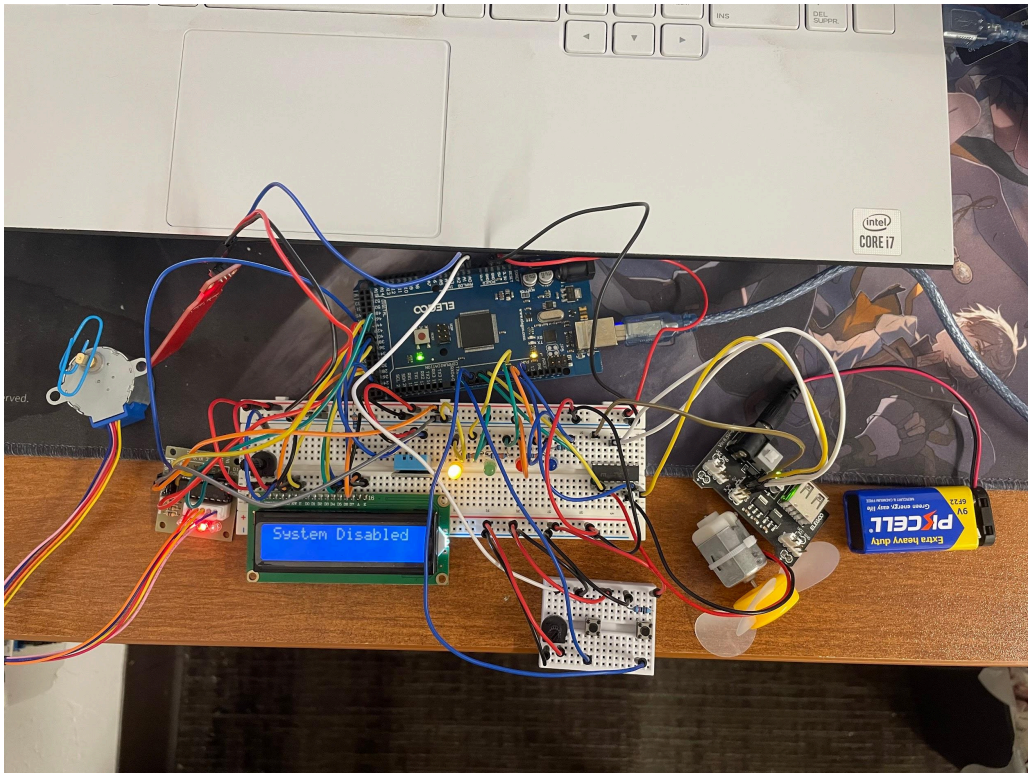
Component Details:

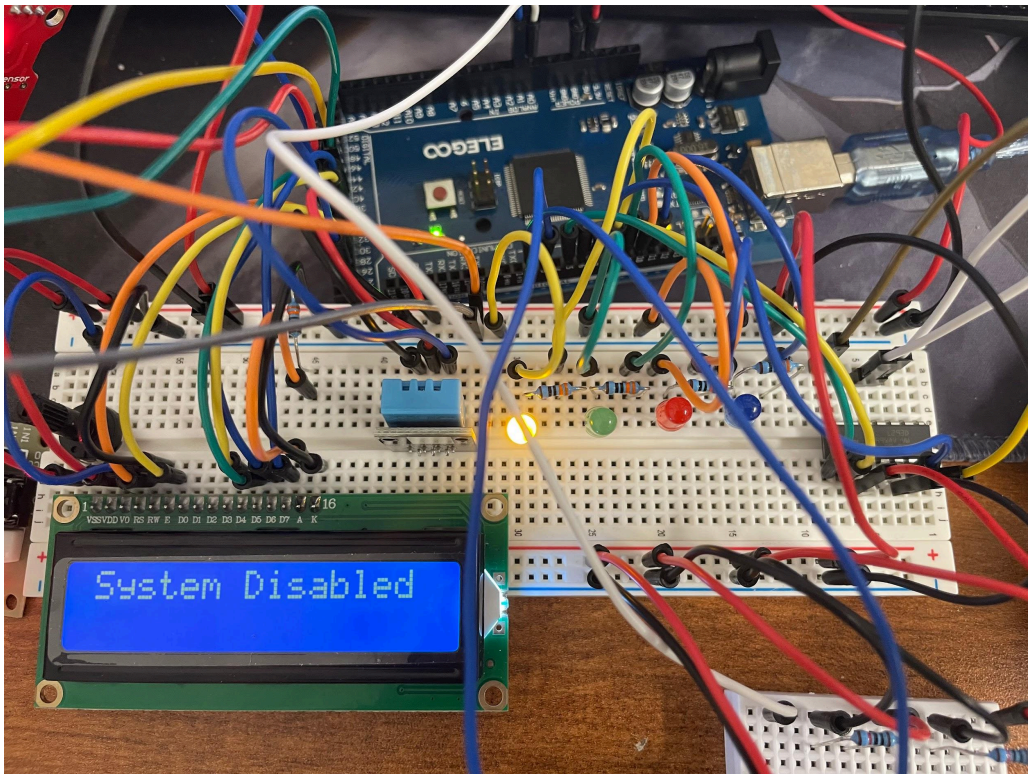
The DHT11 sensor continuously monitors the temperature and humidity of the environment to control the state of the cooler. The liquid crystal display (LCD) shows the user the temperature and humidity measured by the sensor while running or idle. When in error state, an error message is shown on the LCD instead. A potentiometer is connected to control the brightness of the LCD. There is a fan powered by a DC motor that turns on when the temperature is above a certain threshold to cool down the air. A stepper motor with a paperclip on it is also present to act as a vent. The user can turn the other potentiometer to control the direction of the stepper motor. A water sensor detects the level of water it is submerged in and sends the system into error mode if the level becomes too low. There is a start/stop button that cycles the system from disabled to idle, idle to disabled, error to disabled, or running to disabled. The restart button moves the system from error to idle. Lastly, a 9V battery is connected to the circuit as well to add extra power for the DC motor, as well as a L293D unit to control the motor properly.

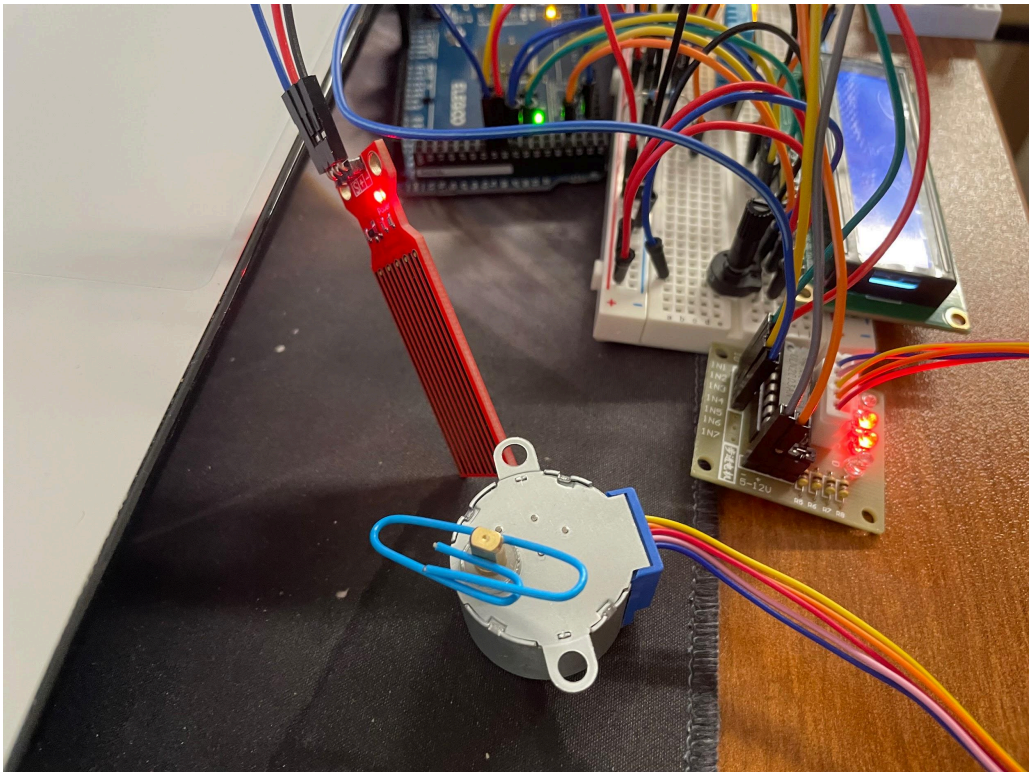
System Overview:

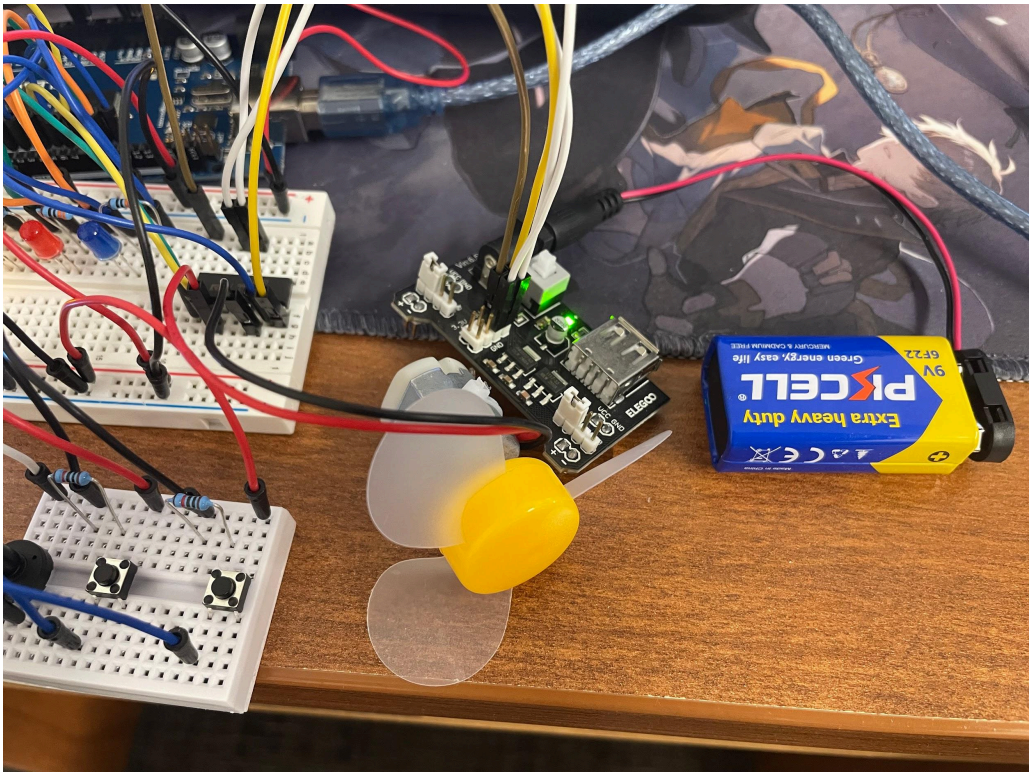
The system works as outlined in the project description with moving between stages according to button presses and temperature, humidity, and water levels. There are some additional constraints on the system as well. A 9V battery is connected to the Arduino for additional power for the DC motor and a L293D unit is also included so the motor can work properly. For the testing video, the thresholds for temperature and humidity are lowered so it shows functionality better for the video. The time delay for the readings on the LCD is also shorter than specified so the demonstration video can see the update more often.

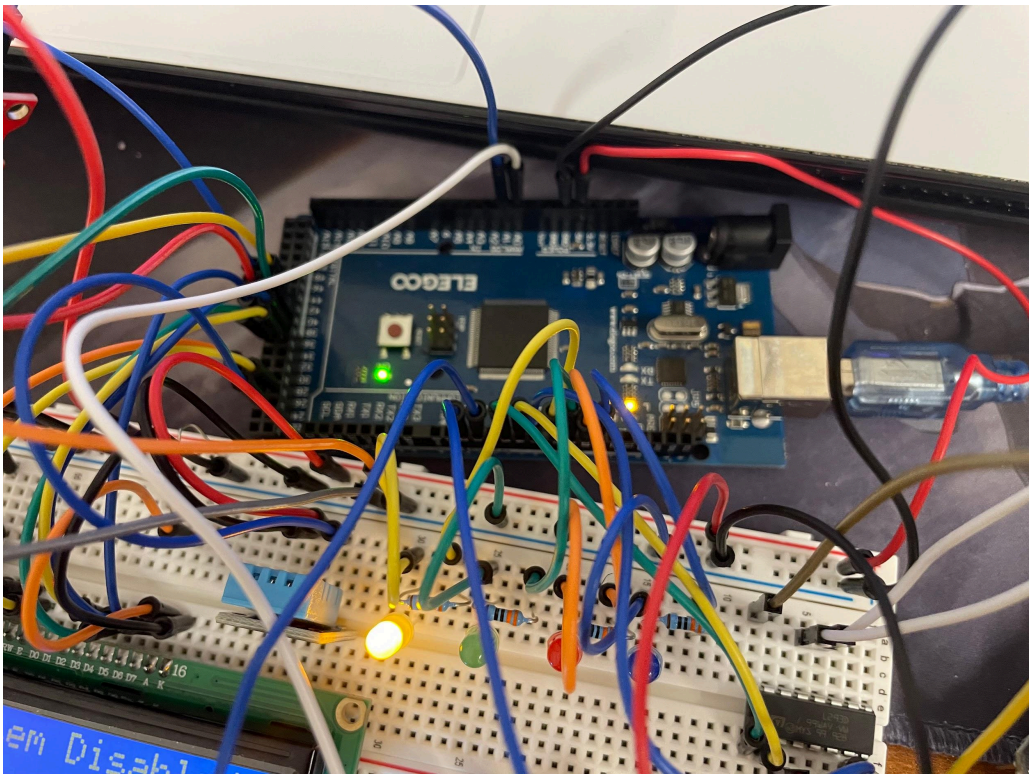
Circuit Images:



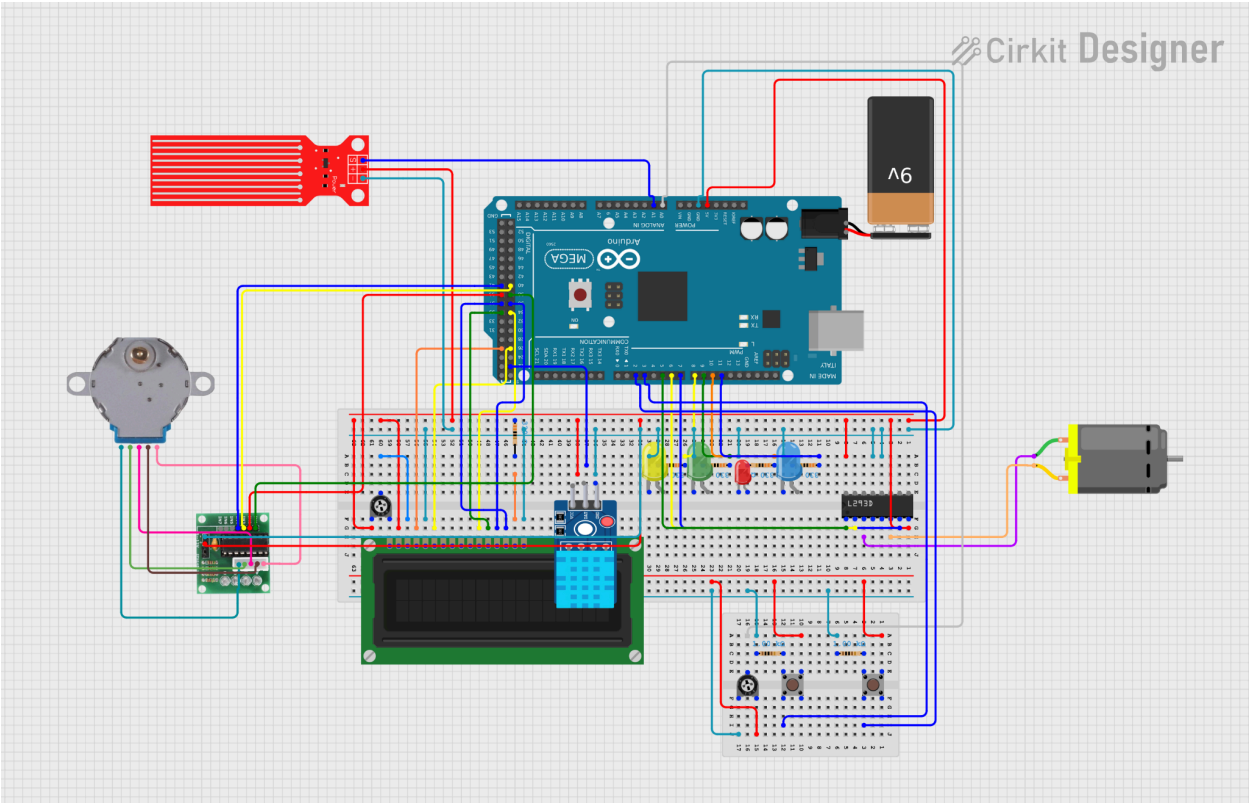





Extra heavy duty
PKCELL
9V
6F22
Green energy, easy life
ELEGOO


ELEGOO

Schematic Diagram:



Note: I couldn't find a module for the external power module so the schematic shows the DC motor connected directly to the Arduino. My actual circuit uses the external power module.

System Demonstration:

Link to demo video on YouTube: https://youtu.be/XonEPD-hhrg

Environmental Impacts:

I tried to make my system as energy efficient as possible by combining operations where possible and avoiding using sensor readings or calculations when unnecessary. For instance, when in disabled mode, no readings from the DHT11 sensor or the vent control potentiometer are taken at all because they don't function in that mode and there is no need to use power on that functionality. Every component is connected to power and grounded safely, and while building the circuit, I made sure to only connect and disconnect wires and components when fully disconnected to power to ensure safe operation and construction. I also used a 9V battery with the separate power module to avoid damaging the Arduino's power output with the DC

motor. I tried to construct the circuit with as few materials as possible by connecting pieces directly to the breadboard when possible to reduce the number of wires needed. I also color-coded wires for certain components and use red and black for power and ground where possible. This makes it easier to keep track of what wires connect to what components. Lastly, the user experience is relatively smooth, with the LCD showing any relevant data and with the buttons and potentiometer on the mini breadboard expansion as a “control panel” that was separate from the rest of the circuit.