Austin Landis and Jacob Bledsoe

CPE 301 Spring 2025

CPE 301 Final Project Lab Report

1. Project Description

    1.1. This project involved building a swamp cooler using the Arduino designed for cooling based on environmental conditions. The system pulls hot air through a water-soaked pad, cooling it through evaporation before circulating it via a fan. It uses a water level sensor to ensure the reservoir is sufficiently filled, a DHT11 sensor to monitor temperature and humidity, and an LCD to display real-time environmental data. A motor adjusts the direction of the airflow through a vent, controlled by a potentiometer or directional buttons. A real-time clock module logs key events like motor activations and vent adjustments. The system operates across four states, disabled, idle, running, and error, with transitions triggered by environmental conditions or user input. In disabled, only the start button is active, while idle monitors temperature, humidity, and water level. If the temperature exceeds a threshold, the system moves to running, activating the fan. It returns to idle once the temperature drops. If the water level falls too low at any time, the system enters error, stopping the fan and requiring user reset after refilling. Each state is indicated by a different LED color on the breadboard, and all state changes are time stamped using the real time clock.

2. Component Details

    2.1. Arduino Mega 2560

        2.1.1. The Arduino Mega 2560 is our microcontroller that manages the execution of all logic, manages inputs, and controls all of our sensors.

    2.2. Water Level Sensor

2.2.1. The water level sensor detects whether or not the water in the reservoir is below the minimum threshold. In this project, it triggers the transition to the error state when water is too low.

2.3. DHT11

2.3.1. The DHT11 humidity sensor measures and records the humidity. For our project, the DHT11 helps determine when the fan motor should be activated. The DHT11 sensor requires the DHT library.

2.4. LCD

2.4.1. This screen displays system status, temperature, humidity, and other info. It connects to six Arduino pins in 4-bit mode and is driven by the LiquidCrystal library.

2.5. Stepper Motor

2.5.1. The stepper motor adjusts the direction of the output air vent. The Arduino sends pulse sequences to rotate the motor a set number of steps either clockwise or counterclockwise.

2.6. Buttons

2.6.1. The buttons control the system state transitions. In the project, the start button takes the system from disabled to idle, the stop button takes the system from an active state to disabled, and the reset button takes the system from error to idle.

2.7. LED Indicators

2.7.1. The LED’s represent the state of the water cooler. yellow represents disabled, blue represents idle, green represents running, and red represents error. The LED’s are controlled by the output pins through resistors.

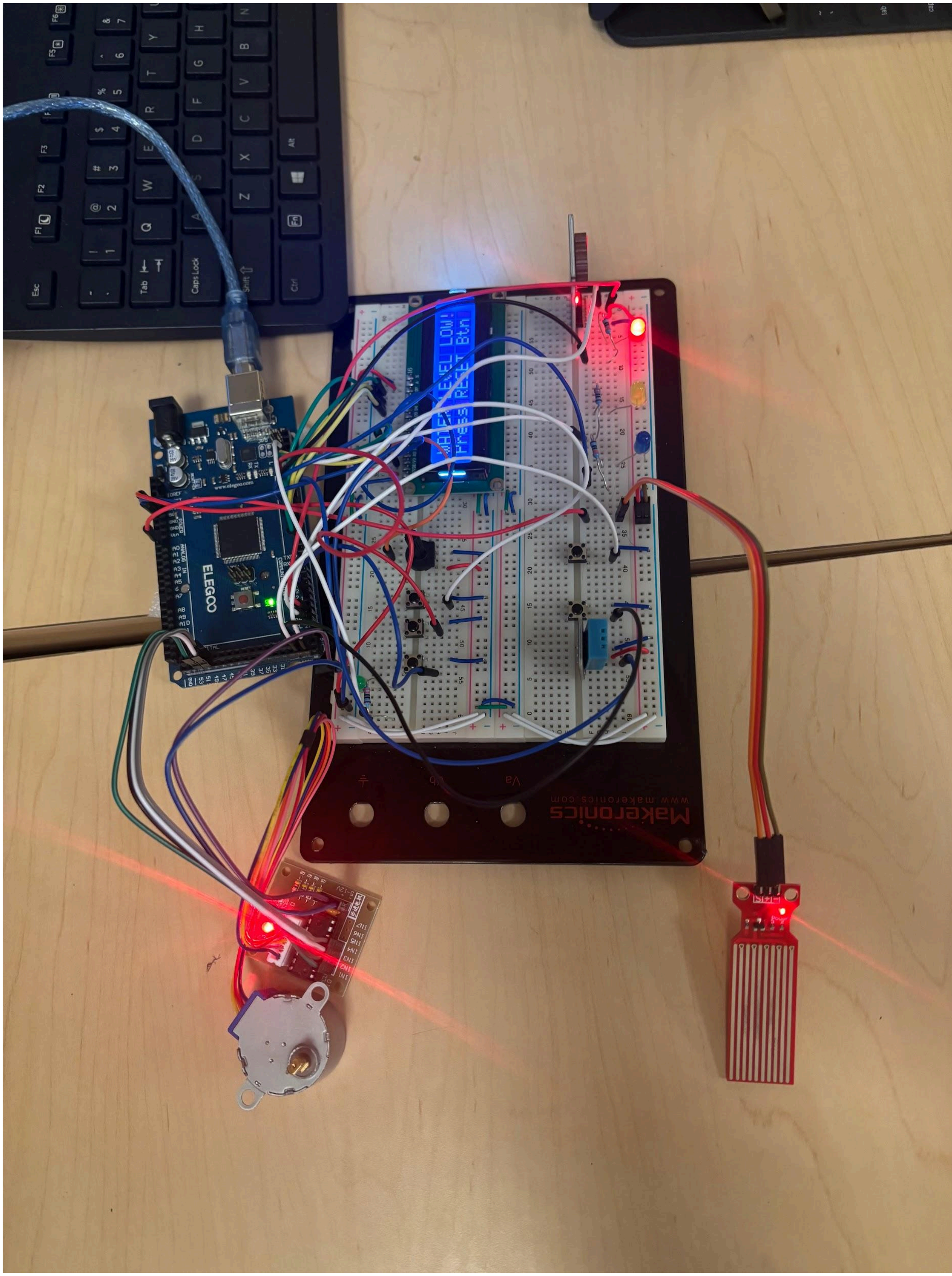
2.8. Real Time Clock (RTC)

2.8.1. The RTC accurately keeps track of time to log events. It timestamps state transition, fan activation, adjustment of the vent, and sends this data to the serial monitor.

3. System Overview

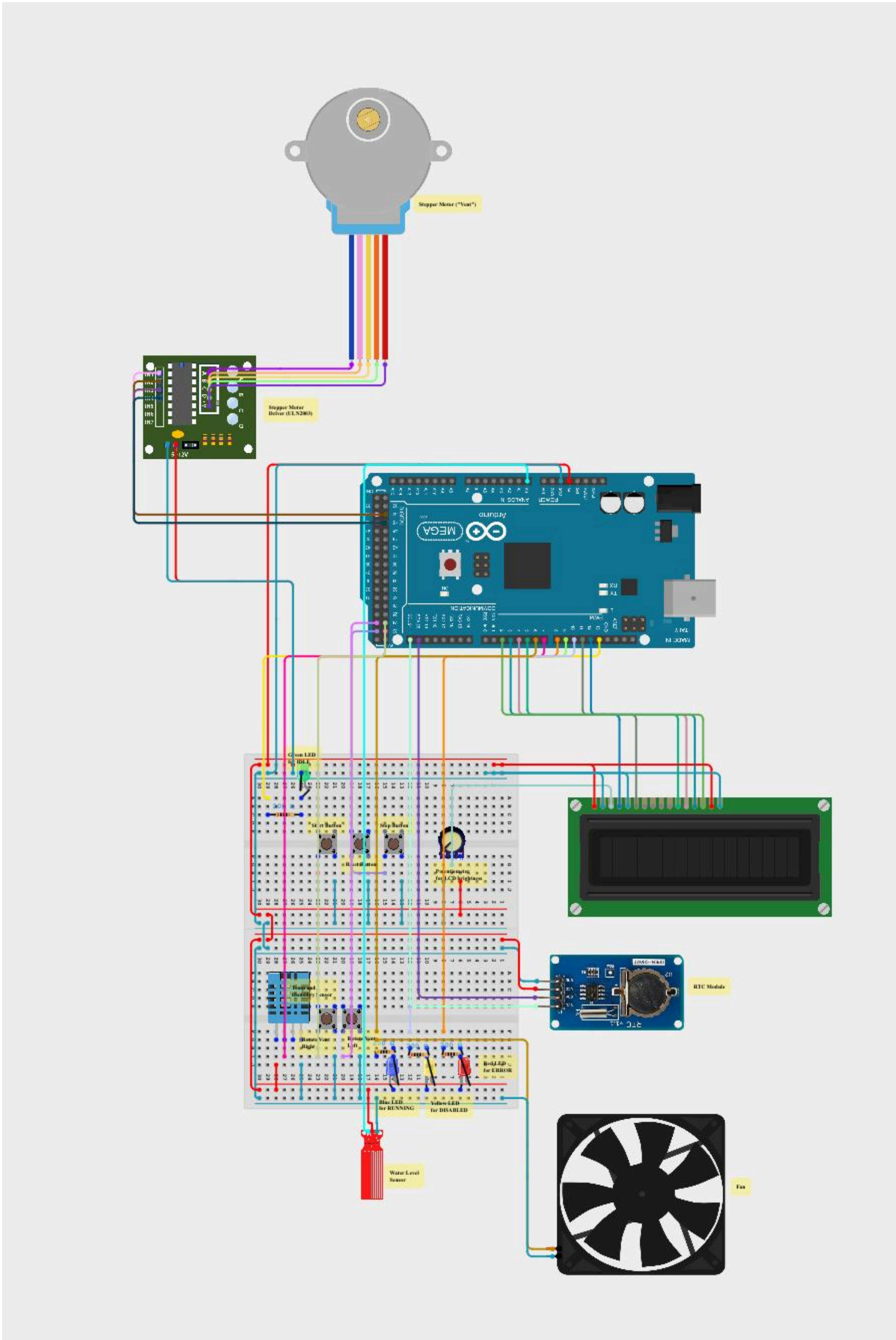
3.1. When trying to implement the fan blade, it caused interference with LEDs and LCD, making the LCD display random characters. We could not find an effective workaround.

4. Circuit Image


WATER LEVEL LOW!
Press RESET Btn
ELEGOO
Makeronics
www.makeronics.com

4.1.

5. Schematic Diagram



5.1. https://app.cirkitdesigner.com/project/4b0371bd-4e49-4cca-b2bc-39e1240fc979

6. System Demonstration

Included in Github Repo