Final Project: Arduino Swamp Cooling System

CPE 301.1102 Embedded Systems Design

Submitted to:

Dr. Bashira Akter Anima

Submitted by:

Christian Culanag & Manuel Morales-Marroquin

May 12th, 2024

1 System Overview

Our project is focused on developing a swamp cooler through the use of an Arduino Mega 2560 and its components. The design utilizes components such as the DHT11 sensor for monitoring temperature and humidity, a water level sensor to ensure efficient operation, and a stepper motor to adjust airflow direction. User interaction is captured through an LCD display and multiple control buttons, with safeguards like separate power supplies for motors and the status indicating LEDs.

2 Design Constraints

- **Operating Temperatures**:
    - **Optimal Range**: The system is designed to function in temperatures from 25°C to 50°C (77°F to 122°F), which is ideal for dry and or hot climates.
    - **Testing Challenges**: Testing at lower temperatures (around 25°C) can be challenging due to reduced evaporation rates. This can make it difficult to accurately gauge the system's efficiency, as it's designed to perform best under much hotter conditions.
- **Humidity Range**:
    - **Specifications:** The DHT11 sensor used in our system measures humidity levels across a range from 20% to 90% RH, with an accuracy of ±5% RH. This capability ensures that our swamp cooler operates optimally in environments with lower humidity, where the evaporation process is more effective.
        - https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/
- **Power Requirements**:
    - **Supply Needs**: The system uses a 5V supply for the Arduino and its sensors, and incorporates a separate 9V DC power module specifically for powering the fan motor and other high-load components.
- **Motor Configuration**:
    - **Design and Integration:** Unlike traditional swamp coolers, the motors in this design are configured to work with a specialized setup that may differ from standard applications. This configuration allows for more controlled and efficient operation but requires careful integration and testing to ensure optimal performance.

3 Experimental Design

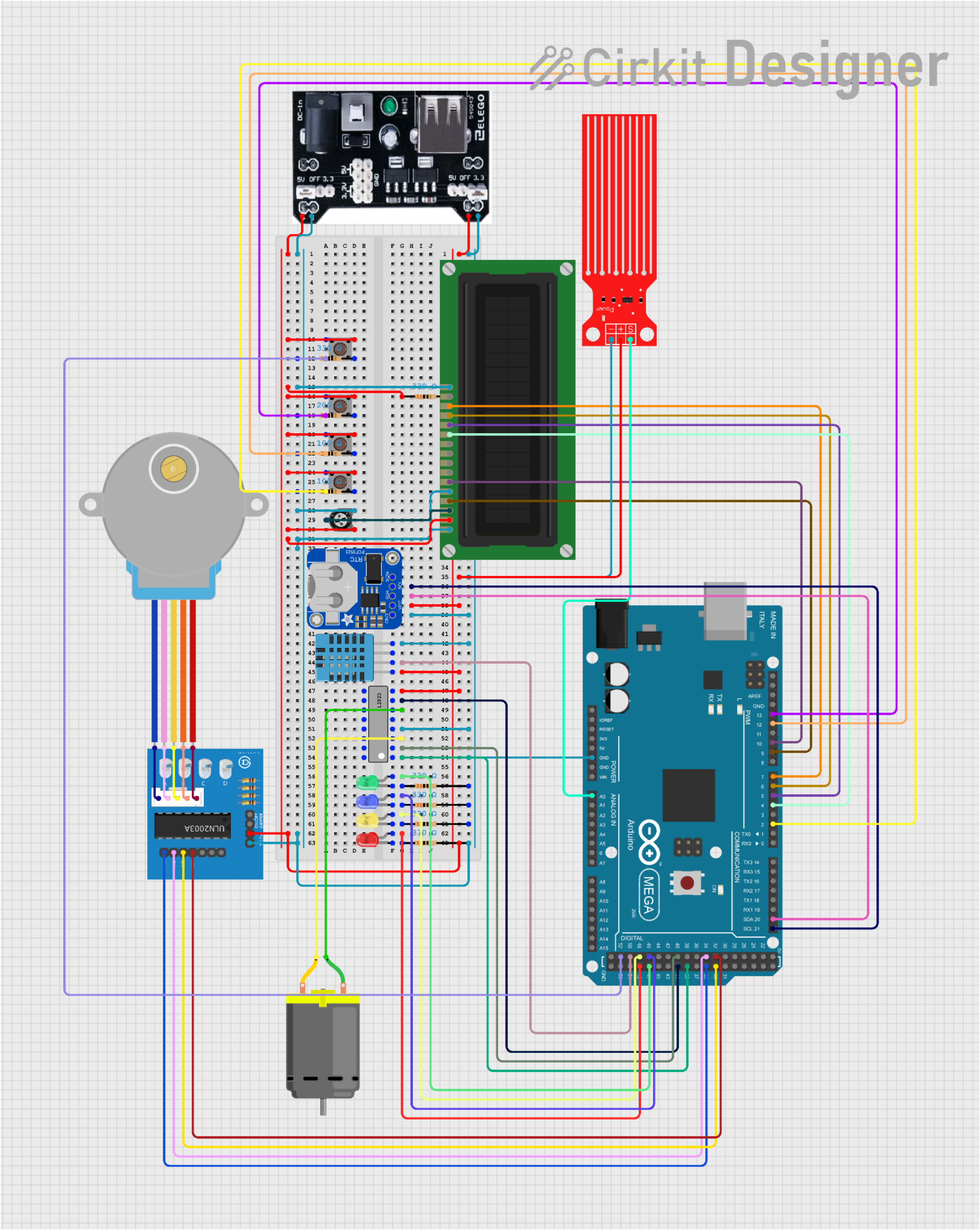


Figure 1: Experimental Design of Arduino Swamp Cooling System in Cirkit Designer.

4 Complete System Schematic

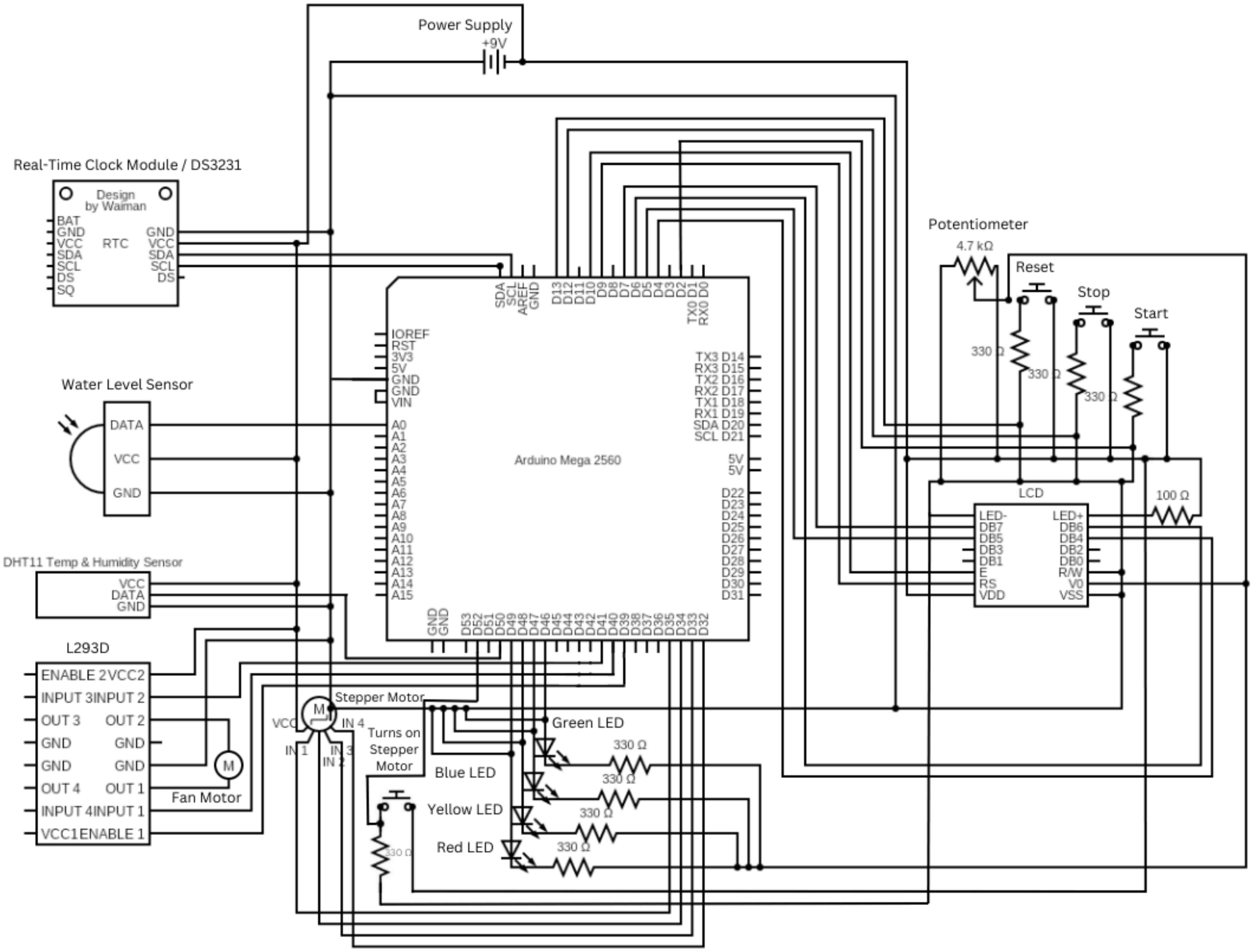


Figure 2: Schematic Design of Arduino Swamp Cooling System in CircuitLab.

5 Physical Implementation

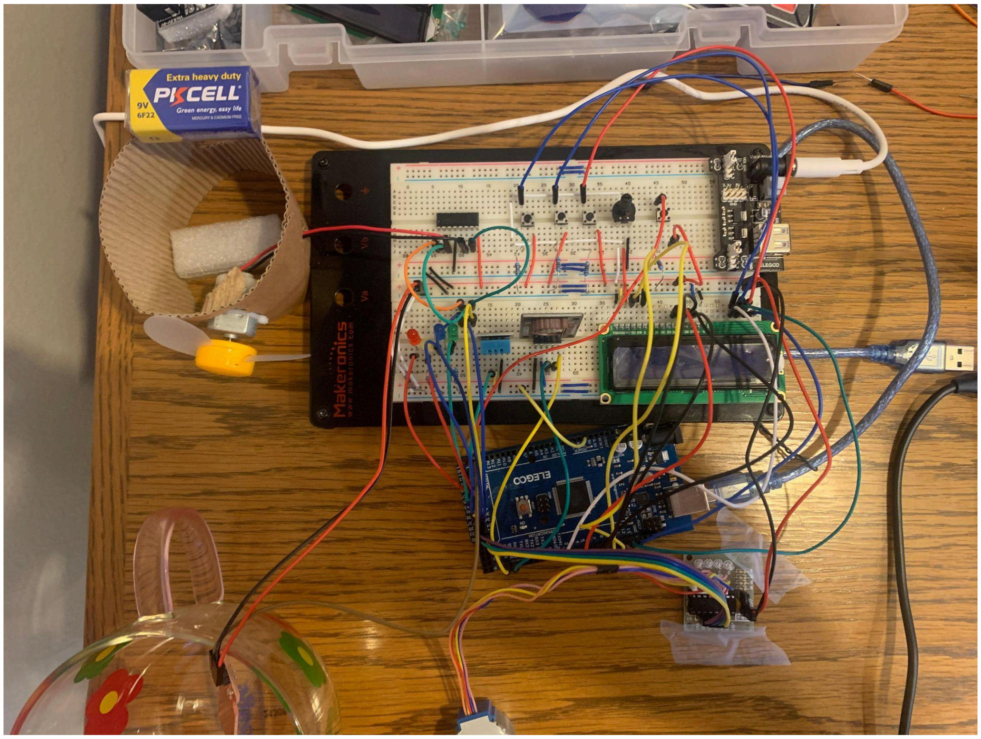


Figure 3: Physical Implementation of Arduino Swamp Cooling System, Top-Down View.

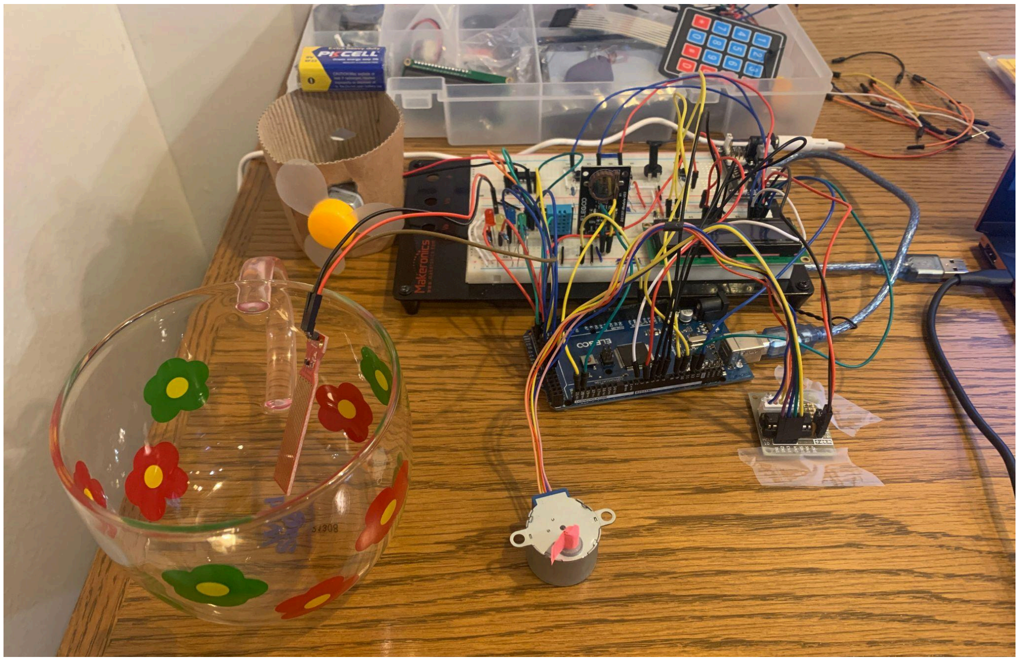
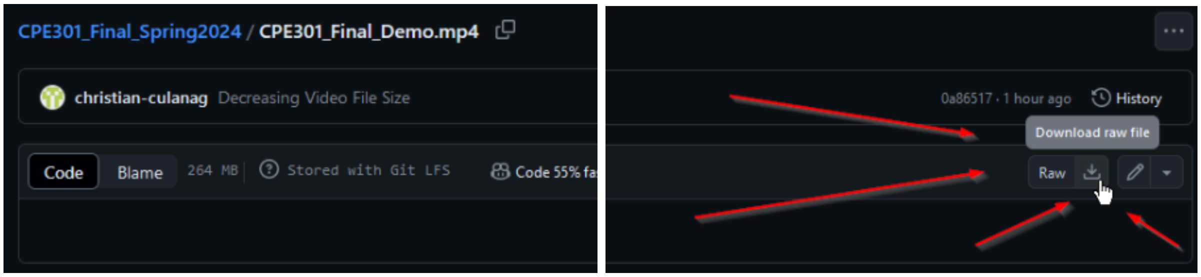


Figure 4: Physical Implementation of Arduino Swamp Cooling System, Side-View.

6 Video Of Operation

- https://github.com/1202-Morales-Marroquin-Manuel/CPE301_Final_Spring2024/blob/main/CPE301_Final_Demo.mp4



- **Instructions:** Access the GitHub Repository, click on "CPE301_Final_Demo.mp4" or use the link above, download the raw file, and view the MP4 file locally.

7 Detailed Component Descriptions and Functionalities

- **Arduino Mega 2560:**
  - **Description:** The main control unit for the swamp cooler. It manages inputs from sensors and directs outputs according to the program.
  - **Functionality:** Runs the system software, coordinating sensors, the display, and motor actions.
  - **Specification/Data Sheets:**
    - https://www.electronicshub.org/wp-content/uploads/2021/01/Arduino-Mega-Pinout.jpg
- **DHT11 Temperature and Humidity Sensor:**
  - **Description:** Measures ambient temperature and humidity.
  - **Functionality:** Provides real-time data to the Arduino to determine if cooling is needed based on predefined thresholds.
  - **Specification/Data Sheets:**
    - https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf
- **Water Level Sensor:**
  - **Description:** Monitors the water level in the cup reservoir.

    - **Functionality:** Alerts the system when the water in the cup drops below a certain level.
    - **Specification/Data Sheets:**
        - https://www.biomaker.org/block-catalogue/2021/12/17/water-level-sensor-tzt-water-level-sensor
- **Stepper Motor:**
    - **Description:** Adjusts the angle of the air outlet vent.
    - **Functionality:** Adjusts where the cool air goes to better spread the cooling.
    - **Specification/Data Sheets:**
        - https://www.mouser.com/datasheet/2/758/stepd-01-data-sheet-1143075.pdf
        - https://pages.pbclinear.com/rs/909-BFY-775/images/Data-Sheet-Stepper-Motor-Support.pdf
- **Fan Motor (DC Motor with Fan Blade):**
    - **Description:** Powers the airflow through the swamp cooler.
    - **Functionality:**By moving air consistently, it helps in cooling and humidifying the air, simulating a swamp cooler's function.
    - **Specification/Data Sheets:**
        - https://cdn-learn.adafruit.com/downloads/pdf/adafruit-arduino-lesson-13-dc-motors.pdf
        - https://leisonmotor.en.made-in-china.com/product/cBVJMabuaPRz/China-FF-030-3V-5V-6V-Mini-Electric-Toy-Brushed-DC-Motor.html
- **LCD Display (16x2 LCD or similar):**
    - **Description:** Displays system status, temperature, humidity, and alerts.
    - **Functionality:** Provides a user interface to monitor and interact with the system.
    - **Specification/Data Sheets:**
        - https://components101.com/displays/16x2-lcd-pinout-datasheet
- **Real-Time Clock Module (DS3231 or similar):**
    - **Description:** Keeps track of the current time and date.
    - **Functionality:** Used to timestamp system events and record data logs.
    - **Specification/Data Sheets:**
        - https://components101.com/modules/ds3231-rtc-module-pinout-circuit-datasheet
- **Separate Power Supply for Motors:**
    - **Description:** Provides power specifically for the motors.
    - **Functionality:** Ensures that the motors receive a stable and isolated power supply.
    - **Specification/Data Sheets:**

        - http://www.handsontec.com/dataspecs/mb102-ps.pdf
- **Push Buttons (for Start, Stop, Reset, and Adjusting Stepper Motor):**
    - **Description:** Manual controls that allow the user to interact with the swamp cooler system.
    - **Functionality:**
        - **Start Button:** Activates the cooling system.
        - **Stop Button:** Deactivates the cooling system or halts operations.
        - **Reset Button:** Resets the system to its default state.
        - **Adjust Button:** Changes the position of the stepper motor, thus adjusting the airflow direction.
            - **Initial Press:** Activates the stepper motor to rotate left by a quarter turn.
            - **Subsequent Press:** Reverses the stepper motor, returning it to the original position.
    - **Specification/Data Sheets:**
        - https://wiki-content.arduino.cc/documents/datasheets/Button.pdf
        - https://components101.com/switches/push-button
- **Potentiometer:**
    - **Description:** A variable resistor that tweaks electronic signal levels.
    - **Functionality:** In this project, it is used to modify the display settings on the LCD, such as contrast or backlight intensity.
    - **Specification/Data Sheets:**
        - https://arduinogetstarted.com/tutorials/arduino-potentiometer
- **LEDs (Red, Green, Blue, Yellow):**
    - **Description:** Small lights that show the system's status.
    - **Functionality**:
        - **Red LED:** Lights up to indicate warnings or errors within the system. Error state, displays an error message that the "*Water level is too low.*"
        - **Green LED:** Lights up when the system is functioning normally and all parameters are stable. Idle state, fan is off.
        - **Blue LED:** Activates when the fan motor is running, signaling active cooling. Running state, the fan motor starts when entering this state and stops when leaving it.
        - **Yellow LED:** Indicates the system is in standby mode or highlights important alerts, such as low water levels. Disabled state, fan is off.
    - **Specification/Data Sheets:**
        - https://wiki-content.arduino.cc/documents/datasheets/LEDRGB-L-154A4SURK.pdf

        - https://www.geeksforgeeks.org/led-blinking-using-arduino/
- **Resistor:**
    - **Description:** A basic component that controls the flow of electricity.
    - **Functionality:** Protects LEDs from getting too much power and helps manage inputs from sensors.
    - **Specification/Data Sheets:**
        - https://wiki-content.arduino.cc/documents/datasheets/Resistors.pdf
        - https://protosupplies.com/product/resistor-1k-5/
- **Jumper Wires, Breadboard/PCB:**
    - **Description:** Wires and a board used to connect and organize components.
    - **Functionality:** Allows for easy adjustments and testing before finalizing the electronic design.
    - **Specification/Data Sheets:**
        - https://components101.com/sites/default/files/component_datasheet/Breadboard%20Datasheet.pdf

8 GitHub Repository

- **Repository Link**: https://github.com/1202-Morales-Marroquin-Manuel/CPE301_Final_Spring2024.git