FINAL PROJECT

CPE 301.1001/301.1102

ROBB NORTHRUP

Project Overview

Operating Temperatures:

- The swamp cooler enters the RUNNING state when the environment temperature exceeds 25 degrees celsius.
- Temperature (as well as humidity) monitoring is achieved using a DHT11 sensor.

Power Requirements:

- The Arduino Mega 2560 microcontroller powers the system.
- A separate power supply provides 5V to both the fan and the stepper motor (which is used for vent control).

Hardware Components:

- An LCD is utilized for displaying system information. Under normal operating conditions, this displays the temperature and humidity of the environment. When the system enters an errored state (too little water in the reservoir), the LCD notifies the user that the water level is too low. The LCD 1602 Module with pin header is utilized in this project.
- The Stepper library controls the stepper motor(along with the ULN2003 Stepper Motor Driver Module are utilized) for vent control.
- DHT library facilitates communication with the DHT11 temperature and humidity sensor.
- RTClib library is utilized for real-time clock functionality, which allows updates to the state of the system to be monitored by the serial monitor along with timestamps. The DS1307 RTC module also allows the temperature and humidity to be updated every minute (instead of being overbearing and doing so every cycle).
- The L293D IC along with a servo motor is used for operation of the fan.

Thresholds:

- Water level threshold is set at 25%, checked against the water level sensor's readings.
- Temperature threshold for activating the cooler is 25 degrees Celsius.

Stepper Motor Configuration:

- The stepper motor is configured with 2048 steps per revolution.

- It rotates a certain angle per cycle, with the angle defined by the constant ANGLE_PER_CYCLE (this is set to 15 degrees, typically).
- The number of steps per cycle is calculated based on the steps per revolution and the angle per cycle.

States and Functionality:

The system operates in different states: DISABLED, IDLE, ERROR_STATE, and RUNNING. These states are outlined below:

1. DISABLED:
    a. Yellow LED enabled
    b. No monitoring of temperature, humidity, or water level
    c. Start button engages IDLE mode
2. IDLE:
    a. Green LED enabled
    b. Monitor temperature, humidity, and water level
    c. Vent position can be controlled
    d. Stop button put system into DISABLED state
    e. High temperatures force the system into the RUNNING state
    f. Low water forces the system into the ERROR state
3. ERROR:
    a. Red LED enabled
    b. Vent motor and fan are off
    c. Error message displayed on LCD
    d. If water level in reservoir sufficient, pressing the reset button will force the system into the IDLE state
4. RUNNING:
    a. Blue LED enabled
    b. Monitor temperature, humidity, and water level
    c. Vent position can be controlled
    d. Stop button put system into DISABLED state
    e. Low temperatures force the system into the IDLE state
    f. Low water forces the system into the ERROR state

Functions:

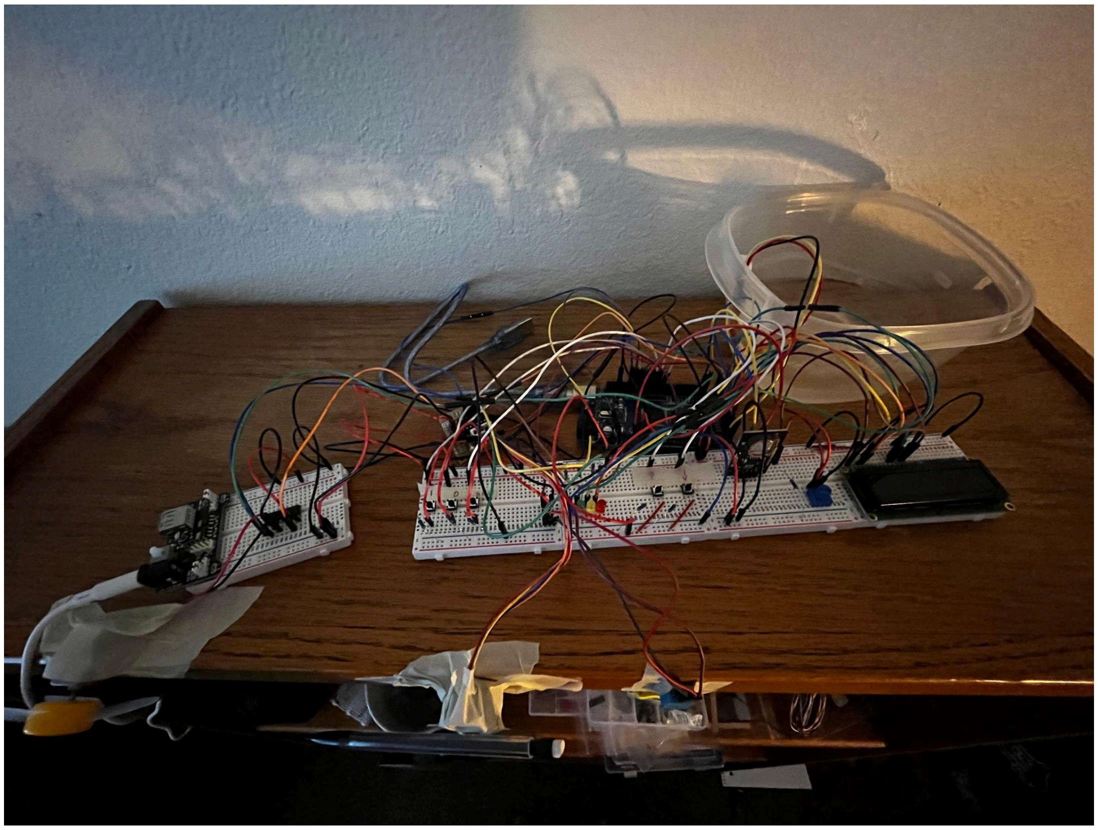
- Functions are defined for tasks like moving the stepper motor, checking water level, updating the LCD, measuring temperature and humidity, and reporting state transitions. These functions handle the core functionalities of the swamp cooler system, making it modular and easier to maintain.

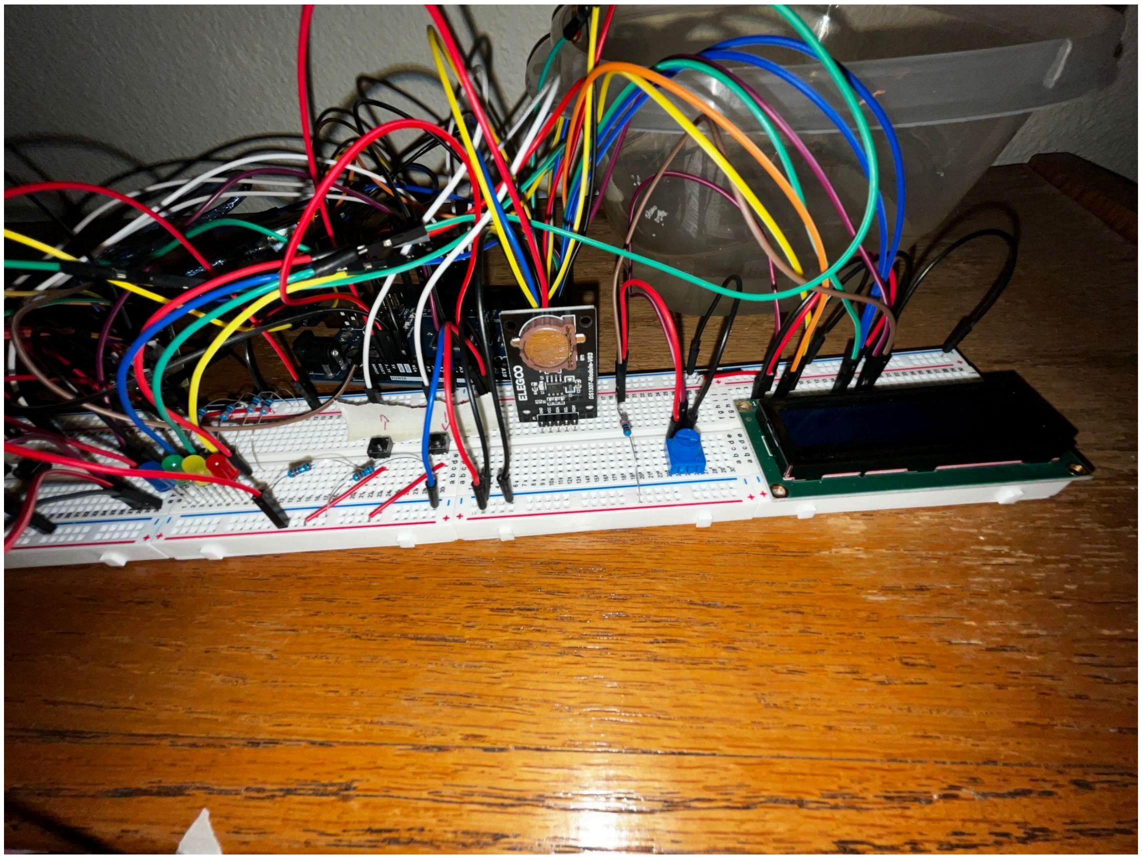
Constraints:

- The system seems to rely heavily on button presses for user interaction, which might not be the most intuitive or user-friendly interface.
- The swamp cooler's functionality might be limited by the accuracy and reliability of the DHT11 sensor for temperature and humidity measurements.
- The system lacks extensive error handling and recovery mechanisms, which might lead to unexpected behavior in certain conditions.
- The system was designed without an enclosure. This "swamp cooler" fails to act as an air conditioner in any real capacity, and therefore this project serves moreso as a prototype and insight into embedded system design.

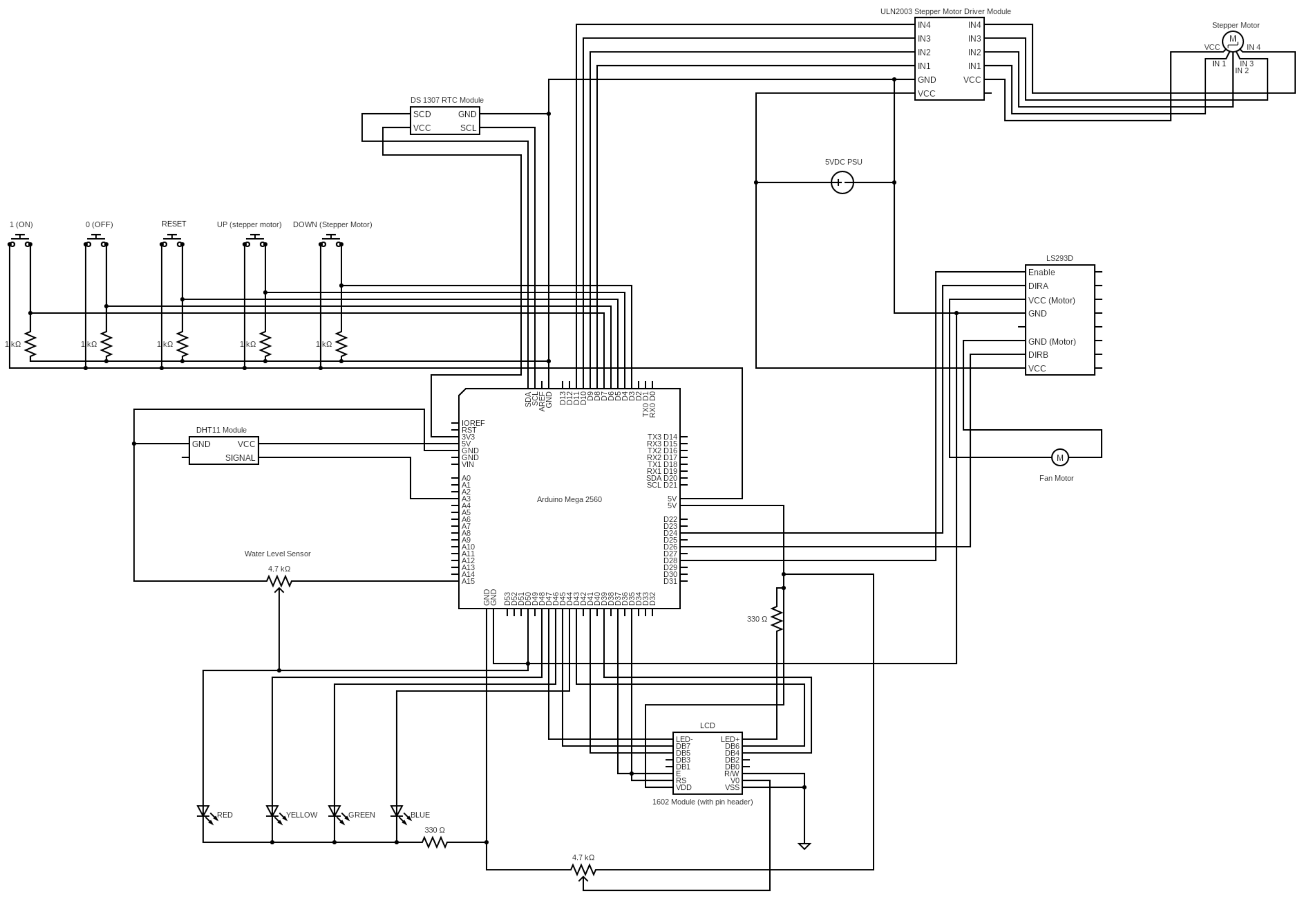
Overall, the design provides a framework for building a basic swamp cooler system using Arduino, with room for further refinement and optimization.

Pictures




ELEGOO

Schematic



(the full schematic file can also be found in the Repo, provided below, for closer inspection)

Repo

https://github.com/NorthrupRobert/FINAL-CPE-301.1001---ROBB-NORTHRUP.git