CPE 301 Final Project

Victor Lopez-Estrada

May 5, 2025

1 Introduction

This project is the culmination of the objectives and disciplines of CPE 301. Our task is to create a swamp cooler system using key topics learned from lecture and applications from the lab section, and apply low-level hardware interfacing and high-level component control using libraries. The main processor is the Arduino ATMega 2560 and the components used are the following:

- Temperature and humidity sensor (DHT11)
- Real-time clock (DS1307)
- Stepper motor (28BYJ-48) with driver (ULN2003)
- Water sensor module
- 3-6V DC motor with fan blade
- 16x2 LCD display (1602A)
- Breadboard power supply module

The design of the project encapsulates the goals and criteria of the class. The figure below shows the block diagram to illustrate the pin connections between the Arduino, the components, and the additional wiring required for logic flow.

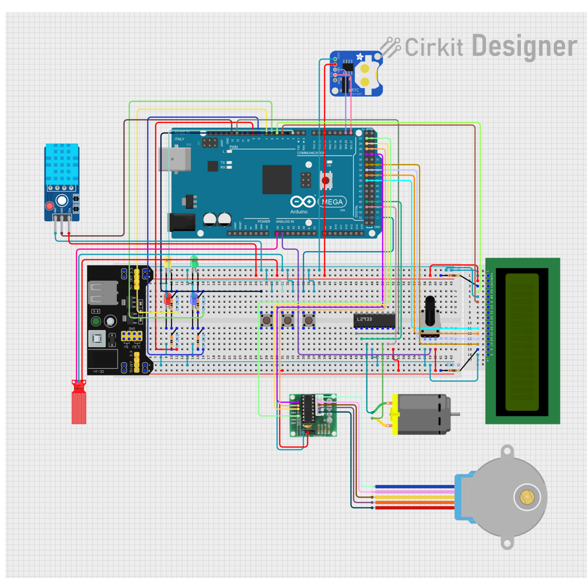

Figure 1: The complete circuit diagram.

2 Project Description

As previously stated, the scope of this project is to build a functional swamp cooler. The purpose of a swamp cooler is to cool areas in drier and hotter climates. A typical swamp cooler will pull air from the outside through a permeable pad soaked in water. This type of cooling system provides a more efficient method of cooling compared to air conditioners.

The cooler for this project will have properties similar and different to a standard swamp cooler:

1. Temperature / Humidity monitor: Temperature and humidity will be monitored via the DHT11 sensor, and both metrics will be displayed on an LCD 16x2 display.
2. Water Level Monitor: The water reservoir level will be monitored via a water sensor module and will alert when the water levels are too low.
3. Fan Control: Our fan and DC motor are built on a stepper motor that can tilt its angle and can be controlled by the user via a potentiometer.
4. Real-time status: Any changes in system state or fan angle will produce a message with the timestamp and status of the system to the Serial monitor in real time.

The system will have several logical states that will change dynamically according to the current conditions of the temperature and water levels. The initial state of the system is called DISABLED. In the DISABLED state, there will not be any monitoring and nothing should display to the LCD display. The user may still choose to tilt the angle of the fan. If the system is in any state other than DISABLED, the user can press the "off" button to force the system back to the DISABLED state.

The next state is IDLE. This state will be the base state that the system will be in. To become IDLE, a button that acts as the "ON" button must be pressed. All monitors will begin to actively check for all thresholds and will act when any threshold has been crossed. A reading of the temperature and humidity will be displayed on the LCD while in this state. If the temperature becomes too high, then the system will transition into the RUNNING state. If the water level is too low, the system will go into the ERROR state. Otherwise, the system will remain in IDLE.

While in the RUNNING state, the system will begin to run the fan to reduce the temperature. The system will actively check the temperature and should return to IDLE once the temperature has lowered below the threshold. The same metrics will be displayed to the LCD while in this state, similar to the IDLE state. Another similarity is the active monitoring of the water levels. If they become to low, the system will transition into the ERROR state.

The last state to cover is the ERROR state. This state occurs when the water levels become too low. An error message will appear on the LCD that indicates this status. To reset the error, the user must first replenish the reservoir. The user can reset the system by pressing the reset button that can reset the state if the water levels are high enough.

3 Component Details

This section is an overview of each component and their use. All components used were part of the Arduino Starter kit but can be found in different styles and variations.

3.1 DHT11 Sensor

A low-cost digital sensor that measures both temperature and humidity with reasonable accuracy. The sensor will be used to track if the temperature crosses a predefined threshold.

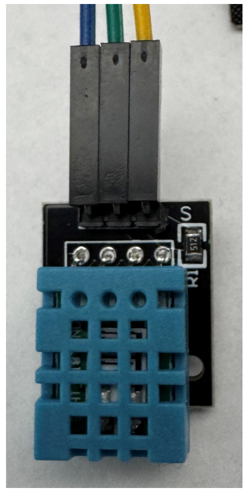

Figure 2: This component is the DHT11 sensor.

3.2 DS1307 RTC Module

A timekeeping chip that tracks seconds, minutes, hours, day, date, month, and year. It's use in this system is to timestamp all activity in real time.

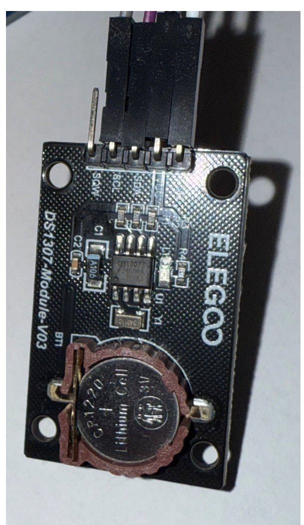

Figure 3: This component is the DS1307 RTC module.

3.3 28BYJ-48 Motor with the ULN2003 Driver

A small, inexpensive stepper motor with a 5V rating, often used for precise position control. This motor is used to tilt the angle of the fan motor and is controlled by the user via a potentiometer.

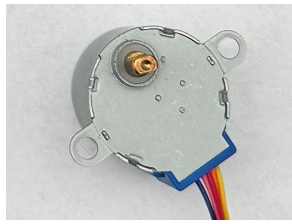

Figure 4: This component is the Stepper motor.

The stepper motor must use a driver module for control of the motor. The ULN2003 driver board allows easy control of the motor via microcontrollers, converting control signals into high-current pulses.

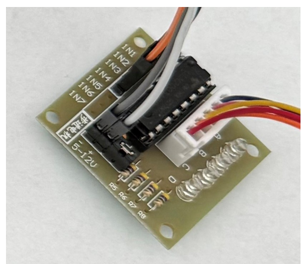

Figure 5: This is the driver module used to control Stepper motor logic.

3.4 Water Sensor Module

A sensor used to detect the presence or level of water. It typically outputs an analog signal (representing water level) and a digital signal (water detected or not), making it useful for leak detection or rain sensing. This sensor is used to check if the water level in the reservoir becomes too low.

Figure 6: This component is the water level sensor module.

3.5 DC Motor with Fan Blade

A simple DC motor that runs on 3–6 volts. When paired with a fan blade, it can be used for ventilation or to simulate wind in small projects. This motor is used in conjunction with a fan blade to cool the environment.

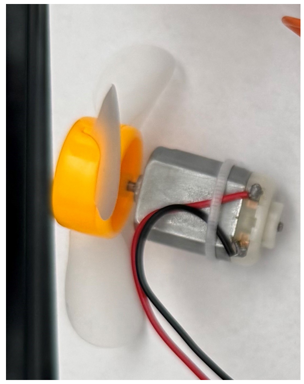

Figure 7: This component is the DC motor with the fan blade attached.

3.6 1602A LCD 16x2 Display

An alphanumeric LCD that can display 16 characters across 2 rows. This LCD will be used to display current temperature, humidity, or error conditions.

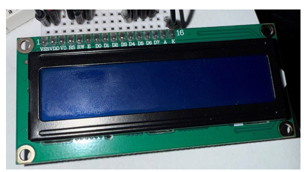

Figure 8: This component is the 1602A LCD 16x2 display.

3.7 Breadboard Power Supply

A module that mounts directly on to a breadboard, providing regulated 3.3V and/or 5V output from a 6–12V input. It simplifies powering small electronics projects on a breadboard. The module will be used to provide enough power to these components.

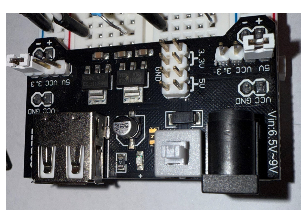

Figure 9: This component is the power supply module.

4 Conclusion

This project successfully demonstrates the design and implementation of a swamp cooler system using the Arduino ATMega 2560 and by interfacing with various components such as a DHT11 sensor, a water level sensor, the DS1307 RTC module, an LCD display, a stepper motor, and a DC motor with a fan blade. The system effectively monitors environmental conditions and dynamically adjusts its state (DISABLED, IDLE, RUNNING, ERROR) in real time. The circuit design also implements standard practices and complex logic systems as shown in the figure below.

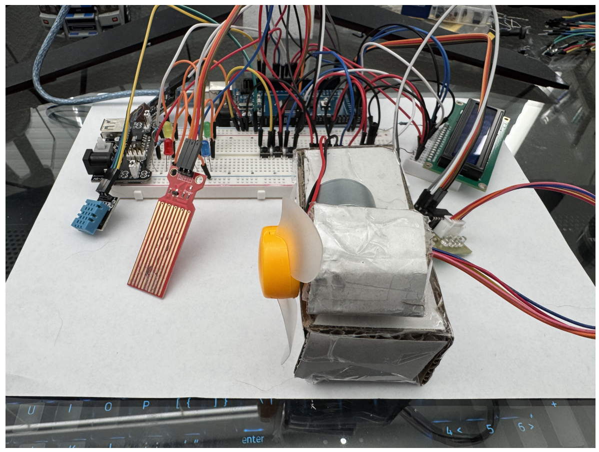

Figure 10: Front point of view of overall project.

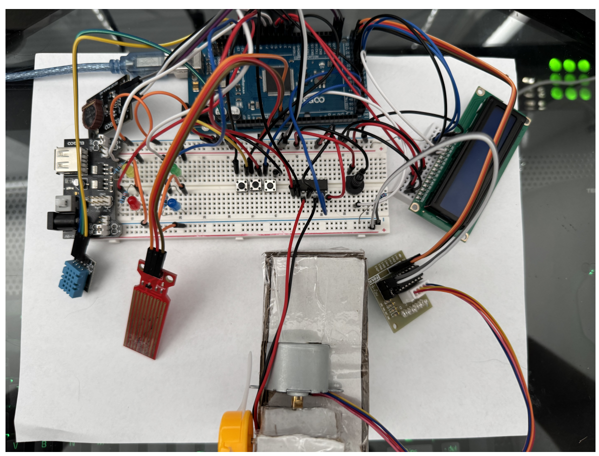

Figure 11: Top point of view of overall project.

Through responsive interrupt handling, real-time feedback, and error detection (for example, low water level), this embedded system reflects the criteria and scope of this project. This project not only reinforces theoretical knowledge from lecture and labs, but also provides hands-on experience with integrating hardware and software in a cohesive, state-driven system. In conclusion, this project gives insight into the discipline of embedded system engineering.