CPE 301 Final Project

James Torres-Davila, Jacob Arce, George Kokkinis, Javier Correa-Martinez

December 16, 2023

Abstract

The goal of the project is to create an evaporation cooling system, also known as a swamp cooler. Evaporation coolers provide a more energy efficient alternative to air conditioners in dry, hot climates. They work by pulling air in from the outside through a pad that is soaked in water. The water evaporation cools and humidifies the air. This project will utilize multiple components such as a liquid crystal display, fan motor, colored LEDs, push buttons, temperature and humidity sensors, and a water level sensor.

1 Experimental Design

In this project we will use an Arduino ATmega2560 to create a swamp cooler. This project will use a liquid crystal display to show the humidity and air temperature, a fan motor powered by an extra supply and a L293D IC, colored LED's that will show the different states of the cooler, push buttons, DHTH temperature and humidity sensors, and a water level sensor. The cooler functions by monitoring the water level in a cup and displays a warning if it gets too low. Also, the system records the time and date when the motor starts or stops.

The liquid crystal display shows the current temperature and humidity in the area. After testing our swamp cooler and using the DS1307 module, we verified a temperature of 20°C. If the water level drops below the threshold, the LCD displays an error message, "ERROR: water level low". We used the LiquidCrystal.h library to determine the pins of the display. Also we employed RTClib.h to establish the DateTime() function for the expected results.

The fan motor works with the L293D IC chip and operates onyl when the system is configured with proper water levels and temperature values. The setFanMotor() function determines the motor's behavior and allows it to switch between the disabled, idle, and running states based on the situation. The motor is accompanied by the power supply which porives an additional direct power soure. Also, using the L293D pinout chart and the control pins, we can control the direction of the motor's rotation.

We used four different colored LED's in our project which corresponded to the different states of the project's components. The yellow LED signifies that system is in a disabled state, the green LED signifies that the system is in an idle state, and the blue LED indicates an active state in which all the requirements are met. The red LED on the other hand, illuminates when there is an error which signifies that the idle state has been disrupted. Each LED color has a specific pin assigned to it to differentiate between the system states.

We used three push buttons that each served different functions. TO toggle between the disabled and enabled states, we used stop and start buttons respectively. We also included a button for when the sysen is inactive to bring it back to its default idle state which was the reset button. The stepper motor and vent direction are controlled by the state of the system.

To the detect the temperature within the temperature and humidity sensor we used a thermistor. The thermistor combines with a capacitive humidity sensor to measured the humidity and transmit a signal to the data pin once the readings are recorded. Utilizing the DHT.h library's read11() function, the system record the temperature and humidity readings from the DHT11 sensor. The sensor is defined with DHT11_PIN 7. The temperature reading alternates between the running and idle states of the system.

The water level sensor monitor the amount of water present from an outside source. Once it detects a liquid then it can switch between the idle, running and error states. The system uses the adc_read() function to measure the water level. If the value measured below the WATER_THRESHOLD value which is the water threshold value, and error state is returned.

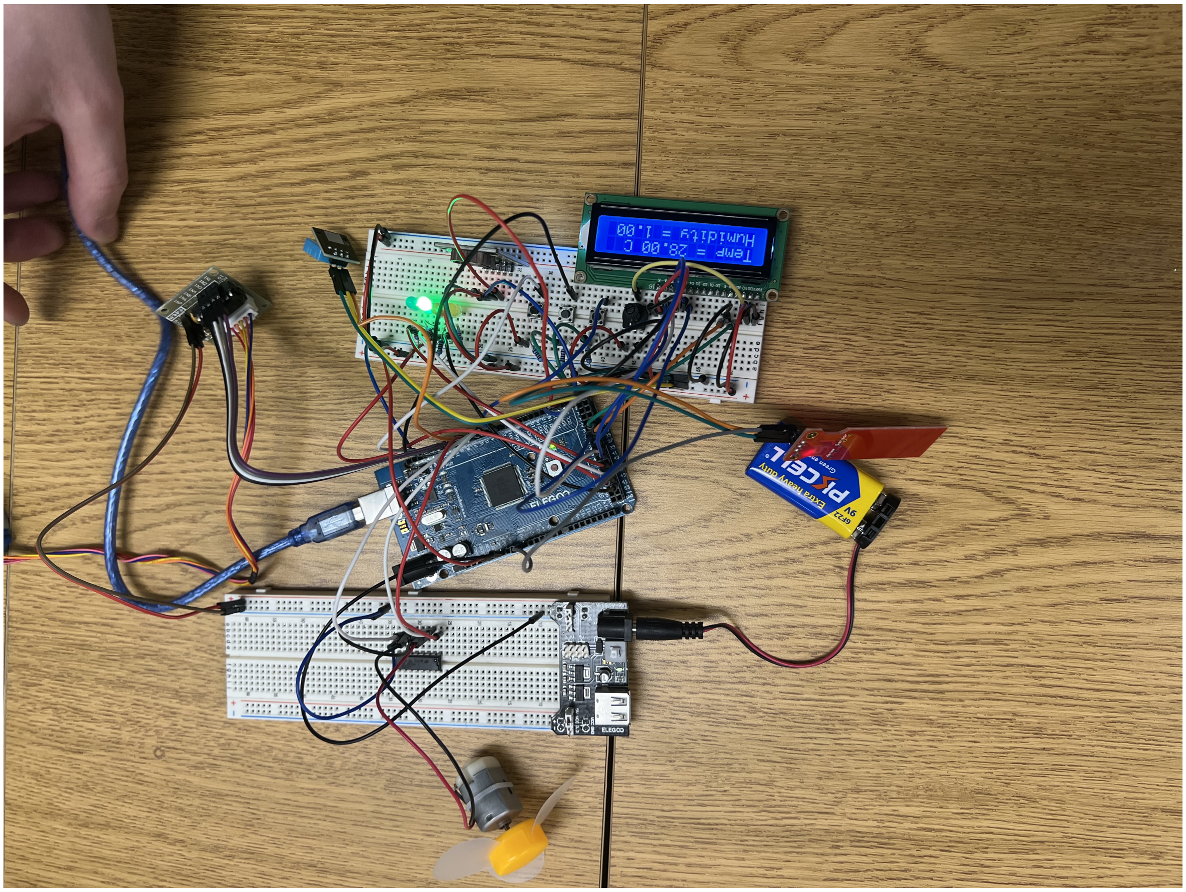

Figure 1: Figure 1: Circuit

2 Results

In this project, we successfully built an evaporative cooler that operated as intended. Each LED correctly signaled the corresponding conditions. The yellow LED lit up when the system was disabled, the green LED turned on when the system was in an idle state, the blue LED illuminated when the system was running. Finally, if there was an error, the red LED lit up. The LCD displayed the status of the build and the current information on air temperature and humidity.