CPE 301 Final Project Report

Group 30

Nicole Lacanlale and Thalia Ysassi

Fall 2025 – December 12, 2025

I) Introduction

The goal of this project is to create an evaporation cooling system by utilizing Arduino components and concepts from CPE 301's lecture and lab. This cooling system is also known as a swamp cooler, which is an efficient cooling system using evaporative cooling from a water source. When the temperature rises, a fan automatically turns on and blows on cool water, causing it to evaporate and lower the surrounding temperature. The circuit mimics a swamp cooler's functions, which are documented below.

II) System Components

This project utilizes various components to function as a water cooler. This section will provide additional detail on each component before building the circuit.

Arduino Mega 2560

A microcontroller board used to operate the embedded system. It is used alongside the Arduino IDE software to operate the system. The finished code for this project will be a .ino file. For more information on the system's pin connections, view the Arduino Pinout: https://www.electronicshub.org/wp-content/uploads/2021/01/Arduino-Mega-Pinout.jpg

DHT11 Temperature and Humidity Sensor

A digital sensor used to measure the temperature and humidity of its surroundings. This simulates the swap cooler's ability to measure the nearby temperature and humidity. Its data can be accessed using the DHT.h library in the Arduino IDE. This component is connected to a breadboard and the Arduino Mega 2560.

Water Sensor

A sensor that detects when water is touching its copper wires. This simulates a swamp cooler's ability to check if its water reserves are running low before operating the fan. Its data can be accessed using the SPI.h library in the Arduino IDE. This component is connected to a breadboard and the Arduino Mega 2560.

Light Crystal Display (LCD)

A screen that displays information the Arduino Mega 2560 receives. It is used in this circuit to display the current temperature and humidity levels, as well as printing possible error messages. Its data can be accessed using the LiquidCrystal.h library in the Arduino IDE. This component is directly connected to the breadboard and the Arduino Mega 2560.

28BYJ-48 Stepper Motor

A 5-wire, 4 phase stepper motor. This simulates a swamp cooler venting system. This component is connected to a stepper motor driver board and breadboard.

ULN2003 Driver Board

A driver board used to manage the stepper motor component. Its data can be accessed using the Stepper.h library in the Arduino IDE. This component connects to the stepper motor and Arduino Mega 2560.

Real Time Clock (RTC)

A module used to keep track time. It is used to keep track of timestamps in between stages of the circuit. Its data can be accessed using the RTClib.h library in the Arduino IDE. This component connects to the breadboard and Arduino Mega 2560.

Fan motor

This component is used to simulate a swamp cooler's fan. This component connects to a fan blade. Additionally, this component must be connected to a separate power supply module, as it can damage the Arduino Mega 2560 when connected directly.

Power Supply Module

A module that provides power to the fan motor component. This is necessary due to the power demands of the fan motor. This component connects directly onto the breadboard and 9v battery.

L293D Motor Driver IC

A motor driver that manages the fan motor. This component connects directly onto the breadboard.

Additionally, this circuit also uses LEDs, resistors, and buttons. All of these components are attached directly onto the breadboard.

III) Schematic Diagram

When building this circuit, a schematic diagram was also created to clearly show the circuit's wiring. It is also useful for future reference if our group decides to rebuild the circuit at a later date.

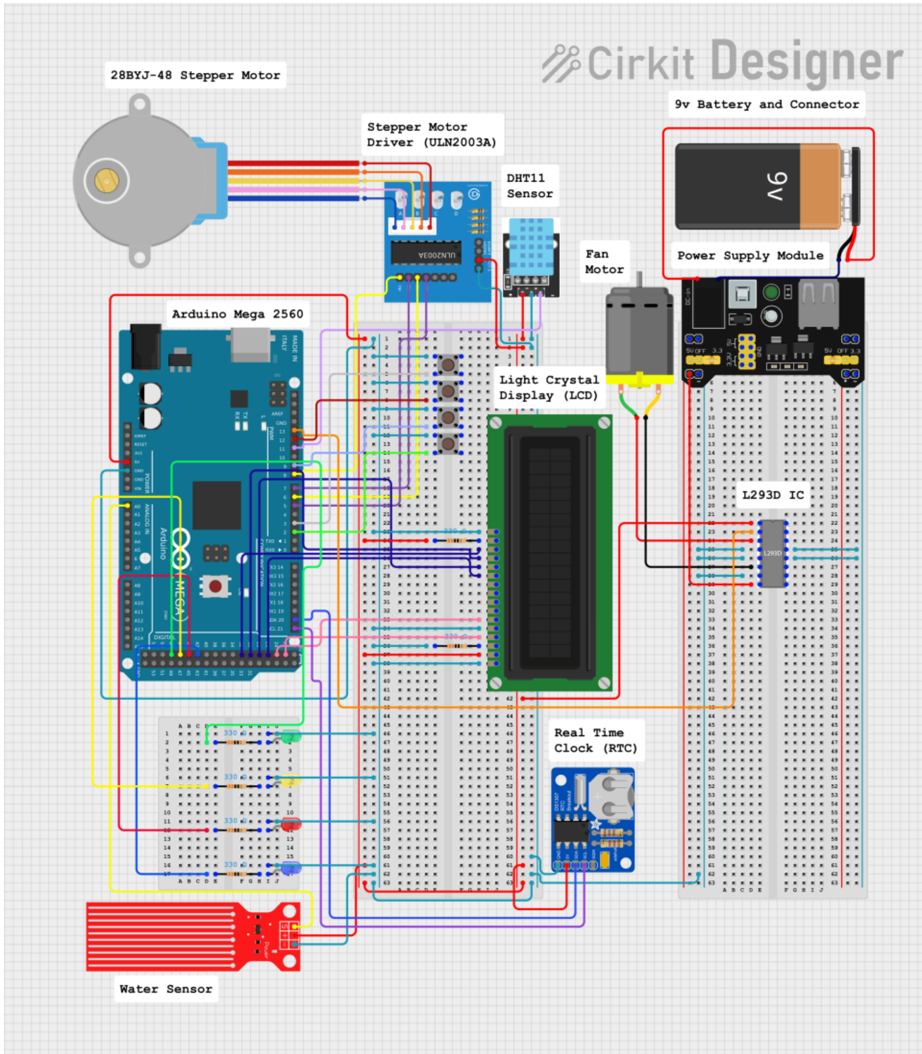

Figure 1: A schematic diagram of the completed swamp cooler.

After designing the diagram, our group rebuilt a physical copy of the circuit. All components were from the ELEGOO Mega R3 starter kit.

IV) Diagram and State Changes

There are four stages implemented in this circuit: disabled, idle, error, and running. Each stage is dependent on the other three stages and is necessary to replicate the swamp cooler's functions. In all stages, the real time clock (RTC) reports the time of each state transition and any changes to the stepper motor position. When the circuit isn't disabled, it continuously monitors and reports the humidity and temperature on the LDC each minute. It also responds to any changes in the vent position.

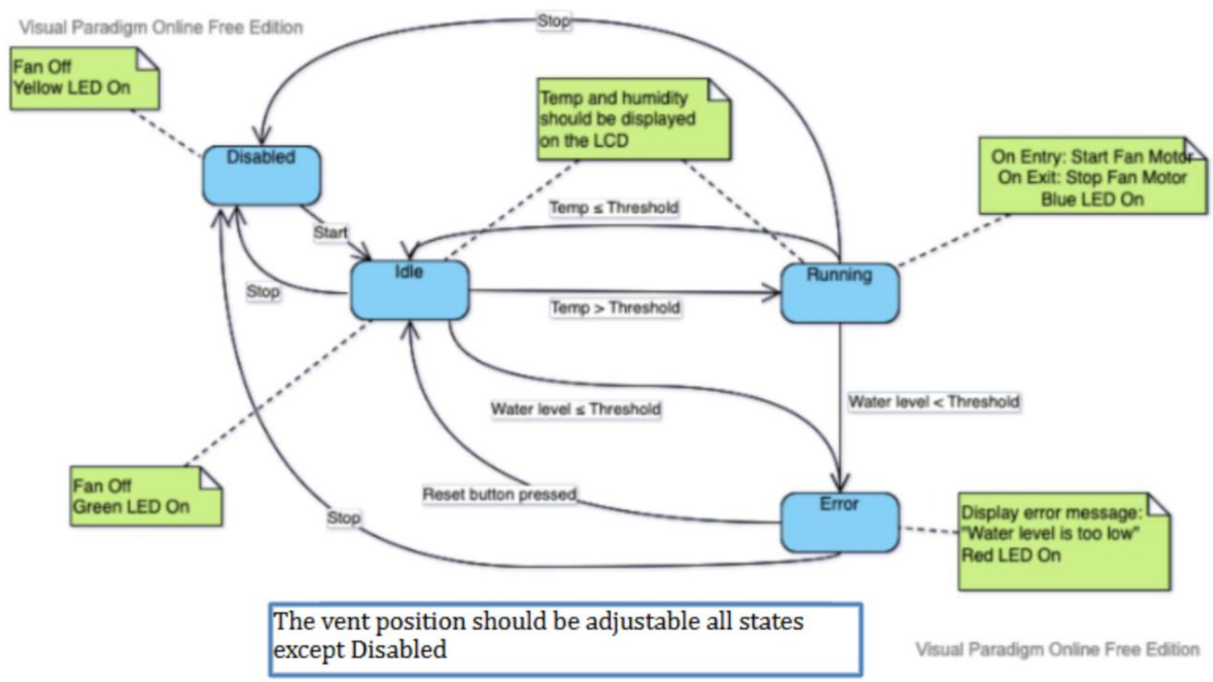

Figure 2: A flowchart of the circuit's four stages and transition requirements.

Disabled

This stage turns off all the components in the circuit and enables the yellow LED. Users enter this stage by pressing the stop button in any of the other three stages. Since the components are turned off, the temperature and water levels are not recorded in this stage. Users can leave this stage by pressing the start button.

Idle

This stage is depicted by the green LED being enabled on the circuit. The circuit enters this stage after the reset button is pressed when the water threshold is met in the error stage, or when the temperature drops below the threshold in the running stage. This stage continuously monitors the water level and temperature. The circuit leaves this stage if the water level drops below the threshold or if the temperature rises.

Error

This stage is depicted by the red LED being enabled on the circuit. The circuit enters this stage if the water level drops below the threshold in the idle or running stages. When in this stage, the fan motor is disabled, and an error message is displayed on the LCD. The circuit leaves this stage when the user replenished the water level above the threshold and presses the reset button.

Running

This stage is depicted by the blue LED being enabled on the circuit. The circuit enters this stage when the temperature rises above the threshold while in the idle stage. When in this stage, the fan motor is enabled. The circuit leaves this stage and returns to idle if the temperature drops below the threshold, or transitions to error if the water level drops below the water threshold.

V) Physical Circuit

The following photos are of the completed circuit at each stage.

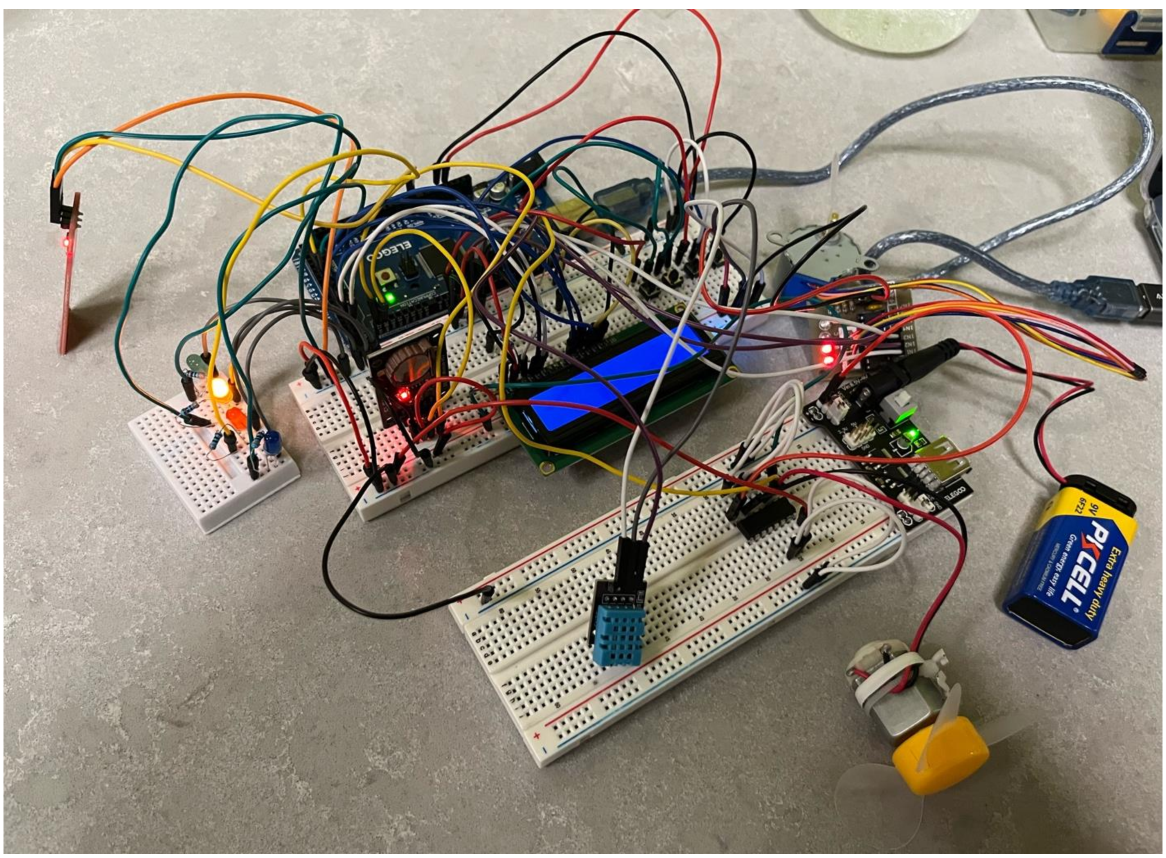

Figure 3: The circuit in the DISABLED stage.

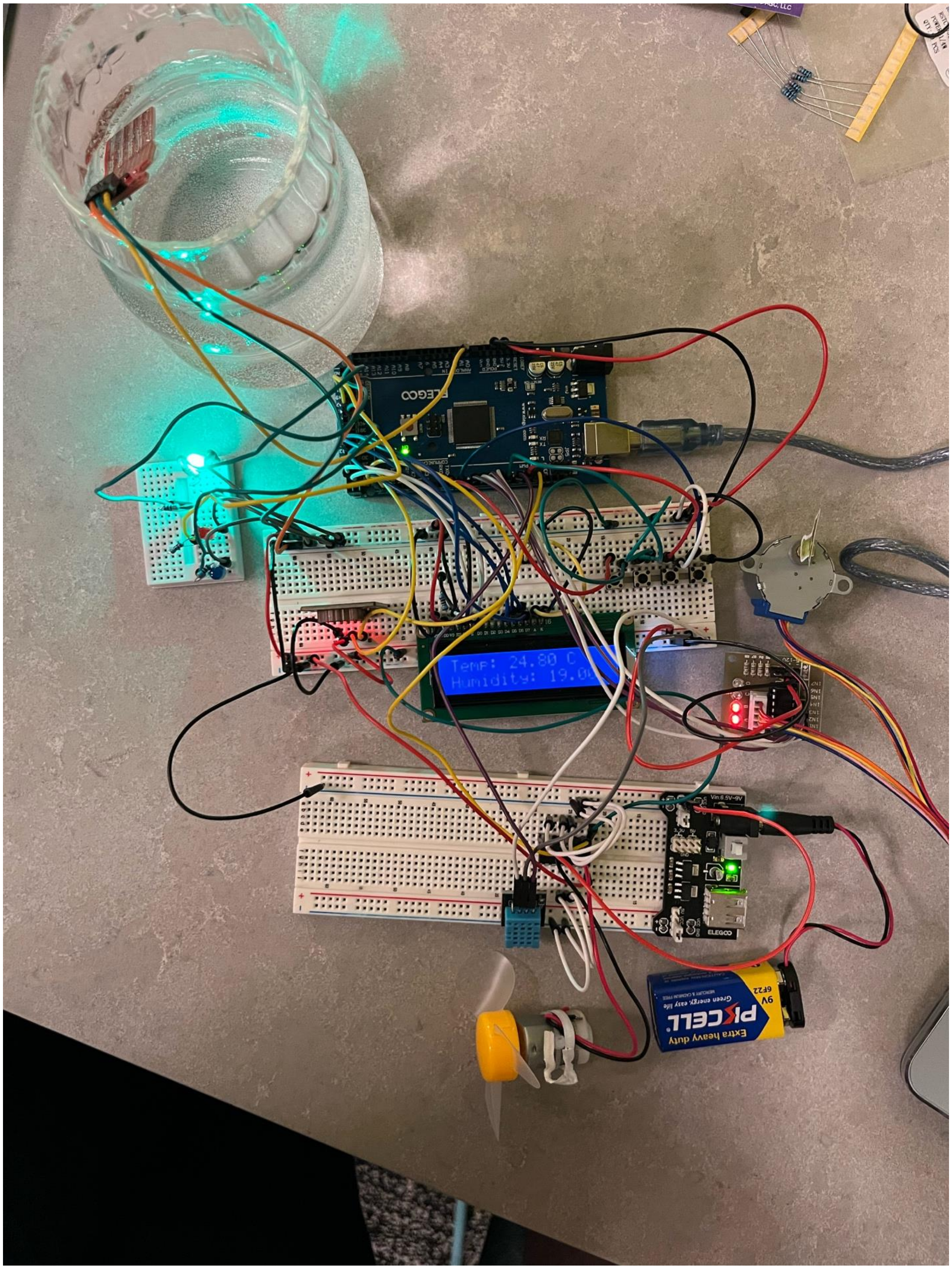

Figure 4: The circuit in the IDLE stage.

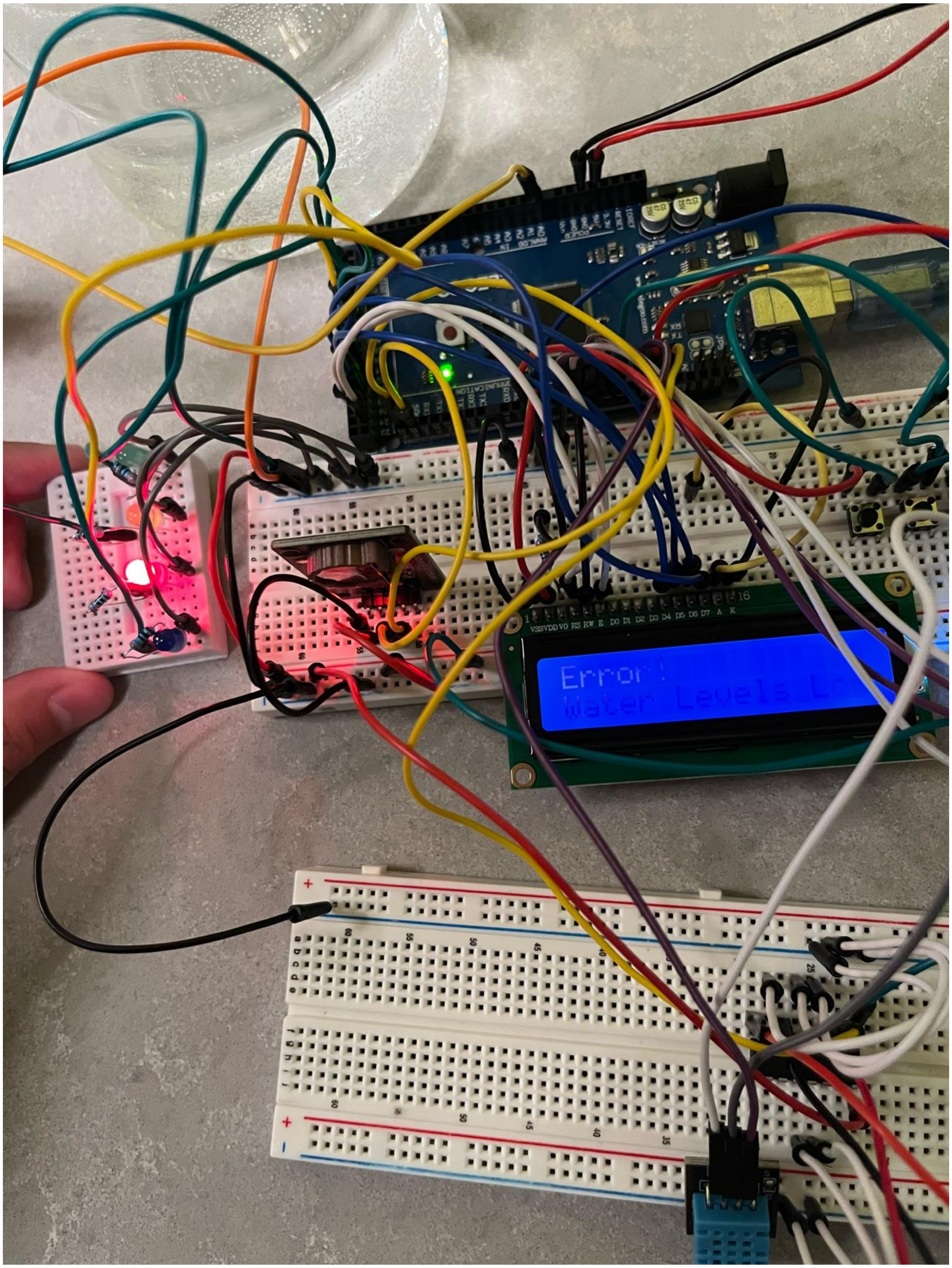

Figure 5: The circuit in the ERROR stage.

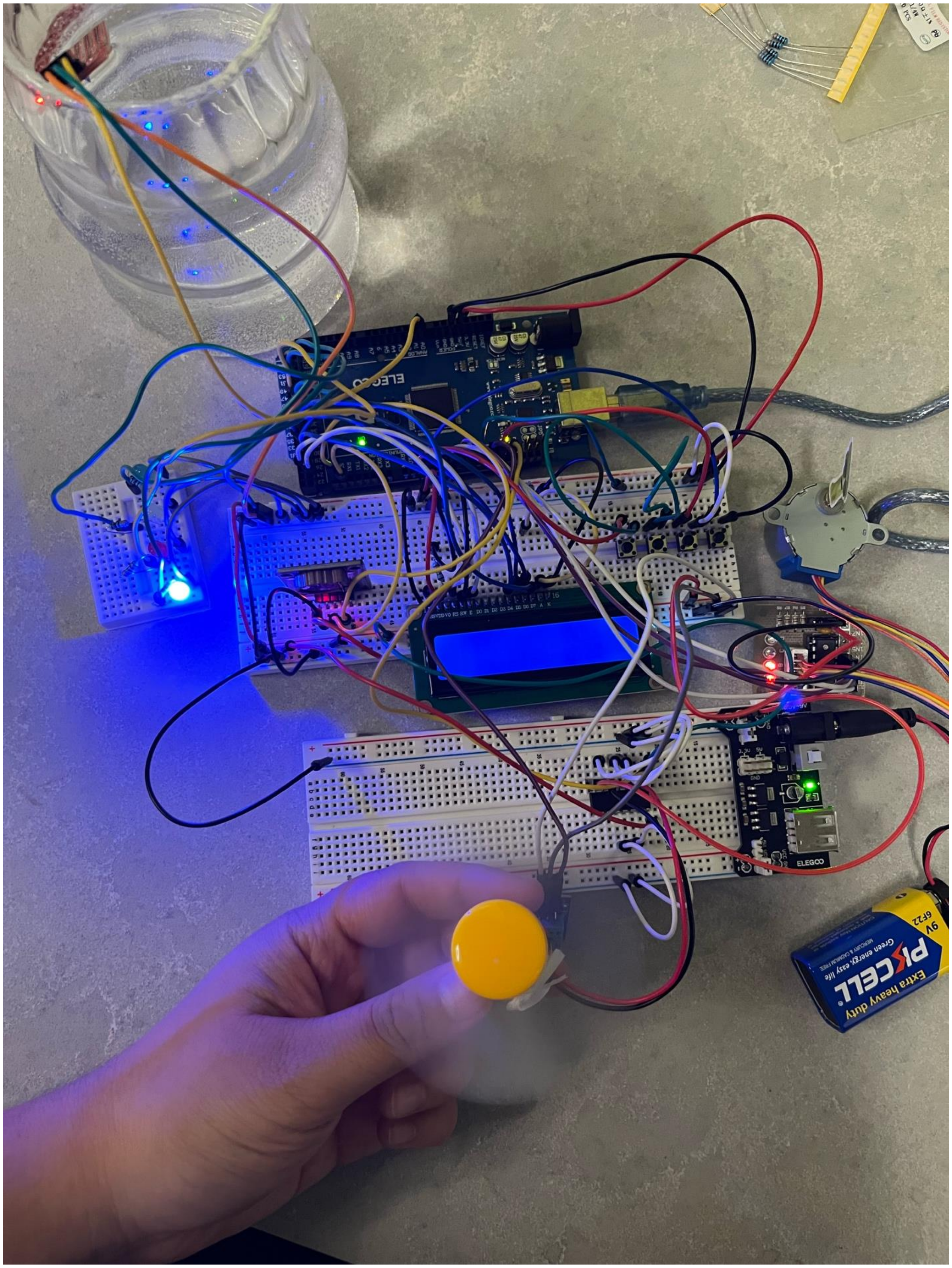

Figure 6: The circuit in the RUNNING stage.

VI) Environmental Impacts

Energy Efficiency

This circuit is designed to be energy efficient in the Arduino .ino file. To maximize efficiency, the code uses the pin registers instead of functions such as serialWrite(). This provides a faster runtime when operating the circuit.

Design Safety

To ensure safety, the circuit was designed with the fan motor to be implemented on a separate breadboard that was connected to the power supply module. This is needed because the fan motor component uses more power than the Arduino can provide, so it would damage the Arduino's output circuitry if connected directly. The power supply module is able to meet the motor's power requirements, so the fan motor, L293D chip, and power supply module are managed on a separate breadboard. This breadboard is then connected to the main breadboard with a common ground, but both boards maintain separate power supplies.

Affordability

This circuit design is affordable, as it was constructed using the ELEGOO Mega R3 starter kit and an additional breadboard. It would cost about $70 dollars for a beginner to buy these supplies. While it may be a pricier cost upfront, the finished circuit can be deconstructed, and users can reuse all materials for future projects.

Sustainability

The circuit is built using modular components on a solderless breadboard. As a result, the final circuit is a sustainable design, as damaged parts can be replaced without rebuilding the entire project. The project is also made from durable materials, as all wires come with a protective rubber coating, and components are built to be reused.

Accessibility

The circuit is accessible because of its ease to build and operate. The circuit is comprised of components that are included in the ELEGOO Mega R3 starter kit. Users can build this project with the starter kit or buy components individually using the component list depicted in the schematic diagram. The code is also available for users, which makes building the circuit easier. It is also simple to operate the circuit, as it is operated with the four buttons and the sensors attached to the circuit.