Final Project (CPE 301)

Grace Frazee & Jakob Kelley

December 2025

1 Project Description

The goal of this project is to design and evaporative cooling system, these are more energy efficient systems of air conditioning that use wet pads that take in hot hair and blow cool air into the home, commonly seen in hot environments. We will be designing one of these using our Arduino Mega 2560, where the Arduino will monitor all systems of the cooler, this includes the air temperature and humidity monitored by the DHT11 sensor, and reservoir water levels monitored by a water sensor. It was also manage a fan and a controllable vent using a stepper motor to imitate the function of an evaporative cooler. The DS1307 will keep a clock system for the Arduino to properly update changes every 60 seconds. It will run through the states ENABLED, DISABLED, RUNNING, and ERROR. All these states will be displayed on a connected LCD that will also show us the other relevant information such as the temp, humidity, and water level, giving the user more access to see what the Arduino is doing. If the air temp gets too warm the system should turn on to try and cool the air, and it will hit an error state if the water levels get too low. The user can reset these errors with a button or enable or disable the system with a button, giving the user ultimate control over how the system will act.

2 Component Details

2.1 Push buttons

- These are simple buttons that complete a circuit at the users discretion. In our project the buttons are used to cycle through 4 states: DISABLED, IDLE, RUNNING, and ERROR. The three buttons represent: START, STOP, and RESTART. Pressing START when the circuit is DISABLED will cause the state to change to IDLE. Pressing STOP at anytime, will make the circuit go back into the DISABLED state. If the RESET button is pressed during the ERROR State, the circuit will go back to being IDLE (reseting itself).

2.2 LEDS

- Small bulbs that light up when small currents are put through them, they come in various colors. Our project uses them to display what state the cooler is in, yellow being the off/disabled state, this becomes green which is our idle state, then there is blue which is running, and finally we use a red led to show an error state.

2.3 Resistors (330 amps)

- Small components added to a circuit to reduce the current going into a module or just to reduce strain on smaller wires, these are used to protect our LEDs from burning out. Another one is used for LCD display.

2.4 Water Level Sensor

- This sensor is used to check for the water level of the reservoir (or any body of water you need to measure level of). It checks the level by using the water as a bridge for each copper trace along the sensor, the higher the water level is on the sensor the more traces that get bridged increasing current and thus showing higher water level.

2.5 Stepper Motor and ULN2003

- The stepper motor is vent control of the cooler, changing position based on the state of the cooler. It is a motor that spins when current is ran through it, being bi directional when the current switches to open and close the vent. This works in tandem with the transistor ULN2003 which acts as a switch for the stepper. The transistor prevents constant flow through the stepper motor only allowing movement when given a current by the Arduino.

2.6 LCD Display

- The LCD display is a large screen that displays text based on which pixels are lit up, it can show text, symbols, and other items. The pixels are lit up by electrical signals given from the Arduino. We use the LCD to display the coolers state, the humidity and the temperature. It is updated one minute at a time with new sensor information.

2.7 Potentiometer

- This is a unique resistor that you can control by spinning the knob on top left or right to increase or decrease voltage. This is used with the LCD to prevent a burnout and change the brightness of the LCD display. Since the LCD is a critical component of the cooler this is also critical to keep the LCD in working order. Another potentiometer is used for the stepper motor (vent) to help control the direction.

2.8 Real-time Clock Module (DS1307)

- This clock module is used to keep track of time and make sure the system can work in real time, so it can respond to external changes properly. It is used so we can have the system update every 60 seconds, without this module we could not update the system to properly follow 60 second update intervals making the cooler work unpredictably in unpredictable time frames.

2.9 Temp/Humidity Sensor DHT11

The temp and humidity is monitored and reported by this sensor. The data it collects is sent to the Arduino and is updated in the system every minute and helps the cooler decide how the vent and fan motor should react to the environmental data to ensure that the area is cooled to the user preference.

2.10 DC Motor, Fan Blade, and L293D

- The DC motor and fan blade are what imitate the fan inside the cooler, it is a generator that spins when a current is applied to it. This motor is paired with the L293D chip that helps the Arduino moderate the motor's speed and which direction it spins, although in this project the fan will only spin one way, but the fan speed is modeled by the code, the Arduino sends these instructions to the chip so that the fan works as intended.

2.11 Power Supply and Adapter

- The power supply and adapter are used to supply power from wall outlets to the fan and stepper motor due to them requiring more power than the Arduino itself can power, without it the fan doesn't work as intended.

3 Environmental Impact

The environmental impact of this evaporative cooler is significantly less than a standard freon based AC unit. The energy needed to make the cool air relies on the natural thermal abilities of water and spinning a fan rather than using a compressor to compress freon to a pressure high enough that it becomes cold. Since water will just evaporate into the air over time it is much less dangerous to the environment than freon, which can damage ozone and is a hazard when breathed in. The lower power consumption saves the grid from extra strain of dozens to hundreds of potential ACs running, reducing how much coal is needed to run power plants. Our system also conserves energy by shutting off if there are any issues with the system, such as low water shutting off the system, or when the desired temperature is reached it will shut off, only turning back on when the temperature fall out of what the cooler is set to. Since this system uses most of its power to move a fan and the actual cooling is done through natural processes, it makes our cooler system very energy efficient, helping reduce any large carbon foot print people may think our cooler has.

4 Schematic Diagram

Link for the schematic:
`https://app.cirkitdesigner.com/project/c53c24d4-5727-4368-a3b1-651ccbbfccff`

4.1 Transition States

Disabled State: occurs at the beginning when the system is just powered up and when the stop button is pressed. It means that the system is inactive. The Yellow LED indicates the Disabled State. All components are off in this state.

Idle State: Occurs when the Start button is pressed when in the Disabled State. The fan is off in this state. In this state, the system is waiting for a condition: water + temperature meeting the designated thresholds. The Green Led indicates this state and the LCD displays the values for water, temperature, and humidity with a 1 minute delay. The stop button changes the state back to disabled. If the sensor values reach a certain value, the state will change to the run state.

Run State: Occurs when the sensors (water and temperature) are working and have reached the specifed values. The fan motor is activated once the temperature is high enough. If water level gets too low, the state will change to error. If the stop button is pressed, the state will change back to disabled. The blue LED indicates the running state.

Error State: Occurs when the water level is too low (according to a threshold decided in the code). the fan motor is off and an error message is displayed on the LCD. The Red LED indicates the error state. By pressing the stop button, the state is changed back to disabled. If the reset button is pressed, the system is returned back to idle.

During all the states, updates are displayed on the serial monitor and uses the Real-time clock Module to get time stamps. The Stepper Motor (vent) is active during all the states except for disabled.

4.1.1 Examples of Transitions

1. Disabled THEN Start button pressed THEN Idle
THEN Sensor Values at Threshold THEN Running
2. Idle THEN Stop button THEN Disabled
3. Running THEN Stop button THEN Disabled
4. Idle/Running THEN Water Level Too low THEN Error
5. Error THEN Reset Button Pressed THEN Idle

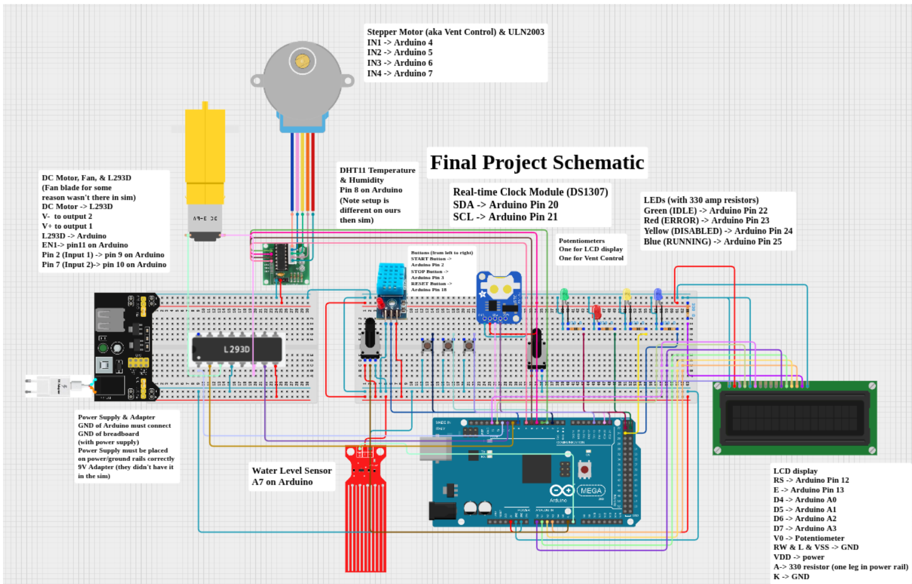

Figure 1: Picture of the Schematic

5 System Demonstration

5.1 Link to Video

`https://photos.app.goo.gl/3J14KYo8edpAbQJZ9`

5.2 Pictures of Final System

5.2.1 Picture of the Circuit as a Whole

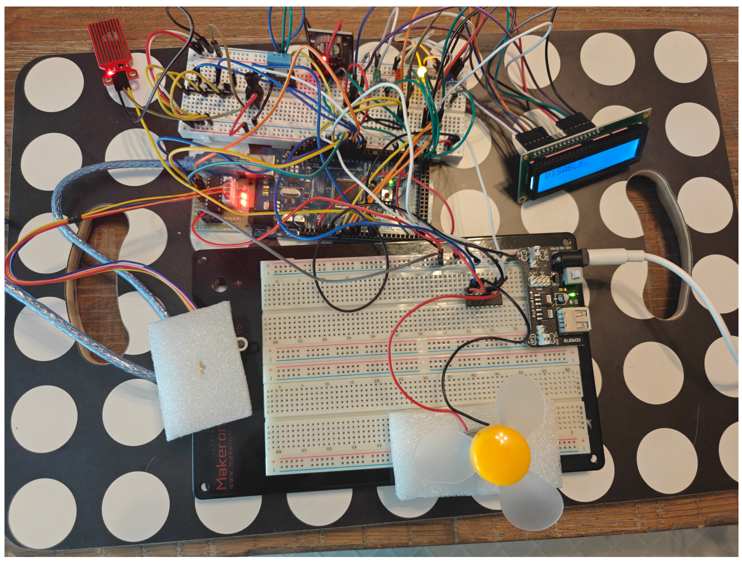

Figure 2: Picture of the Circuit