Final Project Report

Swamp Cooler

Names: Farzana Tanni & Lorie Phan

Professor: Bashira Akter Anima

Due Date: 12/15/2023

Class: CPE 301 Embedded Systems

Overview *1.1*

We were tasked with building a working swamp cooler using the Arduino ATmega 2560 and sensors from the kit. The cooler works by drawing fresh air from the external environment through a pad soaked in water. The evaporation of the water will then be used to cool and humidify the air.

To achieve this task, we created functions that monitor the water levels and let us know when the water levels are too low by printing an alert. Additionally, when the temperature falls out of the range, the fan motor is supposed to be enabled or disabled accordingly. All updates on temperature and humidity levels will be displayed on the LCD screen. The date and time will also be displayed using a clock function.

The system has four different states: disable, idle, error, and running. In the disabled state, the yellow LED will be turned on and there should be no monitoring of temperature or water. In the idle state, the green LED should be on and the water level should be continuously monitored. In the error state, the red LED should be turned on and an error message should be displayed on the LCD. If the reset button is pressed during the error state, then the cooler state will be switched to idle if the water level is above the threshold. Lastly, in the running state, the blue LED should be on and the fan should be running.

System Restraints *1.2*

The main restraints we had to work with were temperature, power usage, complexity of assignment and components available to us. Temperature was hard to work with because depending on how hot or cold our water was, the system would not work correctly, resulting in many changes to our Arduino program. Because of this, it took us a while and messed up our coding in the end. Continuing, power usage was also a big factor on our Arduino circuit not working, as what we are trying to achieve would realistically use more than 5V. This adds onto our next restraint, as the complexity of the assignment was hard to perform since we were given the task of using newer parts in order to make something bigger. Lastly, because of the complexity, parts that were available to us were not enough to perform our assignment. These restraints could be due to over complicating the assignment.

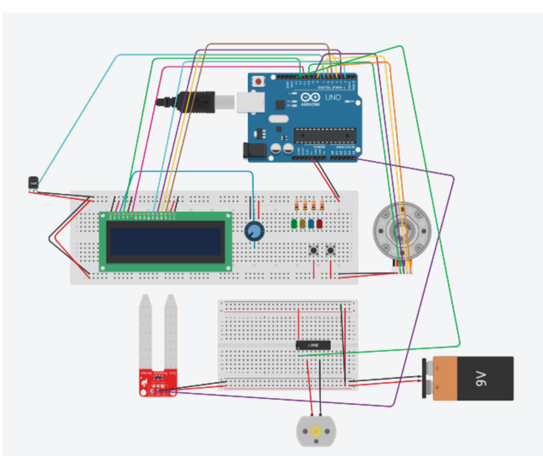
Errors and Problems *1.3*

We had many problems, some were not fixable in the end.

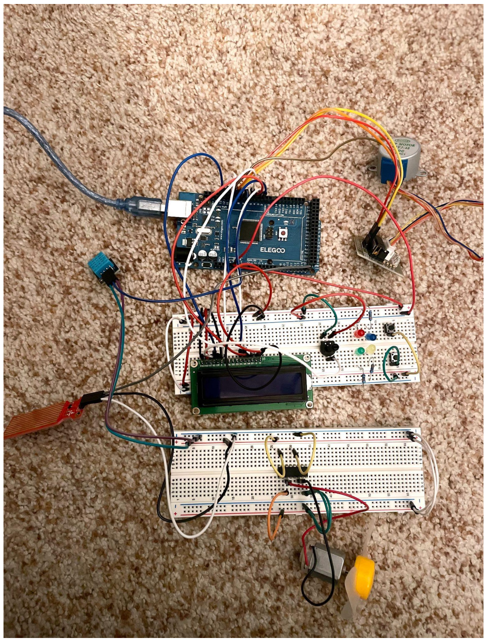
- Coding errors: Several components of our Arduino code were not able to work, potentially due to both of us working on similar functions at the same time. We also rewrote and changed code multiple times, which caused errors due to the overlap of code.
- Wiring and pins: A big error was that since there were many parts, keeping track of the pins and registers was hard, especially when we took it apart multiple times. We also changed around wires often and experimented, which led to confusion on what wires actually worked.
- We also had lots of noise coming from our breadboard, causing us to be worried about our breadboard's functionality.

Figures *1.4*

Technical Diagram



Overall Circuit



Links *1.4*

Github: https://github.com/1001-Tanni-Farzana/FInal-Project-CPE-301

Video: https://youtube.com/shorts/Z1q-77uXWzI?si=fOv-laeZG1SV0sbh