CPE 301: Final Project Overview

By: Eylon Borenstein, Kyle Hedin, Gianluca Hunt

12/12/22

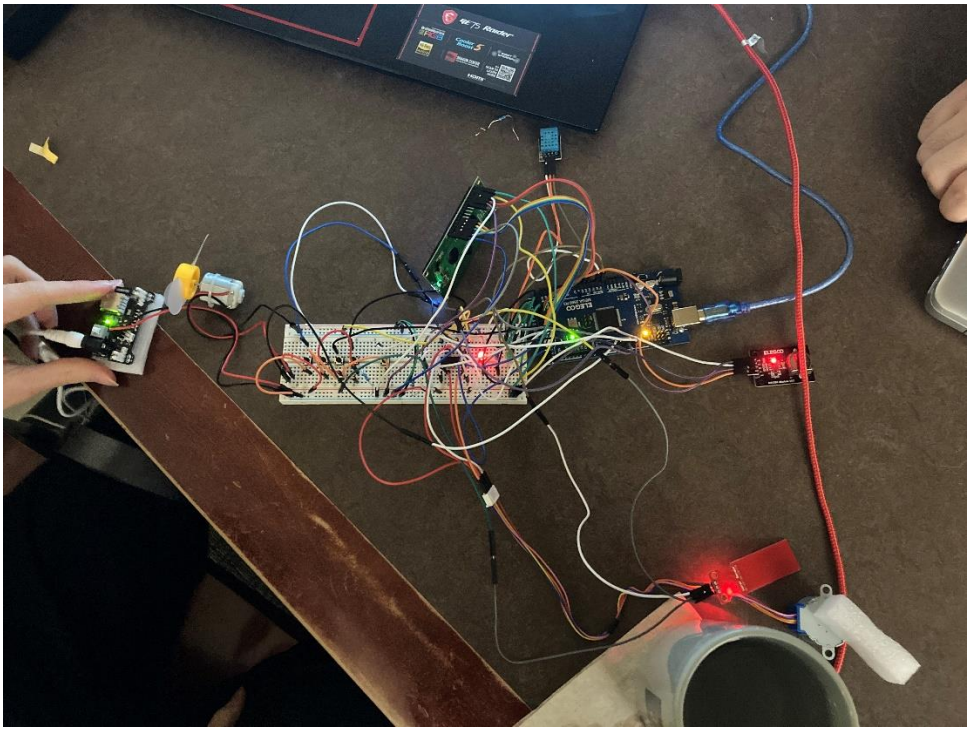
Design Overview

Using the Arduino 2560 mega control board, our group were able to code, build, and utilize a swamp cooler. The swamp cooler utilizes multiple components including the Arduino 2560 mega, a water level sensor, stepper motor, LCD display, temp and humidity sensor, motor and fan, and an external power supply. The swamp cooler has a few different states, depending on the amount of water available for the water level sensor. If the water level is at the correct level, an LED light will turn green, and the temperature and water level will be displayed on the LCD monitor. This means that the swamp cooler in the IDLE state, and is working normally, with the fan turning on.

In the ERROR state, the water level is below the necessary threshold, turning off the motor, and displaying an ERROR message on the LCD screen, along with a red LED light being turned on. In the RUNNING state, the water level is at the correct threshold, but the temperature is below the threshold. This will activate the fan causing the temperature to drop putting it into the RUNNING state. This will also activate the blue LED, and will continue to run until the temperature is below the threshold again. The last state is the DISABLED state in which nothing is being displayed, and a yellow LED is on. In all states except DISABLED, the stepper motor will act as a vent, giving a direction to the air flow. The temperature/humidity sensor is responsible for recording the temperature and displaying it to the LCD screen. The rest of the system will act accordingly to the data given from the temperature sensor, as wall as the water level sensor, and place itself into the correct states depending on threshold levels. A reset button is also installed, which will move the states form ERROR back to IDLE, when the water level and temperature is at the correct threshold, and the button is pressed.

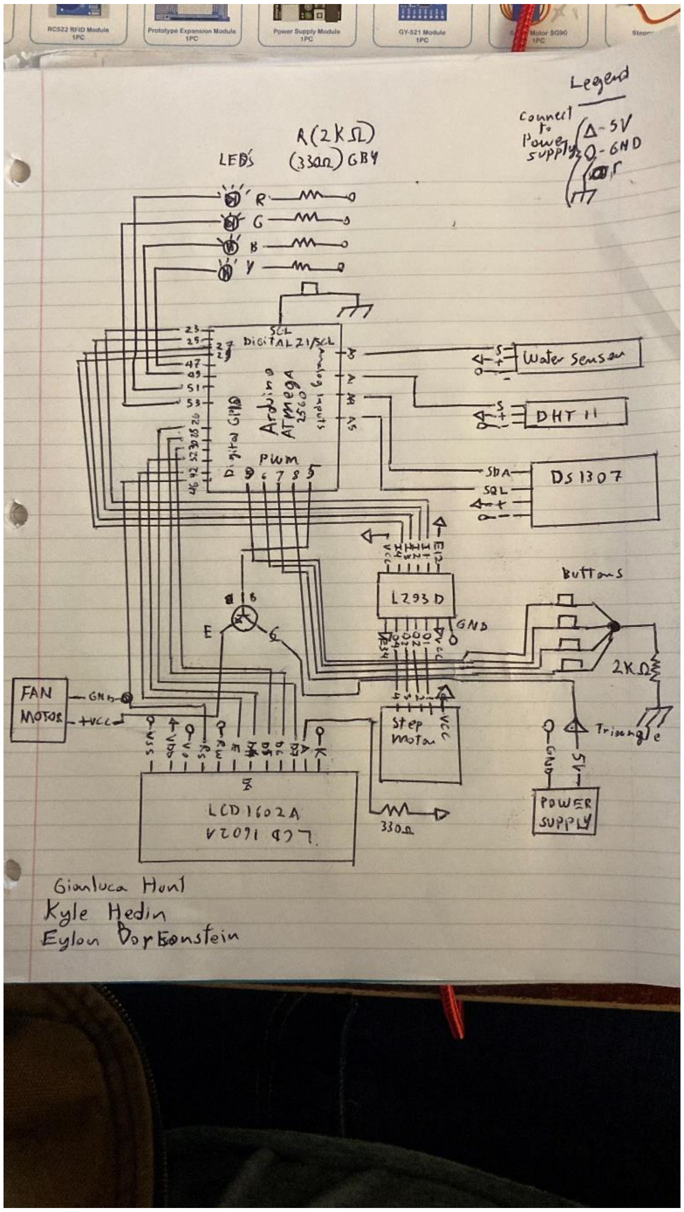
Some of the constraints on the system is the inability to change the temperature and water thresholds manually on the board. It requires changing the code and re-uploading it to the Arduino to be able to change the threshold. Another constraint, is the need to manually replace the water when the levels reach below the threshold, placing it into the ERROR state.

Pictures/ Video Link



https://youtu.be/vQBe8g5abxQ

Schematic



RC522 RFID Module
Prototype Expansion Module
Power Supply Module
GY-521 Module
Legend
Connect to Power supply
Δ-5V
O-GND
LED's
R(2KΩ)
(330Ω) GBY
R
G
B
Y
Digital 21/SCL
Arduino
ATmega 2560
Digital GPIO
PWM
Water Sensor
DHT 11
SDA
SCL
DS1307
L293D
GND
Buttons
2KΩ
FAN MOTOR
GND
+VCC
Step motor
VCC
Triangle
GND
5V
POWER SUPPLY
LCD1602A
330Ω
Gianluca Hunt
Kyle Hedin

Component Sheet

Water Level Detection Sensor Module 1PC

https://curtocircuito.com.br/datasheet/sensor/nivel_de_agua_analogico.pdf

LCD1602 Module (With pin header) 1PC

https://www.waveshare.com/datasheet/LCD_en_PDF/LCD1602.pdf

DHT11 Temperature and Humidity Module 1PC

https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf

Stepper Motor SG90 1PC

https://www.mouser.com/datasheet/2/758/stepd-01-data-sheet-1143075.pdfPower Supply Module 1PC

https://static.rapidonline.com/pdf/73-4538_v1.pdf

Mega 2560 Controller Board 1PC

https://www.arduino.cc/en/uploads/Main/arduino-mega2560-schematic.pdf

DS1307 RTC Module 1PC

https://www.analog.com/media/en/technical-documentation/data-sheets/ds1307.pdf

NPN Transistor S8050

http://media.nkcelectronics.com/datasheet/s8050.pdf

Github Link

https://github.com/eborenstein/CPE-301-Final-Project