CPE 301 Final Project Overview

Morgan Young

Dylan Perkins

CPE 301 - Team Project

Group 404

Project Overview Document

Our goal for this assignment was to keep the implementation of the code as clean, simple, and readable as possible as well as the hardware. There were a lot of moving pieces to this project, so keeping things clean and simple was our first priority.

Design

Constraints

The temperature variable had to be changed depending on the environment we were in and we had to do quite a bit of trial and error to figure out the right number. We eventually landed on 75 for the temperature threshold. The water level sensor was also quite fickle and we had to modify this number several times as well. We ended up landing on 150 for the water level threshold.

Hardware Observations/Challenges

We encountered a lot of hardware-related challenges throughout this project. One of the challenges came from the constant movement that the system had to go through. Moving it from place to place proved to be problematic as wires came loose and affected the outcomes of our trials. In addition, we encountered several problems with

faulty hardware. On a random trial, our yellow LED was not working suddenly. We initially thought that it was a mistake in the code, but it turned out that the LED was just fried. As soon as we replaced it with another LED, it worked. The same happened for the button. In a trial run, the serial monitor showed no input being read from the button, even though it had been working minutes prior. We replaced the button, and it worked just fine.

Code Implementation - Libraries

For this project, we utilized four libraries, denoted as “DHT.h,” “LiquidCrystal.h,” “Wire.h,” and “RTClib.h.” The “DHT.h” series is a library for the temperature/humidity sensory aspect of the project. The “LiquidCrystal.h” library enables the LCD display to be controlled by the Arduino Mega. The “Wire.h” library enables communication between the various hardware pieces and the Arduino Mega. The “RTClib.h” library enables the RTC DS1307 (real-time clock) module.

Naming Conventions - Variables

To keep the naming conventions for our input and output variables as straightforward as possible, we chose chose obvious names for them. Our output variables are respectively initalized as “**yellowLED**,” “**greenLED**,” “**blueLED**,” “**redLED,”** and “**motor**.” We chose a similar naming style for the rest of our input(s) and initialized our button as “**button.”** We initialzied our binary states (0/1 or T/F) in a similar fashion with the button state being represented by “**buttonState”** and the state of the hardware by “**state**.”

Programming Overview

The Setup

The setup initializes all of the LEDs and the motor to the output setting, and the button to the input setting. It begins the digital humidity and tempeature sensing and writes all of the LEDs to their low setting.

The Main Loop

The main function of our program consists primarily of a switch statement that has 3 cases and 1 default case. First, the program checks the humidity, assigns a value to the humidity variable, checks the temperature, assigns a value to the tempature variable, and checks whether the button is pressed or not. Once these values are checked and assigned, it runs the switch statement.

The Switch Statement

Case 1

Case 1 first checks if the value of the water level sensor is less than 150. If it is, it changes the state to 3 and the switch statement reverts to case 3. If it's not, it runs a second check to see if the temperature is above 75. If the temperature is greater than 75, it changes the state to 2 and the switch statement reverts to case 2. If neither of these conditions are met, the system enters the "IDLE" state, which is represented by the green LED. In this state, the temperature and humidity will be displayed in the LCD screen.

Case 2

Case 2 checks if the temperature is less than 75; if it is, it changes the state to 1 and the switch statement reverts to case 1. If it's not, it then checks if the value of the water level sensor is less than 150. If the water level is less than 150, it changes the state to 3 and the switch statement reverts to case 3. If neither of these conditions are met, the system enters the "RUNNING" state which is represented by the blue LED. In this state, the temperature and humidity will be displayed on the LCD screen and the fan will run in order to "cool down" the room.

(NOTE: We had to mess with the threshold for the fan quite a bit, since the temperature of the room we were in kept changing. We ended up landing on 75. In addition, the fan wouldn't stop without us manually stopping it since the little fan in the kit wasn't going to change the temperature of the massive room we were in.)

Case 3

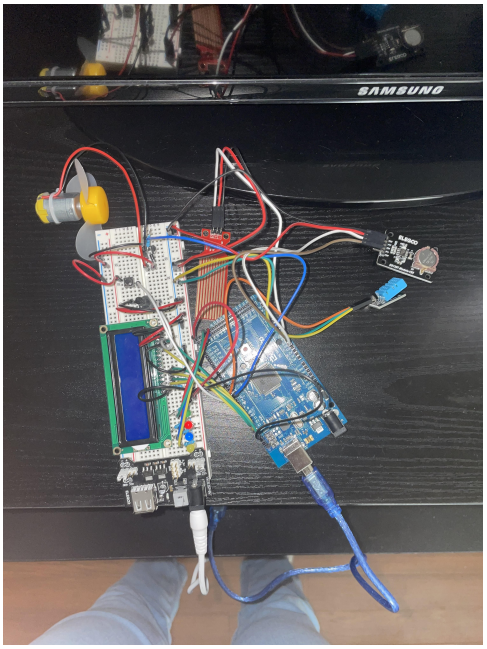
Case 3 checks if the value of the water level is more than 150. If it is, it changes the state to 1 and the switch statement reverts to case 1. Otherwise, the system enters the “ERROR” state, which is represented by the red LED. In this state, the LCD will just display “Error” and nothing else will happen.

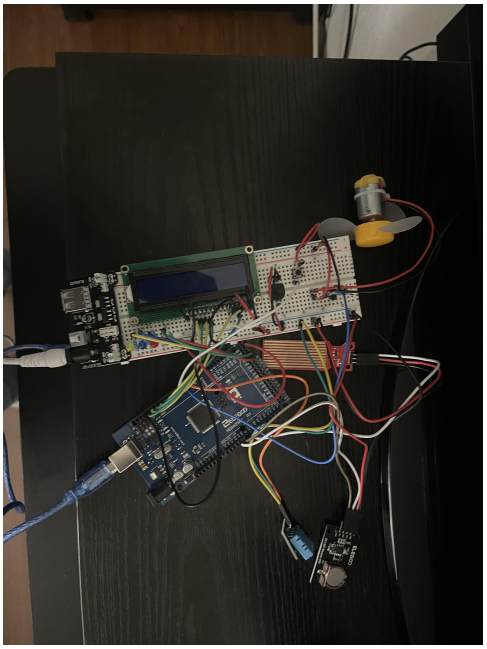
Default Case

The only condition for the default case is for the buttonState to be “HIGH” or in other words, for the button to be pressed. If the button is pressed, the state will be set to 1 and the switch statement will revert to case 1. If not, the system will enter the “DISABLED” state, which is represented by the yellow LED. In this state, the LCD will just display “Disabled” and nothing else will happen.

Demonstration

Photos



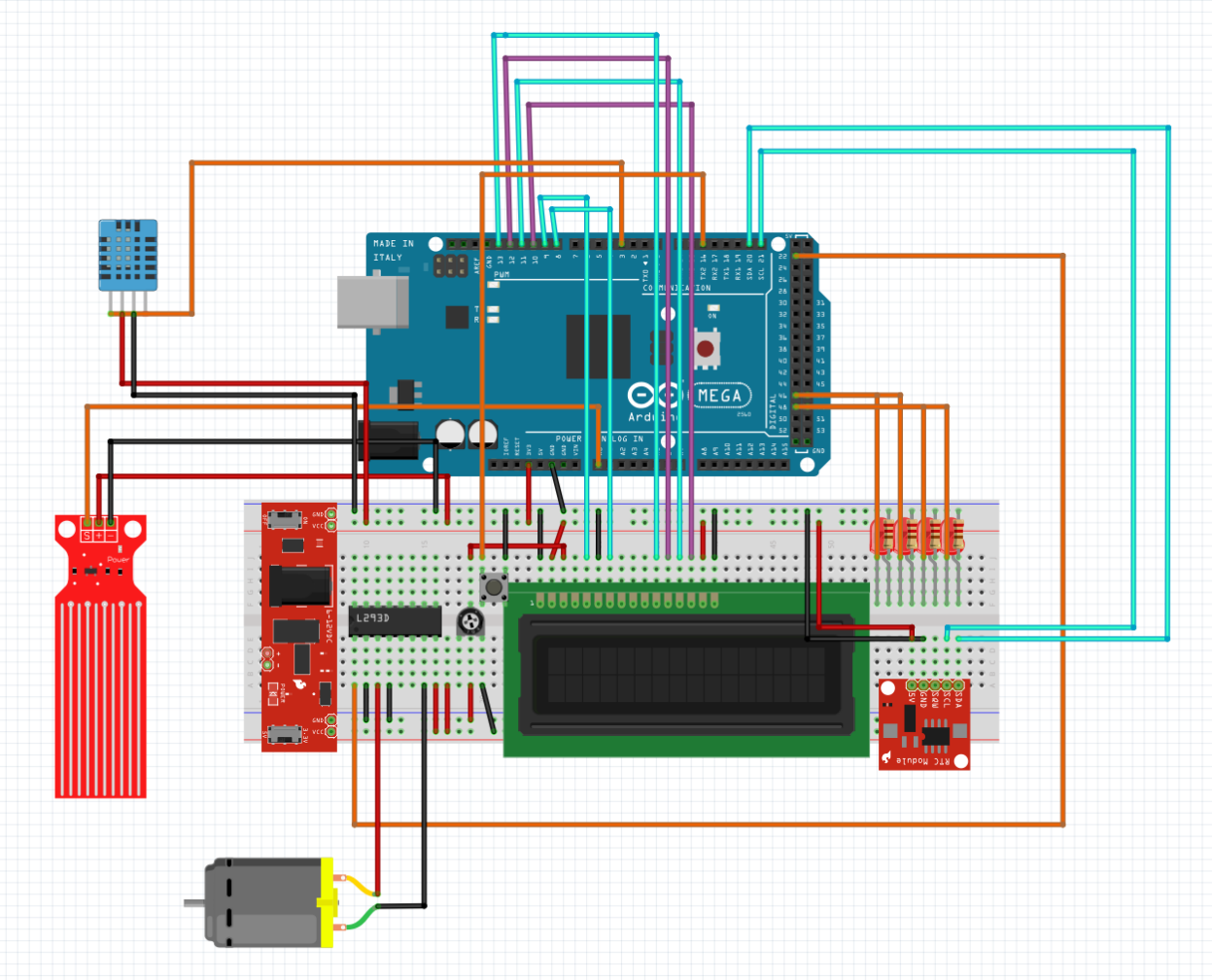


Video

CPE 301 - Final Project Video Demo.MOV	https://drive.google.com/file/d/1sivM8XWg24XtFvs7eFMK0lBFRdRdGg1Z/view?usp=drivesdk

Text link: https://drive.google.com/file/d/1sivM8XWg24XtFvs7eFMK0lBFRdRdGg1Z/view?usp=sharing

Schematics



Relevant Specification Sheets ~

https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf

Text Link: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf

GitHub Link

https://github.com/missmorganyoung/CPE301-Final-Project

Text Link: https://github.com/missmorganyoung/CPE301-Final-Project