CPE 301 Embedded Systems Design FINAL PROJECT

Paris Acosta, Ankeet Singh, Brendan Smith

May 7, 2024

1 Abstract

The project's objective is to build a swamp cooler, or evaporation cooling system. In arid, hot areas, evaporation coolers offer a more energy-efficient substitute for air conditioners. They function by drawing in air from the exterior via a water-soaked pad.

The air gets cooler and more humid due to water evaporation. A water level sensor, temperature and humidity sensors, a fan motor, colorful LEDs, push buttons, and a liquid crystal display are just a few of the parts that this project will make use of.

2 Experimental Design

In this project, we'll build a swamp cooler using an Arduino ATmega2560. This project will make use of colorful LEDs that will show the various states of the cooler, push buttons, a fan motor driven by an additional supply and an L293D IC, a liquid crystal display to show the humidity and air temperature,

DHTH water level sensor, humidity and temperature sensors. The cooler works by keeping an eye on the water level in a cup and alerting the user if it drops too low. Additionally, the system logs the date and time that the motor begins and stops. The area's current humidity and temperature are displayed on the liquid crystal display. Using the DS1307 module, we tested our swamp cooler and confirmed a temperature of 20°C. An error message titled "ERROR: water level low" appears on the LCD if the water level falls below the threshold. To find the display's pins, we utilized the LiquidCrystal.h library. Additionally, we used RTClib.h to create the DateTime() method and get the desired outcomes. The L293D ic chip powers the fan motor, which only turns on when the system is set up with the right water levels and temperature readings. The motor's behavior is determined by the setFanMotor() method, which enables it to alternate between the operating, idle, and disabled states depending on the circumstances. The power supply, which provides an extra direct power source, is attached to the motor. Additionally, the direction of the motor may be adjusted by utilizing the control pins and the L293D pin-out chart. Using the control pins, we are able to regulate the motor's rotational direction. In our project, we employed four distinct colors of LEDs to represent the various stages of the various project components. The system is shown as being in one of three states: disabled (shown by the yellow LED), idle (shown by the green LED), and active (shown by the blue LED) when all conditions are satisfied. On the other hand, when an error occurs, indicating that the idle state has been disturbed, the red LED glows. A unique pin is allocated to every hue of LEDs in order to distinguish between different system states. Three push buttons, each with a distinct purpose, were employed. Toggling between the activated and deactivated states. We utilized the buttons for start and stop, respectively. We also added a reset button, which is used to return the system to its initial idle state when it is not in use. The system's current status governs both the stepper motor and vent direction. We utilized a thermistor to measure the temperature within the temperature and humidity sensor. Once the values are recorded, the thermistor and capacitive humidity sensor work together to measure the humidity and send a signal to the data pin. The DHT11 sensor's temperature and humidity values are recorded by the system via the read11() function of the DHT.h library. DHT11 PIN 7 is used to define the sensor. The system's idle and operating states are alternated in the temperature reading. The water level sensor keeps an eye on the amount of water that comes in from the outside. It can transition between the idle, running, and error modes once it detects a liquid. The water level is determined by the system using the adc read() function. An error status is returned if the value measured is less than the WATER THRESHOLD value, which is the water threshold value.