Christopher Mengu
CPE 301

CPE Final: Water Cooling System

Design:
The system's main functionalities include monitoring water levels, displaying temperature and humidity, controlling a fan motor based on temperature, adjusting the angle of an output vent, enabling/disabling the system, and recording the motor's on/off times.

Components:
- Arduino 2560: Microcontroller used in project
- Water level sensor: Monitors the reservoir's water level and prints an alert if the level is too low.
- Temp/humidity sensor (DHT11): Monitors and displays the air temperature and humidity on an LCD screen.
- Fan motor: Starts and stops based on specified temperature range.
- Stepper motor: Controls the output vent angle, adjusted by the user.
- On/off button: Allows the user to enable or disable the system.
- Reset button: Allows the user to reset from error state
- Real-time clock module: Records the time of state changes
- Potentiometer: Used to control LCD screen brightness and stepper motor
- Various Resistors: Resistors used to control flow
- Battery: Feeds into stepper motor for power.

Libraries Used:
- Temp and Humidity Sensor:
https://reference.arduino.cc/reference/en/libraries/dht-sensor-library/
- LCD Screen:
https://www.arduino.cc/reference/en/libraries/liquidcrystal/
- Real Time Clock Module:
https://www.arduino.cc/reference/en/libraries/rtclib/
- Stepper Motor:
https://www.airspayce.com/mikem/arduino/AccelStepper/

Important Info:
- The threshold for the water detection before it enters an error state is a reading that is 100 above whatever reading it begins at
- The threshold for temperature before it enters the fan turns on ins 22.5 degrees (Celsius)
- The power delivered to the stepper motor is with a 9V battery.

Constraints:
- The system is designed for dry, hot climates and does not work in humid environments.
- The water level monitoring must use the provided water level sensor. Threshold detection can use either an interrupt from the comparator or a sample using the ADC. ADC library use is not allowed.
- Vent direction control must be implemented using the stepper motor. Buttons or a potentiometer can be used for vent control. Arduino libraries for the stepper motor are allowed.

- LCD display must be used for required messages. Arduino library for the LCD is allowed.
- Real-time clock module must be used for event reporting. Arduino library for the clock is allowed.
- The DHT11 sensor must be used for temperature and humidity readings. Arduino library for this sensor is allowed.
- The kit motor and fan blade must be used for the fan motor. A separate power supply board is required to avoid damaging the Arduino output circuitry.

System States:
The cooler continuously cycles through various states (Disabled, Idle, Error, Running) based on user input, temperature changes, and water level changes. State transitions are recorded in the serial monitor.

- Disabled: No monitoring of temperature or water levels occurs. The start button is monitored using an ISR.
- Idle: Water level is continuously monitored, and the system transitions to Error or Running state based on water levels and temperature.
- Error: Fan motor remains off regardless of temperature. A reset button can transition the system back to Idle state if the water level is above the threshold.
- Running: Fan motor is on, and the system transitions to Idle or Error state based on temperature or water level changes.

The provided code initializes and configures the necessary components and system states, including the stepper motor, LCD display, temperature and humidity sensor, real-time clock module, and water level sensor. The code also includes functions to control the fan motor, adjust the vent angle, and monitor the water level.

GitHub:
https://github.com/2111-Mengu-Chris/CPE-301-Final-Project

Demonstration Link:
https://youtu.be/mCP5Qg7tzvs