#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

// Define DHT22 sensor

#define DHTPIN 2 // Digital pin connected to the DHT sensor

#define DHTTYPE DHT22 // DHT22 sensor type

DHT dht(DHTPIN, DHTTYPE);

// Define buzzer and resistor

const int buzzerPin = 7;

const int resistorValue = 1000; // Adjust based on your buzzer and preference

// Define button pins

const int tempUpButtonPin = 3;

const int tempDownButtonPin = 4;

const int humidityUpButtonPin = 5;

const int humidityDownButtonPin = 6;

// Define LCD parameters

const int lcdColumns = 16;

const int lcdRows = 2;

LiquidCrystal\_I2C lcd(0x27, lcdColumns, lcdRows);

// Other variables

float tempSetPoint = 20.0; // Adjusted temperature set point

float humiditySetPoint = 50.0; // Default humidity set point

const float tempStep = 0.5; // Temperature adjustment step

const float humidityStep = 0.5; // Humidity adjustment step

void setup() {

// Start serial communication

Serial.begin(9600);

Serial.println("Setup started...");

// Initialize DHT sensor

dht.begin();

// Set up buzzer pin and button pins

pinMode(buzzerPin, OUTPUT);

pinMode(tempUpButtonPin, INPUT\_PULLUP);

pinMode(tempDownButtonPin, INPUT\_PULLUP);

pinMode(humidityUpButtonPin, INPUT\_PULLUP);

pinMode(humidityDownButtonPin, INPUT\_PULLUP);

// Initialize LCD

lcd.begin(lcdColumns, lcdRows);

lcd.clear();

// Set initial temperature set point

tempSetPoint = 30.0; // Adjust to a value lower than the ambient temperature

}

void loop() {

// Read temperature and humidity from DHT22 sensor

float temperature = dht.readTemperature();

float humidity = dht.readHumidity();

Serial.println("Inside loop...");

Serial.print("Temperature: ");

Serial.println(temperature);

Serial.print("Humidity: ");

Serial.println(humidity);

// Debug print statements for button states

Serial.print("Temp Up Button: ");

Serial.println(digitalRead(tempUpButtonPin));

Serial.print("Temp Down Button: ");

Serial.println(digitalRead(tempDownButtonPin));

Serial.print("Humidity Up Button: ");

Serial.println(digitalRead(humidityUpButtonPin));

Serial.print("Humidity Down Button: ");

Serial.println(digitalRead(humidityDownButtonPin));

// Display values on LCD

displayValues(temperature, humidity);

// Adjust set points based on button presses

adjustSetPoints();

// Check if temperature or humidity exceeds the set points

checkThresholds(temperature, humidity);

// Add a delay to avoid rapid readings

delay(1000);

}

void displayValues(float temp, float humidity) {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Temp: ");

lcd.print(temp);

lcd.print(" C");

lcd.setCursor(0, 1);

lcd.print("Humidity: ");

lcd.print(humidity);

lcd.print(" %");

delay(2000); // Increase the delay to 2 seconds

}

void adjustSetPoints() {

Serial.println("Adjusting set points...");

// Adjust temperature set point

if (digitalRead(tempUpButtonPin) == LOW && tempSetPoint < 100.0) {

tempSetPoint += tempStep;

}

if (digitalRead(tempDownButtonPin) == LOW && tempSetPoint > -50.0) {

tempSetPoint -= tempStep;

}

// Adjust humidity set point

if (digitalRead(humidityUpButtonPin) == LOW && humiditySetPoint < 100.0) {

humiditySetPoint += humidityStep;

}

if (digitalRead(humidityDownButtonPin) == LOW && humiditySetPoint > 0.0) {

humiditySetPoint -= humidityStep;

}

}

void checkThresholds(float temp, float humidity) {

Serial.println("Inside checkThresholds");

// Check if temperature exceeds the set point

if (temp > tempSetPoint) {

activateBuzzer();

lcd.clear();

lcd.print("High Temp! Alert");

delay(2000);

lcd.clear();

}

// Check if humidity exceeds the set point

if (humidity > humiditySetPoint) {

activateBuzzer();

lcd.clear();

lcd.print("High Humidity! Alert");

delay(2000);

lcd.clear();

}

}

void activateBuzzer() {

Serial.println("Activating buzzer...");

// Activate the buzzer with a simple tone

tone(buzzerPin, 1000); // Adjust the frequency as needed

delay(1000); // Buzz for 1 second

noTone(buzzerPin); // Turn off the buzzer

}