ALERT CODE

#include <DHT.h>

#define DHTPIN 7 // DHT11 data pin

#define DHTTYPE DHT11 // DHT11 type

#define MQ2\_PIN A0 // MQ-2 analog pin

#define BUZZER\_PIN 8 // Buzzer pin

#define LED\_PIN 9 // LED pin (indicating safe status)

DHT dht(DHTPIN, DHTTYPE);

// Define threshold values

const int THRESHOLD\_GAS = 300; // Example threshold for gas concentration in ppm

const float THRESHOLD\_TEMP = 30.0; // Example threshold for temperature in °C

void setup() {

Serial.begin(9600); // Initialize Serial Monitor

dht.begin(); // Initialize DHT sensor

pinMode(BUZZER\_PIN, OUTPUT); // Set buzzer pin as output

pinMode(LED\_PIN, OUTPUT); // Set LED pin as output

Serial.println("Setup complete");

}

void loop() {

// Get current time in milliseconds since Arduino started

unsigned long currentTime = millis();

// Read temperature and humidity from DHT11

float temperature = dht.readTemperature(); // Read temperature

float humidity = dht.readHumidity(); // Read humidity

// Read gas concentration from MQ-2

int gasLevel = analogRead(MQ2\_PIN); // Read the gas level

// Check if any DHT11 reads failed and handle it

if (isnan(temperature) || isnan(humidity)) {

Serial.println("Error,Failed to read from DHT sensor!,0,0,0"); // Error flag for DHT failure

} else {

// Output the sensor data in CSV format

Serial.print(currentTime);

Serial.print(",");

Serial.print(temperature);

Serial.print(",");

Serial.print(humidity);

Serial.print(",");

Serial.print(gasLevel);

// Check thresholds for gas and temperature to decide ALERT/SAFE

if (gasLevel > THRESHOLD\_GAS || temperature > THRESHOLD\_TEMP) {

// ALERT state: Turn on buzzer, turn off LED

digitalWrite(BUZZER\_PIN, HIGH); // Buzzer ON

digitalWrite(LED\_PIN, LOW); // LED OFF

Serial.println(",ALERT!"); // Send alert flag

} else {

// SAFE state: Turn off buzzer, turn on LED

digitalWrite(BUZZER\_PIN, LOW); // Buzzer OFF

digitalWrite(LED\_PIN, HIGH); // LED ON (safe)

Serial.println(",SAFE"); // Send safe flag

}

}

delay(1000); // Wait 1 second before the next reading

}

ALERT CODE

#include <DHT.h>

#define DHTPIN 7 // DHT11 data pin

#define DHTTYPE DHT11 // DHT11 type

#define MQ2\_PIN A0 // MQ-2 analog pin

#define BUZZER\_PIN 8 // Buzzer pin

#define LED\_PIN 9 // LED pin (indicating safe status)

DHT dht(DHTPIN, DHTTYPE);

// Define threshold values

const int THRESHOLD\_GAS = 1000; // Example threshold for gas concentration in ppm

const float THRESHOLD\_TEMP = 100.0; // Example threshold for temperature in °C

void setup() {

Serial.begin(9600); // Initialize Serial Monitor

dht.begin(); // Initialize DHT sensor

pinMode(BUZZER\_PIN, OUTPUT); // Set buzzer pin as output

pinMode(LED\_PIN, OUTPUT); // Set LED pin as output

Serial.println("Setup complete");

}

void loop() {

// Get current time in milliseconds since Arduino started

unsigned long currentTime = millis();

// Read temperature and humidity from DHT11

float temperature = dht.readTemperature(); // Read temperature

float humidity = dht.readHumidity(); // Read humidity

// Read gas concentration from MQ-2

int gasLevel = analogRead(MQ2\_PIN); // Read the gas level

// Check if any DHT11 reads failed and handle it

if (isnan(temperature) || isnan(humidity)) {

Serial.println("Error,Failed to read from DHT sensor!,0,0,0"); // Error flag for DHT failure

} else {

// Output the sensor data in CSV format

Serial.print(currentTime);

Serial.print(",");

Serial.print(temperature);

Serial.print(",");

Serial.print(humidity);

Serial.print(",");

Serial.print(gasLevel);

// Check thresholds for gas and temperature to decide ALERT/SAFE

if (gasLevel > THRESHOLD\_GAS || temperature > THRESHOLD\_TEMP) {

// ALERT state: Turn on buzzer, turn off LED

digitalWrite(BUZZER\_PIN, HIGH); // Buzzer ON

digitalWrite(LED\_PIN, LOW); // LED OFF

Serial.println(",ALERT!"); // Send alert flag

} else {

// SAFE state: Turn off buzzer, turn on LED

digitalWrite(BUZZER\_PIN, LOW); // Buzzer OFF

digitalWrite(LED\_PIN, HIGH); // LED ON (safe)

Serial.println(",SAFE"); // Send safe flag

}

}

delay(1000); // Wait 1 second before the next reading

}