**Smart greenhouse management**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

#include <WiFi.h>

#include <ThingSpeak.h>

#include <ESP32Servo.h>

#define WIFI\_SSID "Wokwi-GUEST"

#define WIFI\_PASSWORD ""

#define THINGSPEAK\_API\_KEY "GMYWA336MF4YR4BV"

#define THINGSPEAK\_CHANNEL\_ID 2386314

// Pin Definitions

#define LDR\_PIN         34    //LDR Sensor pin connection

#define YELLOW\_LED\_PIN  32    //Yellow LED pin connection

#define RED\_LED\_PIN     17    //Red LED pin connection

#define SERVO\_PIN       23    //Servo pin connection

LiquidCrystal\_I2C LCD = LiquidCrystal\_I2C(0x27, 20, 4);

DHT dht(4, DHT22);

Servo servo;

const long updateInterval = 1 \* 1000;  // Updates ThingSpeak every seconds

unsigned long lastUpdate = 0;

WiFiClient client;

void setup() {

**Serial**.begin(115200);

  LCD.init();

  delay(1000);

  LCD.backlight();

  LCD.setCursor(0, 0);

  LCD.print("Initializing...");

  pinMode(RED\_LED\_PIN, OUTPUT);

  pinMode(YELLOW\_LED\_PIN , OUTPUT);

  dht.begin();

  servo.attach(23);  // Attach servo to pin 23

  WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(500);

**Serial**.println("Connecting to WiFi...");

  }

  ThingSpeak.begin(client);

}

void loop() {

  unsigned long currentMillis = millis();

  if (currentMillis - lastUpdate >= updateInterval) {

    float humidity = dht.readHumidity();

    float temperature = dht.readTemperature();

    int ldrValue = analogRead(LDR\_PIN);

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

      delay(2000);

      return;

    }

    LCD.clear();

    LCD.setCursor(0, 0);

    LCD.print("Temperature: ");

    LCD.print(temperature, 1);

    LCD.print("C");

    LCD.setCursor(0, 1);

    LCD.print("Humidity: ");

    LCD.print(humidity, 1);

    LCD.print("%");

    LCD.setCursor(0, 2);

    LCD.print("Lux: ");

    LCD.print(ldrValue, 1);

    LCD.print("%");

**Serial**.print("Temperature: ");

**Serial**.print(temperature);

**Serial**.print("C  Humidity: ");

**Serial**.print(humidity);

**Serial**.println("% Light level:");

**Serial**.print(humidity);

    if (temperature > 35.0) {

      servo.write(180);  // Set servo to 180 degrees

    } else {

      servo.write(0);  // Set servo to 0 degrees

    }

    if (humidity < 20.0 || humidity > 60.0 || temperature < 10) {

     digitalWrite(RED\_LED\_PIN, HIGH); // RED LED on

    } else {

      digitalWrite(RED\_LED\_PIN, LOW); // RED LED off

    }

    if (ldrValue < 1913) {

      digitalWrite(YELLOW\_LED\_PIN, LOW); // LED off

    } else {

      digitalWrite(YELLOW\_LED\_PIN, HIGH); // LED on

    }

    ThingSpeak.writeField(THINGSPEAK\_CHANNEL\_ID, 1, temperature, THINGSPEAK\_API\_KEY);

    ThingSpeak.writeField(THINGSPEAK\_CHANNEL\_ID, 2, humidity, THINGSPEAK\_API\_KEY);

    ThingSpeak.writeField(THINGSPEAK\_CHANNEL\_ID, 3, ldrValue, THINGSPEAK\_API\_KEY);

    lastUpdate = 200;

  }

}