**IOT Assignment 3- By P.Pooja Sree**

**19R11A04H8**

Develop a Smart Home Automation project using ESP32.

The tasks involved in completing this project are:

1. Get the Temperature, Humidity from the DHT11 sensor
2. Get the light intensity from LDR
3. Display the light intensity, Temperature, Humidity values on the OLED display.
4. control the lights based on Light intensity ( Control led's as an indication of light)
5. Control the fans based on the temperature and humidity parameters ( Control led's as an indication of fan)

1) Get the Temperature, Humidity from the DHT11 sensor

**code:-**

#include "DHT.h"

#define DHTPIN 4// Digital pin connected to the DHT sensor

// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --

// Pin 15 can work but DHT must be disconnected during program upload.

// Uncomment whatever type you're using!

#define DHTTYPE DHT11 // DHT 11

//#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

//#define DHTTYPE DHT21 // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V

// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1

// to 3.3V instead of 5V!

// Connect pin 2 of the sensor to whatever your DHTPIN is

// Connect pin 3 (on the right) of the sensor to GROUND (if your sensor has 3 pins)

// Connect pin 4 (on the right) of the sensor to GROUND and leave the pin 3 EMPTY (if your sensor has 4 pins)

// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor.

// Note that older versions of this library took an optional third parameter to

// tweak the timings for faster processors. This parameter is no longer needed

// as the current DHT reading algorithm adjusts itself to work on faster procs.

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(9600);

Serial.println(F("DHTxx test!"));

dht.begin();

}

void loop() {

// Wait a few seconds between measurements.

delay(2000);

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht.readHumidity();

// Read temperature as Celsius (the default)

float t = dht.readTemperature();

// Read temperature as Fahrenheit (isFahrenheit = true)

float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t) || isnan(f)) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

// Compute heat index in Fahrenheit (the default)

float hif = dht.computeHeatIndex(f, h);

// Compute heat index in Celsius (isFahreheit = false)

float hic = dht.computeHeatIndex(t, h, false);

Serial.print(F("Humidity: "));

Serial.println(h);

Serial.print(F("% Temperature: "));

Serial.println(t);

Serial.print(F("°C "));

Serial.println(f);

Serial.print(F("°F Heat index: "));

Serial.println(hic);

Serial.print(F("°C "));

Serial.print(hif);

Serial.println(F("°F"));

delay(2000);

}

2) Get the light intensity from LDR

**code:-**

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

int ldrval=analogRead(15);

Serial.print("the LDR value is:");

Serial.println(ldrval);

delay(1000);

}

**3)** Display the light intensity, Temperature, Humidity values on the OLED display

**code:-**

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include "DHT.h"

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 64 // OLED display height, in pixels

#define DHTPIN 4

#define DHTTYPE DHT11

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(115200);

delay(2000);

Serial.println("oled test");

if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {

Serial.println("SSD1306 allocation failed");

Serial.println(F("DHTxx test!"));

dht.begin();

for(;;);

}

delay(2000);

display.clearDisplay();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(0, 10);

// Display static text

}

void loop() {

int a=analogRead(15);

display.print("ldr value is");

display.println(a);

Serial.print("ldr value is");

Serial.println(a);

delay(5000);

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht.readHumidity();

// Read temperature as Celsius (the default)

float t = dht.readTemperature();

// Read temperature as Fahrenheit (isFahrenheit = true)

float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t) || isnan(f)) {

display.println(F("Failed to read from DHT sensor!"));

return;

}

// Compute heat index in Fahrenheit (the default)

float hif = dht.computeHeatIndex(f, h);

// Compute heat index in Celsius (isFahreheit = false)

float hic = dht.computeHeatIndex(t, h, false);

Serial.print(F("Humidity: "));

Serial.println(h);

display.print (F("Humidity: "));

display.print(h);

Serial.print(F("% Temperature: "));

Serial.println(t);

display.print(F("% Temperature: "));

display.print(t);

Serial.print(F("°C "));

Serial.println(f);

display.print(F("°C "));

display.print(f);

Serial.print(F("°F Heat index: "));

Serial.println(hic);

display.print(F("°F Heat index: "));

display.print(hic);

Serial.print(F("°C "));

Serial.print(hif);

display.print(F("°C "));

display.print(hif);

Serial.println(F("°F"));

display.print(F("°F"));

display.display();

delay(2000);

}

**4)** Control the lights based on Light intensity

**Code :-**

void setup() {

// put your setup code here, to run once:

pinMode(4,OUTPUT);

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

int ldrval=analogRead(15);

//int a=digitalRead(4);

Serial.print("the LDR value is:");

Serial.println(ldrval);

delay(1000);

if(ldrval<=2000)

{

digitalWrite(4,LOW);

Serial.println("led is off");

delay(1000);

}

else

{

digitalWrite(4,HIGH);

Serial.println("led is on");

delay(1000);

}}

**Serial monitor statements:-**

the LDR value is:4095

led is on

the LDR value is:4095

led is on

the LDR value is:2019

led is on

the LDR value is:707

led is off

the LDR value is:941

led is off

**5)** Control the fans based on the temperature and humidity parameters

**Code:-**

// Example testing sketch for various DHT humidity/temperature sensors

// Written by ladyada, public domain

// REQUIRES the following Arduino libraries:

// - DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library

// - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit\_Sensor

#include "DHT.h"

#define DHTPIN 4 // Digital pin connected to the DHT sensor

// Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --

// Pin 15 can work but DHT must be disconnected during program upload.

// Uncomment whatever type you're using!

//#define DHTTYPE DHT11 // DHT 11

#define DHTTYPE DHT11 // DHT 22 (AM2302), AM2321

//#define DHTTYPE DHT21 // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V

// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1

// to 3.3V instead of 5V!

// Connect pin 2 of the sensor to whatever your DHTPIN is

// Connect pin 3 (on the right) of the sensor to GROUND (if your sensor has 3 pins)

// Connect pin 4 (on the right) of the sensor to GROUND and leave the pin 3 EMPTY (if your sensor has 4 pins)

// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor.

// Note that older versions of this library took an optional third parameter to

// tweak the timings for faster processors. This parameter is no longer needed

// as the current DHT reading algorithm adjusts itself to work on faster procs.

DHT dht(DHTPIN, DHTTYPE);

void setup() {

Serial.begin(9600);

Serial.println(F("DHTxx test!"));

pinMode(2,OUTPUT);

dht.begin();

}

void loop() {

// Wait a few seconds between measurements.

delay(2000);

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht.readHumidity();

// Read temperature as Celsius (the default)

float t = dht.readTemperature();

// Read temperature as Fahrenheit (isFahrenheit = true)

float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t) || isnan(f)) {

Serial.println(F("Failed to read from DHT sensor!"));

return;

}

// Compute heat index in Fahrenheit (the default)

float hif = dht.computeHeatIndex(f, h);

// Compute heat index in Celsius (isFahreheit = false)

float hic = dht.computeHeatIndex(t, h, false);

Serial.print(F("Humidity: "));

Serial.println(h);

Serial.print(F("% Temperature: "));

Serial.println(t);

Serial.print(F("°C "));

Serial.println(f);

Serial.print(F("°F Heat index: "));

Serial.println(hic);

Serial.print(F("°C "));

Serial.print(hif);

Serial.println(F("°F"));

delay(2000);

if((h<=56)&&(t<32.0))

{

digitalWrite(2,LOW);

delay(2000);

Serial.println("fan is off");

}

else

{

digitalWrite(2,HIGH);

delay(2000);

Serial.println("fan is on");

}

}

**Serial monitor statements:-**

fan is on

Humidity: 56.00

% Temperature: 32.20

°C 89.96

°F Heat index: 36.34

°C 97.42°F

fan is on

Humidity: 56.00

% Temperature: 32.10

°C 89.78

°F Heat index: 36.13

°C 97.03°F