

INFORME PRÀCTICA 5_2+WEB

CODI

```
#include "WiFi.h"
#include "ESPAsyncWebServer.h"
#include <Arduino.h>
#include <Wire.h>
#include "ClosedCube_SHT31D.h"
#include "AsyncTCP.h"
#include <DNSServer.h>
#include "SSD1306Wire.h"

// Replace with your network credentials
const char* ssid = "Xiaomi_11T_Pro";
const char* password = "f5cbd8a82232";

DNSServer dnsServer;
SSD1306Wire display(0x3c, SDA, SCL);
ClosedCube_SHT31D sht31 = ClosedCube_SHT31D();
SHT31D result;

// Create AsyncWebServer object on port 80
AsyncWebServer server(80);

String readSHT31Temperature(){
float t = result.t;
return String(t);
}

String readSHT31Humidity() {

float h = result.rh;
return String(h);
}

const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE HTML><html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4p0pQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
crossorigin="anonymous">
<style>
    html {
        font-family: Arial;
        display: inline-block;
        margin: 0px auto;
        text-align: center;
```

```
        }
    h2 { font-size: 3.0rem; }
    p { font-size: 3.0rem; }
    .units { font-size: 1.2rem; }
    .sht31-labels{
        font-size: 1.5rem;
        vertical-align:middle;
        padding-bottom: 15px;
    }
</style>
</head>
<body>
<h2>ESP32 SHT31 Web Server</h2>
<p>
    <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>
    <span class="sht31-labels">Temperature</span>
    <span id="temperature">%TEMPERATURE%</span>
    <sup class="units">&deg;C</sup>
</p>
<p>
    <i class="fas fa-tint" style="color:#00add6;"></i>
    <span class="sht31-labels">Humidity</span>
    <span id="humidity">%HUMIDITY%</span>
    <sup class="units">%</sup>
</p>
</body>
<script>
setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("temperature").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "/temperature", true);
    xhttp.send();
}, 10000 ) ;

setInterval(function ( ) {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("humidity").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "/humidity", true);
    xhttp.send();
}, 10000 ) ;
</script>
</html>)rawliteral";

// Replaces placeholder with sht31 values
String processor(const String& var){
    /Serial.println(var);
```

```
if(var == "TEMPERATURE"){
    return readSHT31Temperature();
}
else if(var == "HUMIDITY"){
    return readSHT31Humidity();
}
return String();
}

void setup(){
// Serial port for debugging purposes
Serial.begin(115200);

Wire.begin();

// Initialising the UI will init the display too.

Serial.println("ClosedCube SHT3X-D Periodic Mode Example");
Serial.println("supports SHT30-D, SHT31-D and SHT35-D");

sht31.begin(0x44); // I2C address: 0x44 or 0x45

Serial.print("Serial #");
Serial.println(sht31.readSerialNumber());

if (sht31.periodicStart(SHT3XD_REPEATABILITY_HIGH, SHT3XD_FREQUENCY_10HZ)
!= SHT3XD_NO_ERROR)
    Serial.println("[ERROR] Cannot start periodic mode");

display.init();

display.flipScreenVertically();
display.setFont(ArialMT_Plain_10);
// Connect to Wi-Fi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println(".");
    Serial.println(WiFi.status());
}

// Print ESP32 Local IP Address
Serial.println(WiFi.localIP());

// Route for root / web page
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/html", index_html, processor);
});
server.on("/temperature", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", readSHT31Temperature().c_str());
});
server.on("/humidity", HTTP_GET, [](AsyncWebServerRequest *request){
    request->send_P(200, "text/plain", readSHT31Humidity().c_str());
});
```

```
// Start server
Serial.println("Server begin");
server.begin();
Serial.println("End server");
}

void loop(){
result = sht31.periodicFetchData();
// draw the current demo method
float humd = result.rh;
float temp = result.t;

Serial.print("Time:");
Serial.print(millis());
Serial.print(" Temperature:");
Serial.print(temp, 1);
Serial.print("C");
Serial.print(" Humidity:");
Serial.print(humd, 1);
Serial.print("%");

Serial.println();

// clear the display
// draw the current demo method
//demos[demoMode]();
display.clear();

display.setTextAlignment(TEXT_ALIGN_CENTER);
display.setFont(ArialMT_Plain_10);
display.drawString(128/2, 0, "HUMEDAD");
display.setFont(ArialMT_Plain_16);
display.drawString(128/2, 11, String(humd)+"%");
display.setFont(ArialMT_Plain_10);
display.drawString(128/2, 30, "TEMPERATURA");
display.setFont(ArialMT_Plain_16);
display.drawString(128/2, 41, String(temp)+"°C");

display.setFont(ArialMT_Plain_10);
display.setTextAlignment(TEXT_ALIGN_RIGHT);
display.drawString(128, 54, String(millis()/3600000)+String(":")\
+String((millis()/6000)%60)+String(":")\
+String((millis()/1000)%(60)));

// write the buffer to the display
display.display();

delay(100);
}
```

FUNCIONAMENT

Aquesta part és exactament igual que la pràctica 5_2 però li hem afegit una pàgina web per on treu les dades obtingudes del sensor de temperatura. Per això veiem abans del setup() l'estructura de la nostra pàgina web, amb el seu estil de lletra, tamany, alineació...

Abans però, hem d'afegir totes les dades del web server com per exemple el wifi on ens connectem per tal de que ens pugui donar el ip de la pàgina web (*ssid* i el *password*). Per connectar-nos al wifi ho farem amb la funció *WiFi.begin(ssid, password)* i un cop s'ha inicialitzat ens connectem amb el *while(WiFi.status() != WL_CONNECTED){...}* d'aquesta manera no sortirà del bucle fins que es connecti. Un cop connectats, ens donarà el ip de la nostre pàgina web amb la funció *Serial.println(WiFi.localIP())*

FOTO WEB

