UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

Anul II, 2024- 2025

Sistem IoT pentru Monitorizarea și Reglarea Temperaturii

Molnar Andrei Alexandru

Anul II, Facultatea de Automatică si Calculatoare,

Secția Automatică și Informatică Aplicată , seria A, grupa 3

Coordonator: Radu Munteanu

CONTACT: alexmolnar31@gmail.com

0722411665

1 Cuprins

2 Lista de abrevieri și simboluri

ESP32 - Modul microcontroller cu Wi-Fi şi Bluetooth integrat, utilizat pentru aplicaţii IoT.

DHT - Digital Humidity and Temperature, denumire comună pentru senzorii de temperatură şi umiditate (ex: DHT22).

GPIO - General Purpose Input/Output, pini utilizaţi pentru conectarea şi controlul componentelor externe.

HTML - HyperText Markup Language, limbaj folosit pentru structura şi designul interfeţei web.

CSS - Cascading Style Sheets, limbaj pentru stilizarea interfeţei web.

HTTP - HyperText Transfer Protocol, protocol utilizat pentru transferul de date pe web.

LED - Light Emitting Diode, componentă utilizată pentru semnalizare vizuală.

IoT - Internet of Things, reţea de dispozitive conectate care comunică şi partajează date.

SPIFFS - SPI Flash File System, sistem de fişiere utilizat pe ESP32 pentru stocarea datelor.

EEPROM - Electrically Erasable Programmable Read-Only Memory, memorie utilizată pentru stocarea permanentă a datelor.

Wi-Fi - Wireless Fidelity, tehnologie de reţea pentru conectarea dispozitivelor fără fir.

MQTT - Message Queuing Telemetry Transport, protocol pentru comunicarea între dispozitive IoT (menţionat în sugestii).

JSON - JavaScript Object Notation, format utilizat pentru transferul de date structurate între server şi client.

3 Introducere

3.1 Justificarea abordării teme

Tema acestui proiect este justificată de importanţa tot mai mare a monitorizării condiţiilor de mediu în viaţa de zi cu zi. Temperaturile şi nivelurile de umiditate influenţează direct confortul, sănătatea şi funcţionarea corectă a echipamentelor sensibile. Într-o lume tot mai conectată, soluţiile IoT precum acest proiect sunt accesibile şi uşor de implementat, oferind utilizatorilor posibilitatea de a monitoriza şi răspunde rapid la condiţii nefavorabile. Alegerea unui sistem simplu, bazat pe un afişaj digital şi o interfaţă web intuitivă, subliniază dorinţa de a crea un instrument practic, uşor de utilizat de către oricine, dar şi flexibil, adaptabil pentru nevoi mai complexe. Tema se aliniază tendinţelor actuale în automatizare şi tehnologie, fiind relevantă atât pentru uz personal, cât şi educaţional.

3.2 Importanța și aplicabilitatea temei

Proiectul este important pentru că răspunde unei nevoi practice: monitorizarea temperaturii şi umidităţii într-un mod simplu şi accesibil. Poate fi folosit acasă pentru a asigura un mediu confortabil, în birouri pentru a menţine un spaţiu sănătos de lucru, sau în zone tehnice, cum ar fi serverele sau laboratoarele, unde condiţiile de mediu sunt critice. Utilizarea unui afişaj digital îl face uşor de utilizat de oricine, iar interfaţa web oferă flexibilitate şi accesibilitate, astfel încât datele să fie mereu la îndemână. Este un proiect care îmbină funcţionalitatea practică cu uşurinţa în utilizare şi poate fi o bază pentru soluţii mai complexe, fie în aplicaţii personale, fie în scopuri educaţionale.

3.3 Originalitatea lucrării

Acest proiect urmăreşte monitorizarea temperaturii şi umidităţii folosind un senzor DHT22, completat de un sistem de afişare pe un ecran digital OLED, care afişează valorile măsurate în timp real. Datele sunt afişate pe o interfaţă web simplă şi intuitivă, care permite utilizatorului să vizualizeze în timp real valorile senzorului.

Originalitatea proiectului vine din integrarea componentelor într-un mod personalizat şi adaptarea funcţionalităţilor pentru nevoi specifice. De exemplu, pragul de temperatură poate fi ajustat direct din interfaţa web, iar actualizarea datelor se face automat, fără a necesita intervenţia utilizatorului. În loc de soluţii complexe, cum ar fi un sistem de notificare extern, s-a ales o metodă directă de afişare pe un ecran OLED, subliniind astfel caracterul practic şi accesibil al proiectului.

Proiectul se remarcă şi prin aplicabilitatea sa, fiind ideal pentru monitorizarea condiţiilor din locuinţe, birouri sau alte spaţii unde controlul temperaturii este important. Prin combinaţia dintre simplitatea implementării şi flexibilitatea funcţionalităţilor, proiectul reuşeşte să ofere o soluţie uşor de folosit, dar şi suficient de versatilă pentru a fi extinsă în viitor.

4 Descrierea realizării proiectului

4.1 Principiul functional

Proiectul funcţionează pe baza integrării unui senzor de temperatură şi umiditate (DHT22), a unui microcontroler ESP32 şi a unui afişaj digital OLED, completate de o interfaţă web intuitivă. Iată o descriere detaliată a modului în care fiecare componentă contribuie la funcţionarea întregului sistem:

1. Colectarea datelor de la senzorul DHT22

- **Cum funcţionează senzorul**:
  Senzorul DHT22 măsoară atât temperatura, cât şi umiditatea din mediul înconjurător, utilizând un termistor şi un senzor capacitiv.
    - **Temperatura** este măsurată prin variaţia rezistenţei termistorului, care este direct proporţională cu temperatura ambientală.
    - **Umiditatea** este măsurată prin schimbările capacitive ale unui material care absoarbe umiditatea din aer.
- **Citirea datelor**:
  ESP32 citeşte periodic valorile de temperatură şi umiditate de la senzor prin

intermediul unui pin digital (GPIO 27). Acest proces este gestionat de biblioteca DHT, care interpretează semnalele transmise de senzor.

2. Procesarea datelor pe ESP32

- Datele primite de la senzor sunt validate pentru a verifica dacă sunt corecte. În cazul în care citirile nu sunt valide, sistemul returnează o eroare şi afişează valori implicite ("--").
- După validare, valorile măsurate sunt comparate cu un prag prestabilit de temperatură.
    - Dacă temperatura depăşeşte acest prag, LED-ul conectat la ESP32 se aprinde, semnalizând o situaţie de alertă.

3. Afişarea datelor pe interfaţa web

- **Generarea interfeţei web**:
  ESP32 rulează un server web local pe portul 80. Codul HTML al interfeţei este stocat în memoria sa şi transmis către browserul utilizatorului atunci când acesta accesează adresa IP locală a dispozitivului.
- **Actualizarea datelor**:
  Interfaţa utilizează cereri asincrone (AJAX) pentru a solicita periodic date de la ESP32, fără a reîncărca pagina. Aceste cereri interoghează endpoint-urile /temperature şi /humidity pentru a primi valorile curente ale senzorului.
- **Afişarea pe display OLED-ului**:

4. Configurarea pragului de temperatură

- Interfaţa web include funcţionalităţi care permit utilizatorului să ajusteze pragul de temperatură la care LED-ul de alertă s-ar aprinde.

- Această modificare este trimisă către ESP32 sub formă de parametru printr-o cerere HTTP (/update-threshold). ESP32 actualizează valoarea pragului și o salvează în memoria sa.

5. Funcționalitatea generală

1. ESP32 pornește și se conectează la rețeaua Wi-Fi configurată.
2. Serverul web local este inițializat și începe să accepte cereri de la utilizator.
3. La intervale regulate, ESP32 citește datele de la senzorul DHT22, le validează și le compară cu pragurile setate.
4. Datele valide sunt afișate pe interfața web, atat cat si pe display-ul digital conectat la placa ESP32.
5. Utilizatorul poate interacționa cu sistemul prin ajustarea pragului de temperatură direct din interfața web, actualizările fiind aplicate în timp real.

Rezultatul final

Proiectul funcționează ca un sistem simplu și eficient de monitorizare a temperaturii și umidității, oferind utilizatorului acces la informații în timp real și un mod vizual de a identifica condițiile critice prin LED-ul de alertă. Interfața web intuitivă permite monitorizarea și ajustarea parametrilor fără intervenții complexe, asigurând astfel un echilibru între funcționalitate și ușurință în utilizare.

4.2 Partea de cod:

4.2.1 Importarea bibliotecilor

Mai întâi, importăm bibliotecile necesare. Bibliotecile WiFi, ESPAsyncWebServer și ESPAsyncTCP sunt necesare pentru a construi serverul web. Bibliotecile Adafruit_Sensor și DHT sunt necesare pentru a citi datele de la senzorul DHT22.

```
#include "WiFi.h"
#include "ESPAsyncWebServer.h"
#include <ESPAsyncTCP.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
```

Figure 1

4.2.2 Funcții pentru citirea temperaturii și umidității

Am creat două funcții: una pentru citirea temperaturii (readDHTTemperature()) și alta pentru citirea umidității (readDHTHumidity()).

```
String readDHTTemperature() {
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  //float t = dht.readTemperature(true);
  // Check if any reads failed and exit early (to try again).
  if (isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return "--";
  }
  else {
    Serial.println(t);
    return String(t);
  }
}
```

Figure 2

4.3 Procesor

Acum, trebuie să creăm funcția **processor()**, care va înlocui substituenții din textul HTML cu valorile reale de temperatură și umiditate.

```
String processor(const String& var){
  //Serial.println(var);
  if(var == "TEMPERATURE"){
    return readDHTTemperature();
  }
  else if(var == "HUMIDITY"){
    return readDHTHumidity();
  }
  return String();
}
```

Figure 3

4.4 Partea de cod HTML:

4.4.1 Interfata la baza:

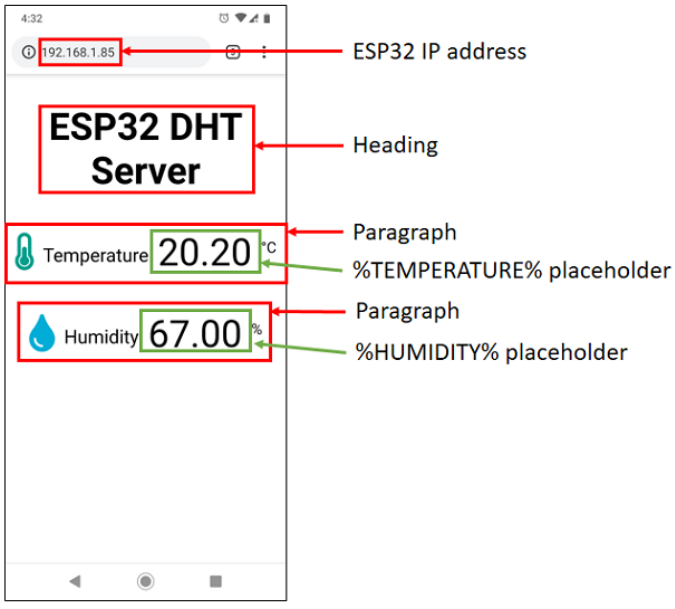


Figure 4

Aşa cum se poate observa în figura de mai sus, pagina web afişează un titlu şi două paragrafe. Un paragraf este folosit pentru a afişa temperatura, iar altul pentru a afişa umiditatea. De asemenea, există două pictograme pentru a stiliza pagina.

4.4.2 Codul HTML

Tot textul HTML, inclusiv stilurile, este stocat în variabila **index_html**. Acum vom analiza textul HTML şi vom vedea ce face fiecare parte.

Următorul tag **<meta>** face ca pagina ta web să fie receptivă (responsive) în orice browser.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Figure 5

Următoarea linie scrie cuvântul „Temperature" pe pagina web.

```
<span class="dht-labels">Temperature</span>
```

Figure 6

Textul **TEMPERATURE** dintre semnele % este un substituent (placeholder) pentru valoarea temperaturii.

```
<span id="temperature">%TEMPERATURE%</span>
```

Figure 7

Aceasta înseamnă că textul **%TEMPERATURE%** este ca o variabilă care va fi înlocuită cu valoarea reală a temperaturii preluată de senzorul DHT. Substituenţii din textul HTML trebuie să fie plasaţi între semnele **%**.

În cele din urmă, adăugăm simbolul pentru grade.

```
<sup class="units">°C</sup>
```

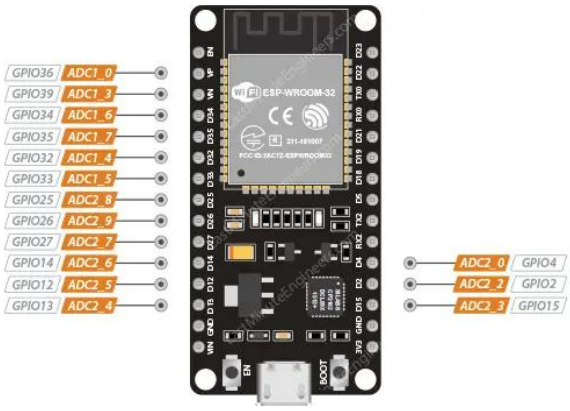
Figure 8

Iara poi vom proceda la fel pentru “UMIDITATE”.

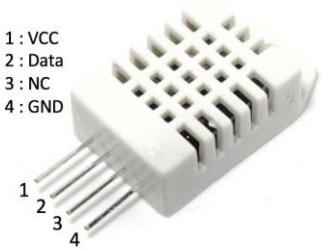
4.5 Schema electrică

Schema circuitului este următoarea:

- ESP32: Alimentat prin portul USB la 5V.



- DHT22:



- Pinul VCC conectat la 3.3V al plăcii ESP32.
- Pinul GND conectat la GND al plăcii ESP32.
- Pinul de date conectat la GPIO 27 al plăcii ESP32, cu un rezistor de pull-up de 4.7kΩ conectat între VCC şi pinul de date.

- Display Digital

Această configuraţie asigură stabilitatea semnalului de ieşire şi o funcţionare optimă a senzorului dupa cum este prezentat in figura 9 de mai jos. Iar in figura 10 este o versiune mai detaliata a acesteia.

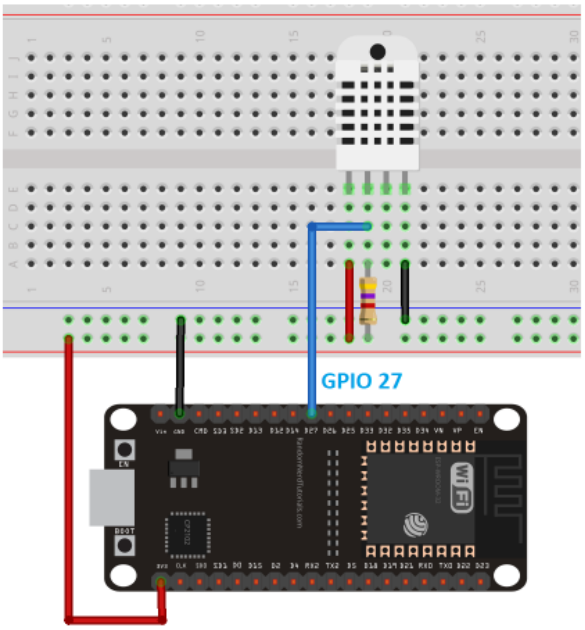


Figure 9

4.6 Componente

4.6.1 ESP32-WROOM-32 Pinout

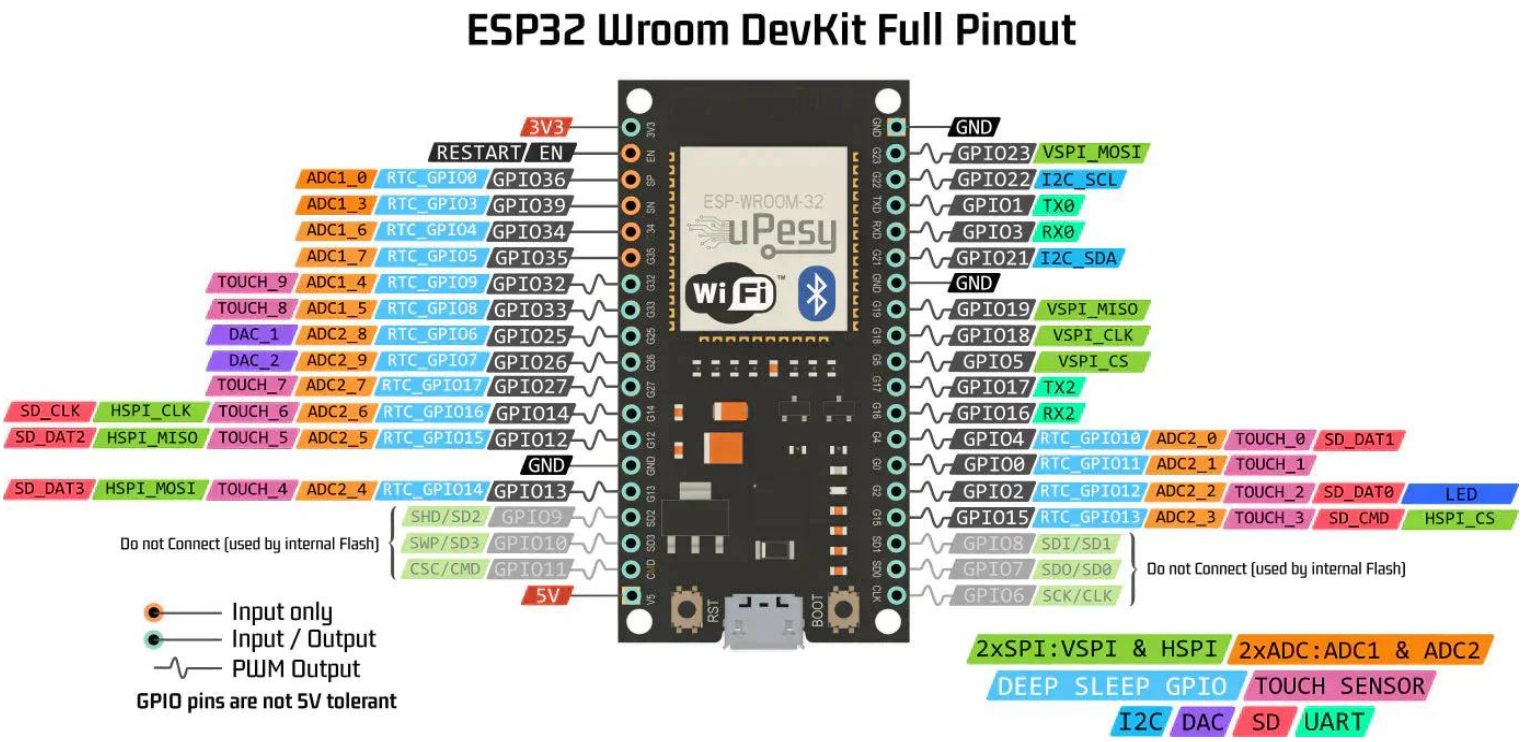


Figure 10

Tabel Specificaţii ESP32-WROOM-32

Categorii	Itemi	Specificaţii
Certificări	RF	See certificates for [64]
	Wi-Fi	Wi-Fi Alliance
	Bluetooth	BQB
	Certificare Verde	RoHS/REACH
Teste	Fiabilitate	HTOL/HTSL/uHAST/TCT/ESD
Wi-Fi	Protocoale	802.11 b/g/n (802.11n până la 150 Mbps)
		Agregare A-MPDU şi A-MSDU şi interval de gardă de 0,4 µs
	Interval de frecvenţă centrală de funcţionare a canalului	2412 ~ 2484 MHz
Bluetooth	Protocoale	Bluetooth v4.2 BR/EDR şi specificația Bluetooth LE
	Radio	Receptor NZIF cu sensibilitate de -97 dBm
		Transmiţător clasa-1, clasa-2 și clasa-3
		AFH
	Audio	CVSD şi SBC
	Interfețe modul	Card SD, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR, contor de impulsuri, GPIO, senzor tactil capacitiv, ADC, DAC, Interfaţă auto cu două fire (TWAI®), compatibilă cu

Hardware		ISO11898-1 (Specificaţia CAN 2.0)
	Cristal integrat	Cristal de 40 MHz
	Memorie flash SPI integrată	4 MB
	Tensiune de operare/alimentare	3.0V - 3.6V
	Curent de operare	Medie: 80 mA
	Curent minim furnizat de sursa de alimentare	500 mA
	Interval de temperatură ambientală recomandat	-40°C până la +85°C
	Dimensiunea pachetului	18 mm x 25.5 mm x 3.10 mm
	Nivel de sensibilitate la umiditate (MSL)	Nivel 3

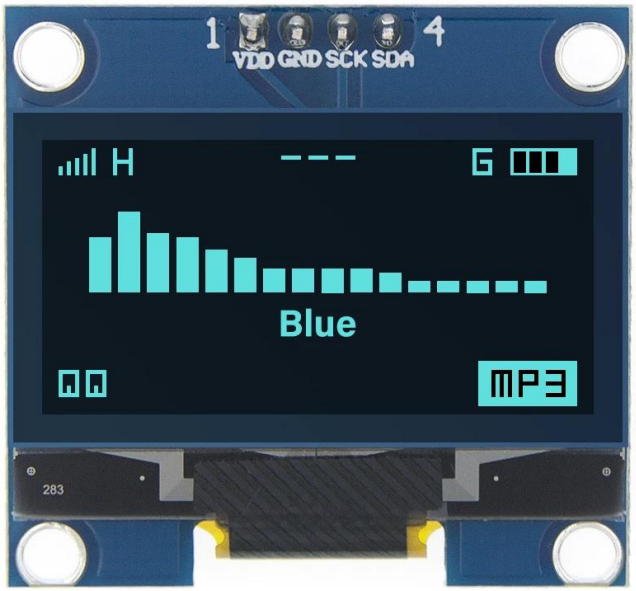
4.6.2 Senzor DHT22

Figure 11

Tabel specificaţii senzor DHT22

Tensiune de alimentare	3V-5V
Intensitate	2.5 mA
Procentaj de umiditate	0-100%
Temperaturi funcţionale	-40 °C + 80 °C
Dimensiuni	27 mm x 59 mm x 13.5 mm

4.6.3 Display OLED 1.3”



y

Figure 12

Tabel specificaţii display OLED 1.3”

Culoare	Albastru
Rezoluţie	128x64
Comunicare serială	I2C
Consum	0.08W

Tensiune alimentare	DC 3V-5V
Temperatură de funcţionare	-30°~70°
Dimensiuni	36x31x4.1 mm
Driver IC	SH1106
Dimensiuni display	34.5x19

4.7 Schema Completa

4.7.1 Schema Electrica

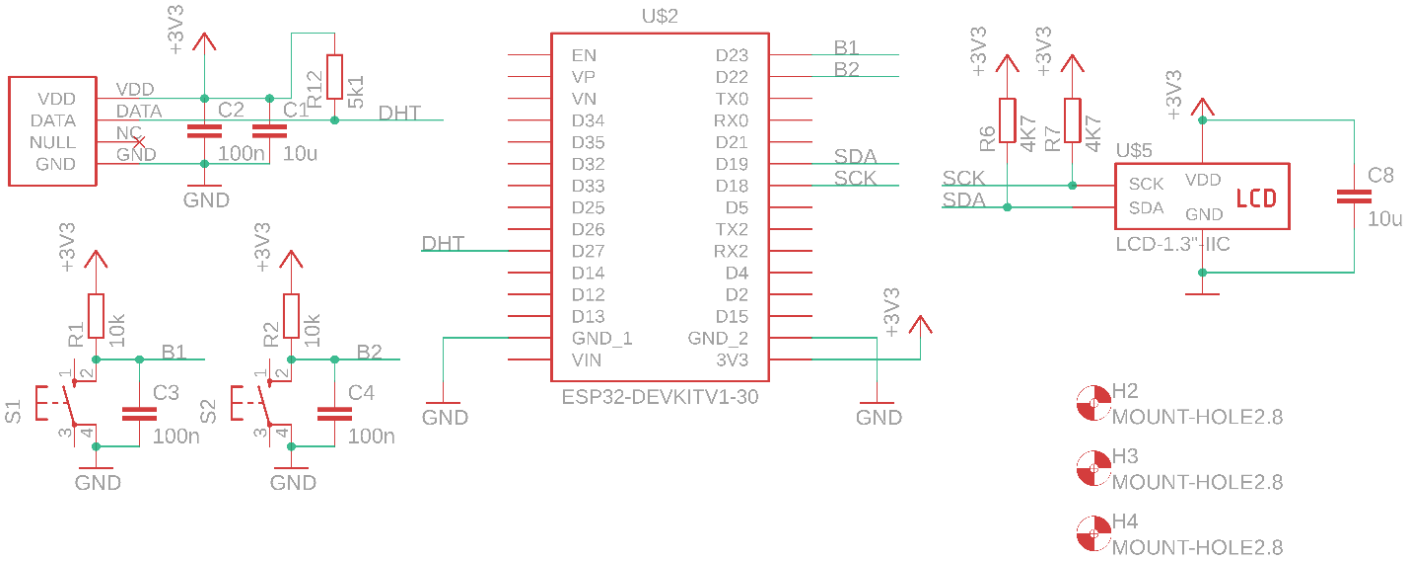


Figure 13

4.7.2 Vedere inferioara a placutei(PCB)

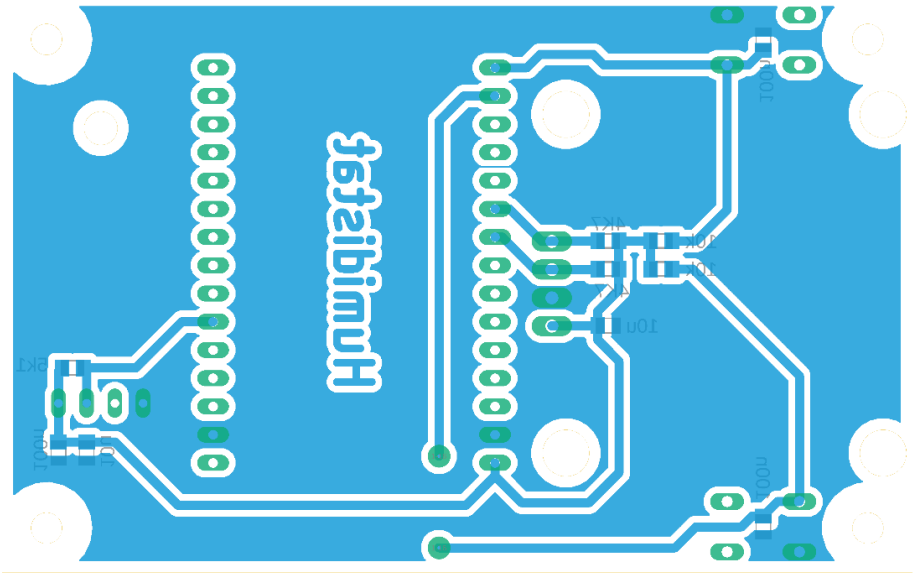


Figure 14

4.7.3 Vedere superioara a placutei

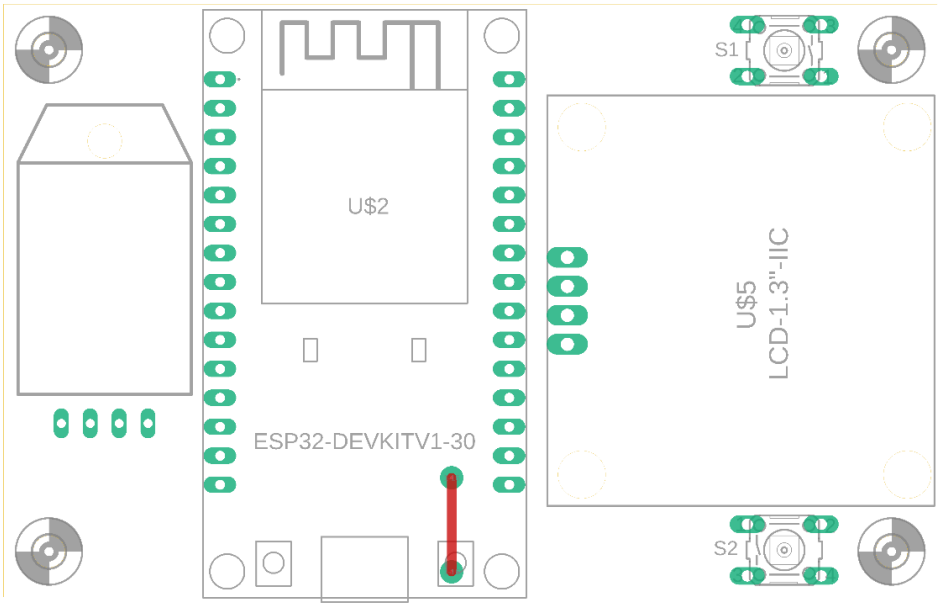


Figure 15

4.8 Carcasa Circuitului

4.8.1 Capac partea din fata

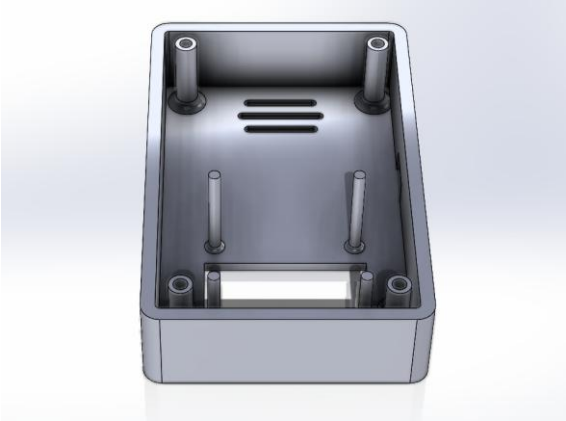


Figure 16

4.8.2 Capac partea din spate

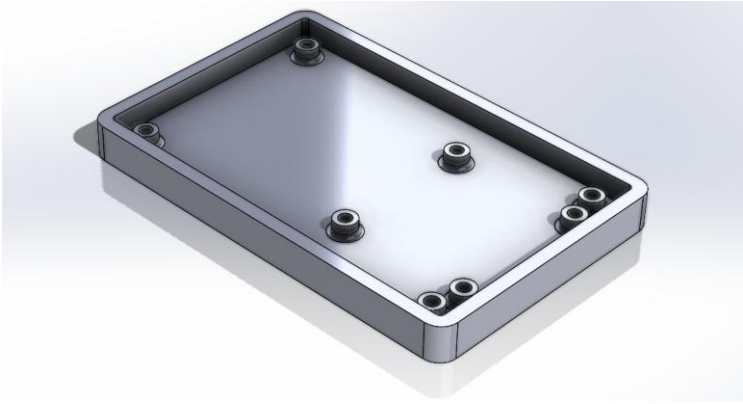


Figure 17

4.8.3 Extensia de buton

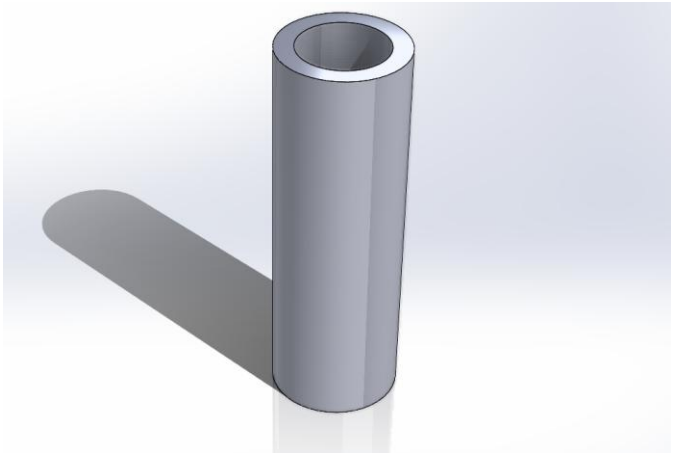


Figure 18

4.8.4 Ansamblu carcasa complet

Figure 19

Figure 20

4.9 Schema explicate

Senzorul DHT22 este alimentat de către microcontroler la +3V3, pinul de date (DATA) este conectat la ESP32 la pinul D27, iar pinul GND este conectat la masa PCB-ului. Rezistența pull-up R12 (5k1) este pentru linia de date a senzorului, conține și condensatorii C1 (10uF) și C2 (100uF) pentru filtrarea tensiunii de alimentare. Pinul NULL nu face nimic, nu a fost conectat la nimic pe plăcuță.

Display-ul OLED are pinul de alimentare este conectat la +3V3, pinul SCK (linia de ceas pentru I2C) este conectat la D18 de pe microcontroler, pinul SDA (linia de date pentru I2C) este conectat la pinul D19 de pe microcontroler, iar pinul GND este conectat la masa PCB-ului. Condensatorul C8 (10uF) este folosit pentru filtrarea alimentarii display-ului, rezistențele pull-up R6 și R7 (amândouă de 4k7) pentru magistrala I2C (SCK și SDA).

H2, H3, H4 sunt găurile de montare de 2.8 mm pentru fixarea fizică a plăcii PCB pe cadrul dispozitivului, având aceeași problemă cu H1. O să apară și restul de

găuri pe vederea inferioară a termostatului pentru prinderea display-ului şi a senzorului DHT22 (fiind de acelaşi diametru).

Butoanele S1 şi S2 sunt conectate la pinii de intrare digitala B1 (D23) şi B2 (D22) a microcontrolerului. Condensatorii C3 şi C4 (100uF) sunt folosiţi pentru debouncing (folosit pentru a nu provoca erori în funcţionalitatea circuitelor şi a microcontrolerelor), iar R1 şi R2 (amândouă de 10k) sunt rezistenţe pull-up pentru butoane.

Microcontrolerul ESP32 în afară de pinii conectaţi la display, senzorul de umiditate şi cele două butoane mai are pinii GND conectaţi la masa PCB-ului şi pinul de 3V3 conectat la alimentarea PCB-ului.

Funcţionalitatea termostatului începe cu alimentarea circuitului de la 3.3V (de la microcontrolerul ESP32) unde C1, C2 şi C8 sunt utilizaţi pentru filtrarea şi stabilizarea tensiunii de alimentare mai puţin condensatorii C3 şi C4 care sunt utilizaţi pentru debouncing, fiecare pin de date care este conectat la microcontroler conţine cate o rezistenţă pull-up pentru o protecţie în plus. Butoanele sunt adăugate pentru a fii programate pentru diferite comezi.

Vederile PCB-ului superioare şi inferioare vor face vizibile conectarea componentelor şi cum au fost concepute cât mai eficient dar şi dimensiunile plăcuţei care sunt de 82 mm x 52 mm, centrul găurilor este la 4 mm poziționat de marginile plăcuţei.

5 Probleme și soluții

1. **Problemă:**
Senzorul poate returna valori NaN sau citiri inexacte, mai ales în medii cu interferenţe electrice sau în cazul unei alimentări instabile.

Soluţii:

- Adaugarea unui condensator de decuplare (ex: 0.1µF) între pinii VCC şi GND ai senzorului pentru a reduce zgomotul.
- Verificarea calităţii conexiunilor şi utilizarea firelor de bună calitate.
- Implementarea unei filtrări software care să elimine citirile aberante prin medierea mai multor valori consecutive.

2. **Problemă:**
ESP32 poate pierde conexiunea la reţea din cauza interferenţelor, a distanţei mari faţă de router sau a unor probleme de configurare.

Soluţii:

- Implementarea unei funcţii de reconectare automată în cazul pierderii conexiunii:
- Alegerea unei reţele Wi-Fi cu un semnal puternic sau utilizarea unui amplificator de semnal.
- Utilizarea unei adrese IP statice pentru ESP32 pentru a reduce timpul de reconectare.

3. **Problemă:**
Interfaţa poate afişa valori eronate sau neactualizate dacă cererile HTTP nu funcţionează corect sau dacă există probleme de sincronizare.

Soluţii:

- Verificarea endpoint-urilor din codul ESP32 şi a logicii JavaScript din pagina HTML.
- Testarea interfaţei cu un browser diferit pentru a elimina eventualele probleme legate de cache.
- Creşterea frecvenţei cererilor HTTP doar dacă este necesar, dar fără a suprasolicita ESP32.

4. **Problemă:**
Dacă mai mulţi utilizatori accesează interfaţa simultan, ESP32 poate întâmpina dificultăţi în gestionarea cererilor.

Soluţii:

- Limitarea numărului de conexiuni simultane la serverul web.
- Utilizarea unui broker MQTT pentru a gestiona traficul în locul cererilor HTTP individuale.

6 Observații privind rezultatele obținute

Rezultatele obţinute arată că proiectul funcţionează bine şi îşi îndeplineşte scopul principal. Senzorul DHT22 a furnizat date precise despre temperatură şi umiditate, iar LED-ul de alertă a răspuns corect, aprinzându-se atunci când temperatura a depăşit pragul stabilit. Totodată, interfaţa web s-a comportat excelent, afişând informaţiile în timp real şi permiţând utilizatorului să ajusteze pragul de temperatură cu uşurinţă.

Conexiunea Wi-Fi a fost stabilă pe toată durata testelor, iar sistemul a reuşit să se reconecteze rapid în caz de întreruperi. Designul simplu al interfeţei a fost intuitiv, oferind acces rapid la informaţii esenţiale, cum ar fi starea LED-ului sau valorile măsurate. În plus, sistemul a fost testat în condiţii critice şi a răspuns prompt, ceea ce îl face de încredere pentru aplicaţii practice.

Din punct de vedere al resurselor, ESP32 a gestionat eficient atât cererile de date, cât şi controlul LED-ului, fără să apară întârzieri sau blocaje. Proiectul s-a dovedit uşor de implementat şi de utilizat, fiind o soluţie accesibilă pentru monitorizarea condiţiilor de mediu. Este, de asemenea, uşor de extins, oferind posibilitatea de a adăuga funcţionalităţi noi, cum ar fi notificările prin e-mail sau integrarea cu alte platforme IoT.

7 Concluzii

7.1 Concluzii generale

Proiectul acesta reprezintă o soluţie simplă, dar eficientă, pentru monitorizarea temperaturii şi umidităţii, oferind un sistem de afişare clar şi intuitiv prin intermediul unui display digital OLED, alături de o interfaţă web uşor de utilizat. Deşi se bazează pe componente accesibile şi tehnici bine cunoscute, modul în care acestea sunt integrate într-un sistem funcţional subliniază versatilitatea şi aplicabilitatea soluţiilor IoT în viaţa de zi cu zi.

Rezultatele obţinute arată că sistemul este stabil, precis şi uşor de extins. Afişajul OLED permite utilizatorului să monitorizeze condiţiile de mediu în timp real, fără a necesita acces permanent la interfaţa web, ceea ce sporeşte confortul şi accesibilitatea. Faptul că poate fi personalizat şi extins îl face util atât pentru utilizare personală, cât şi pentru aplicaţii educaţionale.

Concluzia generală este că proiectul demonstrează cum tehnologii accesibile, precum ESP32, senzorul DHT22 şi afişajul OLED, pot fi utilizate pentru a rezolva probleme reale într-un mod eficient şi accesibil. Cu câteva ajustări şi extinderi, sistemul poate fi adaptat pentru multe alte scopuri, de la monitorizarea serelor până la integrarea în ecosisteme IoT mai complexe. Este un exemplu de cum un proiect mic poate avea un impact practic mare.

7.2 Concluzii privind aspectele economice

Din punct de vedere economic, proiectul este extrem de accesibil. Utilizarea componentelor de bază, precum ESP32, senzorul DHT22 şi afişajul OLED, asigură un cost redus, fără a compromite funcţionalitatea. Spre exemplu, costurile totale pentru realizarea proiectului pot varia între 50 şi 100 RON, în funcţie de sursa componentelor şi de accesorii.

De asemenea, sistemul are un consum energetic minim, ceea ce îl face potrivit pentru utilizare continuă. Afişajul OLED, în special, oferă un consum redus de energie, contribuind la eficienţa generală a sistemului. Această eficienţă economică, combinată cu beneficiile practice, îl recomandă pentru utilizare personală sau în proiecte educaţionale. Comparativ cu soluţii comerciale mai complexe, proiectul oferă o alternativă economică ce poate fi personalizată pentru a răspunde nevoilor utilizatorului.

Aceste concluzii subliniază faptul că lucrarea nu doar că adresează o problemă tehnică, dar o face într-un mod eficient din punct de vedere al costurilor şi al resurselor, aducând astfel valoare adăugată în domeniul aplicaţiilor IoT.

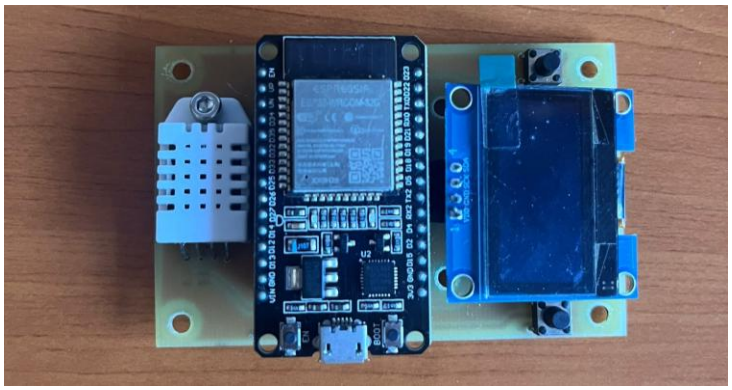


Figure 21

Figure 22

8 Bibliografie

1. Adafruit, „DHT Sensor Library Documentation,” https://github.com/adafruit/DHT-sensor-library.
2. Arduino, „Arduino Reference Guide,” https://www.arduino.cc/reference/en/.
3. Espressif Systems, „ESP-IDF Programming Guide,” https://docs.espressif.com/.
4. G3Schools (HTML și CSS) https://www.w3schools.com
5. GitHub (ESPAsyncWebServer) https://github.com/me-no-dev/ESPAsyncWebServer
6. Hackster.io https://www.hackster.io
7. Random Nerd Tutorials, „ESP32 DHT Server Web Tutorial,” https://randomnerdtutorials.com/.
8. **YouTube** (**Tech with Tim**, **GreatScott**, **DroneBot Workshop**)