Proiect Priză inteligentă cu 3 ieșiri independente

P0. Definirea ideii (direcției) de dezvoltare

Proiectul propune realizarea unei **prize inteligente cu trei ieșiri independente**, fiecare controlabilă individual prin intermediul unui microcontroller conectat la rețea (ex. ESP32). Scopul este de a automatiza alimentarea dispozitivelor electrice, oferind funcția de control prin aplicație mobilă sau interfață web. Acesta este un use-case pentru un sistem de decizie pentru controlul parametrilor monitorizati in incinte urbane, fiind vizat pentru utilizarea în locuințe pentru a putea controla prizele de la distanță.

P1. Identificarea resurselor necesare

Informații: principii de funcționare a releelor, circuite de comutație, protecții la tensiune rețea, programare microcontroller (Arduino/ESP-IDF).

Cerințe hardware:

- Microcontroller ESP32
- 3 Module de Relee (≥16A, 250VAC, cu izolare galvanica)
- Sursă 230V→5V DC izolată
- Trei prize de ieșire și conectori
- Carcasă izolantă + siguranță fuzibilă + borne de conectare
- Cablu de rețea, conectori, cablaj PCB sau breadboard

Cerințe software: IDE Arduino sau PlatformIO, biblioteci: WiFi.h, WebServer.h / MQTT client, cod pentru controlul releelor și interfața de comandă

P2. Stabilirea modelului de dezvoltare

Soluție fizică: se va construi un prototip fizic al prizei inteligente folosind un ESP32 și un modulele de relee montate într-o cutie izolată. Fiecare releu controlează o priză separată.
Soluție software: aplicație web pe server local pe ESP32 pentru control Wi-Fi.
Simulare / testare: înainte de conectarea la tensiunea rețea, sistemul se va testa pe tensiune joasă (5V DC) cu LED-uri în loc de sarcini.
Date colectate: starea prizelor (ON/OFF), tensiunea, comenzi de la utilizator.
Rezultat final: sistem funcțional demonstrativ, capabil să comute independent trei ieșiri prin comenzi wireless, cu izolare electrică și protecție de bază.

P3. Analiza SWOT

Puncte Forte (Strengths)

Control Independent: Oferă control separat pentru fiecare din cele trei ieșiri, sporind flexibilitatea în automatizarea dispozitivelor.

Tehnologie Standardizată: Utilizează un microcontroller popular (ESP32) și medii de dezvoltare comune (IDE Arduino/PlatformIO), facilitând implementarea și documentarea.

Control la Distanță: Funcția de bază de control prin aplicație mobilă sau interfață web răspunde cerinței de bază pentru un dispozitiv inteligent.

Siguranță Prioritară: Include cerințe clare pentru izolare galvanică a releelor, sursă izolată, carcasă izolantă și siguranță fuzibilă, indicând o preocupare pentru protecție.

Strategie de Testare Schedulată: Planificarea testării inițiale la tensiune joasă (5V DC) înainte de conectarea la rețea reduce riscul de deteriorare a componentelor și crește siguranța.

Puncte Slabe (Weaknesses)

Interfață Software Simplă: Soluția software se bazează pe o aplicație web pe server local pe ESP32, care ar putea oferi o experiență de utilizare limitată comparativ cu o aplicație mobilă dedicată sau un serviciu cloud.

Complexitatea Manipulării Tensiunii Rețea: Lucrul cu 230V AC necesită prudență și cunoștințe avansate de protecție electrică, reprezentând un risc tehnic și de siguranță pentru prototip.

Oportunități (Opportunities)

Integrare cu Sisteme Smart Home: Proiectul se încadrează în piața în creștere a dispozitivelor IoT, fiind un use-case pentru controlul parametrilor în incinte urbane. Poate fi integrat cu platforme precum Home Assistant, Google Home sau Alexa.

Dezvoltarea Funcțiilor Avansate: Pot fi adăugate funcții de programare orară, scenarii de automatizare și monitorizare a consumului de energie, transformând-o într-o soluție completă de management energetic.

Optimizarea Carcasei/Designului: Posibilitatea de a dezvolta un design de carcasă elegant, compact și ușor de utilizat, care să se integreze estetic în locuințe.

Sursă de Date pentru Decizii: Datele colectate (starea prizelor, comenzi) pot servi ca bază pentru un sistem de decizie mai complex privind controlul parametrilor în locuință.

Amenințări (Threats)

Reglementări de Siguranță Electrică: Nerespectarea standardelor stricte de siguranță electrică pentru tensiunea de rețea (230V AC) poate duce la defecte, incendii sau riscuri pentru utilizatori.

Concurență de Piață: Există deja o multitudine de prize inteligente comerciale, iar produsul final trebuie să ofere un avantaj competitiv (cost, funcții, fiabilitate).

Defectarea Releelor/Componentelor: Releele trebuie să fie durabile și să reziste la curenții maximi (≥16A) pe termen lung; o subdimensionare sau o calitate slabă ar fi un eșec critic.

Risc de Securitate Cibernetică: Expunerea interfeței web a ESP32 direct pe internet prin port forwarding face dispozitivul vulnerabil la atacuri externe. Fără măsuri de securitate solide (ex. autentificare puternică, HTTPS/TLS), datele sau controlul dispozitivului pot fi compromise.

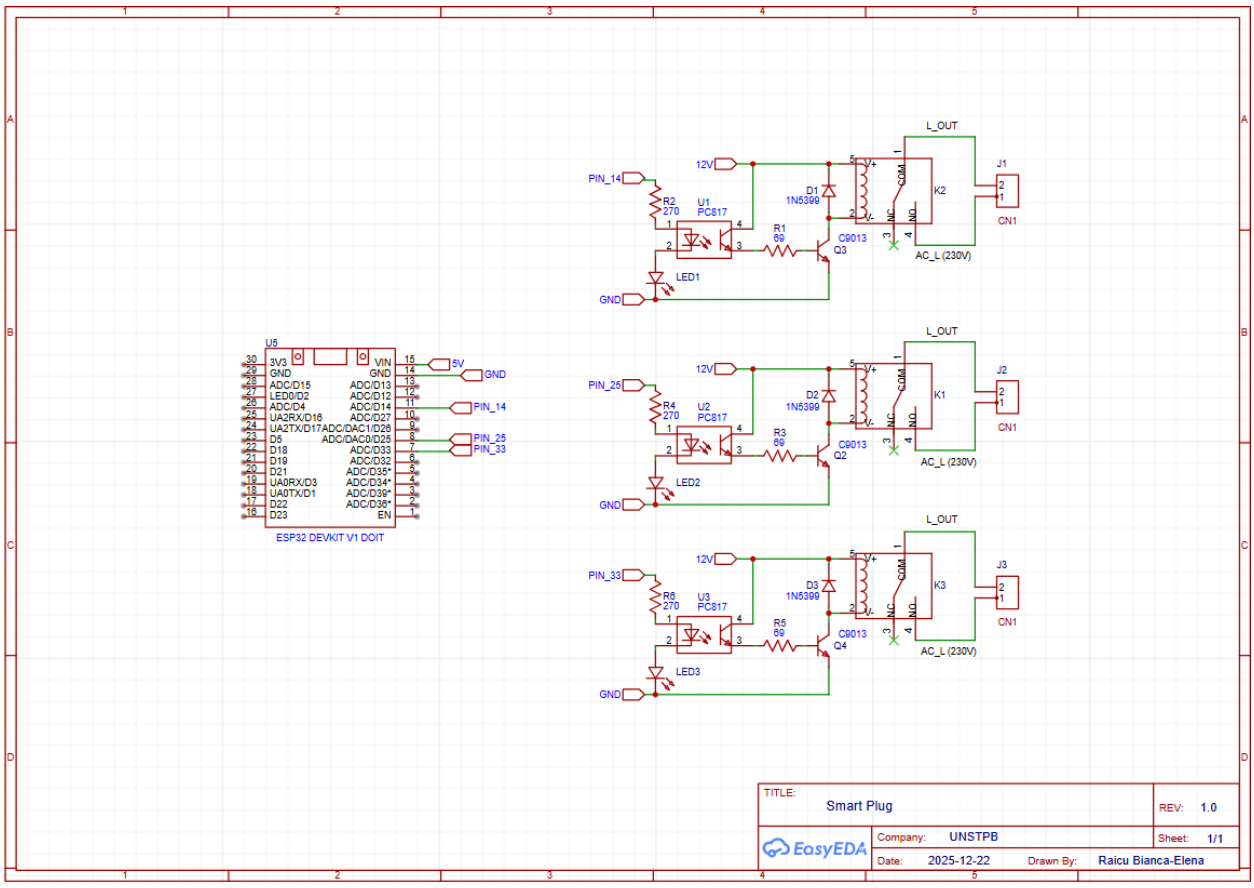
P4. Proiectarea soluției

Proiectarea acestui circuit a fost realizată având ca obiectiv principal siguranța în exploatare și izolarea galvanică între partea de comandă (microcontrolerul ESP32) și partea de putere (sarcina de 230V AC). Circuitul este structurat pe trei canale independente, fiecare fiind controlat de un pin digital al ESP32 (PIN_14, PIN_25, PIN_33).

Pentru protecția microcontrolerului împotriva eventualelor vârfuri de tensiune sau perturbații din rețeaua electrică, am implementat o etapă de izolare folosind optocuploare PC817. Acestea permit transmiterea comenzii prin intermediul luminii, eliminând orice contact electric direct între logica de 3.3V și restul circuitului. În serie cu dioda internă a optocuplerului și LED-ul de stare, am calculat și montat rezistențe de 270 Ω pentru a limita curentul la o valoare optimă de funcționare.

Etapa de execuție folosește tranzistori NPN (C9013) pe post de comutatoare electronice, care amplifică curentul necesar pentru a acționa bobinele releelor de 12V. Fiecare releu este protejat de o diodă de tip flyback (1N5399) montată în paralel cu bobina, având rolul de a disipa energia stocată în câmpul magnetic în momentul decuplării, prevenind astfel distrugerea tranzistorului.

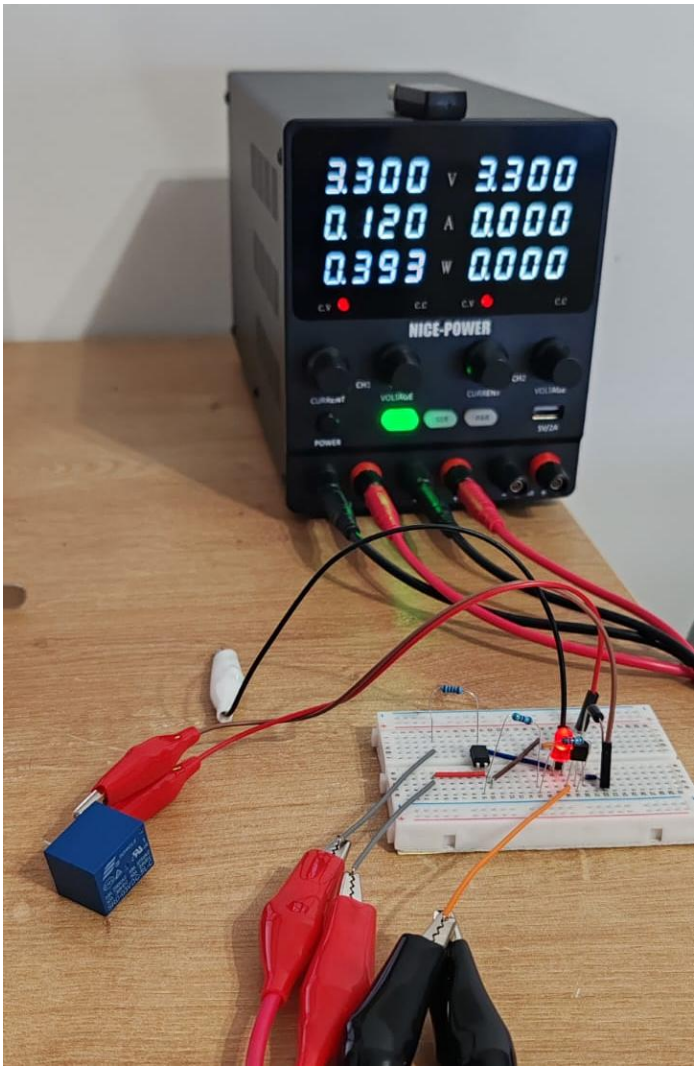
Conectarea la rețeaua de curent alternativ a fost proiectată prin rutarea fazei (AC LINE) prin pinii COM (Common) și NO (Normally Open) ai releelor. Această configurație asigură că priza (LOAD) este alimentată doar atunci când ESP32 trimite un semnal "HIGH" către optocuplor. Terminalele de ieșire (J1, J2, J3) permit o conexiune sigură și modulară pentru consumatorii finali.



P5. Implementare

1. Etapa de Prototipare și Validare pe Breadboard

Această primă etapă a reprezentat fundamentul practic al proiectului, având ca scop principal verificarea integrității componentelor și validarea schemei logice propuse în mediul de simulare. Am utilizat o placă de test (breadboard) pentru a interconecta temporar microcontrolerul ESP32 cu etajul de izolare format din optocuplorul PC817 și tranzistorul de comandă. Pentru alimentarea circuitului și simularea semnalelor, am folosit o sursă de laborator setată la valoarea de 3.3V, specifică logicii ESP32. Această fază a fost critică pentru a observa în timp real comportamentul diodelor de stare (LED-urile) și pentru a măsura curentul absorbit de bobina releului, asigurându-mă că parametrii electrici se încadrează în limitele de siguranță ale componentelor semiconductoare alese.

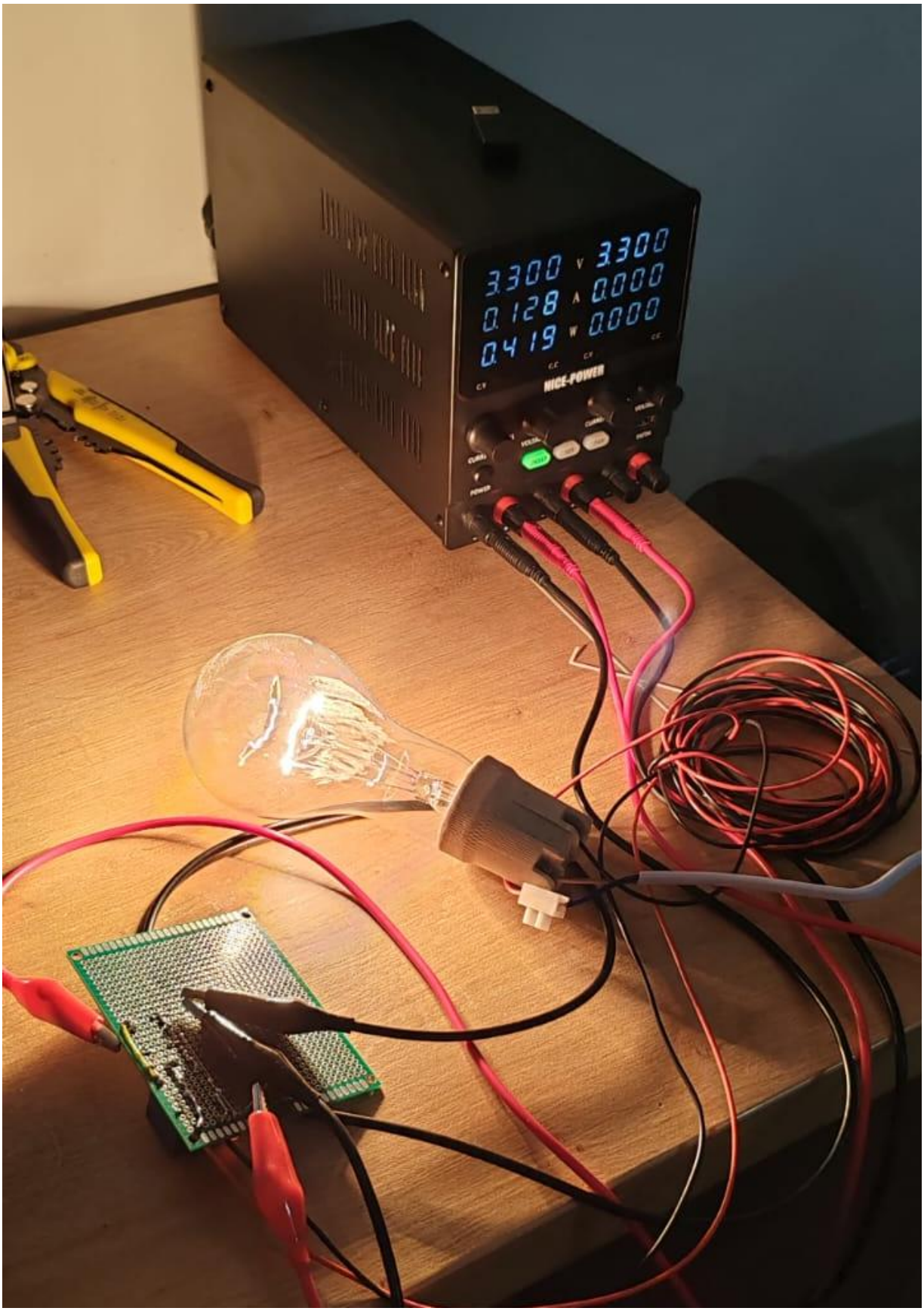


2. Implementarea Hardware pe Placă de Prototipare (Perfboard)

Trecerea de la breadboard la placa de prototipare de tip perfboard a marcat tranzitia către un dispozitiv compact și fiabil. Procesul a implicat o planificare atentă a traseelor pentru a menține o distanță fizică de siguranță între zona de joasă tensiune (controlul de 3.3V și 5V) și zona de înaltă tensiune (traseele pentru 230V AC). În această etapă, am realizat lipiturile definitive pentru releul Songle SRD-03VDC-SL-C și am integrat componentele de protecție, precum dioda flyback 1N5399, care are rolul esențial de a proteja tranzistorul împotriva tensiunilor inverse generate de bobina releului la decuplare. Utilizarea unei plăci cu puncte mi-a permis să obțin un montaj robust, capabil să reziste la solicitări mecanice și să asigure conexiuni electrice durabile, eliminând riscul contactelor imperfecte specifice breadboard-ului. Am adoptat o strategie de asamblare modulară, implementând și testând riguros un singur canal de control înainte de a replica identic circuitul pentru restul canalelor, asigurând astfel validarea fiecărei etape și fiabilitatea întregului montaj final.

3. Testarea în Sarcina și Analiza Performanței Sistemului

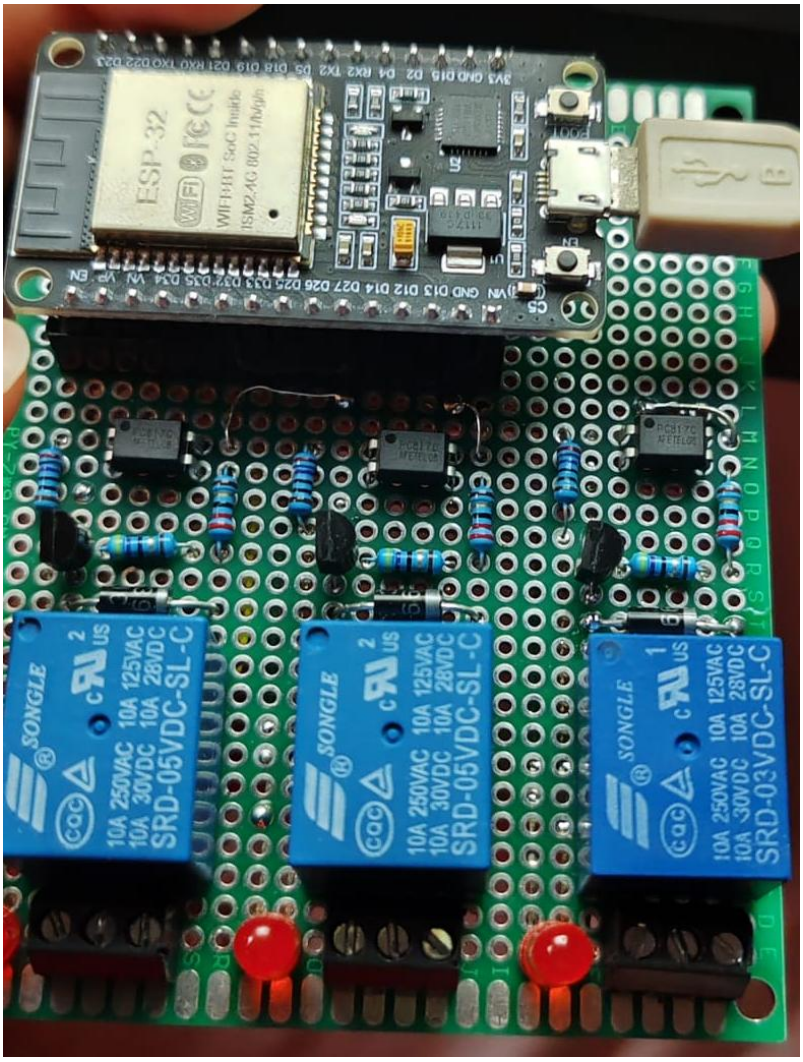
Ultima și cea mai importantă fază a implementării a fost testarea sistemului în condiții de utilizare reală, prin comutarea unei sarcini rezistive reprezentată de un bec incandescent de mare putere. În cadrul acestui test, am monitorizat consumul de energie indicat de sursa de laborator, care a raportat o valoare stabilă de 0.128A și o putere disipată de aproximativ 0.42W în starea activă a releului. Această monitorizare a confirmat faptul că circuitul de comandă proiectat este extrem de eficient și că tranzistorul de comutație nu se supraîncălzește în timpul funcționării prelungite. Succesul acestui test demonstrează că izolarea galvanică prin lumină oferită de optocuplor funcționează impecabil, permițând controlul în siguranță al tensiunii de rețea (230V) fără a pune în pericol microcontrolerul sau utilizatorul.

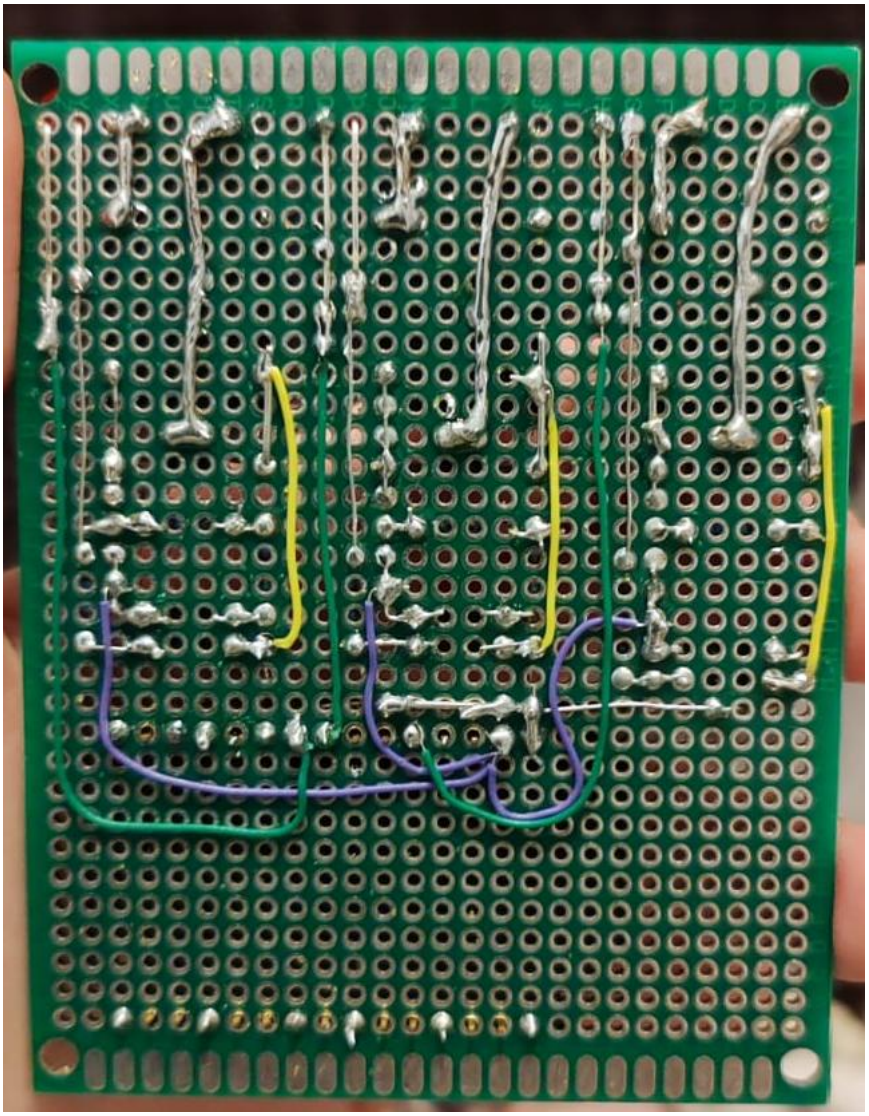

3.300 V 3.300
0.128 A 0.000
0.419 W 0.000
NICE-POWER

4. Asamblarea Unității Centrale de Control

Această etapă a constat în gruparea tuturor componentelor electronice pe placa de prototipare finală, realizând astfel "creierul" sistemului de control. În această fază, am integrat microcontrolerul ESP32 pe socluri dedicate și am finalizat cablajul pentru cele trei canale de control. Se poate observa clar organizarea modulară a circuitului: cele trei optocuploare pentru izolare, tranzistorii de putere și cele trei relee Songle dispuse pentru a facilita accesul la terminalele de ieșire.

Deși partea electronică este acum complet funcțională și testată, acesta reprezintă doar nucleul sistemului. Următorul pas critic în implementare va fi integrarea acestei unități într-o carcasă de protecție și realizarea conexiunilor electrice către prizele externe. Această etapă finală va asigura nu doar un aspect estetic profesional, ci și protecția necesară pentru manipularea în siguranță a tensiunii de 230V AC la care vor fi conectate prizele. Am adaugat și o poză cu lipiturile de pe spate.





5. Programarea Microcontrolerului în Arduino IDE

Partea de inteligență a sistemului a fost dezvoltată în limbajul C++, utilizând un model de comunicare bazat pe protocolul MQTT în locul unui server HTTP clasic, pentru a permite controlul instantaneu de la distanță fără configurări complexe de rețea. Am implementat librăria WiFiManager, care elimină necesitatea stocării datelor Wi-Fi în codul sursă; astfel, la prima pornire sau în absența rețelei, ESP32 generează un portal captive parolat pentru configurarea dinamică a conexiunii. Microcontrolerul gestionează pinii digitali (14, 25, 33) ca ieșiri, utilizând o funcție de callback asincronă care procesează comenzile primite de la interfața Vercel prin intermediul unor topicuri securizate. Această arhitectură asigură o latență minimă și o stabilitate ridicată, transformând dispozitivul într-un nod IoT autentic și portabil.

Am adăugat aici codul pentru logica de control pentru gestionarea comenzilor Remote Toggle: procesarea pachetelor de date sosite pe topicul de comandă și actualizarea în timp real a stării pinilor digitali (GPIO) pentru acționarea prizei.

```
void callback(char* topic, byte* payload, unsigned int length) {
  String message = "";
  for (int i = 0; i < length; i++) message += (char)payload[i];

  Serial.println("Comanda primită: " + message);

  if (message == "1_TOGGLE") {
    state1 = !state1;
    digitalWrite(pin1, state1 ? HIGH : LOW);
    client.publish(topic_status, state1 ? "1:ON" : "1:OFF");
  }
  if (message == "2_TOGGLE") {
    state2 = !state2;
    digitalWrite(pin2, state2 ? HIGH : LOW);
    client.publish(topic_status, state2 ? "2:ON" : "2:OFF");
  }
  if (message == "3_TOGGLE") {
    state3 = !state3;
    digitalWrite(pin3, state3 ? HIGH : LOW);
    client.publish(topic_status, state3 ? "3:ON" : "3:OFF");
  }
}
```

Implementarea completă, incluzând configurațiile de rețea și bibliotecile auxiliare, este disponibilă pentru consultare în repository-ul oficial al proiectului: https://github.com/bia12340/smart_plug.

6. Crearea Interfeței Web (HTML/CSS)

Pentru a oferi o experiență de utilizare intuitivă și sigură, am dezvoltat o aplicație web responsivă utilizând HTML5, CSS3 și JavaScript, găzduită pe platforma Vercel. Interfața adoptă o estetică minimalistă, unde butoanele interactive reflectă în timp real starea releelor prin schimbări cromatice (verde pentru „activ” și gri pentru „inactiv”).

Din punct de vedere tehnic, aplicația nu mai depinde de adresa IP locală a dispozitivului, ci utilizează protocolul WebSockets pentru a comunica cu brokerul MQTT. Pentru protecția accesului, am implementat un strat de securitate bazat pe hashing SHA-256; astfel, parola de acces nu este stocată în format text în codul sursă, ci sub forma unei amprente digitale criptate, verificată asincron la logare. De asemenea, am integrat utilizarea localStorage pentru a menține sesiunea activă pe dispozitivele autorizate, asigurând un echilibru între securitate robustă și ușurință în utilizare.

A. Logica de Securitate (Hashing)

```
async function checkPass() {
    const input = document.getElementById('pass-input').value;

    // Algoritm SHA-256 pentru a cripta parola introdusă
    const msgBuffer = new TextEncoder().encode(input);
    const hashBuffer = await crypto.subtle.digest('SHA-256', msgBuffer);
    const hashArray = Array.from(new Uint8Array(hashBuffer));
    const hashHex = hashArray.map(b => b.toString(16).padStart(2, '0')).join('');
    const parolaCriptata = "3f21e50243a4f1b966e1acf9b60fd0dad7def7c4c5593385e1fca6f3099a4261";

    if(hashHex === parolaCriptata) {
        localStorage.setItem("isLoggedIn", "true");
        document.getElementById('login-screen').style.display = "none";
    } else {
        alert("Cod incorect!");
    }
}
```

B. Conexiunea și Mesajele MQTT (Protocoale)

```
const client = mqtt.connect('wss://broker.hivemq.com:8884/mqtt');
const topicCmd = 'bia_smart_12340/priza/comenzi';
const topicStatus = 'bia_smart_12340/priza/status';

client.on('connect', () => {
    document.getElementById('status').innerText = "Online";
    document.getElementById('status').style.color = "#22c55e";
    client.subscribe(topicStatus);
});
```

C. Sincronizarea UI (Logica asincronă)

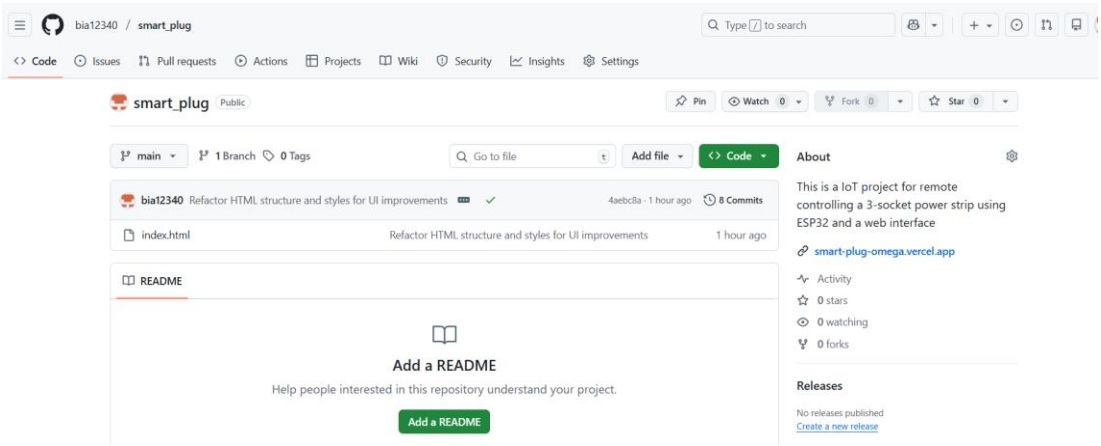
```
client.on('message', (topic, payload) => {
    const msg = payload.toString();
    const [id, state] = msg.split(':');
    updateUI(id, state);
});

function toggle(id) {
    client.publish(topicCmd, id + "_TOGGLE");
}
```

7. Versionarea pe GitHub și Deployment-ul pe Vercel

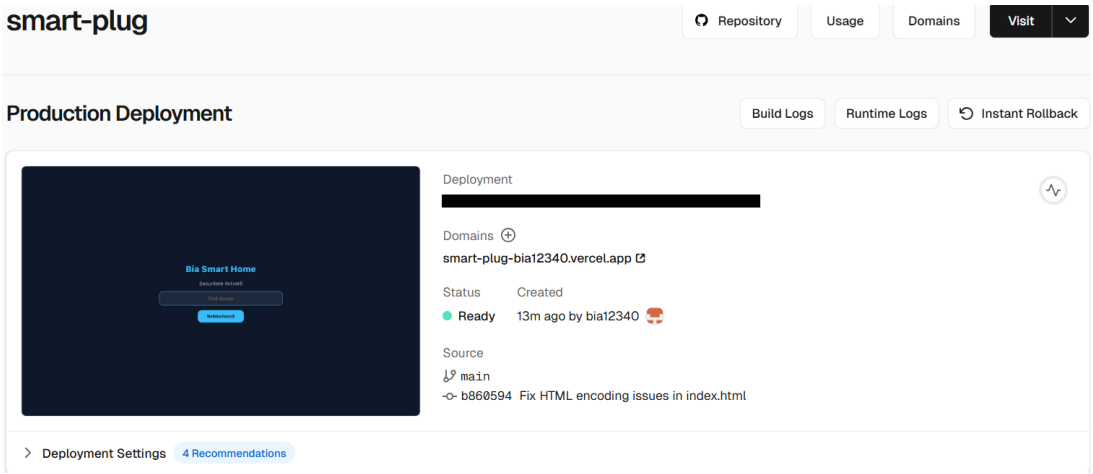
Pentru managementul codului și trasabilitatea versiunilor, am utilizat GitHub, încărcând sursele proiectului într-un repository securizat. Această abordare mi-a permis să automatizez fluxul de lucru prin integrarea cu platforma Vercel, care realizează un deployment automat la fiecare actualizare a codului.

Prin generarea unui link public securizat (HTTPS), am eliminat bariera rețelei locale. Datorită arhitecturii bazate pe WebSockets și brokerul MQTT, interfața găzduită pe Vercel poate comunica instantaneu cu dispozitivul ESP32, indiferent de locația geografică a utilizatorului. Astfel, sistemul beneficiază de o infrastructură stabilă, oferind control global asupra prizelor inteligente de pe orice dispozitiv mobil, fără a compromite securitatea datelor sau viteza de răspuns.

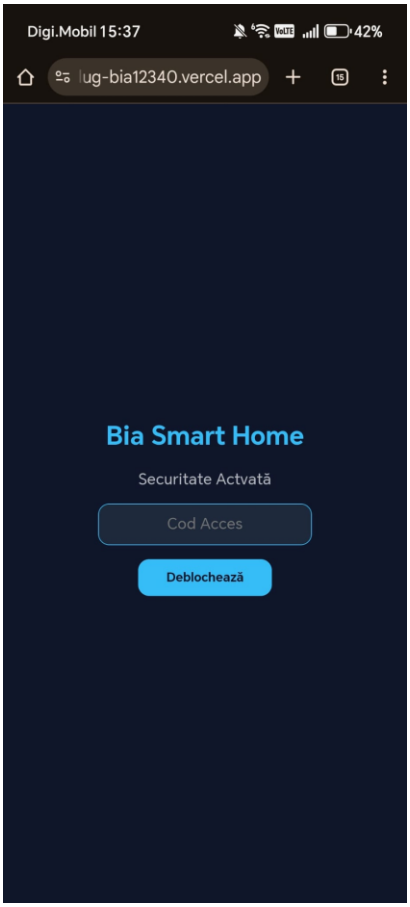
Aici este link-ul de pe Github: https://github.com/bia12340/smart_plug/blob/main/index.html

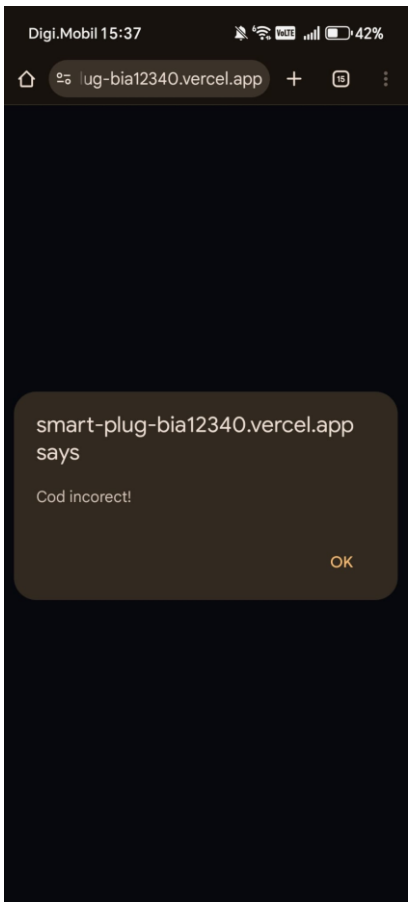


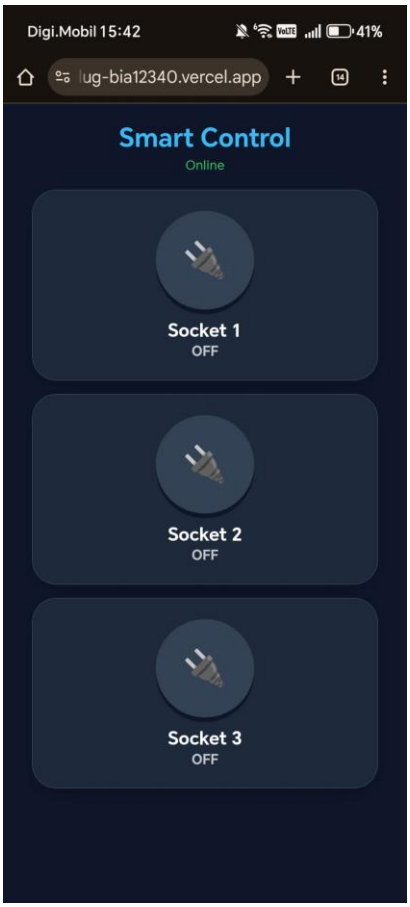
Așa arăta pe platforma Vercel(am cenzurat linkul de deployment în ambele pentru mai multă securitate):



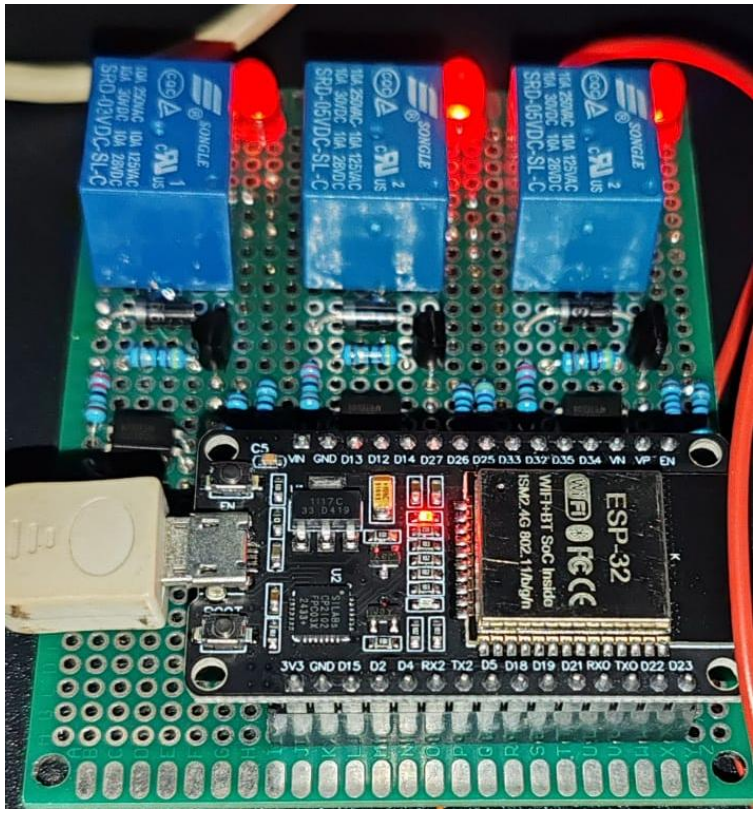
În continuare am adăugat screenshot-uri cu interfața de autentificare, cu mesajul care apare când parola este incorectă, și cu stările activ/pasiv ale canalelor. Am adăugat și o poză cu Esp-ul în care se observă becurile aprinse după ce le-am comutat la activ.


Digi.Mobil 15:37
42%
ug-bia12340.vercel.app
Bia Smart Home
Securitate Actvată
Cod Acces
Deblochează


Digi.Mobil 15:37
42%
ug-bia12340.vercel.app
smart-plug-bia12340.vercel.app says
Cod incorect!
OK

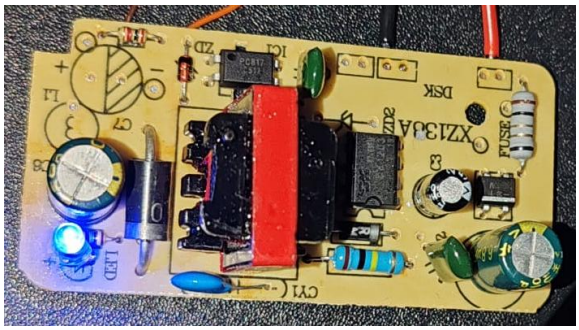

Digi.Mobil 15:42
41%
ug-bia12340.vercel.app
Smart Control
Online
Socket 1
OFF
Socket 2
OFF
Socket 3
OFF

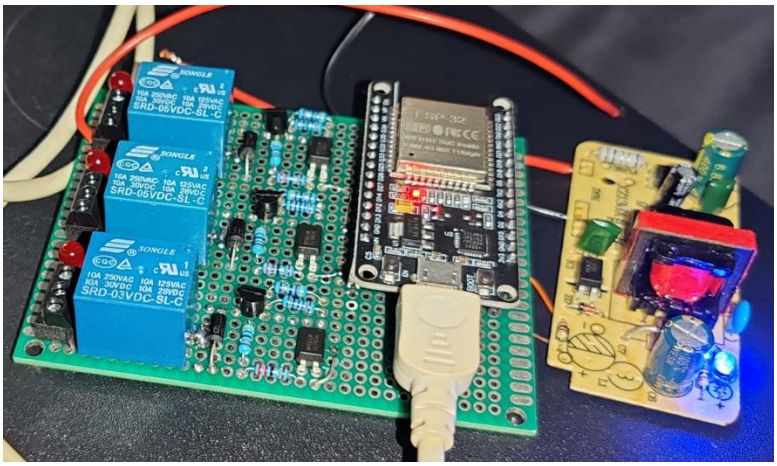

Digi.Mobil 16:32
38%
ug-bia12340.vercel.app
Smart Control
Online
Socket 1
ACTIVE
Socket 2
ACTIVE
Socket 3
ACTIVE


ESP-32
WiFi+BT SoC module
ISM2.4G 802.11b/g/n

8. Sistemul de Alimentare (Sursa AC-DC)

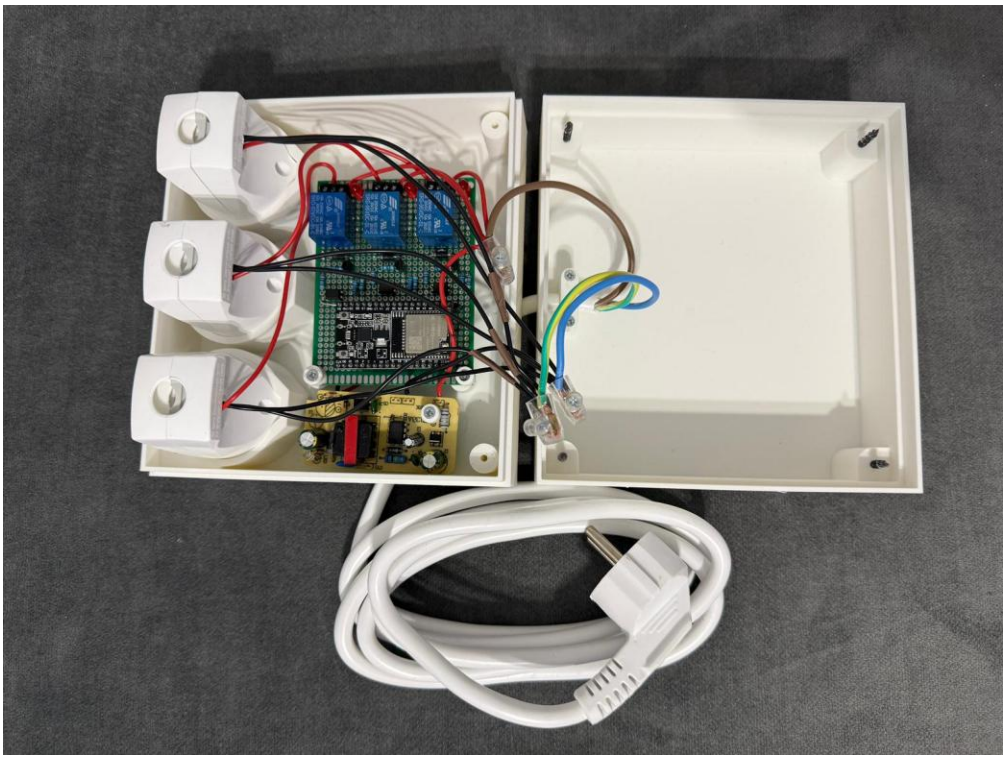
După finalizarea și testarea codului, am integrat în circuit o sursă de alimentare compactă (AC-DC Converter – XZ138A) pentru a transforma curentul alternativ de la priză (230V) în curent continuu de joasă tensiune (5V), necesar microcontrolerului ESP32 și modulelor de relee. Această componentă este vitală deoarece elimină dependența de un cablu USB conectat la calculator, transformând prototipul într-un dispozitiv independent. Sursa include un etaj de filtrare (condensatorii electrolitici vizibili în imagine) și un transformator de izolare, asigurând un curent stabil și protejând componentele logice de fluctuațiile rețelei electrice. Prin adăugarea ei, am finalizat tranziția de la un model alimentat extern la un sistem IoT complet integrat, gata să fie montat într-o carcasă finală.

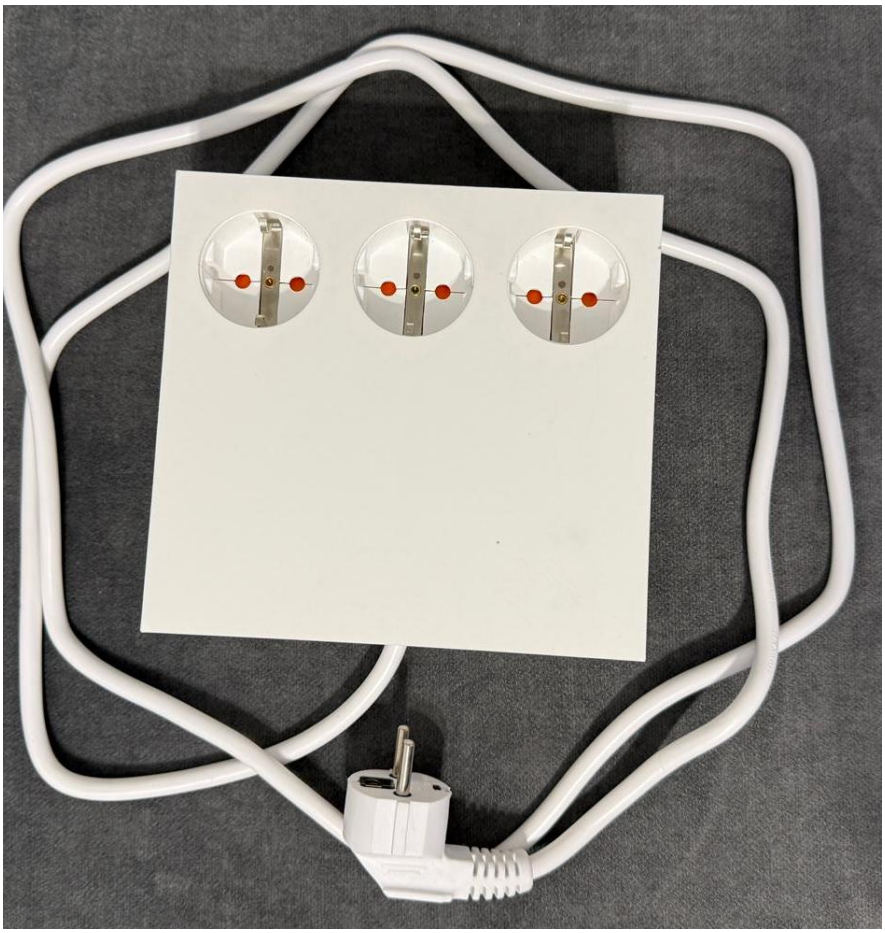




9. Integrarea Hardware și Ansamblul Final

Etapa finală a proiectului a constat în consolidarea circuitului electronic într-o structură compactă și sigură. Componentele logice (ESP32 și sursa XZ138A) au fost integrate împreună cu puntea de relee într-o carcasă proiectată și realizată prin imprimare 3D, special concepută pentru a asigura izolarea electrică și ventilarea corespunzătoare. Conexiunile de putere au fost realizate utilizând conductori de cupru dimensionați pentru sarcini casnice, finalizându-se cu montarea ștecherului de alimentare și a prizelor individuale. Acest ansamblu transformă prototipul dintr-o placă de test într-un dispozitiv IoT gata de utilizare, îmbinând precizia software cu o construcție hardware solidă și estetică.





10. Instrucțiuni de Utilizare și Configurare

Sistemul a fost conceput pentru a fi utilizat intuitiv, procesul de punere în funcțiune fiind împărțit în trei pași simpli:

- Conectarea la rețea: La prima alimentare (sau într-o locație nouă), dispozitivul va încerca timp de 3 secunde să se conecteze la rețeaua salvată. Dacă aceasta nu este disponibilă, priza va genera propriul punct de acces Wi-Fi numit "Configurare_Priza_Smart". Utilizatorul trebuie să se conecteze la această rețea (parola: lumos123) și să introducă datele noii rețele locale în portalul care apare automat.
- Autentificarea: Odată ce dispozitivul este online, se accesează interfața web găzduită pe Vercel. Pentru a activa butoanele de control, utilizatorul trebuie să introducă codul de acces securizat (Lumos3).
- Controlul și Monitorizarea: După logare, starea fiecărei prize poate fi schimbată prin simpla apăsare a butoanelor corespunzătoare. Interfața va afișa în timp real confirmarea comenzii prin schimbarea culorii indicatorilor, confirmând astfel comunicarea bidirecțională între cloud și hardware prin protocolul MQTT.

P6. Bibliografie

https://randomnerdtutorials.com/projects-esp8266/

https://www.w3schools.com/js/

https://www.hivemq.com/mqtt/

https://auth0.com/blog/hashing-passwords-one-way-road-to-security/

https://emn178.github.io/online-tools/sha256.html

https://vercel.com/docs/git

https://www.youtube.com/watch?v=LvweY-6NV6A

https://www.youtube.com/watch?v=VnfX9YJbaU8

https://www.youtube.com/watch?v=DMm20Z7SF4M