

UNIVERSITATEA TEHNICĂ din CLUJ-NAPOCA
FACULTATEA de INGINERIE ELECTRICĂ

**Dezvoltarea unui Sistem Modular dedicat
Managementului în Aplicațiile cu Automatizare**

I. ENUNȚUL TEMEI:

Dezvoltarea unui sistem modular dedicat managementului în aplicațiile cu automatizare bazat pe protocolul ESP-NOW și microcontrolerul ESP32

II. CONȚINUTUL proiectului de diplomă

- a) Piese scrise
- b) Piese desenate
- c) Anexe

III. LOCUL DOCUMENTĂRII:

Universitatea Tehnică din Cluj-Napoca
Facultatea de Inginerie Electrică
Departamentul de Electronică de Putere și Actionări Electrice
Laboratorul de Electronică de Putere

IV. CONDUCĂTOR ȘTIINȚIFIC:

Ș.L.Dr.Ing. Salcu Ionuț Sorin

V. Data emiterii temei: 18.06.2024

VI. Termen de predare: 7.07.2025

Conducător științific,
Ș.L.Dr.Ing. Salcu Ionuț Sorin

Absolvent,
Tudor Popa

Curpins

Curpins	3
1 Rezumat.....	6
2 Introducere	6
2.1 Definirea Problemei	6
2.2 Obiectivele Cercetării.....	7
3 Considerații teoretice.....	8
3.1 Protocolul ESP-NOW	8
3.1.1 Evoluția către Versiunea 2.0	9
3.1.2 Arhitectura Protocolului și Formatele de Frame	9
3.1.3 Capacități Avansate și Aplicații	10
3.1.4 Performanță și Limitări	10
3.1.5 Implementarea și Compatibilitatea.....	11
3.2 Componente Hardware.....	11
3.2.1 Placa de Dezvoltare ESP32	11
3.2.2 Afișaj OLED SSD1306	13
3.3 Arhitectura Rețelei	14
3.4 Tehnologi Web.....	15
3.4.1 WebSocket Protocol.....	15
3.4.2 AsyncWebServer.....	16
3.4.3 HTML5 (HyperText Markup Language)	16
3.4.4 CSS3 (Cascading Style Sheets).....	17
3.4.5 JavaScript	18
4 Evoluția Sistemului - Procesul Iterativ de Dezvoltare	19
4.1 Prototip Initial (V1).....	19
4.1.1 Procesul de realizare.....	19
4.1.2 Provocări întâlnite	20
4.2 Interfața Web (V2)	20
4.2.1 Îmbunătățiri	20
4.2.2 Provocări	20
4.3 Configurare Dinamică (V3)	21
4.3.1 Îmbunătățiri	21
4.3.2 Provocări	22



Dezvoltarea unui Sistem Modular dedicat

Managementului în Aplicațiile cu Automatizare

4.4	Reorganizare (V4).....	22
4.5	Optimizări Finale (V5, V6).....	22
5	Produs Final	23
5.1	Arhitectura Generală a Sistemului	23
5.2	Dispozitivul Master.....	24
5.3	Dispozitivele Slave	25
5.4	Configurare	25
6	Concluzie	26
6.1	Rezumat	26
6.2	Contribuții tehnice.....	27
6.3	Evaluarea performanței	27
6.4	Aplicații practice	28
6.5	Îmbunătățiri posibile	29
6.6	Limitări și considerații	29
6.7	Ultimele cuvinte.....	30
7	Anexe	31
7.1	ANEXA A: Fișiere de cod principal folosite	31
7.2	ANEXA B: Interfața web a primei versiuni.....	31
7.3	ANEXA C: Interfața web a versiunilor doi și trei	32
7.4	ANEXA D: Interfața web finală a modului master	33
7.5	ANEXA E: Interfața web pentru configurarea unui modul	34
8	Bibliografie	35



1 Rezumat

În această lucrare se prezintă proiectarea și implementarea unui sistem de management modular bazat pe protocolul de comunicație ESP-NOW utilizând microcontrolere ESP32. Sistemul folosește o arhitectură master-slave pentru monitorizarea centralizată și în timp real a datelor trimise de cel mult 20 de module diferite. Informația este organizată în structuri de date pentru a menține o eficiență mare în comunicarea modulelor și oferită utilizatorului într-o interfață web interactivă și simplă.

Cuvinte cheie: ESP32, ESP-NOW, Comunicație Wireless, Design Modular

2 Introducere

Sistemele de dispozitive distribuite au revoluționat modul în care modulele inteligente comunică și colaborează pentru a oferi soluții integrate în diverse domenii de aplicare. Sistemele tradiționale se bazează adesea pe conectivitatea WiFi, care poate fi consumatoare de energie și complexă de gestionat în implementări la scară largă cu module diverse și distribuite geografic. Această lucrare prezintă o abordare inovatoare pentru managementul rețelelor modulare folosind protocolul ESP-NOW, care oferă avantaje semnificative în termeni de consum de energie, simplitatea comunicației și robustețea rețelei.

Protocolul ESP-NOW, dezvoltat de Espressif Systems, oferă o metodă de comunicație fără conexiune care permite comunicarea directă dispozitiv-la-dispozitiv fără nevoia unei infrastructuri de rețea tradiționale. Acest lucru îl face deosebit de potrivit pentru rețelele modulare unde diverse tipuri de dispozitive trebuie să comunice eficient, de la simple module de senzori la actuatori complecși, controllere inteligente și dispozitive personalizate pentru aplicații specifice.

2.1 Definirea Problemei

Rețelele actuale de dispozitive distribuite se confruntă cu provocări fundamentale care limitează eficiența și adoptarea pe scară largă a acestor tehnologii în aplicații modulare complexe. Una dintre cele mai semnificative probleme este consumul ridicat de energie, deoarece modulele bazate pe WiFi necesită o putere considerabilă pentru menținerea conexiunii de rețea. Această cerință energetică înaltă devine problematică în special pentru dispozitivele alimentate cu baterii,



modulele autonome sau pentru implementările în locații îndepărtate unde accesul la surse de alimentare constante este limitat.

Complexitatea rețelei reprezintă o altă provocare majoră, întrucât rețelele TCP/IP tradiționale necesită configurare și management complex pentru fiecare tip de modul conectat. Acest aspect devine și mai pronunțat când se dorește implementarea unor rețele mari cu module unde fiecare element trebuie configurat individual și gestionat în cadrul unei infrastructuri de rețea sofisticate. Problemele de scalabilitate apar în mod natural din această complexitate, deoarece rețelele mari cu module eterogene devin din ce în ce mai dificil de gestionat și monitorizat pe măsură ce numărul și diversitatea dispozitivelor crește.

În plus, preocupările privind fiabilitatea sistemului constituie o limitare critică, având în vedere că eșecurile de rețea pot afecta cluster-uri întregi de module cu funcționalități diverse. Această vulnerabilitate poate duce la pierderea datelor importante, întreruperea actuatorilor critici sau la oprirea serviciilor de control automat, fiind deosebit de problematică în aplicații critice unde continuitatea funcționării întregului ecosistem modular este esențială.

2.2 Obiectivele Cercetării

Această cercetare urmărește să dezvolte o arhitectură modulară pentru rețele de dispozitive distribuite folosind protocolul ESP-NOW, care să depășească limitările sistemelor tradiționale bazate pe WiFi prin suportul unui ecosistem divers de module conectate. Un obiectiv fundamental constă în implementarea unui sistem de comunicație master-slave pentru gestionarea optimă a datelor și comenzilor, care să permită coordonarea centralizată a informațiilor provenite de la diverse tipuri de module configurate pentru scopuri diverse.

Cercetarea își propune, de asemenea, să creeze o interfață de management bazată pe web pentru monitorizarea și controlul în timp real, oferind utilizatorilor posibilitatea de a accesa, configura și controla întregul ecosistem modular de la distanță prin intermediul unui browser web standard. Un alt obiectiv esențial este să demonstreze avantajele de scalabilitate și fiabilitate ale sistemului propus prin testarea și validarea performanțelor în diverse scenarii cu module eterogene. În final, cercetarea urmărește să ofere un sistem comprehensiv de configurare și setup care să simplifice procesul de implementare pentru module cu funcționalități diverse și să facă tehnologia accesibilă chiar și utilizatorilor fără cunoștințe tehnice avansate, indiferent de tipul sau complexitatea modulelor conectate.



3 Considerații teoretice

3.1 Protocolul ESP-NOW

ESP-NOW constituie un protocol de comunicație wireless proprietar dezvoltat de Espressif Systems, specific pentru microcontrolerele din familiile ESP32 și ESP8266. Acest protocol implementează o arhitectură peer-to-peer care permite comunicația directă între dispozitive fără necesitatea unei infrastructuri intermediare precum routere Wi-Fi sau access point-uri, oferind o soluție eficientă pentru aplicații care necesită transfer rapid de date cu latență minimă și consum energetic redus, așa cum se poate observa în Figura 3.1.



Figura 3.1 Diagramă arhitecturală ESP-NOW

În prezent, ESP-NOW suportă două versiuni: v1.0 și v2.0. Lungimea maximă a pachetului suportată de dispozitivele v2.0 este de 1470 bytes (**ESP_NOW_MAX_DATA_LEN_V2**), în timp ce lungimea maximă a pachetului suportată de dispozitivele v1.0 este de 250 bytes (**ESP_NOW_MAX_DATA_LEN**). Această evoluție marchează o îmbunătățire substanțială în capacitatea de transfer de date, eliminând una dintre principalele limitări ale versiunii inițiale.

Fundamentul tehnologic al ESP-NOW se bazează pe stratul fizic IEEE 802.11, utilizând aceleași frecvențe radio ca Wi-Fi standard (2.4 GHz), dar implementând un protocol de nivel superior optimizat pentru comunicația punct-la-punct și unul-la-mulți. Spre deosebire de protocoalele Wi-Fi tradiționale, primele cinci straturi superioare din modelul OSI sunt simplificate într-un singur strat în ESP-NOW, astfel datele nu trebuie să fie transmise prin stratul de rețea, stratul de transport, stratul de sesiune, stratul de prezentare și stratul de aplicație. [2]



3.1.1 Evoluția către Versiunea 2.0

Dispozitivele v2.0 sunt capabile să primească pachete atât de la dispozitivele v2.0, cât și de la cele v1.0. În contrast, dispozitivele v1.0 pot primi pachete doar de la alte dispozitive v1.0. Cu toate acestea, dispozitivele v1.0 pot primi pachete v2.0 dacă lungimea pachetului este mai mică sau egală cu 250 bytes. Pentru pachetele care depășesc această lungime, dispozitivele v1.0 vor fie trunchia datele la primii 250 de bytes, fie vor respinge complet pachetul.

ESP-NOW v2.0 suportă conexiunea și controlul dispozitivelor de tipul unu-la-mulți și mulți-la-mulți, putând fi utilizat pentru transmisia în masă a datelor, precum configurarea rețelei, actualizarea firmware-ului și debugging. Această versiune introduce funcționalități avansate care transformă protocolul dintr-un simplu mecanism de comunicație într-o platformă completă pentru managementul ecosistemelor IoT complexe.

Noile capacități includ securitatea prin handshake și comunicația criptată între dispozitivele inițiatoare și cele care răspund, provisionarea Wi-Fi pe dispozitivul inițiator prin aplicație și apoi configurarea rețelei Wi-Fi a dispozitivelor care răspund prin ESP-NOW, precum și debugging-ul dispozitivelor prin comenzi și obținerea de log-uri de la dispozitivele care răspund.

3.1.2 Arhitectura Protocolului și Formatele de Frame

ESP-NOW utilizează un frame de acțiune vendor-specific pentru a transmite datele ESP-NOW. Structura frame-ului include un header MAC de 24 bytes, urmat de un cod de categorie care indică categoria vendor-specific (valoarea 127), organizator identificier-ul unic Espressif (0x18fe34), valori random pentru prevenirea atacurilor de tip relay și conținutul vendor-specific.

Pentru versiunea v2.0, dimensiunea maximă este de 1512 bytes ($1470 + 6 \cdot 7$), iar pentru versiunea v1.0, dimensiunea este de 257 bytes ($250 + 7$). Diferențele structurale dintre versiuni se manifestă în organizarea elementelor vendor-specific, versiunea 2.0 introducând un bit suplimentar "More data" care facilitează gestionarea fragmentării datelor pentru pachete mari.

Stratul de securitate implementează protocolul CCMP (CTR with CBC-MAC Protocol) pentru protejarea frame-ului de acțiune. Dispozitivul Wi-Fi menține o Primary Master Key (PMK) și



Dezvoltarea unui Sistem Modular dedicat Managementului în Aplicațiile cu Automatizare

mai multe Local Master Keys (LMKs, fiecare dispozitiv pereche având o LMK), asigurând un model de securitate distribuit și flexibil. [3]

3.1.3 Capacități Avansate și Aplicații

ESP-NOW poate fi utilizat pentru transmisia în masă a datelor precum actualizarea firmware-ului. Sistemul de OTA (Over-The-Air) implementat în versiunea 2.0 oferă funcționalități avansate precum reluarea actualizării de la punctul de întrerupere: când se folosește ESP-NOW pentru actualizarea firmware-ului, firmware-ul va fi sub-împachetat într-o dimensiune fixă și scris în flash unul câte unul, iar dispozitivul va înregistra pachetele actualizate. Dacă procesul de actualizare este întrerupt, dispozitivul va solicita doar pachetele de firmware rămase pentru a relua actualizarea de la punctul de întrerupere.

ESP-NOW poate suporta actualizarea simultană a mai multor dispozitive. 50 de dispozitive pot fi actualizate în 3 minute. Această capacitate de scalare oferă avantaje semnificative pentru implementarea și mentenanța rețelelor mari de dispozitive IoT, reducând dramatic timpul și resursele necesare pentru actualizările de sistem.

ESP-NOW poate fi utilizat pentru primirea log-urilor de rulare pentru debugging. Poate fi folosit în scenarii unde dispozitivele nu pot fi contactate direct din cauza electricității de înaltă tensiune sau temperaturii ridicate. Cu suportul conexiunilor mulți-la-mulți, inițiatorul poate primi log-uri de la mai multe dispozitive care răspund pentru a diagnostica rapid defecțiunile dispozitivelor.

3.1.4 Performanță și Limitări

Cu o putere de transmisie de până la +21dBm, chiar și dispozitivele care se află la distanță de 200 de metri una de cealaltă într-un spațiu deschis pot fi controlate cu ușurință. Rata de biți implicită a ESP-NOW este de 1 Mbps, optimizată pentru aplicații care prioritizează fiabilitatea și eficiența energetică față de viteza de transfer.

În modul Station, sunt suportați cel mult 10 peers criptați; cel mult 6 în modul SoftAP sau SoftAP + Station; sunt suportați mai mulți peers necriptați. Dacă se folosesc pachete unicast, până la 20 de dispozitive pot fi perecheate și controlate simultan. Pentru aplicații cu cerințe de scalabilitate mai mari, se pot folosi pachete broadcast și se pot furniza adresele de destinație în payload.



Dispozitivul nu poate schimba canalele după conectarea la Wi-Fi. Poate transmite și primi date doar pe canalul Wi-Fi curent. Această limitare impune considerații de design important pentru aplicații hibride care combină conectivitatea Wi-Fi cu comunicația ESP-NOW.

3.1.5 Implementarea și Compatibilitatea

ESP-NOW poate lucra cu Wi-Fi și Bluetooth LE și suportă seriile ESP8266, ESP32, ESP32-S și ESP32-C. Această compatibilitate extinsă permite integrarea flexibilă în ecosisteme IoT existente și facilitează migrarea între diferite generații de hardware fără modificări substanțiale ale protocolului de comunicație.

ESP-NOW oferă o nouă metodă de provisionare pe lângă provisionarea Wi-Fi și Bluetooth. Primul dispozitiv se configurează prin bluetooth pentru rețea, iar celelalte dispozitive nu trebuie să fie configurate cu informațiile SSID/parolă, deoarece primul dispozitiv conectat la rețea poate trimite aceste informații direct la celelalte. Această abordare simplifică dramatic procesul de configurare inițială în rețele cu multiple noduri.

Ecosistemul de dezvoltare pentru ESP-NOW v2.0 include suport pentru componente avansate prin ESP Component Registry, oferind dezvoltatorilor acces la exemple pre-configurate pentru diverse scenarii de utilizare, de la sisteme simple de control până la soluții complete de management pentru rețele IoT complexe.

3.2 Componente Hardware

Implementarea soluțiilor IoT (Internet of Things) moderne necesită o selecție atentă a componentelor hardware care să asigure funcționalitatea dorită, eficiența energetică și capacitățile de comunicație adecvate contextului aplicației. Evoluția microcontrolerelor și a dispozitivelor periferice a permis dezvoltarea de sisteme embedded cu performanțe ridicate și costuri reduse, facilitând democratizarea tehnologiilor de automatizare și monitorizare.

3.2.1 Placa de Dezvoltare ESP32

ESP32 reprezintă o familie de microcontrolere dezvoltată de Espressif Systems, caracterizată prin integrarea nativă a capacităților de comunicație wireless și prin performanțele superioare în procesarea datelor. Arhitectura acestui microcontroler se bazează pe un procesor dual-core



Dezvoltarea unui Sistem Modular dedicat Managementului în Aplicațiile cu Automatizare

Xtensa LX6 care operează la frecvențe de până la 240 MHz, oferind resurse computaționale substanțiale pentru aplicații complexe care necesită procesarea simultană a multiple fluxuri de date. [9]

Capacitățile de conectivitate wireless constituie principalul avantaj competitiv al ESP32, integrând suport nativ pentru standardele Wi-Fi 802.11 b/g/n și Bluetooth Classic/Low Energy (BLE). Implementarea dual-mode a stivei Bluetooth permite comunicația atât cu dispozitive tradiționale, cât și cu senzori și actuatori optimizați pentru eficiența energetică. Subsistemul Wi-Fi suportă multiple moduri de operare, incluzând stația (client), access point și mode mixte, facilitând implementarea de soluții flexibile de networking pentru diverse scenarii de utilizare.

Arhitectura memoriei ESP32 include 520 KB de SRAM intern și suport pentru memorie flash externă de până la 16 MB, oferind spațiu adecvat pentru stocarea programului și a datelor aplicației. Unitatea de management a energiei (PMU) implementează multiple moduri de somn configurabile, permițând optimizarea consumului energetic în funcție de cerințele specifice ale aplicației. Capacitatea de trezire din somn prin diverse surse, inclusiv temporizatoare, întreruperi externe sau pachete de rețea, asigură responsivitatea sistemului în scenarii de monitorizare continuă.

Interfețele periferice (Figura 3.1) disponibile includ multiple canale ADC cu rezoluție de 12 biți, interfețe de comunicație serială (UART, SPI, I2C), generatoare PWM pentru controlul actuatorilor și suport pentru protocoale specializate precum I2S pentru aplicații audio. Flexibilitatea în configurarea pinilor GPIO prin matricea de rutare permite adaptarea optimă la cerințele hardware specifice ale proiectului, minimizând complexitatea circuitului extern și costurile de implementare.

Ecosistemul de dezvoltare pentru ESP32 include suportul pentru multiple framework-uri și medii de programare, de la Arduino IDE pentru prototiparea rapidă până la ESP-IDF (Espressif IoT Development Framework) pentru dezvoltarea profesională. Integrarea cu platforme cloud populare și biblioteca extensivă de exemple facilitează implementarea rapidă a soluțiilor conectate, reducând timpul de dezvoltare și barierele tehnice pentru dezvoltatorii cu experiență limitată în domeniul embedded.



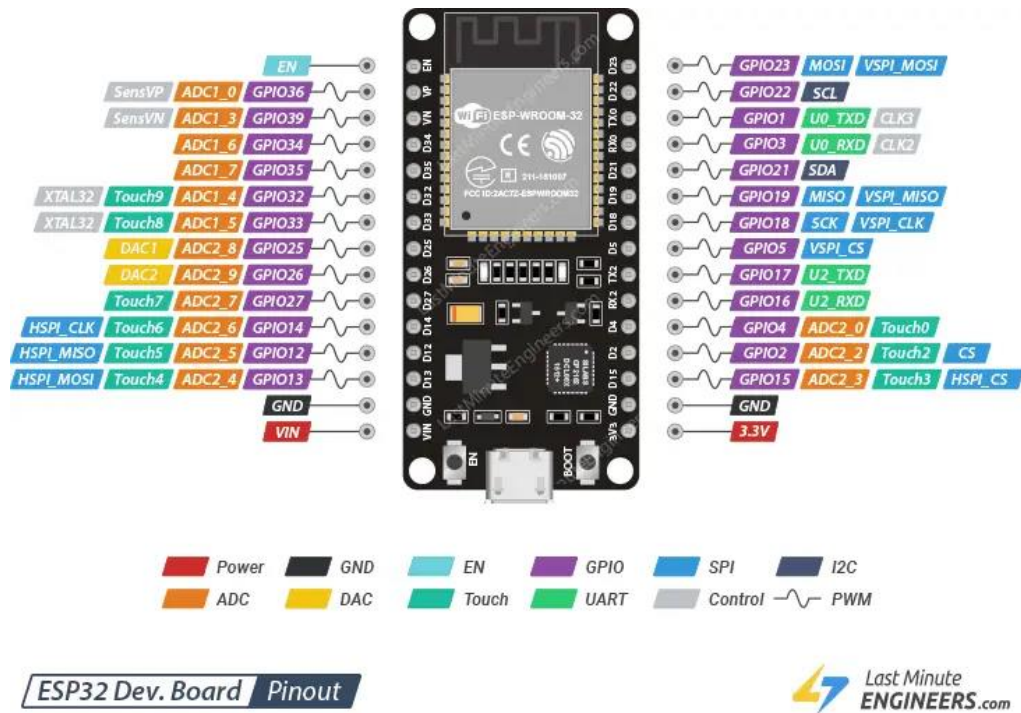


Figura 3.2 Pinout-ul ESP32 DevKit V1

3.2.2 Afișaj OLED SSD1306

Afișajul OLED (Organic Light-Emitting Diode) SSD1306 constituie o soluție de vizualizare compactă și eficientă energetic, bazată pe tehnologia semiconductorilor organici care emit lumină prin aplicarea unei tensiuni electrice. Această tehnologie elimină necesitatea unei surse de iluminare externe, rezultând într-un contrast superior, unghiuri de vizualizare extinse și un profil fizic redus care facilitează integrarea în dispozitive portabile și aplicații cu constrângeri de spațiu.

Controllerul integrat SSD1306 gestionează o matrice de pixeli cu rezoluția de 128x64, organizată în 8 pagini de câte 8 pixeli pe verticală. Arhitectura internă include o memorie RAM grafică (GDDRAM) de 1024 bytes care stochează starea fiecărui pixel, permitând actualizarea selectivă a conținutului fără necesitatea retransmiterii complete a imaginii. Această caracteristică optimizează eficiența comunicației cu microcontrolerul și reduce latența în actualizarea informațiilor afișate.

Interfețele de comunicație suportate includ I2C și SPI, oferind flexibilitate în integrarea cu diverse microcontrolere și optimizarea utilizării pinilor GPIO disponibili. Protocolul I2C permite conectarea multiplelor dispozitive pe același bus, facilitând implementarea de sisteme cu afișaje



Dezvoltarea unui Sistem Modular dedicat Managementului în Aplicațiile cu Automatizare

multiple sau cu senzori suplimentari. Configurabilitatea adresei I2C prin intermediul unui pin dedicat permite evitarea conflictelor în sisteme complexe cu multiple componente pe același bus de comunicație.

Managementul energiei implementat în SSD1306 include multiple moduri de operare, de la funcționarea normală până la starea de sleep profund cu consum minimal. Capacitatea de control individual al luminozității permite adaptarea la condițiile de iluminare ambientală și optimizarea autonomiei în aplicații alimentate cu baterii. Funcțiile de scroll hardware și manipularea directă a memoriei grafice oferă posibilități avansate de animație și efecte vizuale fără solicitarea intensivă a resurselor microcontrolerului.

Dimensiunile fizice compacte și greutatea redusă fac din SSD1306 o soluție ideală pentru dispozitive wearable, sisteme de monitorizare portabile și interfețe utilizator pentru echipamente industriale. Rezistența la vibrații și temperatura extinsă de operare asigură fiabilitatea în medii industriale și aplicații automotive, în timp ce consumul redus de energie și lipsa componentelor mobile contribuie la o durată de viață îndelungată a sistemului.

3.3 Arhitectura Rețelei

Sistemul implementează o topologie centralizată în stea unde nodul master funcționează ca un coordonator central care gestionează comunicația cu toate nodurile slave din ecosistemul modular. Nodurile slave reprezintă diverse tipuri de module care comunică cu master-ul pentru transmiterea datelor, primirea comenzilor sau sincronizarea operațiunilor. [11] O astfel de topologie este bine cunoscută și folosită, exemplul din Figura 3.3 prezentând o structură des întâlnită în departamentele de dezvoltare tehnologică.

Această arhitectură a fost aleasă pentru a optimiza eficiența și simplitatea sistemului, indiferent de diversitatea funcțională a modulelor conectate. În acest fel se construiește o independență între nodurile slave ceea ce reduce posibilitatea de erori și defecțiuni la nivelul lor. Nodul master, de pe altă parte, devine un punct vulnerabil al sistemului, fără opțiuni secundare, apariția unui defect poate duce la ieșirea din funcțiune a întregului sistem.



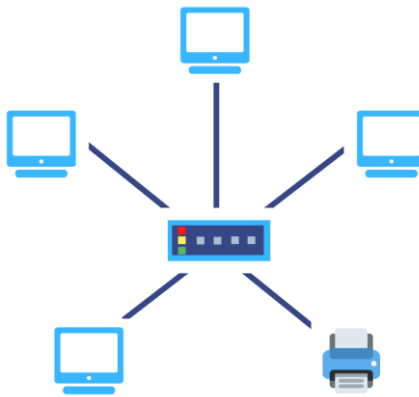


Figura 3.3 Diagramă topologie în stea

3.4 Tehnologi Web

Dezvoltarea aplicațiilor web moderne necesită o înțelegere aprofundată a tehnologiilor fundamentale care constituie baza infrastructurii digitale contemporane. Evoluția rapidă a standardelor web și a protocoalelor de comunicație a generat un ecosistem complex de instrumente și metodologii care permit crearea de soluții interactive, scalabile și performante. [10]

3.4.1 WebSocket Protocol

Protocolul WebSocket reprezintă o tehnologie de comunicație bidirecțională în timp real între client și server, standardizată prin RFC 6455. Spre deosebire de protocolul HTTP tradițional, care funcționează pe baza unui model request-response, WebSocket stabilește o conexiune persistentă care permite schimbul continuu de date între cele două părți implicate în comunicație.

Implementarea tehnologiei WebSocket adresează limitările inerente ale protocolului HTTP în contextul aplicațiilor care necesită actualizări frecvente ale conținutului. Prin eliminarea overhead-ului generat de headerele HTTP repetitive și prin menținerea conexiunii deschise, protocolul oferă o latență considerabil redusă și o eficiență sporită în transmiterea datelor. Această caracteristică îl face ideal pentru aplicații precum chat-urile în timp real, jocurile multiplayer online, platformele de colaborare și sistemele de monitorizare care necesită actualizări instantanee.



Dezvoltarea unui Sistem Modular dedicat Managementului în Aplicațiile cu Automatizare

Procesul de stabilire a conexiunii WebSocket începe cu un handshake HTTP standard, în cadrul căruia clientul transmite o cerere de upgrade la protocolul WebSocket. Serverul confirmă disponibilitatea pentru această tranziție prin returnarea unui cod de status 101 Switching Protocols, moment în care conexiunea HTTP se transformă într-o conexiune WebSocket completamente funcțională. Această abordare asigură compatibilitatea cu infrastructura web existentă, inclusiv cu proxy-urile și firewall-urile care procesează traficul HTTP.

3.4.2 AsyncWebServer

AsyncWebServer constituie o bibliotecă software specializată în gestionarea cererilor HTTP de maniera asincronă, optimizată pentru microcontrolere și sisteme embedded cu resurse limitate. Arhitectura acestei biblioteci se bazează pe paradigma programării asincrone, care permite procesarea simultană a multiple cereri web fără blocarea execuției programului principal.

Implementarea asincronă oferă avantaje semnificative în termeni de performanță și responsivitate, mai ales în contextul aplicațiilor IoT (Internet of Things) unde resursele de procesare și memoria sunt constrained. Prin evitarea operațiunilor blocante, serverul poate gestiona eficient multiple conexiuni simultane, servind conținut static, procesând cereri API și coordonând interacțiuni cu senzori sau actuatori într-un mod fluid și predictibil.

Bibliotecă suportă multiple funcționalități avansate, inclusiv gestionarea automată a fișierelor statice, implementarea WebSocket-urilor pentru comunicația în timp real, autentificarea utilizatorilor și managementul sesiunilor. Flexibilitatea în definirea rutelor și a handler-elor personalizate permite dezvoltatorilor să creeze interfețe web complexe și interactive, adaptate specificului aplicației respective. Compatibilitatea cu standardele web moderne asigură integrarea seamless cu framework-urile frontend contemporane.

3.4.3 HTML5 (HyperText Markup Language)

HTML5 reprezintă cea mai recentă iterație majoră a limbajului de marcare pentru structurarea conținutului web, introducând numeroase elemente semantice și funcționalități native care elimină dependența de plugin-uri externe. Această versiune a standardului aduce îmbunătățiri substanțiale în ceea ce privește accesibilitatea, optimizarea pentru motoarele de căutare și capacitățile multimedia native.



Elementele semantice introduse de HTML5, cum ar fi `<header>`, `<nav>`, `<article>`, `<section>`, `<aside>` și `<footer>`, permit o structurare mai clară și mai semnificativă a documentelor web. Această abordare facilitează interpretarea conținutului atât de către browsere, cât și de către tehnologiile asistive, îmbunătățind experiența utilizatorilor cu nevoi speciale și optimizând indexarea de către motoarele de căutare.

Capacitățile multimedia native ale HTML5 includ elementele `<video>` și `<audio>`, care permit integrarea conținutului multimedia fără necesitatea de plugin-uri suplimentare precum Adobe Flash. Suportul pentru multiple formate de fișiere și funcționalitățile de control programatic oferă dezvoltatorilor un control granular asupra experienței multimedia. API-urile JavaScript asociate permit implementarea de playere personalizate și funcționalități avansate de procesare audio-video.

Tehnologiile de stocare locală introduse de HTML5, incluzând Local Storage, Session Storage și IndexedDB, revoluționează modul în care aplicațiile web gestionează datele pe partea clientului. Aceste soluții permit stocarea persistentă a informațiilor în browserul utilizatorului, reducând necesitatea comunicației frecvente cu serverul și îmbunătățind performanța aplicației, mai ales în scenarii cu conectivitate limitată sau intermitentă.

3.4.4 CSS3 (Cascading Style Sheets)

CSS3 reprezintă evoluția standard-ului de stilizare pentru documentele web, introducând module specializate care extind considerabil posibilitățile de design și animație. Arhitectura modulară permite implementarea incrementală a funcționalităților, asigurând compatibilitatea cu versiunile anterioare și flexibilitatea în adoptarea noilor caracteristici.

Sistemul de layout flexibil introdus prin Flexbox și CSS Grid revoluționează modul în care dezvoltatorii abordează structurarea elementelor pe pagină. Flexbox oferă o soluție elegantă pentru alinierea și distribuirea elementelor într-o dimensiune, fiind ideal pentru componente de tip navbar, carduri și liste. CSS Grid extinde aceste capacități la două dimensiuni, permițând crearea de layout-uri complexe și responsive fără necesitatea framework-urilor externe sau a hack-urilor bazate pe float sau positioning.



Dezvoltarea unui Sistem Modular dedicat Managementului în Aplicațiile cu Automatizare

Funcționalitățile de animație și tranziție native ale CSS3 elimină dependența de JavaScript pentru efecte vizuale simple și moderate. Proprietățile **transition** permit crearea de efecte fluide la modificarea valorilor CSS, în timp ce **@keyframes** și proprietatea **animation** oferă control complet asupra animațiilor complexe. Aceste capacități, combinate cu transformările 2D și 3D, permit crearea de interfețe interactive și angajante fără impactul asupra performanței asociat cu animațiile JavaScript.

Preprocesoarele CSS, cum ar fi Sass, Less și Stylus, extind funcționalitatea nativă a CSS3 prin introducerea de variabile, funcții, mixins și structuri de control. Aceste instrumente facilitează menținerea și scalabilitatea codului CSS în proiecte mari, promovând reutilizarea codului și respectarea principiilor DRY (Don't Repeat Yourself). Sistemele de build moderne integrează aceste preprocesoare în workflow-ul de dezvoltare, automatizând procesul de compilare și optimizare.

3.4.5 JavaScript

JavaScript constituie limbajul de programare fundamental pentru dezvoltarea aplicațiilor web interactive, evoluând de la un simplu limbaj de scripting la o platformă completă pentru dezvoltarea atât pe partea clientului, cât și pe partea serverului. Natura interpretată și dinamica tipizării fac din JavaScript un instrument versatil pentru implementarea logicii aplicației și a interacțiunilor utilizator.

Paradigmele de programare suportate de JavaScript includ stilul procedural, orientat pe obiect și funcțional, oferind dezvoltatorilor flexibilitatea de a alege abordarea cea mai potrivită pentru contextul specific. Suportul nativ pentru closure-uri, higher-order functions și asynchronous programming prin Promise-uri și async/await facilitează implementarea de algoritmi complecși și gestionarea operațiunilor care consumă timp, precum cererile de rețea sau procesarea fișierelor. Ecosistemul JavaScript modern se bazează pe manageri de pachete precum npm și yarn, care facilitează integrarea bibliotecilor și framework-urilor de terți. Framework-urile populare cum ar fi React, Vue.js și Angular oferă structuri organizaționale pentru dezvoltarea aplicațiilor single-page (SPA), promovând reutilizabilitatea componentelor și separarea responsabilităților. Aceste instrumente implementează concepte avansate precum virtual DOM, reactive programming și dependency injection.



Standardizarea prin ECMAScript asigură evoluția predictibilă și compatibilitatea cross-platform a limbajului. Versiunile recente (ES6+) introduc funcționalități moderne precum arrow functions, destructuring, template literals, classes și modules, care îmbunătățesc lizibilitatea codului și productivitatea dezvoltatorilor. Transpilerii precum Babel permit utilizarea acestor funcționalități moderne chiar și în medii care nu le suportă nativ, asigurând compatibilitatea cu browsere mai vechi.

4 Evoluția Sistemului - Procesul Iterativ de Dezvoltare

Procesul de dezvoltare a sistemului a fost caracterizat de o evoluție iterativă sistematică, în care fiecare versiune a reprezentat o răspuns direct la provocările tehnice și limitările funcționale identificate în etapa precedentă, rezultând într-o progresie logică de la un concept simplu de demonstrație către o soluție completă și robustă pentru monitorizarea distribuită. Evoluția sa se poate observa în fișierele de cod prezente în ANEXA A

4.1 Prototip Initial (V1)

4.1.1 Procesul de realizare

Prima versiune a sistemului s-a concentrat pe stabilirea fundamentelor comunicației ESP-NOW prin implementarea unei configurări statice a dispozitivelor cu adrese MAC hardcodate direct în cod. [7] Această abordare, deși limitată în flexibilitate, a permis testarea și validarea conceptelor de bază ale protocolului. Comunicația peer-to-peer ESP-NOW de bază a fost implementată pentru a demonstra viabilitatea transmiterii datelor între dispozitive fără infrastructură WiFi tradițională.

Implementarea simplă WebServer folosind cereri HTTP sincrone a oferit primele capacități de monitorizare la distanță, deși cu limitări în termeni de performanță și experiență utilizator. [4] Configurarea manuală a dispozitivului prin modificarea codului a reprezentat metoda primară de personalizare, necesitând cunoștințe de programare și recompilarea pentru fiecare schimbare.

Interfața web inițială s-a bazat pe pagini HTML statice cu refresh manual, oferind o modalitate simplă dar funcțională de vizualizare a datelor colectate. Afișarea stării dispozitivului cu CSS stilizat a îmbunătățit prezentarea informațiilor, făcând interfața mai atractivă și mai ușor de interpretat. Vizualizarea de bază a datelor modulelor a permis monitorizarea valorilor în timp



Dezvoltarea unui Sistem Modular dedicat Managementului în Aplicațiile cu Automatizare

real, în timp ce afișarea informațiilor de rețea a oferit insight-uri asupra conectivității și stării sistemului. Acesta se poate observa în ANEXA B.

4.1.2 Provocări întâlnite

Lipsa capacității de configurare dinamică a reprezentat cea mai semnificativă limitare, necesitând modificări de cod pentru orice ajustare a setărilor. Scalabilitatea limitată datorită managementului hardcodat al peer-urilor a împiedicat extinderea facilă a rețelei. Lipsa persistenței datelor între reboot-uri a însemnat că orice configurări temporare se pierdeau la restartarea sistemului. Cerința de refresh manual pentru datele actualizate a afectat experiența utilizatorului, în timp ce procesul complex de setup care necesita compilarea codului a făcut sistemul inaccesibil pentru utilizatorii non-tehnici.

4.2 Interfața Web (V2)

4.2.1 Îmbunătățiri

A doua iterație a introdus biblioteca ESPUI pentru a crea o interfață web dinamică care să depășească limitările paginilor statice ale versiunii anterioare. Această bibliotecă a permis dezvoltarea unei interfețe mai sofisticate și interactive, oferind utilizatorilor o experiență mult îmbunătățită. [12] Actualizările UI în timp real prin conexiuni WebSocket au eliminat necesitatea refresh-ului manual, oferind o experiență continuă și fluidă de monitorizare.

Interfața cu tab-uri pentru organizarea mai bună a informațiilor a îmbunătățit semnificativ navigabilitatea și accesibilitatea datelor, permițând utilizatorilor să acceseze rapid informațiile relevante. Tab-urile individuale pentru dispozitive au permis monitorizarea detaliată a fiecărui senzor, oferind informații granulare când era necesar. Tab-ul setări pentru parametrii de configurare a început să introducă capacități de personalizare, deși încă limitate. Actualizările de date în timp real fără refresh de pagină au transformat experiența utilizatorului, oferind informații mereu actuale. Aceasta se poate observa în ANEXA C.

4.2.2 Provocări

Configurarea încă necesita modificarea manuală a codului pentru aspectele fundamentale ale sistemului, limitând accesibilitatea pentru utilizatorii non-tehnici. Lipsa stocării persistente pentru setări a însemnat că personalizările se pierdeau la restart. Experiența limitată de setup



prima dată a rămas o barieră în adoptarea sistemului. Dependența de biblioteca ESPUI a creat unele probleme de stabilitate, în special în scenarii cu multiple conexiuni simultane sau în condiții de rețea instabile.

4.3 Configurare Dinamică (V3)

4.3.1 Îmbunătățiri

Versiunea 3.0 a marcat o revoluție în abordarea configurării sistemului prin introducerea stocării persistente bazate pe EEPROM cu validare checksum, eliminând dependența de modificările manuale ale codului. [6] Această implementare a asigurat că toate setările utilizatorului sunt păstrate în siguranță chiar și în cazul întreruperilor de alimentare sau al restartărilor sistemului. Modul de setup prima dată cu punct de acces dedicat a creat o experiență de configurare inițială accesibilă și intuitivă, permițând utilizatorilor să configureze dispozitivele fără cunoștințe tehnice avansate.

Gestionarea identității dispozitivului prin nume, ID și rol a permis organizarea clară și identificarea facilă a fiecărui element din rețea. Setările de rețea comprehensive, incluzând credențiale WiFi și configurarea punctului de acces, au oferit flexibilitate în diverse scenarii de implementare. Selecția tipului de modul senzor a facilitat personalizarea funcționalității fiecărui dispozitiv conform cerințelor specifice. Managementul adreselor MAC ale dispozitivelor peer a automatizat procesul de înregistrare și recunoaștere în rețea.

Checksum-ul de configurare pentru integritatea datelor a asigurat detectarea și prevenirea corupției informațiilor critice. Capacitatea de factory reset prin butonul hardware a oferit o metodă de recuperare în cazul problemelor de configurare. Atribuirea automată a ID-ului dispozitivului pentru slave-uri a simplificat procesul de setup și a prevenit conflictele de identificare. Sistemul de backup și recuperare a configurației a adăugat un nivel suplimentar de siguranță și fiabilitate.

Eliminarea necesității de recompilare a codului pentru setup a transformat fundamental experiența utilizatorului, făcând sistemul accesibil unui public mult mai larg. Procesul de implementare simplificat a redus timpul și expertiza necesare pentru punerea în funcțiune. Fiabilitatea sistemului îmbunătățită prin stocarea persistentă a eliminat problemele legate de



Dezvoltarea unui Sistem Modular dedicat Managementului în Aplicațiile cu Automatizare

pierderea configurațiilor. Experiența utilizatorului mai bună pentru utilizatorii non-tehnici a deschis noi oportunități de adoptare în diverse domenii de aplicare.

4.3.2 Provocări

În creșterea complexității programului, stabilitatea bibliotecii ESPUI a început să influențeze negativ experiența utilizatorului, creând probleme de conectivitate, latență ridicată (între 30 - 300s) și artefacte vizuale neplăcute.

4.4 Reorganizare (V4)

Versiunea 4.0 a reprezentat o tranziție tehnologică majoră prin înlocuirea ESPUI cu AsyncWebServer pentru cereri HTTP non-blocante, rezolvând problemele de performanță și stabilitate identificate în versiunile anterioare. Această modificare a permis gestionarea mult mai eficientă a cererilor simultane și a îmbunătățit considerabil responsivitatea sistemului. [5]
Implementarea WebSocket pentru streaming de date în timp real a creat o experiență de utilizare fluidă și dinamică, eliminând întârzierile și necesitatea de refresh manual. [14]

API-ul bazat pe JSON (JavaScript Object Notation) pentru configurare și interogări de stare a standardizat comunicația între componente și a facilitat integrarea cu alte sisteme. [13]
Mecanismul eficient de actualizare pentru reducerea lățimii de bandă a optimizat utilizarea resurselor de rețea, fiind deosebit de important pentru rețelele cu multiple dispozitive active simultan.

4.5 Optimizări Finale (V5, V6)

Versiunea 5.0 a extins semnificativ capacitățile de configurare a rețelei prin introducerea modului Access Point cu SSID și parolă personalizate, oferind utilizatorilor control complet asupra identității rețelei create. Modul client WiFi pentru integrarea în rețeaua existentă a permis conectarea la infrastructuri existente, facilitând accesul la internet și integrarea cu alte servicii. Comutarea dinamică a modului de rețea bazată pe configurare a oferit flexibilitate operațională, permițând adaptarea la diverse scenarii de utilizare fără necesitatea recompilării.

Versiunea 6.0 a sistemului a introdus un sistem comprehensiv de afișaj local prin implementarea unui afișaj OLED multi-pagină cu ciclare automată, oferind acces la informații vitale chiar și în absența conectivității la interfața web. [1] Sistemul prezintă starea sistemului, datele senzorilor și



informațiile de rețea într-un format compact și ușor de citit. Feedback-ul vizual pentru sănătatea sistemului și conectivitate oferă indicatori clari despre funcționarea optimă sau problemele care necesită atenție.

5 Produs Final

5.1 Arhitectura Generală a Sistemului

Sistemul implementează o arhitectură master-slave bazată pe protocolul ESP-NOW, utilizând microcontrolere ESP32 pentru toate nodurile din rețea. Comunicația se realizează prin trei protocoale complementare: ESP-NOW pentru transmiterea datelor senzorilor cu latență redusă (1-10ms), WebSocket pentru actualizările în timp real ale interfeței web, și HTTP pentru configurare și setup.

Toate dispozitivele rulează același cod sursă, diferențierea funcționalității fiind realizată prin parametrii de configurare stocați în EEPROM cu verificare checksum pentru integritate. Raza de comunicație variază între 10 și 200 de metri în funcție de condițiile de mediu, iar sistemul suportă până la 20 dispozitive simultane.

Implementarea necesită plăci ESP32 DevKit V1 și display-uri OLED SSD1306 conectate via I2C, schema electrică fiind prezentată în Figura 5.1. Dependențele software includ bibliotecile ESP32 Arduino Core, ESP-NOW, AsyncWebServer, WebSocket și SSD1306 pentru gestionarea display-ului.



Dezvoltarea unui Sistem Modular dedicat Managementului în Aplicațiile cu Automatizare

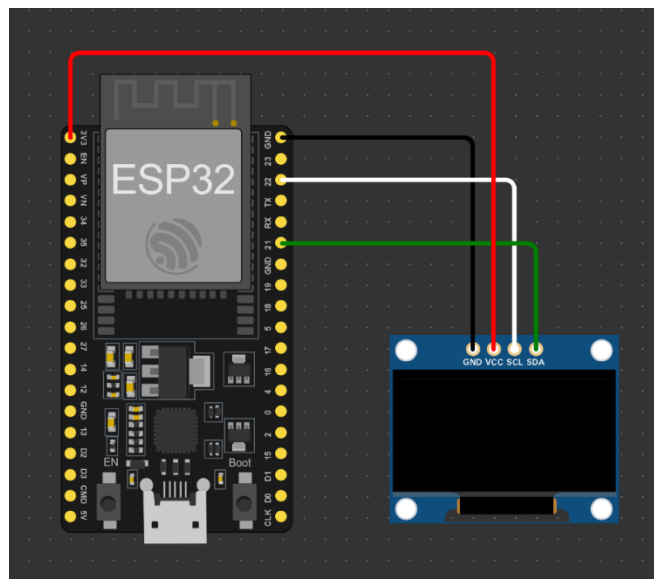


Figura 5.1 Schema electrică a modului simplu

În Figura 5.2 de mai jos sunt prezentate și modulele fizice folosite în dezvoltarea și testarea acestui system.

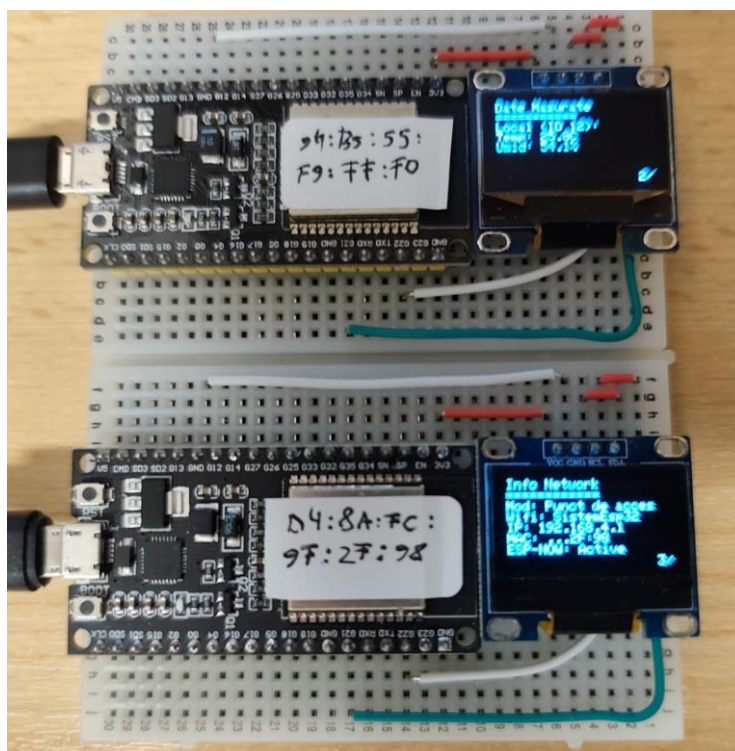


Figura 5.2 Cele două module Esp32 folosite

5.2 Dispozitivul Master

Dispozitivul master servește ca punct central de coordonare al întregii rețele, implementând funcționalități complete de management și interfață utilizator. Server-ul web AsyncWebServer încorporat oferă o interfață accesibilă din orice browser, exemplificată în ANEXA D, eliminând



necesitatea software-ului specializat. Sistemul de configurare bazat pe web permite setup-ul intuitiv prin crearea automată a unui Access Point temporar la prima pornire.

Display-ul OLED multi-pagină oferă monitorizare locală cu rotație automată între trei ecrane: informații de sistem (uptime, dispozitive active), date senzori (temperatura, umiditatea, nivelul de lumină), și status rețea (mod WiFi, IP, MAC). Implementarea din cod permite ciclarea automată la fiecare 3 secunde între aceste pagini.

Funcționalitatea WebSocket asigură actualizări în timp real la fiecare 2 secunde pentru interfața web, transmițând date JSON structurate cu informații despre toate dispozitivele active din rețea. Sistemul de configurare EEPROM păstrează persistent toate setările, incluzând adresele MAC ale dispozitivelor slave, parametrii de rețea și configurația modulelor senzor.

5.3 Dispozitivele Slave

Dispozitivele slave rulează același cod ca master-ul dar cu funcționalități dezactivate. Acestea nu inițializează interfața web, WebSocket sau Puncte de acces, concentrându-se exclusiv pe colectarea și prelucrarea datelor și transmiterea către master.

În cazul lucrării s-au implementat două tipuri de module fictive: *Modul de Temperatură* și *Modulul de Luminozitate*. Acestea simulează valori fictive care apoi sunt trimise, către master pentru a fi afișate.

5.4 Configurare

Configurarea inițială a sistemului urmează o secvență logică și intuitivă care începe cu pornirea dispozitivului în modul de configurare, activat automat la prima utilizare sau prin apăsarea butonului de setup. Crearea punctului de acces pentru interfața de setup oferă o rețea temporară dedicată exclusiv configurării, eliminând necesitatea unei infrastructuri de rețea existente. Configurarea bazată pe web a parametrilor dispozitivului se realizează prin conectarea la această rețea temporară și accesarea unei interfețe simple și ghidate, interfața aceasta fiind regăsită în ANEXA E.

Configurarea rețelei și senzorului permite personalizarea comportamentului sistemului conform cerințelor specifice ale aplicației. Restart-ul automat și funcționarea normală finalizează



Dezvoltarea unui Sistem Modular dedicat Managementului în Aplicațiile cu Automatizare

procesul, trecând sistemul de la configurare la funcționarea productivă fără intervenție suplimentară.

Numele și identificarea dispozitivului permit organizarea clară a rețelelor complexe cu multiple noduri și facilitează managementul pe termen lung. Selecția rolului master sau slave determină funcționalitatea principală a dispozitivului în arhitectura rețelei. Configurarea tipului de modul senzor specifică ce date vor fi colectate și transmise de către acel dispozitiv particular. Setările de conectivitate de rețea definesc modul în care dispozitivul se integrează în infrastructura existentă.

Adresele MAC ale dispozitivelor peer stabilesc relațiile de comunicație în rețeaua ESP-NOW. Intervalele de actualizare și timing permit optimizarea între responsivitatea sistemului și consumul de energie în funcție de cerințele aplicației.

Stocarea EEPROM pentru datele de configurare asigură că toate setările sunt păstrate permanent chiar în cazul întreruperilor de alimentare sau al restartărilor sistemului. Validarea checksum pentru integritatea datelor detectează și previne corupția informațiilor critice, asigurând fiabilitatea pe termen lung. Capacitatea de factory reset oferă o metodă sigură de resetare completă în cazul problemelor sau al necesității de reconfigurare. Backup-ul și restaurarea configurației facilitează migrarea setărilor între dispozitive sau recuperarea după probleme hardware.

6 Concluzie

6.1 Rezumat

Această cercetare a dezvoltat și implementat cu succes un sistem complet de management bazat pe ESP32 care adresează limitările cheie ale rețelelor tradiționale de senzori IoT. Sistemul demonstrează îmbunătățiri semnificative în fiabilitatea comunicației și ușurința implementării comparativ cu soluțiile convenționale bazate pe WiFi.

Interfața prietenoasă, realizată prin sistemul de management bazat pe web cu actualizări în timp real, face tehnologia accesibilă unui public larg, depășind barierele tehnice tradiționale. Configurarea flexibilă care suportă multiple tipuri de aplicații și implementarea ușoară demonstrează adaptabilitatea sistemului la diverse scenarii de utilizare. Design-ul robust,



caracterizat de gestionarea comprehensivă a erorilor și mecanismele de recuperare, asigură funcționarea fiabilă în condiții reale de utilizare.

6.2 Contribuții tehnice

Sistemul aduce mai multe contribuții tehnice importante în domeniul rețelelor de senzori IoT:

Inovația Protocolului: Sistemul demonstrează o implementare eficientă a ESP-NOW pentru rețelele de senzori, optimizând acest protocol pentru cerințele specifice ale aplicațiilor IoT. Modelele de comunicație au fost optimizate pentru eficiența energetică, rezultând în reduceri semnificative ale consumului comparativ cu alternativele tradiționale. Transmiterea fiabilă a datelor cu gestionarea erorilor asigură integritatea informațiilor chiar în condiții de comunicație dificile, contribuind la robustețea generală a sistemului.

Designul Sistemului: Arhitectura modulară dezvoltată oferă flexibilitate pentru diverse aplicații practice. Sistemul eficient de configurare și management simplifică procesul de implementare și întreținere, reducând costurile operaționale pe termen lung. Interfața de monitorizare în timp real cu capabilități web oferă acces imediat la informații critice și control asupra sistemului din orice locație cu conectivitate internet.

Experiența Utilizatorului: Procesul de setup intuitiv a fost proiectat special pentru utilizatorii non-tehnici, eliminând barierele tradiționale în adoptarea tehnologiilor IoT. Capacitățile de monitorizare și management oferă control complet asupra sistemului fără a compromite simplitatea utilizării. Opțiunile de acces local și la distanță asigură flexibilitatea operațională în diverse scenarii de implementare, de la aplicații domestice la instalații industriale complexe.

6.3 Evaluarea performanței

Sistemul demonstrează performanță superioară în mai multe domenii cheie:

Fiabilitatea Comunicației: Rata de succes a livrării mesajelor de 99%+ în medii tipice demonstrează robustețea protocolului ESP-NOW în aplicații practice reale. Această fiabilitate ridicată este crucială pentru aplicații critice unde pierderea datelor poate avea consecințe importante. Gestionarea automată a erorilor și mecanismele de reîncercare asigură că informațiile importante ajung la destinație chiar și în condiții de comunicație dificile.



Dezvoltarea unui Sistem Modular dedicat Managementului în Aplicațiile cu Automatizare

Performanța robustă în medii RF provocatoare, cum ar fi zonele cu multe interferențe wireless, confirmă adaptabilitatea sistemului la diverse condiții de implementare.

Extensibilitatea: Adăugarea ușoară a noilor noduri permite extinderea graduală a rețelei fără necesitatea reconfigurării infrastructurii existente. Managementul centralizat al rețelelor mari simplifică operarea și întreținerea sistemelor complexe cu zeci de dispozitive. Utilizarea eficientă a resurselor asigură că performanța sistemului se menține constantă chiar pe măsură ce numărul de dispozitive conectate crește, fără degradarea funcționalității sau a vitezei de răspuns.

6.4 Aplicații practice

Sistemul are aplicabilitate largă în diverse domenii:

Monitorizarea de Mediu: Sistemul dezvoltat oferă soluții ideale pentru rețelele de stații meteo, permițând colectarea distribuită a datelor climatice pe arii extinse cu costuri reduse și întreținere minimă. Sistemele de senzori agricoli beneficiază de autonomia energetică și fiabilitatea comunicației pentru monitorizarea condițiilor de sol, umiditate și temperatură în timp real. Monitorizarea de mediu industrială poate utiliza sistemul pentru supravegherea parametrilor critici în fabrici, depozite sau alte facilități industriale unde conectivitatea tradițională este problematică.

Managementul Clădirilor: Rețelele de senzori pentru case inteligente pot fi implementate cu ușurință, oferind monitorizarea temperaturii, umidității și luminii în diverse camere fără necesitatea unor instalații complexe de cabluri. Monitorizarea clădirilor de birouri beneficiază de extensibilitatea sistemului pentru supravegherea condițiilor de mediu pe multiple etaje sau secțiuni ale clădirii. Sistemele de management energetic pot utiliza datele colectate pentru optimizarea consumului de energie și îmbunătățirea eficienței operaționale.

Aplicații de Cercetare: Colectarea datelor științifice în domenii precum meteorologia, ecologia sau agricultura poate beneficia de flexibilitatea și fiabilitatea sistemului pentru experimente pe termen lung. Proiectele IoT educaționale găsesc în acest sistem o platformă accesibilă pentru învățarea conceptelor de rețele de senzori și comunicații wireless. Platformele de dezvoltare prototip permit cercetătorilor și dezvoltatorilor să testeze și să valideze noi concepte în domeniul IoT cu un efort minimal de configurare.



6.5 Îmbunătățiri posibile

Mai multe domenii prezintă oportunități pentru dezvoltarea viitoare:

Caracteristici Îmbunătățite: Dezvoltarea viitoare a sistemului poate include suportul pentru tipuri adiționale de senzori, extinzând aplicabilitatea la domenii precum calitatea aerului, vibrațiile sau detectarea mișcării. Integrarea avansată a analizei datelor și învățării automate poate transforma datele brute colectate în insight-uri acționabile și predicții utile pentru utilizatori. Conectivitatea cloud pentru monitorizarea la distanță poate permite accesul la date de oriunde din lume și integrarea cu servicii externe. Dezvoltarea aplicațiilor mobile poate oferi o experiență de utilizare optimizată pentru dispozitivele mobile și notificări în timp real.

Îmbunătățiri de Scalabilitate: Topologiile de rețea ierarhice pot permite organizarea sistemelor cu sute sau mii de dispozitive prin crearea unor nivele multiple de management. Capacitățile de rețea mesh pot oferi redundanță și căi alternative de comunicație, crescând fiabilitatea în rețelele mari și complexe. Configurarea dinamică a rețelei poate permite adaptarea automată la schimbările din topologie și optimizarea rutelor de comunicație pentru eficiența maximă.

6.6 Limitări și considerații

Deși sistemul demonstrează avantaje semnificative, trebuie notate mai multe limitări:

Limitări Tehnice: Limitările de rază ESP-NOW în unele medii pot restringe implementarea în spații foarte mari sau cu obstacole fizice semnificative, necesitând planificarea atentă a poziționării dispozitivelor. Restricția de maximum 20 dispozitive per rețea poate fi limitativă pentru aplicații care necesită sute de senzori, deși această limitare poate fi abordată prin arhitecturi ierarhice în versiuni viitoare. Dimensiunea limitată a pachetului de 250 bytes per transmisiune necesită o planificare atentă a structurii datelor pentru aplicații care colectează volume mari de informații per ciclu de măsurare.

Considerații de Implementare: Managementul atent al adreselor MAC este necesar pentru funcționarea corectă a sistemului, necesitând o evidență precisă a dispozitivelor în rețelele mari. Factorii de mediu, cum ar fi obstacolele fizice, interferențele radio și condițiile meteorologice, pot afecta raza de comunicație și fiabilitatea transmisiei. Setup-ul inițial, deși simplificat prin interfața web, încă necesită un nivel de bază de cunoștințe tehnice pentru configurarea corectă a parametrilor de rețea și pentru înțelegerea conceptelor de bază ale sistemului.



6.7 *Ultimele cuvinte*

Sistemul de management modular pentru rețele de dispozitive IoT bazat pe ESP32 reprezintă un progres semnificativ în tehnologia rețelelor modulare distribuite. Prin utilizarea protocolului ESP-NOW, sistemul atinge eficiență energetică superioară, fiabilitatea comunicației și ușurința implementării comparativ cu abordările tradiționale, oferind în același timp flexibilitatea de a integra o gamă largă de module cu funcționalități diverse.

Interfața comprehensivă de management bazată pe web și sistemul flexibil de configurare îl fac potrivit pentru o varietate extrem de largă de aplicații - de la sisteme simple de monitorizare la ecosisteme complexe de automatizare industrială, case inteligente și soluții IoT personalizate. Natura modulară a sistemului permite integrarea armonioasă a oricărui tip de dispozitiv, de la senzori și actuatori la controllere inteligente și module personalizate.

Succesul acestui proiect demonstrează potențialul protocolului ESP-NOW pentru aplicațiile modulare IoT și oferă o fundație pentru dezvoltarea viitoare în domeniul sistemelor distribuite și al automatizării. Arhitectura modulară și implementarea open-source îl fac o platformă excelentă pentru cercetarea și dezvoltarea ulterioară în ecosistemele IoT modulare, unde diversitatea funcțională și flexibilitatea de integrare sunt esențiale pentru adoptarea pe scară largă a tehnologiilor inteligente.



7 Anexe

7.1 ANEXA A: Fișiere de cod principal folosite

Versiunea 1:

https://github.com/twdoor/ESP32-management-sistem/blob/main/main_code/main_code.ino

Versiunea 2:

https://github.com/twdoor/ESP32-management-sistem/blob/main/main_v2/main_v2.ino

Versiunea 3:

https://github.com/twdoor/ESP32-management-sistem/blob/main/main_v3/main_v3.ino

Versiunea 4:

https://github.com/twdoor/ESP32-management-sistem/blob/main/main_v4/main_v4.ino

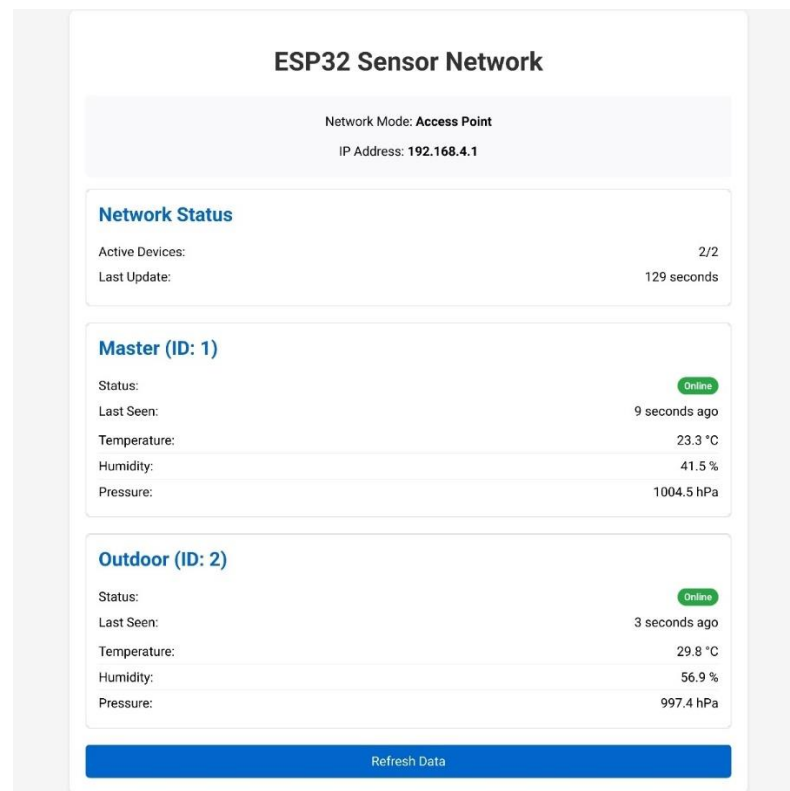
Versiunea 5:

https://github.com/twdoor/ESP32-management-sistem/blob/main/main_v5/main_v5.ino

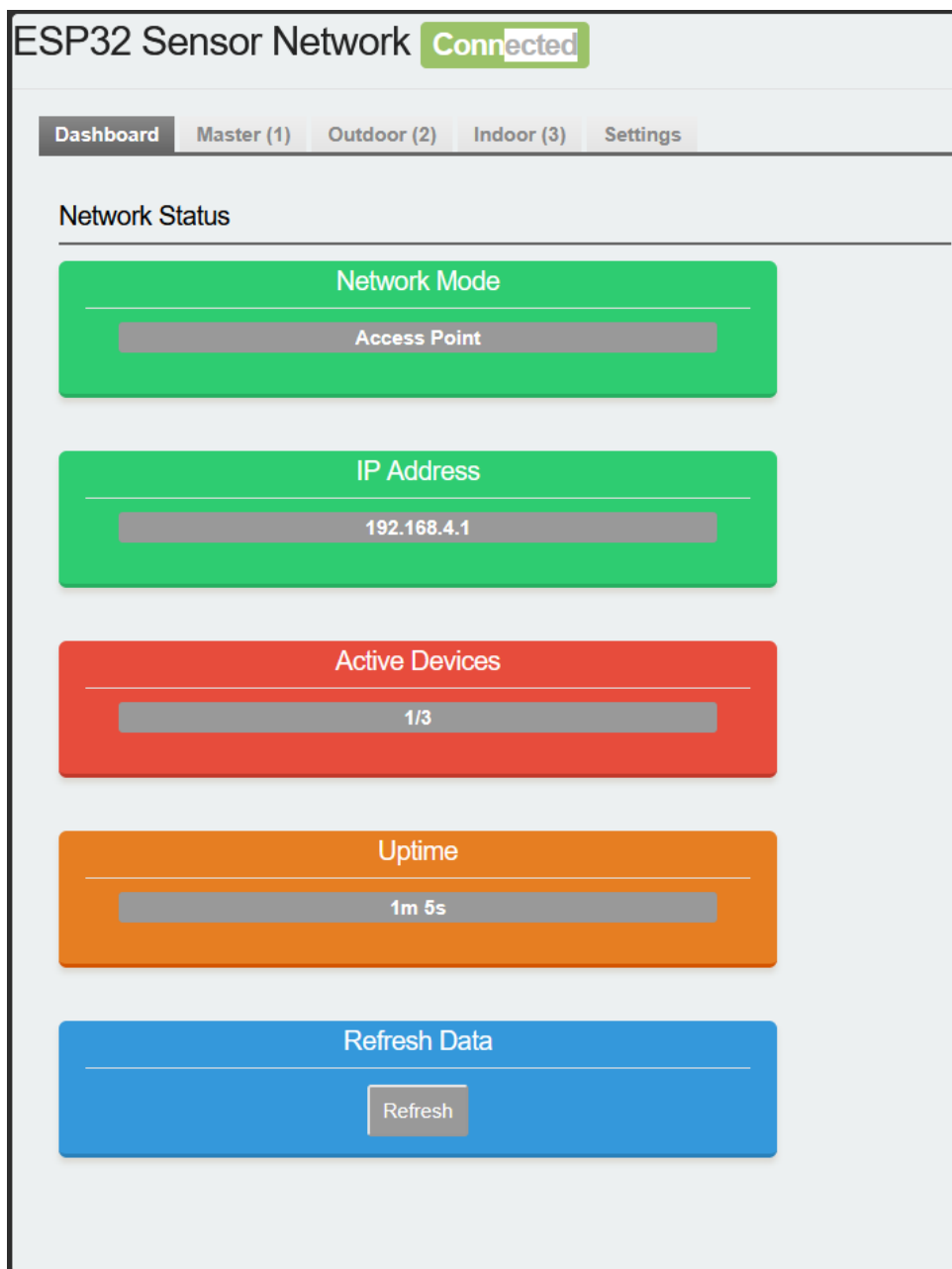
Versiunea 6 (Finală):

https://github.com/twdoor/ESP32-management-sistem/blob/main/main_v6/main_v6.ino

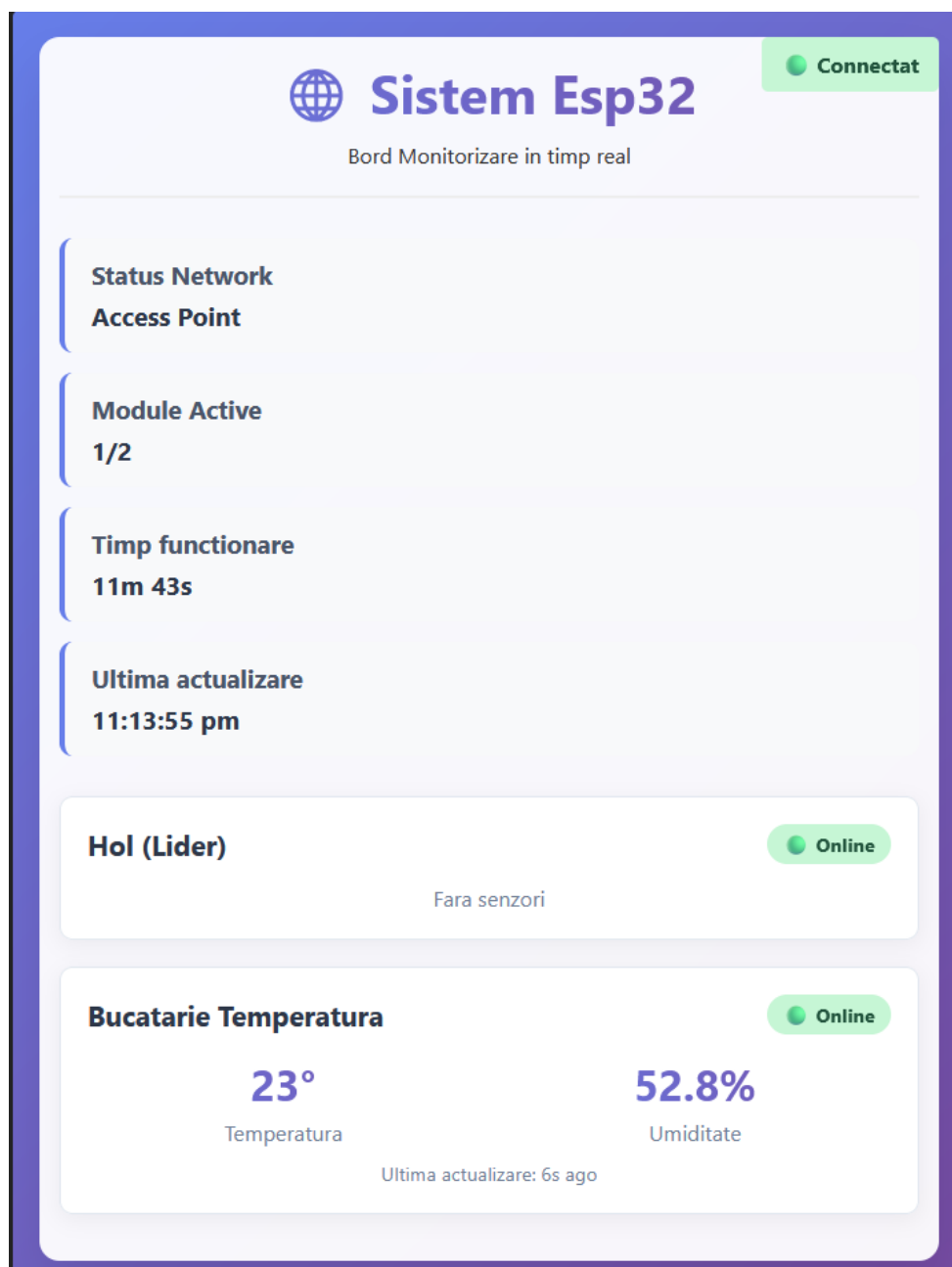
7.2 ANEXA B: Interfața web a primei versiuni




7.3 ANEXA C: Interfața web a versiunilor doi si trei



7.4 ANEXA D: Interfața web finală a modulului master



7.5 ANEXA E: Interfața web pentru configurarea unui modul



Configurare Modul

Configura modulul Esp32

**Info Modul Curent**

MAC Address: 00:00:00:00:00:00
Setup IP: 192.168.4.1

Nume Modul

Bucătărie

☐ **Modul Lider (primește date)**


Tip Modul

Temperatura si Umiditate

Adresa MAC a Liderului

D4:8A:FC:9F:2F:98

Adauga adresa MAC a modului Lider/small>

 **Salveaza si redeschide**



8 Bibliografie

[1]

adafruit/Adafruit_SSD1306. (5 iulie 2025). C++. Adafruit Industries. Data accesării: 6 iulie 2025. [Online]. Disponibil la: https://github.com/adafruit/Adafruit_SSD1306

[2]

„ESP-NOW - ESP32 - — ESP-IDF Programming Guide v5.4.2 documentation”. Data accesării: 6 iulie 2025. [Online]. Disponibil la: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/network/esp_now.html

[3]

„esp-now/User_Guide.md at master · espressif/esp-now”, GitHub. Data accesării: 6 iulie 2025. [Online]. Disponibil la: https://github.com/espressif/esp-now/blob/master/User_Guide.md

[4]

R. Santos, „ESP32 Web Server - Arduino IDE | Random Nerd Tutorials”. Data accesării: 6 iulie 2025. [Online]. Disponibil la: <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>

[5]

„ESP32Async/ESPAsyncWebServer: Async Web Server for ESP32”. Data accesării: 6 iulie 2025. [Online]. Disponibil la: <https://github.com/ESP32Async/ESPAsyncWebServer>

[6]

espressif/arduino-esp32. (6 iulie 2025). C++. Espressif Systems. Data accesării: 6 iulie 2025. [Online]. Disponibil la: <https://github.com/espressif/arduino-esp32>

[7]

S. Santos, „Getting Started with ESP-NOW (ESP32 with Arduino IDE) | Random Nerd Tutorials”. Data accesării: 6 iulie 2025. [Online]. Disponibil la: <https://randomnerdtutorials.com/esp-now-esp32-arduino-ide/>

[8]

„IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, pp. 1–4379, feb. 2021, doi: 10.1109/IEEESTD.2021.9363693.

[9]

Neil Kolban, *kolban-ESP32*. 2017. Data accesării: 6 iulie 2025. [Online]. Disponibil la: <http://archive.org/details/kolban-ESP32>



Dezvoltarea unui Sistem Modular dedicat Managementului în Aplicațiile cu Automatizare

[10]

„(PDF) Responsive Web Design and Its Impact on User Experience”, *ResearchGate*, doi: 10.48175/IJARSCT-9259.

[11]

S. Deepak, H. Anandakumar, S. Pavithra, V. Keerthika, și K. Nandhini, „Performance Analysis of Star Topology for Small Networks Using Riverbed”, în *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, mar. 2022, pp. 2108–2111. doi: 10.1109/ICACCS54159.2022.9785012.

[12]

L. Bachschwell, *s00500/ESPUI*. (3 iulie 2025). C++. Data accesării: 6 iulie 2025. [Online]. Disponibil la: <https://github.com/s00500/ESPUI>

[13]

T. Bray, „The JavaScript Object Notation (JSON) Data Interchange Format”, Internet Engineering Task Force, Request for Comments RFC 8259, dec. 2017. doi: 10.17487/RFC8259.

[14]

A. Melnikov și I. Fette, „The WebSocket Protocol”, Internet Engineering Task Force, Request for Comments RFC 6455, dec. 2011. doi: 10.17487/RFC6455.

