КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ РАДІОФІЗИКИ, ЕЛЕКТРОНІКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ
Кафедра радіотехніки та радіоелектронних систем

«На правах рукопису»

Робота допущена до захисту в ЕК
рішенням кафедри радіотехніки та радіоелектронних систем
від _____ 2025 року, протокол № ____.
Завідувач кафедри доктор фіз.-мат. наук, професор
_____ Ігор АНІСІМОВ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
на тему:
«WI-FI - АНАЛІЗАТОР»

Виконав:
студент 4-го курсу
денної форми навчання
спеціальності 172 – Електронні комунікації та радіотехніка
ОПП «Інформаційна безпека телекомунікаційних систем і мереж»
Шапошник Олександр Ігорович

Науковий керівник:
доцент кафедри радіотехніки
та радіоелектронних систем
Жиров Генадій Борисович _____

Рецензент:
Старший науковий співробітник
науково дослідного центру ВІКНУ
Мирослав Олександрович Коваль _____

Засвідчую, що у цій бакалаврській роботі
немає запозичень з праць інших авторів без
відповідних посилань
студент Шапошник Олександр

Київ – 2025

РЕФЕРАТ

Дипломна робота: 27 с., 2 формули, 5 рис., 10 джерел.

Об’єкт дослідження: процеси моніторингу та аналізу параметрів бездротових локальних мереж (WLAN) стандарту IEEE 802.11 за допомогою портативних вбудованих систем.

Мета проекту: реалізація автономного Wi-Fi сканера на базі мікроконтролерної платформи ESP32. Дане дослідження охоплює розробку апаратної архітектури пристрою, створення програмного забезпечення для сканування радіоефіру, інформацію про отримання та обробку даних щодо точки доступу, а також імплементацію алгоритму для орієнтовної оцінки відстані до джерела сигналу на основі показника рівня прийнятого сигналу (RSSI) і розробку інтерфейсу користувача з метою подальшої візуалізації отриманої інформації на OLED-дисплеї.

Методи: дослідження моделі об’єкту з використанням комп’ютерного моделювання, мікроконтролерного програмування, обробки цифрових сигналів, об’єктно-орієнтованого програмування на мовах C++ та Arduino (C-подібний синтаксис).

У процесі розробки реалізовано наступні задачі:

- сканування доступних Wi-Fi мереж з виведенням SSID, MAC-адреси, типу захисту, каналу, рівня сигналу (RSSI) та орієнтовної відстані до точки доступу;
- обрахунок відстані до Wi-Fi точки доступу на основі потужності сигналу (RSSI);

виведення даних щодо кожної мережі на окремій сторінці OLED-дисплея з подальшим відображенням номеру сторінки;

- перемикання між сторінками за допомогою кнопки;
- функціональність періодичного переміщення з метою знаходження певних змін у довколишньому радіоефірі.

Під час дослідження були розглянуті можливості вбудованої багатозадачності ESP32 й оптимізації процесу цифрової обробки сигналів в режимі реального часу на обмежених обчислювальних ресурсах.

ЗМІСТ

ВСТУП

З огляду стрімкого розвитку й загального впровадження бездротових локальних мереж (WLAN) стандарту IEEE 802.11, постає нагальна потреба у розробці доступних, а також ефективних засобів задля їхньої діагностики, моніторингу і оптимізації. Професійні програмно-апаратні комплекси, які наразі представлені на ринку, є занадто високовартісними та надлишковими в питанні вирішення більшості нагальних повсякденних інженерних задач. В той же час, і мобільні застосунки зазвичай мають обмежений функціонал, а, отже, не забезпечують належної точності вимірювань. З огляду на це, питання розробки портативного автономного пристрою для аналізу Wi-Fi мереж, що об'єднує низьку собівартість з достатньою функціональністю, є назрілою та актуальною. Таким чином, об’єктом дипломної роботи є саме такий пристрій, який реалізований на базі мікроконтролера ESP32.

Практична значущість пристрою полягає в реалізації найважливіших основних функцій: від сканування радіоефіру задля отримання вичерпної інформації щодо точок доступу (SSID, MAC, RSSI, каналу, типу безпеки) до імплементації алгоритму для орієнтовної оцінки відстані на основі рівня прийнятого сигналу (це є важливим щодо планування топології мережі й локалізації джерел сигналу). Запроваджений ергономічний інтерфейс із посторінковим виводом даних та апаратним керуванням підвищує зручність використання пристрою при натурних випробуваннях, а функція періодичного переміщення дозволяє здійснювати моніторинг динамічних змін у радіопросторі. Отже, розроблений Wi-Fi сканер є сукупним комплексним рішенням, що становить практичний жвавий інтерес для інженерно-технічних спеціалістів з обслуговування мереж. Дана розробка також має значну освітню цінність як приклад застосування вбудованих систем задля вирішення прикладних нагальних задач в галузі телекомунікацій.

1. ПРИНЦИПИ РОБОТИ БЕЗДРОТОВИХ МЕРЕЖ WI-FI

1.1. Основи Wi-Fi-зв'язку (IEEE 802.11)

Бездротова технологія, яка загальновідома під комерційною назвою Wi-Fi, базується на сімействі стандартів IEEE 802.11 [1], які розроблені Інститутом інженерів з електротехніки та електроніки (IEEE). Зазначені стандарти визначають функціонування бездротових локальних мереж (WLAN) на фізичному (PHY) й канальному (MAC) рівнях моделі OSI. Пристрій, що спроєктований в даній роботі, допомагає в розумінні та інтерпретації базових принципів роботи засобів аналізу та моніторингу.

Фізичний рівень (PHY) призначений для перетворення цифрових даних (бітів) у радіосигнали задля безпровідної передачі в прямому та зворотньому напрямках. Робота Wi-Fi мереж здійснюється у неліцензованих частотних діапазонах, переважно 2.4 ГГц та 5 ГГц, а із запровадженням стандарту Wi-Fi 6E (802.11ax) – і в діапазоні 6 ГГц. Кожен діапазон поділено на певну кількість каналів – вузьких смуг частот, на яких може впроваджуватись передача даних. Використання різних каналів дозволяє мінімізувати процес взаємної інтерференції поміж сусідніми мережами. Значущою характеристикою на даному рівні є показник рівня прийнятого сигналу (RSSI – Received Signal Strength Indicator). RSSI є відносною величиною, що вимірюється в децибел-міліватах (дБм), і характеризує потужність сигналу, отриманого приймачем. Даний параметр має логарифмічну залежність, він є ключовим задля оцінки якості з'єднання та надалі буде представлений задля орієнтовної локалізації джерела сигналу, оскільки значення його зменшується зі збільшенням відстані й наявності певних перешкод.

Канальний рівень, а саме його підрівень керування доступом до середовища (MAC – Media Access Control), відповідає за організацію доступу до спільного радіосередовища, а також адресацію пристроїв в межах однієї мережі. Будь-який мережевий адаптер має унікальний 48-бітний ідентифікатор – MAC-адресу, що використовується задля ідентифікації відправника і одержувача на цьому рівні. Основою організації мережі є базовий набір служб (BSS – Basic

Service Set), який складається з точки доступу (AP – Access Point) та клієнтських станцій. Кожен BSS розпізнається унікальним ідентифікатором BSSID, яким здебільшого є MAC-адреса тієї самої точки доступу. Задля зручності користувачів кожній мережі надається текстова назва – ідентифікатор набору служб (SSID – Service Set Identifier). Інформація про мережу, включно з її SSID, BSSID, робочим каналом та типом безпеки, періодично транслюється точкою доступу у професійних службових пакетах – кадрах-маяках (Beacon Frames). Якраз перехоплення та аналіз цих кадрів є основним принципом роботи будь-якого Wi-Fi сканера. Цей принцип є провідним і в пристрої, що розробляється.

Також можна зазначити, що стандарт IEEE 802.11 визначає механізми захисту даних, що передаються. З точки зору історії, першим був протокол WEP, який сьогодні є надто вразливим. Новітніми та рекомендованими до використання є протоколи WPA2 (Wi-Fi Protected Access 2) та його оновлена версія WPA3, що використовують надійні алгоритми шифрування, а саме: AES (Advanced Encryption Standard). Ідентифікація типу безпеки є однією з базових функцій сканера, оскільки дозволяє визначити і кваліфікувати рівень захищеності мережі.

1.2. Параметри бездротових мереж: SSID, MAC-адреса, RSSI, канал, тип шифрування

Задля виконання та реалізації комплексного аналізу бездротової мережі, сканер повинен отримувати та інтерпретувати набір ключових параметрів, що детально характеризують кожну виявлену точку доступу. Дані параметри передаються у службових кадрах (Beacon Frames), вони і є основою для подальшої діагностики та оптимізації. Отже, перейдемо до розгляду основних з них:

SSID (Service Set Identifier – Ідентифікатор набору служб) – це відкрите, символьне найменування бездротової мережі, що складається з послідовності до 32 символів. Його основне призначення – надати користувачам можливість безперешкодно ідентифікувати та розрізняти доступні мережі. Хоча SSID і є основним ідентифікатором для кінцевого користувача, він не є унікальним і

може бути однаковим для різних фізичних мереж, або прихованим (коли поле SSID у кадрі-маяку залишається порожнім), що є елементарним заходом безпеки.

MAC-адреса (Media Access Control Address) – це унікальний 48-бітний апаратний ідентифікатор, що надається кожному мережевому пристрою на етапі виробництва. У контексті Wi-Fi мереж MAC-адреса точки доступу відіграє роль BSSID (Basic Service Set Identifier). На відміну від SSID, BSSID є унікальним ідентифікатором конкретної точки доступу на канальному рівні. Саме BSSID дозволяє однозначно розрізняти мережі, навіть якщо вони мають однакові назви (SSID), що є типовим для великих корпоративних мереж з кількома точками доступу.

RSSI (Received Signal Strength Indicator – показник рівня потужності прийнятого сигналу) – це відносна міра потужності сигналу, який приймає антена сканера від точки доступу. RSSI вимірюється в децибел-міліватах (дБм) і є логарифмічною величиною з від'ємними значеннями (наприклад, від -30 дБм – чудовий сигнал, до -90 дБм – дуже слабкий сигнал). Цей параметр є ключовим індикатором якості зв'язку та фізичної відстані до точки доступу. На основі моделі загасання сигналу у вільному просторі (Free-Space Path Loss) можна побудувати математичну залежність між RSSI та відстанню, яка є однією з центральних функцій пристрою, що розробляється.

Канал (Channel) – є визначеною смугою частот у межах радіодіапазону (2.4 ГГц або 5 ГГц), на якій точка доступу здійснює передачу даних. У діапазоні 2.4 ГГц доступно 11-14 каналів (залежно від регіону), проте лише деякі з них, до прикладу, 1, 6 та 11, не перетинаються між собою. Інформація про зайняті канали є критично важливою для оптимізації мережі, оскільки вибір найменш завантаженого каналу дозволяє значною мірою зменшити інтерференцію та підвищити стабільність й швидкість з'єднання.

Тип шифрування (Encryption Type) – це метод, що використовується для захисту даних, які передаються через бездротову мережу. Сканер ідентифікує використовуваний протокол безпеки, який дозволяє миттєво визначити рівень захищеності мережі. Основні типи містять в собі Open (відсутність

шифрування), WEP (застарілий і вразливий протокол), а також сучасні та надійні стандарти WPA/WPA2/WPA3, які використовують стійкі алгоритми шифрування, такі як AES.

1.3. Алгоритми оцінки відстані за RSSI

Однією з найважливіших функціональних можливостей пристрою, що розробляється, є здатність надавати орієнтовну оцінку відстані до точки доступу. Ця функція ґрунтується на фундаментальному фізичному принципі: потужність радіосигналу зменшується зі збільшенням відстані від джерела. Показник RSSI, бувши безпосередньою характеристикою цієї потужності, може бути використаний як вхідний параметр для математичних моделей, які описують це загасання.

Найбільш поширеною моделлю щодо таких розрахунків є логарифмічна модель втрат сигналу при поширенні (Log-Distance Path Loss Model). Дана модель зважає, що втрати потужності сигналу у вільному просторі мають логарифмічну залежність від відстані. [2]

$$RSSI = -10 * n * \log10(d) + A \quad (1.1)$$

де:

- RSSI – виміряне значення рівня потужності прийнятого сигналу в дБм;
- d – відстань від передавача (точки доступу) до приймача (сканера) в метрах;
- A – опорне значення RSSI, тобто потужність сигналу, виміряна на фіксованій референтній відстані, зазвичай 1 метр. Цей параметр є емпіричною константою, що залежить від потужності передавача точки доступу та характеристик антени, і визначається експериментально;
- n – коефіцієнт загасання середовища (path-loss exponent), який характеризує швидкість втрати потужності сигналу при його поширенні в конкретному середовищі. Значення коефіцієнту залежить від умов: для вільного простору $n \approx 2$; для приміщень з перешкодами (стіни, меблі) n може варіюватися в діапазоні від 3 до 4 і вище.

Задля практичного застосування в нашому пристрої дану формулу маємо перетворити для розрахунку відстані d:

$$d = 10^{((A - RSSI) / (10 * n))} \quad (1.2)$$

Незважаючи на свою теоретичну обґрунтованість, метод оцінки відстані за RSSI має певні суттєві обмеження, через які результат варто розглядати як приблизний, а не точний вимір:

- у реальних умовах радіосигнал досягає приймача не лише прямим шляхом, а й через численні відбиття від стін, стелі, підлоги та інших об'єктів. Це призводить до інтерференції (конструктивної та деструктивної), що викликає значні та непередбачувані коливання RSSI, які не пов'язані зі зміною відстані;
- фізичні об'єкти на шляху сигналу спричиняють значне загасання, яке не враховується простою моделлю з єдиним коефіцієнтом n;
- розташування антен передавача та приймача в просторі суттєво впливає на рівень прийнятого сигналу;
- робота інших бездротових пристроїв в тому ж частотному діапазоні може істотно змінювати вимірювання RSSI.

РОЗДІЛ 2. АРХІТЕКТУРА ТА КОМПОНЕНТИ СИСТЕМИ

2.1. Принципова схема пристрою

Для реалізації пристрою аналізу Wi-Fi мереж було використано мікроконтролер ESP32-C. В основі апаратної частини лежить схема підключення цього контролера з урахуванням необхідного живлення, підключення антени, дисплея та елементів керування. Живлення подається на плату з джерела 3.3 В, радіочастотна частина включає в себе антену, підключену через узгоджувальний ланцюг, що складається з конденсаторів та індуктивностей. Це дозволяє коректно працювати з бездротовими мережами у діапазоні 2.4 ГГц. Підключення дисплея здійснюється через інтерфейс I²C, при цьому сигнальні лінії SCL та SDA виведено на виводи GPIO15 та GPIO16 відповідно. Сам дисплей — OLED із роздільною здатністю 128×64 пікселі, на якому відображається інформація про знайдені Wi-Fi мережі. Для зручності користувача реалізовано перемикання між сторінками результатів сканування за допомогою двох кнопок, які підключені до пінів GPIO4 і GPIO5 (див. рис. 2.1).

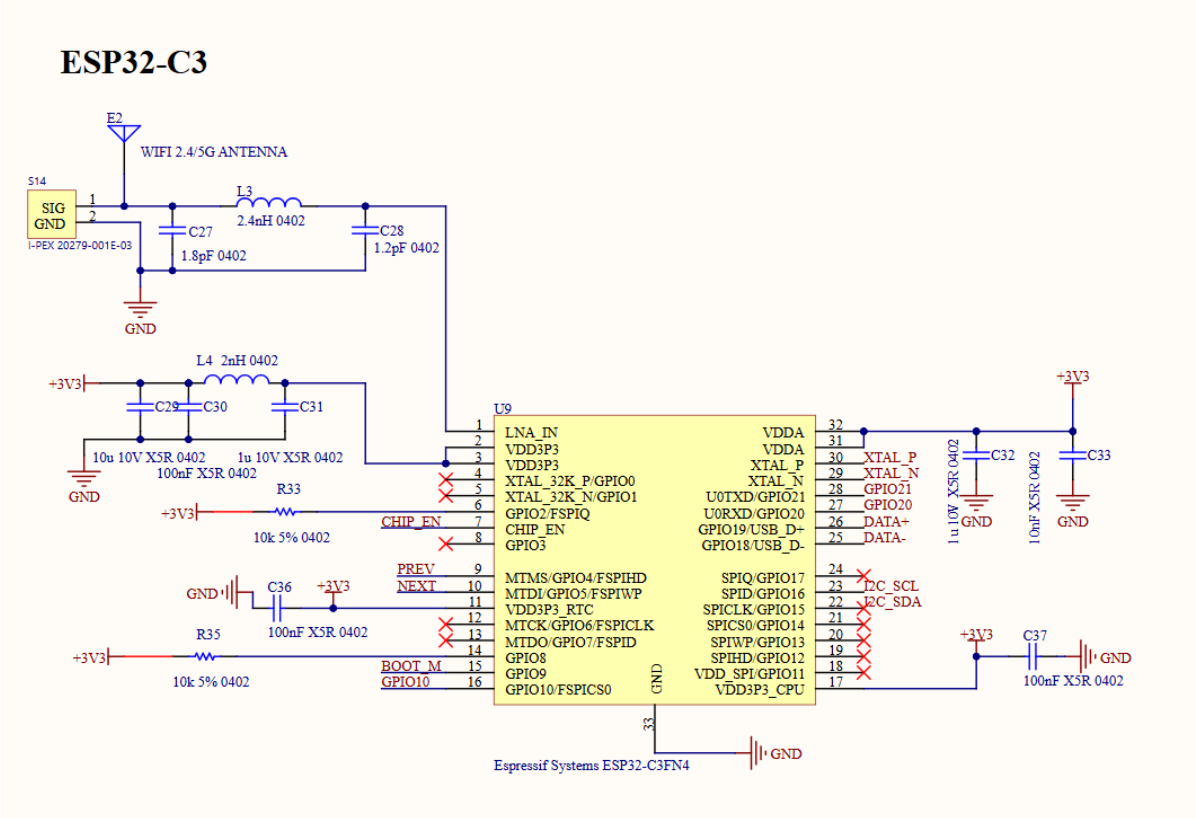

Рис. 2.1. Принципова схема пристрою

2.2. Використане обладнання

ESP32

Основою апаратної частини пристрою, що розробляється, є мікроконтролерна платформа ESP32 від компанії Espressif Systems. Вибір саме цього компонента не є випадковим і обґрунтовується низкою його технічних характеристик, які оптимально відповідають вимогам даного проєкту. ESP32 є потужною системою на кристалі (SoC), що поєднує високу продуктивність, чималий набір периферійних модулів та низьке енергоспоживання, що робить систему ідеальною для створення портативних пристроїв мережевої діагностики.

Основною перевагою ESP32 для даної задачі є інтегрований модуль Wi-Fi, що підтримує стандарти IEEE 802.11 [3]. Наявність вбудованого радіомодуля з повним стеком протоколів TCP/IP суттєво спрощує апаратну реалізацію, усуваючи необхідність у використанні зовнішніх Wi-Fi-шилдів, а це зменшує габарити, вартість та складність кінцевого пристрою. Вбудовані програмні бібліотеки дозволяють безперешкодно реалізувати функціонал сканування мереж, отримуючи доступ до всіх необхідних параметрів, таких, як: SSID, BSSID, RSSI, канал та тип шифрування.

Висока продуктивність забезпечується двоядерною архітектурою Tensilica Xtensa LX6 з тактовою частотою до 240 МГц. Це дозволяє ефективно розподіляти наступні задачі: одне ядро може бути виділене для управління Wi-Fi стеком та виконання інтенсивних операцій сканування радіоефіру, тоді, як друге ядро може паралельно займатися обробкою даних, розрахунком відстані, оновленням інформації на дисплеї, а також обробкою натискань кнопок. Подібний підхід забезпечує високу швидкість реакції інтерфейсу користувача та стабільну роботу пристрою без всіляких затримок.

Для взаємодії із зовнішніми компонентами, такими, як: OLED-дисплей та кнопка управління, ESP32 надає широкий набір периферійних інтерфейсів. Задля підключення дисплея використовуються базові стандартні послідовні інтерфейси, такі, як: I^2C або SPI, - для яких мікроконтролер має апаратну підтримку. Кнопка підключається до одного з багаточисельних контактів

загального призначення (GPIO), які можуть бути налаштовані як і на вхід, так і на вихід.

Нарешті, важливим фактором є енергоефективність та наявність кількох режимів зниженого енергоспоживання, що є критичним для портативного пристрою з автономним живленням. Хоча в активному режимі сканування споживання є значним, для майбутніх версій пристрою ці режими можуть бути використані для подовження часу роботи від акумулятора.

OLED-дисплей (0.96", SSD1306) [4]

Для забезпечення ефективної взаємодії користувача з пристроєм та наочного представлення результатів сканування, було обрано монохромний OLED-дисплей з діагоналлю 0.96 дюйма та роздільною здатністю 128x64 пікселів, керований драйвером SSD1306. Вибір даного компонента для візуалізації інформації обумовлений його технічними характеристиками, які ідеально відповідають вимогам портативного діагностичного пристрою.

По-перше, технологія OLED забезпечує ключові переваги над традиційними LCD-дисплеями. Оскільки кожен піксель є самостійним джерелом світла, OLED-дисплеї не потребують зовнішньої підсвітки. Це призводить до двох важливих наслідків:

Вимкнений піксель не випромінює світла, що створює ідеально глибокий чорний колір і, як наслідок, надзвичайно високу контрастність зображення. Це забезпечує чудову читабельність тексту навіть за умов яскравого навколишнього освітлення.

Споживання енергії прямо пропорційне кількості та яскравості увімкнених пікселів. Для інтерфейсу, де значна частина екрана залишається чорною, це дозволяє суттєво знизити загальне енергоспоживання, що є критичним фактором для пристрою з автономним живленням.

По-друге, компактні розміри (0.96 дюйма) та достатня роздільна здатність (128x64) є оптимальним співвідношенням для портативного пристрою. Такі габарити дозволяють створити ергономічний та кишеньковий корпус, а

роздільної здатності цілком достатньо для чіткого відображення кількох рядків тексту, що є основним форматом виводу даних у даному проєкті.

По-третє, драйвер SSD1306 є одним з найпоширеніших для OLED-дисплеїв малого розміру. Його широка розповсюдженість забезпечує наявність великої кількості готових програмних бібліотек для платформи ESP32. Це значно спрощує процес програмної інтеграції, скорочує час розробки та дозволяє зосередитись на логіці роботи самого застосунку, а не на низькорівневому програмуванні графічного контролера.

Нарешті, підключення дисплея здійснюється через інтерфейс I²C, який вимагає лише двох ліній даних (SDA та SCL) окрім живлення. Це дозволяє мінімізувати використання контактів загального призначення (GPIO) мікроконтролера ESP32, залишаючи їх вільними для підключення інших периферійних пристроїв, таких як кнопки керування.

2.3. Програмне забезпечення

Arduino IDE [5]

Для написання прошивки для мікроконтролера ESP32 я обрав середовище розробки Arduino IDE. Цей вибір був зумовлений кількома важливими причинами: зручністю у використанні, швидкістю розробки та достатньо широким функціоналом — що особливо важливо під час роботи над дипломним проєктом.

Перш за все, Arduino IDE спрощує взаємодію з апаратною частиною. Завдяки використанню бібліотек і зрозумілих функцій, мені не потрібно працювати з низькорівневим кодом і регістрами. Наприклад, щоб налаштувати пін або передати дані по I²C чи SPI, достатньо викликати просту функцію. Це дозволило мені сконцентруватися на головній логіці програми, а не витрачати час на технічні деталі.

Ще одна перевага — це легке підключення підтримки ESP32. Через вбудований менеджер плат я швидко додав офіційний пакет "ESP32 Core for

Arduino", який відкриває доступ до всіх можливостей цієї платформи, включаючи сканування Wi-Fi мереж прямо з середовища Arduino.

Також Arduino має величезну спільноту і багато готових бібліотек, що значно полегшує життя. Наприклад, для роботи з OLED-дисплеєм я використав бібліотеки Adafruit_GFX та Adafruit_SSD1306, які дуже легко підключаються через менеджер бібліотек і добре документовані.

І нарешті, структура програм в Arduino досить проста і логічна: вся ініціалізація відбувається в setup(), а основний код постійно виконується в loop(). У моєму випадку setup() відповідає за налаштування дисплея, серійного порту, Wi-Fi та кнопки, а loop() — за зміну сторінок при натисканні кнопки, оновлення інформації на екрані та періодичне сканування мереж.

Бібліотеки: WiFi.h, Adafruit_SSD1306.h, Wire.h

Ефективна та швидка розробка програмного забезпечення для пристрою стала можливою завдяки використанню стандартних та спеціалізованих бібліотек в середовищі Arduino IDE [6]. Ці бібліотеки надають високорівневий API для взаємодії з апаратною частиною, абстрагуючи розробника від складності низькорівневих протоколів та регістрів. Для реалізації функціоналу Wi-Fi сканера було залучено наступний набір ключових бібліотек.

WiFi.h

Ця бібліотека є стандартною частиною пакета "ESP32 Core for Arduino" і надає повний доступ до функціоналу вбудованого Wi-Fi модуля. Вона є фундаментальною для даного проєкту, оскільки саме через неї реалізовано основну задачу – сканування радіоефіру. У ході роботи були використані наступні її функції:

- WiFi.mode(WIFI_STA): Переводить Wi-Fi модуль у режим станції, що є необхідною умовою для ініціації сканування мереж.

- WiFi.scanNetworks(): Запускає процес активного сканування та повертає кількість виявлених точок доступу.

- Функції для отримання даних:

WiFi.SSID(i), WiFi.BSSIDstr(i), WiFi.RSSI(i), WiFi.channel(i) та WiFi.encryptionType(i), де i – індекс виявленої мережі. Ці функції дозволяють отримати всі необхідні параметри для подальшої обробки та візуалізації.

Wire.h

Стандартна бібліотека Arduino для роботи з послідовним інтерфейсом I²C (Inter-Integrated Circuit). Хоча вона не використовується безпосередньо для виводу графіки, її підключення є обов'язковою вимогою для роботи бібліотеки керування OLED-дисплеєм, оскільки саме Wire.h забезпечує низькорівневу передачу даних між мікроконтролером ESP32 та драйвером дисплея SSD1306 по шині I²C. Основна функція, що використовується – Wire.begin(), яка ініціалізує I²C-з'єднання.

Adafruit_SSD1306.h та **Adafruit_GFX.h** [7]

Цей комплекс з двох бібліотек є де-факто стандартом для роботи з монохромними OLED-дисплеями на базі контролера SSD1306.

- Adafruit_GFX.h є базовою графічною бібліотекою, що надає набір універсальних функцій для роботи з графічними примітивами: виведення тексту, малювання ліній, кіл, прямокутників тощо.

- Adafruit_SSD1306.h є апаратно-залежною надбудовою, яка "навчає" базову бібліотеку GFX працювати конкретно з дисплеєм SSD1306. Вона ініціалізує дисплей та реалізує функцію відправки буфера зображення на екран.

У проєкті використовуються ключові функції цих бібліотек: display.begin() для ініціалізації, display.clearDisplay() для очищення екрана перед оновленням сторінки, display.setTextSize(), display.setTextColor(), display.setCursor() для налаштування параметрів шрифту та позиціонування, display.println() для виводу текстових даних, і, найголовніше, display.display() для фізичного оновлення зображення на екрані.

РОЗДІЛ 3. ОПИС РОБОТИ ПРИСТРОЮ

3.1. Сканування доступних Wi-Fi мереж

Логіка процесу сканування реалізована наступним чином:

Ініціалізація модуля Wi-Fi. При першому запуску пристрою, у програмному блоці setup(), виконується ініціалізація Wi-Fi модуля. За допомогою команди WiFi.mode(WIFI_STA) мікроконтролер переводиться в режим станції (Station Mode). Цей режим є обов'язковою умовою, оскільки саме в ньому ESP32 може виконувати пошук доступних мереж, імітуючи поведінку клієнтського пристрою.

Запуск сканування. Процес сканування ініціюється викликом функції WiFi.scanNetworks(). Ця функція є блокуючою, тобто виконання основного коду призупиняється на час, необхідний модулю для надсилання запитів (Probe Requests) на всіх каналах та прослуховування відповідей від точок доступу (Probe Responses) та їхніх кадрів-маяків (Beacon Frames). По завершенню, функція повертає ціле число – кількість виявлених унікальних мереж.

Збір та структурування даних. Після отримання позитивної відповіді про кількість знайдених мереж (N > 0), організовується цикл, що ітерується від 0 до N-1. На кожній ітерації для мережі з індексом i за допомогою відповідних функцій бібліотеки WiFi.h зчитуються її параметри:

SSID: WiFi.SSID(i)

BSSID (MAC-адреса): WiFi.BSSIDstr(i)

RSSI: WiFi.RSSI(i)

Робочий канал: WiFi.channel(i)

Тип шифрування: WiFi.encryptionType(i)

З метою ефективного зберігання та подальшої маніпуляції, отримані дані для кожної мережі агрегуються та зберігаються в масиві спеціалізованих структур даних (struct).

Сортування результатів. Для підвищення зручності аналізу, після збору даних масив виявлених мереж сортується. В якості основного критерію

сортування було обрано показник RSSI. Сортування виконується за спаданням цього значення, що забезпечує пріоритетне відображення найближчих або найпотужніших мереж на перших сторінках інтерфейсу.

3.2. Отримання параметрів мережі

Після завершення процесу сканування, програмне забезпечення отримує доступ до масиву результатів. На цьому етапі відбувається безпосереднє зчитування, обробка та форматування даних для кожної виявленої мережі перед їх збереженням та подальшим виводом на дисплей. Цей процес є критично важливим для перетворення "сирих" даних, що надаються драйвером, у зрозумілу для користувача інформацію.

SSID: Ідентифікатор мережі зчитується за допомогою функції WiFi.SSID(i). Вона повертає об'єкт типу String, який безпосередньо виводиться на екран. Окремо обробляється випадок так званих "прихованих мереж", коли поле SSID у кадрі-маяку є порожнім. У такій ситуації програма виводить спеціальний текстовий маркер, наприклад, [Hidden Network], для інформування користувача.

MAC-адреса (BSSID): Для отримання цього унікального ідентифікатора точки доступу використовується функція WiFi.BSSIDstr(i). Перевага даної функції полягає в тому, що вона одразу повертає MAC-адресу у вигляді відформатованого рядка стандартного вигляду (наприклад, DE:AD:BE:EF:01:23), що усуває необхідність у додатковому коді для перетворення 6-байтного масиву в шістнадцятковий текстовий формат.

RSSI: Рівень потужності сигналу отримується викликом WiFi.RSSI(i). Функція повертає цілочисельне значення в дБм. Це значення використовується у двох ключових аспектах: по-перше, воно безпосередньо виводиться на екран як основний індикатор якості сигналу; по-друге, воно слугує вхідним параметром для математичної моделі оцінки відстані, описаної в розділі 1.3.

Канал: Номер робочого каналу зчитується за допомогою функції WiFi.channel(i). Вона повертає ціле число, яке напряму відповідає

номеру каналу (наприклад, 1, 6, 11). Це значення не потребує додаткової обробки і готове до відображення.

Тип захисту: Ідентифікація протоколу безпеки є найбільш комплексним процесом з точки зору інтерпретації. Функція WiFi.encryptionType(i) повертає не текстовий рядок, а значення перелічуваного типу (enum) wifi_auth_mode_t. Для перетворення цього службового ідентифікатора у зрозумілий користувачеві формат, у програмному коді реалізовано спеціальну функцію-обробник. Ця функція використовує конструкцію switch-case для зіставлення кожного можливого значення enum з відповідним текстовим описом.

WIFI_AUTH_OPEN перетворюється на рядок "Open".

WIFI_AUTH_WEP на "WEP".

WIFI_AUTH_WPA_PSK на "WPA-PSK".

WIFI_AUTH_WPA2_PSK на "WPA2-PSK".

WIFI_AUTH_WPA_WPA2_PSK на "WPA/WPA2".

3.3. Основна структура програми

Програмний код пристрою розроблено в середовищі Arduino IDE та побудовано на базі стандартної архітектури, що складається з двох основних блоків: setup() та loop().

Функція setup() виконується одноразово при старті системи та відповідає за повну ініціалізацію апаратних та програмних компонентів. У цьому блоці послідовно виконуються наступні дії: ініціалізація послідовного порту для відлагодження, налаштування пінів кнопок (SCROLL_BUTTON_PIN, RESCAN_BUTTON_PIN) у режим входу з активацією внутрішнього підтягуючого резистора (INPUT_PULLUP), ініціалізація OLED-дисплея через шину I^2C. Важливим етапом є виклик функції displayIntroScreen(), що виводить на екран стартову заставку, та негайний виклик функції scanNetworks() для виконання первинного сканування. Це забезпечує наявність актуальних даних одразу після завантаження пристрою.

Функція loop() є нескінченним циклом, що містить основну логіку роботи пристрою. Її структура реалізована у вигляді простого скінченного автомату, що

керується логічним прапором rescanRequested. У нормальному стані (rescanRequested == false) цикл постійно викликає функцію displayNetworkInfo(), що оновлює екран даними про поточну обрану мережу. Якщо ж користувач ініціює пересканування, прапор встановлюється у true, і логіка циклу перемикається на виконання процедури зворотного відліку та подальшого сканування. Також на кожній ітерації циклу перевіряється стан кнопок керування, що забезпечує високу швидкість реакції інтерфейсу. Використання глобальних змінних (currentNetwork, numNetworks, currentPage) дозволяє ефективно керувати станом програми та передавати дані між різними функціями.

3.4. Реалізація циклічного сканування

На відміну від повністю автоматичного періодичного оновлення, в даній реалізації процес сканування є ініційованим користувачем, що дозволяє уникнути непотрібних витрат ресурсів та енергії. Логіка цього процесу реалізована наступним чином:

Ініціація сканування: Користувач натискає на кнопку, підключену до RESCAN_BUTTON_PIN. Це натискання детектується у головному циклі loop().

Встановлення прапора: При підтвердженому натисканні та за умови, що попередній запит на сканування ще не виконується (!rescanRequested), встановлюється логічний прапор rescanRequested = true. Одночасно за допомогою функції millis() фіксується час початку запиту (rescanStartTime = millis()) [8].

Зворотний відлік: Поки прапор rescanRequested встановлений, головний цикл loop() переходить у режим очікування. На екран виводиться інформація про майбутнє сканування та таймер зворотного відліку, реалізований у функції displayCountdown(). Тривалість відліку задається константою COUNTDOWN_DURATION. Цей етап забезпечує візуальний зворотний зв'язок для користувача.

Виконання сканування: Після того, як час, що пройшов з моменту запиту (millis() - rescanStartTime), перевищить COUNTDOWN_DURATION,

викликається функція scanNetworks(). Вона виконує системний виклик WiFi.scanNetworks(), що повертає нову кількість знайдених мереж, та скидає індекси поточної мережі і сторінки на початкові значення.

Скидання прапора: Одразу після завершення сканування прапор rescanRequested скидається у false, що повертає програму до основного режиму відображення інформації.

Такий підхід забезпечує повний контроль користувача над процесом оновлення даних та є ефективним з точки зору управління ресурсами пристрою.

3.5. Програмна реалізація перемикання сторінок

Навігація між сторінками з інформацією про виявлені мережі реалізована за допомогою кнопки, підключеної до SCROLL_BUTTON_PIN. Обробка її натискання відбувається в головному циклі loop() і включає наступні етапи[9]:

Програмний анти-дребезг (Debouncing): Для запобігання хибним багаторазовим спрацюванням, викликаним механічним брязкотом контактів, реалізовано програмний фільтр. Перевірка стану кнопки відбувається лише тоді, коли з моменту останнього зареєстрованого натискання пройшов час, більший за BUTTON_DEBOUNCE_DELAY. При фіксації натискання час lastButtonPressTime оновлюється.

Детектування натискання: Оскільки пін налаштовано з внутрішнім підтягуючим резистором (INPUT_PULLUP), натискання кнопки відповідає низькому логічному рівню (digitalRead(SCROLL_BUTTON_PIN) == LOW).

Циклічне перемикання: При виявленні натискання відбувається оновлення двох ключових змінних.

Змінна currentNetwork, що є індексом мережі в масиві результатів (0-базована), інкрементується з використанням оператора ділення за модулем: currentNetwork = (currentNetwork + 1) % numNetworks;. Це забезпечує циклічний перехід від останньої мережі до першої (з індексом 0), створюючи нескінченну прокрутку списку.

Змінна currentPage, що відображається користувачеві (1-базована), оновлюється за схожою логікою: currentPage = (currentPage % numNetworks) + 1;.

Додавання одиниці забезпечує інтуїтивно зрозумілу для користувача нумерацію сторінок, починаючи з 1.

Ця логіка дозволяє створити простий, надійний та зручний для користувача механізм навігації по всіх знайдених бездротових мережах.

3.6. Оптимізація пам’яті для ESP32

Хоча ESP32 має значно більший обсяг оперативної пам'яті (SRAM) порівняно з попередніми поколіннями мікроконтролерів, її раціональне використання є важливим для забезпечення стабільності роботи [10].

Найбільш значущим аспектом оптимізації є те, що програма не створює в оперативній пам'яті власний масив структур для зберігання повної інформації про всі виявлені мережі. Замість цього, вона покладається на внутрішній буфер бібліотеки WiFi.h. Після виклику WiFi.scanNetworks() результати зберігаються у спеціальній, прихованій від розробника області пам'яті.

Функція displayNetworkInfo() на кожному етапі відображення звертається до цього буфера "на вимогу" за допомогою функцій WiFi.SSID(i), WiFi.BSSIDstr(i), WiFi.RSSI(i) і т.д. Це означає, що в оперативну пам'ять програми копіюється лише інформація про одну, поточну мережу, що відображається, а не про весь список одразу. Такий підхід радикально знижує вимоги до SRAM, особливо при скануванні в середовищі з великою кількістю (десятки) точок доступу.

РОЗДІЛ 4. ПРАКТИЧНІ РЕЗУЛЬТАТ РОБОТИ

Проект був зібраний на макетній платі, живлення забезпечене через power bank через Type-C – підключення, бо в аналізаторі доступних мереж важлива мобільність та гнучкість.

В експерименті аналізується мережа лабораторії факультету радіофізики, електроніки на комп’ютерних систем під назвою «RELAB_2.4». Як можна побачити, дані виводяться стабільно та цілісно, залишається лише перевірити достовірність інформації – найбільше цікавить визначення відстані до роутера завдяки RSSI.

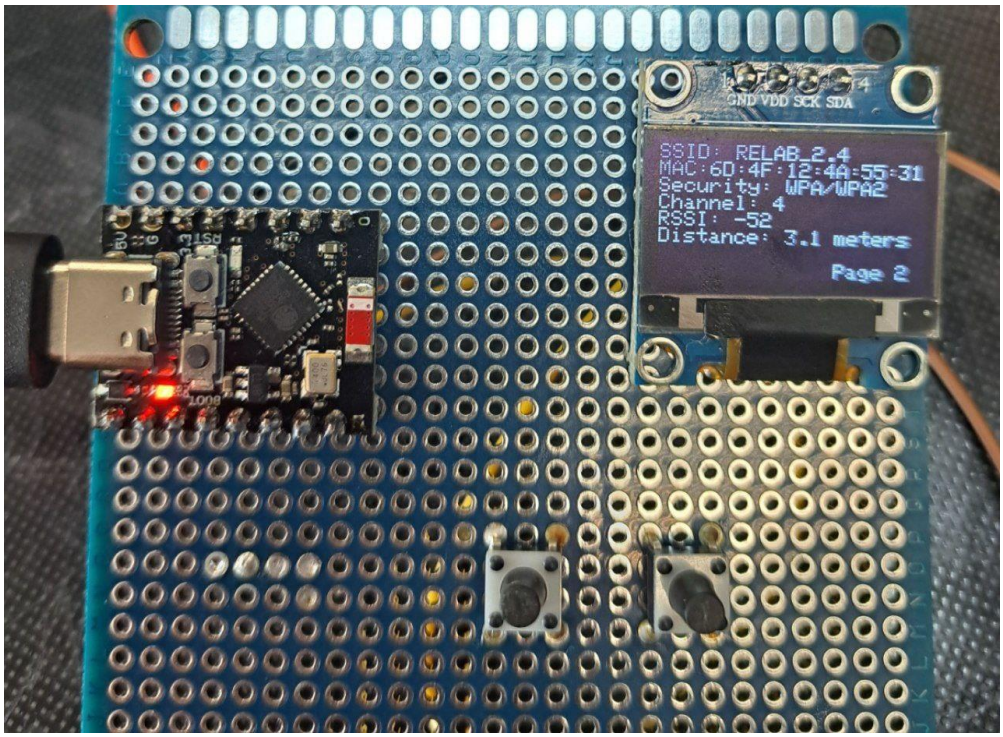

Рис. 4.1. - Зібрана схема

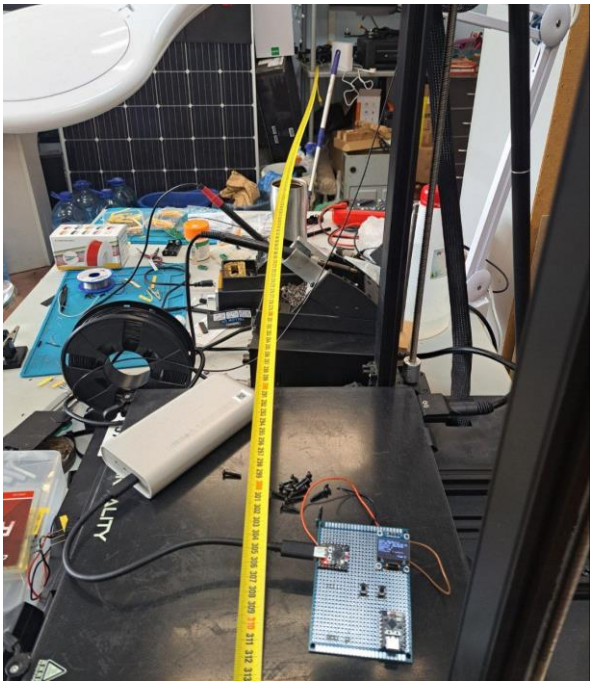

Рис. 4.2. - Фізичний вимір для порівняння

В умовах відсутності перешкод для сигналу, пристрій показує вірну відстань (вимірювання рулеткою дало приблизно 306 см відстані до роутера, пристрій показує 3.1 м). Оскільки величина округлюється до десятих частин метра – експеримент можна вважати успішним.

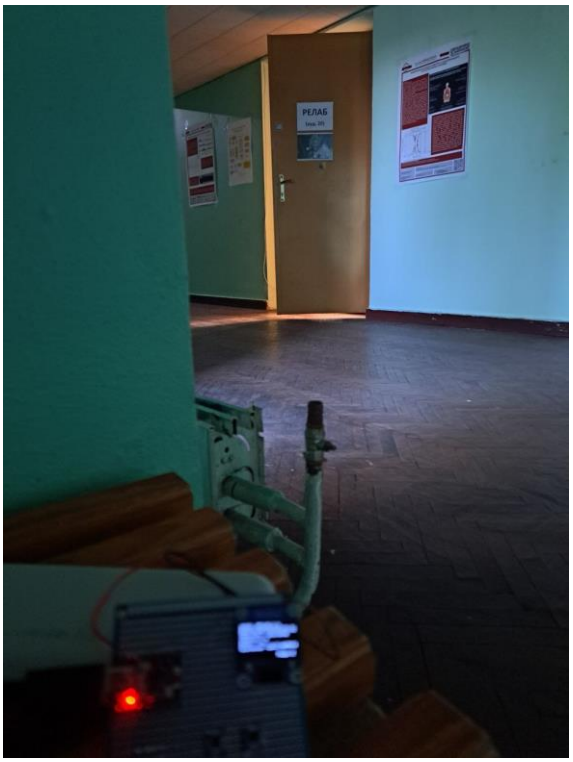

Рис. 4.3. Виміри в умовах наявності перешкод

В другому експерименті аналіз мережі проводився за межами кабінету, де знаходиться роутер – відстань приблизно 3-4 метра, враховуючи стіну. Оскільки через перешкоди сигнал стає набагато слабшим (міняється величина RSSI), пристрій показав неправильну відстань, а саме 22.5 метри.

Так відбулось через сталі коефіцієнти у формулі обчислення відстані з використанням параметру RSSI.

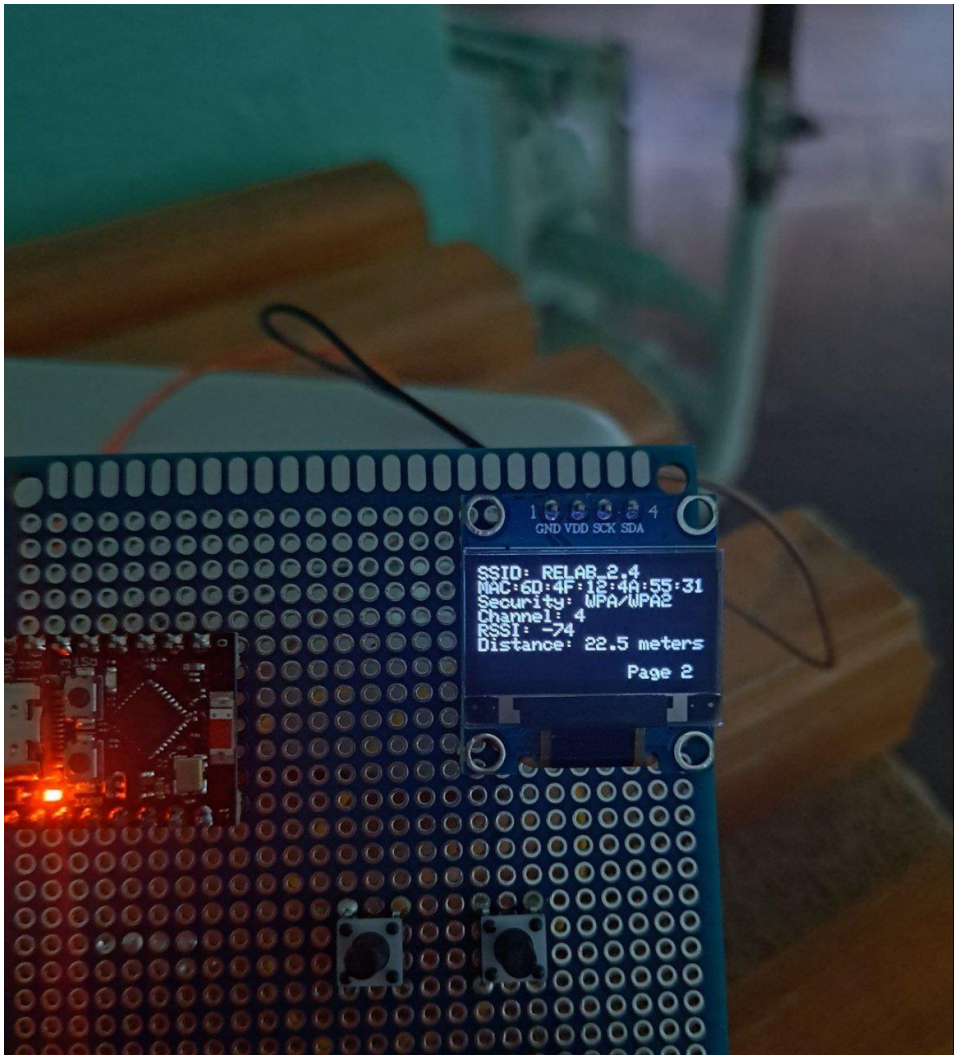

Рис. 4.4 - Покази аналізатора в другому експерименті

ВИСНОВОК

У результаті виконання курсової роботи було розроблено автономний пристрій для сканування Wi-Fi мереж на базі мікроконтролера ESP32. Пристрій реалізує функціонал збору інформації про точки доступу (SSID, MAC-адреса, тип шифрування, RSSI, канал), виведення кожної мережі на окрему сторінку OLED-дисплея, оцінювання відстані до джерела сигналу за рівнем потужності (RSSI), а також циклічне оновлення інформації через взаємодію з користувачем.

Були досліджені ключові принципи функціонування стандарту IEEE 802.11, алгоритми обрахунку відстані у бездротовому середовищі та особливості цифрової обробки сигналів у реальному часі. Завдяки використанню ESP32, вдалося реалізувати стабільну багатозадачну систему із мінімальними затримками, що є ефективним прикладом застосування вбудованих платформ у телекомунікаційній практиці.

Результати експериментів підтвердили працездатність пристрою: точність розрахунків є прийнятною у вільному просторі та чітко демонструє вплив перешкод на достовірність оцінок. Отримана система може бути корисною для інженерів, які займаються проектуванням, обслуговуванням та моніторингом бездротових мереж, а також як навчальний приклад інтеграції апаратного та програмного забезпечення в рамках телекомунікаційних технологій.

Перелік джерел посилання

1. IEEE Std 802.11™ URL: https://standards.ieee.org/ieee/802.11/

(дата звернення: 03.05.2025)

2. Asymptotic Performance Limits of Switches with Buffered Crossbars Supporting Multicast Traffic

URL: https://ieeexplore.ieee.org/document/4146869

(дата звернення: 18.05.2025)

3. Espressif ESP32 Technical Reference Manual

URL: https://espressif.com/en/support/documents/technical-documents

(дата звернення: 07.05.2025)

4. Monochrome OLED Breakouts

URL: https://learn.adafruit.com/monochrome-oled-breakouts

(дата звернення: 08.05.2025)

5. Arduino IDE Documentation URL: https://docs.arduino.cc/

https://www.arduino.cc/reference/en/

(дата звернення: 07.05.2025)

6. Arduino Language Reference

URL: https://docs.arduino.cc/language-reference/
(дата звернення: 03.05.2025)

7. Adafruit SSD1306 and GFX Library Documentation

URL: https://learn.adafruit.com/monochrome-oled-breakouts/arduino-library-and-examples

(дата звернення: 07.05.2025)

8. Basics of PWM (Pulse Width Modulation)

URL: https://docs.arduino.cc/learn/microcontrollers/analog-output/

(дата звернення: 05.05.2025)

9. Espressif GPIO & RTC GPIO

URL: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/gpio.html

(дата звернення: 08.05.2025)

10. Heap Memory Allocation

URL: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/mem_alloc.html

(дата звернення: 15.05.2025)