Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

Програмно-технічний засіб контролю доступу до складських приміщень з RFID та веб сервером на платформі ESP32
Назва теми

Рівень вищої освіти *перший (бакалаврський)*

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *123 «Комп'ютерна інженерія»*
Шифр, назва

Освітня програма *«Комп'ютерна інженерія та програмування»*
Назва

Шифр КвРКІ 022065.22.02.05 ПЗ

Виконав здобувач IV курсу, група КІ2-22-2 _____ Владислав НЕЗГОДА
Підпис Ініціали, прізвище

Керівник ДФ _____ Юрій ВОЙЧУР
Науковий ступінь, учене звання Підпис Ініціали, прізвище

Нормоконтролер канд. фіз.-мат. наук, доц. _____ Тетяна КИСІЛЬ
Науковий ступінь, учене звання Підпис Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІІС _____ Ольга ПАВЛОВА
«19» _червня_ 2026 р. підпис Ініціали, прізвище
дата

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ
Завідувачка кафедри КІІС
_____ Ольга ПАВЛОВА

" 10 " 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Незгоді Владиславу Андрійовичу
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб контролю доступу до складських приміщень з RFID та веб сервером на платформі ESP32
Керівник проекту (роботи) Войчур Юрій Олексійович, ДФ
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Аналіз предметної області та огляд відомих рішень
Проєктування програмно-технічного засобу контролю доступу до складських приміщень з RFID та веб сервером на платформі ESP32
Симуляція програмно-технічного засобу контролю доступу до складських приміщень з RFID та веб сервером на платформі ESP32

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
Узагальнена структура та схема симуляції програмно-технічного засобу
Схема електрична
Інтерфейсні вікна результатів симуляції

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «_10_»_____01_____2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2026	виконано
4	Робота над розділом 2 – вибір компонентів для проєктування програмно-технічного засобу	01.04.2026	виконано
5	Робота над розділом 3 – реалізація та тестування програмно-технічного засобу	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	виконано
7	Попередній захист ВКР	25.05.2026	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач_____ _____Владислав НЕЗГОДА_____
Підпис Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи_____ _____Юрій ВОЙЧУР_____
Підпис Ім'я, ПРІЗВИЩЕ

№ рядка	формат	Позначення	Найменування	Кіл. листів	№ екз	Примітка
			Текстові документи			
1		КвРКІ 022065.22.02.05 ПЗ	Пояснювальна записка	60		
			Графічні матеріали			
2		КвРКІ 022065.22.02.05 Е8	Узагальнена структура та схема симуляції програмно-технічного засобу	1		
3		КвРКІ 022065.22.02.05 Е3	Схема електрична	1		
4		КвРКІ 022065.22.02.05 Е8	Інтерфейсні вікна результатів симуляції	1		

Зм	Арк	№ докум	Підпис	Дата	КвРКІ 22065.22.02.17 ВП		
Розробив		Незгода			Відомість проекту	Літера	Аркуш 1 / Аркушів 1
Перевір.		Войчур				У	
Н. контр.		Кисіль				ХНУ, КІ2-22-2	
Затв.		Павлова		01.06			

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб контролю доступу до складських приміщень з RFID та веб сервером на платформі ESP32».

Автор роботи: Владислав НЕЗГОДА.

Керівник роботи: Юрій ВОЙЧУР.

Пояснювальна записка: 60 с., 28 рис., 4 табл., 3 дод., 63 джерела.

Графічна частина: 3 креслення.

RFID, БЕЗКОНТАКТНИЙ ДОСТУП, МІКРОКОНТРОЛЕР ESP32, СЕРВЕР.

Традиційні методи охорони, такі як механічні замки та фізична охорона не забезпечують належного рівня контролю, не ведуть автоматизованого обліку відвідувань і не дозволяють гнучко управляти правами доступу окремих працівників. Стрімкий розвиток концепції Інтернету речей та здешевлення мікроконтролерних платформ відкривають нові можливості для створення доступних електронних систем контролю доступу з мережевою інтеграцією. Незважаючи на наявність комерційних рішень у цій сфері, більшість із них є надмірно дорогими для малих і середніх підприємств або не забезпечують необхідної гнучкості налаштування. Це зумовлює актуальність розробки доступного, відмовостійкого та функціонально повного програмно-технічного засобу контролю доступу на базі RFID-технології та веб-сервера.

Метою дипломної роботи є проєктування програмно-технічного засобу контролю доступу до складських приміщень на платформі ESP32 з використанням RFID-ідентифікації, що забезпечує автоматизовану авторизацію персоналу, реєстрацію подій та дистанційне адміністрування через веб-інтерфейс.

Підпис здобувача

30.05.2026

Дата

ЗМІСТ

ВСТУП

Актуальність дослідження. Забезпечення фізичної безпеки складських приміщень є одним із пріоритетних завдань підприємств логістичної, виробничої та торговельної галузей. Несанкціонований доступ до складів призводить до матеріальних збитків, порушення технологічних процесів та витоку комерційно чутливої інформації. Традиційні методи охорони, такі як механічні замки та фізична охорона не забезпечують належного рівня контролю, не ведуть автоматизованого обліку відвідувань і не дозволяють гнучко управляти правами доступу окремих працівників. Стрімкий розвиток концепції Інтернету речей та здешевлення мікроконтролерних платформ відкривають нові можливості для створення доступних електронних систем контролю доступу з мережевою інтеграцією. Незважаючи на наявність комерційних рішень у цій сфері, більшість із них є надмірно дорогими для малих і середніх підприємств або не забезпечують необхідної гнучкості налаштування. Це зумовлює актуальність розробки доступного, відмовостійкого та функціонально повного програмно-технічного засобу контролю доступу на базі RFID-технології та веб-сервера.

Метою дипломної роботи є проєктування програмно-технічного засобу контролю доступу до складських приміщень на платформі ESP32 з використанням RFID-ідентифікації, що забезпечує автоматизовану авторизацію персоналу, реєстрацію подій та дистанційне адміністрування через веб-інтерфейс.

Об’єктом дослідження є система контролю та обліку доступу персоналу до складських приміщень на основі технологій радіочастотної ідентифікації та Інтернету речей.

Предметом дослідження є методи та засоби проєктування програмно-технічного засобу контролю доступу з RFID-ідентифікацією, локальним резервним зберіганням даних та серверною частиною для централізованого управління і моніторингу.

Для досягнення поставленої мети в роботі вирішуються такі завдання: аналіз предметної галузі систем контролю доступу та огляд існуючих технологій ідентифікації і програмних рішень; формування вимог до розроблюваного засобу та обґрунтування його архітектури; вибір елементної бази та розробка принципової електричної схеми пристрою; розробка програмного забезпечення мікроконтролера ESP32 з реалізацією двоступеневої авторизації та мережевої взаємодії; розробка серверної частини на базі Python Flask з REST API та веб-інтерфейсом адміністратора; верифікація розробленого рішення шляхом симуляції у середовищі Wokwi з підключенням до локального сервера через HTTP-тунель ngrok.

Практична цінність роботи полягає в тому, що розроблений програмно-технічний засіб може бути впроваджений на підприємствах малого та середнього бізнесу як доступна альтернатива дорогим комерційним системам контролю доступу, забезпечуючи автоматизовану RFID-ідентифікацію, ведення журналу подій, дистанційне адміністрування через браузер та відмовостійкість завдяки резервній локальній авторизації на SD-картці.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ВІДОМИХ РІШЕНЬ

1.1 Аналіз предметної галузі систем контролю доступу

Контроль доступу є однією з фундаментальних складових комплексної системи фізичної безпеки будь-якого підприємства. Під системою контролю доступу (СКД) розуміється сукупність програмно-технічних засобів, організаційних заходів та правил, що регламентують можливість переміщення осіб або транспортних засобів у межах об’єкта, що охороняється. Основне завдання системи контролю доступу полягає у забезпеченні доступу до охоронюваних зон виключно авторизованому персоналу, одночасно ведучи повний облік всіх фактів проходу із прив’язкою до конкретної особи та часу події.

Складські приміщення є об’єктами підвищеної уваги з точки зору фізичної безпеки. Це пов’язано з концентрацією матеріальних цінностей – товарів, сировини, готової продукції, обладнання на обмеженій площі. Несанкціонований доступ до складу може призвести до розкрадання, псування майна, порушення технологічного процесу або витоку комерційно чутливої інформації про залишки та рух товарів. За даними досліджень у сфері роздрібної торгівлі та логістики, значна частка втрат товарно-матеріальних цінностей на підприємствах пов’язана саме з внутрішніми крадіжками, здійсненими персоналом, що має або отримує несанкціонований доступ до складських зон.

Традиційні методи охорони складів механічні замки, охоронці на входах, відеоспостереження мають суттєві обмеження. Механічні замки не забезпечують диференційованого доступу для різних категорій персоналу і не ведуть журналу відвідувань. Охоронці є дорогим і суб’єктивним рішенням, схильним до людського фактору. Відеоспостереження фіксує факти подій, але не попереджає їх і потребує значних ресурсів для аналізу записів. Системи контролю доступу на базі електронної ідентифікації вирішують усі ці проблеми одночасно: забезпечують автоматизований контроль з диференційованими правами доступу,

ведуть детальний електронний журнал та інтегруються з іншими підсистемами безпеки підприємства.

Сучасні системи контролю доступу для складських приміщень виконують широкий спектр функцій. Ідентифікація та автентифікація персоналу є первинною функцією, таким чином, що система визначає особу працівника та перевіряє його право на доступ до конкретної зони у конкретний час. Аудит та протоколювання подій забезпечує формування детального журналу всіх фактів проходу з інформацією про особу, час та напрямок руху, що є необхідним для розслідування інцидентів та підтвердження присутності персоналу. Управління правами доступу дозволяє адміністратору гнучко налаштовувати права окремих співробітників або груп, обмежуючи доступ за зонами, часом доби або днями тижня. Інтеграція з суміжними системами такими як обліком робочого часу, відеоспостереженням, системами пожежної сигналізації перетворює СКД на елемент комплексної інфраструктури безпеки підприємства.

Ринок систем контролю доступу демонструє стале зростання у всьому світі. Драйверами розвитку галузі є зростання вимог до безпеки на підприємствах, здешевлення електронних компонентів і мікроконтролерів, поширення концепції Інтернету речей (IoT), яка дозволяє інтегрувати СКД у загальну мережеву інфраструктуру підприємства, а також зростання попиту на хмарні рішення для централізованого управління доступом на розподілених об’єктах. Малий та середній бізнес, зокрема логістичні компанії та склади, дедалі активніше впроваджує доступні електронні системи контролю доступу замість традиційних механічних засобів охорони.

1.2 Огляд технологій ідентифікації в системах контролю доступу

Основою будь-якої системи контролю доступу є технологія ідентифікаці, тобто спосіб, яким система розпізнає особу або об’єкт, що намагається отримати доступ. На сьогодні у практичних СКД застосовуються чотири основні

технології ідентифікації: оптичне зчитування штрих-кодів, магнітні картки, радіочастотна ідентифікація (RFID) та біометричні методи. Кожна з технологій має характерні переваги та обмеження, що визначають область її застосування.

Ідентифікація за штрих-кодом є найпростішою і найдешевшою технологією. Картка або пропуск містить надрукований штрих-код, який зчитується оптичним сканером при піднесенні до зчитувача. Основним недоліком є необхідність прямої видимості між кодом і сканером, чутливість до забруднення та механічного пошкодження носія, а також висока вразливість до підробки – штрих-код можна скопіювати звичайним принтером. Через ці обмеження штрих-кодові системи практично не застосовуються у сучасних системах контролю доступу і витіснені більш надійними технологіями.

Магнітні картки з магнітною смугою (стандарт ISO 7811) широко використовувалися у СКД протягом 1980–2000-х років. Інформація зберігається на магнітній смузі у вигляді намагнічених ділянок і зчитується при проведенні карткою через зчитувач. Недоліки технології є суттєвими для застосувань у складських умовах: висока чутливість до магнітних полів і механічного зносу, необхідність безпосереднього контакту з зчитувачем, відносна простота копіювання даних з магнітної смуги за допомогою доступного обладнання. Магнітні картки поступово виходять з ужитку у нових інсталяціях, поступаючись місцем RFID-технологіям.

Біометричні системи ідентифікують особу за унікальними фізіологічними характеристиками – відбитком пальця, геометрією долоні, малюнком вен, зображенням обличчя або райдужної оболонки ока. Головною перевагою біометрії є неможливість передачі ідентифікатора іншій особі на відміну від картки, яку можна позичити або вкрасти. Однак біометричні системи мають суттєві обмеження для промислового застосування: висока вартість обладнання, чутливість до змін фізіологічного стану (забруднені або травмовані руки), складність розгортання та обслуговування, а також юридичні та етичні питання щодо збору та зберігання біометричних даних персоналу. Крім того, час

ідентифікації у деяких біометричних системах може бути вищим порівняно з RFID, що є критичним при організації потоків персоналу.

Радіочастотна ідентифікація (RFID) займає домінуюче положення на ринку сучасних систем контролю доступу завдяки оптимальному поєднанню вартості, надійності, швидкодії та зручності використання. RFID-картка не потребує контакту зі зчитувачем і може бути прочитана крізь одяг або гаманець. Технологія стійка до забруднень і механічних впливів, оскільки мікрочіп захищений пластиковим корпусом картки. Сучасні RFID-картки підтримують криптографічний захист даних, що суттєво ускладнює їх несанкціоноване копіювання (рис. 1.1). Таблиця 1.1 містить порівняльний аналіз розглянутих технологій ідентифікації за ключовими критеріями.

Таблиця 1.1 – Порівняльний аналіз технологій ідентифікації

Критерій	Штрих-код	Магнітна картка	RFID	Біометрія
Відстань зчитування	Контакт	Контакт	До 100 см	Контакт / до 50 см
Стійкість до забруднень	Низька	Низька	Висока	Середня
Можливість підробки	Висока	Висока	Середня	Низька
Вартість інфраструктури	Низька	Низька	Середня	Висока
Потреба у фіз. контакті	Так	Так	Ні	Ні / Так
Запис даних на носій	Ні	Обмежено	Так	Ні

Аналіз таблиці 1.1 підтверджує, що RFID-технологія є оптимальним вибором для системи контролю доступу до складських приміщень. Вона поєднує безконтактне зчитування, стійкість до промислових умов експлуатації, середній рівень захисту від підробки при прийнятній вартості інфраструктури. Для підвищення рівня безпеки RFID може використовуватися у комбінації з PIN-кодом (двофакторна автентифікація), однак для більшості складських застосувань однофакторна RFID-ідентифікація є достатньою.

Рисунок 1.1 – Функціонування інформаційної системи із застосуванням RFID технології

1.3 Аналіз RFID-технології

Радіочастотна ідентифікація (Radio Frequency Identification, RFID) це технологія автоматичної ідентифікації об’єктів за допомогою радіохвиль. Система RFID складається з двох основних компонентів: транспондера (мітки або картки) та зчитувача (рідера). Транспондер містить мікрочіп з пам’яттю і антену, зчитувач генерує електромагнітне поле і зчитує дані з мітки при її потраплянні у зону дії антени.

За діапазоном робочих частот RFID-системи поділяються на кілька класів, кожен з яких має специфічні характеристики і область застосування. Низькочастотні системи (LF, 125–134 кГц) мають невелику відстань зчитування

до 10 см і низьку швидкість передачі даних, проте добре проникають крізь металеві поверхні і рідини. Вони широко застосовуються у системах ідентифікації тварин та ранніх системах контролю доступу. Високочастотні системи (HF, 13.56 МГц) забезпечують відстань зчитування до 1 метра, вищу швидкість передачі даних та підтримку криптографічного захисту. Саме на цій частоті працює переважна більшість сучасних систем контролю доступу, транспортних карток та платіжних систем. Ультрависокочастотні системи (UHF, 860–960 МГц) забезпечують зчитування на відстані до 10 метрів і застосовуються у логістиці для відстеження вантажів та управління складськими запасами.

Стандарт ISO/IEC 14443, що регулює роботу безконтактних смарт-карток на частоті 13.56 МГц, визначає два типи карток: Type A та Type B, що відрізняються методом модуляції та протоколом антиколізії. Найбільш поширеними картками стандарту ISO/IEC 14443 Type A є Mifare Classic виробництва NXP Semiconductors. Mifare Classic 1K містить 1 кБ пам’яті, розподіленої на 16 секторів по 4 блоки, кожен сектор захищений двома 48-бітними ключами автентифікації. Mifare Ultralight є спрощеною версією без криптографічного захисту з 512 бітами пам’яті, призначеною для одноразових або малобюджетних застосувань.

Мікросхема MFRC522 виробництва NXP Semiconductors є одним з найпоширеніших RFID-зчитувачів для діапазону 13.56 МГц у сфері вбудованих систем і DIY-проєктів. Вона підтримує всі стандарти ISO/IEC 14443 Type A/B і може зчитувати унікальний ідентифікатор картки (UID), читати та записувати дані у захищені сектори пам’яті Mifare Classic, а також виконувати взаємну автентифікацію з карткою за алгоритмом Crypto1. Мікросхема підключається до мікроконтролера по шині SPI, I2C або UART, має вбудований генератор поля 13.56 МГц та схему узгодження з антеною. Ефективна відстань зчитування стандартних карток ISO/IEC 14443A складає 30–50 мм залежно від розміру антени.

Принцип роботи пасивних RFID-карток (без власного джерела живлення) ґрунтується на явищі електромагнітної індукції. Зчитувач генерує змінне електромагнітне поле, яке індукує електрорушійну силу в антені картки. Отримана енергія заряджає конденсатор і живить мікрочіп картки. Картка модулює навантаження на своїй антені (load modulation), що впливає на характеристики поля зчитувача і дозволяє передавати дані від картки до зчитувача без власного джерела живлення. Цей принцип забезпечує фактично необмежений термін служби пасивних карток за відсутності механічного пошкодження.

RFID-технологія використовується в багатьох сферах діяльності, де потрібна автоматизація процесів, контроль та ідентифікація об’єктів. RFID-технологія використовується:

– у роздрібній торгівлі – для контролю переміщення товарів, проведення інвентаризації та зменшення ризику крадіжок;

– у системах контролю та управління доступом – для обмеження доступу до офісів, житлових будинків, університетів, готелів та інших об’єктів;

– у безконтактних платіжних системах – для швидкого здійснення оплат;

– у громадському транспорті – для роботи електронних квитків та автоматизації оплати проїзду;

– у сільському господарстві – для чипування та ідентифікації домашніх і сільськогосподарських тварин;

– у логістиці – для обліку вантажів, відстеження їх переміщення та контролю доставки;

– у бібліотеках та архівах – для автоматизації обліку книг і документів;

– у промисловості – для контролю виробничих процесів, обліку сировини та доступу персоналу;

– у фармацевтичній сфері – для перевірки справжності лікарських препаратів;

– у системах захисту товарів – для боротьби з підробками та незаконним продажем продукції.

З точки зору безпеки RFID-системи мають певні вразливості. Основним ризиком є несанкціоноване перехоплення даних або копіювання ідентифікаторів.

Однією з базових вразливостей є дистанційне зчитування (eavesdropping). Оскільки передача даних між міткою та зчитувачем відбувається через електромагнітне поле, зловмисник, використовуючи спеціалізоване обладнання з антенами з високим коефіцієнтом підсилення, може перехопити сигнал або ініціювати приховане зчитування UID (Unique Identifier) картки на відстані від декількох десятків сантиметрів до метра. Це дозволяє отримати ідентифікатор користувача без фізичного контакту з карткою.

Найбільш поширеною загрозою для систем початкового рівня є клонування ідентифікаторів. Більшість бюджетних систем, подібних до тієї, що реалізована в роботі, використовують для автентифікації лише серійний номер картки (UID). На сьогоднішній день існують так звані «магічні картки» (UID-changeable cards), які дозволяють програмно перезаписувати блок 0, де зберігається ідентифікатор. Це робить можливим створення повного дубліката легітимної картки за лічені секунди за допомогою портативних пристроїв (наприклад, Flipper Zero або Proxmark3).

Проте, рівень захищеності суттєво зростає при використанні сучасних стандартів: Mifare Classic або Mifare DESFire / Plus. Хоча алгоритм шифрування Crypto1 у картках Mifare Classic вважається застарілим і вразливим до атак типу Nested або Darkside, використання захищених секторів пам’яті та унікальних ключів доступу (Key A / Key B) замість простого зчитування UID значно підвищує поріг входження для зловмисника. Щодо Mifare DESFire / Plus, то ці карти використовують стійке шифрування AES-128, що практично унеможливлює клонування на сучасному рівні розвитку обчислювальної техніки.

Для умов типового підприємства, де розгортається дана система, авторизація за UID є компромісним рішенням між вартістю впровадження та необхідним рівнем безпеки. Мінімізація ризиків у такому випадку досягається шляхом комплексних організаційних заходів:

– використання відеоспостереження у зонах зчитування для детекції підозрілої активності;

– логування всіх спроб доступу (включаючи невдалі) на SD-карту або сервер, що дозволяє проводити ретроспективний аналіз безпеки;

– обмеження часу дії карток та оперативне анулювання доступу для втрачених ідентифікаторів у базі даних.

1.4 Огляд платформ для реалізації вбудованих систем контролю доступу

Вибір апаратної платформи для реалізації вбудованої системи контролю доступу є критичним рішенням, що впливає на функціональні можливості, складність розробки, вартість та масштабованість кінцевого рішення. На сучасному ринку доступний широкий спектр платформ від простих мікроконтролерів без операційної системи до повноцінних одноплатних комп'ютерів під керуванням Linux, а також спеціалізовані контролери доступу з готовим програмним забезпеченням.

Сімейство мікроконтролерів Arduino (зокрема Arduino Uno та Arduino Mega) є найпопулярнішою платформою для початківців у сфері вбудованих систем завдяки простоті програмування та великій кількості навчальних матеріалів (рис. 1.2). Однак для задач контролю доступу з мережевою взаємодією Arduino має суттєве обмеження, що полягає у відсутності вбудованого модуля WiFi або Ethernet [15-20]. Для підключення до мережі необхідний окремий модуль (ESP8266, W5100 тощо), що ускладнює схему, збільшує розміри та вартість пристрою. Тактова частота процесора ATmega (16 МГц для Uno, Arduino

Mega) є достатньою для простих задач, однак може бути обмежуючим фактором при одночасній обробці SPI-периферії та мережевого стеку.

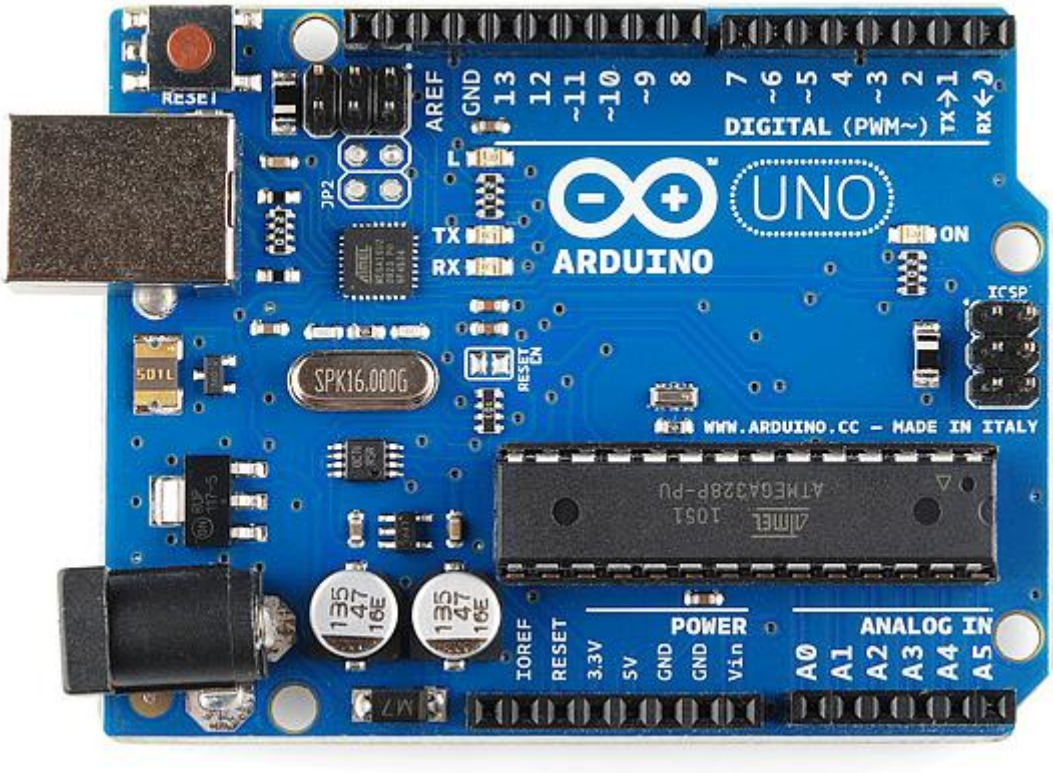


Рисунок 1.2 – Мікроконтролер Arduino Uno [21]

Одноплатний комп’ютер Raspberry Pi (зокрема моделі 3B+ та 4) є потужною платформою з повноцінною операційною системою Linux, чотириядерним процесором ARM та вбудованим WiFi і Bluetooth. Raspberry Pi дозволяє запускати складне програмне забезпечення, таке як веб-сервери, бази даних, системи розпізнавання облич безпосередньо на пристрої. Однак для задачі контролю доступу на одному складі можливості Raspberry Pi є надлишковими. Платформа потребує операційної системи, що збільшує час завантаження до кількох десятків секунд і створює ризики збоїв ОС. Вартість Raspberry Pi 4 значно перевищує вартість ESP32, а споживання електроенергії є вищим. Для промислових застосувань Raspberry Pi є менш надійним вибором порівняно з

мікроконтролерними платформами через відсутність апаратного watchdog та чутливість до некоректного вимкнення живлення.

Рисунок 1.3 – Одноплатний комп’ютер Raspberry Pi [22]

Мікроконтролер ESP32 виробництва Espressif Systems є оптимальною платформою для IoT-застосувань, де потрібне поєднання обчислювальної потужності, мережевої підключеності та низького енергоспоживання при помірній вартості. ESP32 містить двоядерний процесор Xtensa LX6 з тактовою частотою до 240 МГц, вбудовані модулі WiFi 802.11 b/g/n та Bluetooth 4.2/5.0, 520 кБ SRAM та 4–16 МБ Flash-пам’яті залежно від модифікації. Наявність вбудованого WiFi-стеку з підтримкою TCP/IP, TLS та HTTP усуває необхідність у зовнішніх мережевих модулях і значно спрощує схему. Широка підтримка периферійних інтерфейсів (SPI, I2C, UART, PWM, ADC, DAC) дозволяє підключити всі необхідні компоненти, зокрема RFID-зчитувач, SD-карту, бузер, світлодіод без додаткових мікросхем. Таблиця 1.2 містить порівняльний аналіз розглянутих платформ.

Таблиця 1.2 – Порівняльний аналіз апаратних платформ

Критерій	ESP32	Arduino Mega	Raspberry Pi	Готовий контролер
WiFi вбудований	Так	Ні	Так	Залежить
Потужність CPU	240 МГц, 2 ядра	16 МГц, 1 ядро	1.5 ГГц, 4 ядра	Обмежена
Підтримка SPI	Так	Так	Так	Залежить
Вартість (грн)	~350	~400	~1800	~3000+
Енергоспоживання	Низьке	Низьке	Високе	Середнє
Гнучкість налашт.	Висока	Висока	Дуже висока	Низька
Екосистема бібліотек	Широка	Широка	Дуже широка	Вузька

Готові контролери доступу, такі як ZKTeco серії Pro або Hikvision DS-K1T, є спеціалізованими апаратно-програмними рішеннями, призначеними для швидкого впровадження систем контролю доступу без необхідності розробки власного програмного забезпечення. Такі пристрої мають вбудоване ПЗ, готовий веб-інтерфейс для налаштування та адміністрування, а також підтримують поширені протоколи обміну даними, зокрема Wiegand та OSDP, що дозволяє інтегрувати їх із електронними замками, турнікетами, зчитувачами карток та іншими компонентами системи безпеки.

Перевагою готових контролерів є простота встановлення, швидке введення в експлуатацію та наявність технічної підтримки від виробника. Крім цього, такі системи зазвичай мають готові механізми журналювання подій, керування користувачами та інтеграції з відеоспостереженням.

Разом із тим, використання готових рішень має і певні недоліки. Насамперед це висока вартість обладнання та ліцензійного програмного забезпечення, особливо при розгортанні системи на великій кількості точок доступу. Також функціональні можливості таких контролерів обмежуються програмним забезпеченням, передбаченим виробником, що ускладнює реалізацію нестандартної логіки роботи або інтеграцію зі специфічними сервісами.

Альтернативним підходом є створення власної системи контролю доступу на базі мікроконтролера ESP32. Такий варіант дозволяє значно зменшити вартість системи та забезпечує високу гнучкість у налаштуванні й модернізації. ESP32 підтримує бездротові технології Wi-Fi та Bluetooth, має достатню обчислювальну потужність і кількість інтерфейсів для підключення RFID-зчитувачів, електронних замків, датчиків та інших периферійних пристроїв.

Власна розробка на базі ESP32 дозволяє реалізувати індивідуальну бізнес-логіку, створити власний веб-інтерфейс, інтегрувати систему з базами даних, хмарними сервісами або мобільними застосунками. Крім того, така система може бути легко модифікована та розширена відповідно до потреб користувача без залежності від конкретного виробника обладнання чи обмежень готового програмного забезпечення.

1.5 Аналіз існуючих програмних рішень для контролю доступу

Ринок програмного забезпечення для систем контролю доступу охоплює широкий спектр рішень – від комерційних корпоративних платформ до відкритих проєктів з відкритим кодом. Аналіз існуючих рішень є необхідним кроком для виявлення прогалин, які заповнює розроблюваний засіб, та обґрунтування доцільності власної розробки.

Серед комерційних рішень провідне місце займають платформи ZKTeco, Hikvision та Bosch Security Systems (рис. 1.4-1.6). ZKTeco пропонує лінійку

пристроїв і програмного забезпечення ZKBioSecurity – комплексну платформу управління доступом з підтримкою RFID, біометрії, відеоверифікації та інтеграції з HR-системами. Програмна частина розгортається на Windows Server і потребує ліцензування, вартість якого для малих підприємств може бути невиправданою. Hikvision iVMS-4200 є популярним рішенням для відеоспостереження з модулем контролю доступу, що підтримує пристрої власного виробництва. Обмеженням є жорстка прив’язка до екосистеми Hikvision та складність інтеграції зі стороннім обладнанням. Bosch Building Technologies пропонує enterprise-рівня рішення для великих корпоративних об’єктів з розвиненою підтримкою, проте вартість ліцензій і впровадження робить ці системи недоступними для малого бізнесу.

Рисунок 1.4 – Платформа управління доступом з підтримкою RFID від ZKTeco

Рисунок 1.5 – Контролер доступу від Bosch Security Systems

Рисунок 1.6 – Термінал контролю доступу Hikvision

Серед рішень з відкритим кодом (open-source) виділяється проєкт OpenACS (Open Access Control System), що реалізує базові функції контролю доступу на базі мікроконтролерів Arduino з підключенням до Linux-сервера. Проєкт має обмежену документацію і спільноту, не підтримується активно і не пристосований до сучасних мікроконтролерних платформ з вбудованим WiFi. Інший відкритий проєкт – Tasmota для ESP8266/ESP32. Даний проєкт є універсальною прошивкою для IoT-пристроїв з підтримкою MQTT, яка може бути адаптована для задач контролю доступу, проте потребує значної доробки для реалізації специфічної логіки авторизації та ведення журналу.

Хмарні IoT-платформи, такі як AWS IoT, Google Cloud IoT, Microsoft Azure IoT Hub надають інфраструктуру для підключення вбудованих пристроїв до хмари з підтримкою протоколів MQTT та HTTP. Ці платформи є потужними, але надлишковими для локального розгортання системи контролю доступу на

одному складі. Вони потребують постійного інтернет-з’єднання, мають вартість, що залежить від обсягу трафіку та кількості пристроїв, і створюють залежність від хмарного провайдера. Для малого підприємства переваги хмарної платформи не виправдовують її складності та вартості.

Аналіз існуючих рішень виявляє характерну прогалину: комерційні системи є потужними, але дорогими і негнучкими; відкриті рішення є застарілими або недостатньо спеціалізованими; хмарні платформи є надлишковими для локального застосування. Розроблюваний засіб займає нішу доступного, гнучкого і повністю контрольованого рішення для малих та середніх підприємств, яке розгортається локально без хмарних залежностей і може бути адаптоване під конкретні вимоги замовника.

1.6 Постановка задачі

На підставі проведеного аналізу предметної галузі, огляду технологій ідентифікації, апаратних платформ та існуючих програмних рішень сформульовано задачу дипломної роботи: розробити програмно-технічний засіб контролю доступу до складських приміщень на базі мікроконтролера ESP32 з використанням RFID-технології та веб-сервера для управління і моніторингу.

Розроблюваний засіб повинен забезпечувати автоматизовану ідентифікацію персоналу за RFID-картками стандарту ISO/IEC 14443A на частоті 13.56 МГц з використанням зчитувача MFRC522. Система має надавати або відмовляти у доступі на основі перевірки унікального ідентифікатора картки за базою авторизованих користувачів, зберігати повний журнал подій сканування з прив’язкою до часу та статусу доступу, а також забезпечувати звукову та світлову індикацію результату авторизації.

Серверна частина системи має реалізовувати REST API для прийому даних від мікроконтролера та управління базою користувачів. Веб-інтерфейс має забезпечувати перегляд журналу подій і списку авторизованих користувачів,

додавання та видалення записів через браузер без необхідності встановлення спеціалізованого програмного забезпечення. Система має зберігати дані у файловому форматі CSV, що забезпечує простоту читання, редагування та резервного копіювання без залежності від СУБД.

Важливою вимогою є забезпечення відмовостійкості системи: при недоступності мережевого з'єднання або серверної частини мікроконтролер має автоматично перемикатися на локальну перевірку авторизації за копією бази даних, збереженою на SD-картці. Це гарантує безперервність роботи системи контролю доступу незалежно від стану мережевої інфраструктури підприємства.

Для підтвердження коректності розроблених технічних та програмних рішень передбачається проведення симуляції повного функціонального ланцюжка системи з використанням онлайн-симулятора Wokwi для апаратної частини та локального Flask-сервера з публічним доступом через HTTP-тунель ngrok. Симуляція дозволить верифікувати алгоритми авторизації, мережеву взаємодію та роботу веб-інтерфейсу без необхідності фізичного монтажу апаратного комплексу на етапі розробки.

Таким чином, розроблюваний засіб є комплексним рішенням, що охоплює апаратну схему, програмне забезпечення мікроконтролера та серверну частину, і спрямований на забезпечення надійного, економічно доступного та гнучкого контролю доступу до складських приміщень для підприємств малого та середнього бізнесу.

2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ КОНТРОЛЮ ДОСТУПУ ДО СКЛАДСЬКИХ ПРИМІЩЕНЬ З RFID ТА ВЕБ СЕРВЕРОМ НА ПЛАТФОРМІ ESP32

2.1 Формування вимог до програмно-технічного засобу

Проєктування будь-якого програмно-технічного засобу розпочинається з формування чіткого переліку функціональних і технічних вимог, які визначають межі системи, її очікувану поведінку та критерії успішності реалізації. У контексті засобу контролю доступу до складських приміщень вимоги формувалися з урахуванням специфіки середовища експлуатації, необхідного рівня безпеки, зручності адміністрування та економічної доцільності рішення.

Основною функціональною вимогою є автоматизована ідентифікація персоналу за допомогою RFID-карток стандарту ISO/IEC 14443A. Система має зчитувати унікальний ідентифікатор (UID) картки, звіряти його з базою авторизованих користувачів у режимі реального часу та надавати або відмовляти у доступі залежно від результату перевірки. Час реакції системи від піднесення картки до отримання відповіді не повинен перевищувати двох секунд за умови наявності мережевого з’єднання.

Вимоги до надійності передбачають наявність механізму резервної авторизації на випадок втрати з’єднання з сервером. У такому режимі мікроконтролер має самостійно перевіряти UID картки за локальною копією бази даних, збереженою на SD-картці. Це забезпечує безперервність роботи системи навіть при тимчасовій недоступності мережі або серверної частини.

До вимог щодо реєстрації подій відноситься обов’язкове логування кожного факту сканування з фіксацією часової мітки, ідентифікатора картки та статусу доступу. Журнал подій має зберігатися як локально на SD-картці, так і передаватися на центральний сервер для централізованого моніторингу. Адміністратор системи повинен мати змогу переглядати журнал через веб-

інтерфейс у браузері без встановлення спеціалізованого програмного забезпечення.

Вимоги до адміністрування передбачають можливість додавання та видалення авторизованих користувачів через веб-інтерфейс або REST API без перезавантаження системи та втручання у програмний код мікроконтролера. Система має підтримувати необмежену кількість облікових записів, обмежену лише обсягом носія SD-карти. Доступ до адміністративного інтерфейсу має бути захищений і доступний лише в межах довіреної мережі або через захищений тунель.

Технічні вимоги включають живлення системи від зовнішнього джерела напругою 12 В постійного струму з внутрішнім перетворенням до 5 В та 3.3 В для живлення окремих компонентів. Система має забезпечувати звукову індикацію результату авторизації за допомогою бузера та світлову індикацію за допомогою світлодіода. Усі компоненти мають бути електрично сумісними, промислово доступними та придатними для монтажу на друкованій платі.

2.2 Узагальнена структура програмно-технічного засобу

З метою виконання поставлених вимог запропоновано структуру програмно-технічного засобу контролю доступу до складських приміщень з RFID та вебсервером на платформі ESP32, що включає апаратну та програмну складові, які взаємодіють між собою в межах єдиної кіберфізичної системи. Апаратна частина побудована на базі мікроконтролера ESP32, який виконує функції центрального керуючого елемента та забезпечує обробку даних, комунікацію з периферійними модулями і передачу інформації до серверної частини через бездротову мережу Wi-Fi. До ESP32 підключено RFID-модуль RC522, що реалізує зчитування унікальних ідентифікаторів користувацьких карток за допомогою інтерфейсу SPI, який забезпечує швидку та надійну передачу даних. Паралельно використовується модуль SD-карти, також

підключений через SPI-шину, який виконує функцію локального зберігання даних, зокрема журналу подій та списку дозволених користувачів, що забезпечує можливість автономної роботи системи у випадку відсутності мережевого з’єднання. Блок світлової та звукової індикації призначений для візуального та акустичного інформування про результат перевірки доступу, що підвищує зручність експлуатації системи. Живлення всіх компонентів забезпечується відповідним блоком стабілізації напруги, який формує необхідний рівень 3,3 В для коректної роботи електронних модулів.

Програмна частина системи реалізована у вигляді розподіленої клієнт-серверної архітектури, де ESP32 виступає як клієнтський пристрій, що здійснює зчитування RFID-міток, первинну обробку даних, локальне логування та передачу інформації на сервер за допомогою HTTP-запитів. Серверна частина реалізована на базі веб-фреймворку Flask і забезпечує обробку запитів, перевірку ідентифікаторів користувачів, збереження даних у файловій системі та формування відповіді щодо дозволу або відмови у доступі. Дані про користувачів і події зберігаються у відповідних файлах, що виконують роль спрощеної бази даних. Крім того, сервер формує веб-інтерфейс, доступний через браузер, який дозволяє здійснювати моніторинг подій, перегляд журналу доступу та управління користувачами.

Взаємодія між апаратною та програмною частинами здійснюється через мережу Інтернет із використанням тунелювання, що забезпечує доступ до локального сервера з боку пристрою.

Отже, розроблена структура забезпечує стабільну роботу системи контролю доступу, поєднуючи можливості ESP32 та серверної частини. Це дозволяє ефективно обробляти дані, зберігати інформацію про користувачів і події, а також зручно переглядати їх через веб-інтерфейс.

Узагальнена структура проєктованого програмно-технічного засобу наведено на рис. 2.1.

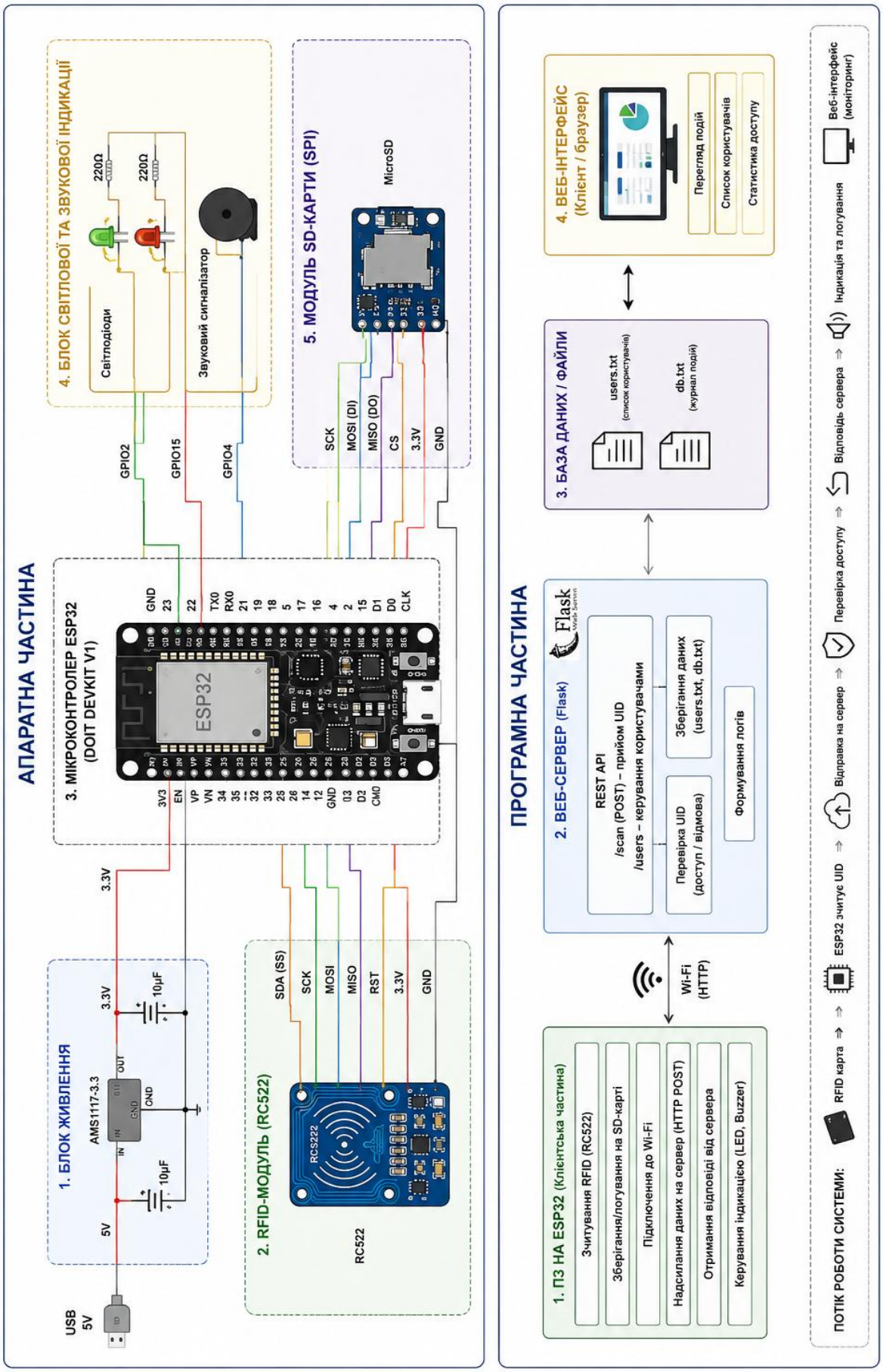


Рисунок 2.1 – Узагальнена структура проєктованого засобу

2.3 Схема електрична принципова

Електрична принципова схема розробленого програмно-технічного засобу доступу до складських приміщень з RFID та веб сервером на платформі ESP32 виконана у середовищі EasyEDA і охоплює шість функціональних блоків: мікроконтролер ESP32, RFID-модуль, модуль SD-карти, блоки світлової та звукової індикації, а також блок живлення.

Схему електричну проєктованого прграмно-технічного засобу доступу до складських приміщень наведено на рис. 2.2.

Центральним елементом схеми є мікроконтролер ESP32 у виконанні DevKit C з 30 виводами (U1). Мікроконтролер отримує живлення +5 В через пін VIN та формує внутрішню шину +3.3 В для живлення периферії.

З мікроконтролера виведені сигнальні лінії шини SPI – MOSI (пін 23), MISO (пін 19), SCK (пін 18) – які є спільними для RFID-модуля та SD-карти. Лінія вибору мікросхеми SDA/SS RFID-модуля підключена до піна 5, а лінія CS SD-карти – до окремого піна мікроконтролера, що запобігає конфліктам на спільній шині. Пін RST RFID-модуля підключений до піна 21. Окремий вихід мікроконтролера (пін 22, позначений LED) керує блоком світлової індикації, а вихід BUZZER – блоком звукової сигналізації. Піни TX0 та RX0 підключені до відповідних ліній послідовного монітора для налагодження.

RFID-модуль (U4) виконаний на базі мікросхеми MFRC522 і підключений до мікроконтролера по шині SPI. Модуль живиться від шини +3.3 В, сформованої внутрішнім стабілізатором ESP32. Лінії MOSI, MISO, SCK та SDA є спільними з SD-картою і розводяться по шині SPI з роздільним керуванням через CS. Пін RST підключений до відповідного виводу мікроконтролера і дозволяє програмно скидати модуль. Антенний вивід (SCA) підключений до зовнішньої антени зчитувача, розташованої на лицьовій панелі пристрою. Модуль забезпечує зчитування RFID-карток на частоті 13.56 МГц на відстані до 50 мм.

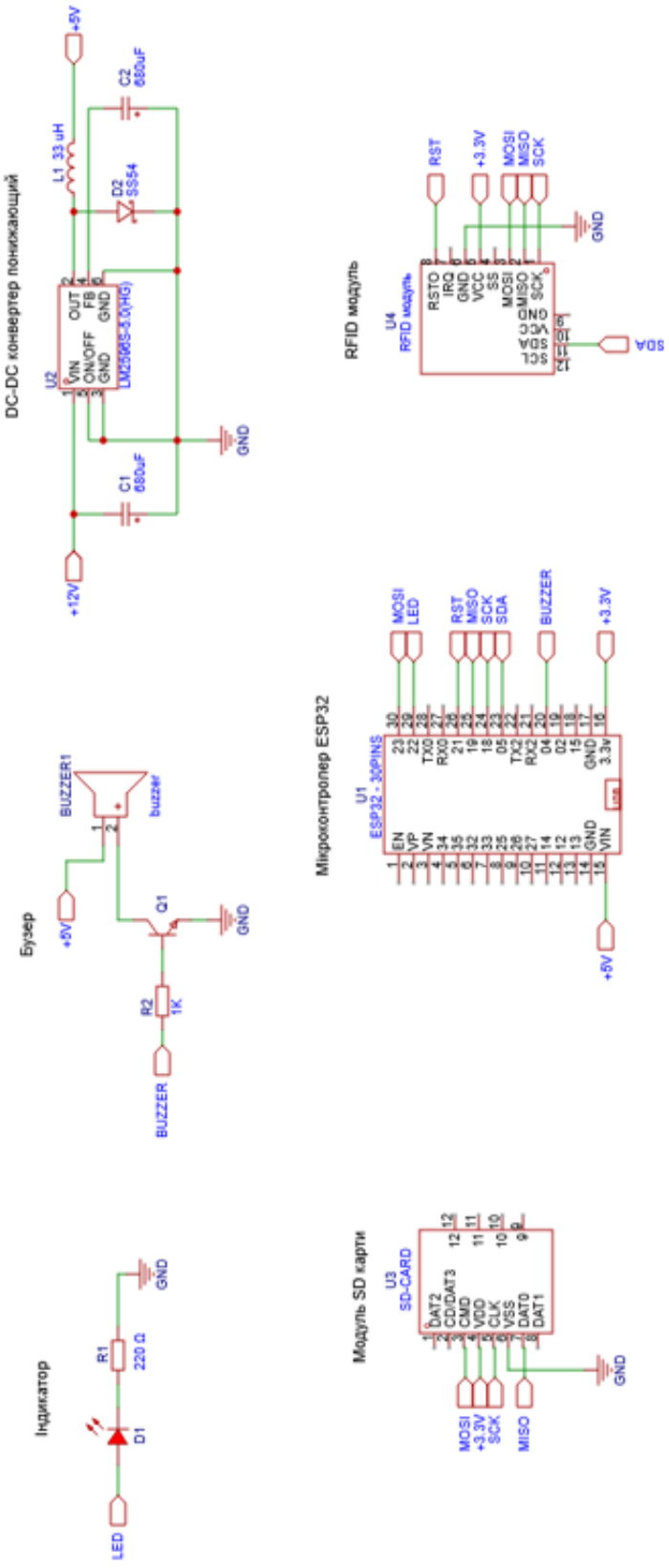


Рисунок 2.2 – Схема електрична проєктованого прграмно-технічного засобу доступу до складських приміщень

Блок живлення реалізований на основі імпульсного DC-DC перетворювача LM2596S-5.0 (U2) понижуючого типу. Вхідна напруга +12 В подається через електролітичний конденсатор C1 ємністю 680 мкФ, що забезпечує згладжування пульсацій вхідної напруги. Мікросхема LM2596S-5.0 є фіксованою версією перетворювача з вихідною напругою 5 В, тому зовнішній дільник напруги не потрібен. До виводу FB підключена відповідна ланцюг зворотного зв'язку. Індуктор L1 номіналом 33 мкГн забезпечує накопичення енергії у циклі перемикання. Діод SS54 (D2) типу Шотткі використовується як зворотній діод для зниження втрат комутації. На виході встановлений згладжуючий конденсатор C2 ємністю 680 мкФ. Вихідна напруга +5 В подається на мікроконтролер та бузер.

Модуль SD-карти (U3) підключений до тієї самої SPI-шини мікроконтролера через власний пін вибору CS. Живлення модуля здійснюється від шини +3.3 В. На модуль виведені лінії DAT0–DAT3, CMD та CLK відповідно до стандарту SD-карти, проте у режимі SPI використовуються лише лінії MOSI (DI), MISO (DO), SCK та CS. Наявність виводу CD/DAT3 забезпечує програмне виявлення вставленої картки. SD-карта використовується для зберігання файлів бази даних users.txt та журналу подій db.txt у форматі CSV, що забезпечує незалежне локальне сховище даних.

Блок звукової індикації складається з пасивного бузера BUZZER, підключеного до колектора транзистора Q1 типу NPN через живлення +5 В. База транзистора підключена до виходу мікроконтролера через струмообмежувальний резистор R2 номіналом 1 кОм, що забезпечує надійне керування транзистором без перевантаження виходу мікроконтролера. Транзисторний ключ необхідний оскільки пасивний бузер споживає струм, що перевищує допустимий вихідний струм піна ESP32. При подачі логічної одиниці транзистор відкривається і через бузер протікає струм, що генерує звуковий сигнал.

Блок світлової індикації складається зі світлодіода D1 та послідовно з'єднаного струмообмежувального резистора R1 номіналом 220 Ом. Анод світлодіода підключений до виходу LED мікроконтролера, катод – до загального проводу GND. Резистор 220 Ом обмежує струм через світлодіод на рівні близько 10 мА при напрузі живлення 3.3 В, що є оптимальним для стандартних індикаторних світлодіодів і забезпечує достатню яскравість без ризику виходу з ладу.

2.4 Аналіз та вибір апаратних компонентів

Вибір апаратних компонентів здійснювався за критеріями відповідності технічним вимогам, доступності на ринку України, наявності документації та бібліотек для інтеграції з платформою Arduino/ESP-IDF, а також економічної доцільності. Нижче наведено детальний аналіз кожного компонента системи.

Мікроконтролер ESP32 DevKit C v4 (U1) є основним обчислювальним елементом системи (рис. 2.3). Побудований на базі двоядерного процесора Xtensa LX6 з тактовою частотою до 240 МГц, ESP32 забезпечує значний обчислювальний ресурс для одночасної обробки SPI-периферії, WiFi-стеку та HTTP-клієнта. Ключовою характеристикою даного мікроконтролера є вбудований модуль WiFi стандарту 802.11 b/g/n з підтримкою протоколів TCP/IP, що дозволяє реалізувати мережеву взаємодію без додаткових компонентів. Обсяг Flash-пам'яті складає 4 МБ, оперативна пам'ять – 520 кБ SRAM. Мікроконтролер має 34 програмованих піни введення-виведення з підтримкою SPI, I2C, UART, PWM та ADC. Напруга живлення 3.3 В (вхід VIN підтримує 5 В через вбудований LDO-стабілізатор).

ESP32 був обраний як оптимальна платформа завдяки поєднанню вбудованого WiFi, достатнього числа пінів для підключення всієї периферії по шині SPI, широкої екосистеми бібліотек (MFRC522, SD, HTTPClient) та низької вартості.

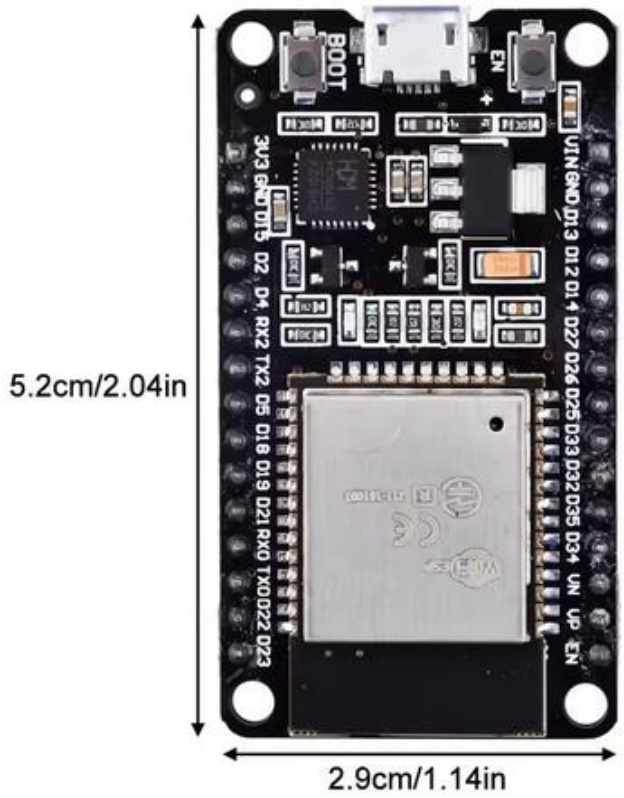


Рисунок 2.3 – Мікроконтролер ESP32 DevKit C v4

RFID-модуль на базі мікросхеми MFRC522 (U4) є спеціалізованим контролером зчитування безконтактних карток (рис. 2.4). Модуль працює на несучій частоті 13.56 МГц і підтримує картки стандартів ISO/IEC 14443A та ISO/IEC 14443B, зокрема широко розповсюджені картки типу Mifare Classic та Mifare Ultralight. Ефективна відстань зчитування становить до 50 мм і залежить від розміру антени та типу картки. Інтерфейсом підключення є SPI з максимальною тактовою частотою 10 МГц, що забезпечує швидкий обмін даними з мікроконтролером. Напруга живлення модуля 3.3 В, споживаний струм у режимі зчитування не більше 26 мА. Модуль містить вбудований крипто-процесор для роботи із захищеними секторами карток. Вибір MFRC522 обумовлений його надзвичайно широким розповсюдженням у системах контролю доступу початкового та середнього рівня, наявністю зрілої бібліотеки MFRC522 для Arduino з відкритим кодом, а також низькою вартістю модуля.

Рисунок 2.4 – RC522 RFID модуль

Модуль SD-карти (U3) реалізує інтерфейс між мікроконтролером та SD-карткою у режимі SPI (рис. 2.5). Модуль містить буферні схеми для узгодження рівнів напруги між логікою 3.3 В мікроконтролера та SD-карткою, а також виведений пін виявлення картки (CD). Підтримуються SD-картки стандартних форматів SDSC та SDHC з файловою системою FAT16/FAT32. Максимальна тактова частота SPI – 25 МГц. Живлення модуля 3.3 В або 5 В залежно від версії. Модуль SD-карти обраний як найпростіший та найдешевший спосіб додати енергонезалежне сховище даних до системи на базі ESP32, при цьому бібліотека SD з комплекту Arduino IDE забезпечує повноцінну роботу з файловою системою FAT.

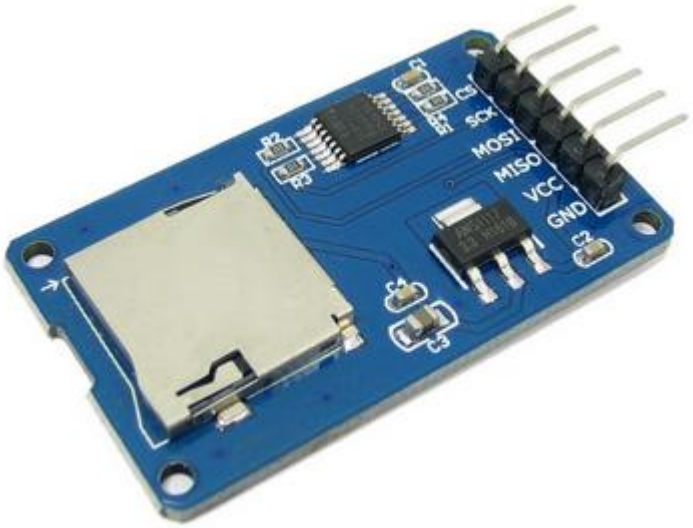


Рисунок 2.5 – RC522 RFID модуль

DC-DC перетворювач LM2596S-5.0 (U2) є імпульсним понижуючим регулятором напруги виробництва Texas Instruments (або сумісних виробників) (рис. 2.6). Мікросхема забезпечує вихідний струм до 3 А при вхідній напрузі від 4.5 до 40 В, що робить її придатною для живлення від акумуляторів 12 В або промислових джерел живлення. Фіксована версія LM2596S-5.0 забезпечує стабілізовану вихідну напругу 5 В з точністю ±4%. Частота перемикання – 150 кГц. ККД перетворювача у типовому режимі роботи складає 73–88% залежно від навантаження. На відміну від лінійних стабілізаторів, імпульсний принцип роботи забезпечує мінімальне тепловиділення при значному перепаді вхідної та вихідної напруг. LM2596S обраний через поширеність, доступність готових модулів з обв’язкою та надійну роботу у широкому діапазоні навантажень.

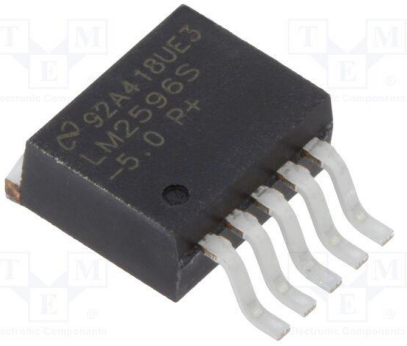


Рисунок 2.6 – DC-DC перетворювач LM2596S-5.0

Діод Шотткі SS54 (D2) використовується як зворотній (freewheeling) діод у схемі DC-DC перетворювача. Максимальний прямий струм діода – 5 А, максимальна зворотна напруга – 40 В. Завдяки технології Шотткі пряме падіння напруги складає лише 0.4–0.55 В при номінальному струмі, що значно знижує теплові втрати порівняно зі звичайними p-n діодами. Швидке відновлення (відсутність накопиченого заряду) забезпечує мінімальні комутаційні втрати на частоті 150 кГц. Конденсатори C1 та C2 ємністю 680 мкФ кожен є електролітичними і забезпечують відповідно фільтрацію вхідної напруги та згладжування вихідної напруги перетворювача. Індуктор L1 номіналом 33 мкГн є ключовим елементом, що накопичує та передає енергію у кожному циклі перемикання.

Транзистор Q1 типу NPN використовується як електронний ключ для керування пасивним бузером. Необхідність транзисторного ключа обумовлена тим, що максимальний вихідний струм одного піна ESP32 становить 40 мА, тоді як пасивний бузер може споживати до 50–80 мА, що перевищує допустиме навантаження на вивід мікроконтролера. Для реалізації ключа обрано транзистор 2N2222A у корпусі TO-92 — класичний NPN біполярний транзистор загального призначення з широкою доступністю та добре документованими характеристиками.

Основні технічні параметри транзистора 2N2222A: максимальний колекторний струм I_C = 600 мА, максимальна напруга колектор-емітер U_CE = 40 В, максимальна напруга колектор-база U_CB = 75 В, коефіцієнт підсилення за постійним струмом h_21E = 100–300 при I_C = 10 мА, напруга насичення колектор-емітер U_CE(sat) ≈ 0.3 В при I_C = 150 мА, максимальна розсіювана потужність P_max = 500 мВт.

Розрахунок режиму роботи транзисторного ключа виконується наступним чином. Струм через бузер при напрузі живлення +5 В та внутрішньому опорі бузера $R_{buz} \approx 100$ Ом складає: $I_C = \frac{U}{R_{buz}} = \frac{5}{100} = 50$мА.

Для надійного насичення транзистора необхідно забезпечити базовий струм з урахуванням коефіцієнта насичення $k_{sat} = 5 - 10$ (для гарантованого входу у насичення): $I_{\text{В}} = \frac{I_C}{h_{21E} \cdot k_{sat}} = \frac{50}{100 \cdot 0.1} = 5\text{мА}$ (мінімум).

Фактичний базовий струм при вихідній напрузі піна ESP32 $U_{\text{вих}} = 3{,}3$ В та напрузі база-емітер $U_{BE} = 0.7$ В: $I_{\text{В}} = \frac{U_{\text{вих}} - U_{BE}}{R_2} = \frac{3.3-0.7}{100} = 2{,}6\text{мА}$.

Отриманий базовий струм 2.6 мА h_{21E}=100 забезпечує колекторний струм до 260 мА у лінійному режимі, що значно перевищує необхідні 50 мА і гарантує насичення транзистора.

Струмообмежувальний резистор R_1=220 Ом захищає світлодіод D1 від надмірного струму. При напрузі живлення U=3.3 В (вихід піна ESP32) та типовому прямому падінні напруги на червоному світлодіоді U_{LED}=1.8–2.0 В струм через світлодіод складає: $I_{LED} = \frac{U - U_{LED}}{R_1} = \frac{3.3-1.9}{220} \approx 6.4$ мА. Отримане значення знаходиться в межах 5–20 мА, що є оптимальним діапазоном для стандартних індикаторних світлодіодів діаметром 3–5 мм і забезпечує достатню яскравість при мінімальному навантаженні на вивід мікроконтролера.

2.5 Орієнтовна вартість апаратних компонентів

Для оцінки економічної доцільності розробленого засобу складено кошторис апаратних компонентів на основі актуальних роздрібних цін на ринку України станом на 2026 рік. Ціни наведені у гривнях і є орієнтовними (реальна вартість може відрізнятися залежно від постачальника та курсу валют на момент розрахунку).

Орієнтовна вартість апаратних компонентів проєктованого програмно-технічного засобу контролю доступу до складських приміщень з RFID та веб сервером на платформі ESP32 наведено у таблиці 2.1.

Таблиця 2.1 – Орієнтовна вартість апаратних компонентів

Компонент	Кількість	Ціна за од. (грн)	Загальна вартість (грн)
ESP32 DevKit C v4	1	350	350
RFID-модуль MFRC522	1	120	120
Модуль SD-карти	1	85	85
DC-DC перетворювач LM2596S-5.0	1	95	95
Індуктор L1 33 мкГн	1	20	20
Діод SS54 (D2)	1	15	15
Конденсатор 680 мкФ (×2)	2	25	50
Бузер пасивний	1	40	40
Транзистор NPN (Q1)	1	12	12
Резистор 220 Ом (R1)	1	2	2
Резистор 1 кОм (R2)	1	2	2
Світлодіод червоний (D1)	1	5	5
SD-карта 8 ГБ	1	120	120
Корпус та монтажні матеріали	1	200	200
Разом			1116

Загальна орієнтовна вартість апаратних компонентів становить близько 1116 гривень, що є конкурентоспроможним показником порівняно з готовими комерційними рішеннями систем контролю доступу початкового рівня, вартість

яких, як правило, починається від 2000–3000 гривень без урахування програмного забезпечення та налаштування. Розроблений засіб забезпечує повний контроль над програмним кодом і можливість масштабування без додаткових ліцензійних витрат.

2.6 Проєктування програмної архітектури системи

Проєктування програмної частини системи охоплює три взаємопов'язані рішення: вибір протоколу передачі даних між мікроконтролером і сервером, визначення структури та формату зберігання даних, а також вибір серверної платформи. Кожне з цих рішень безпосередньо впливає на складність реалізації, надійність і можливість подальшого розширення системи.

Для передачі даних між ESP32 та серверною частиною обрано протокол HTTP з архітектурним стилем REST. Альтернативою розглядався протокол MQTT, який широко застосовується в IoT-системах і забезпечує низьке енергоспоживання та ефективну передачу повідомлень через брокер. Однак MQTT потребує окремого брокера (наприклад, Mosquitto або HiveMQ) як проміжного вузла, що ускладнює інфраструктуру. WebSocket також розглядався як варіант двостороннього з'єднання в реальному часі, проте для задачі контролю доступу постійне з'єднання не є необхідним – кожне сканування є атомарною подією. HTTP/REST натомість не потребує додаткової інфраструктури, підтримується бібліотекою HTTPClient з комплекту ESP32 Arduino SDK, добре документований і дозволяє легко тестувати ендпоінти через браузер або утиліту curl. Запит-відповідь модель цілком відповідає сценарію використання: ESP32 надсилає POST-запит з UID картки і отримує JSON-відповідь зі статусом доступу.

Для зберігання даних обрано файловий підхід на базі формату CSV з двома файлами на SD-картці та їх дзеркальними копіями на сервері. Файл users.txt містить два поля – uid та name і зберігає перелік авторизованих користувачів.

Файл db.txt є append-only журналом подій з полями timestamp, uid та status. Альтернативою розглядалося використання вбудованої SQLite або серверної реляційної СУБД. Проте файловий CSV-формат має суттєві переваги у контексті даного проєкту: він читається без спеціалізованого програмного забезпечення, безпосередньо підтримується стандартною бібліотекою SD на ESP32, не потребує встановлення СУБД на сервері і легко редагується вручну на етапі налагодження. Обсяг даних у системі контролю доступу для складу є відносно невеликим, тому відсутність індексів та складних запитів не є критичним обмеженням.

Серверна частина реалізована на мікрофреймворку Python Flask. Серед альтернатив розглядалися Node.js з Express та Python Django. Node.js з Express є продуктивним рішенням для асинхронних застосунків, однак для даного проєкту асинхронність не є критичною вимогою, а Python є більш звичним середовищем для швидкого прототипування наукових і технічних застосунків. Django, на відміну від Flask, є повнофункціональним фреймворком з вбудованою ORM, адмін-панеллю та системою міграцій, однак надлишковий для простого REST API з файловим сховищем. Flask забезпечує мінімальну кількість залежностей, просту маршрутизацію, вбудований шаблонізатор Jinja2 для генерації веб-інтерфейсу та розгортається єдиною командою без додаткового налаштування. Ці властивості роблять його оптимальним вибором для реалізації серверної частини прототипу системи контролю доступу.

Щодо клієнтської частини, то програмне забезпечення мікроконтролера ESP32 розроблено на мові C++ у середовищі Arduino IDE та організовано за принципом модульності: кожна функціональна підсистема реалізована у вигляді окремої функції з чітко визначеним інтерфейсом. Така організація коду спрощує тестування, налагодження та подальше розширення функціональності без ризику порушення роботи інших модулів.

Модуль ініціалізації виконується одноразово у функції setup() і забезпечує послідовний запуск усіх підсистем у строго визначеному порядку. Спочатку

ініціалізується шина SPI з явним зазначенням пінів SCK, MISO та MOSI, що усуває залежність від дефолтних налаштувань платформи. Потім деактивується пін CS RFID-модуля (переводиться у стан HIGH) для запобігання конфліктам на шині під час ініціалізації SD-карти. Після успішного монтування файлової системи SD ініціалізується RFID-зчитувач, встановлюється WiFi-з'єднання та виводиться поточний вміст бази авторизованих користувачів.

Модуль мережевої взаємодії реалізує функцію sendScan(), яка формує HTTP POST-запит до серверного ендпоінта /scan і передає UID зчитаної картки. Функція обробляє відповідь сервера у форматі JSON та повертає рядок зі статусом доступу. Передбачена обробка помилок: при недоступності сервера або мережевому збої функція повертає значення error, після чого основний цикл автоматично перемикається на локальну перевірку за файлом users.txt на SD-картці. Така двоступенева архітектура забезпечує відмовостійкість системи при нестабільному мережевому з'єднанні.

Основний цикл loop() реалізує скінченний автомат з двома станами: очікування картки та обробка сканування. У стані очікування мікроконтролер безперервно опитує RFID-зчитувач з мінімальною затримкою 10 мс. При виявленні нової картки зчитується її серійний номер, виконується перевірка авторизації (серверна або локальна), формується звуковий та світловий сигнал відповідно до результату, подія реєструється у журналі db.txt, після чого система повертається у стан очікування. Такий підхід забезпечує швидкодію та мінімальне споживання ресурсів у режимі очікування.

2.7 Висновки до другого розділу

У другому розділі сформовано перелік функціональних і технічних вимог до програмно-технічного засобу контролю доступу, що охоплюють автоматизовану ідентифікацію персоналу за RFID-картками стандарту ISO/IEC 14443A, резервну офлайн-авторизацію на основі локальної копії бази даних,

обов’язкову реєстрацію усіх подій сканування, адміністрування через веб-інтерфейс без втручання у програмний код, а також технічні вимоги до живлення, індикації та часу реакції системи. Розроблено та детально описано електричну принципову схему пристрою у середовищі EasyEDA, що включає п’ять функціональних блоків: мікроконтролер ESP32 як центральний обчислювальний елемент, RFID-модуль MFRC522 для зчитування безконтактних карток, модуль SD-карти для локального зберігання даних, блок індикації зі світлодіодом і бузером та імпульсний блок живлення на базі мікросхеми LM2596S-5.0 з вхідною напругою 12 В. Проведено детальний аналіз і обґрунтовано вибір кожного апаратного компонента з наведенням ключових технічних характеристик, зокрема параметрів SPI-інтерфейсу, частотного діапазону RFID-зчитувача, ККД імпульсного перетворювача та електричних характеристик транзисторного ключа. Складено орієнтовний кошторис апаратної частини, загальна вартість якої становить близько 1116 гривень, що підтверджує економічну доцільність власної розробки порівняно з готовими комерційними рішеннями систем контролю доступу початкового рівня. Обґрунтовано вибір протоколу HTTP/REST для передачі даних між мікроконтролером і сервером, визначено структуру файлового сховища даних у форматі CSV та обрано мікрофреймворк Flask як серверну платформу з порівняльним аналізом альтернатив. Описано архітектуру програмного забезпечення мікроконтролера з обґрунтуванням модульної організації коду, порядку ініціалізації периферії та логіки двоступеневої авторизації із автоматичним перемиканням між серверним і локальним режимами перевірки.

3 СИМУЛЯЦІЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСОБУ КОНТРОЛЮ ДОСТУПУ ДО СКЛАДСЬКИХ ПРИМІЩЕНЬ З RFID ТА ВЕБ СЕРВЕРОМ НА ПЛАТФОРМІ ESP32

3.1 Загальна схема реалізації програмно-технічного засобу

З метою виконання перевірки ефективності і функціональності запропонованого програмно-технічного засобу контролю доступу до складських приміщень було проведена його реалізація. Оскільки фізичне розгортання апаратного комплексу не завжди є можливим на етапі розробки та тестування, для перевірки коректності алгоритмів і архітектурних рішень використовувався комплекс інструментів, що дозволив відтворити повний функціональний ланцюжок системи без реального обладнання.

Загальна схема реалізації системи охоплює три ключові компоненти: апаратну частину у вигляді симульованого мікроконтролера ESP32 з підключеними модулями RFID та SD-карти, мережевий тунель для забезпечення публічної доступності локального сервера, а також серверну частину у вигляді веб-застосунку на базі Python Flask. Взаємодія між компонентами відбувається через протокол HTTP, що забезпечує стандартизований обмін даними та можливість подальшого масштабування системи.

Для симуляції апаратної частини було використано онлайн-платформу Wokwi. Це є спеціалізованим середовищем для симуляції мікроконтролерів Arduino та ESP32, яке підтримує широкий спектр периферійних пристроїв, зокрема RFID-модуль MFRC522 та модуль SD-карти. Платформа дозволяє виконувати реальний код мікроконтролера у браузері, з повноцінною підтримкою бібліотек SPI, SD та WiFi, що робить її придатною для тестування комунікаційних протоколів між вбудованою системою та зовнішнім сервером.

Для забезпечення мережевої взаємодії між симульованим ESP32 та локальним сервером використовується утиліта ngrok, яка створює захищений HTTP-тунель між локальним портом машини розробника та публічною адресою

в мережі Інтернет. Це дозволяє симульованому мікроконтролеру, що працює у хмарному середовищі Wokwi, надсилати HTTP-запити на публічний URL, який прозоро перенаправляє трафік до локального Flask-сервера. Загальна схема пайплайну наведена на рисунку 3.1.

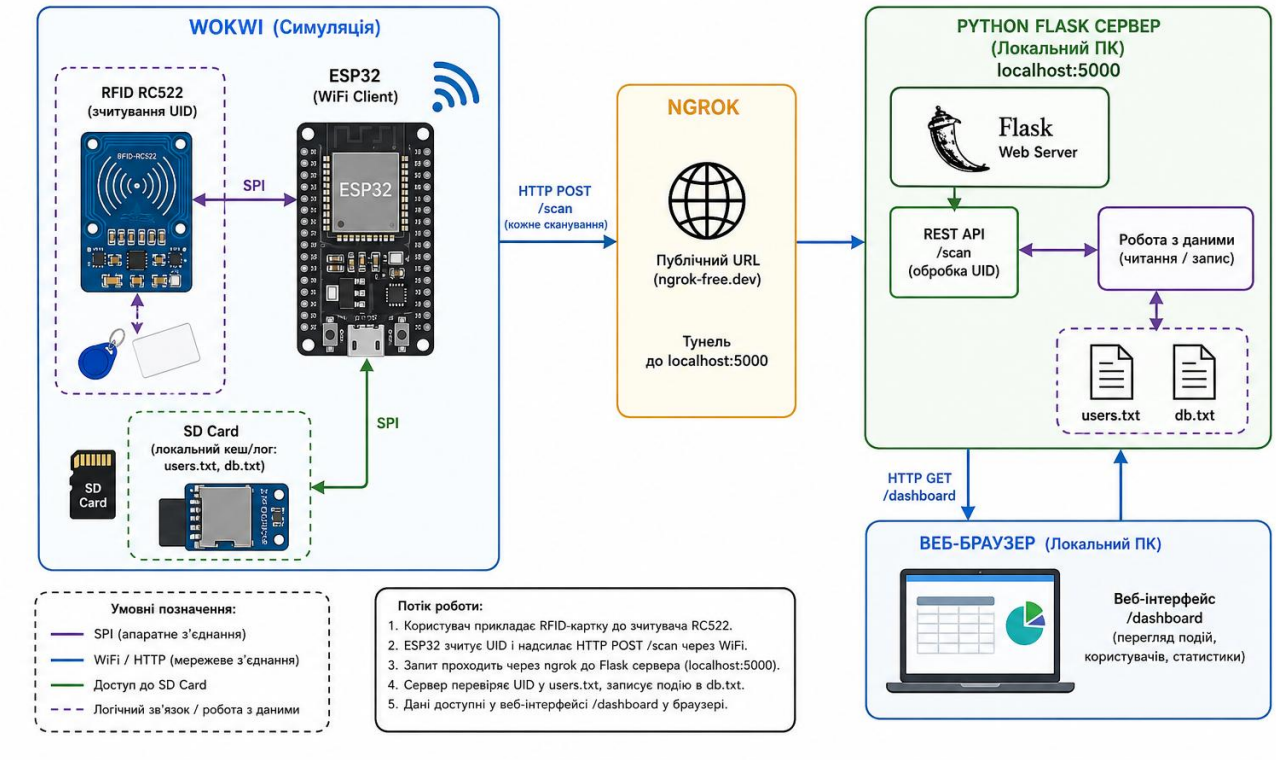


Рисунок 3.1 – Схема симуляції програмно-технічного засобу контролю доступу до складських приміщень за RFID та веб сервером на платформі ESP32

Потік даних у створеній симуляційні системі є лінійним та чітко визначеним. Після зчитування RFID-картки модулем RC522 мікроконтролер ESP32 формує HTTP POST-запит, що містить унікальний ідентифікатор картки (UID) у форматі шістнадцяткового рядка, та надсилає його на ендпоінт /scan публічного ngrok-URL. Сервер Flask обробляє запит, звіряє отриманий UID зі списком авторизованих користувачів у файлі users.txt, фіксує подію у журналі db.txt та повертає JSON-відповідь зі статусом доступу (allowed або unknown) і іменем користувача. Мікроконтролер отримує відповідь та відображає результат

у послідовному моніторі, а також дублює запис у локальний файл на SD-картці як резервну копію журналу подій.

3.2 Реалізація апаратної частини у Wokwi

Wokwi – це сучасна хмарна платформа для моделювання електронних схем та вбудованих систем, яка дозволяє виконувати розробку та тестування пристроїв безпосередньо у веб-браузері. На відміну від традиційних симуляторів, орієнтованих суто на аналогові процеси, Wokwi базується на поцикловій емуляції архітектур мікроконтролерів (AVR, ESP32, ARM) за допомогою технологій WebAssembly. Це забезпечує високу точність виконання програмного коду (firmware) у реальному часі, дозволяючи тестувати складні алгоритми взаємодії з периферією без використання фізичного обладнання.

Технічною особливістю середовища є підтримка цифрових інтерфейсів передачі даних, таких як SPI, I2C та UART, а також наявність вбудованого віртуального логічного аналізатора для моніторингу сигналів на пінах мікроконтролера. Платформа дозволяє гнучко налаштовувати конфігурацію компонентів через JSON-файли, що дає можливість створювати точні схеми з’єднань та перевіряти цілісність логічних рівнів сигналів. Використання Wokwi на етапі проектування дозволяє значно скоротити час на налагодження апаратної частини та уникнути помилок при монтажі реального прототипу.

Вибір середовища Wokwi для реалізації проекту, порівняно з іншими популярними платформами, такими як Autodesk Tinkercad, обумовлений декількома ключовими факторами:

– підтримка мікроконтролерів ESP32. Tinkercad орієнтований переважно на початковий рівень і базові плати Arduino (архітектура AVR). Wokwi, у свою чергу, забезпечує повноцінну підтримку родини ESP32, що є критичним для даного проекту через необхідність використання більшого об’єму пам’яті та

високої тактової частоти для роботи з файловими системами та RFID-інтерфейсами;

– розширена бібліотека складних компонентів. На відміну від Tinkercad, де номенклатура компонентів обмежена базовими датчиками, Wokwi дозволяє моделювати специфічні цифрові модулі, такі як RFID-зчитувач MFRC522 та модулі SD-карт. Це дає змогу перевірити логіку роботи протоколу SPI у складній конфігурації з декількома веденими (slave) пристроями.

– точність емуляції та програмна гнучкість. Wokwi використовує сучасний рушій, що дозволяє завантажувати будь-які стандартні бібліотеки Arduino IDE або ESP-IDF без необхідності їх адаптації. У Tinkercad часто виникають проблеми з несумісністю складних бібліотек через спрощену модель емуляції процесора.

– інструменти налагодження. Wokwi надає доступ до віртуального логічного аналізатора, що дозволяє бачити реальні таймінги передачі пакетів по шині SPI. Це дозволяє виявити колізії або помилки у виборі CS-пінів, які практично неможливо відстежити в Tinkercad.

– робота з файловою системою. Платформа Wokwi підтримує створення віртуальних образів SD-карт (FAT12/FAT16/FAT32), що дозволяє тестувати реальний запис та читання файлів під час симуляції, що є недоступним у більшості інших хмарних симуляторів.

Апаратна частина системи реалізована у симуляторі Wokwi у вигляді схеми, що включає мікроконтролер ESP32 DevKit C v4, RFID-модуль MFRC522 та модуль SD-карти. Усі три компоненти підключені до спільної шини SPI мікроконтролера, проте кожен пристрій має власний пін вибору мікросхеми (Chip Select): GPIO 5 для RFID-модуля та GPIO 4 для SD-карти (рис. 3.2). Спільне використання шини SPI при чіткому керуванні CS-пінами є стандартною практикою у вбудованих системах і дозволяє скоротити кількість використаних пінів мікроконтролера. З’єднання компонентів наведено на таблиці 3.1.

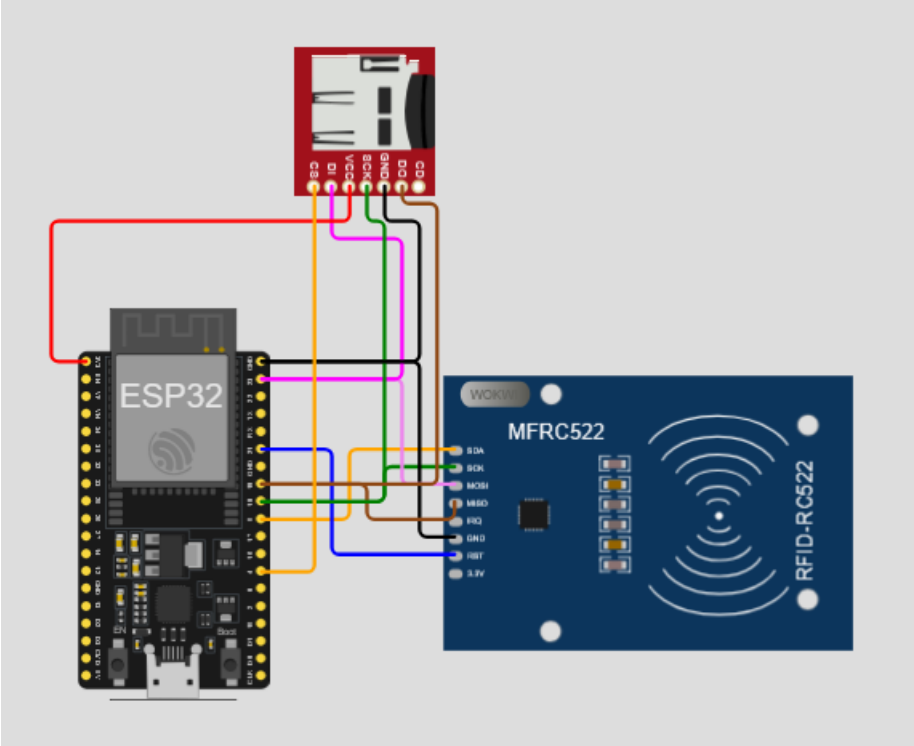


Рисунок 3.2 – Схема з’єднань компонентів у Wokwi


```
{
  "version": 1,
  "author": "Владислав Незгода",
  "editor": "wokwi",
  "parts": [
    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": -38.4, "left": 100.84, "at
    { "type": "board-mfrc522", "id": "rfid1", "top": -2.65, "left": 301.04, "attrs": {}
    {
      "type": "wokwi-microsd-card",
      "id": "sd1",
      "top": -179.83,
      "left": 217.87,
      "rotate": 90,
      "attrs": {}
    }
  ],
  "connections": [
    [ "esp:TX", "$serialMonitor:RX", "", [] ],
    [ "esp:RX", "$serialMonitor:TX", "", [] ],
    [ "esp:18", "rfid1:SCK", "green", [ "h67.2", "v-19.2" ] ],
    [ "rfid1:SDA", "esp:5", "orange", [ "h-57.6", "v38.4" ] ],
    [ "esp:GND.2", "rfid1:GND", "black", [ "h86.4", "v96" ] ],
    [ "rfid1:RST", "esp:21", "blue", [ "h-67.2", "v-57.6" ] ],
    [ "esp:19", "rfid1:MISO", "#8f4814", [ "h57.6", "v19.2", "h48" ] ],
    [ "esp:23", "rfid1:MOSI", "violet", [ "h76.8", "v57.6" ] ],
    [ "esp:23", "sd1:DI", "magenta", [ "h76.8", "v-76.8", "h-38.49" ] ],
    [ "esp:3V3", "sd1:VCC", "red", [ "h-19.05", "v-76.8", "h163.34" ] ],
    [ "esp:18", "sd1:SCK", "green", [ "h67.2", "v-153.6", "h-9.61" ] ],
    [ "sd1:GND", "esp:GND.2", "black", [ "v19.2", "h19.09", "v76.8" ] ],
    [ "esp:19", "sd1:DO", "#8f4814", [ "h96", "v-153.6", "h-19.09" ] ],
    [ "esp:4", "sd1:CS", "orange", [ "h0" ] ]
  ],
  "dependencies": {}
}
```


Рисунок 3.3 – Вміст файлу diagram.json у Wokwi

Таблиця 3.1 – З’єднання компонентів

№	Компонент 1	Пін	Компонент 2	Пін	Інтерфейс / Призначення
1	ESP32	GPIO18	RFID RC522	SCK	SPI (Clock)
2	ESP32	GPIO23	RFID RC522	MOSI	SPI (Master → Slave)
3	ESP32	GPIO19	RFID RC522	MISO	SPI (Slave → Master)
4	ESP32	GPIO5	RFID RC522	SDA (SS)	Chip Select RFID
5	ESP32	GPIO21	RFID RC522	RST	Reset
6	ESP32	GND	RFID RC522	GND	Живлення
7	ESP32	GPIO18	MicroSD	SCK	SPI (Clock)
8	ESP32	GPIO23	MicroSD	DI (MOSI)	SPI (Master → Slave)
9	ESP32	GPIO19	MicroSD	DO (MISO)	SPI (Slave → Master)
10	ESP32	GPIO4	MicroSD	CS	Chip Select SD
11	ESP32	3.3V	MicroSD	VCC	Живлення
12	ESP32	GND	MicroSD	GND	Живлення

Ініціалізація периферії у коді мікроконтролера виконувалась у строго визначеному порядку з метою уникнення конфліктів на шині. Спочатку пін CS RFID-модуля програмно переводився у стан HIGH, що деактивує пристрій на шині, після чого виконувалась ініціалізація SD-карти. Лише після успішного завершення ініціалізації SD запускається ініціалізація RFID-модуля через виклик PCD_Init(). Такий порядок є критично важливим, оскільки спроба ініціалізувати обидва пристрої одночасно призводить до колізій на шині SPI та відмови ініціалізації SD-карти.

Програмна логіка мікроконтролера реалізує наступний алгоритм роботи. У головному циклі loop() система безперервно опитує RFID-зчитувач на наявність нової картки. При виявленні картки зчитується її серійний номер, що перетворюється на рядок у форматі XX:XX:XX:XX з використанням шістнадцяткового кодування. Отриманий UID надсилається на Flask-сервер через HTTP POST-запит. У разі недоступності сервера або відсутності з'єднання з WiFi система автоматично перемикається на локальну перевірку: UID порівнюється зі списком дозволених користувачів у файлі users.txt на SD-картці. Результат кожного сканування незалежно від джерела авторизації фіксується у файлі db.txt у форматі CSV з полями timestamp, uid та status.

Однією із головних функцій є реалізація передач ідентифікатора RFID-мітки з мікроконтролера ESP32 на віддалений сервер та обробку відповіді. На початковому етапі перевіряється наявність підключення до мережі Wi-Fi, і у разі його відсутності функція завершує роботу з поверненням статусу помилки. Якщо з'єднання встановлено, формується HTTP POST-запит до серверного ресурсу /scan із передачею UID у тілі запиту у форматі application/x-www-form-urlencoded, а також додається службовий заголовок для коректної роботи через тунель ngrok. Після надсилання запиту аналізується код відповіді сервера: у випадку успішного виконання (код 200) зчитується тіло відповіді та виконується пошук ключових значень, що визначають статус доступу (дозволено або невідомо), і відповідне значення повертається як результат роботи функції. У разі помилки запиту або некоректної відповіді повертається статус "error". Перед завершенням функції HTTP-з'єднання коректно закривається, що забезпечує стабільність роботи системи та ефективне використання ресурсів. Алгоритм взаємодії ESP32 із сервером наведено на рис. 3.4.

WiFi-підключення у Wokwi реалізувалось через гостьову мережу Wokwi-GUEST, яка надається симулятором автоматично та забезпечує доступ симульованого ESP32 до мережі Інтернет. Це дозволяє мікроконтролеру виконувати реальні HTTP-запити до зовнішніх сервісів, зокрема до публічного

ngrok-URL, без будь-яких додаткових налаштувань мережевої інфраструктури. Слід зазначити, що симулятор Wokwi підтримує лише HTTP (без шифрування TLS) для вихідних запитів у безкоштовному режимі, тому URL ngrok-тунелю використовується з протоколом HTTP.

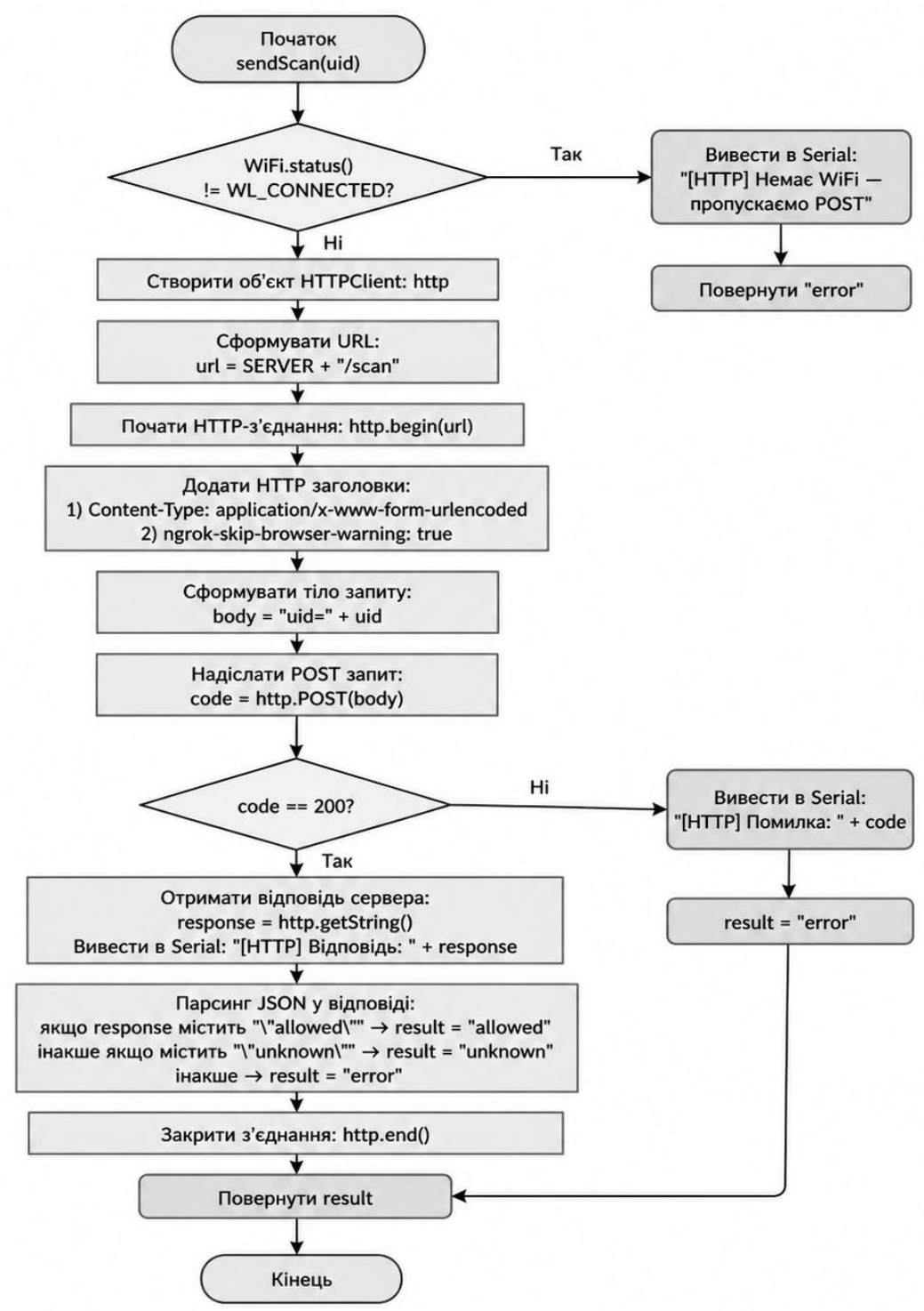


Рисунок 3.4 – Алгоритм взаємодії ESP32 із сервером

Конфігурація схеми у Wokwi описується файлом diagram.json, що містить перелік компонентів та їх з'єднань. Цей підхід дозволяє відтворити схему на будь-якому пристрої без ручного розміщення компонентів, а також є зручним для версійного контролю у системах на зразок Git. Бібліотеки MFRC522 та SD, що використовуються у коді, підтримуються Wokwi нативно і не потребують ручної інсталяції в середовищі симулятора.

3.3 Реалізація серверної частини

Серверна частина системи реалізована у вигляді веб-застосунку на основі мікрофреймворку Flask мови програмування Python. Flask було обрано з огляду на його простоту розгортання, мінімальні залежності та достатню функціональність для реалізації REST API у прототипному середовищі. Сервер запускається локально на порту 5000 та надає набір HTTP-ендпоінтів для обробки запитів від мікроконтролера і взаємодії з веб-інтерфейсом.

Зберігання даних організовано у вигляді двох текстових файлів у форматі CSV. Файл users.txt містить список авторизованих користувачів з полями uid та name, де uid є унікальним ідентифікатором RFID-картки, а name – ім'ям власника. Файл db.txt є журналом усіх подій сканування і містить поля timestamp, uid та status, де timestamp це Unix-час події, а status приймає значення allowed або unknown залежно від результату перевірки. Такий підхід до зберігання даних забезпечує простоту читання і редагування файлів вручну, що є зручним на етапі розробки та тестування.

Основним ендпоінтом для взаємодії з мікроконтролером є POST /scan, який приймає параметр uid у тілі запиту у форматі application/x-www-form-urlencoded. При отриманні запиту сервер виконує пошук переданого UID у файлі users.txt, визначає статус доступу та записує подію у db.txt. Відповідь повертається у форматі JSON і містить поля status та name. Окрім основного ендпоінта, сервер реалізує ендпоінти GET /users та GET /db для отримання

повного вмісту відповідних файлів, POST /add для додавання нового користувача та POST /delete для його видалення за UID.

Веб-інтерфейс системи доступний за адресою /dashboard і реалізований у вигляді HTML-сторінки, що генерується на стороні сервера засобами шаблонізатора Jinja2, вбудованого у Flask. Інтерфейс відображає поточний список авторизованих користувачів з можливістю їх видалення, форму для додавання нового користувача, а також таблицю останніх двадцяти подій сканування з відображенням часу, UID та статусу доступу. Статус allowed відображається зеленим кольором, unknown – червоним, що забезпечує наочність при моніторингу подій у реальному часі.

Публічна доступність локального сервера забезпечується засобами утиліти ngrok, що встановлюється окремо та запускається паралельно з Flask. Ngrok створює стійкий HTTP-тунель між локальним портом 5000 та публічною адресою виду http://[ідентифікатор].ngrok-free.app, яка залишається незмінною протягом усієї сесії роботи тунелю. Ця адреса прописується у коді мікроконтролера як цільовий сервер для HTTP-запитів, що дозволяє симульованому ESP32 у хмарному середовищі Wokwi взаємодіяти з локальним Flask-сервером так само, як це робило б реальне фізичне обладнання у виробничому середовищі.

Таким чином, було реалізовано схему симуляції, яка повністю відтворює функціональну поведінку кінцевої системи, починаючи від зчитування RFID-картки та передачі даних через мережу до серверної обробки, авторизації та відображення результатів у веб-інтерфейсі.

3.4 Тестування програмно-технічного засобу

Тестування системи виконувалося покроково відповідно до функціональних сценаріїв: запуск інфраструктури (Flask-сервер та ngrok-тунелю), ініціалізація симуляції у Wokwi, перевірка реєстрації невідомих карток,

додавання авторизованих користувачів та підтвердження коректності надання доступу. Кожен етап супроводжувався фіксацією стану системи через знімки екрану Serial-монітора, веб-інтерфейсу та термінального виводу ngrok.

Першим кроком тестування є перевірка початкового стану веб-інтерфейсу. На рисунку 3.5 зображено дашборд одразу після запуску Flask-сервера: список користувачів порожній, журнал сканувань не містить жодного запису. Форма додавання користувача вже містить попередньо заповнені поля – UID та ім'я, проте запис до бази ще не було здійснено. Цей стан підтверджує коректну ініціалізацію файлів users.txt та db.txt при першому запуску сервера.

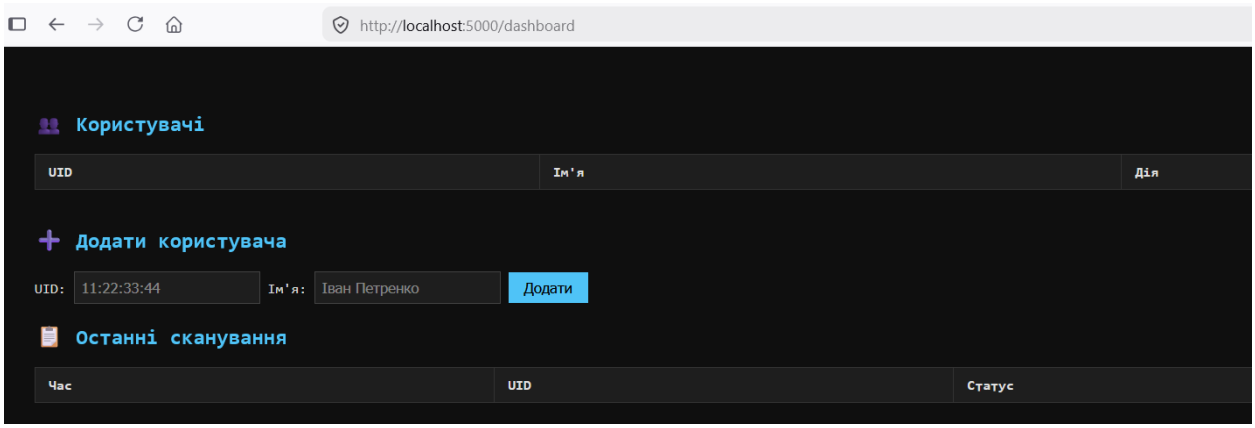


Рисунок 3.5 – Початковий стан веб-інтерфейсу /dashboard

Паралельно з Flask-сервером у окремому терміналі запускалась утиліта ngrok. На рисунку 3.6 наведено термінальний вивід ngrok одразу після встановлення тунелю (можна відмітити метадані статус сесії, регіон та затримку).

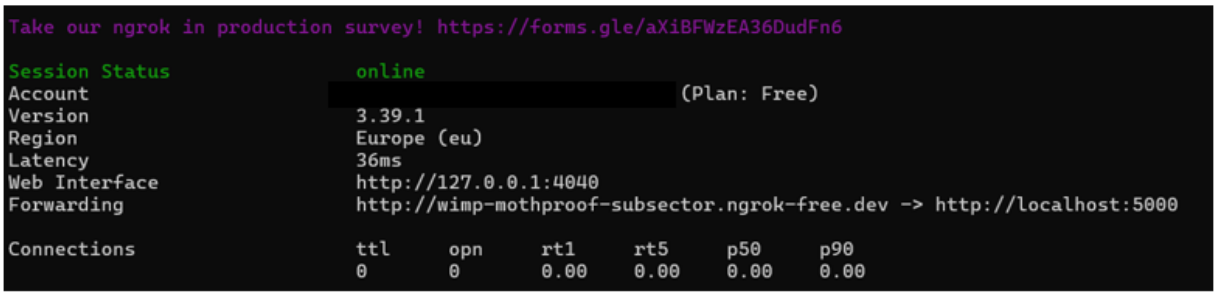


Рисунок 3.6 – Термінальний вивід ngrok після встановлення тунелю

Рядок Forwarding підтверджує, що адреса http://wimp-mothproof-subsector.ngrok-free.dev прозоро перенаправляє вхідний трафік на локальний порт 5000. На цьому етапі лічильник з'єднань ще нульовий, оскільки симуляція Wokwi не запущена.

Наступним кроком був запуск симуляції у Wokwi. На рисунку 3.7 зображено середовище симулятора зі схемою та Serial-монітором у нижній частині екрана. Після успішної ініціалізації SD-карти та RFID-модуля мікроконтролер підключався до гостьової WiFi-мережі Wokwi-GUEST і отримував IP-адресу 10.10.0.2. Після встановлення з'єднання у послідовному моніторі виводилось поточний вміст файлу users.txt, де присутній один запис – 11:22:33:44,TestUser. Система переходила у режим очікування картки з повідомленням «[READY] Прикладіть картку...».

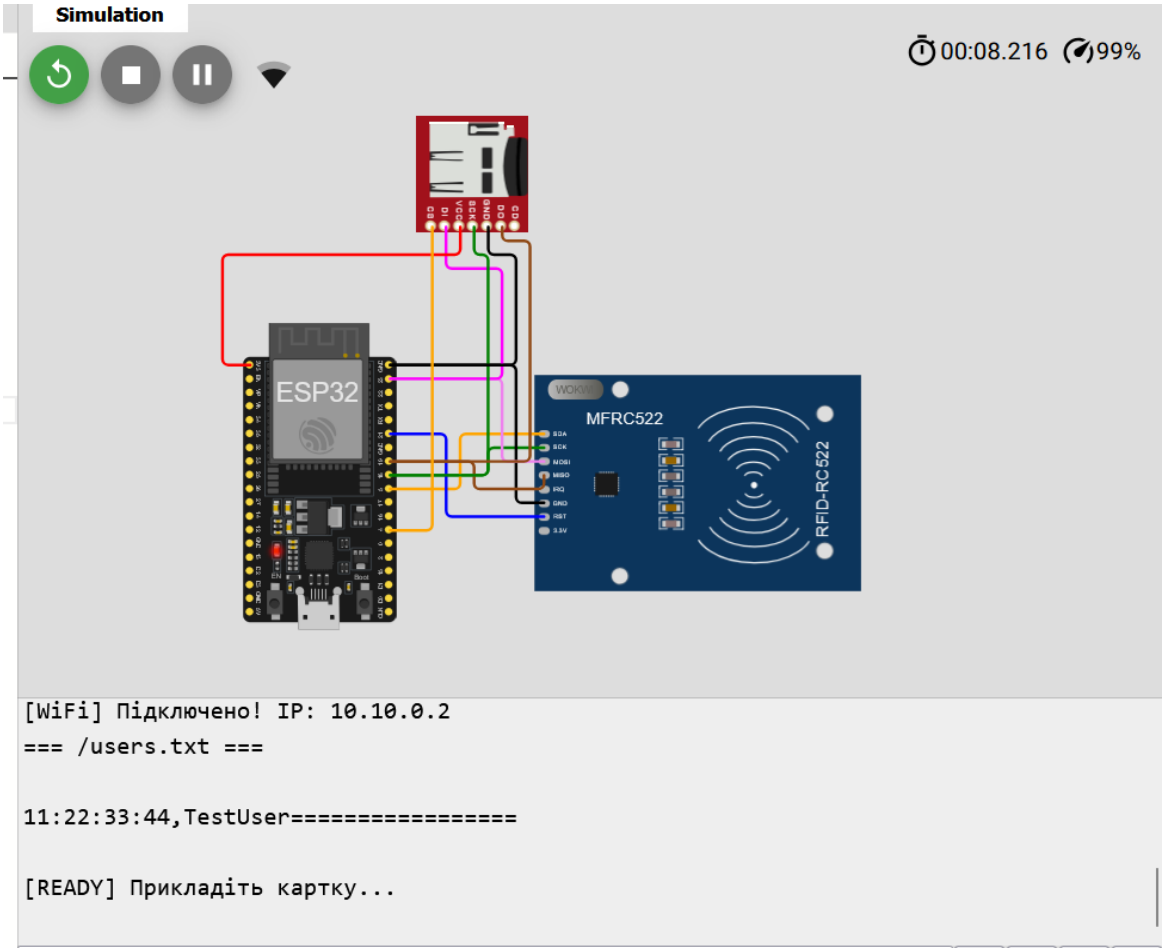


Рисунок 3.7 – Запуск симуляції Wokwi, успішна ініціалізація та підключення до WiFi

Перший сценарій тестування передбачав сканування невідомої картки, тобто такої, UID якої відсутній у файлі users.txt. На рисунку 3.8 веб-інтерфейс відображає результат першого зафіксованого сканування: UID 01:02:03:04 з часовою міткою 1777787321 та статусом unknown, позначеним червоним кольором. Список авторизованих користувачів залишається порожнім, що підтверджує коректну роботу логіки перевірки: якщо UID не знайдено у users.txt, система присвоює події статус unknown та фіксує її у журналі db.txt без надання доступу.

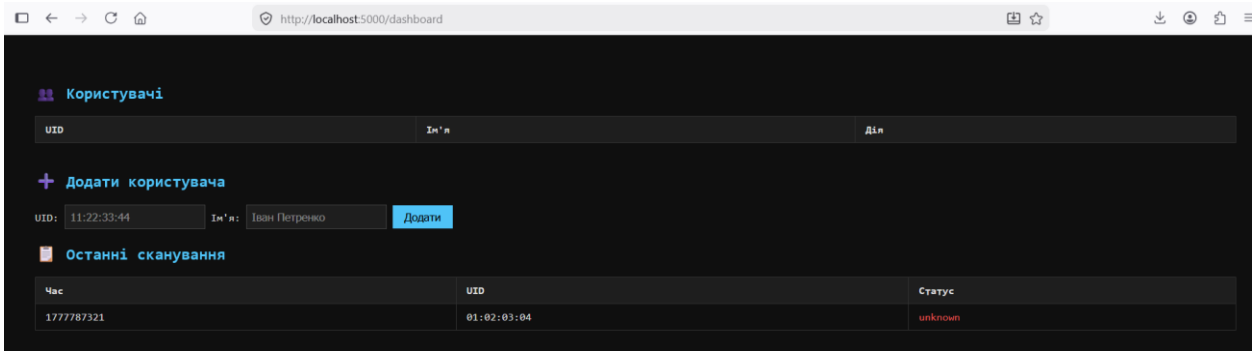


Рисунок 3.8 – Дашборд після першого сканування невідомої картки

Адміністративна панель ngrok також підтверджувала реєстрацію веб-інтерфейсу. На рисунку 3.9 зображено сторінку Endpoints & Traffic Policy хмарної консолі ngrok: ендпоінт http://wimp-mothproof-subsector.ngrok-free.dev відображається як активний зі статусом Public, агент версії 3.39.1 та регіоном Europe. Це підтверджує, що тунель функціонує коректно і є публічно доступним з мережі Інтернет.

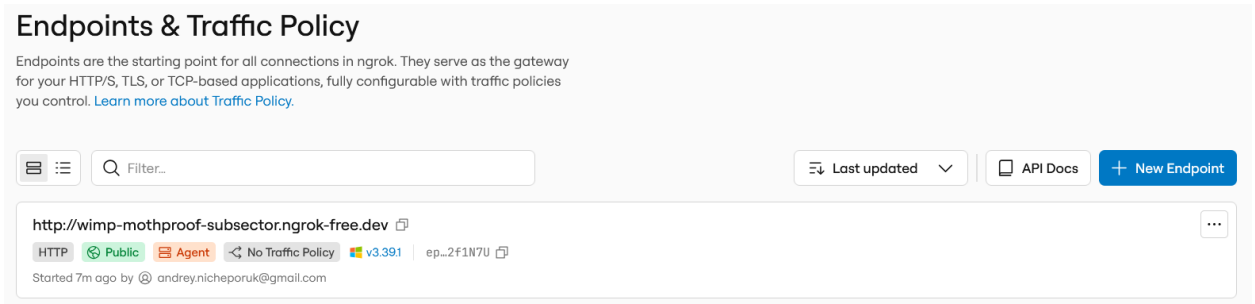


Рисунок 3.9 – Адміністративна консоль ngrok з активним публічним ендпоінтом

На рисунку 3.10 наведено термінальний вивід ngrok після декількох сканувань карток симульованим ESP32. У секції HTTP Requests зафіксовано три успішні POST-запити на ендпоінт /scan з відповіддю 200 OK, надісланих о 08:48:40, 08:49:11 та 08:50:00 за київським часом. Наявність коду 200 для кожного запиту підтверджує, що Flask-сервер коректно обробляв вхідні запити від мікроконтролера та повертав валідні JSON-відповіді. Лічильник загальних з'єднань (ttl) становив 3, що відповідає кількості зафіксованих запитів.

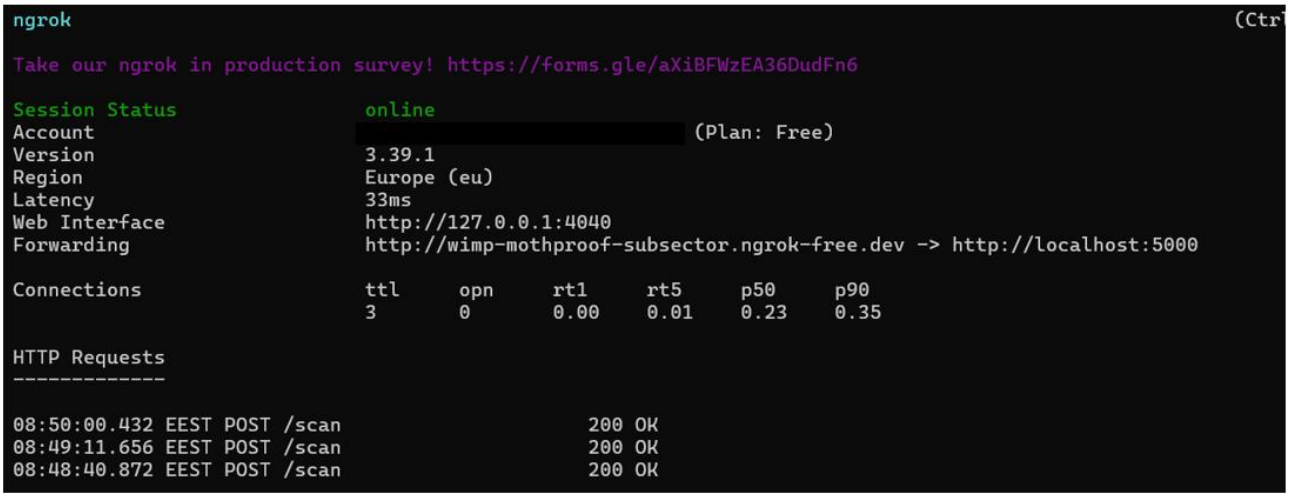


Рисунок 3.10 – Термінал ngrok з журналом HTTP-запитів від ESP32

Після накопичення декількох подій сканування веб-інтерфейс відображає повну таблицю журналу. На рисунку 3.11 дашборд показував п'ять записів у розділі «Останні сканування»: картки з UID 01:02:03:04 та 11:22:33:44 сканувалися по декілька разів, і всі події отримали статус unknown. Список авторизованих користувачів залишався порожнім, що підтверджує, що жодну картку ще не було зареєстровано у системі як авторизовану. Всі статуси відображаються червоним кольором відповідно до логіки кольорового маркування інтерфейсу.

Для перевірки функціональності REST API щодо додавання користувачів було виконано POST-запит через утиліту curl. На рисунку 3.11 зображено команду, надіслану безпосередньо до локального Flask-сервера: POST /add з параметрами uid=55:66:77:88 та name=TestUser2. Використання curl дозволило

перевірити роботу API незалежно від веб-інтерфейсу та підтвердити, що ендпоінт коректно приймає запити у форматі application/x-www-form-urlencoded.

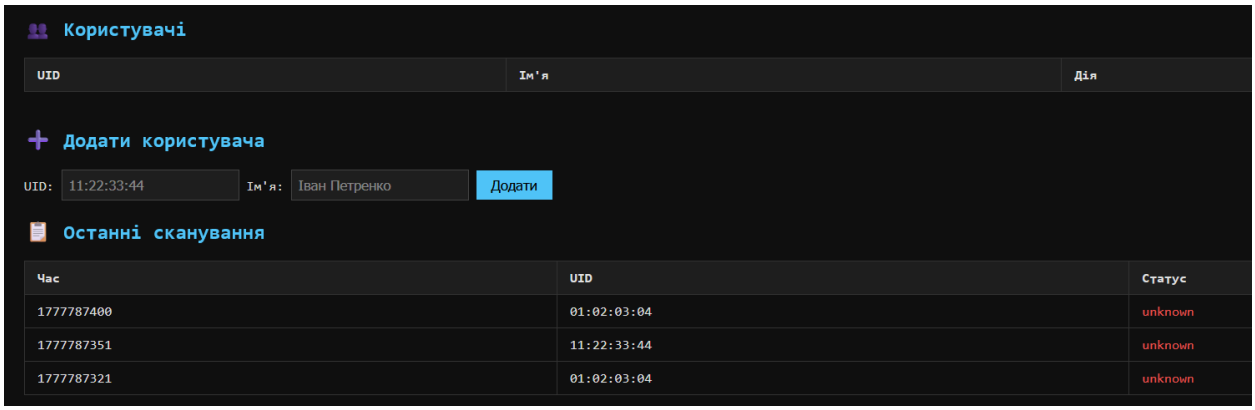


Рисунок 3.11 – Журнал сканувань з кількома подіями зі статусом unknown

```
>curl -X POST http://localhost:5000/add -d "uid=55:66:77:88&name=TestUser2"
```

Рисунок 3.12 – Додавання нового користувача через curl POST-запит

Результат виконання команди curl підтверджується запитом до ендпоінта GET /users. На рисунку 3.13 браузер відображає JSON-відповідь сервера, що містить один об’єкт з полями name зі значенням «TestUser2» та uid зі значенням «55:66:77:88». Коректна структура відповіді підтверджує, що запис був успішно збережений у файлі users.txt та повертається сервером у належному форматі для подальшої обробки клієнтами.

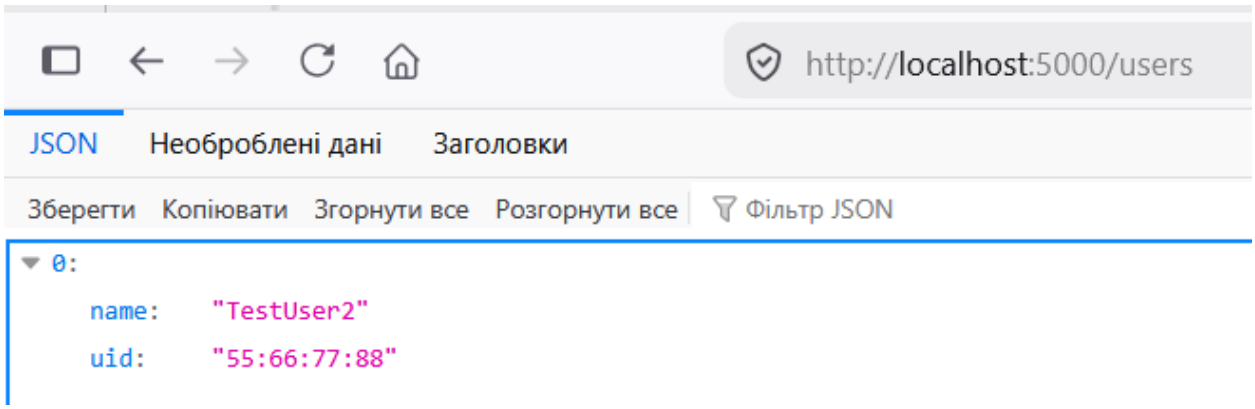


Рисунок 3.13 – JSON-відповідь ендпоінта /users після додавання користувача

Після реєстрації користувача з UID 55:66:77:88 у системі було виконано повторне сканування відповідної картки у Wokwi. На рисунку 3.14 дашборд відображає оновлений стан системи: у розділі «Користувачі» з'явився запис TestUser2 з кнопкою видалення, а в журналі сканувань перший рядок з часовою міткою 1777787925 тепер відображає статус allowed зеленим кольором. Попередні чотири події зі статусом unknown залишаються у журналі незмінними, що демонструє незворотність логування – жоден запис не перезаписується при зміні прав доступу.

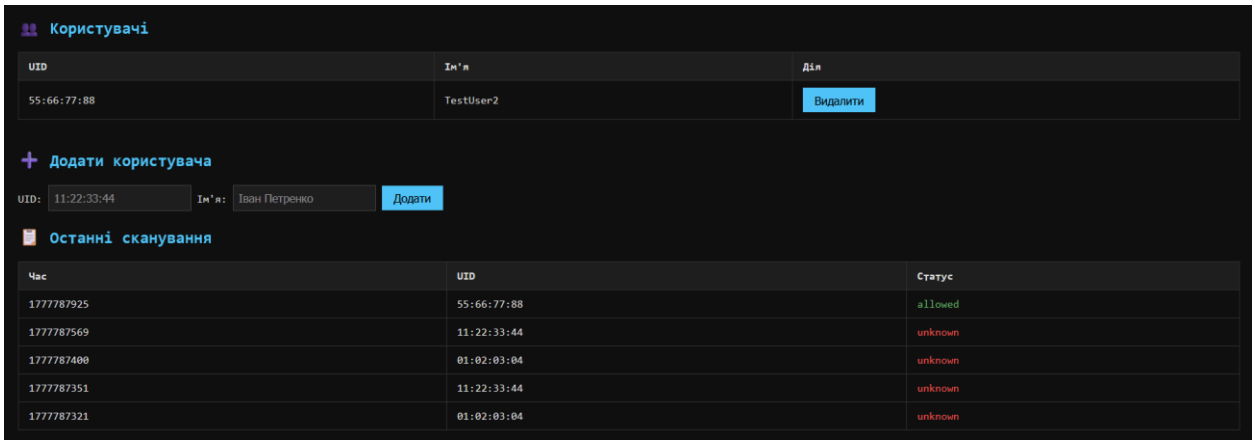


Рисунок 3.14 – Дашборд після успішної авторизації зареєстрованої картки

Коректність присвоєння статусу allowed підтверджується також на стороні симулятора. На рисунку 3.14 зображено вікно Wokwi з активним діалогом MFRC522 RFID Reader: обрана картка Yellow Card з UID 55:66:77:88.

У нижній частині Serial-монітора видно відповідь сервера зі статусом allowed, повідомлення «[ACCESS] ДОЗВОЛЕНО» та рядок журналу «[DB] 152033,55:66:77:88, allowed», що підтверджує одночасний запис події у локальний файл db.txt на симульованій SD-картці у симуляторі Wokwi. Таким чином, обидві підсистеми – серверна та апаратна – синхронно фіксують результат авторизації.

Завершальний знімок тестування (рис. 3.15) демонструє термінал ngrok після повного циклу роботи системи: зафіксовано п'ять успішних POST-запитів на /scan з кодом 200 OK у різний час сесії. Загальна кількість з'єднань (ttl)

становить 5, медіанна затримка обробки запиту (p50) – 0.23 секунди, 90-й перцентиль (p90) – 0.35 секунди. Ці показники свідчать про стабільну роботу тунелю та прийнятну затримку мережевого рівня для прикладних задач контролю доступу в умовах симуляції.

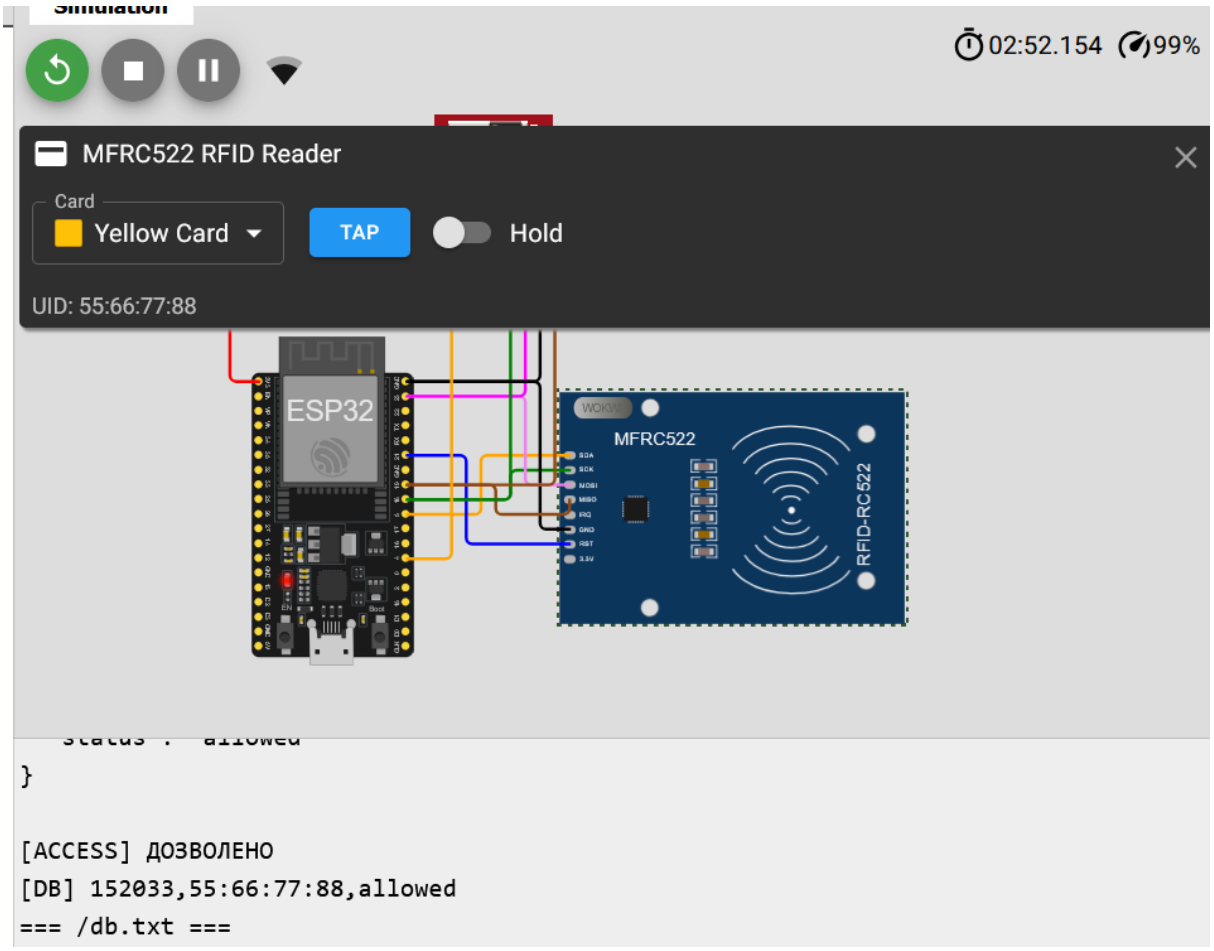


Рисунок 3.15 – Serial-монітор Wokwi з підтвердженням дозволеного доступу

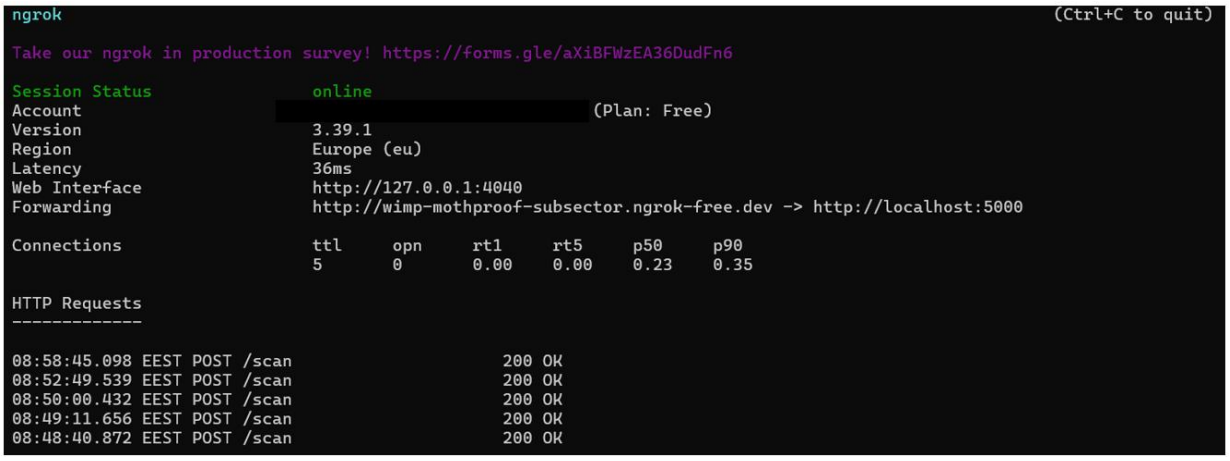


Рисунок 3.16 – Підсумковий журнал ngrok із п’ятьма успішними запитами

За результатами тестування підтверджено коректну роботу всіх функціональних компонентів системи: ініціалізація апаратної частини у Wokwi, встановлення WiFi-з'єднання та надсилання HTTP-запитів через ngrok-тунель, обробка запитів Flask-сервером з визначенням статусу доступу, збереження подій у файловій базі даних, відображення актуального стану у веб-інтерфейсі, а також додавання та верифікація авторизованих користувачів через REST API. Система продемонструвала стабільну роботу протягом усього тестового сеансу без помилок на жодному з рівнів пайплайну.

3.5 Висновки до третього розділу

У третьому розділі розроблено та перевірено повний функціональний прототип програмно-технічного засобу контролю доступу до складських приміщень в умовах програмної симуляції. Описано загальну архітектуру системи та обґрунтовано вибір інструментів симуляції, зокрема платформи Wokwi для відтворення апаратної частини на базі мікроконтролера ESP32 з модулями RFID RC522 та SD-карти. Реалізовано апаратну схему з коректним керуванням спільною шиною SPI та роздільними пінами вибору мікросхеми для кожного периферійного пристрою. Розроблено ПЗ мікроконтролера, що реалізує зчитування RFID-карток, передачу даних на сервер через WiFi та HTTP, локальне резервне логування подій на SD-картці, а також автоматичне перемикання на офлайн-режим у разі недоступності сервера. Розгорнуто серверну частину на базі Python Flask з реалізацією REST API для обробки сканувань, управління списком авторизованих користувачів та ведення журналу подій у файловій базі даних формату CSV. Забезпечено публічну доступність локального сервера через HTTP-тунель ngrok, що дозволило симульованому мікроконтролеру надсилати реальні мережеві запити до серверної частини. Проведено тестування системи з фіксацією результатів на кожному етапі, яке підтвердило коректну роботу всіх компонентів пайплайну від зчитування картки у симуляторі до відображення статусу доступу у веб-інтерфейсі.

ВИСНОВКИ

За результатами виконання кваліфікаційної роботи було спроєктовано програмно-технічний засіб контролю доступу до складських приміщень на платформі ESP32 з використанням RFID-ідентифікації, що забезпечує автоматизовану авторизацію персоналу, реєстрацію подій та дистанційне адміністрування через веб-інтерфейс.

У процесі виконання роботи було вирішено всі поставлені завдання. Зокрема, проведено аналіз предметної галузі систем контролю доступу, здійснено огляд існуючих технологій ідентифікації та програмних рішень, за результатами якого обґрунтовано вибір RFID-технології на частоті 13.56 МГц та платформи ESP32 як оптимальних для реалізації поставленої задачі. Сформовано перелік функціональних і технічних вимог до розроблюваного засобу, розроблено електричну принципову схему пристрою у середовищі EasyEDA, що охоплює п'ять функціональних блоків: мікроконтролер ESP32, RFID-модуль MFRC522, модуль SD-карти, блок індикації та імпульсний блок живлення на базі LM2596S-5.0. Проведено аналіз та обґрунтовано вибір кожного апаратного компонента з наведенням ключових технічних характеристик, складено орієнтовний кошторис апаратної частини загальною вартістю близько 1116 гривень.

Розроблено програмне забезпечення мікроконтролера ESP32 мовою C++ у середовищі Arduino IDE з реалізацією двоступеневої авторизації: при наявності мережевого з'єднання перевірка виконується на сервері, при його відсутності – автоматично перемикається на локальну базу даних SD-карти. Розроблено серверну частину на базі Python Flask з REST API, що реалізує обробку сканувань, управління списком авторизованих користувачів та ведення журналу подій у форматі CSV, а також веб-інтерфейс адміністратора для моніторингу та керування системою через браузер.

Верифікацію розробленого рішення здійснено шляхом повної симуляції функціонального ланцюжка системи у середовищі Wokwi з підключенням до локального Flask-сервера через HTTP-тунель ngrok. У симуляторі відтворено апаратну схему, що включає мікроконтролер ESP32, RFID-модуль MFRC522 та модуль SD-карти, підключені до спільної шини SPI з роздільними пінами вибору мікросхеми. Симульований мікроконтролер підключався до гостьової WiFi-мережі Wokwi-GUEST та надсилав реальні HTTP POST-запити на публічний ngrok-URL, який прозоро перенаправляв трафік до локального Flask-сервера.

Тестування охопило всі ключові функціональні сценарії системи. Перевірено коректну ініціалізацію усіх підсистем у визначеному порядку та успішне встановлення мережевого з'єднання. Підтверджено правильність обробки сканування невідомої картки – система коректно присвоювала статус unknown та фіксувала подію у журналі. Перевірено функціональність REST API шляхом додавання авторизованого користувача та верифікації запису. Після реєстрації користувача підтверджено, що повторне сканування тієї самої картки отримує статус allowed, який відображається у веб-інтерфейсі зеленим кольором та фіксується у журналі. Перевірено коректність відображення журналу подій та списку користувачів на дашборді, який був доступний як локально, так і через публічний ngrok-URL.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. 9 IoT Innovations That Make Smart Warehouses Possible. *IoT For All*. URL: https://www.iotforall.com/9-iot-innovations-that-make-smart-warehouses-possible (дата звернення: 25.04.2026).

2. IoT in Warehouses. *Davra*. URL: https://www.davra.com/iot-in-warehouses/ (дата звернення: 25.04.2026).

3. IoT in Warehouse Management: Explained. *SafetyCulture*. URL: https://safetyculture.com/topics/internet-of-things/iot-in-warehouse-management (дата звернення: 25.04.2026).

4. Warehouse Logistics: The Transformative Power of IoT in Modern Warehouses. *TagoIO*. URL: https://tago.io/blog/warehouse-logistics (дата звернення: 25.04.2026).

5. How IoT Can Be Used in Warehouses. *Quality Material Handling Inc.* URL: https://www.qmhinc.com/iot-in-warehouses/ (дата звернення: 25.04.2026).

6. How to Create IoT-enabled Smart Warehouse. *ELEKS*. URL: https://eleks.com/blog/iot-enabled-smart-warehouse/ (дата звернення: 25.04.2026).

7. Affia I., Aamer A. An Internet of Things-based smart warehouse infrastructure: design and application. *Journal of Science and Technology Policy Management*. 2022. Vol. 13, no. 1. P. 90–109. DOI: https://doi.org/10.1108/JSTPM-08-2020-0117

8. Al-Fuqaha A., Guizani M., Mohammadi M., Aledhari M., Ayyash M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys and Tutorials*. 2015. Vol. 17, no. 4. P. 2347–2376. DOI: https://doi.org/10.1109/COMST.2015.2444095

9. Atzori L., Iera A., Morabito G. The Internet of Things: A survey. *Computer Networks*. 2010. Vol. 54, no. 15. P. 2787–2805. DOI: https://doi.org/10.1016/j.comnet.2010.05.010

10. Azanha A., Vivaldini M., Pires S. R. I., Camargo Junior J. B. de. Voice picking: analysis of critical factors through a case study in Brazil and the United States.

International Journal of Productivity and Performance Management. 2016. Vol. 65, no. 5. P. 723–739. DOI: https://doi.org/10.1108/IJPPM-11-2015-0163

11. Barreto L., Amaral A., Pereira T. Industry 4.0 implications in logistics: an overview. *Procedia Manufacturing*. 2017. Vol. 13. P. 1245–1252. DOI: https://doi.org/10.1016/j.promfg.2017.09.045

12. Chibuye M., Phiri J. A Remote Sensor Network using Android Things and Cloud Computing for the Food Reserve Agency in Zambia. *International Journal of Advanced Computer Science and Applications*. 2017. Vol. 8, no. 11. DOI: https://doi.org/10.14569/ijacsa.2017.081150

13. Ding G., Wu Q., Zhang L., Lin Y., Tsiftsis T. A., Yao Y. D. An Amateur Drone Surveillance System Based on the Cognitive Internet of Things. *IEEE Communications Magazine*. 2018. Vol. 56, no. 1. P. 29–35. DOI: https://doi.org/10.1109/MCOM.2017.1700452

14. Internet of Things, IoT. IT.ua. URL: https://www.it.ua/knowledge-base/technology-innovation/internet-veschej-internet-of-things-iot (дата звернення: 09.05.2026).

15. Інтернет речей у світі (IoT). SELECTOR.SPACE. URL: https://blog.selector.space/internet-of-things/ (дата звернення: 09.05.2026).

16. Що таке Інтернет речей? *World Vision*. URL: https://worldvision.com.ua/chto-takoe-internet-veshchey/ (дата звернення: 09.05.2026).

17. Інтернет речей: чим він може бути корисний для бізнесу. *Metinvest Digital*. URL: https://metinvest.digital/ua/page/internet-veshchej-chem-on-mozhet-byt-polezen-dlya-biznesa (дата звернення: 09.05.2026).

18. Кілька найпопулярніших сфер використання інтернету речей і де можна здобути потрібну освіту. *Національний університет «Львівська політехніка»*. URL: https://lpnu.ua/news/kilka-naipopuliarnishykh-sfer-vykorystannia-internetu-rechei (дата звернення: 09.05.2026).

19. Що таке Інтернет речей або IoT? *Fiberroad*. URL: https://fiberroad.com/uk/resources/resources/what-is-the-internet-of-things-iot/ (дата звернення: 09.05.2026).

20. Інтернет речей. *Dacpol*. URL: https://www.dacpol.eu/ua/blog/post/internet-rechej.html (дата звернення: 09.05.2026).

21. Контролер Arduino UNO R3 [HM-4820]. *Хобі Манія*. URL: https://hobbymania.com.ua/tovar.php?id_tovar=4820 (дата звернення: 09.05.2026).

22. Порівняння одноплатних комп'ютерів Raspberry Pi. *VD MAIS*. URL: https://vdmais.ua/uk/articles/porivnyannya-odnoplatnyh-komp-yuteriv-raspberry-pi/ (дата звернення: 09.05.2026).

23. Gupta B. B., Quamara M. An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols. *Concurr. Comput. Pract. Exp.* 2020. Vol. 32. P. e4946.

24. Mahmmod B. M., Naser M. A., Al-Sudani A. H. S., Alsabah M., Mohammed H. J., Alaskar H., Almarshad F., Hussain A., Abdulhussain S. H. Patient monitoring system based on internet of things: A review and related challenges with open research issues. *IEEE Access*. 2024. Vol. 12. P. 132444–132479.

25. Molaei F., Rahimi E., Siavoshi H., Afrouz S. G., Tenorio V. A comprehensive review on internet of things (IoT) and its implications in the mining industry. *Am. J. Eng. Appl. Sci.* 2020. Vol. 13. P. 499–515.

26. Sikarwar S., Yadav B. Opto-electronic humidity sensor: A review. *Sens. Actuators A Phys.* 2015. Vol. 233. P. 54–70.

27. Surantha N., Atmaja P., Wicaksono M. A review of wearable internet-of-things device for healthcare. *Procedia Comput. Sci.* 2021. Vol. 179. P. 936–943.

28. Seneviratne S., Hu Y., Nguyen T., Lan G., Khalifa S., Thilakarathna K., Hassan M., Seneviratne A. A survey of wearable devices and challenges. *IEEE Commun. Surv. Tutor.* 2017. Vol. 19. P. 2573–2620.

29. Vijayan V., Connolly J. P., Condell J., McKelvey N., Gardiner P. Review of wearable devices and data collection considerations for connected health. *Sensors*. 2021. Vol. 21. P. 5589.

30. Stavropoulos T. G., Papastergiou A., Mpaltadoros L., Nikolopoulos S., Kompatsiaris I. IoT wearable sensors and devices in elderly care: A literature review. *Sensors*. 2020. Vol. 20. P. 2826.

31. Pantelopoulos A., Bourbakis N. G. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* 2009. Vol. 40. P. 1–12.

32. Kumar S., Tiwari P., Zymbler M. Internet of Things is a revolutionary approach for future technology enhancement: A review. *J. Big Data*. 2019. Vol. 6. P. 111.

33. Al-Fuqaha M., Guizani M., Mohammadi A., Atya M., Hossain M. M., Mumtaz J. Internet of things (IoT): A review. *IEEE Commun. Surv. Tutor.* 2017. Vol. 20. P. 1647–1685.

34. Patel K. K., Patel S. M., Scholar P. Internet of things-IOT: Definition, characteristics, architecture, enabling technologies, application & future challenges. *Int. J. Eng. Sci. Comput.* 2016. Vol. 6. P. 6122–6131.

35. Zanella A., Bui N., Castellani A., Vangelista L., Zorzi M. Internet of things for smart cities. *IEEE Internet Things J.* 2014. Vol. 1. P. 22–32.

36. Minerva R., Biru A., Rotondi D. Towards a definition of the Internet of Things (IoT). *IEEE Internet Initiat.* 2015. Vol. 1. P. 1–86.

37. Kadhim K. T., Alsahlany A. M., Wadi S. M., Kadhum H. T. An overview of patient's health status monitoring system based on internet of things (IoT). *Wirel. Pers. Commun.* 2020. Vol. 114. P. 2235–2262.

38. Silva B. N., Khan M., Han K. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustain. Cities Soc.* 2018. Vol. 38. P. 697–713.

39. Ali S. M., Noghanian S., Khan Z. U., Alzahrani S., Alharbi S., Alhartomi M., Alsulami R. Wearable and Flexible Sensor Devices: Recent Advances in Designs, Fabrication Methods, and Applications. *Sensors*. 2025. Vol. 25. P. 1377.

40. Singh G. Wearable IoT (w-IoT) artificial intelligence (AI) solution for sustainable smart-healthcare. *Int. J. Inf. Manag. Data Insights*. 2025. Vol. 5. P. 100291.

41. Nikpour M., Yousefi P. B., Jafarzadeh H., Danesh K., Shomali R., Asadi S., Lonbar A. G., Ahmadi M. Intelligent energy management with iot framework in smart cities using intelligent analysis: An application of machine learning methods for complex networks and systems. *J. Netw. Comput. Appl.* 2025. Vol. 235. P. 104089.

42. El-Afifi M. I., Sedhom B. E., Padmanaban S., Eladl A. A. A review of IoT-enabled smart energy hub systems: Rising, applications, challenges, and future prospects. *Renew. Energy Focus*. 2024. Vol. 51. P. 100634.

43. Muthuramalingam S., Bharathi A., Rakesh Kumar S., Gayathri N., Sathiyaraj R., Balamurugan B. IoT based intelligent transportation system (IoT-ITS) for global perspective: A case study. *Internet Things Big Data Anal. Smart Gener.* 2019. Vol. 154. P. 279–300.

44. Tiwari V., Mishra A., Tiwari S. Role of data safety and perceived privacy for acceptance of IoT-enabled technologies at smart tourism destinations. *Curr. Issues Tour.* 2024. Vol. 27. P. 3079–3094.

45. Sheng M., He Y., Chen K. Edge computing for internet of things: A survey. *IEEE Access*. 2020. Vol. 8. P. 178019–178041.

46. Kim D. H., Lee J. H., Choi S. J. Energy harvesting for energy-efficient internet of things (IoT). *IEEE Access*. 2016. Vol. 4. P. 4981–4991.

47. Rodríguez-Martínez E. A., Flores-Fuentes W., Achakir F., Sergiyenko O., Murrieta-Rico F. N. Vision-based navigation and perception for autonomous robots: Sensors, SLAM, control strategies, and cross-domain applications—A review. *Eng*. 2025. Vol. 6. P. 153.

48. Shi W., Dustdar S. The promise of edge computing. *Computer*. 2016. Vol. 49. P. 78–81.

49. Mukhopadhyay S. C., Tyagi S. K. S., Suryadevara N. K., Piuri V., Scotti F., Zeadally S. Artificial intelligence-based sensors for next generation IoT applications: A review. *IEEE Sens. J.* 2021. Vol. 21. P. 24920–24932.

50. Badidi E., Moumane K., el Ghazi F. Opportunities, Applications, and Challenges of Edge-AI Enabled Video Analytics in Smart Cities: A Systematic Review. *IEEE Access*. 2023. Vol. 11. P. 80543–80572.

51. Mshali H., Lemlouma T., Moloney M., Magoni D. A survey on health monitoring systems for health smart homes. *Int. J. Ind. Ergon.* 2018. Vol. 66. P. 26–56.

52. Benfradj A., Thaljaoui A., Moulahi T., Khan R. U., Alabdulatif A., Lorenz P. Integration of artificial intelligence (AI) with sensor networks: Trends, challenges, and future directions. *J. King Saud Univ.-Comput. Inf. Sci.* 2024. Vol. 36. P. 101892.

53. Shajari S., Kuruvinashetti K., Komeili A., Sundararaj U. The emergence of ai-based wearable sensors for digital health technology: A review. *Sensors*. 2023. Vol. 23. P. 9498.

54. Subhan F., Mirza A., Su'ud M. B. M., Alam M. M., Nisar S., Habib U., Iqbal M. Z. Ai-enabled wearable medical internet of things in healthcare system: A survey. *Appl. Sci.* 2023. Vol. 13. P. 1394.

55. White R. M. A sensor classification scheme. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*. 1987. Vol. 34. P. 124–126.

56. Janudin N., Kasim N. A. M., Knight V. F., Halim N. A., Noor S. A. M., Ong K. K., Yunus W. M. Z. W., Norrrahim M. N. F., Misenan M. S. M., Razak M. A. I. A. et al. Sensing techniques on determination of chlorine gas and free chlorine in water. *J. Sens.* 2022. Vol. 2022. P. 1898417.

57. Ali S. M., Noghanian S., Khan Z. U., Alzahrani S., Alharbi S., Alhartomi M., Alsulami R. Wearable and Flexible Sensor Devices: Recent Advances in Designs, Fabrication Methods, and Applications. *Sensors*. 2025. Vol. 25. P. 1377.

58. Nikpour M., Yousefi P. B., Jafarzadeh H., Danesh K., Shomali R., Asadi S., Lonbar A. G., Ahmadi M. Intelligent energy management with iot framework in smart

cities using intelligent analysis: An application of machine learning methods for complex networks and systems. *J. Netw. Comput. Appl.* 2025. Vol. 235. P. 104089.

59. El-Afifi M. I., Sedhom B. E., Padmanaban S., Eladl A. A. A review of IoT-enabled smart energy hub systems: Rising, applications, challenges, and future prospects. *Renew. Energy Focus*. 2024. Vol. 51. P. 100634.

60. Singh G. Wearable IoT (w-IoT) artificial intelligence (AI) solution for sustainable smart-healthcare. Int. J. Inf. Manag. Data Insights. 2025. Vol. 5. P. 100291.

61. EasyEDA. *EasyEDA*. URL: https://passport.easyeda.com (дата звернення: 09.05.2026).

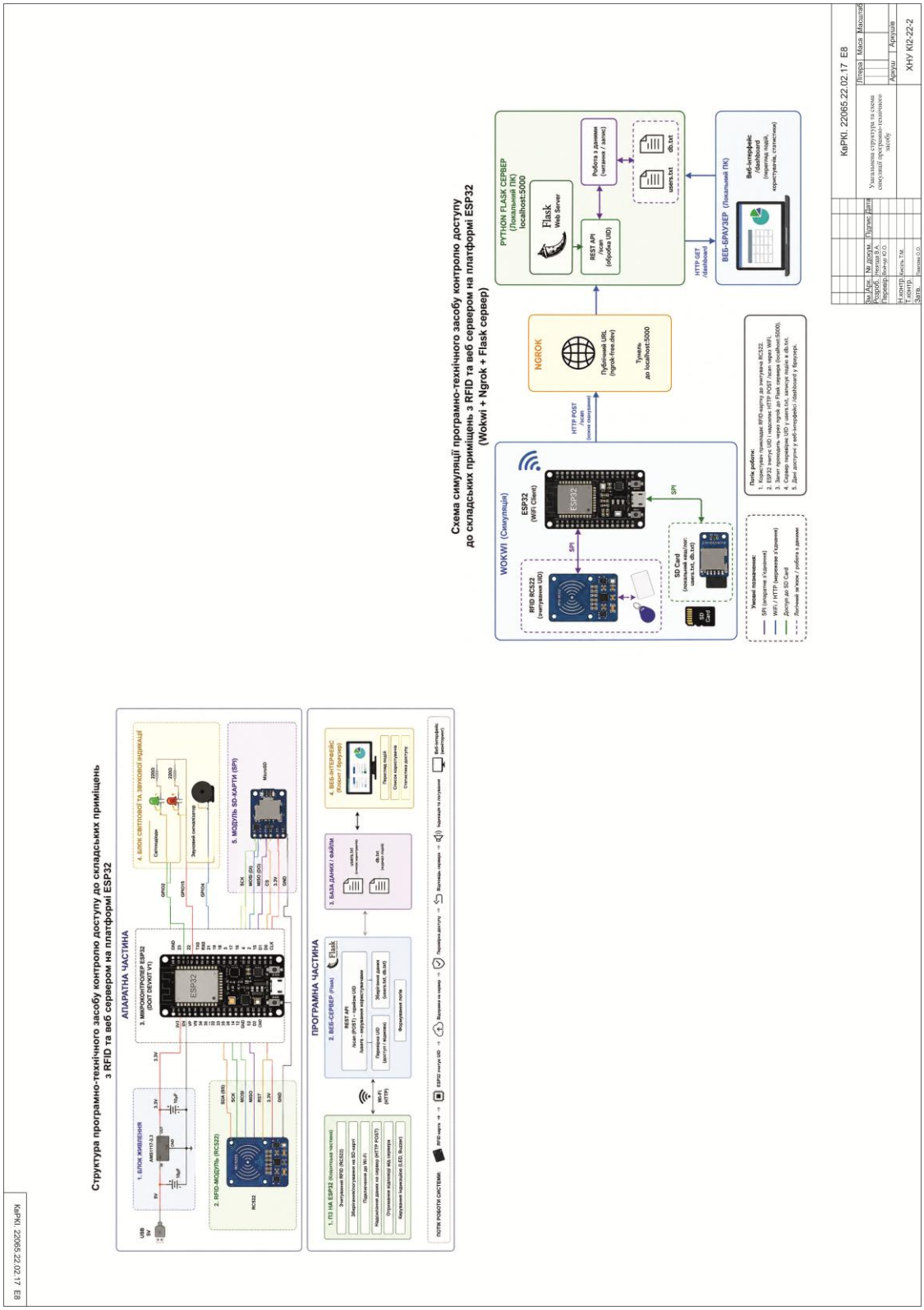
62. Wokwi - World's most advanced ESP32 Simulator. *Wokwi*. URL: https://wokwi.com (дата звернення: 09.05.2026).

63. REST API: що це простими словами + приклади коду 2025. *DAN IT Education*. URL: https://dan-it.com.ua/uk/blog/shho-take-rest-api-i-yak-vin-praczyuye/ (дата звернення: 09.05.2026).

ДОДАТОК А

(обов’язковий)

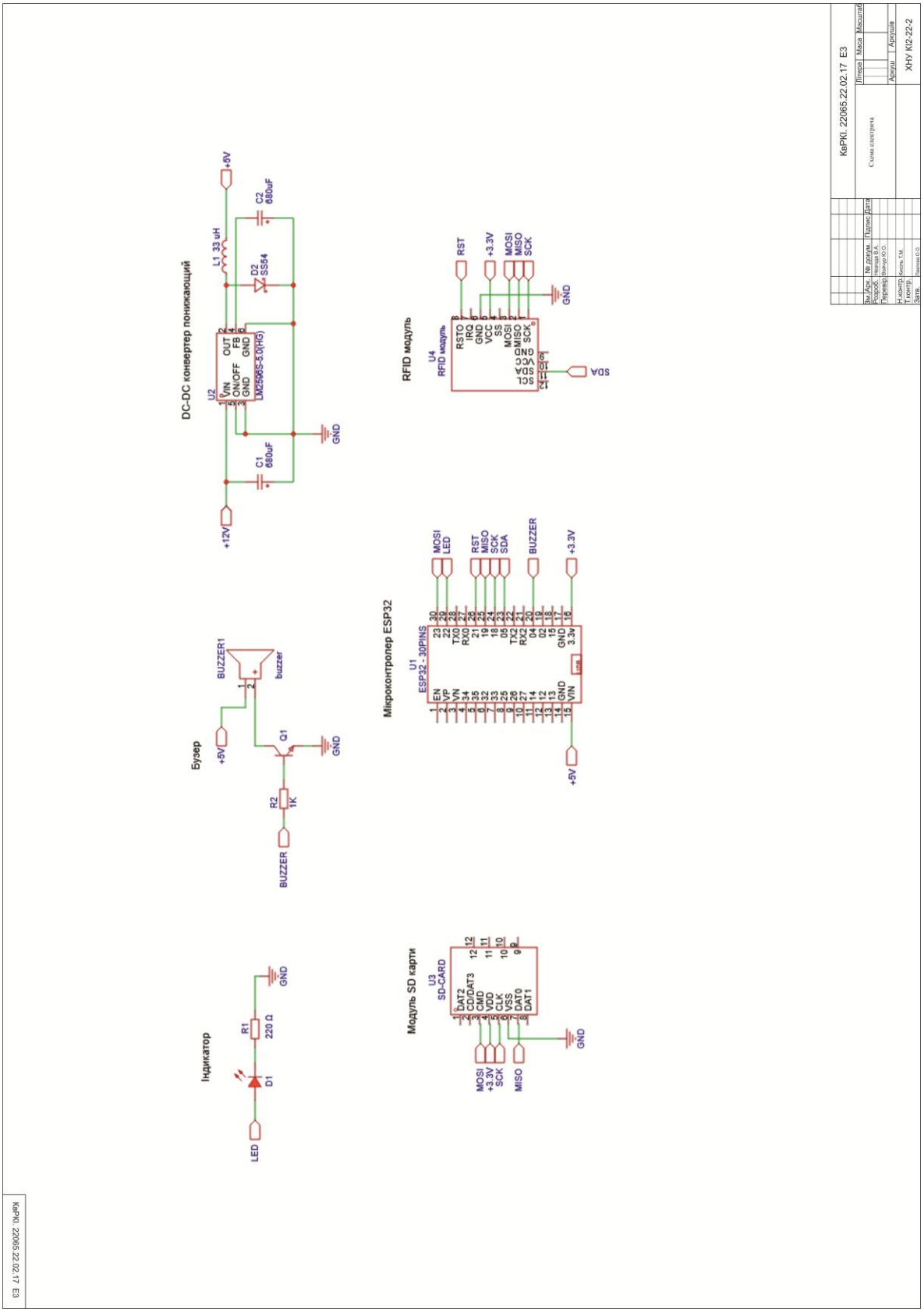
Копія креслення «Узагальнена структура та схема симуляції програмно-технічного засобу»



ДОДАТОК Б

(обов’язковий)

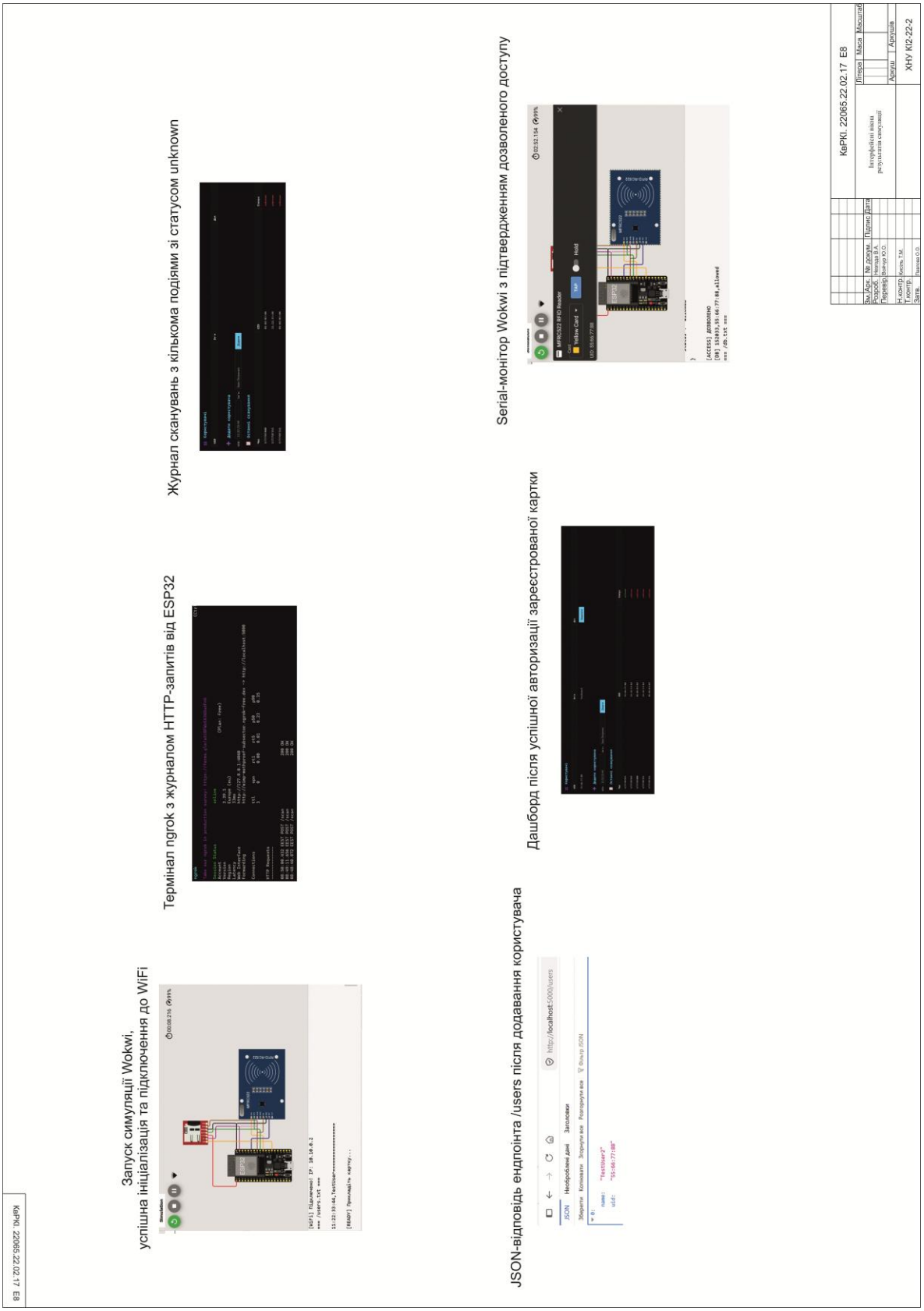
Копія креслення «Схема електрична»



ДОДАТОК В

(обов’язковий)

Копія креслення «Інтерфейсні вікна результатів симуляції»



Зав. кафедри КІІС
д-р. філософії Ользі ПАВЛОВІЙ

Владислав НЕЗГОДА
ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-2

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року

Tue May 26 08:58:47 EEST 2026, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (http://ap.km.ua) v-15.701

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 15%**

	Документ		Сумарний збіг по Базі Даних	
ID: 272269 Назва: БКР Програмно-технічний засіб контролю доступу до складських приміщень з RFID та веб сервером на платформі ESP32 Додано в БД: 2026-05-26 Автора: Владислав НЕЗГОДА Керівники: Юрій ВОЙЧУР Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	83187	595	1870 (2%)	25 (4%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Владислав НЕЗГОДА

Співавтор:

Назва: Програмно-технічний засіб контролю доступу до складських приміщень з RFID та веб сервером на платформі ESP32

Експерт: Юрій ВОЙЧУР

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:8.08%

Коефіцієнт подібності 2:3.15%

Мікропробіли: 9

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-05-25 23:52:20.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2026-05-26 Доцент Андрій Нічепорук

Дата експерт

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-технічний засіб контролю доступу до складських приміщень з RFID та веб сервером на платформі ESP32
Автор Владислав НЕЗГОДА
Освітня програма Комп'ютерна інженерія та програмування
Рівень вищої освіти перший (бакалаврський)
Спеціальність 123 Комп'ютерна інженерія
Науковий керівник: ДФ, Юрій ВОЙЧУР

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;

2) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 8,08% і адресується до 33 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

27.05.2026

Завідувач кафедри _____ Ольга ПАВЛОВА
Підпис Ім'я, ПРІЗВИЩЕ

Гарант освітньої програми _____ Андрій НІЧЕПОРУК
Підпис Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи _____ Юрій ВОЙЧУР
Підпис Ім'я, ПРІЗВИЩЕ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Владислав НЕЗГОДА

Тема: Програмно-технічний засіб контролю доступу до складських приміщень з RFID та веб сервером на платформі ESP32

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 60

1. Короткий зміст роботи та прийнятих рішень: Метою дипломної роботи є проєктування програмно-технічного засобу контролю доступу до складських приміщень на платформі ESP32 з використанням RFID-ідентифікації, що забезпечує автоматизовану авторизацію персоналу, реєстрацію подій та дистанційне адміністрування через веб-інтерфейс.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи проведено аналіз предметної області систем контролю доступу та огляд відомих рішень, а саме: проаналізовано предметну галузь систем контролю доступу; досліджено технології ідентифікації в системах контролю доступу; виконано аналіз RFID-технології; розглянуто платформи для реалізації вбудованих систем; проаналізовано існуючі програмні рішення для контролю доступу.

У другому розділі кваліфікаційної роботи виконано проєктування програмно-технічного засобу контролю доступу до складських приміщень на основі ESP32 та RFID, а саме: сформовано вимоги до системи; розроблено узагальнену структуру програмно-технічного засобу; створено електричну принципову схему; виконано аналіз та вибір апаратних компонентів; розраховано орієнтовну вартість реалізації; спроєктовано програмну архітектуру системи; сформовано висновки до розділу.

У третьому розділі кваліфікаційної роботи виконано симуляцію та тестування програмно-технічного засобу контролю доступу, а саме: розроблено загальну схему реалізації системи; виконано симуляцію апаратної частини у середовищі Wokwi; реалізовано серверну частину системи; проведено тестування програмно-технічного засобу та аналіз отриманих результатів.

5. Негативні сторони роботи: До недоліків роботи можна віднести обмежену перевірку системи в умовах реального експлуатаційного середовища та відсутність оцінювання стійкості до нештатних ситуацій і спроб несанкціонованого доступу.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на достатньому технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре С (75)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Пасічник Олександр Анатолійович,

доцент кафедри КН, ХНУ

"___"_____2026 р. _____ (підпис)