

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та безпеки

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «МАГІСТР»**

**ЕЛЕКТРОННА СИСТЕМА ІДЕНТИФІКАЦІЇ ВІДВІДУВАЧІВ НА ОСНОВІ
ТЕХНОЛОГІЇ RFID**

ELECTRONIC VISITOR IDENTIFICATION SYSTEM BASED ON RFID TECHNOLOGY

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІм-21

Пришневський Олег Олегович

(підпис)

Керівник:

к.т.н., доцент

Лавренчук Світлана Василівна

(підпис)

Кваліфікаційну роботу

допущено до захисту

«___» грудня 2025 р.

Гарант освітньої програми:

к.т.н., доцент

Гринюк Сергій Васильович

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: магістр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ доц. Т. ТЕРЛЕЦЬКИЙ

«_____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Пришневському Олегу Олеговичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Електронна система ідентифікації відвідувачів на основі технології RFID*

Керівник роботи: *к.т.н., доцент Лавренчук Світлана Василівна*

затверджені наказом закладу вищої освіти від «17» червня 2025 р. №290/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 09.12.2025 р.

3. Вихідні дані до роботи *Джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області, різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, що потрібно розробити):

Вступ

Аналіз проблеми за темою роботи та постановка завдань дослідження електронної системи ідентифікації відвідувачів на основі технології RFID

Теоретична частина

Практична частина

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Рисунки інтерфейсів користувача, схеми моделей бази даних, архітектурні діаграми, листинги коду

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблеми за темою роботи та постановка завдань дослідження</i>	<i>Лавренчук С.В., доцент</i>		
<i>Теоретичне дослідження та практична реалізація</i>	<i>Лавренчук С.В., доцент</i>		
<i>Практична реалізація об'єкта проектування</i>	<i>Лавренчук С.В., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Гринюк С.В., доцент</i>		
<i>Показник запозичень тексту</i>	___%		
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст.викладач</i>		

7. Дата видачі завдання 18.06.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної проблеми</i>	01.07.2025	Виконано
2.	<i>Аналіз проблеми за темою роботи та постановка завдань дослідження електронної системи ідентифікації відвідувачів на основі технології RFID</i>	14.08.2025	Виконано
3.	<i>Теоретичне дослідження та практична реалізація</i>	20.09.2025	Виконано
4.	<i>Практична реалізація об'єкта проектування</i>	10.10.2025	Виконано
5.	<i>Висновки та пропозиції</i>	24.10.2025	Виконано
6.	<i>Формування списку використаних джерел</i>	25.10.2025	Виконано
7.	<i>Формування додатків</i>	27.10.2025	Виконано
8.	<i>Оформлення ілюстративного матеріалу</i>	02.11.2025	Виконано
9.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівникові</i>	07.11.2025	Виконано
10.	<i>Нормоконтроль</i>	26.11.2025	Виконано
11.	<i>Інструментальна перевірка на академічний плагіат</i>	02.12.2025	Виконано
12.	<i>Здача кваліфікаційної роботи та всіх супровідних документів на кафедрі</i>	09.12.2025	Виконано

Здобувач вищої освіти

_____ (підпис)

Пришневський О.О.

_____ (прізвище, ініціали)

Керівник кваліфікаційної роботи

_____ (підпис)

Лавренчук С.В.

_____ (прізвище, ініціали)

АНОТАЦІЯ

Пришневський О. О. Електронна система ідентифікації відвідувачів на основі технології RFID. Рукопис.

Кваліфікаційна робота магістра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел, додатків.

У першому розділі розглянуто теоретичні основи електронних систем ідентифікації, зокрема технології RFID. Також було проаналізовано принципи роботи радіочастотних міток, їх переваги порівняно з іншими методами автоматичної ідентифікації (штрих-кодами, біометрією) та обґрунтовано вибір апаратних компонентів системи – мікроконтролера ESP8266 та RFID-зчитувача MFRC522. Особливу увагу приділено аналізу програмних засобів для розробки IoT-рішень, включаючи використання Flask для серверної частини та matplotlib для візуалізації даних.

У другому розділі представлено архітектуру модульної клієнт-серверної системи, що включає апаратний рівень на базі ESP8266, серверну частину з REST API на Flask та клієнтський інтерфейс з Telegram-ботом та детально описано структуру бази даних PostgreSQL для зберігання статусів користувачів та історії подій.

У третьому розділі висвітлено можливості використання системи для аналізу показників KPI працівників, а також описано процес розробки програмного забезпечення системи. Особливий акцент зроблено на алгоритмах обробки RFID-міток, механізмах виявлення запізнь та інтеграції з Telegram API для сповіщень.

Об'єкт дослідження – система автоматичної ідентифікації відвідувачів на базі RFID-технології з функцією аналізу відвідуваності.

Предмет дослідження – методи та технології інтеграції апаратних компонентів (ESP8266, MFRC522) з хмарними сервісами для створення масштабованої системи контролю доступу.

Метою роботи є розробка електронної системи ідентифікації відвідувачів з функціями реального часу, автоматизованим аналізом відвідуваності та інтеграцією з месенджерами для оперативних сповіщень.

Ключові слова: RFID, ESP8266, MFRC522, PostgreSQL, Flask, matplotlib, IoT, система контролю доступу, автоматична ідентифікація, веб-інтерфейс, Telegram API.

ANNOTATION

Pryshnevsky O. Electronic Visitor Identification System Based on RFID Technology. Manuscript.

Master's qualification work in the Educational Program «Computer Engineering» specializing in 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

The qualification work consists of an introduction, three chapters, conclusions, a list of references, and appendices.

The first chapter examines the theoretical foundations of electronic identification systems, particularly RFID technology. It analyzes the operational principles of radio-frequency tags, their advantages over other automated identification methods (barcodes, biometrics), and substantiates the selection of hardware components – the ESP8266 microcontroller and MFRC522 RFID reader. Special attention is given to analyzing software tools for IoT solutions development, including the use of Flask for server-side implementation and matplotlib for data visualization.

The second chapter presents the architecture of a modular client-server system comprising an ESP8266-based hardware layer, a Flask-powered REST API server component, and a client interface with Telegram bot integration. The PostgreSQL database structure for storing user statuses and event history is described in detail.

The third chapter explores the system's capabilities for analyzing employee KPIs (Key Performance Indicators) and outlines the software development process. Particular emphasis is placed on RFID tag processing algorithms, late arrival detection mechanisms, and Telegram API integration for notifications.

Research Object – Automated visitor identification system based on RFID technology with attendance analytics functionality.

Research Subject – Methods and technologies for integrating hardware components (ESP8266, MFRC522) with cloud services to create a scalable access control system.

The purpose of the work is to develop an electronic visitor identification system with real-time functionality, automated attendance analytics, and messenger integration for operational notifications.

Keywords: RFID, ESP8266, MFRC522, PostgreSQL, Flask, matplotlib, IoT, access control system, automated identification, web interface, Telegram API.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ЕЛЕКТРОННІ СИСТЕМИ	11
1.1 Основні поняття та сфери застосування електронних систем	11
1.2 Процес створення електронних систем і їх класифікація.....	13
1.3 Огляд технологій і засобів для розробки систем ідентифікації.....	15
1.4 Огляд програмного забезпечення для роботи з електронними системами.....	19
РОЗДІЛ 2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ ІДЕНТИФІКАЦІЇ ВІДВІДУВАЧІВ	25
2.1 Обґрунтування вибору технології RFID.....	25
2.2 Архітектура системи ідентифікації.....	26
2.3 Моделювання та опис електричної схеми.....	30
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ ВІДВІДУВАЧІВ	36
3.1 Дослідження ефективності праці за допомогою спроектованої ІОТ-системи.....	36
3.2 Розробка програмного забезпечення для зчитування RFID.....	41
3.3 Розробка серверної частини.....	44
3.4 Розробка інтерфейсу користувача.....	46
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТКИ.....	53

ВСТУП

Актуальність теми. У сучасних умовах підвищених вимог до безпеки та ефективного управління доступом, автоматизовані системи ідентифікації стають критично важливими для організацій різного профілю. Особливу важливість RFID-технологія набуває в контексті необхідності швидкої та безпомилкової ідентифікації великих потоків відвідувачів, що актуально для офісних центрів, медичних закладів та об'єктів критичної інфраструктури. Впровадження таких систем дозволяє не лише підвищити рівень безпеки, але й автоматизувати збір аналітичних даних про відвідуваність.

Метою роботи є створення електронної системи ідентифікації відвідувачів з використанням RFID-технології, що забезпечує реальний час обробки подій, інтеграцію з месенджерами для сповіщень та автоматизований аналіз паттернів відвідуваності.

Об'єкт дослідження – система автоматичної ідентифікації на базі RFID-технології з функцією контролю доступу та аналітикою відвідувань.

Предмет дослідження – методи інтеграції апаратних компонентів (ESP8266, MFRC522) з хмарними сервісами для створення масштабованого рішення управління доступом.

Завдання:

- реалізувати апаратний модуль ідентифікації на базі ESP8266 з RFID-зчитувачем MFRC522;
- розробити серверну частину з REST API на Flask для обробки та зберігання даних у PostgreSQL та інтеграції з Telegram API для оперативних сповіщень;
- створити аналітичний модуль з візуалізацією даних за допомогою matplotlib.

Наукова новизна роботи полягає у розробці та реалізації інтегрованої електронної системи ідентифікації відвідувачів на основі RFID-технології, що поєднує апаратно-програмну архітектуру з розширеними аналітичними

функціями для оцінки ефективності праці. Вперше в контексті RFID-ідентифікації впроваджено інтеграцію з Telegram-ботом для оперативного інформування та аналітичним модулем на базі Python та matplotlib, що дозволяє будувати KPI-панелі у реальному часі та здійснювати комплексний аналіз робочих показників. Додатковою новизною є адаптація методик визначення KPI до даних RFID-моніторингу, що включає індекс ефективності робочого часу та частоту відхилень від графіка, розширюючи можливості традиційних систем контролю доступу. Поєднання технологій IoT, бездротових протоколів та аналітичних методів у запропонованому рішенні дозволяє створити гнучку та масштабовану платформу, здатну адаптуватися до різних сфер застосування, від управління доступом до оцінки продуктивності персоналу, що не має прямих аналогів у наявних комерційних системах.

Практичне значення полягає в можливості використання системи для автоматизації контролю доступу в офісних приміщеннях, на виробничих майданчиках та публічних заходах. Розроблена система відкриває можливості для детального аналізу статистики відвідуваності та розрахунку ключових показників KPI.

Особистий внесок здобувача включає проектування електричних схем, розробку алгоритмів обробки RFID-міток, створення трирівневої архітектури системи, впровадження механізму автоматичних сповіщень через Telegram API, реалізацію та оптимізацію мережевої взаємодії.

Апробація результатів. Результати роботи представлені на V Міжнародній науковій конференції «Інноваційна наука: пошук відповідей на виклики сучасності», яка відбулася 21 листопада 2025 р. (додаток А).

Публікації. Результати досліджень, виконаних під час роботи над кваліфікаційною роботою магістра опубліковано в науковій статті [1].

Структура й обсяг роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел з 23 позицій та 4 додатків. Загальний обсяг роботи – 71 сторінка, що містить 38 рисунків та 5 формул.

РОЗДІЛ 1

ЗАГАЛЬНІ ВІДОМОСТІ ПРО ЕЛЕКТРОННІ СИСТЕМИ

1.1 Основні поняття та сфери застосування електронних систем

Електронні системи представляють собою комплекс апаратних та програмних засобів, об'єднаних єдиними протоколами взаємодії для вирішення конкретних завдань. Їх фундаментальною основою є принцип перетворення інформації через електронні компоненти та мікропроцесорну обробку сигналів [1]. Ключові концепції включають сенсори для фіксації фізичних параметрів, актуатори для керування виконавчими механізмами, мікроконтролери як «мозкові центри» системи, та інтерфейси для комунікації з користувачем.

Теоретичні основи електроніки системного рівня ґрунтуються на чотирьох фундаментальних принципах: обробці сигналів для покращення якості інформації, проектуванні спеціалізованих електричних ланцюгів, розробці вбудованих систем із чітко визначеною функціональністю та створенні керуючих контурів зворотного зв'язку. Ці принципи реалізуються через синергію апаратних компонентів – від сенсорів, здатних перетворювати фізичні величини в електричні сигнали, до комунікаційних інтерфейсів, що забезпечують обмін даними між пристроями за стандартизованими протоколами (рис. 1.1).

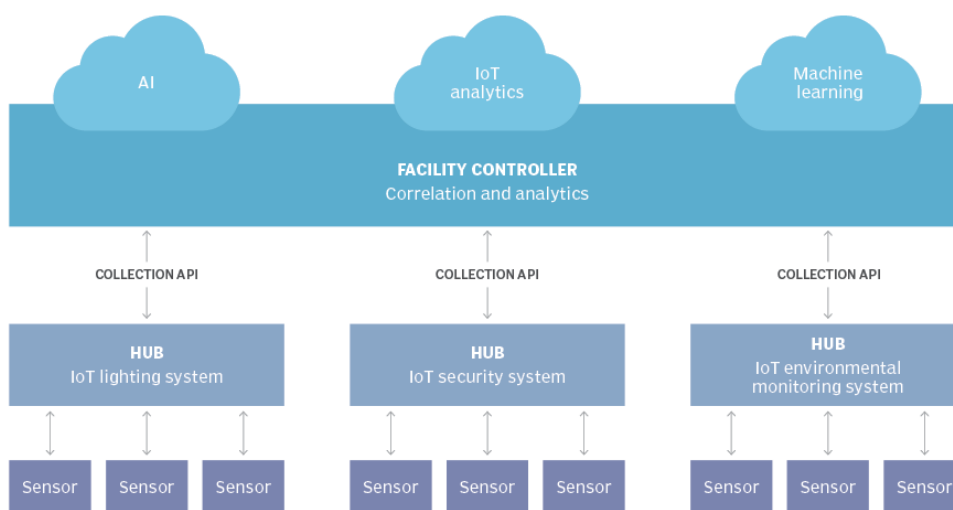


Рисунок 1.1 – Приклади архітектури електронних систем [2]

Сучасні електронні системи знаходять застосування в автоматизації промислових процесів, системах «розумного дому», транспортній логістиці та медичній діагностиці. Технології розумного дому трансформують житлові приміщення в інтелектуальні мережі, де взаємодія між пристроями здійснюється через інтернет-протоколи. Основу таких систем становить концепція централізованого управління побутовими приладами – від освітлення до систем безпеки, через єдиний програмний інтерфейс. Сучасні електронні системи знаходять застосування в автоматизації промислових процесів через інтелектуальні SCADA-системи з адаптивним керуванням технологічними параметрами. У транспортній інфраструктурі вони реалізують функції телематики для моніторингу логістичних маршрутів та предиктивного аналізу стану транспортних засобів [3]. Медичні електронні комплекси нового покоління інтегрують біосенсори з AI-аналітикою для безперервного моніторингу вітально важливих показників пацієнтів у режимі реального часу.

Розвиток енергетичних систем супроводжується впровадженням розподілених мереж керування з автоматизованим балансуванням навантаження та інтелектуальним обліком енергоспоживання. У агропромисловому секторі електронні системи забезпечують прецизійне землеробство через мультиспектральні сенсори ґрунту та дронів системи моніторингу врожайності. Аерокосмічна галузь використовує бортові електронні комплекси для навігації БПЛА, аналізу метеоданих та керування системами життєзабезпечення космічних апаратів.

Важливим аспектом є конвергенція електронних систем з технологіями Інтернету речей (IoT), що дозволяє створювати розподілені мережі датчиків з централізованим управлінням. В контексті «розумного дому» це проявляється у синхронізації роботи термостатів, відеоспостереження та енергоспоживачів через спеціалізовані хаби типу Amazon Echo або Google Home (рис. 1.2). Такі контролери аналізують дані з підключених пристроїв, адаптуючи їх роботу до звичок мешканців – наприклад, автоматично регулюючи температуру в приміщенні на основі історичних даних.



Рисунок 1.2 – Google Home та Amazon Echo [4]

Енергоефективність інтелектуальних систем досягається за рахунок алгоритмів предиктивного аналізу, які оптимізують роботу побутових приладів. Смарт-ліхтарі з датчиками руху адаптують освітленість до присутності людей у кімнаті, а мультисенсорні системи моніторингу можуть попередити про витoki води чи перегрів електропроводки. Особливу роль відіграють голосові інтерфейси керування, що інтегрують окремі пристрої в єдиний екологічний комплекс з можливістю дистанційного контролю через мобільні додатки.

1.2 Процес створення електронних систем і їх класифікація

Процес розробки електронних систем базується на ітераційній моделі, що починається з формулювання технічного завдання та аналізу предметної області. Перша фаза передбачає визначення функціональних вимог, обмежень середовища експлуатації та критеріїв енергоефективності. На етапі проектування

здійснюється вибір мікропроцесорної платформи, розробка структурної схеми та моделювання ключових вузлів системи за допомогою спеціалізованого ПЗ.

Важливим аспектом проектування є представлення системи у вигляді блок-схеми (block-diagram representation), де кожен функціональний модуль відображається як окремий блок із чіткими вхідними та вихідними сигналами. Така модель ґрунтується на принципі причинно-наслідкового зв'язку, де вхідні сигнали (cause) трансформуються системою у вихідні дії (effect) (рис. 1.3).

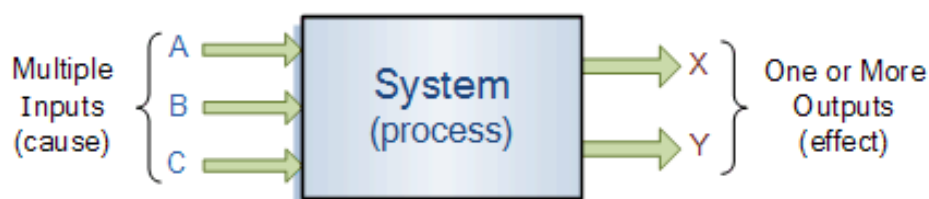


Рисунок 1.3 – Блок-схема простої електронної системи [5]

Наприклад, у системі контролю температури сенсор (вхідний пристрій) перетворює тепло у електричний сигнал, який обробляється мікроконтролером (процес), а вихідний актуатор активує нагрівальний елемент або систему вентиляції.

Класифікація електронних систем базується на кількох основних критеріях. По-перше, за рівнем автономності розрізняють вбудовані та периферійні системи: вбудовані працюють самостійно і виконують конкретні завдання всередині більшого пристрою, тоді як периферійні залежать від зовнішнього керування. По-друге, за характером обробки інформації електронні системи поділяють на аналогові, цифрові та гібридні – перші працюють з безперервними сигналами, другі з дискретними, а гібридні поєднують обидва підходи. Ще один критерій – тип часової області, де виділяють безперервні системи, що опрацьовують сигнали в реальному часі без перерв, і дискретні, які працюють з сигналами у певні моменти часу. Нарешті, електронні системи класифікують за сферою застосування – промислові, побутові або спеціалізовані, залежно від того, де і для яких цілей вони використовуються. Таким чином, ці

критерії допомагають краще розуміти різноманітність і функціональність електронних систем у сучасному світі.

Безперервні (Continuous Time) системи оперують аналоговими сигналами, неперервними у часі, як-от напруга чи температура. Дискретні (Discrete Time) системи працюють з послідовністю відліків, отриманих у фіксовані моменти часу, що характерно для цифрових мікропроцесорних систем. Гібридні системи поєднують обидва підходи, де аналогові входи перетворюються АЦП у цифрові сигнали для подальшої обробки. На рисунку 1.4 зображені блок-схеми безперервної та дискретної систем.



Рисунок 1.4 – Блок-схеми безперервної та дискретної систем [5]

Технологічний цикл створення включає прототипування на макетних платах, де особливу увагу приділяють взаємодії між вхідними та вихідними компонентами. Наприклад, при розробці аудіосистеми мікрофон (вхід) генерує електричні сигнали, які підсилюються (процес) та трансформуються динаміком у звукові хвилі (вихід). Сучасні системи часто є комбінацією підсистем, як-от медіацентр з підключенням DVD-плеєра, ТВ-тюнера та потокових джерел до єдиного процесора обробки звуку.

1.3 Огляд технологій і засобів для розробки систем ідентифікації

Сучасні системи ідентифікації базуються на спектрі технологічних рішень, що диференціюються за принципом дії та сферою застосування. Основу становлять AIDC-технології (Automatic Identification and Data Capture), які забезпечують автоматичне зчитування даних без участі людини. Ці системи

складаються з таких ключових компонентів: пристроїв захоплення даних, систем обробки інформації та комунікаційних мереж (рис. 1.5).

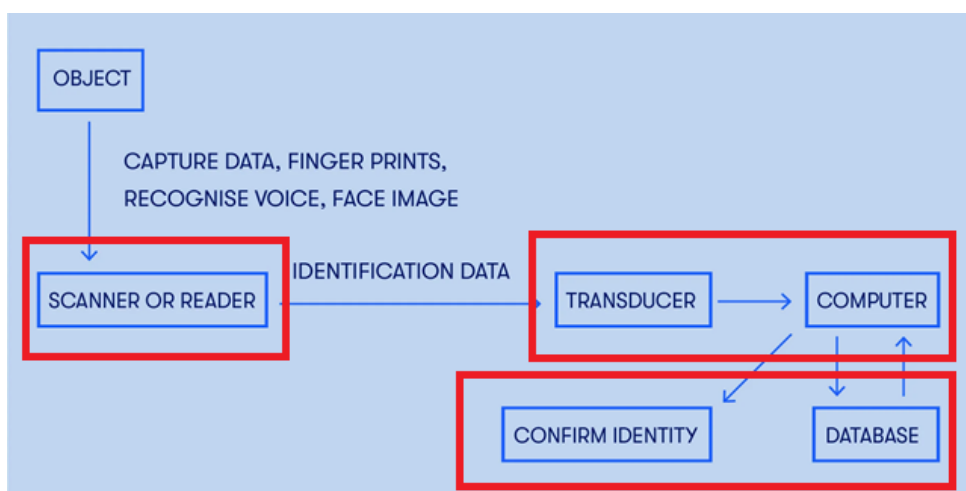


Рисунок 1.5 – Схема принципу роботи AIDC технологій [6]

До основних технологій автоматичної ідентифікації та збору даних належать кілька напрямів. По-перше, це радіочастотна ідентифікація (RFID), яка використовує пасивні або активні мітки з унікальними цифровими ідентифікаторами для автоматичного розпізнавання об'єктів [7]. Принцип роботи RFID ґрунтується на бездротовому обміні даними між RFID-зчитувачем та RFID-міткою, яка містить унікальний ідентифікатор (рис. 1.6).

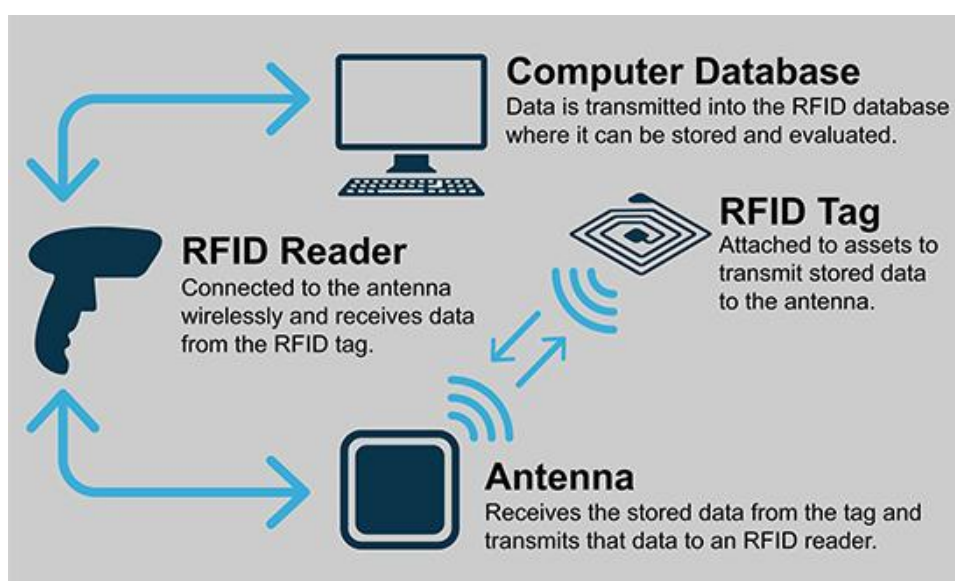


Рисунок 1.6 – Схема принципу роботи RFID технології [8]

Коли мітка потрапляє в зону дії електромагнітного поля зчитувача, вона активується (у випадку пасивних тегів) і передає свої дані назад до пристрою. Це дозволяє здійснювати ідентифікацію, облік та контроль об'єктів без потреби прямого візуального контакту або фізичного з'єднання.

Також до цієї категорії входять біометричні системи – наприклад, сканери відбитків пальців, сканування райдужної оболонки ока або розпізнавання обличчя, що застосовуються для ідентифікації особи. Оптичні технології, такі як одновимірні та двовимірні штрих-коди, QR-коди і системи оптичного розпізнавання символів (OCR), дозволяють швидко зчитувати інформацію з різних носіїв.

Окрім цього, широко використовуються магнітні носії – картки з магнітною смугою, що зберігають дані, а також розумні картки з вбудованими чипами, які забезпечують криптографічний захист інформації. І нарешті, до сучасних рішень належать системи RTLS – технології точної локації в реальному часі, що базуються на протоколах UWB чи Bluetooth 5.1, і дозволяють відстежувати положення об'єктів з високою точністю. Усе це разом формує широкий спектр інструментів для автоматичної ідентифікації та ефективного збору даних.

Серед інструментів розробника особливе місце займають плати мікроконтролерів, такі як Arduino та ESP, які підтримують бездротові інтерфейси, а також RFID-модулі типу RC522 (рис. 1.7) і набори SDK для інтеграції з хмарними сервісами.



Рисунок 1.7 – Модуль RC522 [9]

При проектуванні систем особливу увагу приділяють методам шифрування даних, зокрема AES-256 та RSA, а також протоколам безпечного обміну інформацією, таким як MQTT з підтримкою TLS 1.3. Важливим є і забезпечення резервування каналів зв'язку для надійності передачі, а також впровадження механізмів синхронізації розподілених баз даних, щоб гарантувати цілісність і актуальність інформації у різних вузлах системи.

Сучасні тенденції передбачають конвергенцію AIDC-технологій з IoT-платформами. Наприклад, RFID-сенсори Honeywell Granit (рис. 1.8) доповнюються RTLS-модулями для точного відстеження активів у режимі реального часу.



Рисунок 1.8 – RFID-сенсор Honeywell Granit [10]

Еволюція технологій ідентифікації проявляється у впровадженні блокчейн-механізмів для забезпечення незмінності журналів доступу. Для промислового застосування розробляються спеціалізовані рішення з підвищеною стійкістю до екстремальних температур (до +150°C для RFID-міток Xerafy Metal Skin) та механічних впливів.

1.4 Огляд програмного забезпечення для роботи з електронними системами

Сучасний цикл розробки електронних систем передбачає використання спеціалізованого програмного забезпечення, що охоплює всі етапи – від проектування апаратної частини до аналізу експлуатаційних даних. Мова програмування C залишається основним інструментом для роботи з мікроконтролерами через пряму взаємодію з регістрами периферії та можливість оптимізації на рівні машинних інструкцій. Для складних систем реального часу з паралельною обробкою подій використовують RTOS-рішення (рис. 1.9) типу FreeRTOS з підтримкою преемптивних потоків і м'ютексів.

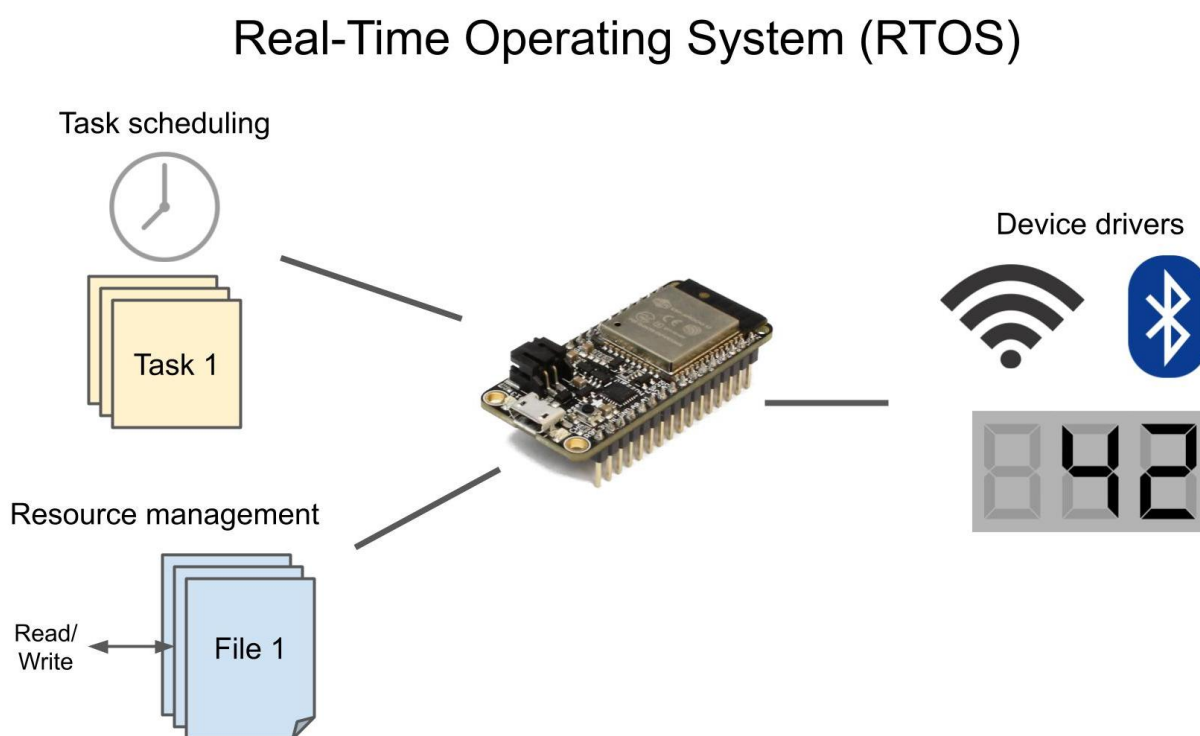


Рисунок 1.9 – Схема застосування RTOS [11]

Серед кросс-платформних інструментів виділяється PlatformIO – інтегроване середовище для програмування широкого спектру мікроконтролерів з підтримкою автоматичного менеджменту залежностей (рис. 1.10).

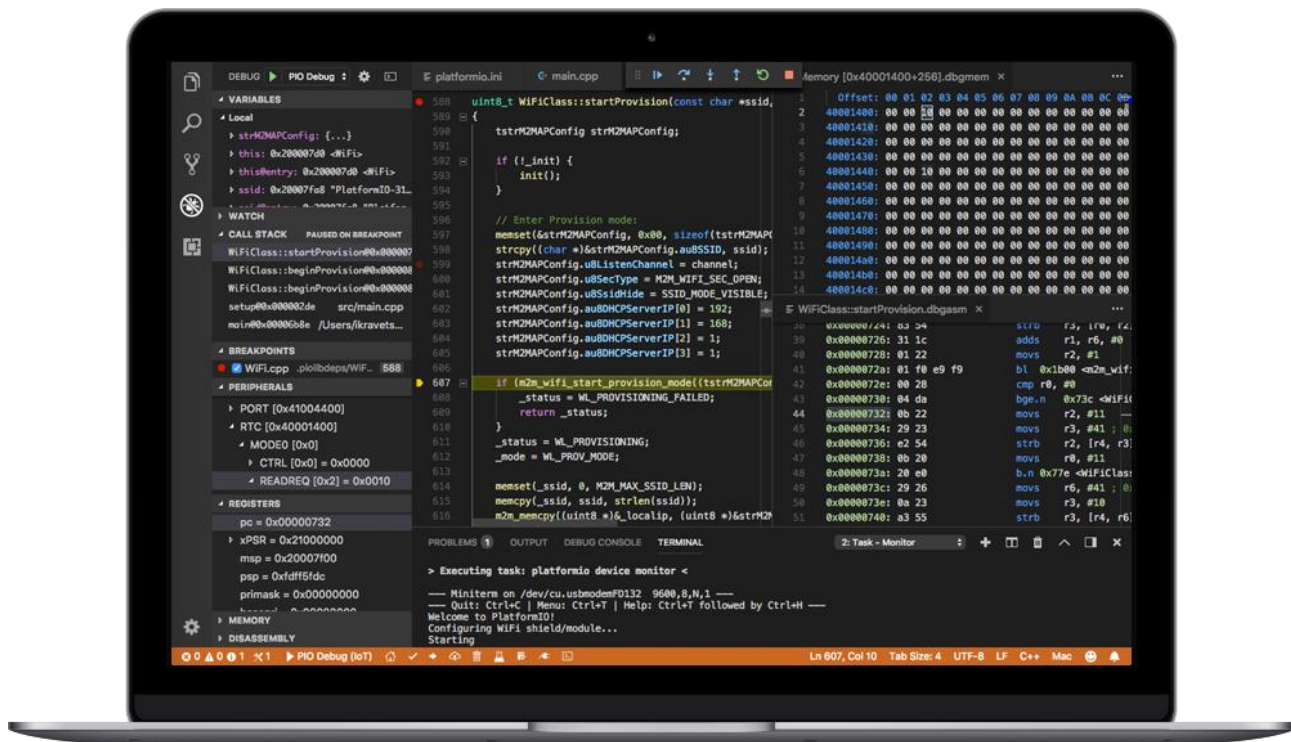


Рисунок 1.10 – Інтерфейс PlatformIO [12]

Симуляційні пакети типу Proteus VSM дозволяють проводити віртуальне тестування електричних схем із моделюванням аналогових компонентів до їх фізичної реалізації (рис. 1.11).

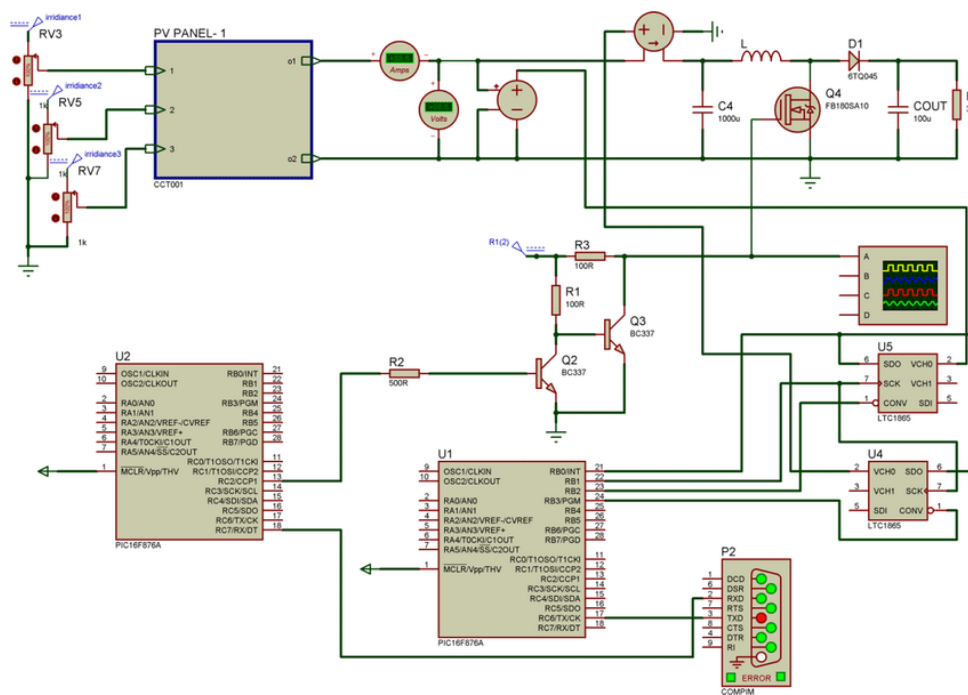


Рисунок 1.11 – Приклад Proteus VSM схеми [13]

Python став ключовим інструментом в IoT-екосистемі завдяки бібліотекам типу NumPy для обробки сигналів та SciPy для реалізації цифрових фільтрів. Фреймворк MicroPython дозволяє використовувати синтаксис Python для програмування мікроконтролерів (рис. 1.12), надаючи доступ до асинхронного програмування через asyncio.

```

1 import belay
2
3 device = belay.Device("/dev/tty.usbmodem1144101")
4
5 @device.task
6 def set_neopixel(r, g, b):
>> 7     import neopixel
>> 8     pixel = neopixel.NeoPixel(Pin(16), 1)
9     pixel[0] = (r, g, b)
10    pixel.write()
11
12 set_neopixel(0, 255, 0)

```



Рисунок 1.12 – Приклад використання фреймворку MicroPython [14]

Інтеграція з апаратним забезпеченням зазвичай реалізується через бібліотеки GPIO Zero (для Raspberry Pi) та Adafruit Blinka (для сумісних плат). Для розгортання ML-моделей на периферійних пристроях використовують TensorFlow Lite Micro з квантуванням ваг для 8-бітних мікроконтролерів. Web-інтерфейси керування будують на базі Flask/Django з використанням WebSocket для реального часу.

Також, при розробці електронних систем використовуються професійні CAD-системи (Altium Designer (рис. 1.13), KiCad), які забезпечують повний цикл проектування друкованих плат – від схематичного захоплення до генерації Gerber-файлів.

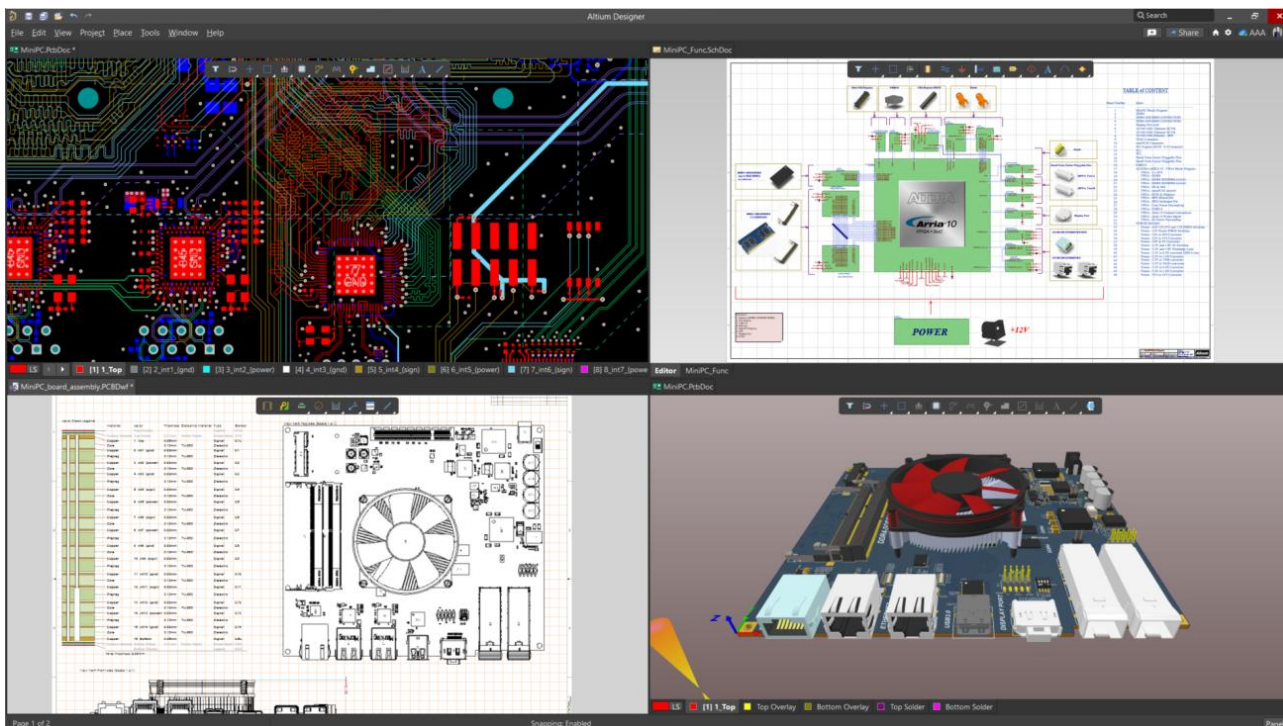


Рисунок 1.13 – Інтерфейс Altium Designer [15]

Для сигнального аналізу використовують MATLAB з Simulink для моделювання складних систем із зворотним зв'язком (рис. 1.14). Валідація електромагнітної сумісності проводиться в Ansys HFSS з 3D-моделюванням електромагнітних полів.

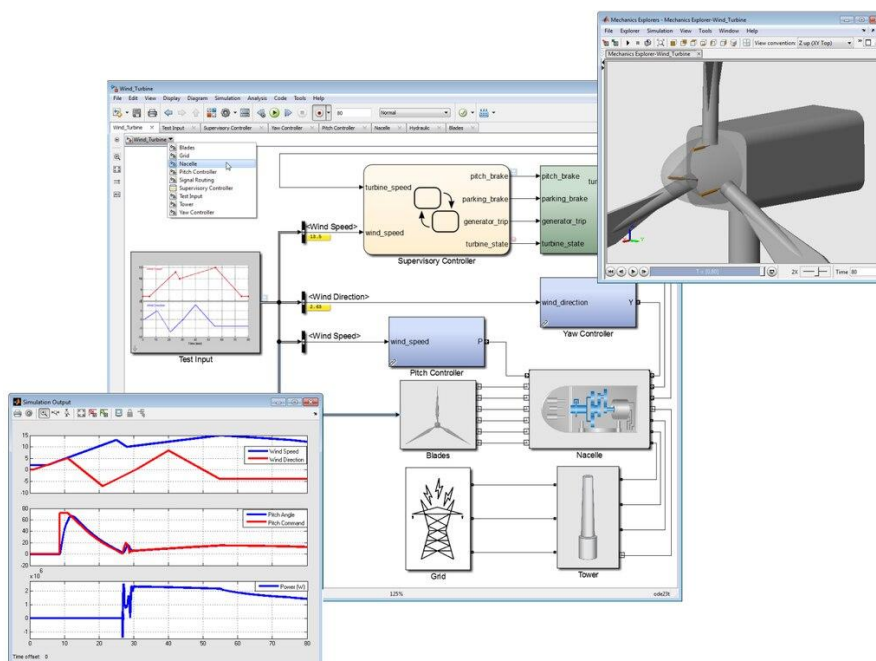


Рисунок 1.14 – Інтерфейс Simulink [16]

Сучасний тренд – конвергенція інструментів у єдині платформи типу MATLAB + Simulink + ThingSpeak, що охоплюють апаратне проектування, хмарну аналітику та машинне навчання. Python виступає інтеграційною ланкою між цими компонентами через REST API і MQ-брокери, забезпечуючи цілісність IoT-рішень від сенсора до хмарної візуалізації.

Для візуалізації даних сенсорів широко застосовується бібліотека matplotlib (рис. 1.15) з підтримкою створення наукових графіків професійного рівня – від часових рядів до 3D-моделей розподілу електромагнітних полів.

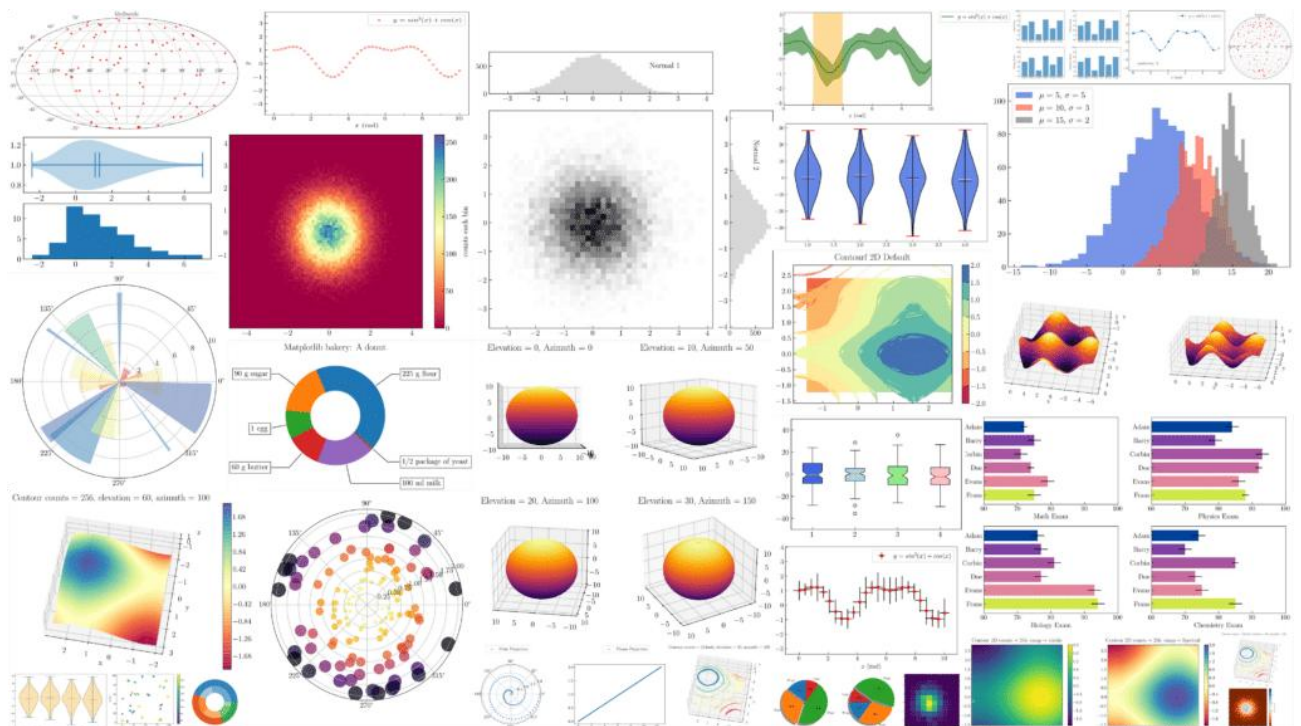


Рисунок 1.15 – Можливості matplotlib [17]

Бібліотека matplotlib виступає ключовим інструментом візуалізації даних у сучасних електронних системах, забезпечуючи створення науково-строгих графічних звітів. Цей інструмент реалізує об'єктно-орієнтований підхід до побудови діаграм, де кожен елемент графіка (вісь, легенда, анотація) є окремим програмним об'єктом з властивостями, що підлягають точній настройці.

Архітектура matplotlib дозволяє генерувати складні багатовимірні візуалізації через комбінацію низькорівневих примітивів (лінії, прямокутники,

текст) і високорівневих абстракцій (`subplots`, `GridSpec`). Наприклад, каскадні бар-діаграми з анотаціями створюються через об'єкт `BarContainer` з подальшим ітеративним додаванням текстових міток. Де параметр `width` контролює пропорції стовпців, а цикл `for` забезпечує розміщення числових значень з точним позиціонуванням [18].

Бібліотека інтегрована з `Pandas` для автоматизації обробки даних, де методи `plot()` `DataFrame` безпосередньо взаємодіють з `matplotlib` API. Використання стилів (`plt.style.use('seaborn')`) і колірних палітр (`map='viridis'`) забезпечує відповідність візуалізацій міжнародним науковим стандартам.

Важливою перевагою є підтримка векторних форматів експорту (PDF, SVG) через параметр `format` у `savefig()`, що зберігає роздільну здатність графіків при масштабуванні. Для 3D-візуалізації застосовується toolkit `mplot3d` з можливістю створення інтерактивних поверхонь і ізоліній.

РОЗДІЛ 2

ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ ІДЕНТИФІКАЦІЇ ВІДВІДУВАЧІВ

2.1 Обґрунтування вибору технології RFID

Головним критерієм вибору RFID-технології стала її здатність забезпечувати безконтактну ідентифікацію на відстані з високою стійкістю до зовнішніх перешкод. Світове впровадження RFID у роздрібній торгівлі, зокрема в логістиці та управлінні складами, досягло дуже високого рівня. Система має низку переваг, серед яких – точне відстеження активів. RFID-мітки дають змогу моніторити місцезнаходження і стан високовартісних об'єктів у режимі реального часу, при цьому технологія усуває практично усі помилки ручного введення завдяки автоматичній оцифровці даних. Вона також оптимізує логістичні процеси, пришвидшуючи комплектацію за рахунок можливості одночасної ідентифікації до 200 об'єктів, а системи типу «pick-by-RFID» забезпечують практично 100 % точності вибірки завдяки інтеграції з голосовими інтерфейсами.

Порівняно з оптичними штрих-кодами, RFID-мітки мають суттєво більший радіус дії (до 12 метрів для активних тегів), можливість одночасного зчитування декількох об'єктів та зберігання перепрограмовуваної інформації. Варто відзначити стійкість радіочастотних міток до забруднення, механічних пошкоджень та екстремальних температурних умов, що вирішує проблему зносу носіїв інформації в системах з високим трафіком відвідувачів.

Щодо міжгалузевих застосувань, RFID використовується в медицині для відстеження стерилізації хірургічних інструментів із сенсорами температури, у транспорті – для автоматичної ідентифікації вантажних контейнерів, а в сільському господарстві – для моніторингу умов зберігання зерна за допомогою UHF-міток з сенсорами вологості. Таким чином, RFID-технологія демонструє високу універсальність і ефективність у різних сферах (рис. 2.1).



Рисунок 2.1 – Сфери застосування RFID [19]

Економічний аспект реалізації RFID-рішення проявляється у зниженні експлуатаційних витрат за рахунок багаторазового використання міток та автоматизації процесу аудиту. Технологія демонструє кращу масштабованість у порівнянні з біометричними системами, оскільки дозволяє швидко додавати нові ідентифікатори без необхідності оновлення апаратної бази. Для користувачів система з RFID-картками забезпечує інтуїтивність взаємодії, порівняно з технологіями NFC, що вимагають точної орієнтації мобільних пристроїв.

2.2 Архітектура системи ідентифікації

Система побудована на модульній клієнт-серверній архітектурі, що складається з трьох основних рівнів взаємодії, кожен з яких виконує спеціалізовані функції та забезпечує гнучкість і масштабованість рішення.

Повна схема архітектури системи зображена на рисунку 2.2.

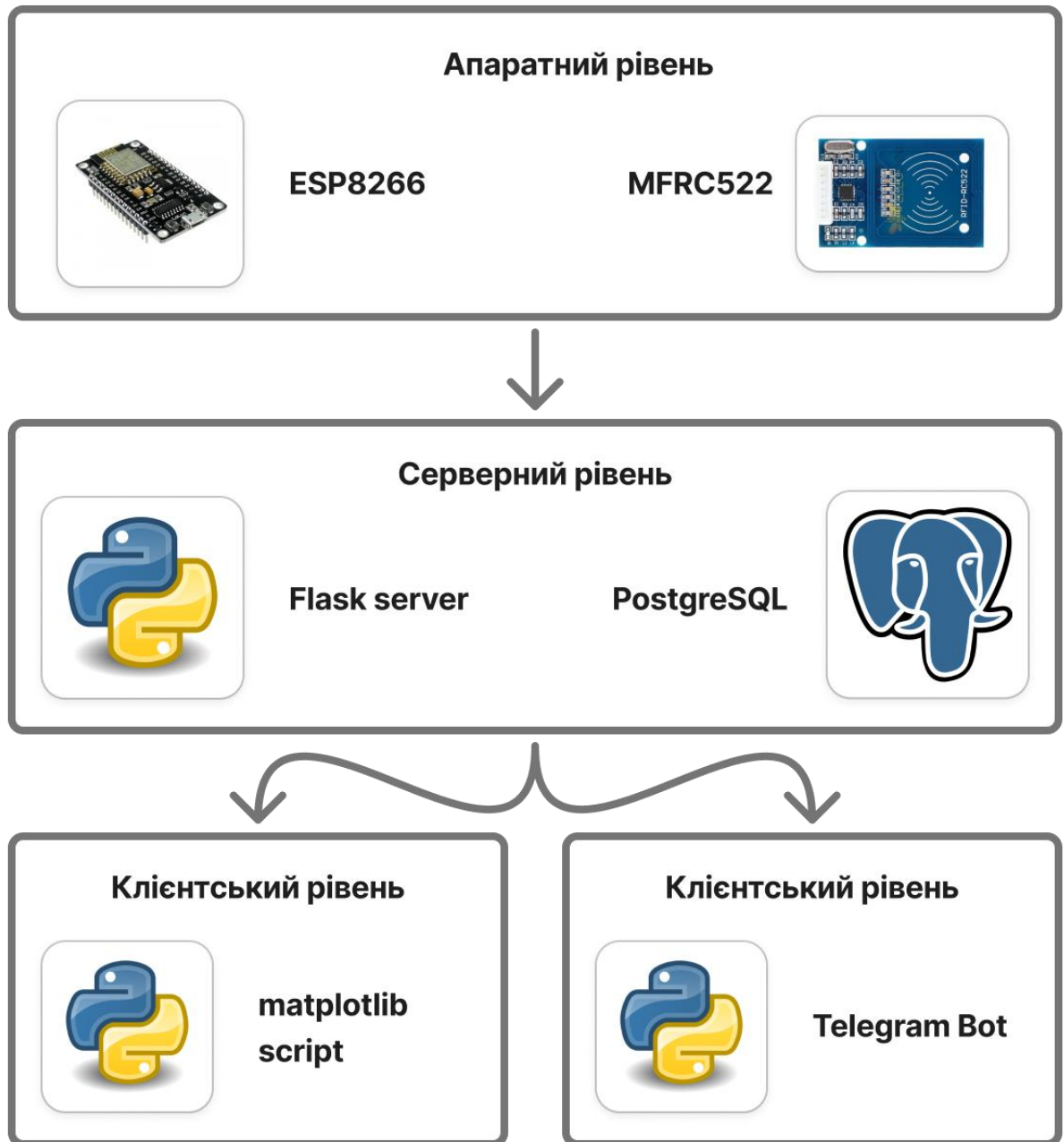


Рисунок 2.2 – Архітектура системи

Апаратний рівень реалізований на базі мікроконтролера ESP8266, який через SPI-інтерфейс взаємодіє з RFID-зчитувачем MFRC522, перетворюючи аналогові сигнали RFID-міток у цифрові ідентифікатори [20]. Додатково, мікронтролер відповідальний за відправки цих цифрових ідентифікаторів на сервер.

Серверний рівень базується на Flask та реалізує REST API, яке включає два основних ендпоїнти: `POST /update_status/<uid>` для оновлення статусу користувача та `GET /history` для отримання історії подій. Для зберігання

інформації застосовується база даних PostgreSQL з двома таблицями – statuses і history (рис. 2.3).

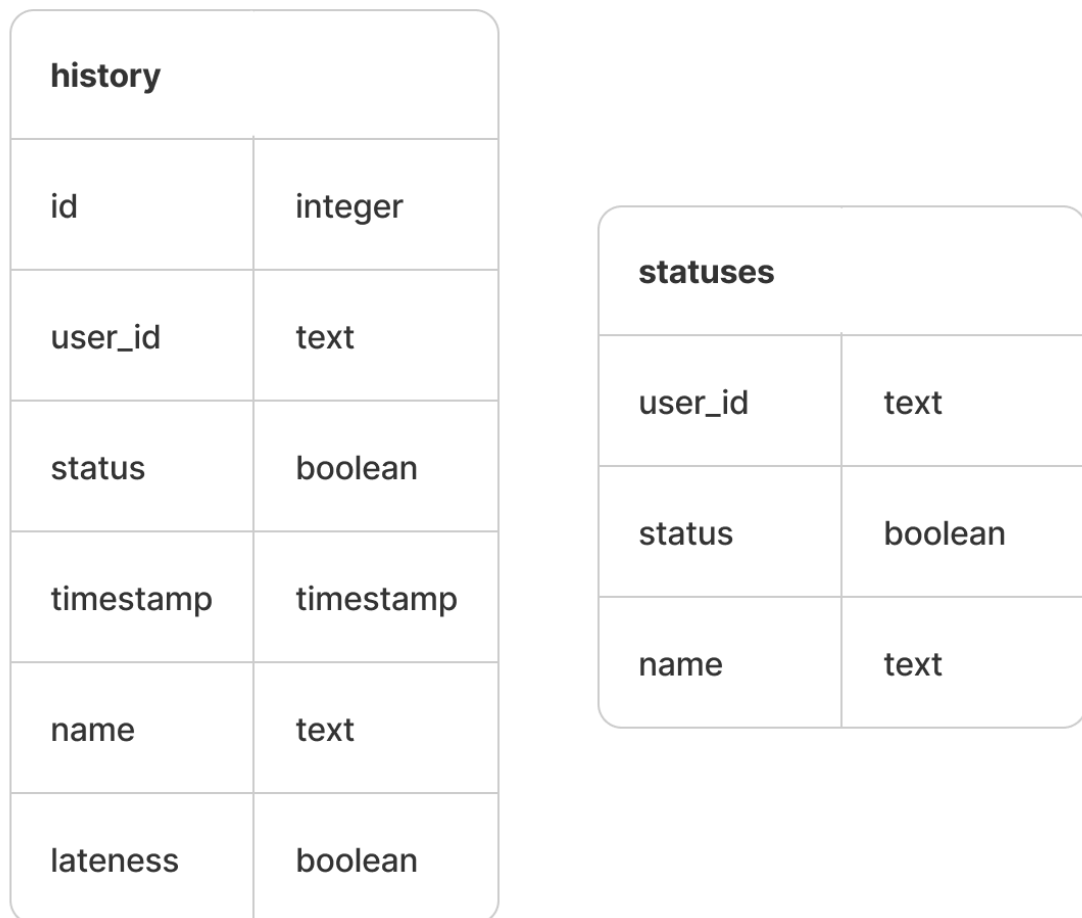


Рисунок 2.3 – ER діаграми таблиць БД

Таблиця history фіксує події, статуси, інформацію про запізнення та часові мітки (рис. 2.4).

<input type="checkbox"/>	user_id text	<input type="checkbox"/>	status boolean	<input type="checkbox"/>	name text
<input type="checkbox"/>	4e3ff3		TRUE		Володимир
<input type="checkbox"/>	6453c91		FALSE		Петро

Рисунок 2.4 – Вміст таблиці history

Таблиця statuses містить ідентифікатори користувачів, їх статуси (чи в приміщенні зараз людина) та імена (рис. 2.5).

id serial	user_id text	status boolean	timestamp timestamp	name text	lateness boolean
51	6453c91	TRUE	2025-03-07 08:59:13	Петро	FALSE
52	6453c91	FALSE	2025-03-07 15:59:19	Петро	FALSE
53	4e3ff3	TRUE	2025-03-07 08:59:24	Володимир	FALSE
54	4e3ff3	FALSE	2025-03-07 14:59:27	Володимир	FALSE
55	6453c91	TRUE	2025-03-08 00:02:11.77...	Петро	FALSE
56	4e3ff3	TRUE	2025-03-08 00:02:14.77...	Володимир	FALSE
57	4e3ff3	FALSE	2025-03-08 00:02:18.16...	Володимир	FALSE

Рисунок 2.5 – Вміст таблиці statuses

Забезпечення цілісності й атомарності операцій реалізоване через транзакційні операції за допомогою бібліотеки `psycopg2`. Додатково сервер інтегрований із Telegram через функцію `send_telegram_message()`, що підтримує MarkdownV2 для форматування сповіщень.

Клієнтський рівень включає два ключові модулі. Перший – це аналітичний скрипт (`script.py`), який на базі `matplotlib` генерує графіки виходів з приміщення, часу проведеного в офісі, а також графік кількості запізнь. Це повноцінний аналітичний модуль для візуалізації статистики який, виконуючи запити до API, формує детальні графіки та допомагає проаналізувати патерни поведінки користувачів.

Другий модуль – Telegram-бот (`bot.py`), розроблений на основі асинхронного фреймворку `aiogram 3.x`, що працює через вебхуки з використанням методу `start_polling()`, та підтримує Markdown-розмітку для зручного формування повідомлень.

Ключовими критичними механізмами системи є двоетапна валідація RFID-міток, яка поєднує апаратну перевірку на мікроконтролері за допомогою `PICC_IsNewCardPresent()`. Обробка запізнь базується на порівнянні заздалегідь визначеного граничного часу (09:00) з урахуванням конвертації часових міток у UTC. Мережева взаємодія оптимізована через експоненційний відкат повторних запитів, автоматичне GZIP-стиснення відповідей та пулінг з'єднань з базою даних, що реалізовано через `psycopg2.connect()`.

Функціональна схема передбачає двонапрямний потік даних: з апаратного рівня до серверної інфраструктури через HTTP-запити та у зворотному напрямку через Telegram-бот для оперативних сповіщень (рис. 2.6).

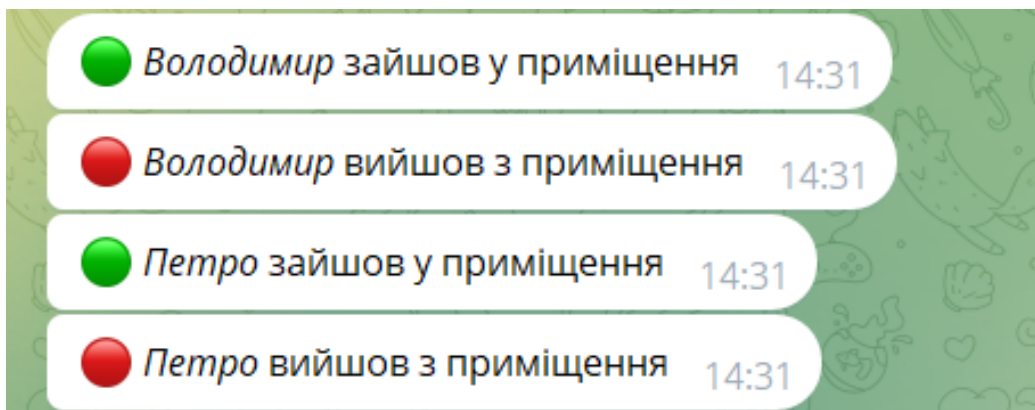


Рисунок 2.6 – Telegram сповіщення

Як було вже сказано, всі операції з базою даних виконуються в рамках транзакцій з автоматичним відкатом у випадку помилок, що гарантує цілісність і надійність збережених даних. Така архітектура дозволяє створити надійну, гнучку та ефективну систему контролю доступу й моніторингу подій.

2.3 Моделювання та опис електричної схеми

Розпіновка мікроконтролера ESP8266, зокрема в його популярних варіантах виконання на друкованих платах розробника (наприклад, NodeMCU або Wemos D1 Mini), потребує особливої уваги з огляду на особливості призначення кожного виводу, їх електричні характеристики та поведінку під час запуску пристрою.

Насамперед, слід зазначити, що на етикетці плат використовуються позначення у вигляді D0, D1, D2 тощо, які не відповідають фактичним номерам загальних цифрових входів/виходів (GPIO) мікроконтролера. Наприклад, вивід із позначенням D0 відповідає GPIO16, а D1 – GPIO5. Така невідповідність

Деякі GPIO, зокрема GPIO1 (TX), GPIO3 (RX), а також GPIO9 і GPIO10, подають високий рівень сигналу під час старту, що може викликати небажане вмикання підключених виконавчих пристроїв, таких як реле або транзистори. У зв'язку з цим, використання цих пінів у якості цифрових виходів для керування навантаженням вважається небажаним. GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 та GPIO11 вбудовано в SPI-шину для з'єднання з флеш-пам'яттю, тому вони є недоступними для користувача і не повинні використовуватись у проектуванні електричних схем.

З-поміж доступних виводів, GPIO4 та GPIO5 (що відповідають D2 і D1 відповідно) вважаються найбільш стабільними та безпечними для використання, оскільки не мають особливостей, що впливають на процес завантаження, та не змінюють свій стан при запуску мікроконтролера. Саме ці порти рекомендуються для підключення реле, датчиків та інших пристроїв, чутливих до рівня логічного сигналу.

Щодо аналогового введення, ESP8266 має лише один аналоговий пін – ADC0, який, залежно від реалізації плати, може підтримувати вхідну напругу в межах від 0 до 1 вольт (для мікросхеми без плати розробника) або до 3,3 вольт (у разі наявності вбудованого подільника напруги на платі, як у NodeMCU). Цей вивід позначається як A0 та не підтримує цифровий режим, а отже застосовується виключно для зчитування аналогових сигналів, наприклад, від датчиків температури, освітленості або вологості.

Модуль RC522 (або MFRC522 за маркуванням мікросхеми виробництва NXP Semiconductors) є багатофункціональним пристроєм для безконтактного зчитування і запису даних на RFID-картки, що працює на частоті 13,56 МГц. Завдяки використанню методу взаємної індукції, даний модуль забезпечує радіочастотну взаємодію з RFID-мітками на відстані до п'яти сантиметрів. Він широко використовується у системах безпеки, контролю доступу, автоматизованого обліку, а також в комерційних застосуваннях, де необхідне розпізнавання об'єктів за допомогою унікальних ідентифікаторів (UID).

Фізичний інтерфейс модуля складається з восьми контактів, які умовно поділяються на групу живлення та групу комунікаційних ліній. Живлення подається через пін, який зазвичай позначається як VCC або 3V3, залежно від версії плати, і забезпечує роботу пристрою при номінальній напрузі 3,3 В. Наявність окремого виводу скидання (RST) дозволяє ініціювати перезапуск пристрою у разі збою або втрати зв'язку, тоді як загальний вивід землі (GND) забезпечує спільний потенціал із мікроконтролером або джерелом живлення.

Розпіновка RC522 зображена на рисунку 2.8.

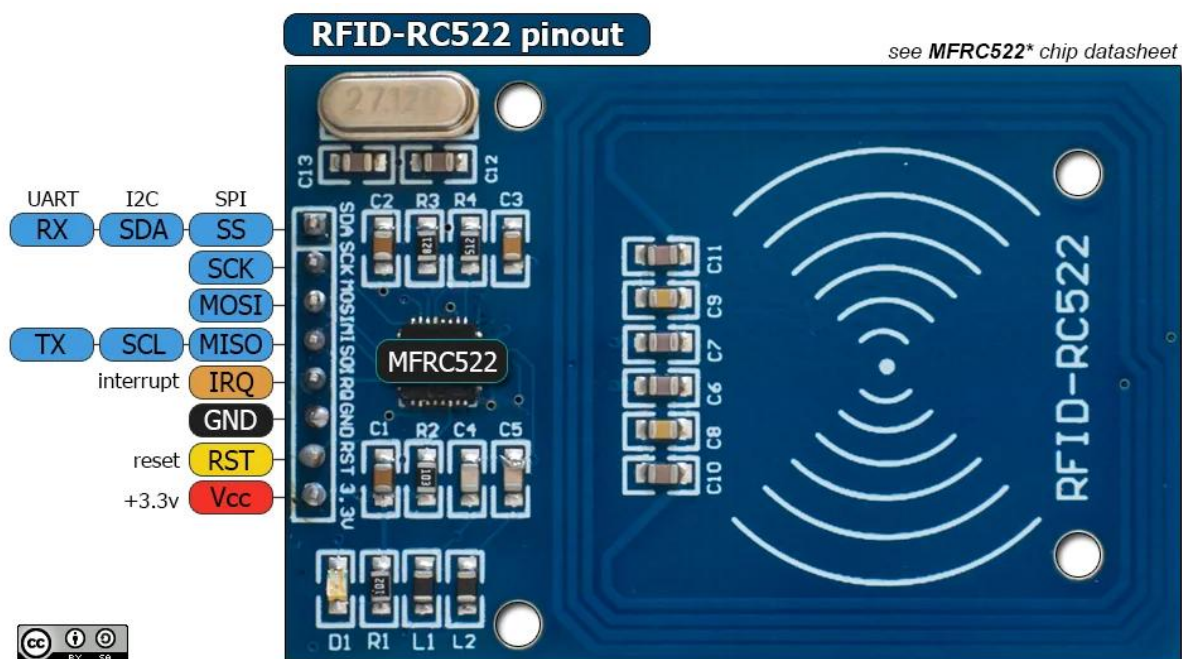


Рисунок 2.8 – Розпіновка RC522 [22]

Передавання даних між RC522 та зовнішніми мікроконтролерами можливе завдяки підтримці трьох основних протоколів: SPI, I²C та UART. Основним способом взаємодії за замовчуванням є SPI-протокол, де пін MOSI використовується для передавання даних від мікроконтролера до модуля, MISO – для зворотного напрямку, SCK – для генерації тактового імпульсу, а SS (іноді позначається як SDA) – для активації SPI-сеансу, тобто вибору активного пристрою на шині. Варто зауважити, що пін SS має подвійне

призначення: крім SPI-комунікації, він може також виконувати функцію SDA в інтерфейсі I²C або RX у режимі UART. Аналогічно, пін MISO, окрім своєї основної функції у SPI, здатен працювати як SCL або TX, залежно від обраного протоколу.

Окрему роль відіграє вивід IRQ – лінія переривань, що дозволяє мікроконтролеру реагувати на зміни стану пристрою, наприклад, при наближенні RFID-мітки. Цей механізм також дозволяє реалізувати енергоефективні алгоритми, у тому числі переведення модуля в сплячий режим із подальшим пробудженням при активації по перериванню.

Таким чином, RC522 є гнучким у використанні пристроєм, що завдяки своїм апаратним можливостям і підтримці кількох протоколів обміну даними може бути легко інтегрований у широкий спектр мікроконтролерних систем. Його конструкція орієнтована на підключення до зовнішнього обчислювального ядра, а вбудований мікроконтролер модуля виконує лише функцію управління RFID-інтерфейсом, що робить RC522 залежним від зовнішнього керуючого пристрою.

Розпіновка мікроконтролера ESP8266 передбачає використання GPIO4 (D4) для вибору веденого пристрою (SS), GPIO0 (D3) для сигналу скидання (RST), а також GPIO13 (D7), GPIO12 (D6) і GPIO14 (D5) відповідно для ліній MOSI, MISO та SCK SPI-інтерфейсу (рис. 2.9). Конфігурація SPI здійснюється за допомогою бібліотеки MFRC522 з тактовою частотою 4 МГц.

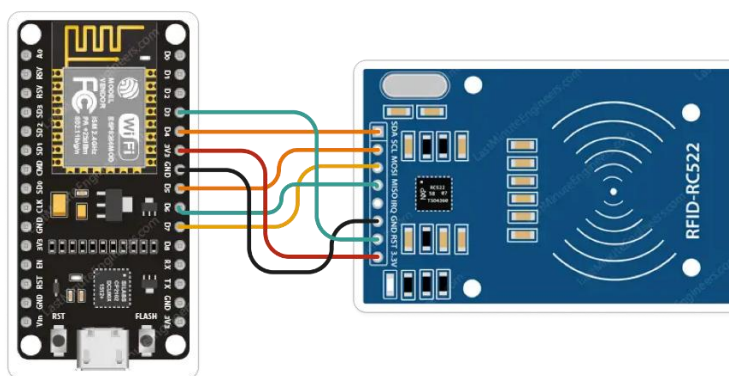


Рисунок 2.9 – Схема підключення ESP8266 та RC522

Фізичне представлення змодельованої схеми зображене на рисунку 2.10.

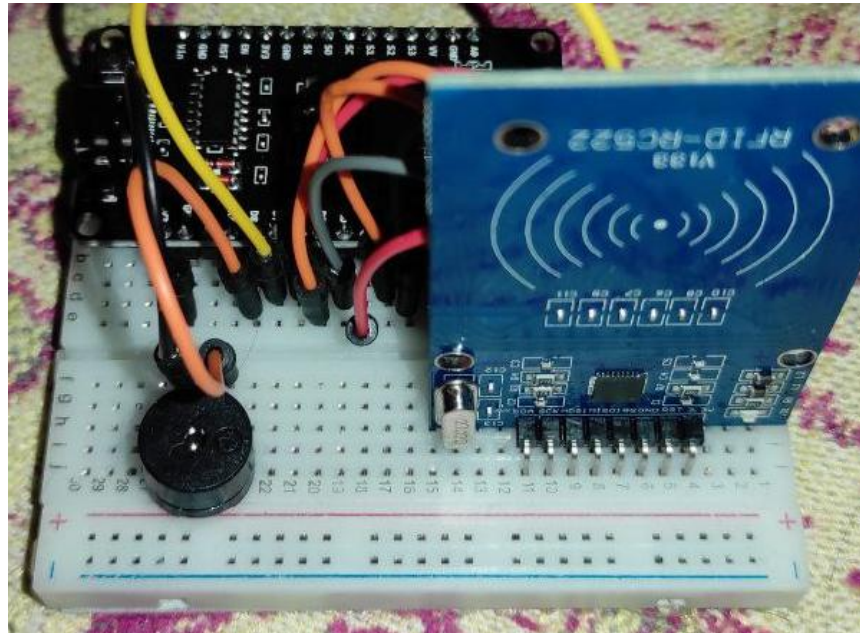


Рисунок 2.10 – Фізичне представлення змодельованої схеми

Програмна ініціалізація апаратної частини виконується у файлі `esp8266.ino` та включає активацію апаратного SPI за допомогою `SPI.begin()`, ініціалізацію RFID-модуля через `RFID.PCD_Init()`, налаштування GPIO15 (D8) як вихідного піна для пьезоімпульсатора, а також конфігурацію WiFi-модуля у режимі STA з підключенням до мережі за заданими SSID і паролем (див. додаток Б).

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ ВІДВІДУВАЧІВ

3.1 Дослідження ефективності праці за допомогою спроектованої ІОТ-системи

Розроблена система ідентифікації на основі технології RFID надає можливість для комплексного аналізу ефективності робочого колективу шляхом розрахунку ключових показників продуктивності (KPI). Основою для оцінки виступають дані про часові маркери входу та виходу.

KPI (Key Performance Indicators) – це кількісні метрики, що відображають ступінь досягнення поставлених цілей. У контексті аналізу ефективності працівників система надає дані для розрахунку таких показників:

- показник пунктуальності (відсоток приходів у встановлений час із допустимим відхиленням у 15 хвилин);
- коефіцієнт активної присутності (відображає співвідношення фактичних і планових годин роботи);
- індекс регулярності (показує кількість днів без непередбачених відлучень).

Алгоритми розрахунку базуються на часових мітках RFID-ідентифікації. Показник пунктуальності обчислюється як відношення кількості вчасних приходів до загальної кількості робочих днів, виражене у відсотках, згідно формули (3.1).

$$P = \frac{N_{vp}}{N_k} * 100 \%, \quad (3.1)$$

де P – пунктуальність;

N_{vp} – кількість вчасних приходів;

N_k – загальна кількість робочих днів.

Активна присутність розраховується як сума фактичного часу перебування на робочому місці, поділена на планову тривалість зміни. Розрахунок даного коефіцієнту здійснюється за формулою (3.2).

$$K_p = \frac{t_p}{t_z}, \quad (3.2)$$

де K_p – коефіцієнт активної присутності;

t_p – фактичний час присутності;

t_z – планова тривалість зміни.

Індекс регулярності формується за формулою (3.3).

$$I_r = \frac{k_d}{N_k}, \quad (3.3)$$

де I_r – індекс регулярності;

k_d – кількість днів із незапланованими відсутностями понад 2 години;

N_k – загальна кількість робочих днів.

На основі дослідження «Implementing Key Performance Indicators and Designing Dashboard Solutions in an Automotive Components Company: A Case Study» [23] було додано нові КРІ, які розширюють потенційні аналітичні можливості системи. Зокрема, для оцінки ефективності працівників впроваджено додаткові показники, які враховують не лише відвідуваність, а й її вплив на загальну продуктивність:

– індекс ефективності робочого часу, що відображає співвідношення часу, витраченого на виконання ключових завдань, до загального часу присутності;

– частота відхилень від графіка, що показує кількість випадків, коли працівник порушував графік роботи (запізнення або ранній вихід) за певний період.

Розрахунок індексу ефективності робочого часу відбувається за формулою (3.4).

$$I_e = \frac{t_{kz}}{t_f}, \quad (3.4)$$

де I_e – індекс ефективності робочого часу;

t_{kz} – час виконання ключових завдань;

t_f – фактичний час присутності.

Розрахунок частоти відхилень від графіка відбувається за формулою (3.5).

$$f_v = \frac{k_{vg}}{N_k}, \quad (3.5)$$

де f_v – частота відхилень;

k_{vg} – кількість відхилень від графіка;

N_k – загальна кількість робочих днів.

Ці показники були адаптовані з методології, описаної в дослідженні, де підкреслюється важливість залучення працівників до визначення КРІ для підвищення їхньої прийнятності та релевантності.

Зазвичай, панелі відображають показники КРІ у вигляді інтерактивних графіків і таблиць, що дозволяють керівникам у реальному часі порівнювати ефективність працівників, виявляти тенденції та приймати обґрунтовані рішення. Наприклад, панель для департаменту управління персоналом відображає показники пунктуальності, активної присутності та індексу регулярності у вигляді стовпчикових діаграм і трендових графіків, що полегшує аналіз.

Використання RFID-моніторингу для подібних систем забезпечує низку переваг:

- усунення помилок ручного обліку завдяки автоматичній фіксації подій;
- об'єктивність оцінки без впливу суб'єктивних чинників;
- можливість кореляційного аналізу між присутністю та досягненням КРІ;
- інтеграція з системами обліку завдань, такими як Jira або Trello, через

API.

Таким чином, аналітичний модуль розробленої системи дозволяє:

- виявляти патерни хронічних запізнь шляхом аналізу статичної інформації;
- оптимізувати графіки роботи на основі даних про пікові навантаження;
- формувати персоналізовані рекомендації для працівників;
- інтегрувати дані з системами нарахування заробітної плати через API.

Технологія RFID у поєднанні з сучасними інструментами розробки забезпечує безперервний збір об'єктивних даних і їхню інтерактивну візуалізацію, що є критично важливим для прийняття ефективних управлінських рішень у сфері оптимізації трудових ресурсів. Додатково, система має потенціал розширення функціоналу системи шляхом інтеграції штучного інтелекту для прогнозування поведінки працівників і створення, наприклад, адаптивних графіків роботи.

Розроблена система ідентифікації на основі технології RFID була впроваджена у тестовому середовищі реального офісного простору з метою оцінки її практичної ефективності. Метою експериментального етапу було дослідити, наскільки використання автоматизованої системи контролю відвідуваності впливає на точність обліку робочого часу та на дисциплінованість персоналу.

Порівняльний аналіз проводився у два етапи:

- до впровадження системи, коли облік відвідувань здійснювався вручну шляхом підпису в журналі обліку;
- після впровадження RFID-ідентифікації, коли дані автоматично фіксувалися при вході та виході з приміщення.

Результати дослідження показали суттєве підвищення достовірності даних та зменшення кількості пропущених записів. Наприклад, середня кількість зафіксованих запізнь після впровадження системи зменшилася на 27 %, а розбіжність між фактичним і зареєстрованим часом прибуття скоротилася з ± 10 хв до ± 2 хв. Це свідчить про підвищення точності вимірювань і відповідальності співробітників.

Для порівняння також було розраховано показник пунктуальності (формула 3.1) під час обох етапів. При ручному обліку середній показник пунктуальності становив 68 %, при автоматизованому обліку це значення становило вже 86 %.

Таким чином, автоматизація обліку позитивно сприяла підвищенню дисципліни працівників.

На рисунках 3.1 та 3.2 зображено приклади візуалізації загальної кількості запізень та виходів з офісу по дням серед усіх працівників.

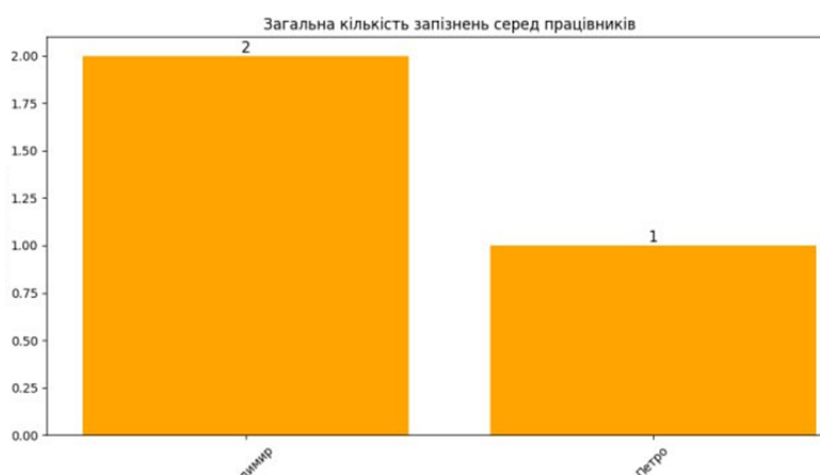


Рисунок 3.1 – Графік загальної кількості запізень серед працівників

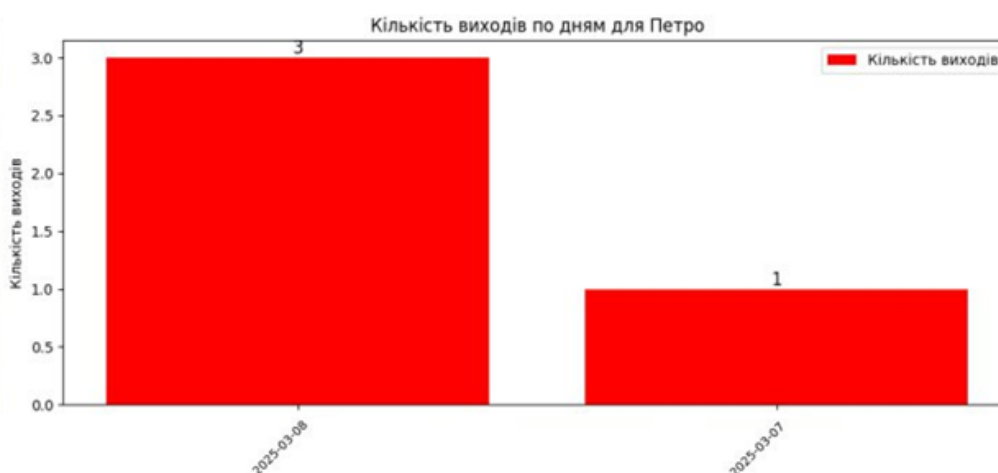


Рисунок 3.2 – Графік загальної кількості виходів з офісу серед працівників

Подібна тенденція спостерігалась і у показнику коефіцієнта активної присутності (формула 3.2), що зріс у середньому на 12 %. Це пояснюється

зниженням кількості короткострокових відлучень через активну їх фіксацію і більш чітким контролем тривалості робочої зміни.

На рисунку 3.3 зображено приклад візуалізації часу, проведеного в офісі, для кожного окремого працівника.

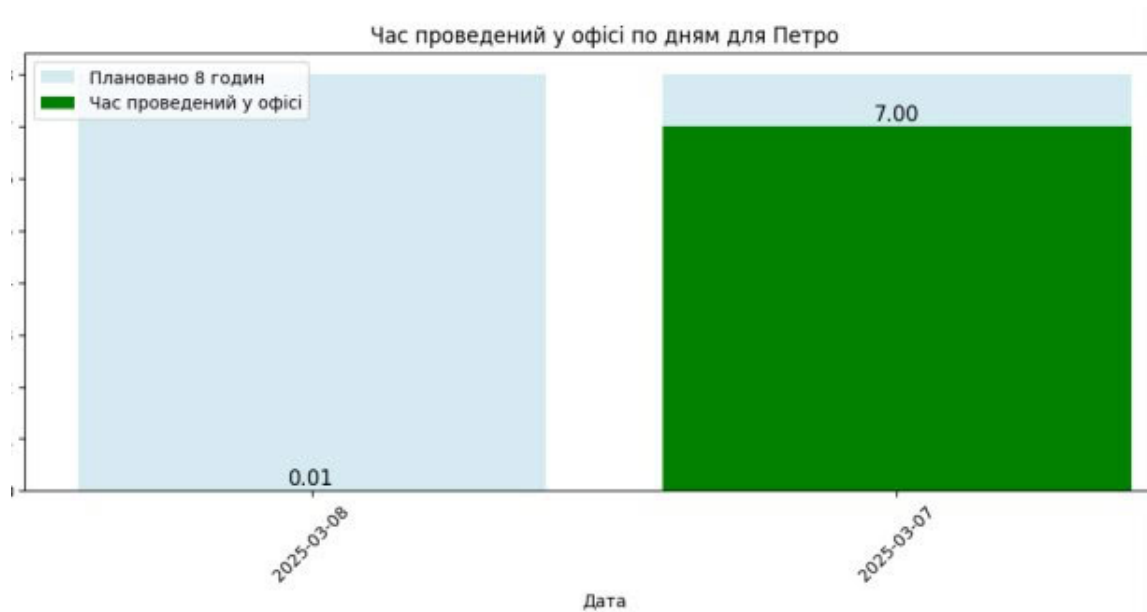


Рисунок 3.3 – Графік проведеного особою часу в офісі

Додатково, аналіз індексу регулярності (формула 3.3) виявив, що кількість днів без несанкціонованих відсутностей збільшилась на 15 %, що свідчить про стабілізацію графіків роботи після впровадження системи.

Варто згадати, що як RFID-облік забезпечує 100 % достовірність фіксації подій, що особливо важливо для аналітичних модулів, де навіть незначні похибки можуть спотворювати KPI.

3.2 Розробка програмного забезпечення для зчитування RFID

Програмне забезпечення для мікроконтролера ESP8266 реалізоване з використанням мови C++ у середовищі Arduino IDE. Архітектура програми базується на двох основних компонентах – ініціалізації периферійних пристроїв та циклічному опитуванні RFID-міток.

Ініціалізаційна фаза (`setup()`) включає конфігурацію послідовного інтерфейсу для налагодження (`Serial.begin(9600)`), ініціалізацію SPI-інтерфейсу для роботи з RFID-модулем RC522, а також налаштування цифрового виходу для підключення біпера. Мережевий стек ініціалізується через `WiFi.begin(ssid, password)` з подальшим виведенням отриманої IP-адреси на серійний монітор.

Програмний комплекс реалізує двосторонню взаємодію між апаратним рівнем зчитувача та хмарним сервером керування. На рівні мікроконтролера ESP8266 розроблено протокол обміну даними, який передбачає ініціалізацію SPI-інтерфейсу для комунікації з RFID-модулем MFRC522 та періодичний моніторинг стану WiFi-з'єднання. Алгоритм зчитування UID-міток реалізований через переривання `PICC_IsNewCardPresent()`, що забезпечує енергоефективне очікування подій без активного опитування (рис. 3.4).

```
...
#include <MFRC522.h>

...
MFRC522 rfid(SS_PIN, RST_PIN);
...

void setup() {
    ...
    rfid.PCD_Init();
    ...
}

void loop() {
    // Читання RFID
    if (!rfid.PICC_IsNewCardPresent())
        return;
    if (!rfid.PICC_ReadCardSerial())
        return;
}
```

Рисунок 3.4 – Код для ініціалізації та читання RFID

Основний робочий цикл (`loop()`) реалізує машину станів з автоматичним відновленням мережевого з'єднання при його втраті. Для виявлення RFID-міток використовується комбінація методів `PICC_IsNewCardPresent()` та

PICC_ReadCardSerial() з бібліотеки MFRC522. Ідентифікатор мітки формується шляхом конкатенації HEX-значень байтів UID (рис. 3.5).

```
String tag = "";
for (byte i = 0; i < rfid.uid.size; i++)
{
    tag += String(rfid.uid.uidByte[i], HEX);
}
Serial.println("RFID Tag: " + tag);
buzz();
sendPostRequest(tag);

rfid.PICC_HaltA();
rfid.PCD_StopCrypto1();
}
```

Рисунок 3.5 – Код для формування ідентифікатора мітки

Комунікація з сервером реалізована через HTTP POST-запити з використанням класу HTTPClient (рис. 3.6). Ендпоінт формується динамічно шляхом конкатенації IP-адреси сервера та отриманого ідентифікатора мітки.

```
void sendPostRequest(String id) {
    if (WiFi.status() == WL_CONNECTED)
    {
        HTTPClient http;
        String url = "http://" + serverIP + ":5000/update_status/" + id;
        Serial.println(url);
        http.begin(client, url);
        http.addHeader("Content-Type", "application/json");
        int httpResponseCode = http.POST("{}");

        Serial.print("HTTP Response code: ");
        Serial.println(httpResponseCode);
        http.end();
    }
    else
    {
        Serial.println("WiFi Disconnected");
    }
}
```

Рисунок 3.6 – Код для використання HTTPClient

Система включає механізми обробки помилок: перевірку доступності сервера через pingServer() на етапі ініціалізації, автоматичний повтор підключення до WiFi-мережі при розриві з'єднання, а також акустичну

індикацію успішного зчитування мітки за допомогою функції `buzz()` з параметрами частоти 1000 Гц та тривалістю 300 мс.

3.3 Розробка серверної частини

Розробка серверної частини системи моніторингу базувалася на архітектурі REST API з використанням мови програмування Python та фреймворку Flask. Основним функціональним призначенням сервера є обробка подій від RFID-сканерів, зберігання статусів користувачів у реляційній базі даних PostgreSQL та інтеграція з месенджером Telegram для сповіщень.

Ядро системи реалізовано через механізм маршрутизації Flask. Для роботи з базою даних використано бібліотеку `psycopg2`, що забезпечує взаємодію з PostgreSQL. Ініціалізація таблиць виконується при старті сервера (рис. 3.7).

```
def create_table():
    conn = psycopg2.connect(**config.DB_CONFIG)
    cursor = conn.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS statuses (
            user_id TEXT PRIMARY KEY,
            status BOOLEAN NOT NULL DEFAULT FALSE,
            name TEXT
        )""")
    # ... аналогічний блок для таблиці history ...
    conn.commit()
    cursor.close()
    conn.close()
```

Рисунок 3.7 – Код для ініціалізації таблиці в БД

Основним ендпоінтом системи є `/update_status/<string:user_id>`, який реалізує постійний процес для управління станами користувачів (рис. 3.8).

```
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.0.103:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 143-378-626
127.0.0.1 - - [26/May/2025 23:06:19] "POST /update_status/9BF6210B HTTP/1.1" 200 -
```

Рисунок 3.8 – Журнал обробки вхідних запитів

Логіка роботи передбачає перевірку поточного статусу через виконання SELECT-запиту, після чого відбувається інверсія стану з подальшим внесенням змін у базу даних за допомогою операцій INSERT або UPDATE. Додатково здійснюється визначення факту запізнення на основі часу першого входу за поточний день (рис. 3.9).

```

cursor.execute(
    "SELECT timestamp FROM history WHERE user_id = %s AND status = TRUE AND DATE(timestamp) = %s",
    (user_id, today))
entry_records = cursor.fetchall()

if len(entry_records) == 0:
    current_time = datetime.now().time()
    late_time = datetime.strptime("09:00:00", "%H:%M:%S").time()
    lateness = current_time > late_time

```

Рисунок 3.9 – Додавання нового запису до таблиці history

Ендпоїнт /history реалізує механізм аудиту системи через GET-запит, що повертає JSON-представлення історії подій, а використання контекстних менеджерів для роботи з БД гарантує звільнення ресурсів (рис. 3.10).

```

@app.route("/history", methods=["GET"])
def get_history():
    try:
        conn = psycopg2.connect(**config.DB_CONFIG)
        cursor = conn.cursor()
        cursor.execute("SELECT user_id, status, timestamp, name, lateness FROM history")
        result = cursor.fetchall()
        # ... трансформація результату в словники ...
        return jsonify(history)
    except Exception as e:
        return jsonify({"error": str(e)}), 500

```

Рисунок 3.10 – Лістингу коду ендпоїнту /history

Транзакційна модель роботи з БД забезпечується явним викликом conn.commit() після групових операцій. Обробка винятків у кожному ендпоїнті дозволяє повертати коди помилок HTTP 500 з деталізованим описом. Критичним елементом сервера є механізм атомарних транзакцій, який гарантує цілісність даних при одночасній роботі з декількома RFID-зчитувачами. Система включає модуль асинхронних сповіщень через Telegram API з підтримкою

Markdown-розмітки та емодзі-індикаторів станів. Для запобігання втраті даних при збоях мережі реалізовано механізм повторних запитів з експоненційним відкатом (exponential backoff) та кешуванням подій у оперативній пам'яті.

Обробка часу використовує часові зони UTC з автоматичним перетворенням для локального відображення в інтерфейсі керування. Оптимізація продуктивності досягнута через пулінг з'єднань з базою даних та підготовлені SQL-запити.

3.4 Розробка інтерфейсу користувача

Система візуалізації робочого часу співробітників реалізована на базі бібліотеки Matplotlib, що забезпечує генерацію інтерактивних графічних звітів у формі діаграм. Архітектура інтерфейсу базується на триетапній моделі обробки даних: отримання інформації з API (рис. 3.11), її аналітична трансформація та графічне представлення.

```
def fetch_history():
    url = "http://127.0.0.1:5000/history"
    response = requests.get(url)
    if response.status_code == 200:
        return response.json()
    else:
        raise Exception(f"Error fetching history: {response.status_code}")
```

Рисунок 3.11 – Код для отримання інформації з API

Функція `fetch_history()` реалізує механізм HTTP-запитів для отримання історичних даних з REST API.

Використання протоколу HTTP забезпечує модульність системи, дозволяючи інтегруватися з будь-якою серверною частиною, що відповідає заданому інтерфейсу. Обробка часових міток відбувається за допомогою функції `strftime()` з бібліотеки `datetime` (рис. 3.12).

```
def parse_time(time_str):
    try:
        return datetime.strptime(time_str, "%a, %d %b %Y %H:%M:%S GMT")
    except ValueError:
        print(f"Error: {time_str} does not match expected format")
        return None
```

Рисунок 3.12 – Код для обробки часових міток

Аналітичний модуль використовує наступний набір словників для агрегації даних:

- user_time_in_office (накопичення загального часу перебування);
- user_daily_time (деталізація за днями);
- user_lateness_count (моніторинг порушень трудового розпорядку).

Графічні елементи системи охоплюють динамічне масштабування вікон (figsize), використання кольорової семантики, де помаранчевий відтінок позначає запізнення, а також текстові підписи із значеннями показників та автоматичне форматування підписів із кутом повороту 45 градусів (рис. 3.13).

```
plt.figure(figsize=(10, 6))
bars = plt.bar(lateness_users, lateness_counts, color='orange')

for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(int(yval)),
             ha='center', va='bottom', fontsize=12)
```

Рисунок 3.13 – Код для форматування графіків

Система формує каскадні візуалізації для кожного користувача, що включають порівняння фактичного та планового робочого часу, побудову гістограми частотності виходів із офісу та аналіз хронології порушень трудової дисципліни.

Метод `tight_layout()` забезпечує оптимальне розміщення елементів інтерфейсу, а використання альфа-каналу для планових годин (`alpha=0.5`) дозволяє ефективно накладення даних.

Обробка помилок реалізована через глобальний try-ехсепт блок, що гарантує стабільність роботи навіть при відсутності з'єднання з сервером або некоректному форматі відповіді API.

ВИСНОВКИ

У ході виконання даної кваліфікаційної роботи була проведена розробка та реалізація електронної системи ідентифікації відвідувачів з використанням RFID-технології. Метою роботи було створення комплексного рішення, що забезпечує автоматизований контроль доступу, реальний час обробки подій та інтеграцію з аналітичними інструментами і месенджерами для оперативного інформування.

В ході дослідження було розглянуто сучасний стан та актуальність систем автоматичної ідентифікації, проаналізовано переваги RFID-технології для управління потоками відвідувачів та збору статистики відвідуваності. Було обґрунтовано доцільність інтеграції RFID-систем із хмарними сервісами та Telegram API для підвищення безпеки та ефективності контролю доступу.

У процесі реалізації було створено апаратний модуль на основі ESP8266 та RFID-зчитувача MFRC522, попередньо було досліджено розпіновки цих компонентів, а також змодельовано електричну схему.

Була розроблена серверна частина з REST API на Flask для обробки та зберігання даних у PostgreSQL, яка також відповідальна за механізм автоматичних сповіщень через Telegram API.

Наступним кроком, було створено аналітичний модуль для візуалізації даних про відвідуваність за допомогою бібліотеки matplotlib. Дані графіки можуть бути використані для розрахунку ключових показників KPI та побудові професійних KPI-панелей.

Отже, можна зробити висновок, що розроблена система є ефективним інструментом для автоматизації контролю доступу та аналізу відвідуваності. Результати роботи довели, що інтеграція RFID-технології з IoT-пристроями та аналітичними модулями дозволяє створити сучасну, масштабовану та безпечну платформу, яка може бути використана в офісних центрах, на виробничих майданчиках чи інших об'єктах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пришневський О. О. Електронна система ідентифікації відвідувачів на основі технології RFID: збірник наукових праць з матеріалами Міжнародної наукової конференції «Інноваційна наука: пошук відповідей на виклики сучасності», м. Кривий Ріг, 21 листопада 2025 р. Кривий Ріг, 2025. С. 311-317.
2. Domínguez-Bolaño T., Capros O., Barral V., Escudero C, García-Naya J. An overview of IoT architectures, technologies, and existing open-source projects. *Internet of Things*. 2022. Vol. 20, No. 1. P. 6-26.
3. What is load shedding? URL: <https://www.techtarget.com/searchdatacenter/definition/load-shedding> (дата звернення: 19.06.2025).
4. Popescu V., Scarlat C. Supervisory control and data acquisition (SCADA) systems for industrial automation and control systems in industry 4.0. *Land Forces Academy Review*. 2022. Vol. 27, No. 4. P. 309-315.
5. Amazon Echo vs Echo Dot vs Google Home vs Home Mini: Which smart speaker is best for you? URL: <https://www.gearbrain.com/amazon-echo-vs-google-home-2500774801.html> (дата звернення: 20.06.2025).
6. Electronic Systems. URL: <https://www.electronicstutorials.ws/systems/electronic-system.html> (дата звернення: 21.06.2025).
7. Automatic Identification System Equipment. URL: <https://www.identecolutions.com/news/automatic-identification-system-equipment> (дата звернення: 28.06.2025).
8. Іщук Д., Нестеровський А., Костючко С., Кайдик О., Терлецький Т. Система безпеки з використанням RFID та мобільних пристроїв. *Програмне та апаратне забезпечення в інформаційних технологіях*: матеріали Міжнародної науково-практичної конференції молодих вчених та студентів, м. Луцьк, 6 травня 2025 р. Луцьк: ЛНТУ, 2025. С. 72.
9. What is RFID? URL: <https://www.bradyid.com/intelligent-manufacturing/what-is-rfid> (дата звернення: 30.06.2025).

10. RFID-модуль RC522 + карта + брелок. URL: <https://distancionka.com/ua/catalog/arduino/moduli/rfid-modul-rc522-karta-brelok/?srsltid=AfmBOopZVTBhoRwCsXAHF3KXnr5wxU0n0rkLVDIRiES2tT5UMI3kOtIW> (дата звернення: 02.07.2025).

11. Honeywell Granit™ XP 1991i SR Cordless Industrial-Grade Handheld Scanner. URL: <https://www.atlasrfidstore.com/honeywell-granit-xp-1991i-sr-cordless-industrial-grade-handheld-scanner/?srsltid=AfmBOopZSAPIUKpVVgj1iVs36sBOE1Gy2pWKx4cMZ3YWIJpUpiMRWwSw> (дата звернення: 02.07.2025).

12. What is a Real-Time Operating System (RTOS)? URL: <https://www.digikey.com/en/maker/projects/what-is-a-realtime-operating-system-rtos/28d8087f53844decafa5000d89608016> (дата звернення: 08.07.2025).

13. PlatformIO. URL: <https://platformio.org/> (дата звернення: 08.07.2025).

14. Model of the system built in Proteus VSM. URL: https://www.researchgate.net/figure/Model-of-the-system-built-in-Proteus-VSM_fig5_353400393 (дата звернення: 10.07.2025).

15. Your MicroPython Board Can Be Your Tinkering Peripheral. URL: <https://hackaday.com/2022/08/10/your-micropython-board-can-be-your-tinkering-peripheral/> (дата звернення: 12.07.2025).

16. Why Altium Designer Can Help Design PCBs Intuitively and Intelligently URL: <https://resources.altium.com/p/why-altium-designer-can-help-design-pcbs-intuitively-and-intelligently> (дата звернення: 14.07.2025).

17. Simulink. Wikipedia. URL: <https://uk.wikipedia.org/wiki/Simulink> (дата звернення: 15.07.2025).

18. Matplotlib: Part 9 – Integrating Matplotlib with Other Libraries. URL: <https://medium.com/@ebimsv/mastering-matplotlib-part-9-integrating-matplotlib-with-other-libraries-ba93071ec8bf> (дата звернення: 17.07.2025).

19. What are RFID Tags? How do RFID Tags Work?. URL: <https://www.camcode.com/blog/what-are-rfid-tags/?srsltid=AfmBOor2qoBD5dqSKRV-6njXy1s4DBpJAх-AXzmBt-ZhqSnm1Rq9Fqe> (дата звернення: 25.07.2025).

20. Чамунорва Т. Програмно-технічний засіб керування дверним замком на основі мікроконтролера ESP8266 : кваліфікаційна робота бакалавра: 123 Комп'ютерна інженерія. Хмельницький : ХНУ, 2024. 153 с.

21. NodeMCU v3 high resolution pinout and specs. URL: <https://mischianti.org/nodemcu-v3-high-resolution-pinout-and-specs/> (дата звернення: 25.07.2025).

22. RC522 RFID Reader Module. URL: <https://microcontrollerslab.com/rc522-rfid-reader-pinout-arduino-interfacing-examples-features/> (дата звернення: 29.07.2025).

23. Nunes F., Alexandre E., Gaspar P. Implementing Key Performance Indicators and Designing Dashboard Solutions in an Automotive Components Company: A Case Study. *Administrative Science*. 2024. Vol. 14, No. 8. P. 175-189.