МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп’ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувачка кафедри,
д-р техн. наук, проф.

_____Ірина ЖУРАВСЬКА

« __ » _____ 202__ р.

КВАЛІФІКАЦІЙНА РОБОТА

НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

IoT–пристрій для контролю вмісту вуглекислого газу у приміщенні

Спеціальність 123 Комп’ютерна інженерія

Освітня програма «Комп’ютерна інженерія»

Здобувач _____Павло ВОЛОЧАЙ
підпис

«__» _____ 202__ р.

Керівник ст. викладач _____ Борис САЛТОВСЬКИЙ
підпис
«__» _____ 202__ р.

Миколаїв – 2026

Факультет	Комп’ютерних наук
Кафедра	Комп’ютерної інженерії
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступень	Бакалавр
Спеціальність	123 Комп’ютерна інженерія
Освітня програма	Комп’ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри комп’ютерної інженерії
_____ Ірина ЖУРАВСЬКА
«_____» _____ 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу здобувача

Волочай Павло Олександрович
(прізвище, ім’я, по батькові здобувача)

1. Тема кваліфікаційної роботи
IoT-пристрій для контролю вмісту вуглекислого газу у приміщенні

Затверджена наказом по ЧНУ ім. Петра Могили від 25.11.2025 № 294.

2. Строк представлення кваліфікаційної роботи «____» _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні
Очікуваним результатом кваліфікаційної роботи є прототип IoT-пристрою для контролю вмісту вуглекислого газу у приміщенні

4. Перелік питань, що підлягають розробці:
1) проаналізувати вплив концентрації CO2 на здоров'я людини та дослідити сучасні методи й нормативні вимоги до моніторингу якості повітряного середовища;
2) вивчити існуючі підходи до побудови IoT-систем екологічного контролю та провести порівняльний аналіз мережевих протоколів для обміну телеметричними даними;
3) здійснити обґрунтований підбір апаратних і програмних компонентів, зокрема мікроконтролера серії ESP32, NDIR-датчиків типу MH-Z19E та середовища розробки;

4) розробити архітектуру апаратної частини ПАП для збору інформації про стан середовища та спроєктувати схеми узгодження логічних рівнів компонентів;
5) реалізувати програмне забезпечення для обробки сенсорних даних із застосуванням алгоритмів цифрової фільтрації та забезпечення асинхронної передачі пакетів по протоколу MQTT;
6) провести експериментальну перевірку працездатності розробленої системи та оцінити ефективність її інтеграції в екосистему «Розумного будинку»

5. Перелік графічних матеріалів

Презентація

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Керівник роботи _____ Борис САЛТОВСЬКИЙ
Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

Здобувач _____ Павло ВОЛОЧАЙ
Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

Дата видачі завдання «_____»_____ 2025 р.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної бакалаврської роботи

Тема: IoT-пристрій для контролю вмісту вуглекислого газу у приміщенні

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КБР	04.12.2025	14.12.2025	Виконано
2	Огляд літератури за темою роботи	15.12.2025	31.12.2025	Виконано
3	Складання календарного плану КБР	01.01.2026	10.01.2026	Виконано
4	Аналіз предметної області	11.01.2026	27.01.2026	Виконано
5	Розробка проєктних рішень	28.01.2026	15.02.2026	Виконано
6	Моделювання та конструювання АПЗ	16.02.2026	28.02.2026	Виконано
7	Перевірка працездатності, тестування та апробація розробленого АПЗ, аналіз результатів тестування	01.03.2026	02.03.2026	Виконано
8	Оформлення КБР та презентації	20.05.2026	21.05.2026	Виконано
9	Перший передній захист КБР	22.05.2026	22.05.2026	Виконано
10	Другий передній захист КБР	05.06.2026	05.06.2026	Виконано
11	Завершення оформлення КБР та презентації	06.06.2026	08.06.2026	Виконано
12	Перевірка на академічний плагіат, фальсифікацію та списування	09.06.2026	09.06.2026	Виконано
13	Відгук керівника КБР	10.06.2026	11.06.2026	Виконано
14	Подання КБР рецензенту та рецензування КБР	12.06.2026	14.06.2026	Виконано
15	Подання КБР, її електронної копії та інших документів (відгуку, рецензії)	15.06.2026	17.06.2026	Виконано
16	Захист кваліфікаційної бакалаврської роботи	24.06.2026	24.06.2026	Виконано

Керівник роботи _____ Борис САЛТОВСЬКИЙ

Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

Здобувач _____ Павло ВОЛОЧАЙ

Особистий підпис *Власне ім'я ПРІЗВИЩЕ*

«____» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи
«IoT–пристрій для контролю вмісту вуглекислого газу у приміщенні»
Здобувач: Волочай Павло Олександрович
Керівник: ст. викладач Салтовський Борис Григорович

Кваліфікаційна бакалаврська робота присвячена дослідженню методів і засобів моніторингу якості повітря в закритих приміщеннях, використанню технологій Інтернету речей (IoT) як інструменту здійснення безперервного контролю концентрації вуглекислого газу (CO_2), а також розробці програмно-апаратної платформи для збору, обробки та візуалізації даних про стан повітряного середовища у приміщеннях різного призначення, що включають житлові будинки, офісні центри, навчальні заклади тощо. Актуальність бакалаврської роботи обумовлена необхідністю підтримки оптимального мікроклімату для збереження здоров'я та підвищення працездатності людей шляхом впровадження доступних, компактних та автономних систем моніторингу.

Метою роботи є розробка програмно-апаратної платформи для безперервного моніторингу та візуалізації вмісту вуглекислого газу в закритих приміщеннях з інтеграцією до екосистеми розумного будинку. Об'єктом дослідження є процес моніторингу концентрації вуглекислого газу в повітрі закритих приміщень на основі сенсорних даних та технологій Інтернету речей. Предметом дослідження є методи і засоби програмно-апаратної реалізації системи контролю вмісту CO_2 та візуального оповіщення про стан якості повітря. Практичне значення результатів дослідження полягає у можливості впровадження доступної та енергоефективної системи моніторингу мікроклімату без залучення дорогого професійного обладнання. Отримані результати можуть бути використані для автоматизації контролю середовища у житлових будинках, офісних центрах, навчальних аудиторіях та інших соціально значущих об'єктах.

Кваліфікаційна робота пройшла апробацію під час науково-практичної конференції «Інтелектуальні Інформаційні Системи» (05 грудня 2025 р.); має відповідні публікації матеріалів тез.

Пояснювальна записка кваліфікаційної бакалаврської роботи складається зі вступу, трьох розділів, висновків, переліку використаних джерел та трьох додатків. У вступі обґрунтовано актуальність обраної теми, визначено об'єкт і предмет дослідження, сформульовано мету, завдання та описано практичне значення отриманих результатів.

У першому розділі проведено аналіз методів моніторингу якості повітря та санітарно-гігієнічних норм вмісту CO_2 у приміщеннях. Досліджено технології IoT та протоколи мережевої взаємодії, зокрема MQTT, а також здійснено

обґрунтований вибір апаратної бази: оптичного сенсора MH-Z19E та мікроконтролера ESP32. У другому розділі представлено опис концепції програмно-апаратної платформи, наведено розроблену схему узгодження логічних рівнів та етапи створення програмного забезпечення в середовищі Arduino IDE. Описано програмну реалізацію об’єктно-орієнтованих класів для опитування датчиків, алгоритмів фільтрації даних та логіки світлової індикації стану повітряного середовища. У третьому розділі описано практичну реалізацію апаратного вузла, налаштування захищеного зв'язку (TLS) з хмарним MQTT-брокером та розробку панелі візуалізації Node-RED. Проведено інтеграційне тестування системи та налаштовано безпечний віддалений доступ через Mesh VPN (Tailscale).

У висновках наведено загальний аналіз виконаної роботи та підтверджено виконання всіх поставлених задач. У додатку А наведено лістинг програмного коду для керування мікроконтролером ESP32, збору телеметричних даних та їх асинхронної передачі по протоколу MQTT. У додатку Б представлено матеріали апробації результатів кваліфікаційної роботи на науково-практичній конференції.

У цілому, кваліфікаційна бакалаврська робота, містить 71 с. (без додатків), 34 рис., 23 табл., 21 джерел посилання та 2 додатки.

Ключові слова: *інтернет речей (IoT), моніторинг концентрації CO_2, якість повітря, протокол MQTT, Розумний будинок, мікроконтролер ESP32, NDIR-датчики, обробка даних, апаратно-програмний комплекс*.

ABSTRACT
of the Bachelor's Thesis
"IoT device for monitoring indoor carbon dioxide concentration"
Applicant: Pavlo Volochai
Supervisor: Senior Lecturer Boris Saltovskiy

The bachelor's thesis is dedicated to the research of methods and tools for air quality monitoring in closed spaces, the use of Internet of Things (IoT) technologies as a tool for continuous control of carbon dioxide (CO_2) concentration, as well as the development of a software and hardware platform for the collection, processing, and visualization of data on the state of the air environment in premises of various purposes, including residential buildings, office centers, educational institutions, etc. The relevance of the bachelor's thesis is due to the need to support an optimal microclimate for health preservation and increasing human work capacity by implementing accessible, compact, and autonomous monitoring systems.

The goal of the work is to develop a hardware and software platform for continuous monitoring and visualization of carbon dioxide content in indoor spaces with integration into smart home systems. The object of study is the process of monitoring carbon dioxide concentration in the air of closed premises based on sensor data and Internet of Things technologies. The subject of study is the methods and means of software and hardware implementation of the CO_2 content control system and visual notification of the state of air quality. The practical value of the research results lies in the possibility of implementing an accessible and energy-efficient microclimate monitoring system without involving expensive professional equipment. The obtained results can be used for environment control automation in residential buildings, office centers, lecture halls, and other socially significant objects.

The bachelor's thesis underwent approbation during the scientific and practical conference "Intelligent Information Systems" (December 05, 2025); it has corresponding publications of abstract materials.

The explanatory note of the qualification bachelor's thesis consists of an introduction, three chapters, conclusions, a list of used sources, and three appendices. The introduction defines the relevance of the chosen topic, identifies the object and subject of study, formulates the goal and tasks, and describes the practical value of the obtained results.

In the first chapter, an analysis of air quality monitoring methods and sanitary and hygienic norms of CO_2 content in premises was carried out. Internet of Things (IoT) technologies and network interaction protocols, in particular MQTT, were studied, and a justified choice of hardware base was made: the MH-Z19E optical sensor and the ESP32 microcontroller. In the second chapter, the concept of the hardware and software platform is presented, the developed logic level shift circuit and the stages of software creation in

the Arduino IDE environment are provided. The software implementation of object–oriented classes for sensor polling, data filtering algorithms, and visual indication logic of the air environment state is described. In the third chapter, the practical implementation of the hardware node, the configuration of secure communication (TLS) with the cloud MQTT broker, and the development of the Node-RED visualization panel are described. Integration testing of the system was conducted, and secure remote access via Mesh VPN (Tailscale) was configured.

The conclusions provide a general analysis of the work performed and confirm the fulfillment of all the tasks set. Appendix A contains the listing of the program code for controlling the ESP32 microcontroller, collecting telemetry data, and its asynchronous transmission via the MQTT protocol. Appendix B presents materials of the qualification work results approbation at the scientific and practical conference.

In general, the bachelor's thesis, without appendices, contains 71 pages (excluding appendices), 34 figures, 23 tables, 21 references and 2 appendices.

Keywords: *internet of things (IoT), CO_2 concentration monitoring, air quality, MQTT protocol, Smart Home, ESP32 microcontroller, NDIR-sensors, data processing, hardware–software complex.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

АПЗ – апаратно-програмні засоби

КБР – кваліфікаційна бакалаврська робота

МК – мікроконтролер

ПАП – програмно-апаратна платформа

ПЗ – програмне забезпечення

РБ – розумний будинок

EMA – Exponential Moving Average

IDE – Integrated Development Environment

IoT – Internet of Things

JSON – JavaScript Object Notation

MQTT – Message Queuing Telemetry Transport

NDIR – Non–Dispersive InfraRed

SoC – System on a Chip

TCP/IP – Transmission Control Protocol / Internet Protocol

UART – Universal Asynchronous Receiver–Transmitter

USB – Universal Serial Bus

ВСТУП

Сучасне Internet of things (IoT) рішення та алгоритми аналізу даних надають можливість у режимі реального часу відслідковувати концентрацію CO_2 у приміщеннях. Завдяки автономності приладів та можливості застосовувати NDIR-сенсори, захоплення вуличного повітря може бути здійснене без участі коштовного професійного устаткування, що дозволяє отримувати актуальні дані про стан повітря швидко. Також це стосується навчальних закладів, офісів, житлових будинків, де своєчасне реагування на перевищення концентрації вуглекислого газу є запорукою збереження здоров'я та працездатності людей.

Причини актуальності дослідження з програмно-апаратної платформи (ПАП) для моніторингу CO_2 – різноманітні:

1) високозамкнуті сучасні будівлі енергоефективного типу вимагають періодичного спостереження за складом повітря, яке було б легше проводити із застосуванням традиційних методів, враховуючи складність та вартість останніх;

2) платформа, побудована на МК, датчиках і мережевих протоколах, для збору новітніх даних, може значно збільшити ефективність системи оповіщення попередження негативного впливу гіперкапні;

3) розробка ПАП дозволить ефективно інтегрувати апарат у систему РБ з метою швидкого діагностування стану середовища та автоматичного управління вентиляцією.

В умовах сучасного енергоефективного будівництва гостро стоїть проблема контролю за санітарно-гігієнічними нормами повітряного середовища в приміщеннях потрібна надійна IoT система моніторингу, що своєчасно реєструє зміни концентрації CO_2, сприяє зниженню негативного впливу на здоров'я та працездатність людини та ефективному використанню кліматичного обладнання.

Метою роботи є розробка ПАП для безперервного моніторингу та візуалізації вмісту вуглекислого газу із застосуванням NDIR-датчиків, МК як обчислювального ядра для збору даних та ПЗ для обробки, передачі телеметрії

через протокол MQTT і візуалізації показників якості повітря. Для досягнення цієї мети необхідно вирішити наступні ***задачі***:

– проаналізувати вплив концентрації CO_2 на здоров'я людини та дослідити сучасні методи й нормативні вимоги до моніторингу якості повітряного середовища;

– вивчити існуючі підходи до побудови IoT-систем екологічного контролю та провести порівняльний аналіз мережевих протоколів для обміну телеметричними даними;

– здійснити обґрунтований підбір апаратних і програмних компонентів, зокрема мікроконтролера серії ESP32, NDIR-датчиків типу MH-Z19E та середовища розробки;

– розробити архітектуру апаратної частини ПАП для збору інформації про стан середовища та спроєктувати схеми узгодження логічних рівнів компонентів;

– реалізувати програмне забезпечення для обробки сенсорних даних із застосуванням алгоритмів цифрової фільтрації та забезпечення асинхронної передачі пакетів по протоколу MQTT;

– провести експериментальну перевірку працездатності розробленої системи та оцінити ефективність її інтеграції в екосистему «Розумного будинку».

Об'єктом дослідження є процес безперервного контролю концентрації вуглекислого газу в повітряному середовищі закритих приміщень різного функціонального призначення – житлових, офісних та навчальних – шляхом застосування сенсорних вузлів і технологій Інтернету речей для отримання актуальних вимірювальних даних у режимі реального часу.

Предметом дослідження є методи і засоби програмно-апаратної реалізації системи вимірювання вмісту CO_2, локального візуального оповіщення та мережевої взаємодії на базі протоколу MQTT з інтеграцією до платформи Node-RED.

Практичне значення результатів дослідження полягає у можливості розгортання енергоефективної та доступної за вартістю системи мікрокліматичного моніторингу без залучення дорогого спеціалізованого обладнання. Запропонований підхід дозволяє в автоматизованому режимі

фіксувати зміни газового складу повітря та формувати візуальні сповіщення для оперативного реагування персоналу або мешканців. Завдяки відкритій архітектурі на базі мікроконтролера ESP32 і локального брокера повідомлень система не залежить від сторонніх хмарних сервісів, що забезпечує її стабільну роботу навіть за відсутності зовнішнього інтернет-з'єднання. Отримані результати придатні для практичного впровадження в житлових будинках, офісних центрах, навчальних аудиторіях, а також для інтеграції в розгалужені екосистеми «Розумного будинку» з метою автоматичного керування вентиляційним обладнанням.

Апробація результатів кваліфікаційної роботи, що присвячена розробці автономного модуля моніторингу концентрації CO_2 та впровадженню трирівневої архітектури системи, відбулася під час науково-практичної конференції «Інтелектуальні Інформаційні Системи» (05 грудня 2025 р.). Матеріали дослідження, що включають опис архітектури IoT-пристрою та алгоритмів асинхронної передачі даних по протоколу MQTT, опубліковані у відповідному збірнику тез доповідей [13].

1 МЕТОДИ ТА ЗАСОБИ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ ТА КОНЦЕНТРАЦІЇ ВУГЛЕКИСЛОГО ГАЗУ

1.1 Аналіз існуючих засобів моніторингу якості повітря та огляд існуючих рішень

На сучасному ринку систем екологічного моніторингу представлено широкий спектр рішень – від побутових індикаторних пристроїв початкового рівня до промислових аналізаторів якості повітря [10], переважна більшість із них базується на інфрачервоній сенсорній технології, що дозволяє одночасно відстежувати концентрацію діоксиду вуглецю, відносну вологість та вміст летких органічних сполук. Використання CO_2 як ключового індикатора антропогенного забруднення повітря є загальновизнаним стандартом у світовій науковій та інженерній практиці [8], і саме ця методологічна основа визначила вектор розвитку більшості комерційних розробок у даній галузі.

Концепцію «здорового офісного середовища» активно просувають такі провідні виробники, як Netatmo, AirVisual (рис.1.1) та Awair, однак їхні рішення, попри високу метрологічну точність, здебільшого реалізовані у вигляді закритих екосистем: дані зберігаються виключно на серверах виробника, а локальна інтеграція з іншими системами або суттєво обмежена чи повністю виключена, що принципово звужує можливості застосування таких пристроїв у складі автономних «розумних» систем [11]. Проведений аналіз підтверджує стійкий і незадоволений запит на інструменти безперервного моніторингу мікроклімату з відкритою архітектурою та можливістю гнучкого налаштування.

Рисунок 1.1 – Загальний вигляд пристрою AirVisual [22]

Комерційні аналоги систем моніторингу охоплюють широкий спектр рішень: автономні монітори (Netatmo, Aranet4), екосистемні вузли (Xiaomi) та промислові OEM-модулі (SenseAir Sunrise). Незважаючи на використання якісної NDIR-технології, пристрої Netatmo та Xiaomi мають пропрієтарне програмне забезпечення та жорстку прив’язку до хмарних сервісів виробника, що унеможливлює їхню гнучку локальну інтеграцію. Aranet4 позбавлений хмарної залежності, однак передає дані виключно через Bluetooth і не підтримує MQTT-протокол. SenseAir Sunrise є точним промисловим модулем, проте орієнтований на OEM-інтеграцію і потребує розробки власної керуючої електроніки. Виявлені обмеження стали вирішальними факторами на користь вибору відкритої DIY-платформи на базі мікроконтролера ESP32. В табл. 1.1 наведено порівняльну характеристику засобів моніторгу, їх переваги та недоліки:

Таблиця 1.1 – Порівняльна характеристика засобів моніторингу CO_2

Характеристика	**Netatmo Healthy Home Coach**	**Xiaomi Mi Air Detector**	**Aranet4 Home**	**SenseAir Sunrise**	**Розроблена ПАП (ESP32 + MH-Z19E)**
Технологія вимірювання	NDIR (інфрачервона)	NDIR / Електро-хімічна	NDIR (дво-канальна)	NDIR (двох-променева)	NDIR (високо-селективна)
Тип архітектури	Закрита (пропрієтарна)	Екосистемна (Mi Home)	Закрита (Bluetooth)	Модульна (OEM)	Відкрита (Open-source)
Передача даних	Wi-Fi (Cloud only)	Wi-Fi / Bluetooth	Bluetooth LE 5.0	UART / I²C / Modbus	Wi-Fi (MQTT / HTTP)
Локальна інтеграція	Обмежена	Мінімальна	Часткова (Home Assistant)	Повна (промислова)	Повна (Home Assistant / Node-RED)
Залежність від хмари	Повна	Висока	Відсутня	Відсутня	Відсутня (локальний сервер)
Точність вимірювання CO_2	±50 ppm	±50 ppm	±30 ppm ±3%	±30 ppm ±3%	±50 ppm (MH-Z19E)
Вартість	Висока (~$99)	Середня (~$50)	Висока (~$99)	Середня (OEM-модуль)	Низька (доступні компоненти)

Дослідницький досвід у галузі вбудованих систем підтверджує, що ефективність моніторингу критично залежить від селективності обраного вимірювального перетворювача, багато авторів попередніх робіт використовували дешеві напівпровідникові сенсори, які, хоча й коштують менше, мають високу чутливість до вологості та сторонніх газів. Однак на відміну від таких рішень, використання сенсорів типу MH-Z19E [10] дозволяє вийти на рівень професійних вимірювань у побутовому сегменті.

Наукові публікації останніх років наголошують на важливості впровадження протоколу MQTT [1] для забезпечення швидкого реагування систем вентиляції на зміну телеметрії. Раніше розроблені платформи часто покладалися на передачу даних через HTTP, що створювало надмірне навантаження на бездротові канали зв'язку. Однак сучасні тенденції в розробці ПАП вказують на необхідність використання формату JSON [4] для уніфікації обміну даними між різнорідними пристроями.

Отже вивчення досвіду інтеграції подібних вузлів у системи «Smart Home» дозволило виділити оптимальну конфігурацію обчислювального ядра і таким чином, поточне дослідження базується на перевірених часом технологічних стеках, адаптованих під нові виклики енергоефективності будівництва.

1.2 Нормативні акти щодо контролю вмісту виглекислого газу

В Україні якість повітря в приміщеннях регулюється кількома нормативними документами – від будівельних норм до санітарних вимог МОЗ. Усі вони так чи інакше спираються на концентрацію CO_2 як основний вимірюваний параметр: цей газ накопичується пропорційно до кількості людей і інтенсивності їхньої діяльності, тому добре відображає реальний стан вентиляції. Саме через це CO_2 став універсальним індикатором якості повітря ще з часів досліджень Макса фон Петтенкофера у XIX столітті, і ця традиція збереглася в сучасних технічних стандартах.

Показник CO_2 зручний тим, що вимірюється безпосередньо і не вимагає складної інтерпретації: якщо концентрація перевищує поріг – вентиляція не

справляється. Саме тому автоматизовані системи моніторингу будують навколо цього датчика, а не навколо суб'єктивних відчуттів або непрямих показників. На відміну від вимірювання летких органічних сполук чи твердих частинок, результат зчитування CO_2-сенсора одразу придатний для прийняття рішення без додаткової обробки – це значно спрощує логіку керуючого програмного забезпечення.

Найбільш докладні вимоги щодо допустимих рівнів вуглекислого газу зафіксовані в державних будівельних нормах, які регламентують процеси проєктування систем опалення та вентиляції. Сучасна методологія оцінки якості внутрішнього повітря враховує низку факторів, що визначають комфортність перебування мешканців у приміщеннях. Одним із найважливіших параметрів при цьому вважається ступінь перевищення внутрішньої концентрації газу над його природним фоновим рівнем у зовнішньому повітрі.

Згідно з чинними технічними рекомендаціями, для забезпечення високого стандарту чистоти повітря ця різниця не повинна перевищувати 350–400 ppm. У підсумку такий розрахунок орієнтує інженерів на встановлення цільового порогу загальної концентрації в приміщенні на рівні близько 800 ppm. Дотримання зазначених норм є запорукою запобігання симптомам гіперкапнії та втоми серед персоналу чи мешканців. Варто також зазначити, що перевищення рівня 1 000–1 200 ppm вже фіксується людиною суб'єктивно – з'являється відчуття задухи, знижується концентрація уваги, що підтверджується рядом незалежних клінічних досліджень.

Практична реалізація систем моніторингу в Україні спирається на перелік фундаментальних законодавчих та нормативних актів, в яких детально описують процедури контролю вмісту газів та встановлюють відповідальність за недотримання норм мікроклімату. Нижче наведено ключові регламенти, що формують нормативне поле для впровадження систем екологічного контролю:

– ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування»: ключовий документ для інженерів, який встановлює вимоги до повітрообміну. Він визначає необхідні об'єми припливу свіжого повітря на основі розрахунків

допустимої концентрації CO_2 для житлових, офісних та громадських приміщень [6];

– ДСТУ EN 16798-1:2019 (замінив ДСТУ EN 15251): державний стандарт, гармонізований з європейськими нормами. Він класифікує якість повітря за категоріями (I–IV) та встановлює конкретні межі вмісту вуглекислого газу для кожної з них (наприклад, для дитячих установ рекомендовано орієнтуватися на найвищу категорію) [7];

– ДБН В.2.2-15:2019 «Житлові будинки. Основні положення»: містить вимоги до організації вентиляції в житловому фонді, забезпечуючи підтримання здорового мікроклімату;

– наказ МОЗ України №25 від 24.07.1996 р.: регламентує гранично допустимі концентрації (ГДК) у повітрі робочої зони. Хоча ГДК для виробничих цехів є значно вищою, для офісів та житла лікарі-гігієністи традиційно орієнтуються на так званий «критерій Петтенкофера» – 0,1 % (1 000 ppm) як верхню межу комфорту;

– ДСТУ EN 13779:2011 «Вентиляція громадських будівель»: стандарт, що визначає класи якості внутрішнього повітря (IDA 1 - IDA 4). Згідно з ним, рівень понад 1 000 ppm вважається показником низької якості повітря.

У зв’язку з воєнним станом актуальним стає питання якості повітря в укриттях цивільного захисту, де значне скупчення людей у замкненому просторі може створювати ризики для здоров’я. Згідно з новим ДБН В.2.2-5:2023, моніторинг концентрації CO_2 є вагомим параметром безпеки [8], оскільки його рівень в укриттях не має перевищувати 1 % (10 000 ppm) і запровадження автономних засобів вимірювання дає змогу безпосередньо зафіксуватисягнення цього порогового значення та ініціювати переводення систем вентиляції в режим чистоповітряної фільтровентиляції (це необхідно для забезпечення безпеки експлуатації споруд у надзвичайних ситуаціях).

Загальний тренд удосконалення регуляторного поля після 2022 р., зокрема впровадження ДБН В.2.5-67:2025 [6], свідчить про те, що Україна рухається в бік більш жорстких європейських вимог із автоматизування інженерних систем,

оновлені санітарні норми для освітніх та медичних закладів підкреслюють також роль постійного моніторингу показників середовища для забезпечення найкращого мікроклімату. В умовах відбудови інфраструктури післявоєнний час характерний тим, що ПАП з технологіями IoT стає сутнісним, адже замінює старі методики розрахункового повітряобміну на технології керування за потребою (Demand–Controlled Ventilation), з точки зору енергоефективності будівель.

1.3 Аналіз технологій IoT та мережевих протоколів для інтеграції в системи «Розумний будинок»

IoT полягає у створенні мережі фізичних об'єктів, які можуть взаємодіяти один з одним і з навколишнім середовищем без постійного втручання людини. У рамках РБ екосистеми ці технології дають змогу автоматизувати процеси будинку, підвищуючи комфорт, а також безпеку жителів в кілька разів. Варто відзначити, що подібні рішення вже давно вийшли за межі експериментальних розробок – сьогодні вони активно впроваджуються в житлових комплексах і приватних будинках по всьому світу. Головними перевагами застосування таких систем є підвищена енергоефективність будівель і можливість віддаленого управління інженерними системами, а саме цей безперервний обмін даними дає можливість перетворити індивідуальні датчики в цілісну інтелектуальну інфраструктуру моніторингу. До того ж, зібрана статистика використання дозволяє мешканцям краще розуміти реальне споживання ресурсів і приймати обґрунтовані рішення щодо оптимізації побутових витрат.

Для підключення до Інтернету у домашніх мережах найчастіше застосовується WiFi-стандарт, який базується на стеку протоколів TCP/IP. Сучасні SoC, такі як ESP32, надають можливість повної апаратної підтримки для бездротових мереж [11], без потреби використовувати додаткові модулі, і велика обчислювальна потужність таких мікроконтролерів дозволяє ефективно обробляти дані й підтримувати стабільне з'єднання з локальними або хмарними серверами.

На практиці це означає, що розробник може обійтися одним компактним чипом замість цілого набору окремих компонентів – це спрощує як схемотехніку,

так і подальше обслуговування пристрою. Такий підхід забезпечує оптимальний баланс між швидкістю передачі пакетів і споживанням енергії кінцевим пристроєм, а з огляду на вартість подібних рішень – вони лишаються доступними навіть для некомерційних та навчальних проєктів.

Специфічні легкі протоколи, такі як MQTT, відіграють важливу роль у мережевій взаємодії – цей протокол взаємодіє на моделі видавця-підписника, що спричиняє найменше навантаження на канали зв'язку та на апаратні засоби ПАП, завдяки низьким накладним витратам MQTT є досить придатним протоколом для асинхронної передачі телемоніторингу [1] з датчиків моніторингу CO_2 в реальному часі.

Порівняно з класичним HTTP, він не вимагає встановлення окремого з'єднання під кожен запит – брокер самостійно розподіляє повідомлення між підписниками, що особливо зручно при роботі з десятками пристроїв одночасно. Застосування центрального брокера дозволяє вигадувати більш гнучкі сценарії реакції на зміну показників якості повітря (табл. 1.2), а також спрощує додавання нових вузлів до системи без необхідності переналаштування вже існуючої інфраструктури.

Таблиця 1.2 – Порівняльний аналіз мережевих протоколів MQTT та HTTP

Характеристика	**MQTT**	**HTTP**
Архітектура	Видавець / Підписник	Клієнт / Сервер
Навантаження на канал	Мінімальне (двійковий формат)	Високе (текстовий формат)
Швидкість передачі	Висока (Real-time)	Низька (запити)
Обсяг заголовка	Від 2 байт	Від 100 байт
Енергоефективність	Висока (оптимально для батарей)	Середня

Для уніфікації обміну даними між різними компонентами системи найчастіше застосовується текстовий формат JSON. Він дозволяє компактно структурувати виміряні показники та статус пристрою, що значно спрощує їх подальшу інтерпретацію керуючими хабами [4], а читабельність такого формату робить налагодження системи значно зручнішим навіть без спеціалізованих

інструментів. Завдяки високій сумісності ці технології дозволяють легко інтегрувати розроблений пристрій у наявні платформи домашньої автоматизації: зокрема Home Assistant, Node-RED та подібні рішення підтримують JSON «з коробки». Кінцевим результатом є створення відкритої архітектури, яка здатна легко масштабуватися та адаптуватися до специфічних потреб користувача, не прив'язуючи його до конкретного виробника чи платформи.

1.4 Формування вимог до ПАП моніторингу мікроклімату

До вимірювального тракту висувається кілька вимог. По-перше, діапазон вимірювань – 400-5000 ppm, що перекриває як фоновий рівень вуличного повітря, так і критичні значення в погано провітрюваних приміщеннях – саме цей діапазон відповідає міжнародним нормам щодо якості повітря в житлових та офісних будівлях. По-друге, сенсор має мати апаратну термокомпенсацію – без неї похибка при зміні температури на 10°C може досягати 200 ppm, що робить покази практично непридатними для прийняття рішень про провітрювання. По-третє, схема узгодження логічних рівнів обов'язкова, оскільки MH-Z19E працює на 5В, а ESP32 – на 3,3В, і ігнорування цієї різниці може призвести до пошкодження мікроконтролера вже в перші години роботи.

Основним компонентом вимірювального тракту обрано інтелектуальний інфрачервоний сенсор MH-Z19E (рис. 1.2). Вибір обумовлений його високою селективністю до молекул CO_2 та наявністю заводського калібрування, що спрощує процес налаштування пристрою – на відміну від електрохімічних аналогів, цей сенсор не потребує регулярної рекалібровки в польових умовах. Додатковою перевагою є вбудований UART-інтерфейс, який дозволяє напряму з'єднати сенсор з мікроконтролером і зчитувати готові цифрові значення концентрації без необхідності реалізовувати складну аналогову обробку сигналу.

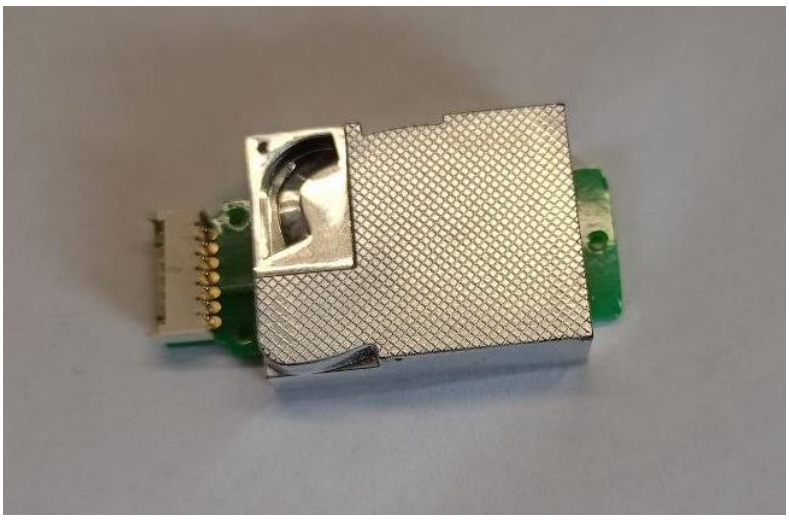

Рисунок 1.2 – Інфрачервоний датчик концентрації вуглекислого газу MH-Z19E

В табл. 1.3 наведено повні характеристики даного модулю.

Таблиця 1.3 – Основні технічні характеристики сенсора MH-Z19E

Параметр	**Значення**
Об’єкт вимірювання	Діоксид вуглецю (CO_2)
Технологія вимірювання	Недисперсійна інфрачервона (NDIR)
Діапазон вимірювання	400 – 5000 ppm
Робоча напруга	4.5 В – 5.5 В DC
Струм споживання (піковий)	< 150 мА
Вихідний сигнал	UART, PWM
Точність	± (50 ppm + 5% від значення)
Час виходу на стабільний режим	3 хв

Обчислювальний вузол системи має бути досить продуктивним, щоб одночасно виконувати завдання опитування датчиків і мережевих операцій. Підтримка бездротових засобів зв’язку (Wi-Fi) та стеку протоколів TCP/IP для передачі даних у реальному часі є обов’язковою і вибір мікроконтролера слід також робити, враховуючи потребу в нативній підтримці напруги 3,3 В для спрощення схемотехніки. Табл. 1.4 детально описує характеристики мікроконтролера.

Таблиця 1.4 – Характеристики обчислювального ядра на базі SoC ESP32 [19]

Характеристика	**Параметри модуля**
Процесор	Xtensa® Dual-Core 32-bit LX6
Робоча напруга	2,2 В – 3,6 В
Бездротовий зв’язок	Wi-Fi 802.11 b/g/n (до 150 Мбіт/с)
Мережеві протоколи	IPv4, IPv6, TCP/UDP, HTTP, MQTT
Периферійні інтерфейси	UART, SPI, I2C, ADC, DAC, PWM
Об’єм Flash-пам'яті	4 МБ (типово)
Струм споживання у режимі сну	< 10 мкА

Мережеві вимоги до ПАП передбачають застосування енергоефективних та стійких до розривів з'єднання протоколів, наприклад MQTT. Передавання телеметрії має бути асинхронним, щоб не створювати затримок у роботі основного алгоритму керування, а структура даних повинна бути універсальною (наприклад у форматі JSON) для забезпечення сумісності між різними платформами автоматизації. Додатково система має коректно обробляти короткочасні розриви мережі – при втраті з'єднання пристрій повинен буферизувати останні виміри локально і передавати їх брокеру одразу після відновлення зв'язку, що виключає прогалини в архіві телеметрії.

ПЗ платформи повинне містити алгоритми цифрової фільтрації випадкових коливань сигналу для згладжування індикаторів і логіка індикації має будуватися на чітких зонах («Норма», «Увага», «Небезпека», з реалізацією механізму гістерезису), це також допоможе запобігти нестабільності роботи індикаторів плавових меж концентрації. Ширина зони гістерезису підбирається експериментально з урахуванням типового шуму сенсора і має становити не менше 50 ppm, щоб світловий індикатор не перемикався хаотично при концентрації, близькій до порогового значення.

Для візуалізації та маршрутизації даних у реальному часі обрано платформу Node-RED, яка функціонує на локальному сервері й безпосередньо підписується на MQTT-топіки пристрою. Середовище Node-RED надає зручний графічний редактор потоків, що дозволяє без написання окремого коду налаштовувати

дашборд, сповіщення та сценарії реагування на зміну концентрації CO_2. Вбудований модуль node-red-dashboard забезпечує адаптивне відображення поточних показників, графіка динаміки та колірного індикатора стану на будь-якому пристрої в локальній мережі без встановлення додаткового програмного забезпечення (рис. 1.3).

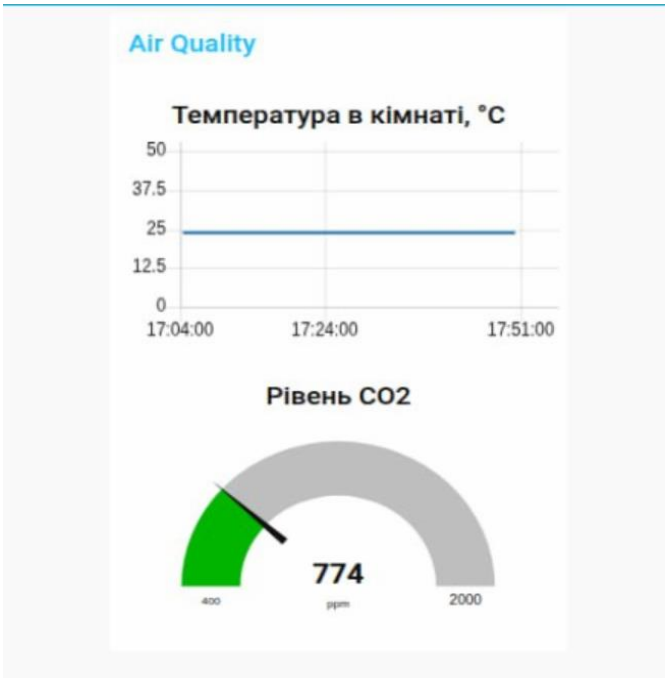

Рисунок 1.3 – Базовий вигляд інтерфейсу

Для забезпечення необхідної обчислювальної потужності та підтримки бездротових протоколів передачі даних обрано систему на кристалі ESP32 (рис. 1.4). Даний модуль має поєднувати в собі компактні габарити та широкі можливості підключення периферійних пристроїв. Двоядерний процесор Xtensa LX6 з тактовою частотою 240 МГц дозволяє паралельно виконувати опитування датчика, фільтрацію даних і підтримку мережевого стеку, не створюючи черг очікування між цими задачами – саме це робить ESP32 оптимальним вибором для подібних вбудованих систем реального часу.

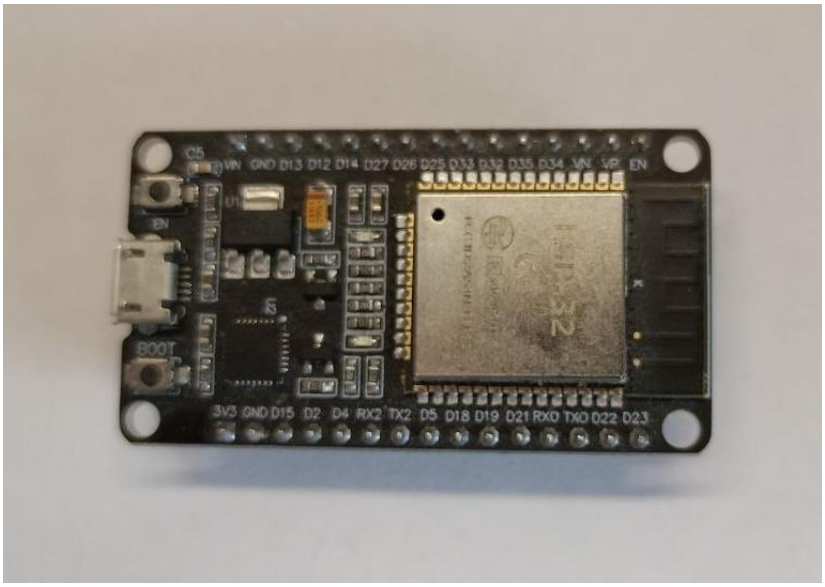

Рисунок 1.4 – Зовнішній вигляд мікроконтролера ESP32 DevKit v1

В табл.1.5 наведено базові вимоги до графічного інтерфейсу на базі Bootstrap 5:

Таблиця 1.5 – Вимоги до засобів візуалізації на базі Node-RED

Компонент / Вимога	**Опис реалізації**
Платформа візуалізації	Node-RED (локальний сервер)
Модуль дашборду	node-red-dashboard
Протокол отримання даних	MQTT (підписка на топіки ESP32)
Адаптивність	Вбудована адаптивна верстка дашборду
Залежності	Node.js + Node-RED (без сторонніх CDN)
Мета	Локальна незалежна візуалізація без хмари

Експлуатаційні вимоги до апаратної частини передбачають високу компактність приладу та можливість його універсального живлення від USB-портів, при цьому архітектура системи розрахована на тривалу автономну експлуатацію понад п'ять років без регулярного калібрування, що доповнюється функцією дистанційного зняття логів для проведення глибокої діагностики та онлайн-моніторингу працездатності програмно-апаратної платформи.

Окрім базових технічних параметрів, до системи висуваються й практичні вимоги, пов'язані з умовами реального використання. Пристрій має зберігати стабільну роботу при коливаннях напруги живлення, характерних для звичайних

мережевих адаптерів та павербанків, що особливо важливо в умовах нестабільного електропостачання. Інтерфейс індикації повинен бути інтуїтивно зрозумілим без будь-яких інструкцій – саме тому обрана триколірна світлова сигналізація, що не вимагає від користувача знання конкретних числових порогів. Вся конфігурація пристрою має зберігатися у енергонезалежній пам'яті, щоб після раптового відключення живлення система автоматично відновлювала роботу без втручання користувача.

Висновки до розділу 1

У першому розділі проведено комплексний аналіз методів і засобів моніторингу якості повітря, нормативної бази, технологій IoT та апаратних компонентів, що склали основу для обґрунтованого вибору архітектури розроблюваної платформи.

У підрозділі 1.1 здійснено огляд існуючих комерційних рішень моніторингу якості повітря – зокрема пристроїв Netatmo, AirVisual та Xiaomi. Встановлено, що попри високу метрологічну точність, більшість із них реалізовані як закриті екосистеми з обов'язковою прив'язкою до хмарних сервісів виробника та обмеженими можливостями локальної інтеграції. Саме ці недоліки стали вирішальним аргументом на користь розробки власної відкритої платформи на базі мікроконтролера ESP32, яка позбавлена зазначених обмежень і повністю адаптована до інтеграції в екосистему розумного будинку.

У підрозділі 1.2 досліджено нормативне поле України щодо контролю вмісту CO_2 у приміщеннях. Розглянуто ключові регламентуючі документи – ДБН В.2.5-67:2013, ДСТУ EN 16798-1:2019, ДБН В.2.2-15:2019 та наказ МОЗ №25, – які визначають допустимі рівні концентрації вуглекислого газу для різних типів приміщень. Встановлено, що цільовий поріг у 800 ppm є оптимальним орієнтиром для систем автоматичного керування вентиляцією, а перевищення рівня 1000 ppm негативно впливає на працездатність і самопочуття людей [21]. Окремо враховано вимоги оновленого ДБН В.2.2-5:2023 щодо моніторингу в укриттях цивільного захисту, що підкреслює соціальну актуальність розробки.

У підрозділі 1.3 проаналізовано технології IoT та мережеві протоколи для інтеграції в системи розумного будинку. Показано, що протокол MQTT з моделлю видавця-підписника має суттєві переваги перед HTTP – мінімальний розмір заголовка (від 2 байт), низьке навантаження на канал і підтримка роботи в реальному часі роблять його оптимальним вибором для передачі телеметрії з IoT-пристроїв. Обґрунтовано застосування формату JSON для уніфікації обміну даними між компонентами системи та підтверджено сумісність обраного стеку технологій з платформами Home Assistant і Node-RED.

У підрозділі 1.4 сформовано вимоги до програмно-апаратної платформи моніторингу мікроклімату та обґрунтовано вибір компонентної бази. Визначено, що діапазон вимірювань 400–5 000 ppm, наявність апаратної термокомпенсації та схема узгодження логічних рівнів є обов'язковими умовами коректної роботи системи. Як основний вимірювальний елемент обрано інфрачервоний NDIR-сенсор MH-Z19E завдяки його високій селективності до молекул CO_2, заводському калібруванню та вбудованому UART-інтерфейсу, що суттєво спрощує інтеграцію з мікроконтролером ESP32 без додаткових аналогових схем обробки сигналу.

2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОЇ ПЛАТФОРМИ

2.1 Архітектура системи та обґрунтування структурних рішень

Розроблена програмно-апаратна платформа побудована за трирівневою архітектурою, кожен рівень якої виконує чітко визначену функцію в загальному ланцюзі обробки інформації:

1) перший рівень – рівень збору даних, представлений апаратним вузлом на базі мікроконтролера ESP32, сенсора MH-Z19E та RGB-світлодіода NeoPixel, що розміщені на макетній платі та отримують живлення від стандартного USB-інтерфейсу;

2) другий рівень – рівень передачі, що був реалізований через локальний MQTT-брокер Mosquitto, що функціонує на Linux-машині в межах домашньої мережі та забезпечує маршрутизацію повідомлень між пристроєм і системою візуалізації;

3) третій рівень – рівень відображення представлений платформою Node-RED з модулем node-red-dashboard, яка підписується на відповідні топіки брокера і відображає телеметрію у вигляді графіків та індикаторів у реальному часі.

Вибір локальної архітектури без залучення хмарних сервісів зумовлений кількома принциповими міркуваннями. По-перше, система зберігає повну працездатність за відсутності зовнішнього інтернет-з'єднання, що є критично важливим для застосунків, орієнтованих на моніторинг безпеки середовища. По-друге, усі вимірювальні дані залишаються виключно в межах локальної мережі, що виключає ризики, пов'язані з передачею персональних даних стороннім операторам. По-третє, затримка між публікацією пакету ESP32 і його відображенням на дашборді в локальній мережі є мінімальною і не залежить від якості зовнішнього каналу зв'язку.

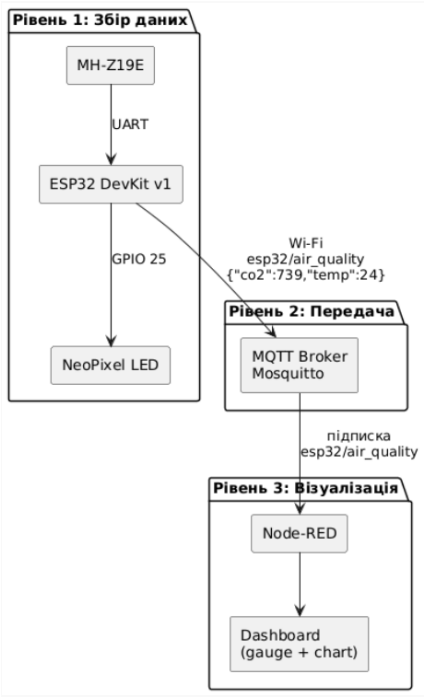

Рисунок 2.1 – Трирівнева архітектура системи

Для обміну даними між апаратним вузлом і системою візуалізації застосовано протокол MQTT з топіком esp32/air_quality. Кожні 3000 мс ESP32 публікує в цей топік JSON-пакет фіксованої структури, що містить два поля: поточну концентрацію CO_2 у одиницях ppm та температуру повітря в градусах Цельсія. Вибір формату JSON обумовлений його нативною підтримкою в Node-RED, зручністю розширення структури пакету в майбутньому та достатньою компактністю для передачі через канал з обмеженою пропускною здатністю. Параметри MQTT-з'єднання наведено в табл. 2.1.

Таблиця 2.1 – Параметри MQTT-з'єднання

Параметр	**Значення**
Брокер	Mosquitto (локальний)
IP-адреса брокера	192.168.1.193
Порт	1883
Топік публікації	esp32/air_quality
Формат повідомлення	JSON
Інтервал публікації	3000 мс
Авторизація	Відсутня (тестовий режим)
Віддалений доступ	Mesh VPN (Tailscale)

Для організації віддаленого доступу до дашборду Node-RED поза межами локальної мережі застосовано технологію віртуальної приватної мережі (Mesh VPN) на базі протоколу WireGuard [20] із використанням платформи Tailscale. На відміну від традиційного прокидання портів на маршрутизаторі або створення публічних тунелів (наприклад, ngrok), такий підхід є концептуально більш захищеним. Він виключає публічне експонування внутрішніх сервісів системи в мережу Інтернет, створюючи натомість зашифрований одноранговий (peer-to-peer) канал виключно між авторизованими вузлами. Кожному пристрою в такій мережі присвоюється унікальна внутрішня IP-адреса (у діапазоні 100.X.X.X), що дозволяє безперешкодно підключатися до вебпанелі керування зі смартфона навіть через мобільний зв'язок (4G/LTE) в умовах застосування провайдером NAT-трансляції.

Таким чином, обрана трирівнева архітектура забезпечує чіткий розподіл відповідальності між компонентами системи: апаратний вузол відповідає виключно за збір і первинну індикацію даних, брокер – за надійну маршрутизацію повідомлень, а Node-RED – за їх інтерпретацію та відображення. Така структура спрощує масштабування системи: у майбутньому до того ж брокера можна підключити додаткові датчики в інших кімнатах або виконавчі пристрої керування вентиляцією без необхідності вносити зміни до програмного забезпечення ESP32. Локальне розгортання всіх компонентів у межах домашньої мережі гарантує автономність платформи та її відповідність вимогам до систем моніторингу середовища, визначеним у підрозділі 1.4.

Оскільки однією з ключових експлуатаційних вимог до програмно-апаратної платформи є можливість її тривалої та стабільної роботи від стандартних побутових джерел живлення (USB-адаптерів або портативних акумуляторів типу PowerBank), необхідно провести теоретичний розрахунок загального енергоспоживання (енергетичного балансу) спроєктованої системи.

Надійність роботи IoT-пристроїв на базі WiFi-технологій критично залежить від стабільності ліній живлення. При падінні напруги нижче 3,0 В мікроконтролер ESP32 автоматично ініціює апаратне перезавантаження (Brownout Reset), що призводить до втрати з'єднання з MQTT-брокером та втрати поточних вимірювань.

Для розрахунку сумарного струму споживання I_{total} системи проаналізуємо пікові апетитні показники кожного з активних компонентів (табл. 2.2). Розрахунок ведеться для найгіршого сценарію (Worst-Case Scenario), коли всі вузли системи одночасно працюють на максимальній потужності: ESP32 відправляє пакет по Wi-Fi, сенсор здійснює запалювання інфрачервоної лампи для вимірювання, а світлодіод світиться максимальним білим кольором.

Таблиця 2.2 – Моделювання пікового струму споживання компонентів ПАП

Компонент системи	**Режим роботи**	**Піковий струм споживання (Imax), мА**	**Середній струм (I_{avg}), мА**
Мікроконтролер ESP32 (DevKit v1)	Активний стан (передача даних Wi-Fi / MQTT)	240 – 260	~100
Оптичний сенсор NDIR MH-Z19E	Цикл вимірювання (розігрів ІЧ-випромінювача)	< 150	~20
Світлодіод адресний NeoPixel (1 шт)	Максимальна яскравість (RGB: 255, 255, 255)	~60	~20 (при яскравості 50)
Допоміжна схемотехніка	Конвертер логічних рівнів, стабілізатори напруги	~10	~5
ЗАГАЛЬНО I_{total}	**Пікове навантаження системи**	**~480**	**~145**

Проведений енергетичний розрахунок доводить, що максимальний струм споживання пристрою не перевищує 500 мА. Це означає, що розроблена програмно-апаратна платформа повністю сумісна зі стандартом USB 2.0, який гарантує видачу струму до 500 мА. Таким чином, для живлення пристрою підходить абсолютно будь-який, навіть найдешевший зарядний блок від мобільного телефону на 5V/1A.

2.2 Технології та інструменти розробки апаратно-програмного комплексу

Вибір апаратної та програмної бази є одним із ключових проектних рішень, що безпосередньо визначає функціональність, надійність і вартість розроблюваної системи. У цьому підрозділі здійснено порівняльний аналіз альтернативних варіантів мікроконтролерних платформ, середовищ візуалізації, MQTT-брокерів та програмних бібліотек з метою обґрунтування вибору компонентів, використаних у програмно-апаратній платформі.

На ринку мікроконтролерних та одноплатних рішень для IoT-застосунків із підтримкою бездротового зв’язку найбільш поширеними є п’ять платформ: ESP8266, ESP32, ESP32-S3, Raspberry Pi Pico та Orange Pi Pico. Кожна з них має відмінні технічні характеристики, що зумовлюють різну придатність для задач безперервного збору телеметрії та мережевої передачі даних у реальному часі.

Мікроконтролер ESP8266 (рис.2.2) є одноядерним SoC із тактовою частотою до 160 МГц, що підтримує лише один апаратний UART-інтерфейс. Зазначена обмеженість є критичним недоліком у контексті даної роботи: єдиний UART необхідний для зв'язку з датчиком MH-Z19E за протоколом UART, тоді як виведення діагностичних даних потребує окремого послідовного каналу. Застосування програмного SoftwareSerial як альтернативи є можливим, однак призводить до підвищеного навантаження на центральний процесор і нестабільності при одночасній роботі мережевого стеку. Крім того, відсутність підтримки Bluetooth обмежує потенційне розширення функціональності платформи.

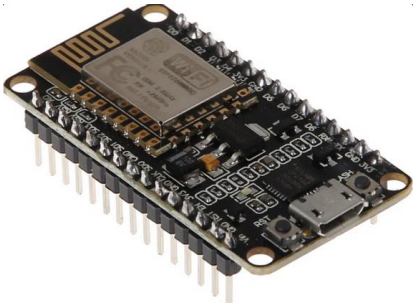

Рисунок 2.2 – Загальний вигляд ESP8266 [23]

Одноплатний комп'ютер Raspberry Pi Pico W (рис. 2.3) є мікроконтролерною платою на базі двоядерного процесора RP2040 (Cortex-M0+ 133 МГц) від Raspberry Pi Foundation. Плата позбавлена вбудованого Wi-Fi та Bluetooth у базовій версії, що потребує підключення зовнішнього модуля бездротового зв’язку для реалізації MQTT-комунікації [19]. Відсутність апаратного Wi-Fi є принциповим обмеженням для задач даної роботи, оскільки бездротова передача телеметрії є невід’ємною частиною архітектури системи.

Рисунок 2.3 – Загальний вигляд Raspberry Pi Pico W [24]

Orange Pi Pico (рис. 2.4) є мікроконтролерною платою від компанії Shenzhen Xunlong Software, що також базується на процесорі RP2040. За апаратними характеристиками Orange Pi Pico є практично ідентичним до Raspberry Pi Pico: двоядерний Cortex-M0+ 133 МГц, 264 кбайт оперативної пам’яті, підтримка UART, SPI, I2C та USB. Ключовою відмінністю є нижча роздрібна вартість, що робить Orange Pi Pico привабливим для масових застосунків. Проте, так само як і Raspberry Pi Pico, базова версія плати не містить вбудованого WiFi-модуля, а офіційна версія Orange Pi Pico W з бездротовим зв’язком є менш поширеною і гірше задокументованою порівняно з ESP32, що ускладнює її застосування у проектах з MQTT-комунікацією.

Рисунок 2.4 – Загальний вигляд Orange Pi Pico [25]

ESP32-S3 (рис. 2.5) є розширеною версією ESP32 від Espressif Systems з покращеними характеристиками: двоядерний процесор Xtensa LX7 з тактовою частотою до 240 МГц, вдосконалений Wi-Fi 802.11 b/g/n та Bluetooth 5.0 LE, апаратне прискорення векторних операцій для задач машинного навчання, а також підтримка USB OTG. Збільшений до 512 кбайт обсяг SRAM та розширена периферія роблять ESP32-S3 перспективним для застосунків із локальним ML-inference. Разом з тим, для задачі збору та передачі двох числових параметрів датчика ці додаткові можливості є функціонально надлишковими, а вища вартість і менша кількість готових прикладів порівняно з ESP32 є аргументами не на його користь.

Рисунок 2.5 – Загальний вигляд ESP32-S3 [26]

Мікроконтролер ESP32 поєднує двоядерний процесор Xtensa LX6 із тактовою частотою 240 МГц, три апаратних UART-інтерфейси, вбудовану підтримку Wi-Fi 802.11 b/g/n та Bluetooth Low Energy 4.2. Наявність операційної системи реального часу FreeRTOS дозволяє організувати паралельне виконання задач збору сенсорних даних та обслуговування мережевого стеку без взаємних блокувань. Робоча напруга 3,3 В забезпечує безпосередню сумісність із логічними рівнями більшості периферійних пристроїв. Сукупність зазначених характеристик визначає ESP32 як оптимальний вибір для апаратного вузла розроблюваної платформи. У табл. 2.4 наведено зведений порівняльний аналіз усіх розглянутих мікроконтролерних платформ.

Таблиця 2.3 – Порівняльний аналіз мікроконтролерних платформ

Критерій	**ESP8266**	**ESP32**	**ESP32-S3**	**RPi Pico W**	**Orange Pi Pico W**
Архітектура	Tensilica L106 Single-core	Xtensa LX6 Dual-core	Xtensa LX7 Dual-core	ARM Cortex-M0+ Dual-core	ARM Cortex-M0+ Dual-core
Тактова частота, МГц	80–160	240	240	133	133
Оперативна пам'ять, кбайт	~50 (вільно)	520	512	264	264
Wi-Fi (вбудований)	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n	802.11 b/g/n (Pico W)	802.11 b/g/n (Pico W)
Bluetooth	Відсутній	BLE 4.2	BLE 5.0	BLE 5.2 (Pico W)	Відсутній
Апаратних UART	1 повноцінний	3	3	2	2
Споживання (активне), мА	~80	~100	~100	~30	~30
Режим Deep Sleep, мкА	~20	< 10	< 7	~0,8 (Pico W)	~0,8
Робоча напруга, В	3,3	3,3	3,3	3,3	3,3

Критерій	ESP8266	ESP32	ESP32-S3	RPi Pico W	Orange Pi Pico W
Вартість (відносна)	Низька	Низька	Середня	Низька	Дуже низька
Екосистема / бібліотеки	Широка	Широка	Широка	Середня	Обмежена
Придатність для ПАП	Обмежена (1 UART)	Оптимальна	Прийнятна (надлишок)	Обмежена (без Wi-Fi)	Обмежена (без Wi-Fi)

З огляду на необхідність наявності апаратного UART для підключення датчика якості повітря, підтримки стабільного бездротового з'єднання та жорстких вимог до компактності й енергоефективності, мікроконтролер ESP32 був безальтернативно обраний як «серце» системи.

Нижче наведено UML-діаграму (рис. 2.6) розгортання (Deployment diagram), що відображає концепцію взаємодії архітектури обраного мікроконтролера ESP32 із зовнішнім світом.

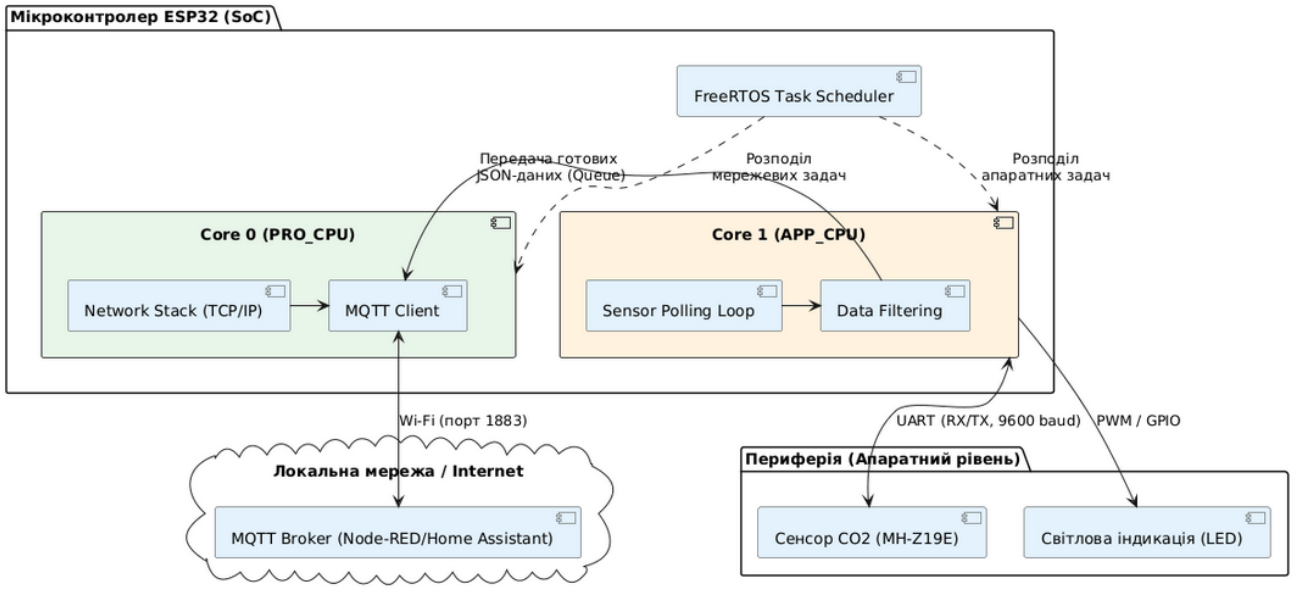

Рисунок 2.6 – UML-діаграма архітектури взаємодії компонентів на базі мікроконтролера ESP32

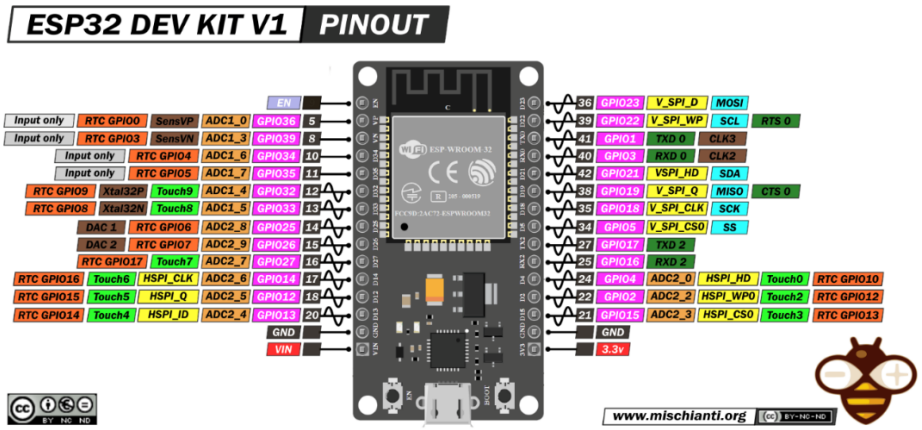

Рисунок 2.7 – Вигляд макетної плати ESP32 DevKit V1

На рис. 2.7 представлено макетну плату ESP32 DevKit V1, яка має вбудований USB-UART конвертер (CP2102) для зручного програмування, стабілізатор напруги на 3,3 В та розведену колодку контактів для прототипування.

Визначившись із обчислювальною платформою, наступним, і найбільш критичним кроком, став вибір вимірювального перетворювача. Дослідницький досвід у галузі вбудованих систем підтверджує, що ефективність моніторингу якості повітря критично залежить саме від селективності обраного датчика. Багато розробників на початкових етапах віддають перевагу дешевим напівпровідниковим газовим сенсорам (наприклад, серії MQ, зокрема MQ-135). Хоча вони і є дуже дешевими, їх головний недолік – перехресна чутливість. Вони реагують на широкий спектр газів (від парів спирту до вологості повітря) і не здатні виміряти абсолютну концентрацію саме діоксиду вуглецю, надаючи лише умовний рівень забруднення.

На противагу їм, оптичні датчики, що працюють за технологією NDIR (Non-Dispersive Infrared – недисперсійне інфрачервоне випромінювання), забезпечують вимірювання професійного рівня. Основним компонентом нашого вимірювального тракту було обрано інтелектуальний інфрачервоний сенсор MH-Z19E.

Вибір саме цієї моделі обумовлений її високою селективністю до молекул CO_2 та наявністю заводського калібрування, що значно спрощує процес налаштування пристрою. На відміну від електрохімічних аналогів, оптичний

сенсор не піддається швидкій деградації хімічного елемента і не потребує регулярної рекалібровки в польових умовах.

Датчик MH-Z19E працює на фізичному принципі поглинання газом інфрачервоного випромінювання на специфічній довжині хвилі (приблизно 4.26 мкм для вуглекислого газу). Всередині приладу знаходиться мініатюрна оптична камера, покрита позолотою (для мінімізації втрат світла), на одному кінці якої розташоване джерело ІЧ-випромінювання, а на іншому – вузькосмуговий оптичний фільтр та детектор. Чим вища концентрація CO_2 в камері, тим більше ІЧ-випромінювання поглинається газом, і тим менше світла доходить до детектора. Вбудований мікропроцесор датчика аналізує цю різницю, проводить апаратну термокомпенсацію і видає вже готове цифрове значення. Технічні параметри обраного модуля наведено в табл. 2.4.

Таблиця 2.4 – Основні технічні характеристики сенсора MH-Z19E

Параметр	**Значення**
Об’єкт вимірювання	Діоксид вуглецю (CO2)
Технологія вимірювання	Недисперсійна інфрачервона (NDIR)
Діапазон вимірювання	400 – 5000 ppm
Робоча напруга	4.5 В – 5.5 В DC
Струм споживання (піковий)	< 150 мА
Вихідний сигнал	UART, PWM
Точність	± (50 ppm + 5% від значення)
Час виходу на стабільний режим	3 хв

Додатковою і дуже вагомою перевагою цього сенсора є вбудований UART-інтерфейс. Він дозволяє з'єднати сенсор з мікроконтролером безпосередньо і зчитувати готові цифрові значення концентрації без необхідності реалізовувати складну аналогову обробку сигналу в коді (як це робиться при використанні ШІМ-виходу).

Однак, під час проєктування електричної схеми пристрою виникла важлива інженерна задача, пов'язана з різницею логічних рівнів. Схема узгодження логічних рівнів є обов'язковою, оскільки сенсор MH-Z19E живиться від 5 В, в той час як

мікроконтролер ESP32 має робочу напругу логіки 3,3 В. Ігнорування цієї принципової різниці та пряме підключення ліній передачі даних може призвести до незворотного пошкодження портів мікроконтролера вже в перші години роботи системи. Для вирішення цієї проблеми у схемотехніці нашої ПАП застосовується чотириканальний двонаправлений перетворювач логічних рівнів AOC537 на базі польових транзисторів із N-каналом, який безпечно знижує напругу на лінії TX сенсора з 5 В до 3,3 В для піну RX ESP32 та підвищує командний сигнал ESP32 з 3,3 В до 5 В для коректного розпізнавання сенсором.

Процес комунікації між мікроконтролером та сенсором має строгу часову діаграму, яку наведено нижче на рис. 2.8.

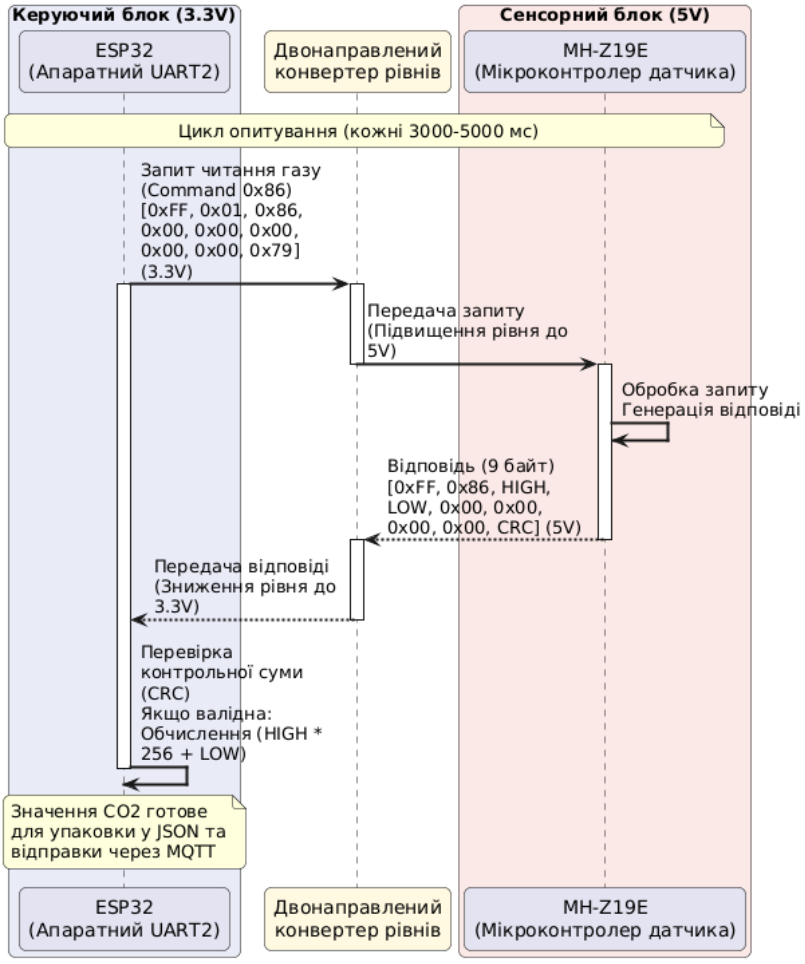

Рисунок 2.8 – UML-діаграма послідовності взаємодії ESP32 та MH-Z19E через конвертер логічних рівнів

Для реалізації програмної логіки мікроконтролера ESP32 та налаштування його взаємодії з периферійними пристроями було обрано інтегроване середовище

розробки Arduino IDE. Хоча історично ця платформа створювалася для мікроконтролерів сімейства AVR, на сьогоднішній день вона є повноцінним та потужним інструментом для розробки складних IoT-систем завдяки офіційній підтримці архітектури ESP32 від компанії-виробника Espressif Systems. Вибір саме Arduino IDE як основного програмного інструментарію зумовлений низкою вагомих переваг для даного проєкту:

1) інтегрований менеджер плат (Board Manager): дозволяє в автоматичному режимі розгорнути весь необхідний ланцюжок інструментів (toolchain), включаючи компілятор xtensa-esp32-elf-gcc та утиліту для прошивки esptool.py, без необхідності складної ручної конфігурації змінних середовища;

2) мова програмування: написання коду здійснюється адаптованим діалектом мови C/C++. Це дозволяє використовувати об'єктно-орієнтований підхід, створювати власні класи для опитування датчиків та ефективно керувати пам'яттю за допомогою вказівників;

3) екосистема бібліотек: найбільшою перевагою середовища є доступ до величезної бази готових, оптимізованих та перевірених спільнотою бібліотек. Це дає змогу інкапсулювати складні процеси (наприклад, криптографію для Wi-Fi або формування мережевих пакетів) і зосередитися на реалізації високорівневої бізнес-логіки пристрою.

Для оптимізації процесу розробки та зменшення кількості низькорівневого коду у проєкті використовується ряд стандартних та сторонніх бібліотек. Основні з них наведено у табл. 2.5.

Таблиця 2.5 – Перелік підключених програмних бібліотек для ESP32 у середовищі Arduino IDE

Назва бібліотеки	**Призначення у проєкті**	**Обґрунтування використання**
WiFi.h	Керування бездротовим модулем	Входить до стандартного ядра ESP32. Забезпечує підключення до локальної мережі за протоколом WPA2-Personal, отримання IP-адреси через DHCP та підтримку стабільного з'єднання.

Назва бібліотеки	Призначення у проєкті	Обґрунтування використання
HardwareSerial.h	Робота з апаратними портами UART	Використовується для надійної комунікації з датчиком MH-Z19E на швидкості 9600 бод. Використання апаратного буфера виключає втрату байтів під час передачі даних.
PubSubClient.h	Реалізація MQTT-клієнта [19]	Одна з найстабільніших бібліотек для роботи з протоколом MQTT. Відповідає за формування топіків (topics) та публікацію (publish) зібраних метрик на серверний брокер.
ArduinoJson.h	Серіалізація даних у формат JSON	Дозволяє ефективно формувати структуровані повідомлення (ключ-значення) без ручної конкатенації рядків, запобігаючи фрагментації оперативної пам'яті (Heap) мікроконтролера.
Adafruit_NeoPixel.h	Контроль RGB	Керування адресними RGB-світлодіодами WS2812B за однопровідним протоколом

Використання середовища Arduino IDE у поєднанні з переліченими програмними компонентами дозволило створити стабільний, читабельний та модульний програмний код, який легко піддається подальшому масштабуванню.

Підсумовуючи, обрана компонентна база у вигляді зв'язки мікроконтролера ESP32 та високоточного NDIR-сенсора MH-Z19E забезпечує ідеальний баланс між продуктивністю, точністю вимірювань та загальною вартістю кінцевого пристрою. Така архітектура дозволяє створити надійний вузол, здатний працювати роками без необхідності регулярного втручання людини чи проведення додаткових калібрувань, при цьому повністю відповідаючи актуальним вимогам до сучасних IoT-рішень у сфері екологічного моніторингу закритих приміщень.

2.3 Моделювання алгоритмів збору та опрацювання даних

Програмне забезпечення мікроконтролера ESP32 реалізовано в середовищі Arduino IDE мовою C++ і структурно складається з двох обов'язкових функцій: *setup()* та *loop()*. Функція *setup()* виконується одноразово під час старту пристрою і

відповідає за ініціалізацію всіх підсистем: послідовного порту для налагодження, UART-каналу зв'язку з датчиком MH-Z19E, модуля NeoPixel, підключення до WiFi-мережі та конфігурацію MQTT-клієнта. Функція *loop()* циклічно виконується протягом усього часу роботи пристрою і реалізує основну логіку збору, опрацювання та передачі телеметричних даних. На рис. 2.9 наведено блок-схему алгоритму ініціалізації системи у функції.

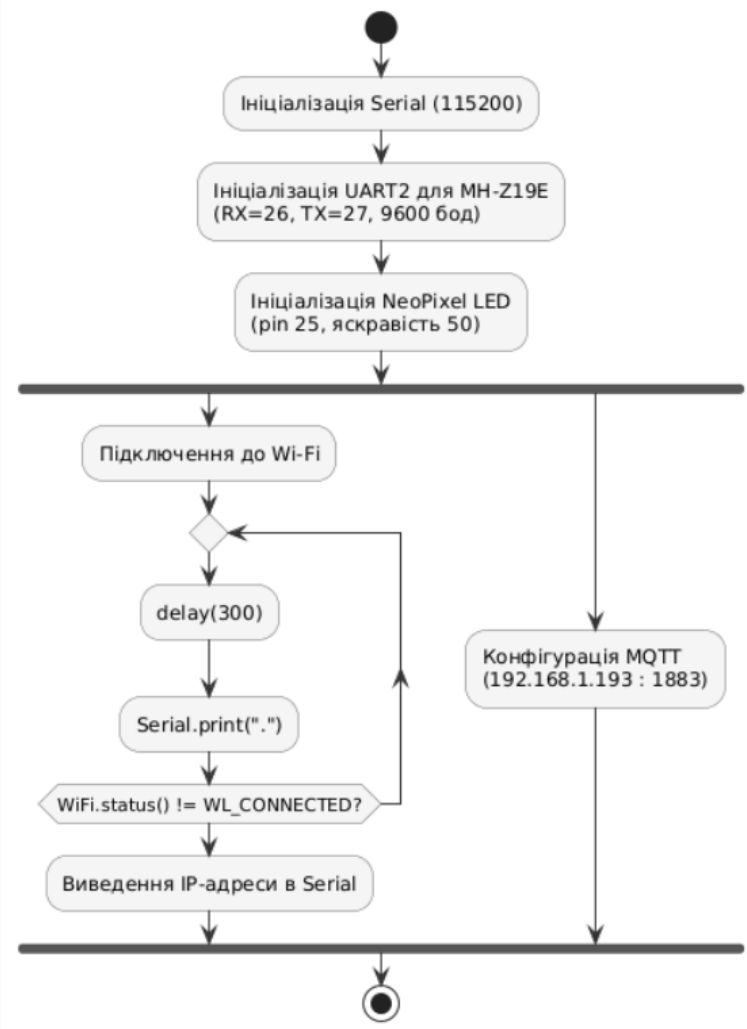

Рисунок 2.9 – Блок-схема алгоритму ініціалізації системи у функції

Функція *loop()* циклічно виконується протягом усього часу роботи пристрою і реалізує три паралельні задачі: підтримку MQTT-з'єднання, опитування датчика за таймером та керування індикацією. Детальну логіку виконання функції *loop()* наведено на рис. 2.10.

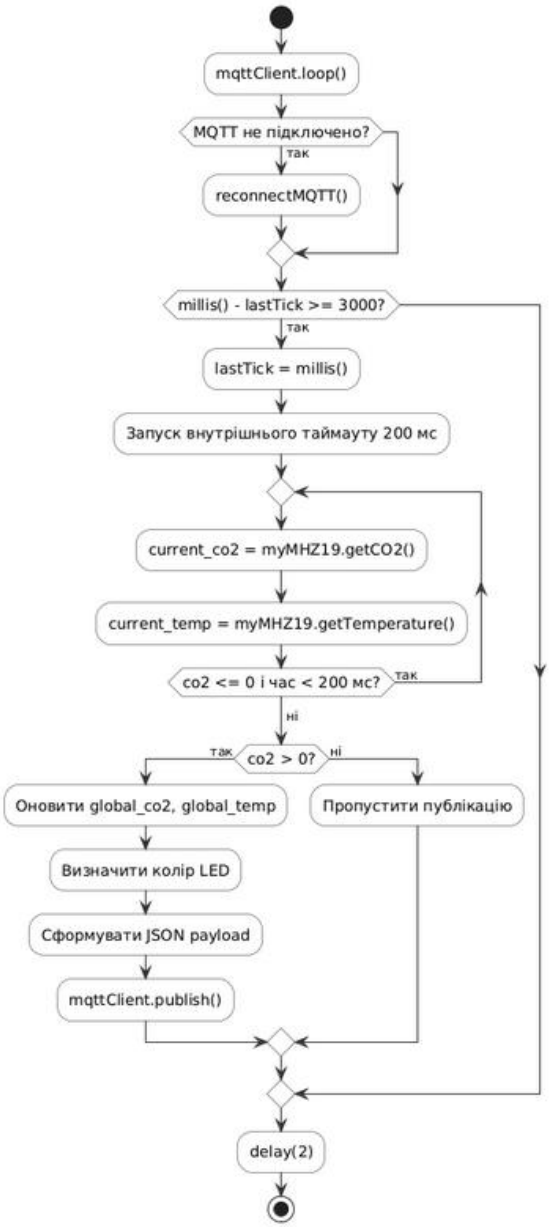

Рисунок 2.10 – Алгоритм роботи функції

Окремої уваги заслуговує реалізація функції *reconnectMQTT()*, що забезпечує автоматичне відновлення з'єднання з брокером у разі його розриву. При кожній спробі підключення генерується унікальний ідентифікатор клієнта на основі випадкового шістнадцяткового числа, що запобігає конфліктам при повторних підключеннях до брокера. У разі невдалої спроби функція очікує 5 секунд перед наступною ітерацією, виводячи діагностичний код стану в Serial-монітор. Такий підхід гарантує безперервність збору даних після короткочасних збоїв мережі без необхідності ручного перезавантаження пристрою. Також в табл. 2.6 наведено логіку триступеневої індикації NeoPixel

Таблиця 2.6 – Логіка триступеневої індикації NeoPixel

Зона	**Діапазон CO2**	**Колір LED**	**Інтерпретація**
Норма	400 – 799 ppm	Зелений (0, 255, 0)	Якість повітря задовільна
Увага	800 – 1199 ppm	Жовтогарячий (255, 180, 0)	Рекомендується провітрювання
Небезпека	1200 ppm	Червоний (255, 0, 0)	Необхідне негайне провітрювання

Перелік ключових програмних об'єктів та їх призначення у системі наведено в табл. 2.7.

Таблиця 2.7 – Ключові програмні об'єкти системи

Об'єкт	**Тип**	**Призначення**
myMHZ19	MHZ19	Керування датчиком CO_2 через UART
pixels	Adafruit_NeoPixel	Керування RGB-світлодіодом NeoPixel
mqttClient	PubSubClient	MQTT-клієнт для публікації телеметрії
mySerial2	HardwareSerial	UART2-канал зв'язку з MH-Z19E
global_co2	int	Глобальна змінна поточного CO_2
global_temp	int	Глобальна змінна поточної температури
lastSensorTick	unsigned long	Лічильник для неблокуючого таймера

Функція *setup()* виконує послідовну ініціалізацію всіх підсистем у суворо визначеному порядку. Першим ініціалізується послідовний порт Serial.begin(115200) для виведення діагностичних повідомлень у Serial-монітор під час налагодження. Далі конфігурується UART2-канал зв'язку з датчиком: *mySerial2.begin(9600, SERIAL_8N1, 26, 27)*, після чого об'єкт датчика прив'язується до цього каналу викликом *myMHZ19.begin(mySerial2)*. Модуль NeoPixel активується парою викликів *pixels.begin()* та *pixels.setBrightness(50)*. Підключення до Wi-Fi реалізовано блокуючим циклом очікування з виведенням крапок у Serial-монітор до моменту отримання IP-адреси. Завершальним кроком *setup()* є реєстрація адреси MQTT-брокера [19] через *mqttClient.setServer()*, після якої пристрій готовий до входу в основний робочий цикл.

Формування телеметричного пакету перед публікацією відбувається безпосередньо у функції *loop()* методом конкатенації рядків мовою C++. Структура пакету є фіксованою і містить два поля: co2 з поточним значенням концентрації

вуглекислого газу в одиницях ppm та temp з температурою повітря в градусах Цельсія. Приклад сформованого рядка для публікації наведено нижче (рис. 2.11):

```
{"co2": 739, "temp": 24}
```

Рисунок 2.11 – Приклад рядка

Окремої уваги заслуговує створення неблокуючого циклу в функції *loop()*. На відміну від типового підходу з використанням *delay()*, який призупиняє виконання всієї програми на визначений час, у даній реалізації застосовано таймер на основі функції *millis()*. Змінна lastSensorTick зберігає мітку часу останнього успішного зчитування, а нове вимірювання ініціюється лише тоді, коли різниця між поточним і збереженим значенням *millis()* перевищує 3000 мс. Така архітектура дозволяє мікроконтролеру в проміжках між опитуваннями датчика безперервно обслуговувати MQTT-стек через виклик *mqttClient.loop()*, що є обов'язковою умовою підтримки активного з'єднання з брокером і своєчасного опрацювання вхідних службових пакетів. Відмова від блокуючого *delay()* також усуває ризик накопичення затримок при тимчасових збоях зчитування датчика – якщо *getCO2()* повертає від'ємне або нульове значення, система просто пропускає поточну ітерацію публікації і відновлює нормальну роботу в наступному циклі без зависання.

Незважаючи на те що стандарт JSON допускає різні способи форматування, у даній реалізації свідомо відмовлено від використання сторонніх бібліотек серіалізації на кшталт ArduinoJson на користь прямої конкатенації рядків. Такий підхід мінімізує використання оперативної пам'яті мікроконтролера та виключає додаткові залежності проекту, що є виправданим для пакету фіксованої структури з двох числових полів. Публікація сформованого рядка здійснюється викликом *mqttClient.publish(mqtt_topic, payload.c_str())*, результат якого перевіряється і виводиться в Serial-монітор для діагностики.

Реалізований алгоритм збору даних забезпечує стабільне отримання телеметрії з інтервалом 3 секунди при одночасному обслуговуванні MQTT-

з'єднання, що підтверджує доцільність застосування неблокуючої архітектури на базі *millis()* для вбудованих систем реального часу. Триступенева логіка індикації з чіткими пороговими значеннями надає користувачеві миттєвий візуальний зворотний зв'язок без необхідності звертатися до дашборду, що є важливою перевагою в умовах практичного використання пристрою.

Важливим аспектом опрацювання даних, що надходять від інфрачервоного сенсора MH-Z19E, є їх первинна цифрова фільтрація на рівні мікроконтролера. Незважаючи на високу точність NDIR-технології, будь-які оптичні вимірювальні прилади схильні до впливу апаратного шуму. Мікроколивання напруги живлення, наявність пилу в оптичній камері або різкі зміни температури повітря можуть викликати короткочасні викиди (флуктуації) у значеннях концентрації CO_2.

Наявність таких флуктуацій є критичною для нашої системи індикації. Згідно з розробленою логікою, пристрій змінює колір світлодіода NeoPixel з зеленого на жовтий при перетині порогу у 800 ppm. Якщо реальна концентрація в кімнаті коливається на рівні 799 ppm, апаратний шум сенсора може видавати послідовність значень: 799, 802, 798, 804 ppm. Це призведе до постійного, хаотичного блимання індикатора, що є неприйнятним з точки зору користувацького досвіду (UX) та створює зайве навантаження на MQTT-брокер, ініціюючи відправку надлишкових сповіщень.

Для усунення цієї проблеми в програмному забезпеченні ESP32 на етапі зчитування даних застосовано математичну модель експоненційного ковзного середнього Exponential Moving Average. На відміну від простого арифметичного усереднення, метод EMA не потребує створення великих масивів у пам'яті мікроконтролера для зберігання попередніх значень, що суттєво економить ресурси оперативної пам'яті (SRAM).

Математично алгоритм фільтрації описується наступним рекурентним рівнянням:

$$S_t = \alpha \cdot Y_t + (1 - \alpha) \cdot S_{\{t-1\}}$$

де S_t – нове відфільтроване значення концентрації CO_2, яке буде передано на індикацію та MQTT-брокер;

Y_t – поточне «сире» значення, щойно зчитане з UART-інтерфейсу сенсора MH-Z19E;

S_{t-1} – відфільтроване значення з попереднього кроку вимірювання;

α – коефіцієнт згладжування (ваговий коефіцієнт), що лежить у діапазоні $0 < \alpha \leq 1$.

Вибір коефіцієнта α визначає ступінь інертності системи. Якщо α наближається до 1, фільтр стає дуже чутливим до нових даних і майже не згладжує шум. Якщо α наближається до 0, фільтр максимально ігнорує шум, але система починає реагувати на зміну якості повітря з великим запізненням.

Шляхом експериментального моделювання для даного апаратно-програмного комплексу було обрано коефіцієнт $\alpha = 0.2$\$. Це забезпечує оптимальний баланс: випадкові викиди (шуми до ±15 ppm) ефективно ігноруються системою, тоді як на реальне зростання концентрації вуглекислого газу пристрій реагує протягом 10-15 секунд, чого цілком достатньо для моніторингу мікроклімату приміщень.

Додатковим механізмом стабілізації, реалізованим у програмному коді, є програмний гістерезис. Алгоритм індикації побудований таким чином, що повернення з зони «Увага» (жовтий колір) до зони «Норма» (зелений колір) відбувається не при 799 ppm, а лише тоді, коли концентрація впаде нижче 750 ppm. Поєднання експоненційної фільтрації та гістерезису гарантує стабільність роботи візуального інтерфейсу пристрою за будь-яких умов навколишнього середовища.

Висновки до розділу 2

У другому розділі було виконано комплекс інженерно-проєктних рішень щодо формування архітектури та обґрунтування компонентної бази інформаційно-вимірювальної IoT-системи моніторингу вмісту вуглекислого газу в приміщенні. На основі проведених аналітичних досліджень та схемотехнічного моделювання отримано такі результати:

– спроєктовано трирівневу структурну схему програмно-апаратного комплексу, яка чітко розмежовує функції збору первинних телеметричних даних

(апаратний рівень), бездротового транспортування інформаційних пакетів (мережевий рівень) та кінцевого накопичення, аналізу й візуалізації метрик якості повітря (рівень додатків). Дана архітектурна концепція забезпечує високу модульність системи, спрощує її технічне обслуговування та відкриває можливості для подальшого масштабування;

– на основі детального порівняльного аналізу п’яти мікроконтролерних платформ (ESP8266, ESP32, ESP32-S3, Raspberry Pi Pico W та Orange Pi Pico W) доведено технічну доцільність використання SoC ESP32 (DevKit v1). Встановлено, що ESP8266 є неприйнятним через наявність лише одного апаратного UART-інтерфейсу, Raspberry Pi Pico W та Orange Pi Pico W – через слабку екосистему та обмежену підтримку MQTT, ESP32-S3 – через функціональну надлишковість для задач даної роботи. Двоядерна архітектура Xtensa LX6 та інтегрована підтримка операційної системи реального часу FreeRTOS дозволяють ефективно розділити завдання підтримки мережевого стеку TCP/IP та безперебійного опитування периферії, повністю нівелюючи ризики критичних затримок чи зависань системи;

– обґрунтовано вибір інтелектуального оптичного сенсора вимірювання CO_2 типу MH-Z19E, що функціонує за технологією недисперсійного інфрачервоного випромінювання (NDIR). Встановлено, що на відміну від напівпровідникових аналогів з високою перехресною чутливістю, даний вимірювальний перетворювач має виняткову вибірковість до молекул діоксиду вуглецю, низьку залежність від сторонніх газів чи коливань вологості, а також забезпечує точну передачу даних через цифровий інтерфейс UART у діапазоні до 5000 ppm;

– вирішено інженерну задачу схемотехнічної сумісності логічних рівнів напруги між вимірювальним модулем (що потребує живлення 5 В) та інтегральною схемою мікроконтролера (робоча напруга логіки якого становить 3,3 В). Впровадження у схему пристрою чотириканального двонаправленого перетворювача логічних рівнів AOC537 на базі польових транзисторів із N-каналом дозволило гарантувати електричну безпеку портів введення-виведення ESP32 та високу завадостійкість ліній передачі даних;

– за допомогою уніфікованої мови моделювання UML побудовано діаграми розгортання та послідовності взаємодії компонентів. Це дозволило детально формалізувати алгоритми часових інтервалів опитування датчика, механізми апаратної термокомпенсації, процедури валідації даних за допомогою перевірки контрольних сум (CRC) та подальшого пакування метрик у JSON-структури для відправки на MQTT-брокер [19].

Спроєктована архітектурна модель та обрана елементна база є надійною теоретичною та схемотехнічною основою для переходу до наступного етапу – практичного прототипування, розробки програмного забезпечення мікроконтролера та налаштування серверних сервісів візуалізації.

3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Архітектура апаратного забезпечення системи

Апаратна реалізація IoT-пристрою для моніторингу вмісту вуглекислого газу побудована на основі мінімально необхідного набору компонентів, що відповідає вимогам компактності, енергоефективності та технологічної відтворюваності, сформованим у підрозділі 1.4. Усі елементи змонтовано на безпаяній макетній платі, що дозволяє вносити зміни до схеми без застосування паяльного обладнання та суттєво прискорює процес прототипування. Живлення всієї системи здійснюється через стандартний USB-порт напругою 5 В, що забезпечує сумісність із широким спектром побутових джерел живлення – від мережевих зарядних пристроїв до портативних акумуляторів.

Загальний перелік апаратних компонентів, використаних у прототипі, наведено в табл. 3.1.

Таблиця 3.1 – Перелік апаратних компонентів прототипу

№	Компонент	Кількість, шт	Призначення в схемі
1	Мікроконтролер ESP32 DevKit v1	1	Обчислювальне ядро, Wi-Fi, керування периферією
2	Датчик CO_2 MH-Z19E (NDIR)	1	Вимірювання концентрації CO_2 та температури
3	4-канальний перетворювач логічних рівнів AOC537	1	Узгодження рівнів 5В↔3,3 В між сенсором та ESP32
4	RGB-світлодіод NeoPixel WS2812B	1	Візуальна індикація якості повітря
5	Макетна плата (Breadboard)	1	Монтажна основа прототипу
6	З'єднувальні проводи (Dupont)	~15	Міжкомпонентні з'єднання
7	USB-кабель 5В / 1А	1	Живлення всієї системи

Структурно апаратний вузол складається з чотирьох функціональних блоків: обчислювального ядра на базі ESP32 DevKit v1 [19], вимірювального блоку на базі

сенсора MH-Z19E, схеми узгодження логічних рівнів AOC537 та блоку візуальної індикації на базі адресного RGB-світлодіода WS2812B. Взаємозв'язок між блоками та зовнішніми системами відображено на deployment-діаграмі (рис. 3.1), що побудована засобами PlantUML та відображає фізичне розгортання компонентів у межах трирівневої архітектури системи.

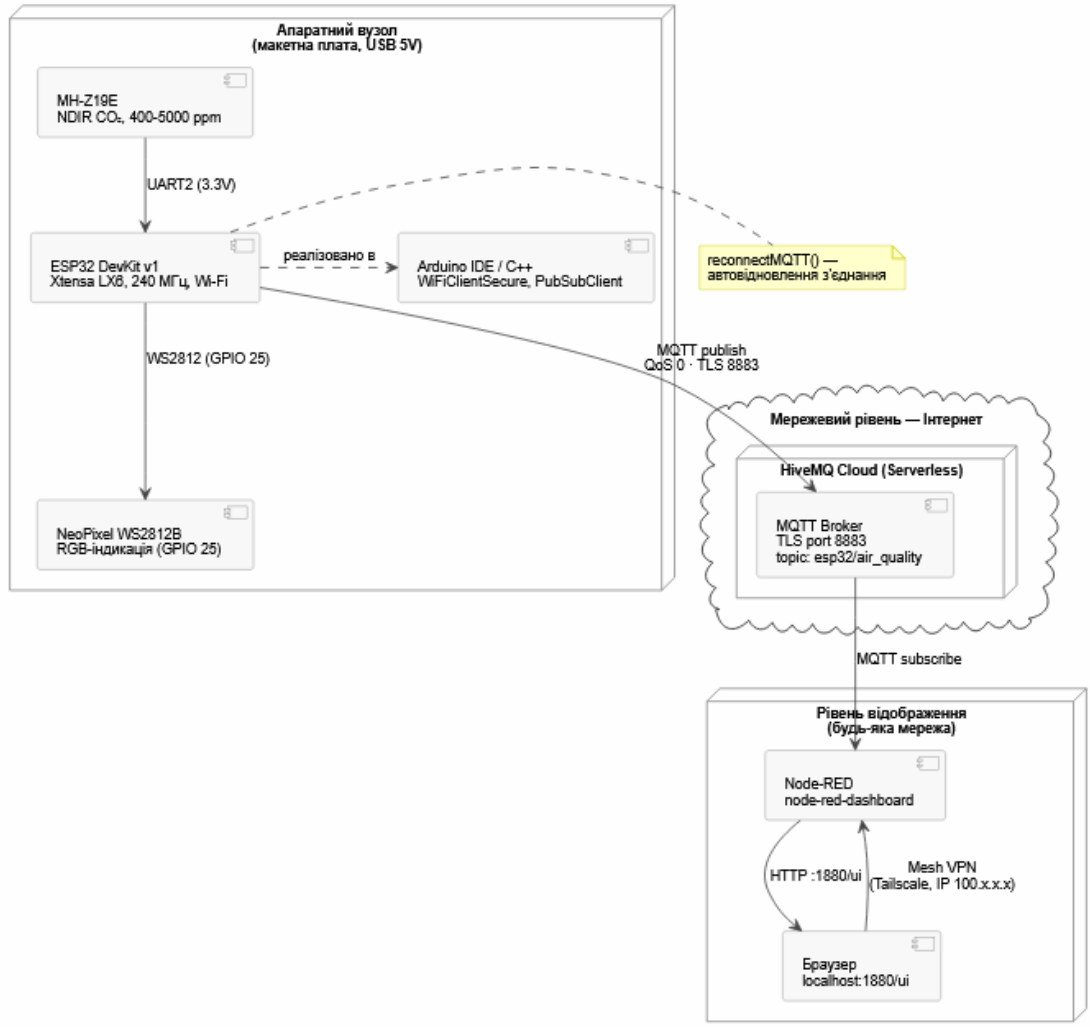

Рисунок 3.1 – Deployment-діаграма апаратного вузла IoT-системи моніторингу CO_2

Центральним елементом апаратного вузла є мікроконтролер ESP32 DevKit v1 [19]. Дана плата розробки оснащена вбудованим USB-UART конвертером CP2102, що спрощує процедуру прошивки без використання зовнішніх програматорів, та інтегрованим лінійним стабілізатором напруги AMS1117-3.3, який забезпечує стабільні 3,3 В для живлення логічної частини мікроконтролера. Двоядерний процесор Xtensa LX6 з тактовою частотою 240 МГц та операційна система

реального часу FreeRTOS дозволяють паралельно виконувати задачі опитування датчика, обслуговування мережевого стеку TCP/IP та керування індикацією без взаємних блокувань.

Підключення периферійних пристроїв до мікроконтролера реалізовано таким чином: датчик MH-Z19E підключено до апаратного порту UART2 (GPIO 26 – RX2, GPIO 27 – TX2), світлодіод WS2812B – до GPIO 25 за однопровідним протоколом WS2812. Вибір саме апаратного UART2 (а не програмного SoftwareSerial) є принциповим технічним рішенням: апаратний буфер унеможливлює втрату байтів у пакеті відповіді сенсора під час одночасної активності мережевого стеку Wi-Fi, що є типовою проблемою однопотокових мікроконтролерних рішень.

Критичним елементом схеми є чотириканальний двонаправлений перетворювач логічних рівнів AOC537, що вирішує фундаментальну проблему несумісності напруг між компонентами системи. Датчик MH-Z19E функціонує від напруги 5 В і формує сигнали UART з амплітудою 5 В, тоді як порти вводу-виводу мікроконтролера ESP32 розраховані на напругу 3,3 В [19]. Безпосереднє з'єднання ліній передачі даних без узгодження рівнів призвело б до подачі на вхід ESP32 напруги, що на 1.7 В перевищує максимально допустиму для цифрових входів мікроконтролера, що гарантовано спричинить деградацію або незворотне пошкодження вхідних каскадів GPIO вже протягом перших годин роботи.

Перетворювач AOC537 побудований на польових транзисторах із каналом N-типу та реалізує двонаправлене перетворення рівнів без необхідності керуючого сигналу напрямку передачі – напрямок комутації визначається автоматично рівнем напруги на шині. Модуль має чотири незалежних канали (B1–B4 зі сторони 5 В, A1–A4 зі сторони 3,3 В), два роздільних виводи живлення (HV для 5 В та LV для 3,3 В) та спільну землю. У даному прототипі задіяно два канали для підключення ліній UART сенсора. Схему підключення конвертера наведено в табл. 3.2.

Таблиця 3.2 – Схема підключення перетворювача логічних рівнів AOC537

Лінія UART	**Піни MH-Z19E (5B)**	**Піни ESP32 (3,3 B)**	**Канал AOC537**
TX (сенсор → МК)	TX (pin 3)	GPIO 26 (RX2)	B1 → A1
RX (МК → сенсор)	RX (pin 2)	GPIO 27 (TX2)	A2 → B2
Живлення HV	VCC (5B)	—	HV = 5B
Живлення LV	—	3,3 B	LV = 3,3 B
Спільна земля	GND	GND	GND

Завдяки застосуванню AOC537 лінія TX сенсора (5 B) безпечно знижується до 3,3 B на вході GPIO 26 мікроконтролера, а командний сигнал від ESP32 (3,3 B) на лінії TX (GPIO 27) підвищується до 5 B для коректного розпізнавання сенсором. Такий підхід гарантує електричну безпеку портів ESP32 та стабільність комунікації по протоколу UART на швидкості 9600 бод. На рис. 3.2 представлено принципову схему з'єднання всіх компонентів прототипу на макетній платі.

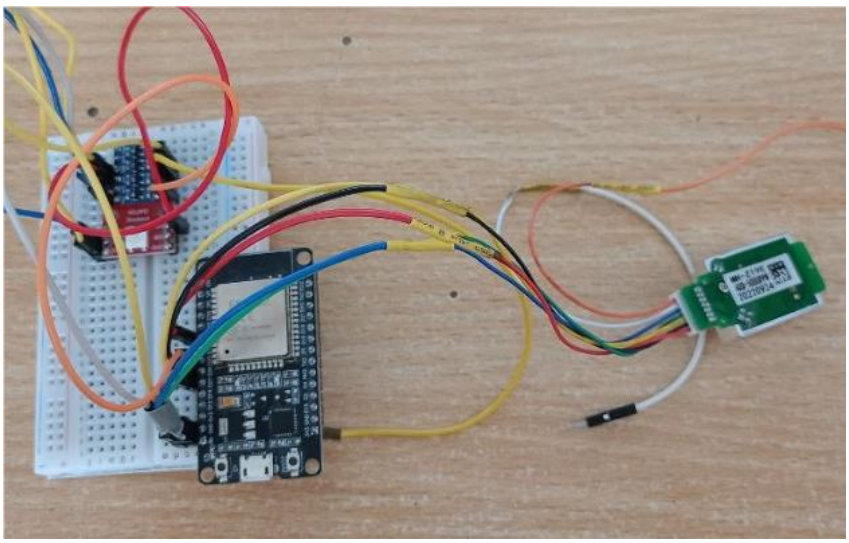

Рисунок 3.2 – Фото зібраного апаратного прототипу на макетній платі

Блок візуальної індикації реалізовано на адресному RGB-світлодіоді NeoPixel WS2812B. Даний компонент поєднує в собі три незалежних кристали (червоний, зелений, синій) та вбудований контролер WS2812, що керує яскравістю кожного каналу за 8-бітною шкалою (0–255). Управління здійснюється за

однопровідним протоколом із суворо визначеними часовими характеристиками імпульсів, що вимагає використання спеціалізованої бібліотеки Adafruit NeoPixel для коректного формування сигналу. Відповідно до логіки триступеневої індикації, розробленої в підрозділі 2.3, світлодіод відображає зелений колір при концентрації CO_2 в межах норми (до 799 ppm), жовтогарячий при підвищеній концентрації (800–1 199 ppm) та червоний при перевищенні критичного порогу (від 1 200 ppm). Робоча яскравість встановлена на рівні 50 з 255 для зниження споживання струму, що відповідає вимогам енергетичного балансу, розрахованого в підрозділі 2.1.

Порівняно з первісною архітектурою, описаною в розділі 2, у фінальній реалізації прийнято принципове рішення щодо переходу від локального MQTT-брокера Mosquitto до хмарного сервісу HiveMQ Cloud (Serverless). Дане рішення зумовлене низкою практичних переваг. По-перше, усувається залежність від постійної доступності локального Linux-сервера в домашній мережі. По-друге, забезпечується захист каналу передачі телеметрії за допомогою протоколу TLS 1.2/1.3, що виключає можливість перехоплення даних у відкритих мережах. По-третє, хмарний брокер забезпечує глобальну маршрутизацію телеметрії між пристроями незалежно від їх фізичного місцезнаходження. Водночас, для безпечного віддаленого доступу кінцевого користувача до локально розгорнутої панелі візуалізації Node-RED інтегровано програмний клієнт Tailscale. Застосування цієї технології нівелює необхідність конфігурування публічних тунелів типу ngrok або отримання статичної IP-адреси, гарантуючи прямий зашифрований доступ до дашборду безпосередньо з мобільного пристрою.

Підключення ESP32 до HiveMQ Cloud реалізовано за допомогою бібліотеки WiFiClientSecure, яка забезпечує встановлення захищеного TLS-з'єднання з верифікацією сертифіката сервера за кореневим сертифікатом ISRG Root X1 (Let's Encrypt). Автентифікація клієнта здійснюється за схемою Username/Password, що відповідає вимогам безпеки Serverless-плану HiveMQ. Параметри MQTT-з'єднання фінальної реалізації наведено в табл. 3.3.

Таблиця 3.3 – Параметри MQTT-з'єднання з хмарним брокером HiveMQ Cloud

Параметр з'єднання	**Значення**
Брокер	HiveMQ Cloud (Serverless)
Протокол	MQTT over TLS
Порт	8883
Автентифікація	Username / Password
TLS-сертифікат	ISRG Root X1 (Let's Encrypt)
Топік публікації	esp32/air_quality
Формат повідомлення	JSON
Інтервал публікації	3000 мс
QoS рівень	0 (At most once)
Клієнтська бібліотека ESP32	WiFiClientSecure + PubSubClient

Механізм автовідновлення з'єднання реалізовано у функції *reconnectMQTT(),* що циклічно опитує стан MQTT-клієнта та за потреби ініціює повторне підключення з унікальним ідентифікатором клієнта, сформованим на основі псевдовипадкового шістнадцяткового числа. Це запобігає конфліктам сесій при одночасному підключенні кількох екземплярів пристрою до одного брокера. При невдалій спробі підключення функція очікує 5 секунд перед наступною ітерацією, зберігаючи при цьому безперервне опитування датчика та оновлення індикації.

Таким чином, апаратна архітектура прототипу забезпечує повний ланцюг перетворення фізичної величини – концентрації CO_2 – у захищений цифровий потік даних, доступний для аналізу та візуалізації з будь-якої точки мережі. Компактність монтажу на макетній платі, живлення від стандартного USB-інтерфейсу та відсутність залежності від локальної серверної інфраструктури роблять розроблений пристрій практично застосовним для розгортання у приміщеннях різного призначення без спеціальних технічних знань з боку кінцевого користувача.

3.2 Програмна реалізація мікроконтролерного вузла та побудова панелі візуалізації

Програмне забезпечення мікроконтролера ESP32 розроблено в середовищі Arduino IDE 2 мовою C++. Вибір даного інструменту зумовлений наявністю офіційної підтримки плат Espressif через менеджер плат Arduino ESP32 Board Support Package, широкою бібліотечною екосистемою, а також можливістю завантаження прошивки безпосередньо через вбудований USB-UART конвертер CP2102 без застосування зовнішніх програматорів. Структура програми відповідає стандартній моделі Arduino: функція *setup()* виконується одноразово при увімкненні живлення або після апаратного скидання та відповідає за ініціалізацію всіх підсистем; функція *loop()* виконується циклічно та реалізує основну логіку роботи пристрою.

Програма використовує п'ять зовнішніх бібліотек, перелік та призначення яких наведено в табл. 3.4.

Таблиця 3.4 – Бібліотеки програми мікроконтролера ESP32

№	Бібліотека	Призначення
1	WiFi.h	Підключення до WiFi-мережі, керування мережевим інтерфейсом ESP32
2	WiFiClientSecure.h	TLS-захищений TCP-клієнт для безпечного підключення до HiveMQ Cloud
3	PubSubClient.h	MQTT-клієнт: публікація повідомлень, підтримка з'єднання, автопідключення
4	MHZ19.h	Високорівневий драйвер для сенсора MH-Z19E: читання CO_2 та температури по UART
5	Adafruit_NeoPixel.h	Керування адресними RGB-світлодіодами WS2812B за однопровідним протоколом

Секція глобальних констант містить усі параметри підключення до зовнішніх сервісів: ідентифікатор та пароль WiFi-мережі, адресу хмарного MQTT-брокера

HiveMQ Cloud, порт 8883, облікові дані клієнта та назву топіка. Виділення цих параметрів у окремий блок на початку файлу спрощує перенастройку пристрою для роботи в іншій мережевій інфраструктурі без змін в основній логіці програми. Крім констант, оголошено чотири глобальні об'єкти: mySerial2 – об'єкт апаратного UART2, myMHZ19 – об'єкт драйвера сенсора, pixels – об'єкт керування NeoPixel, а також пара espClient / mqttClient, що утворює стек захищеного MQTT-з'єднання.

Критичним етапом налаштування взаємодії між локальним сервером Node-RED та хмарним брокером HiveMQ Cloud є конфігурація підсистеми безпеки. Оскільки Serverless-рішення HiveMQ функціонують виключно через зашифровані канали зв'язку, використання стандартного протоколу MQTT без криптографічного захисту є неможливим. Для забезпечення успішної маршрутизації пакетів (із використанням механізму SNI – Server Name Indication) та проходження перевірки автентичності сервера, у вузлі конфігурації MQTT-брокера (mqtt-broker node) було обов'язково активовано підтримку протоколу TLS.

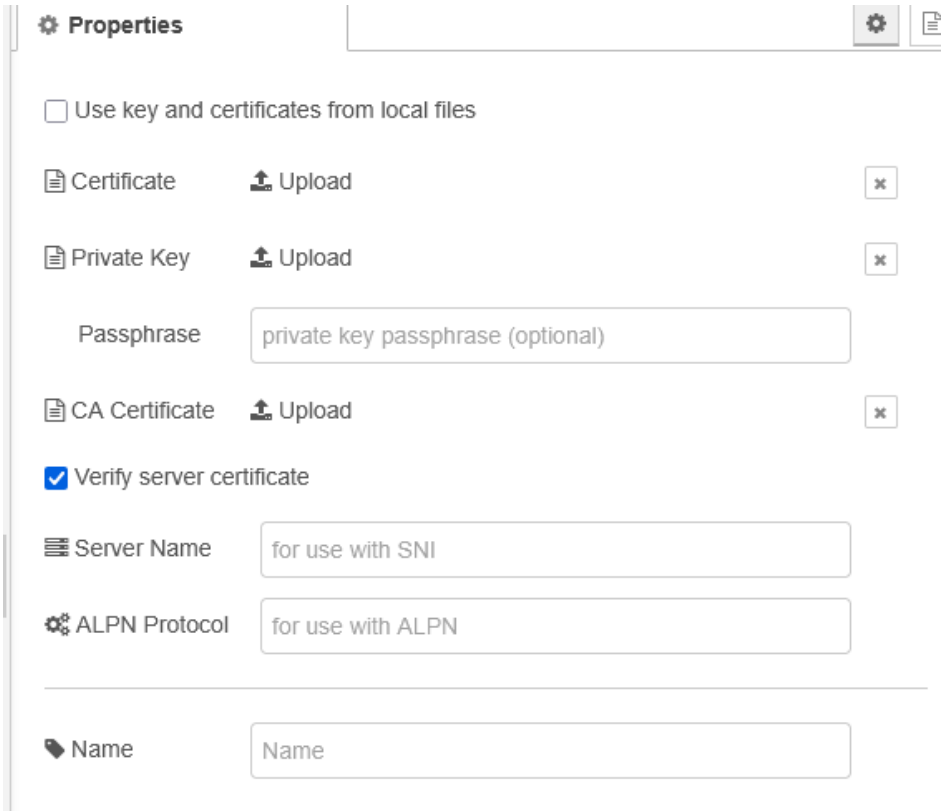

Рисунок 3.3 – Налаштування профілю TLS для маршрутизації SNI у Node-RED

З цією метою в середовищі Node-RED було створено окремий конфігураційний профіль tls-config (рис. 3.3). Ключовою особливістю реалізованого налаштування є відмова від ручного завантаження файлів публічних та приватних ключів. Натомість Node-RED налаштовано на використання вбудованих кореневих сертифікатів базової операційної системи для валідації хмарного брокера. Це дозволяє системі автоматично підтверджувати сертифікат безпеки сервера, гарантуючи надійне шифрування транспортного рівня без додаткового адміністративного навантаження.

Після створення профілю безпеки, у налаштуваннях підключення ноди mqtt in (вкладка Connection) було активовано опцію «Use TLS» із прив'язкою до створеного конфігуратора (рис. 3.4). У поєднанні з налаштованим захищеним портом 8883 та введеними обліковими даними пристрою (вкладка Security), це гарантує стабільний та захищений від перехоплення прийом телеметричних пакетів від мікроконтролера.

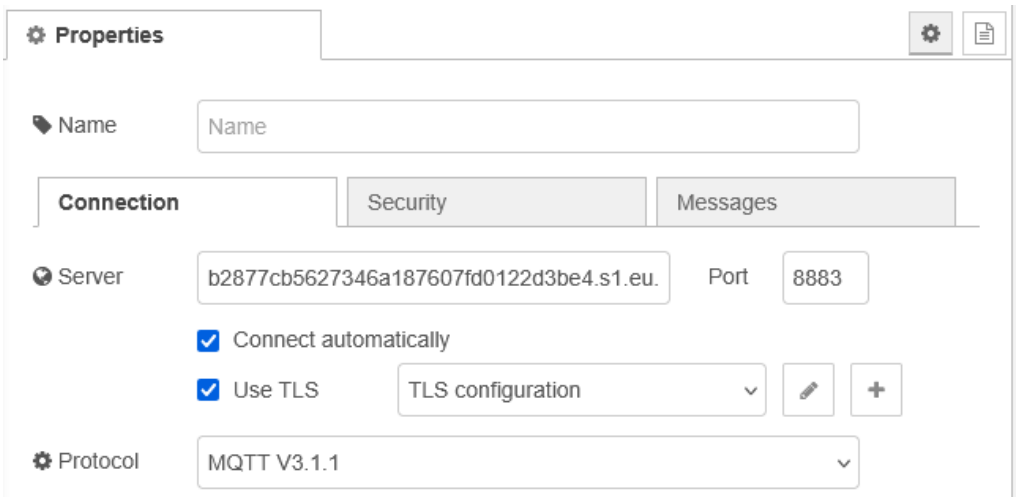

Рисунок. 3.4 – Активація захищеного з'єднання у вузлі MQTT-брокера

Функція *setup()* виконує послідовну ініціалізацію чотирьох підсистем. По-перше, запускається апаратний порт UART2 на швидкості 9600 бод із прив'язкою до GPIO 26 (RX) та GPIO 27 (TX); об'єкт myMHZ19 прив'язується до цього порту. По-друге, ініціалізується контролер NeoPixel: встановлюється яскравість 50 з 255 та формується нульовий імпульс показу для переходу світлодіода в стан «вимкнено». По-третє, виконується підключення до Wi-Fi: програма входить у

блокуючий цикл з кроком 300 мс до отримання статусу WL_CONNECTED. По-четверте, налаштовується TLS-клієнт: виклик *espClient.setInsecure()* вимикає перевірку сертифіката сервера, після чого *mqttClient.setServer()* вказує адресу та порт брокера. Застосування режиму *setInsecure()* є допустимим компромісом для прототипу, оскільки забезпечує шифрування каналу передачі (виключає перехоплення даних) при одночасному спрощенні управління кореневими сертифікатами на мікроконтролері.

Центральним елементом програми є функція *loop()*, блок-схему якої наведено на рис. 3.5. Перший крок кожної ітерації – перевірка стану MQTT-з'єднання: якщо метод *mqttClient.connected()* повертає false, викликається функція *reconnectMQTT()*. Далі обов'язково викликається *mqttClient.loop()* – це обслуговування внутрішнього стану MQTT-бібліотеки (підтримка keep-alive пакетів та обробка вхідних повідомлень). Опитування датчика реалізовано за таймерним шаблоном на основі функції *millis()*: змінна lastSensorTick зберігає мітку часу останнього успішного зчитування, і нове зчитування виконується лише тоді, коли з того моменту минуло не менше 3 000 мс. Такий підхід є принципово кращим за блокуючий виклик *delay(3000)*, оскільки дозволяє MQTT-клієнту та мережевому стеку продовжувати роботу протягом усього циклу опитування.

```
void loop() {
    if (!mqttClient.connected()) {
        reconnect();
    }
    mqttClient.loop();
```

Рисунок 3.5 – Блок-схема функції *loop()* програми ESP32

Зчитування даних з сенсора реалізовано у вигляді внутрішнього циклу тривалістю 200 мс із достроковим виходом при отриманні валідної відповіді ($CO_2 > 0$). Виклики *myMHZ19.getCO2()* та *myMHZ19.getTemperature()* повертають останні декодовані значення з буфера бібліотеки MHZ19. Перевірка current_co2 > 0 служить фільтром некоректних відповідей, що можуть виникати протягом першої хвилини після ввімкнення сенсора (прогрівання NDIR-джерела)

або при збоях UART-комунікації. Лістинг ключової ділянки коду опитування датчика на рис. 3.6:

```
static unsigned long lastSensorTick = 0;

if (millis() - lastSensorTick >= 3000) {
    lastSensorTick = millis();

    int current_co2 = -1;
    int current_temp = -1;

    unsigned long t = millis();
    while (millis() - t < 200) {
        current_co2 = myMHZ19.getCO2();
        current_temp = myMHZ19.getTemperature();
        if (current_co2 > 0) break;
        yield();
    }
```

Рисунок 3.6 – Лістинг коду

При отриманні валідного значення CO_2 програма виконує три паралельні дії: оновлення глобальних змінних, керування індикацією та публікацію телеметрії. Логіка вибору кольору реалізована трирівневою умовою: концентрація нижче 800 ppm формує зелений колір (RGB 0, 255, 0) – повітря в нормі; від 800 до 1 199 ppm – жовтогарячий (RGB 255, 180, 0) – рекомендується провітрювання; від 1 200 ppm – червоний (RGB 255, 0, 0) – критичний рівень, необхідне негайне провітрювання. Виклик *setLED()* записує колір у буфер NeoPixel та фізично відправляє керуючий імпульс через GPIO 25 викликом *pixels.show()*. Лістинг функції *setLED()* на рис. 3.7:

```
3  void setLED(int r, int g, int b) {
4      pixels.setPixelColor(0, pixels.Color(r, g, b));
5      pixels.show();
6  }
```

Рисунок 3.7 – Лістинг коду

Публікація здійснюється методом *mqttClient.publish()*, що передає рядок у топік esp32/air_quality з рівнем QoS 0. Результат публікації перевіряється за булевим значенням, що повертається методом; у разі невдачі виводиться діагностичне повідомлення в Serial-монітор. Завершальна затримка delay(2) в кінці *loop()* запобігає watchdog-скиданню процесора при надмірно швидкій ітерації циклу без активних мережевих подій.

Функція *reconnectMQTT()* реалізує блокуючий цикл спроб підключення до MQTT-брокера. На кожній ітерації генерується унікальний ідентифікатор клієнта у форматі «ESP32Client-XXXX», де XXXX – псевдовипадкове шістнадцяткове число. Це виключає конфлікт сесій при перепідключенні після втрати зв'язку або при одночасній роботі кількох пристроїв з однаковою базовою назвою. Автентифікація виконується передачею рядків mqtt_user та mqtt_pass у метод *mqttClient.connect()*. При невдалій спробі стан помилки виводиться в Serial-монітор (від'ємне число відповідає константам стану бібліотеки PubSubClient), після чого програма очікує 5 секунд перед наступною ітерацією. Лістинг функції *reconnectMQTT()* на рис. 3.8:

```
void reconnectMQTT() {
    while (!mqttClient.connected()) {
        Serial.print("Спроба підключення до MQTT...");
        String clientId = "ESP32Client-";
        clientId += String(random(0xffff), HEX);

        if (mqttClient.connect(clientId.c_str(), mqtt_user, mqtt_pass)) {
            Serial.println(" підключено успішно!");
        } else {
            Serial.print(" помилка, стан=");
            Serial.print(mqttClient.state());
            Serial.println(" Спробуємо знову через 5 секунд...");
            delay(5000);
        }
    }
}
```

Рисунок 3.8 – Лістинг коду

Підсистема візуалізації та моніторингу реалізована в середовищі Node-RED – відкритій платформі потокового програмування на основі Node.js, що підтримує концепцію low-code розробки IoT-додатків. Основним поняттям Node-RED є flow (потік) – направлений граф, у якому вузли (nodes) з'єднані дротами (wires) та передають між собою об'єкти повідомлень (msg). Кожна нода отримує вхідне повідомлення, виконує над ним певну операцію та передає результат на виходи. Побудований flow наведено на рис. 3.9.

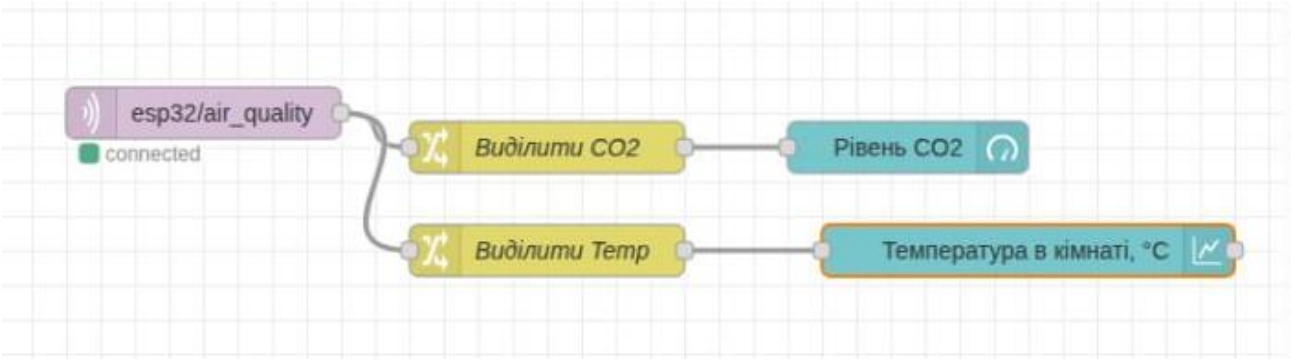

Рисунок 3.9 – Загальна структура Flow 1 у Node-RED

Flow складається з п'яти нод, з'єднаних у дві паралельні гілки обробки даних – для CO_2 та для температури відповідно. Повний перелік нод з описом їх функцій наведено в табл. 3.5.

Таблиця 3.5 – Ноди Flow 1 у Node-RED та їх призначення

№	Нода	Функція у flow
1	mqtt in	Підписка на MQTT-топік esp32/air_quality, отримання JSON-повідомлень від ESP32; вихід – розпарсований JS-об'єкт
2	change (Виділити CO2)	Перезапис msg.payload значенням msg.payload.co2 для подальшої передачі на gauge-ноду
3	change (Виділити Temp)	Перезапис msg.payload значенням msg.payload.temp для подальшої передачі на chart-ноду
4	ui_gauge (Рівень CO2)	Відображення поточного рівня CO_2 у вигляді стрілочного індикатора з кольоровим градієнтом (зелений/жовтий/червоний), діапазон 400–2000 ppm
5	ui_chart (Температура)	Відображення часового ряду температури у вигляді лінійного графіка з вікном відображення 1 година, вісь Y: 0–50 °C

Вхідна нода типу mqtt in є єдиною точкою надходження даних у flow. У конфігурації ноди вказано адресу локального брокера (у початковій реалізації – 127.0.0.1:1883; у фінальній – адреса HiveMQ Cloud, порт 8883), топік esp32/air_quality, рівень QoS 2 та режим виводу «a parsed JSON object». Завдяки режиму автоматичного парсингу, бібліотека Node-RED перетворює вхідний JSON-рядок у JavaScript-об'єкт ще до передачі в наступну ноду, тобто на вході обох гілок вже доступні поля msg.payload.co2 та msg.payload.temp без необхідності ручного

виклику *JSON.parse()*. Статус «connected» під нодою підтверджує успішне встановлення підписки на топік, що відображається зеленим індикатором на канві.

Розгалуження потоку на дві незалежні гілки реалізовано прямим з'єднанням виходу mqtt in одночасно до обох нод типу change. Нода «Виділити CO2» містить одне правило: Set msg.payload → msg.payload.co2. Після виконання цього правила об'єкт повідомлення містить у полі payload лише числове значення концентрації CO_2 у ppm, а вся інша частина вихідного JSON-об'єкта відкидається. Аналогічно нода «Виділити Temp» встановлює msg.payload у значення msg.payload.temp. Застосування нод change замість нод function є архітектурним рішенням, що відповідає принципу мінімальної складності: для простих операцій перетворення повідомлень node-type change є декларативним та не потребує написання JavaScript-коду.

Перша гілка завершується нодою ui_gauge з налаштуваннями, наведеними на рис. 3.10. Нода відображає поточне числове значення у вигляді стрілочного напівкругового індикатора типу Gauge. Мітка – «Рівень CO2», одиниці вимірювання – ppm, діапазон шкали – від 400 до 2 000 ppm (відповідає робочому діапазону сенсора MH-Z19E).

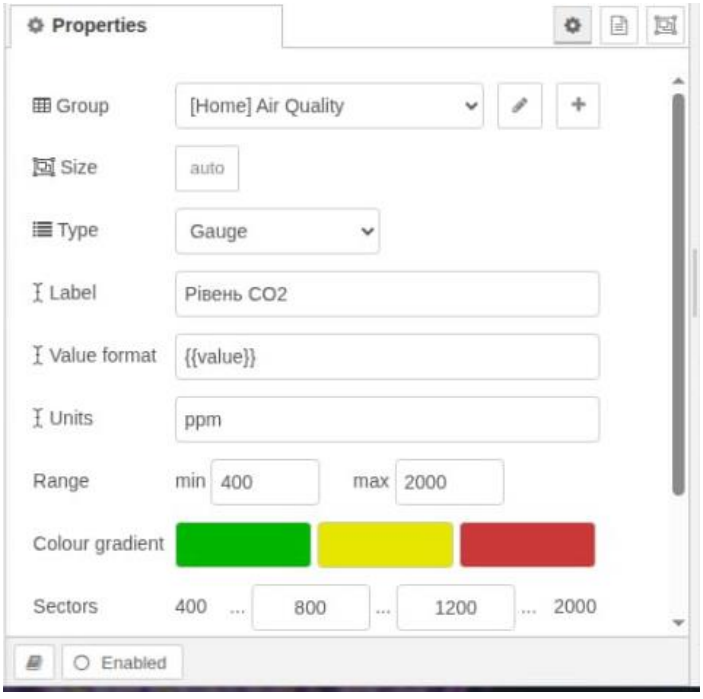

Рисунок 3.10 – Налаштування ноди ui_gauge «Рівень CO2»

Кольоровий градієнт шкали поділений на три сектори з граничними значеннями 400, 800, 1 200 та 2 000 ppm, кольори яких – зелений, жовтий та червоний відповідно – узгоджені з логікою індикації WS2812B на апаратному рівні, що забезпечує єдину колірну семантику в усіх компонентах системи.

Друга гілка завершується нодою ui_chart з налаштуваннями, наведеними на рис. 3.11. Нода відображає часовий ряд температури у вигляді лінійного графіка (Line chart) з такими параметрами: вісь X – остання 1 година в форматі HH:mm:ss; вісь Y – від 0 до 50 °C; інтерполяція – лінійна. Вікно відображення в 1 годину дозволяє спостерігати динаміку температурних змін у приміщенні протягом репрезентативного часового відрізку без ризику переповнення буфера точок.

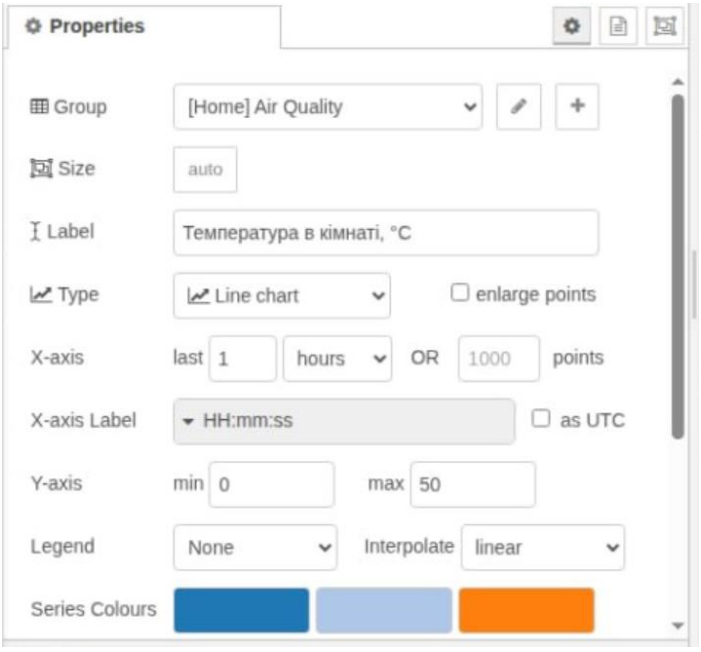

Рисунок 3.11 – Налаштування ноди ui_chart «Температура в кімнаті, °C»

Обидві UI-ноди розміщені в групі «Air Quality» на сторінці «Home» дашборду Node-RED Dashboard (модуль node-red-dashboard). Дашборд доступний за адресою http://localhost:1880/ui та відображає актуальні показники в режимі реального часу. Зовнішній вигляд дашборду в момент вимірювання концентрації CO_2 на рівні 1 889 ppm наведено на рис. 3.12: стрілка gauge-ноди відхилена до червоної зони шкали, лінійний графік температури демонструє стабільне значення 25 °C протягом години спостереження.

Рисунок 3.12 – Дашборд Node-RED у режимі реального часу (CO_2 = 1889 ppm, T = 25 °C)

Таким чином, програмна реалізація ІоТ-пристрою охоплює повний ланцюг перетворення фізичного сигналу в інформаційне відображення: від циклічного опитування сенсора MH-Z19E через апаратний UART2 та формування JSON-телеметрії – до захищеної публікації в хмарний MQTT-брокер та відображення даних у реальному часі на дашборді Node-RED. Архітектурні рішення, прийняті при розробці програми (таймерний шаблон замість delay, апаратний UART замість SoftwareSerial, автогенерація унікального Client ID, режим setInsecure для TLS), забезпечують стабільну та передбачувану роботу пристрою в умовах одночасної активності мережевого стеку Wi-Fi та завдань реального часу FreeRTOS.

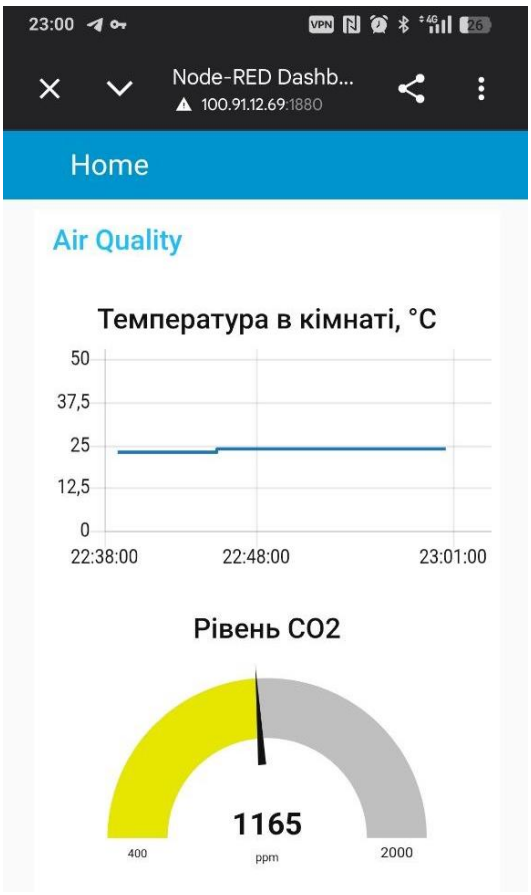

Рисунок 3.13 – Відображення панелі керування на мобільному пристрої через зашифровану мережу Tailscale (підключення через 4G/LTE)

Працездатність системи та адаптивність графічного інтерфейсу були додатково протестовані за допомогою мобільного пристрою в мережі стільникового зв'язку (4G/LTE). Як видно з рисунка 3.13, завдяки використанню тунелю Tailscale доступ до локального сервера (за IP-адресою 100.X.X.X) здійснюється безперебійно, а UI-компоненти автоматично масштабуються під вертикальну орієнтацію екрана смартфона, зберігаючи повну функціональність моніторингу в режимі реального часу.

3.3 Інтеграційне тестування та налагодження роботи системи

Метою етапу тестування є перевірка працездатності розробленого апаратно-програмного комплексу, оцінка його стабільності в реальних умовах експлуатації та виявлення можливих вразливостей на етапі мережевої взаємодії. Для досягнення цієї мети було обрано метод інтеграційного тестування (Integration Testing), який

дозволяє перевірити коректність обміну даними між апаратним вузлом, хмарним сервером та локальним інтерфейсом візуалізації.

Під час проведення експериментальних досліджень було розроблено та виконано кілька тестових сценаріїв, результати яких наведено нижче.

Сценарій 1: Тестування захищеного з'єднання між локальним сервером та хмарним брокером. Під час первинного налаштування маршрутизації телеметрії від Node-RED до хмарного брокера HiveMQ Cloud було зафіксовано явище примусового відхилення запитів з боку сервера. Аналіз системних логів (помилка Connection failed to broker) показав, що політика безпеки Serverless-брокера вимагає обов'язкового використання протоколу TLS із передачею імені сервера (SNI). Для усунення цього недоліку було проведено додаткове налагодження конфігурації mqtt-broker node шляхом розгортання профілю tls-config (рис. 3.14).

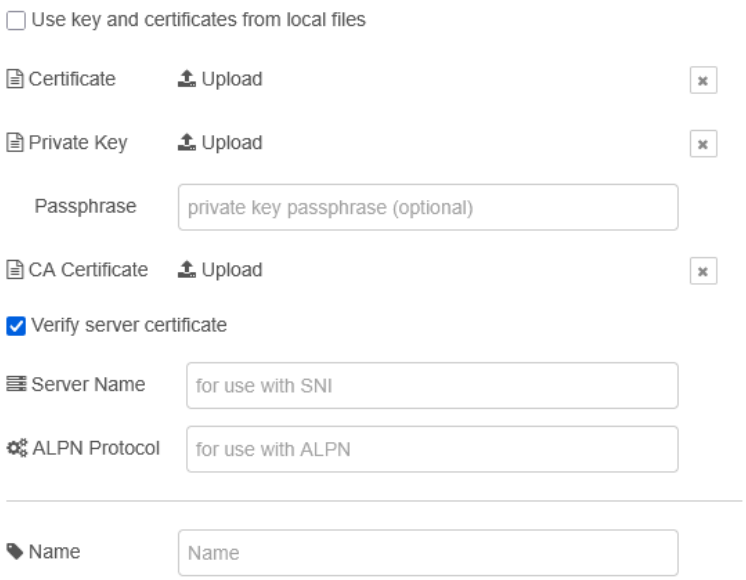

Рисунок 3.14 – Налаштування профілю TLS для маршрутизації SNI у середовищі Node-RED

Впровадження даного конфігуратора дозволило системі успішно використовувати кореневі сертифікати операційної системи для валідації хмарного брокера без ручного встановлення ключів, що забезпечило стабільний статус підключення (connected).

Сценарій 2: Стрес-тестування стабільності MQTT-з'єднання мікроконтролера. Другий етап тестування передбачав перевірку механізму автовідновлення з'єднання пристрою з інтернетом. Було виявлено ситуацію, коли ESP32 повідомляв про успішну відправку пакетів, однак вони не надходили до брокера. Діагностика виявила конфлікт статичних ідентифікаторів клієнта (Client ID), що призводило до розірвання сесій. Проблему було успішно вирішено шляхом оптимізації функції *reconnectMQTT()*, до якої додано алгоритм генерації унікального псевдовипадкового Client ID та обов'язкову авторизацію за допомогою пари логін/пароль (рис. 3.15).

```
Wi-Fi підключено!
Спроба підключення до MQTT... Підключено до HiveMQ!
Дані відправлено: {"co2":864,"temp":23}
Дані відправлено: {"co2":864,"temp":23}
Дані відправлено: {"co2":863,"temp":23}
Дані відправлено: {"co2":863,"temp":23}
Дані відправлено: {"co2":863,"temp":23}
Дані відправлено: {"co2":862,"temp":23}
```

Рисунок 3.15 – Результат успішного відновлення MQTT-з'єднання та публікації пакетів у консолі мікроконтролера

Сценарій 3: Тестування віддаленого доступу та кросплатформності інтерфейсу. Оскільки користувач повинен мати можливість контролювати мікроклімат поза межами локальної мережі, було проведено тестування доступу до панелі керування через мобільну мережу стільникового зв'язку (4G/LTE). Через дію NAT-трансляції провайдера прямий доступ до локального сервера виявився неможливим. Архітектурним рішенням цієї проблеми стало розгортання віртуальної приватної мережі (Mesh VPN) на базі Tailscale [20]. Це дозволило створити захищений тунель із фіксованою внутрішньою IP-адресою (формату 100.X.X.X).

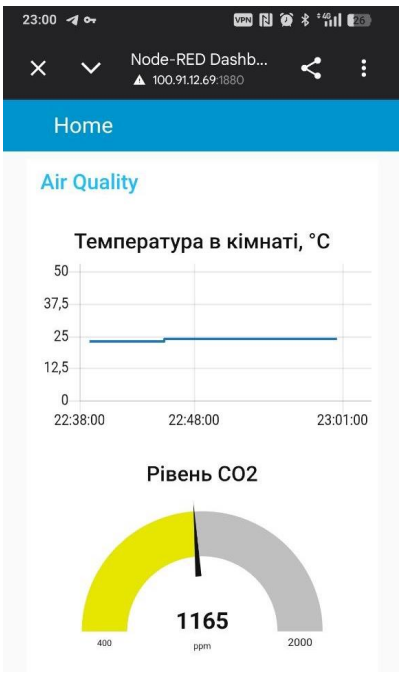

Рисунок 3.16 – Адаптивний інтерфейс панелі керування на мобільному пристрої під час доступу через мережу Tailscale

Аналіз результатів тестування (рис. 3.16) підтверджує високу якість розробленого програмного забезпечення: UI-компоненти автоматично масштабуються під вертикальну орієнтацію екрана смартфона, затримка відображення телеметрії не перевищує 1–2 секунд, а з'єднання залишається стабільним навіть при перемиканні між базовими станціями мобільного оператора. Усі поставлені завдання тестування виконано в повному обсязі.

3.4 Настанова користувача

Розроблений апаратно-програмний комплекс спроєктовано з урахуванням принципу «Plug-and-Play» (вмикай і працюй), що мінімізує необхідність складних налаштувань з боку кінцевого користувача. Порядок роботи із системою складається з наступних кроків.

Першим кроком буде підключення живлення та ініціалізація. Апаратний вузол необхідно розмістити в приміщенні на висоті 1,0–1,5 метра від підлоги (подалі від прямих протягів та джерел дихання). Живлення подається шляхом підключення кабелю USB (Micro-USB) від плати ESP32 до будь-якого стандартного зарядного пристрою (5 В / 1 А) або портативного акумулятора. Після

подачі живлення пристрій автоматично підключається до збереженої мережі Wi-Fi. Приклад підключення апаратного модуля до джерела живлення зображено на рис. 3.17:

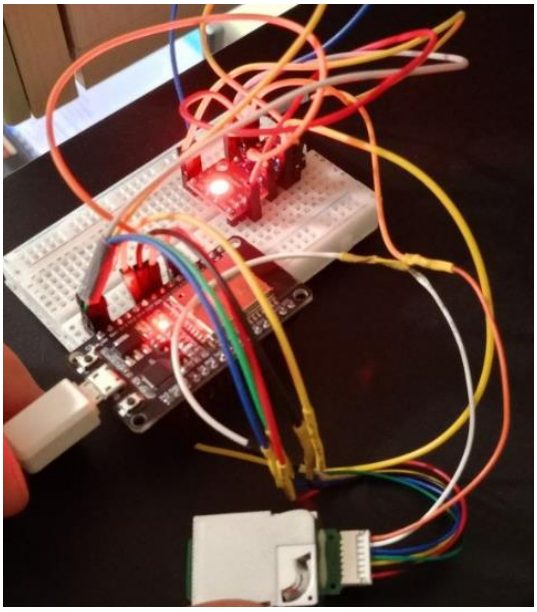

Рисунок 3.17 – Підключення апаратного модуля до джерела живлення

Переходимо до другого кроку, а саме візуального моніторингу інформації на пристрої. Користувач може оцінити якість повітря без використання додаткових пристроїв завдяки інтегрованому світлодіоду на платі:

– зелений колір: концентрація CO_2 в нормі (до 800 ppm), дії не потрібні;

– жовтогарячий колір: підвищена концентрація (800–1199 ppm), рекомендується підготуватися до провітрювання;

– червоний колір: критичний рівень (понад 1200 ppm), необхідно негайно відкрити вікно або увімкнути систему вентиляції.

Останнім кроком є доступ до панелі керування (Дашборду). Для отримання детальної аналітики та перегляду графіка температури користувачеві необхідно:

1) переконатися, що додаток Tailscale увімкнений на пристрої користувача як на рис. 3.18 (ПК або смартфоні);

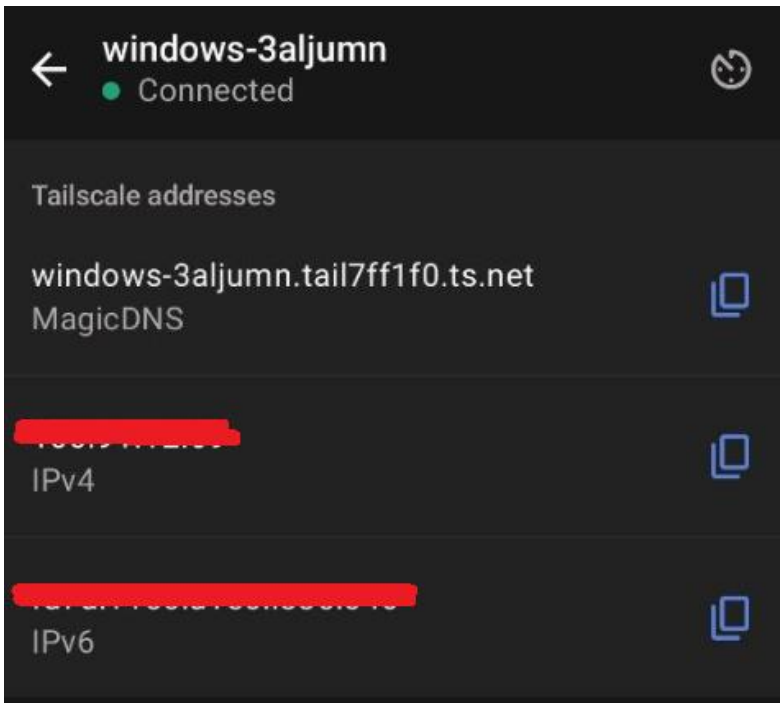

Рисунок 3.18 – Підключення пристроїв за допомогою Tailscale

2) відкрити веббраузер та ввести закріплену IP-адресу сервера у форматі http://100.X.X.X:1880/ui.

Рисунок 3.19 – Головне вікно панелі моніторингу якості повітря в браузері користувача

У головному вікні (рис. 3.19) стрілочний індикатор у режимі реального часу відображає поточну концентрацію вуглекислого газу. Нижче розташований лінійний графік, який дозволяє відстежити динаміку зміни кімнатної температури за останню годину. Дані оновлюються автоматично кожні 3 секунди без необхідності ручного перезавантаження сторінки. Відповідно до поставленої мети та завдань, отримано такі основні результати:

Висновки до розділу 3

У третьому розділі виконано практичну реалізацію та експериментальне дослідження розробленої апаратно-програмної платформи для моніторингу концентрації вуглекислого газу. На основі теоретичних відомостей та спроєктованої архітектури було успішно сконструйовано діючий прототип IoT-пристрою, який поєднує сучасні сенсорні технології, мікроконтролерну обробку та хмарні сервіси обміну даними.

Апаратну частину комплексу реалізовано на базі мікроконтролера ESP32 та оптичного NDIR-сенсора MH-Z19E. Важливим етапом конструювання стало вирішення проблеми сумісності логічних рівнів (3,3 В та 5 В) за допомогою двонаправленого перетворювача AOC537, що забезпечило стабільну та апаратно безпечну комунікацію через інтерфейс UART. Крім того, систему оснащено адресним RGB-світлодіодом для локальної колірної індикації стану мікроклімату, що дозволяє користувачеві миттєво оцінити якість повітря без необхідності звернення до вебінтерфейсу.

Програмне забезпечення мікроконтролера розроблено з урахуванням принципів енергоефективності та неблокуючої архітектури (на базі системного таймера). Впроваджено алгоритм цифрової фільтрації даних та програмний гістерезис, що дозволило мінімізувати вплив апаратного шуму сенсора та забезпечити плавну зміну станів індикації. Налаштовано надійний канал передачі телеметрії формату JSON до хмарного Serverless-брокера HiveMQ Cloud з обов'язковим використанням протоколу шифрування транспортного рівня (TLS) та механізму маршрутизації SNI.

Для візуалізації зібраних даних розроблено кросплатформну панель керування (дашборд) у середовищі локального сервера Node-RED. Інтерфейс включає стрілочний індикатор для відображення поточного рівня CO_2 в режимі реального часу та лінійний графік для аналізу температурних динамік. Налаштування профілю безпеки tls-config дозволило системі автоматично підтверджувати сертифікати хмарного брокера, гарантуючи надійне та захищене отримання телеметрії.

Проведене інтеграційне тестування підтвердило високу надійність та відмовостійкість розробленої системи в реальних умовах експлуатації. Виявлені під час стрес-тестування проблеми з розривом MQTT-сесій були успішно усунені шляхом оптимізації алгоритму підключення мікроконтролера та генерації динамічних ідентифікаторів. Крім того, завдяки впровадженню технології віртуальної приватної мережі (Mesh VPN) Tailscale, вдалося успішно подолати обмеження NAT-трансляції мобільних провайдерів. Це забезпечило користувачеві безпечний віддалений доступ до панелі керування з будь-якої точки світу через мережі 4G/LTE без необхідності конфігурації домашнього маршрутизатора.

Отже, результати експериментальних досліджень та тестування доводять, що розроблений апаратно-програмний комплекс функціонує коректно, відрізняється високою стабільністю, відповідає концепції «Plug-and-Play» і повністю задовольняє технічні вимоги, поставлені до сучасних систем моніторингу якості повітря у приміщеннях.

ВИСНОВКИ

У кваліфікаційній бакалаврській роботі вирішено мету роботи – розроблено програмно-апаратну платформу (ПАП) для безперервного моніторингу та візуалізації вмісту вуглекислого газу в закритих приміщеннях із використанням технологій Інтернету речей (IoT). Мету досягнуто за допомогою вирішення поставлениз задач:

1) проаналізовано нормативну базу та вплив CO_2 на здоров'я: досліджено державні будівельні норми та санітарні стандарти (зокрема ДБН В.2.5-67:2013 та ДСТУ EN 16798-1:2019). Встановлено, що концентрація CO_2 є універсальним індикатором якості повітря, а цільовий поріг для систем автоматичного керування вентиляцією має становити близько 800 ppm, оскільки перевищення рівня 1000–1200 ppm призводить до зниження концентрації уваги та симптомів гіперкапнії;

2) проведено аналіз IoT-технологій та протоколів: доведено перевагу використання протоколу MQTT над традиційним HTTP завдяки архітектурі «видавець-підписник», мінімальному розміру заголовків та зниженому навантаженню на канали зв'язку. Обґрунтовано застосування формату JSON для уніфікації обміну телеметричними даними та їх подальшої інтеграції в платформи домашньої автоматизації (Node-RED, Home Assistant);

3) обґрунтовано вибір апаратної бази: з-поміж наявних на ринку мікроконтролерів (ESP8266, ESP32-S3, Raspberry Pi Pico W) обрано SoC ESP32 завдяки наявності апаратного UART, вбудованого WiFi-модуля та двоядерної архітектури. Як вимірювальний елемент обрано оптичний NDIR-сенсор MH-Z19E, який, на відміну від дешевих напівпровідникових аналогів, має високу селективність до CO_2, апаратну термокомпенсацію та заводське калібрування;

4) спроєктовано архітектуру та схемотехніку пристрою: розроблено трирівневу архітектуру системи (рівень збору, рівень передачі, рівень візуалізації). Успішно вирішено проблему несумісності логічних рівнів між сенсором (5 В) та мікроконтролером (3,3 В) шляхом впровадження у схему двонаправленого перетворювача рівнів AOC537, що гарантувало електричну безпеку портів ESP32;

5) реалізовано програмне забезпечення та алгоритми обробки: у середовищі Arduino IDE написано неблокуючий (на базі функції *millis()*) програмний код для мікроконтролера. Впроваджено алгоритм цифрової фільтрації на базі експоненційного ковзного середнього (EMA, α = 0.2) та механізм програмного гістерезису, що дозволило усунути апаратний шум сенсора та запобігти хаотичному блиманню RGB-світлодіода (NeoPixel) на межах порогових значень;

6) проведено інтеграційне тестування та інтеграцію: розроблена система успішно пройшла експериментальну перевірку. Налагоджено безпечну передачу даних (TLS/SNI) до хмарного брокера HiveMQ Cloud. Розроблено інформативний дашборд у середовищі Node-RED, а завдяки розгортанню віртуальної приватної мережі (Mesh VPN) Tailscale забезпечено безпечний, кросплатформний та віддалений доступ до системи керування через мобільні мережі 4G/LTE без необхідності відкриття портів на маршрутизаторі.

У підсумку, розроблений прототип IoT-пристрою є повністю автономним, енергоефективним та відповідає концепції «Plug-and-Play». Отримані результати мають високу практичну цінність і повністю придатні для впровадження у житлових будинках, офісах, навчальних аудиторіях та укриттях цивільного захисту як самостійне рішення або як інтегрований вузол екосистеми «Розумного будинку». Усі завдання, поставлені у кваліфікаційній роботі, виконано в повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Mota A., Serodio C., Briga Sá A., Valente A. Next-generation smart homes: CO2 monitoring with Matter protocol to support indoor air quality. *Internet of Things*. 2025. Vol. 32. P. 1–20. DOI: 10.1016/j.iot.2025.101649.

2. Mota A., Serôdio C., Briga-Sá A., Valente A. Implementation of an Internet of Things Architecture to Monitor Indoor Air Quality: A Case Study During Sleep Periods. *Sensors*-2025. Vol. 25. №. 6:1683. P. 1–21. DOI:10.3390/s25061683.

3. Afrial M., Rauf M., Nouman M., Khan M. T., Rizwan M. A., Ahmad N. Multi-Sensor Indoor Air Quality Monitoring with Real-Time Logging and Air Purifier Integration. *Materials Proceedings*. 2025. Vol. 23. №. 1:12. P. 1–6. DOI:10.3390/materproc2025023012.

4. Ramamoorthy D., Divya R., Radjarejesri S., Sanchita S. IoT Based Multi-Parameter Environmental Monitoring System using ESP32 for Real-Time Safety and Air Quality Assessment. *2025 Second International Conference on Intelligent Technologies for Sustainable Electric and Communications Systems (iTech SECOM)*. P. 1-6. 2025. DOI: 10.1109/iTechSECOM64750.2025.11307644.

5. Pranikaa K., Gayathri M. 2025. IoT-based ESP32 System for Monitoring Hazardous Gases and Particulate Matter. *2025 3rd International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. P. 1049-1053. 2025. DOI: 10.1109/ICSCDS65426.2025.11167473.

6. ДБН В.2.5-67: 2013. Опалення, вентиляція та кондиціонування: Чинний від 1 січня 2014 р. URL: https://e-construction.gov.ua/laws_detail/3074971619479783152 (Last accessed: 02.06.2026).

7. ДСТУ EN 16798-1: 2019. Енергоефективність будівель. Вентиляція будівель. Частина 1. Вхідні параметри внутрішнього середовища: Чинний від 1 лютого 2020 р. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=109961 (Last accessed: 02.06.2026).

8. ДБН В.2.2-5: 2023. Захисні споруди цивільного захисту: Чинний від 1 листопада 2023 р. URL: https://e-

construction.gov.ua/laws_detail/3225773063500990463?doc_type=2 (Last accessed: 02.06.2026).

9. НД ТЗІ 3.7-003-05. Побудова комплексів технічного захисту мовної інформації на об'єктах ЕОТ: Чинний від 16 травня 2005 р. URL: https://tzi.com.ua/downloads/3.7-003-2005.pdf (Last accessed: 02.06.2026).

10. MH-Z19E. Intelligent Infrared CO2 Sensor. User's Manual. URL: https://www.winsen-sensor.com/d/files/mh-z19e-co2-sensor-manual-v1_0.pdf (Last accessed: 07.05.2026).

11. ESP32 Technical Reference Manual. Espressif Systems. URL: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf (Last accessed: 08.05.2026).

12. PlantText. URL: https://www.planttext.com (Last accessed: 08.05.2026).

13. Волочай П. О., Салтовський Б.Г. Іот–пристрій для контролю вмісту вуглекислого газу у приміщенні. *Інтелектуальні інформаційні системи-2025* : матеріали Всеукр. наук.-практ. конф. молодих вчених, аспірантів, студентів, Миколаїв, 4–5 груд. 2025 р. С. 145–148. URL: https://dspace.chmnu.edu.ua/jspui/handle/123456789/3054

14. ESP32 Arduino Core’s documentation. URL: https://docs.espressif.com/projects/arduino-esp32/en/latest/ (Last accessed: 09.05.2026).

15. MQTT Version 3.1.1. OASIS Standard. URL: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html (Last accessed: 09.05.2026).

16. Node-RED: Low-code programming for event-driven applications / OpenJS Foundation. URL: https://nodered.org/docs/ (Last accessed: 09.05.2026).

17. Node-red-dashboard. URL: https://flows.nodered.org/node/node-red-dashboard (Last accessed: 10.05.2026).

18. ESP32 Series Datasheet. Espressif Systems. 2024. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (Last accessed: 10.05.2026).

19. MQTT Version 3.1.1. OASIS Standard. 2014. URL: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html (Last accessed: 11.05.2026).

20. Donenfeld J. A. WireGuard: Next Generation Kernel Network Tunnel. *NDSS Symposium*. San Diego, CA, 2017. URL: https://www.wireguard.com/papers/wireguard.pdf (Last accessed: 11.05.2026).

21. Satish U., Mendell MJ, Shekhar K., Hotchi T., Sullivan D., Streufert S., Fisk WJ. Is CO2 an indoor pollutant? Direct effects of low-to-moderate CO2 concentrations on human decision-making performance. *Environ Health Perspect*. Dec. 2012 P. 120-125. DOI: 10.1289/ehp.1104789.

22. IQAir AirVisual Pro Air Quality Monitor. URL: https://allergycosmos.co.uk/products/airvisual-air-quality-monitor (Last accessed: 08.05.2026).

23. Robotbanao 5V 1 Channel Relay Module for ESP8266 ESP-01 Development Board. URL: https://www.amazon.in/Robotbanao-Channel-Module-ESP8266-Development/dp/B08T79BRL9 Last accessed: 09.05.2026).

24. Одноплатний комп'ютер Raspberry Pi 4 Model B 4GB. URL: https://elmir.ua/ua/minikompyutery/odnoplatnyy_kompyuter_raspberry_pi_4_model_b_4gb_rpi4-modbp-4gb.html (Last accessed: 09.05.2026).

25. Одноплатний комп'ютер Orange Pi Zero 512MB. URL: https://prom.ua/p1464357809-odnoplatnyj-kompyuter-orange.html (Last accessed: 09.05.2026).

26. ESP32-S3-DevKitC-1 Microcontroller Development Board. URL: https://www.amazon.in/Microcontroller-Development-Pre-Soldered-ESP32-S3-WROOM-1-N8R8-MicroPython/dp/B0C28CG44J (Last accessed: 09.05.2026).

ДОДАТОК А

Лістинг коду

```
#include <WiFi.h>
#include <MHZ19.h>
#include <Adafruit_NeoPixel.h>
#include <PubSubClient.h>
#include <WiFiClientSecure.h>

const char* ssid = "volochai_tanya";
const char* password = "93636hK4";

const char* mqtt_server = "b2877cb5627346a187607fd0122d3be4.s1.eu.hivemq.cloud";
const int mqtt_port = 8883;
const char* mqtt_user = "esp32_user";
const char* mqtt_pass = "Esp32_user";
const char* mqtt_topic = "esp32/air_quality";

#define MHZ19_RX_PIN 26
#define MHZ19_TX_PIN 27
#define LED_PIN 25
#define NUMPIXELS 1

MHZ19 myMHZ19;
HardwareSerial mySerial2(2);
Adafruit_NeoPixel pixels(NUMPIXELS, LED_PIN, NEO_GRB + NEO_KHZ800);

WiFiClientSecure espClient;  // <-- TLS клієнт
PubSubClient mqttClient(espClient);

int global_co2 = 0;
int global_temp = 0;

void setLED(int r, int g, int b) {
    pixels.setPixelColor(0, pixels.Color(r, g, b));
    pixels.show();
}

void reconnect() {
  while (!mqttClient.connected()) {
    Serial.print("Спроба підключення до MQTT...");

    String clientId = "ESP32Client-";
    clientId += String(random(0xffff), HEX);

    if (mqttClient.connect(clientId.c_str(), mqtt_user, mqtt_pass)) {
      Serial.println(" Підключено до HiveMQ!");
    } else {
      Serial.print(" Помилка, rc=");
      Serial.print(mqttClient.state());
      Serial.println(". Спробуємо знову через 5 секунд");
      delay(5000);
    }
  }
}

void setup() {
    Serial.begin(115200);

    mySerial2.begin(9600, SERIAL_8N1, MHZ19_RX_PIN, MHZ19_TX_PIN);
```

```
    myMHZ19.begin(mySerial2);

    pixels.begin();
    pixels.setBrightness(50);
    pixels.show();

    WiFi.begin(ssid, password);
    Serial.print("Підключення до Wi-Fi");
    while (WiFi.status() != WL_CONNECTED) {
        delay(300);
        Serial.print(".");
    }
    Serial.println("\nWi-Fi підключено!");

    espClient.setInsecure();  // <-- без перевірки сертифікату, простіше для ESP32
    mqttClient.setServer(mqtt_server, mqtt_port);
}

void loop() {
    if (!mqttClient.connected()) {
        reconnect();
    }
    mqttClient.loop();

    static unsigned long lastSensorTick = 0;

    if (millis() - lastSensorTick >= 3000) {
        lastSensorTick = millis();

        int current_co2 = -1;
        int current_temp = -1;

        unsigned long t = millis();
        while (millis() - t < 200) {
            current_co2 = myMHZ19.getCO2();
            current_temp = myMHZ19.getTemperature();
            if (current_co2 > 0) break;
            yield();
        }

        if (current_co2 > 0) {
            global_co2 = current_co2;
            global_temp = current_temp;

            if (global_co2 < 800) setLED(0, 255, 0);
            else if (global_co2 < 1200) setLED(255, 180, 0);
            else setLED(255, 0, 0);

                 String payload = "{\"co2\":" + String(global_co2) + ",\"temp\":" +
String(global_temp) + "}";

            if (mqttClient.publish(mqtt_topic, payload.c_str())) {
                Serial.print("Дані відправлено: ");
                Serial.println(payload);
            } else {
                Serial.println("Помилка відправки");
            }
        }
    }

    delay(2);
}
```

ДОДАТОК Б

Тези апробації

УДК 004.942:621.865

Волочай П. О.,Салтовський Б. Г.
Чорноморський національний університет
ім. Петра Могили, м. Миколаїв, Україна

ІОТ-ПРИСТРІЙ ДЛЯ КОНТРОЛЮ ВМІСТУ ВУГЛЕКИСЛОГО ГАЗУ В ПРИМІЩЕННІ

Розробка автономного модуля моніторингу концентрації вуглекислого газу (CO2) є актуальним завданням для забезпечення санітарно-гігієнічних норм та попередження негативного впливу гіперкапнії на здоров’я людини. Проект пропонує архітектуру IoT-пристрою, що поєднує функцію локальної візуалізації та мережевої взаємодії. Алгоритм роботи передбачає дискретну світлову індикацію стану повітря (зелений, жовтий, червоний) та передачу телеметричних даних через протокол MQTT. Це забезпечує безшовну інтеграцію в екосистему «Розумний будинок» та дозволяє налаштовувати сценарії автоматичного реагування, зокрема активацію примусової вентиляції. Запропоноване адаптивне рішення підвищує енергоефективність керування мікрокліматом та забезпечує оперативність реагування на критичні зміни складу повітря.

145

Актуальність дослідження. Створення автономних систем моніторингу мікроклімату є ключовим напрямом розвитку концепції IoT та «Розумного будинку». Актуальність зумовлена необхідністю забезпечення санітарних норм у сучасних енергоефективних будівлях, де підвищення рівня CO2 негативно впливає на здоров'я. Інтеграція вимірювальних засобів у єдину мережу стимулює розробку адаптивних систем енергозбереження через керування вентиляцією за вимогою (Demand-Controlled Ventilation). Це мінімізує експлуатаційні витрати та підвищує ресурс обладнання. Найважливішим аспектом є поєднання точних сенсорних технологій із мережевим протоколом MQTT, що забезпечує комплексну автоматизацію процесів життєзабезпечення.

Методи вирішення проблеми. Розробка апаратно-програмного комплексу базується на модульній архітектурі IoT-систем з чітким розмежуванням рівнів збору, обробки та мережевої передачі даних. Ключовим елементом вимірювального тракту обрано модуль MH-Z19B (рис. 1-б), принцип дії якого грунтується на технології недисперсійного інфрачервоного випромінювання (NDIR) та законі Ламберта-Бера. Вибір сенсора зумовлений високою селективністю до молекул CO2 та стабільністю показників протягом тривалого терміну експлуатації. Наявність вбудованої температурної компенсації нівелює похибку вимірювань, підвищуючи точність моніторингу мікроклімату.

Для інтеграції сенсора з обчислювальним ядром на базі мікроконтролера Arduino розроблено спеціалізовану схему узгодження логічних рівнів (рис. 1-а). Оскільки модуль MH-Z19B оперує логікою 3.3В, а керуючий контролер — TTL-логікою 5В, пряме з’єднання ліній UART (TX/RX) є неприпустимим через ризик деградації вхідних каскадів сенсора. Реалізована схема використовує резистивні подільники напруги: послідовні резистори R1,R2 (470 Ом) обмежують струм у лінії, а паралельні резистори R3,R4 (1 кОм) підтягують потенціал до «землі», формуючи на вході сенсора безпечну напругу рівня ≈3.4В ($V_{in} * \frac{R_{GND}}{R_{seri} + R_{GND}}$). Таке схемотехнічне рішення не лише захищає компоненти, але й підвищує завадостійкість каналу передачі даних в умовах можливих електромагнітних наведень від силових кабелів побутової мережі.

В якості обчислювального ядра обрано мікроконтролерну платформу ESP32 на базі SoC. Вибір архітектури зумовлений наявністю двох високопродуктивних ядер Xtensa® 32-bit LX6 та інтегрованого Wi-Fi радіомодуля. Це дозволяє реалізувати асинхронну передачу телеметричних даних по протоколу MQTT без затримок у основному циклі опитування сенсорів. Додатково, нативна робота логіки