

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет робототехніки та приладобудування
Кафедра комп'ютерно-інтегрованих оптичних та навігаційних систем

До захисту допущено:
Завідувач кафедри
_____ Надія БУРАУ
«___» _____ 2026 р.

Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Комп'ютерно-інтегровані системи та
технології в приладобудуванні»
спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та
робототехніка»
на тему: «Автоматизована система моніторингу кліматичних показників»

Виконав:

студент III курсу, групи ПГ-ПЗ1

Мельник Максим Анатолійович _____

Керівник:

К.т.н., доцент Цибульник Сергій Олексійович _____

Рецензент:

К.т.н., доцент Безугла Наталя Василівна _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2026

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет робототехніки та приладобудування

Кафедра комп'ютерно-інтегрованих оптичних та навігаційних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

Освітньо-професійна програма «Комп'ютерно-інтегровані системи та технології в приладобудуванні»

ЗАТВЕРДЖУЮ

Завідувачка кафедри

_____ Надія БУРАУ

«___» _____ 2026 р.

ЗАВДАННЯ

на дипломну роботу студенту

Мельнику Максиму Анатолійовичу

1. Тема роботи «Автоматизована система моніторингу кліматичних показників», керівник роботи Цибульник Сергій Олексійович, к.т.н., доцент, затверджені наказом по університету від «___» _____ 20__ р. № _____

2. Термін подання студентом роботи 05.06.2026р.

3. Вихідні дані до роботи: точність визначення температури $\pm 2^{\circ}\text{C}$; точність визначення вологості $\pm 5\%$; мінімальна температура $+10^{\circ}\text{C}$; максимальна температура нагріву $+50^{\circ}\text{C}$; мінімальна вологість 20%; максимальна вологість 100%; частота оновлення даних графічного інтерфейсу користувача – 2 Гц.

4. Зміст роботи:

1) Огляд стану проблеми.

2) Огляд та вибір елементної бази.

- 3) Проектування елементів складальної одиниці.
 - 4) Розроблення графічного інтерфейсу користувача.
 - 5) Апаратне та програмне налагодження системи.
 - 6) Перевірка працездатності системи та аналіз результатів.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): таблиці, графіки, рисунки, тощо.
6. Дата видачі завдання 16.03.2026р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
	Огляд стану проблеми	23.03.2026	
	Огляд та вибір елементної бази	30.03.2026	
	Проектування елементів складальної одиниці	06.04.2026	
	Розроблення графічного інтерфейсу користувача	20.04.2026	
	Апаратне та програмне налагодження системи	18.05.2026	
	Перевірка працездатності системи та аналіз результатів	25.05.2026	
	Оформлення пояснювальної записки	05.06.2026	

Студент

Максим МЕЛЬНИК

Керівник

Сергій ЦИБУЛЬНИК

АНОТАЦІЯ

У кваліфікаційній роботі розглядається розроблення автоматизованої системи моніторингу кліматичних показників, спеціально адаптованої для використання у приміщеннях. Головною метою роботи є забезпечення безперервного та точного контролю за температурою та вологістю повітря, що є важливим фактором для підтримання належних умов зберігання майна та запобігання його псуванню через порушення режимів температури та вологості.

Апаратна частина базується на мікроконтролері ESP32. Вибір мікроконтролера зумовлений його високою обчислювальною потужністю та наявністю вбудованого модуля Wi-Fi, що дозволяє бездротову передачу показників без використання додаткових комунікаційних шляхів. Це забезпечує компактність, енергоефективність та надійність кінцевого вузла моніторингу.

Програмна складова проєкту охоплює написання мікропрограмного забезпечення для зчитування й первинної обробки сенсорних даних, а також організацію їх відправки на віддалений вебсервер. Для зручної взаємодії з системою реалізовано вебінтерфейс, який дозволяє користувачу дистанційно відстежувати стан мікроклімату у приміщенні шляхом наочної візуалізації отриманої інформації у вигляді динамічних графіків у режимі реального часу.

Запропоноване апаратно-програмне рішення відрізняється простотою розгортання, гнучкістю до масштабування та економічною доступністю, що робить його ефективним інструментом для автоматизації контролю клімату у приміщеннях.

Ключові слова: моніторинг мікроклімату, автоматизована система, ESP32, датчики температури та вологості, вебінтерфейс, інтернет речей (IoT).

ABSTRACT

This qualification paper focuses on the development of an automated climate monitoring system specifically adapted for indoor environments. The primary objective of this work is to ensure the continuous and accurate monitoring of air temperature and humidity. This is a crucial factor in maintaining proper storage conditions for assets and preventing property damage caused by temperature and humidity fluctuations.

The hardware architecture is based on the ESP32 microcontroller. The selection of this microcontroller is justified by its high computational performance and integrated Wi-Fi module, enabling the wireless transmission of readings without the need for additional communication interfaces. This ensures the compactness, energy efficiency, and reliability of the final monitoring node.

The software component of the project involves developing firmware for acquiring and pre-processing sensor data, as well as configuring its transmission to a remote web server. To facilitate user interaction with the system, a web interface has been implemented. This allows the user to remotely track the indoor microclimate through clear, real-time visual representation of the acquired data using dynamic charts.

The proposed hardware-software solution is characterized by its ease of deployment, scalability, and cost-effectiveness, making it a highly efficient tool for the automation of indoor climate control.

Keywords: microclimate monitoring, automated system, ESP32, temperature and humidity sensors, web interface, Internet of Things (IoT).

ПЕРЕЛІК СКОРОЧЕНЬ

АЦП - аналогово-цифровий перетворювач.

МК - мікроконтролер.

API (Application Programming Interface) - прикладний програмний інтерфейс.

BLE (Bluetooth Low Energy) - технологія бездротового зв'язку з низьким енергоспоживанням.

HTTP (HyperText Transfer Protocol) - протокол передачі гіпертексту.

I2C (Inter-Integrated Circuit) - послідовна симетрична шина даних для зв'язку між інтегральними схемами.

IoT (Internet of Things) - Інтернет речей.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 ОГЛЯД ІСНУЮЧИХ СИСТЕМ ТА ЗАСОБІВ МОНІТОРИНГУ КЛІМАТИЧНИХ ПОКАЗНИКІВ.....	11
1.1 Аналіз проблеми контролю мікроклімату у приміщеннях	11
1.2 Огляд комерційних систем моніторингу мікроклімату	13
1.2.1 Побутові екосистеми.....	13
1.2.2 Охоронно-промислові комплекси	15
1.2.3 Спеціалізовані інтелектуальні системи моніторингу	16
1.2.4 Порівняльний аналіз комерційних рішень.....	17
1.3 Огляд технологій бездротової передачі даних	19
1.4 Аналіз мікроконтролерів для побудови вузлів моніторингу.....	20
1.5 Огляд методів обробки та візуалізації даних у системах моніторингу	23
1.5.1 Методи попередньої обробки даних на стороні мікроконтролера.....	23
1.5.2 Аналіз мережевих протоколів для передачі в реальному часі	24
1.5.3 Технології візуалізації та побудови динамічних графіків	25
1.6 Обґрунтування та вибір елементної бази системи.....	26
1.6.1 Обґрунтування вибору мікроконтролера	26
1.6.2 Порівняльний аналіз та вибір первинних перетворювачів	27
1.6.3 Організація вузла живлення та виконавчих механізмів	30
РОЗДІЛ 2 РОЗРОБЛЕННЯ ТА ДОСЛІДЖЕННЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ СИСТЕМИ МОНІТОРИНГУ	34
2.1 Розроблення принципової електричної схеми пристрою.....	34
2.2 Розрахунок енергетичного балансу та теплових режимів системи.....	36
2.3 Конструктивне виконання та компонування пристрою	39
2.4 Технологія збірки, монтажу та просторової інтеграції компонентів	47
2.5 Розроблення графічного інтерфейсу користувача.....	50

2.5.1 Архітектура програмного забезпечення вебсервера.....	51
2.5.2 Аналіз протоколів передачі даних для	52
2.5.3 Структура та верстка графічного інтерфейсу.....	53
2.5.4 Візуалізація даних у реальному часі	58
2.6 Апаратне та програмне налагодження системи	59
2.6.1 Розроблення мікропрограмного забезпечення та алгоритму роботи	60
2.6.2 Налагодження апаратної частини та калібрування сенсорів	63
2.6.3 Програмне керування виконавчим механізмом.....	65
2.7 Експериментальне дослідження працездатності системи	68
2.7.1 Методика проведення експерименту.....	69
2.7.2 Аналіз точності кліматичних показників.....	72
2.7.3 Оцінка роботи вебінтерфейсу та виконавчих механізмів.....	74
Висновки до другого розділу	75
ВИСНОВКИ.....	76
ПЕРЕЛІК ПОСИЛАНЬ	78
ДОДАТОК А	82
ДОДАТОК Б.....	83

ВСТУП

У світі, де все більше проявляються ознаки зміни клімату, а особливо в Україні, моніторинг температури та вологості у приміщеннях стає надзвичайно важливим для збереження якості товарів, сировини, чутливого обладнання та комфорту людей.

Температура та вологість повітря є визначальними фізичними параметрами, відхилення яких від норми може призвести до псування продукції, розвитку цвілі, корозії металів або пошкодження електроніки, а традиційні методи контролю, що передбачають періодичну ручну перевірку показників, є неефективними [1], бо не здатні забезпечити безперервність спостереження та оперативність інформування про критичні зміни середовища.

Через це усе більше актуальності набувають автоматизовані системи моніторингу параметрів мікроклімату. Використання концепції Internet of Things (IoT) та бездротових технологій дозволяє не лише в реальному часі спостерігати за зміною параметрів середовища, а й опрацьовувати ці дані на віддалених серверах для подальшого аналізу. Для великих приміщень бездротові рішення є особливо вигідними, оскільки вони знімають потребу у прокладанні складних кабельних мереж.

Актуальність теми зумовлена необхідністю створення доступного, гнучкого та надійного інструменту для віддаленого моніторингу приміщень. Побудова системи на базі мікроконтролера ESP32, який має вбудований модуль Wi-Fi та достатню обчислювальну потужність, дозволяє реалізувати компактний вузол збору даних. Такий вузол здатен безперервно відправляти показники вологості та температури на вебсервер, де вони візуалізуються у вигляді динамічних графіків у режимі реального часу, забезпечуючи зручний контроль з будь-якого пристрою.

Метою дипломної роботи є розроблення та створення діючого прототипу автоматизованої системи моніторингу кліматичних показників для приміщень із можливістю віддаленого доступу до даних через вебінтерфейс.

Для досягнення поставленої мети в роботі передбачено вирішення наступних основних завдань: провести огляд стану проблеми, здійснити огляд та вибір елементної бази, виконати проєктування елементів складальної одиниці, здійснити розроблення графічного інтерфейсу користувача, провести апаратне та програмне налагодження системи, а також виконати перевірку працездатності системи та аналіз результатів.

РОЗДІЛ 1

ОГЛЯД ІСНУЮЧИХ СИСТЕМ ТА ЗАСОБІВ МОНІТОРИНГУ КЛІМАТИЧНИХ ПОКАЗНИКІВ

1.1 Аналіз проблеми контролю мікроклімату у приміщеннях

Мікроклімат у приміщеннях - це набагато більше, ніж просто питання комфорту. Від того, наскільки стабільні температура та вологість, напряду залежить здоров'я людини [2], продуктивність на роботі, а також збереження дорогоцінного обладнання і товарів. Усі ці показники тісно взаємопов'язані. В Україні базові санітарні норми регулюються стандартом ДСН 3.3.6.042-99, який задає чіткі межі для різних робочих зон [3]. Проте, як зазначають автори фахових підручників (наприклад, Б. М. Гончаренко у праці «Автоматизація виробничих процесів») [1], для надійного контролю цих параметрів на виробництвах чи складах самого лише існування норм недостатньо - потрібне впровадження ефективних технічних засобів та автоматизованих систем.

Проблема забезпечення стабільного мікроклімату є критично важливою для широкого спектра приміщень, що підтверджується низкою вітчизняних нормативних документів та публікацій.

У житлових та офісних зонах відхилення температури й вологості від оптимальних значень безпосередньо впливають на функціональний стан людей, причому медичні дослідження [2] демонструють, що такі коливання здатні спричиняти порушення координації, слабкість, головні болі та створювати сприятливе середовище для респіраторних інфекцій.

Для складських і логістичних центрів стабільність мікроклімату прямо визначає безпеку зберігання, адже згідно з чинними державними будівельними нормами, недотримання жорстких температурно-вологісних режимів неминуче призводить до корозії металів, розвитку плісняви, а також деградації паперу чи медикаментів [4].

В агропромислових об'єктах, зокрема в теплицях, недотримання кліматичних норм вкрай негативно позначається на фізіології рослин, оскільки навіть мінімальні відхилення гальмують базові біохімічні процеси, що напряду знижує кількість і якість продукції та вимагає впровадження спеціалізованих систем автоматизації моніторингу [5].

У таких приміщеннях, як серверні кімнати та центри обробки даних, вимоги є найсуворішими, оскільки відповідно до інженерних стандартів, відхилення показників від вузько заданих нормативів загрожує не просто перегрівом обладнання, а зупинкою критичних ІТ-систем та безповоротною втратою даних [6].

До розвитку цифрових рішень за мікрокліматом стежили вручну: працівники періодично знімали показання звичайних термометрів чи гігromетрів і записували їх у журнали [7]. Цей підхід безнадійно застарів. По-перше, людині властиво помилятися, вона не може чергувати 24/7, тому вночі чи у вихідні приміщення фактично залишаються без контролю. По-друге, втрачається оперативність: якщо посеред ночі зламається опалення, про це дізнаються лише зранку, коли збитки вже будуть незворотними.

Оптимальним виходом із цієї ситуації є перехід на автоматизовані системи моніторингу, побудовані на філософії Internet of Things(IoT). Відповідно до досліджень Ю. Є. Грудзинського [7], впровадження IoT-технологій наділяє кожен вимірювальний датчик здатністю до прямого мережевого підключення. Завдяки цьому архітектура системи стає децентралізованою та гнучкою, а її елементи можуть безперешкодно обмінюватися інформацією через інтернет [7]. Сучасні мікроконтролери, що зображені на рис. 1.1 дозволяють розгорнути мережу бездротових датчиків, які безперервно та з високою точністю збирають інформацію і передають її на сервер по Wi-Fi [8]. Це повністю знімає рутину з персоналу, дозволяє будувати наочні графіки в реальному часі, миттєво розсилати

тривожні сповіщення та створює надійний фундамент для автоматичного керування кліматичною технікою [8].



Рисунок 1.1 - Сучасні мікроконтролери ESP32, STM32, Raspberry Pi 5

1.2 Огляд комерційних систем моніторингу мікроклімату

Сучасний ринок рішень на базі Internet of Things (IoT) пропонує величезний вибір готових систем для моніторингу параметрів навколишнього середовища. За своєю архітектурою, протоколами зв'язку, відкритістю програмного забезпечення та вартістю вони суттєво відрізняються. Як зазначається у дослідженнях архітектури бездротових сенсорних мереж [8], для зручності аналізу весь масив доступних рішень можна умовно поділити на три великі категорії: масові побутові системи, професійні охоронно-промислові комплекси та вузькоспеціалізовані платформи для агропромислового і логістичного секторів.

1.2.1 Побутові екосистеми

Найбільш доступним і масовим сегментом є побутові системи розумного дому. Такі виробники, як Xiaomi з її екосистемою Mi Home рис. 1.2 [9], або численні бренди, що працюють на платформі Tuya Smart, пропонують доволі

компактні датчики температури та вологості, які часто оснащуються енергоефективними дисплеями на електронному чорнилі.



Рисунок 1.2 – Екосистема Mi Home [9]

Зв'язок у таких мережах здебільшого будується на енергоощадних протоколах BLE або Zigbee [10]. До безперечних переваг цих систем належать дуже низька вартість окремих вузлів, інтуїтивно зрозуміле налаштування через смартфон та вражаюча автономність - датчики здатні працювати від однієї мініатюрної батарейки більше року.

Однак для професійного використання на складах чи виробництвах вони підходять вкрай погано. Головною перепоною стає закритість їхньої програмної інфраструктури: інтегрувати такі датчики у сторонні системи підприємства або витягнути з них «сирі» дані на власний незалежний сервер майже неможливо без застосування неофіційних модифікацій [10]. Крім того, радіус дії BLE та Zigbee є досить обмеженим, що в умовах великих складських приміщень із металевими стелажми вимагає встановлення десятків проміжних ретрансляторів (хабів), а це суттєво знижує загальну надійність мережі.

1.2.2 Охоронно-промислові комплекси

Значно вищий рівень надійності пропонують професійні системи безпеки та автоматизації. Яскравим прикладом є українська компанія Ajax Systems, чиї рішення сьогодні активно впроваджуються як у великих офісах, так і на складських комплексах [11]. Для контролю мікроклімату в їхній лінійці передбачено датчик LifeQuality, зовнішній вигляд якого зображено на рис. 1.3 [11]. Цей пристрій з високою точністю вимірює не лише температуру та вологість, а й рівень концентрації вуглекислого газу



Рисунок 1.3 - Датчик якості повітря Ajax LifeQuality [11]

Перевагою таких рішень є використання висококласних швейцарських сенсорів та безпрецедентна надійність бездротової передачі даних. Відповідно до заявлених виробником характеристик, фірмовий радіопротокол Jeweller забезпечує стабільний обмін інформацією на відстанях до 1500 метрів,

зберігаючи стійкість зв'язку навіть за наявності значних електромагнітних завад [11]. Користувач отримує доступ до детальної аналітики та зручних графіків через фірмовий додаток. Водночас головним фактором, що стримує масове використання таких систем на невеликих підприємствах, є їхня ціна. Розгортання мережі датчиків та придбання центральної панелі керування потребує значних капіталовкладень. До того ж система залишається пропрієтарною, що серйозно обмежує можливості її глибокої кастомізації під специфічні інженерні задачі.

1.2.3 Спеціалізовані інтелектуальні системи моніторингу

Коли мова йде про комплексний моніторинг великих промислових об'єктів або агрокомплексів, найчастіше застосовуються спеціалізовані ІТ-рішення, розроблені «під ключ». Огляд таких продуктів чітко демонструє сучасний вектор розвитку професійного моніторингу.

Наприклад, українська компанія Intellias спеціалізується на розробці систем промислового Internet of Things (IoT), зборі даних та генеруванні рекомендацій щодо керування середовищем [12]. Приклад типового інтерфейсу такої системи наведено на рис. 1.4 [13].

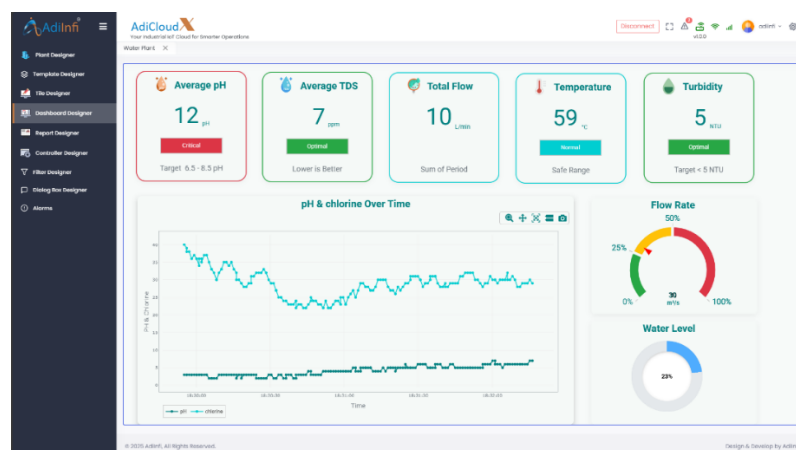


Рисунок 1.4 - Приклад графічного вебінтерфейсу промислової платформи моніторингу IoT [13]

Ця платформа агрегує ключові показники мікроклімату, такі як температура та вологість, дозволяючи операторам миттєво реагувати на критичні зміни та оптимізувати робочі процеси в режимі реального часу.

Також на міжнародному ринку присутні комплекси від компанії Inson Green, яка створює професійні автоматизовані рішення для розумних теплиць (Smart Greenhouses) рис. 1.5, поєднуючи контроль різних параметрів клімату у єдину замкнену екосистему з можливістю дистанційного моніторингу [14]. Такі рішення доводять свою високу ефективність саме в умовах нестабільного клімату та жорстких вимог до зберігання продукції.



Рисунок 1.5 - Розумна теплиця компанії Inson Green

1.2.4 Порівняльний аналіз комерційних рішень

Для кращого розуміння місця нашої майбутньої розроблення серед існуючих альтернатив, зведемо ключові характеристики розглянутих систем у єдину порівняльну таблицю 1.1.

Таблиця 1.1 - Порівняльна характеристика комерційних систем моніторингу мікроклімату

Характеристика	Побутові (Xiaomi, TuYa)	Охоронні (Ajax Systems)	Спеціалізовані (Intellias)
Основний протокол зв'язку	BLE, Zigbee, Wi-Fi	Jeweller (пропрієтарний)	Wi-Fi, LoRaWAN, Ethernet
Масштабованість	Низька / Середня	Висока (до 100+ датчиків)	Дуже висока (хмарна архітектура)
Відкритість екосистеми (API)	Сильно обмежена	Обмежена	Висока (гнучка інтеграція)
Вартість впровадження	Низька	Висока	Дуже висока (проектна розроблення)
Аналітика та експорт даних	Базова	Детальна (в межах додатку)	Професійна, з генерацією звітів

Проведений аналіз наочно демонструє серйозний дисбаланс на ринку готових рішень. Побутові екосистеми приваблюють своєю низькою ціною, проте їхня програмна закритість та слабкий радіосигнал роблять їх непридатними для розгортання у великих складських приміщеннях. Натомість професійні комплекси забезпечують бездоганну надійність, глибоку аналітику та зводять до мінімуму помилки персоналу, але їхня вартість часто є непідйомною для малого та середнього бізнесу.

Саме тому розроблення власної автоматизованої системи на базі мікроконтролера ESP32 є абсолютно виправданим кроком. Такий підхід дозволить взяти найкраще від обох світів: відкритість даних, стабільну передачу по Wi-Fi та зручну вебвізуалізацію від промислових систем, зберігши при цьому низьку собівартість та гнучкість, характерні для самостійних IoT-рішень [10].

1.3 Огляд технологій бездротової передачі даних

Побудова сучасної системи моніторингу неможлива без організації надійного каналу зв'язку між вимірювальними вузлами та сервером. Використання традиційних дротових ліній (наприклад, промислових шин RS-485 або Ethernet) у приміщеннях великої площі є економічно недоцільним через значну вартість монтажних робіт, складність прокладання кабелів повз стелажні конструкції та неможливість швидкого переміщення датчиків за потреби. [10]

На сьогодні в сегменті Internet of Things (IoT) застосовуються протоколи BLE, Zigbee, LoRaWAN та Wi-Fi. Кожна з цих технологій розроблялася під специфічні задачі та пропонує свій баланс між дальністю дії, енергоспоживанням та пропускнуою здатністю.

Технологія Bluetooth Low Energy [15] вирізняється наднизьким енергоспоживанням, що дозволяє автономним сенсорам працювати від мініатюрних елементів живлення протягом кількох років. Проте радіус стабільної дії BLE зазвичай обмежується кількома десятками метрів. В умовах наявності металевих конструкцій та перешкод це вимагає встановлення щільної мережі ретрансляторів [8]. Схожу проблему частково вирішує стандарт Zigbee, який підтримує комірчасту топологію мережі, де кожен пристрій із постійним живленням може передавати сигнал далі по ланцюжку [8]. Незважаючи на високу надійність такого підходу, розгортання мережі Zigbee вимагає обов'язкової наявності спеціалізованого апаратного шлюзу (координатора) для виходу в Інтернет та агрегації даних.

Для об'єктів значної площі розглядають технологію LoRaWAN [8], яка забезпечує передачу даних на відстані до кількох кілометрів при мінімальних витратах енергії. Вона ідеально підходить для агропромислового сектору, моніторингу полів або великих відкритих майданчиків. Однак її використання для моніторингу в межах однієї будівлі або складського комплексу є

надлишковим, оскільки протокол має вкрай низьку швидкість передачі даних і також потребує дорогих базових станцій для підключення до мережі.

З іншого боку, стандарт Wi-Fi [8] є бездротовою технологією для передавання даних між пристроями, яка забезпечує прямий доступ до локальної мережі або інтернету без необхідності розгортання проміжних специфічних шлюзів. Хоча класичний Wi-Fi має вище енергоспоживання порівняно з BLE або Zigbee, ця проблема ефективно вирішується на рівні програмного забезпечення шляхом переведення мікроконтролера в режим глибокого сну між сеансами зчитування та відправки даних. Крім того, більшість сучасних приміщень вже мають розгорнуту інфраструктуру маршрутизаторів для роботи персоналу та внутрішньої техніки. Це дозволяє легко інтегрувати нові вузли моніторингу в існуючу мережу практично з нульовими витратами на мережеве обладнання.

1.4 Аналіз мікроконтролерів для побудови вузлів моніторингу

Основою будь-якого сучасного вузла моніторингу є мікроконтролер - компактний обчислювальний пристрій, який приймає інформацію з датчиків, аналізує її та керує передачею цих даних на сервер. Від правильного вибору цієї платформи залежить не лише швидкодія та стабільність усієї системи, а й її розміри, енергоспоживання та складність написання програмного коду [16]. Сьогодні в сегменті розроблення систем Internet of Things(IoT) та автоматизації використовують три основні сімейства плат: класичні моделі Arduino, що зображені на рис. 1.6 , а також модулі на базі мікросхем ESP8266 продемонстрований на рис. 1.7 [16].



Рисунок 1.6 – Сімейство Arduino

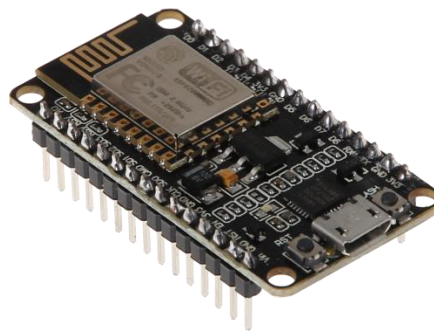


Рисунок 1.7 – Мікроконтролер ESP8266

Класичні плати сімейства Arduino (наприклад, Uno, Nano чи Mega) використовують для навчання та побудови базової автоматики. Їхньою головною перевагою є надзвичайна простота у використанні, доступність та наявність великої кількості цифрових і аналогових входів для підключення різноманітних сенсорів. Проте для побудови сучасних мережевих пристроїв їхніх ресурсів відверто недостатньо [16]. Вони базуються на застарілих 8-бітних мікропроцесорах із тактовою частотою всього 16 МГц та мають критично малий обсяг пам'яті. Головним же їхнім недоліком є відсутність вбудованих засобів бездротового зв'язку. Щоб така плата могла передавати дані по Wi-Fi, до неї потрібно підключати додаткові зовнішні модулі, що неминуче збільшує фізичні

розміри пристрою, здорожчує його та знижує надійність системи через зайві дротові з'єднання.

Справжню революцію у сфері доступного IoT здійснили плати на базі чіпа ESP8266 [17]. Їхньою ключовою інновацією стало об'єднання в одному недорогому кристалі повноцінного мікроконтролера та функціонального модуля Wi-Fi [17]. Це дало змогу розробникам створювати мініатюрні пристрої, здатні самостійно підключатися до роутера та передавати інформацію в інтернет. Втім, з часом проявилися й обмеження цієї платформи [17]: наявність лише одного аналогового входу, відносно невелика кількість універсальних пінів та обмежена обчислювальна потужність, якої іноді не вистачає для одночасної стабільної роботи вебсервера та постійного опитування кількох вимогливих датчиків.

Для побудови бездротових систем моніторингу вибирають мікроконтролер ESP32. Ця платформа є прямим, значно вдосконаленим нащадком ESP8266. Чіп ESP32 зображений на рис. 1.6 оснащено потужним двоядерним процесором із тактовою частотою до 240 МГц та в рази більшим обсягом оперативної пам'яті [8].



Рисунок 1.8 – Процесор ESP32.

Найважливішою ж його перевагою є наявність одразу двох вбудованих бездротових технологій - Wi-Fi та Bluetooth (включно з енергоефективним BLE) [17]. Плата має десятки вільних виводів, апаратно підтримує всі необхідні для датчиків інтерфейси (I2C, SPI, UART) та містить багатоканальний АЦП [18]. Крім того, ESP32 підтримує просунуті режими глибокого енергозбереження, що дозволяє пристрою споживати мізерні мікроампери струму під час пауз між вимірюваннями [18].

Враховуючи поставлені перед роботою завдання - а саме необхідність безперервного та точного зчитування кліматичних показників, їх первинної обробки та стабільної відправки на віддалений сервер по бездротовій мережі, використання платформи ESP32 є найбільш раціональним і безальтернативним рішенням [17]. Вона гарантує ідеальний баланс між високою обчислювальною потужністю, самодостатністю комунікаційних інтерфейсів та доступною ціною, дозволяючи створити надійний, швидкий і сучасний вузол моніторингу.

1.5 Огляд методів обробки та візуалізації даних у системах моніторингу

Ефективність функціонування сучасних автоматизованих систем моніторингу залежить не лише від точності первинних вимірювань, а й від обраної стратегії обробки, передачі та представлення інформації. У системах на базі Internet of Things (IoT) цей процес розділяється на три рівні: польовий (збір даних мікроконтролером), мережевий (передача даних) та прикладний (хмарна обробка та візуалізація) [8].

1.5.1 Методи попередньої обробки даних на стороні мікроконтролера

Сучасні цифрові сенсори генерують потік даних, який може містити випадкові шуми або короточасні викиди. Як зазначає В. В. Войтко, використання

методів попередньої фільтрації безпосередньо на мікроконтролері дозволяє суттєво підвищити достовірність інформації перед її відправкою на сервер [9]. Для систем кліматичного контролю найчастіше застосовують[9]:

- *Метод рухомого середнього*: дозволяє згладити графік показників, нівелюючи випадкові похибки окремих вимірювань.
- *Метод медіанної фільтрації*: ефективний для відсікання аномальних пікових значень, що можуть виникати через електромагнітні наведення на довгих лініях зв'язку.

1.5.2 Аналіз мережевих протоколів для передачі в реальному часі

Для виконання вимоги щодо оновлення даних із частотою 2 Гц, критично важливий є вибір протоколу з мінімальними накладними витратами. Традиційний протокол ння кожної нової точки графіка клієнт має ініціювати нове з'єднання, що при частоті 0.5 с може спричинити перевантаження вебсервера.

Більш ефективним для завдань моніторингу є протокол MQTT (Message Queuing Telemetry Transport). Як підкреслює Ю. Є. Грудзинський, концепція «видавець-підписник» дозволяє пристрою надсилати дані лише при їх зміні або за коротким таймером, мінімізуючи об'єм мережевого трафіку [7]. Однак для забезпечення максимальної плавності графіків безпосередньо у веббраузері найбільш раціональним є використання технології WebSockets [18]. На відміну від HTTP, WebSockets створюють постійне двостороннє з'єднання, через яке сервер може «штовхати» дані клієнту миттєво після їх отримання від ESP32.

1.5.3 Технології візуалізації та побудови динамічних графіків

Кінцевим етапом обробки є візуалізація, яка має забезпечити наочність трендів температури та вологості. У сучасній веброзробці для цього застосовуються спеціалізовані графічні бібліотеки:

- *Chart.js*[19]: легка та гнучка бібліотека на базі HTML5 Canvas. Вона ідеально підходить для відображення часових рядів (Time Series) і дозволяє реалізувати ефект «пливучого» графіка, коли нові дані додаються праворуч, а старі зсуваються ліворуч без мерехтіння сторінки. Приклад можливості побудови графіків зображений на рис. 1.9;

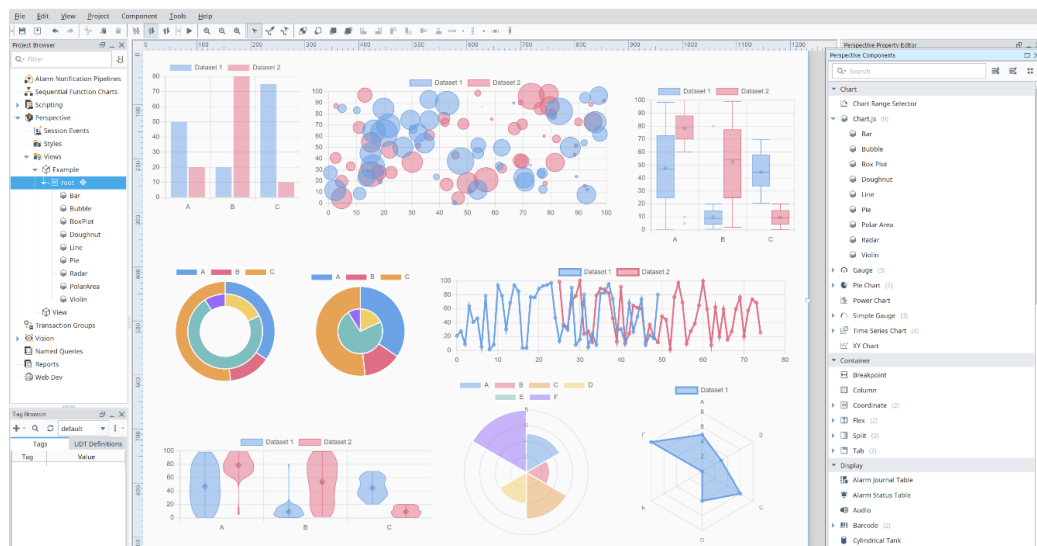


Рисунок 1.9 – Можливості побудови графіків бібліотеки Chart.js

- *Google Charts*[20]: пропонує широкий вибір інструментів, але вимагає постійного завантаження скриптів із зовнішніх серверів Google, що може бути небажаним для автономних локальних систем. Приклад можливості побудови графіків зображений на рис. 1.8.



Рисунок 1.10 – Можливості побудови графіків бібліотеки Google Charts

Застосування бібліотеки Chart.js у поєднанні з WebSockets дозволяє досягти частоти оновлення понад 2 Гц, що забезпечує високу плавність відображення перехідних процесів у приміщенні.

1.6 Обґрунтування та вибір елементної бази системи

Для реалізації автоматизованої системи моніторингу кліматичних показників необхідно підібрати елементну базу, яка не лише забезпечить стабільну роботу пристрою, а й гарантовано задовольнить жорсткі вимоги технічного завдання. Основними критеріями вибору є точність визначення температури не гірше $\pm 2^{\circ}\text{C}$ у діапазоні від $+10^{\circ}\text{C}$ до $+50^{\circ}\text{C}$ та вологості не гірше $\pm 5\%$ у діапазоні від 20% до 100%. Крім того, апаратна платформа має володіти достатньою обчислювальною потужністю для забезпечення частоти оновлення даних у графічному інтерфейсі на рівні 2 Гц.

1.6.1 Обґрунтування вибору мікроконтролера

Основою вимірювального вузла системи є центральний обчислювальний мікроконтролер. Традиційні 8-бітні плати сімейства Arduino (Uno, Nano) мають

низьку тактову частоту (16 МГц) та обмежений обсяг пам'яті, а головне - позбавлені вбудованих мережевих інтерфейсів, що потребує підключення зовнішніх модулів і суттєво знижує надійність пристрою [16]. Більш сучасні чіпи серії ESP8266 мають інтегрований Wi-Fi модуль [17], проте вони обмежені малою кількістю універсальних виводів (GPIO) та наявністю лише одного аналогового входу, що унеможлиблює подальше розширення або модернізацію системи.

З огляду на це, як головний обчислювальний модуль обрано відлагоджувальну плату ESP32 DevKit v1. Чіп ESP32 оснащено потужним двоядерним 32-бітним процесором із тактовою частотою до 240 МГц та значним обсягом оперативної пам'яті [17]. Обчислювальних ресурсів даного чіпа цілком достатньо для паралельного опитування датчиків, первинної фільтрації даних та підтримки постійного WebSocket-з'єднання з вебсервером.

Таким чином, порівняльний аналіз обчислювальних платформ показує, що мікроконтролер ESP32 є найбільш раціональним і безальтернативним рішенням для даної роботи. Його висока швидкодія та вбудований мережевий стек дозволяють використати його як єдине обчислювальне ядро, здатне забезпечити стабільну обробку та відправку кліматичних показників у реальному часі з необхідною частотою 2 Гц без ризику переповнення буфера чи затримок.

1.6.2 Порівняльний аналіз та вибір первинних перетворювачів

Для забезпечення заданої точності моніторингу критично важливим є вибір первинних вимірювальних перетворювачів навколишнього середовища. Було проведено аналіз найбільш поширених на ринку цифрових сенсорів [9]: DHT11, DHT22, BME280 та SHT31-D. Основні технічні параметри цих пристроїв зведено в порівняльну таблицю 1.2.

Таблиця 1.2 - Порівняльна характеристика цифрових датчиків мікроклімату

Параметр сенсора	DHT11	DHT22	BME280	SHT31-D
Діапазон вимірювання температури, °C	0...+50	-40...+80	-40...+85	-40...+125
Похибка вимірювання температури, °C	±2.0	±0.5	±0.5	±0.2
Діапазон вимірювання вологості, %	20...90	0...100	0...100	0...100
Похибка вимірювання вологості, %	±5.0	±2.0	±3.0	±2.0
Інтерфейс зв'язку з МК	Single-wire	Single-wire	I2C / SPI	I2C
Максимальна частота опитування, Гц	0.5	0.5	10	10

Згідно з отриманими даними, датчик DHT11 має граничну точність ($\pm 2.0^{\circ}\text{C}$) та обмежений діапазон вимірювання вологості (до 90%), що суперечить вимогам ТЗ. Модель DHT22 має низьку частоту опитування (0.5 Гц), через що технічно неможливо досягти частоти оновлення графічного інтерфейсу у 2 Гц. Сенсор BME280 повністю задовольняє вимоги, проте містить надлишковий для даної задачі датчик атмосферного тиску, що безпосередньо здорожчує конструкцію кінцевого пристрою.

Оптимальним вибором є датчик SHT31-D від компанії Sensirion, який працює по високошвидкісній цифровій шині I2C [21], зображений на рис. 1.11. Його заводська похибка ($\pm 0.2^{\circ}\text{C}$ для температури та $\pm 2\%$ для вологості) забезпечує високу точність вимірювань, а максимальна частота опитування до 10 Гц повністю покриває динамічні вимоги вебінтерфейсу [21]. Для розширення функціонала системи додатково обрано напівпровідниковий датчик якості повітря MQ-135, що зображений на рис. 1.12. Він підключається до аналогового входу мікроконтролера і дозволяє фіксувати наявність шкідливих газів [22].

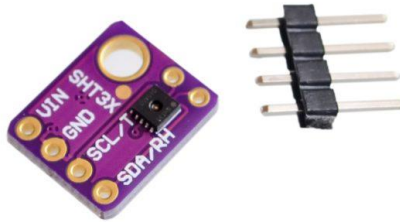


Рисунок 1.11 - Датчик SHT31-D



Рисунок 1.12 - Датчик якості повітря MQ-135

Отже, вибір цифрового сенсора SHT31-D є повністю обґрунтованим, оскільки його метрологічні характеристики суттєво перевищують мінімальні вимоги технічного завдання (забезпечуючи точність температури $\pm 0.2^{\circ}\text{C}$ замість $\pm 2^{\circ}\text{C}$, а вологості - $\pm 2\%$ замість $\pm 5\%$). Це гарантує високу достовірність первинних даних, а додавання напівпровідникового датчика MQ-135 дозволяє модернізувати систему до рівня комплексної платформи автоматизованого екологічного моніторингу приміщень.

1.6.3 Організація вузла живлення та виконавчих механізмів

Для забезпечення тривалої безперервної роботи автоматизованої системи проєктом передбачено використання стаціонарного імпульсного блока живлення з вихідними параметрами 12 В та 2 А, що зображений на рис. 1.13.



Рисунок 1.13 - Імпульсний блок живлення

Вибір такої потужності обґрунтований інтеграцією в систему активного виконавчого пристрою - відцентрового вентилятора (равлика) MX-3010 із робочою напругою 12 В [23], що зображений на рис. 1.14.



Рисунок 1.14 - Вентилятор MX-3010

Він призначений для примусового охолодження об'єкта та організації локальної циркуляції повітряного потоку навколо вимірювальних сенсорів.

Оскільки цифрова логіка мікроконтролера ESP32 та датчиків потребує низьковольтного живлення, до схеми введено DC-DC понижуючий конвертер на базі мікросхеми LM2596 [24], який зображений на рис. 1.15.



Рисунок 1.15 - Понижуючий конвертер на базі мікросхеми LM2596

Він забезпечує стабільне перетворення напруги 12 В у 5 В із високим коефіцієнтом корисної дії. Використання імпульсного конвертера замість лінійного стабілізатора мінімізує виділення паразитного тепла всередині корпусу пристрою, що повністю виключає тепловий вплив на покази високоточного датчика SHT31-D [10]. Для комутації 12-вольтового вентилятора від цифрового виводу МК (3.3 В) обрано силовий польовий транзистор IRLZ44N [25], що зображений на рис. 1.16.



Рисунок 1.16 - Силовий польовий транзистор IRLZ44N

Цей MOSFET належить до класу пристроїв із логічним рівнем керування (Logic-Level), що дозволяє повністю відкривати його канал безпосередньо сигналом ШІМ з GPIO мікроконтролера.

Запропонована конфігурація вузла живлення та комутаційних елементів створює надійну й безпечну апаратну основу. Поєднання імпульсного регулятора LM2596 та транзистора IRLZ44N дозволяє ефективно розділити логічну та силову частини системи, забезпечуючи стабільну напругу без теплових завад для сенсорів.

Отже, для реалізації автоматизованої системи обрано наступний перелік компонентів: головним обчислювальним модулем слугуватиме плата ESP32 DevKit v1 із вбудованим Wi-Fi, а прецизійним перетворювачем мікроклімату — цифровий I2C-датчик температури та вологості SHT31-D. Контроль якості повітря забезпечить напівпровідниковий сенсор MQ-135. Живлення системи здійснюватиметься від імпульсного блоку на 12 В та 2 А із застосуванням DC-DC понижуючого конвертера LM2596 для стабільного формування логічних 5 В. Для активної аспірації сенсорів обрано відцентровий вентилятор MX-3010, апаратно-програмне керування яким реалізовано через силовий N-канальний транзистор IRLZ44N.

Основою надійного функціонування апаратної частини системи моніторингу є коректне електричне узгодження всіх обраних елементів. Головним завданням на етапі розроблення принципової схеми було забезпечення безконфліктного обміну даними між мікроконтролером ESP32 та первинними перетворювачами, а також безпечне керування силовим навантаженням.

Рисунок 2.1 - Принципова електрична схема вузла моніторингу

Підключення цифрового датчика температури та вологості SHT31-D реалізовано за допомогою синхронної послідовної шини I2C. Відповідно до технічної документації мікроконтролера ESP32, для апаратної реалізації шини I2C за замовчуванням використовуються виводи GPIO21 (SDA — лінія даних) та GPIO22 (SCL — лінія тактування) [17]. Живлення датчика здійснюється від стабілізованої лінії 3.3 В, яку забезпечує внутрішній лінійний стабілізатор (LDO) самої плати ESP32 DevKit, оскільки споживання сенсора SHT31-D у режимі вимірювання становить менше 1000 мкА [21]. Підтягуючі резистори для ліній I2C не застосовувалися на зовнішній макетній платі, оскільки вони інтегровані безпосередньо на модулі датчика.

Інтеграція датчика якості повітря MQ-135 реалізована шляхом підключення його аналогового виходу до порту GPIO34 мікроконтролера. Цей вивід підключено до внутрішнього 12-бітного аналогово-цифрового перетворювача (АЦП1) ESP32. Живлення підігрівача (Heater) напівпровідникового сенсора MQ-135 вимагає напруги 5 В та струму близько 150 мА [22], тому його підключено безпосередньо до загальної лінії 5 В, яка формується модулем LM2596.

Особливої уваги під час проєктування вимагав вузол керування виконавчим механізмом - відцентровим вентилятором MX-3010, що працює від напруги 12 В [23]. Комутація здійснюється за допомогою N-канального польового транзистора IRLZ44N, який виконує функцію електронного ключа. Сток (Drain) транзистора з'єднано з негативним контактом вентилятора, а витік (Source) - із загальною шиною заземлення (GND). Керуючий сигнал подається на затвор (Gate) з порту GPIO26 мікроконтролера. Оскільки IRLZ44N належить до класу транзисторів із логічним рівнем керування (Logic-Level Gate Drive), напруги 3.3 В від порту ESP32 достатньо для повного відкриття переходу без використання додаткових транзисторних драйверів [25].

Для загального зниження напруги живлення від мережевого адаптера (12 В) до рівня логіки (5 В) застосовано DC-DC перетворювач на мікросхемі LM2596 [24]. Всі компоненти об'єднані спільною шиною заземлення (GND) для унеможливлення виникнення паразитних різниць потенціалів, що гарантує високу точність роботи АЦП та цифрових шин передачі даних.

2.2 Розрахунок енергетичного балансу та теплових режимів системи

Надійність роботи будь-якого пристрою IoT критично залежить від стабільності його вузла живлення [9]. Падіння напруги під час пікових навантажень (наприклад, під час активної передачі даних по Wi-Fi) може призвести до перезавантаження мікроконтролера або спотворення показів АЦП [17]. Для підтвердження правильності вибору джерела живлення та DC-DC перетворювача проведемо розрахунок енергетичного балансу системи для режиму максимального навантаження.

Енергетичний розрахунок доцільно розділити на два контури: низьковольтний логічний контур (5 В) та загальний силовий контур (12 В).

1. Розрахунок споживання низьковольтного контуру (5 В)

До цього контуру підключені мікроконтролер ESP32, цифровий датчик SHT31-D та напівпровідниковий датчик MQ-135. Максимальний струм споживання мікроконтролера ESP32 в режимі активної передачі даних (Tx) за протоколом 802.11b/g/n становить = 240 мА, проте в моменти встановлення з'єднання можливі короточасні сплески струму до 500 мА [17]. Для забезпечення надійності в розрахунок приймаємо пікове значення. Струм споживання нагрівального елемента датчика якості повітря MQ-135 становить = 150 мА [22]. Споживанням датчика SHT31-D (≈ 1 мА) можна знехтувати [21], але для точності врахуємо його в загальній сумі.

Загальний максимальний струм логічного контуру обчислюється за формулою:

$$I_{5V} = I_{ESP} + I_{MQ} + I_{SHT} \quad 2.1$$

Підставивши значення, отримаємо:

$$I_{5V} = 500 + 150 + 1 = 651 \text{ мА} = 0.651 \text{ А} \quad 2.2$$

Максимальна споживана потужність логічного контуру при напрузі 5 В становить:

$$P_{5V} = U_1 \cdot I_{5V} = 5 \cdot 0.651 = 3.255 \text{ Вт} \quad 2.3$$

2. Втрати на DC-DC перетворювачі та тепловий розрахунок Живлення 5 В формується з вхідної напруги 12 В за допомогою імпульсного понижуючого конвертера LM2596. Згідно з технічною документацією, середній коефіцієнт корисної дії ККД цієї мікросхеми при різниці напруг 12 В та 5 В становить $\approx 80\%$ (або 0.8) [24]. Потужність, яку конвертер споживає від первинного джерела 12 В, визначається як:

$$P_{LM_in} = \frac{P_{5V}}{\eta} = \frac{3.255}{0.8} \approx 4.07 \text{ Вт} \quad 2.4$$

Різниця між спожитою та відданою потужністю розсіюється у вигляді тепла:

$$P_{loss} = P_{LM_in} - P_{5V} = 4.07 - 3.255 = 0.815 \text{ Вт} \quad 2.5$$

Мікросхема LM2596 випускається в корпусі ТО-263, типовий тепловий опір якого (перехід-середовище) без використання додаткового масивного радіатора становить $\approx 50^\circ\text{C}/\text{Вт}$ [24]. Очікуваний нагрів мікросхеми відносно температури навколишнього середовища складе:

$$\Delta T = P_{loss} \cdot R_{th} = 0.815 \cdot 50 \approx 40.75^\circ\text{C} \quad 2.6$$

При максимальній температурі повітря в приміщенні згідно з ТЗ ($+50^\circ\text{C}$), температура кристала мікросхеми досягне 90.75°C , що знаходиться в безпечних межах (максимально допустима температура кристала для LM2596 становить $+125^\circ\text{C}$) [24]. Це доводить, що система не потребує активного охолодження плати конвертера.

3. Розрахунок загального споживання системи (12 В)

Струм, який DC-DC конвертер споживає від лінії 12 В, становить:

$$I_{LM_in} = \frac{P_{LM_in}}{U_{in}} = \frac{4.07}{12} \approx 0.339 \text{ А} \quad 2.7$$

До лінії 12 В також підключено виконавчий механізм - відцентровий вентилятор MX-3010. Його робочий струм становить $= 150 \text{ мА}$ (0.15 А) [23]. Загальний максимальний струм, який споживає пристрій від блоку живлення, дорівнює:

$$I_{total} = I_{LM_in} + I_{fan} = 0.339 + 0.150 = 0.489 \text{ А} \quad 2.8$$

Загальна споживана потужність всього комплексу становить:

$$P_{total} = U_{in} \cdot I_{total} = 12 \cdot 0.489 = 5.868 \text{ Вт} \quad 2.9$$

4. Обґрунтування вибору мережевого блоку живлення

Для проєкту обрано імпульсний блок живлення з номінальною вихідною напругою 12 В та максимальним струмом 2 А. Його номінальна потужність складає:

$$P_{PSU} = 12 \cdot 2.0 = 24 \text{ Вт} \quad 2.10$$

Коефіцієнт запасу потужності, що гарантує відсутність перевантажень та перегріву самого блоку живлення, обчислюється як:

$$K_{res} = \frac{P_{PSU}}{P_{total}} = \frac{24}{5.868} \approx 4.09 \quad 2.11$$

Проведені розрахунки доводять, що сумарна пікова потужність системи становить близько 5.87 Вт. Використання блоку живлення потужністю 24 Вт (12 В, 2 А) забезпечує чотирикратний запас надійності. Це дозволить блоку живлення працювати у полегшеному тепловому режимі (без перегріву), повністю виключить просідання напруги під час активації Wi-Fi передавача ESP32 [17], а також залишає значний потенціал для подальшого масштабування системи.

2.3 Конструктивне виконання та компонування пристрою

Точність автоматизованої системи моніторингу залежить не лише від характеристик обраних електронних компонентів, а й від їх правильного

просторового компонування. Головною інженерною проблемою під час проєктування є уникнення теплового впливу від обчислювальних та силових модулів на високоточні первинні перетворювачі [9].

Конструктивно пристрій розміщено у єдиному металевому корпусі прямокутної форми, що закривається верхньою кришкою. Використання металу забезпечує високу механічну міцність, а також виконує роль електромагнітного екрана, що мінімізує вплив зовнішніх радіозавад на чутливі вимірювальні кола. Знімна верхня кришка надає зручний доступ до внутрішніх компонентів для їх обслуговування чи налагодження. Вигляд зверху та з боку корпусу зображений на рис.2.2 та рис. 2.3.



Рисунок 2.2 - Зовнішній вигляд зверху прямокутного металевого корпусу з відкритою кришкою.



Рисунок 2.3 - Зовнішній вигляд збоку прямокутного металевого корпусу з закритою кришкою.

Внутрішній об'єм корпусу функціонально розділений на дві умовні зони:

1. Системо-технічна зона. Тут розміщено відлагоджувальну плату мікроконтролера ESP32, імпульсний понижуючий конвертер LM2596 та польовий транзистор IRLZ44N. Під час активної роботи Wi-Fi модуля та перетворення напруги ці елементи виділяють паразитне тепло. Для запобігання перегріву в цій частині корпусу передбачено пасивні вентиляційні щілини.

2. Вимірювальна зона. Максимально віддалена від системотехнічної зони. Тут закріплено цифровий датчик SHT31-D та напівпровідниковий датчик газу MQ-135. Фізичне рознесення зон виключає пряму теплопередачу через друковані плати та конвекційні потоки всередині корпусу [1].

Для забезпечення мінімальної інерційності вимірювань застосовано метод активної аспірації [5]. Відцентровий вентилятор MX-3010 встановлено таким чином, що він здійснює примусовий забір повітря із зовнішнього середовища через спеціальні отвори в корпусі та направляє спрямований повітряний потік

безпосередньо на чутливий елемент датчика SHT31-D. Це не лише забезпечує достовірність показів, а й запобігає застою повітря навколо сенсорів. Кріплення вентилятору MX-3010 зображене на рис. 2.4.

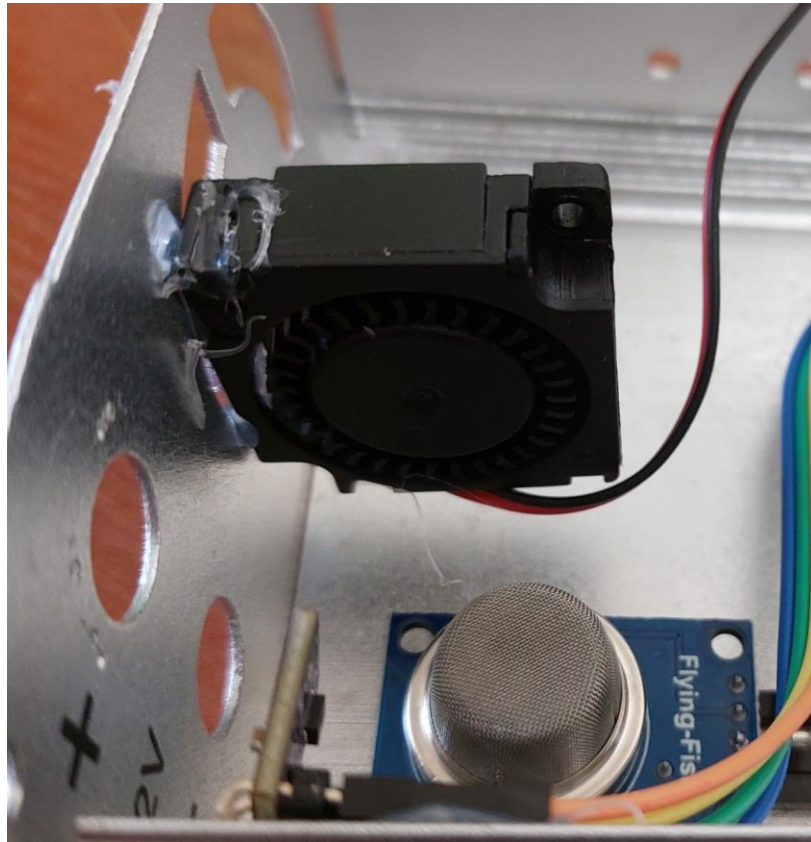


Рисунок 2.4 — Кріплення вентилятору MX-3010

Для надійної апаратної взаємодії між модулями реалізовано наступну схему електричних підключень:

- Сенсор мікроклімату SHT31-D. Зв'язок здійснюється за цифровою шиною I2C: контакти SDA та SCL підключені до портів GPIO21 та GPIO22 мікроконтролера ESP32 відповідно. Живлення датчика реалізовано від лінії 3.3 В. Підключення датчику SHT31-D зображене на рис. 2.6.



Рисунок 2.5 - Підключення датчика SHT31-D]

- Сенсор якості повітря MQ-135. Аналоговий вихід датчика (A0) з'єднано з портом GPIU34, який підтримує функціонал АЦП. Живлення 5 В подається безпосередньо від перетворювача напруги. Підключення датчику MQ-135 зображене на рис. 2.6.

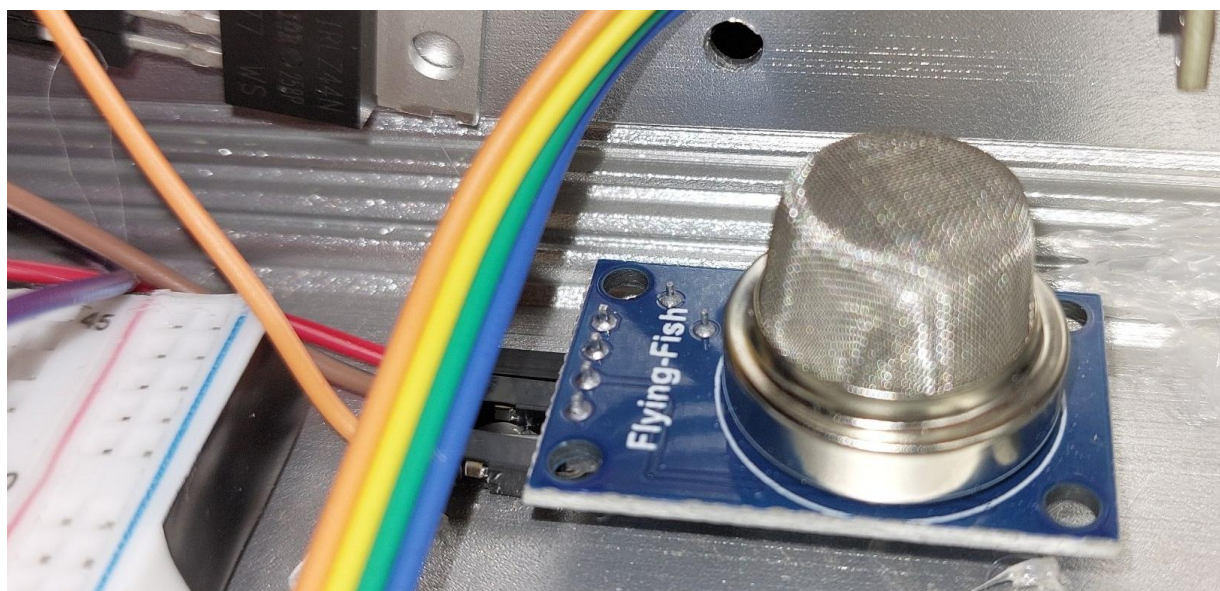


Рисунок 2.6 - Комутація датчика MQ-135

- Вузол керування вентилятором. Апаратна комутація реалізована на базі N-канального MOSFET транзистора IRLZ44N. Керуючий ШІМ-сигнал з порту GPIO26 надходить на затвор (Gate) транзистора, вентилятор підключено до стоку (Drain), а витік (Source) виведено на спільну "землю". Підключення вузлу керування вентилятором зображене на рис. 2.7.

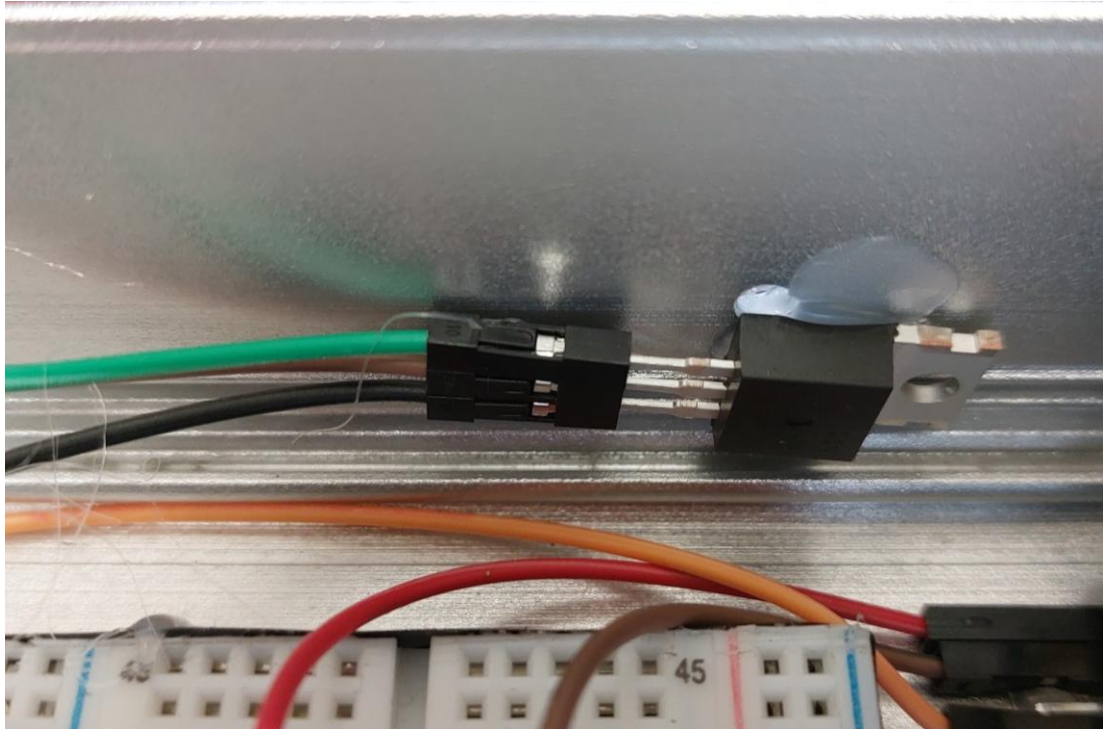


Рисунок 2.7 – Підключення транзистора IRLZ44N.

- Підсистема живлення. Зовнішня напруга 12 В від роз'єму DC 5.5x2.5 мм, що виведений на задню стінку корпусу, подається на вентилятор та на вхід понижуючого конвертера LM2596. Конвертер стабілізує напругу до рівня 5 В, необхідного для живлення мікроконтролера ESP32 та датчика газу. Підключення підсистеми живлення зображене на рис. 2.8.

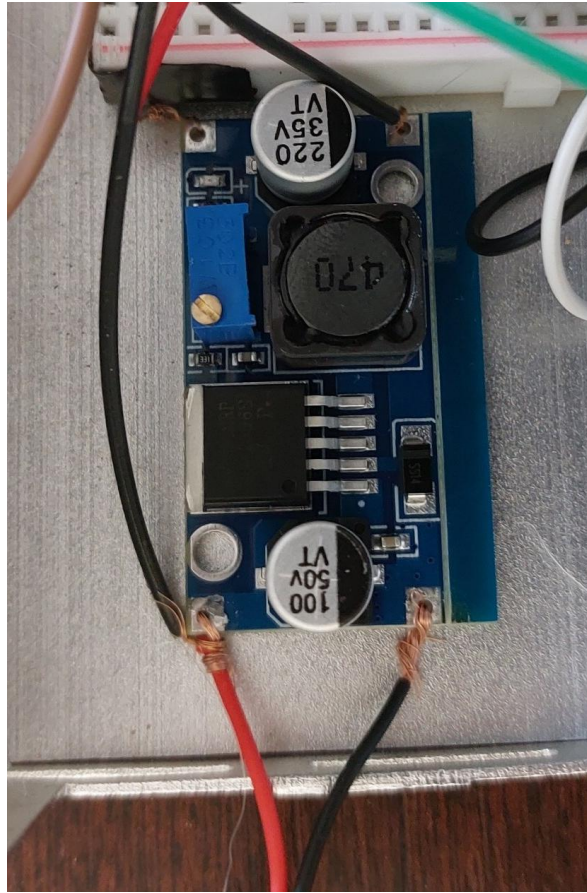


Рисунок 2.8 - Підключення блоку живлення до перетворювача LM2596]

Усі електронні компоненти зафіксовані за допомогою термоклею. Електричні з'єднання між модулями виконано гнучкими ізольованими провідниками типу Dupont. Це забезпечує високу ремонтпридатність системи та стійкість до механічних вібрацій, які створює вентилятор під час роботи. Повну збірку зображено на рис. 2.9 з відкритою кришкою та на рисунку 2.10 повністю зібрану систему.

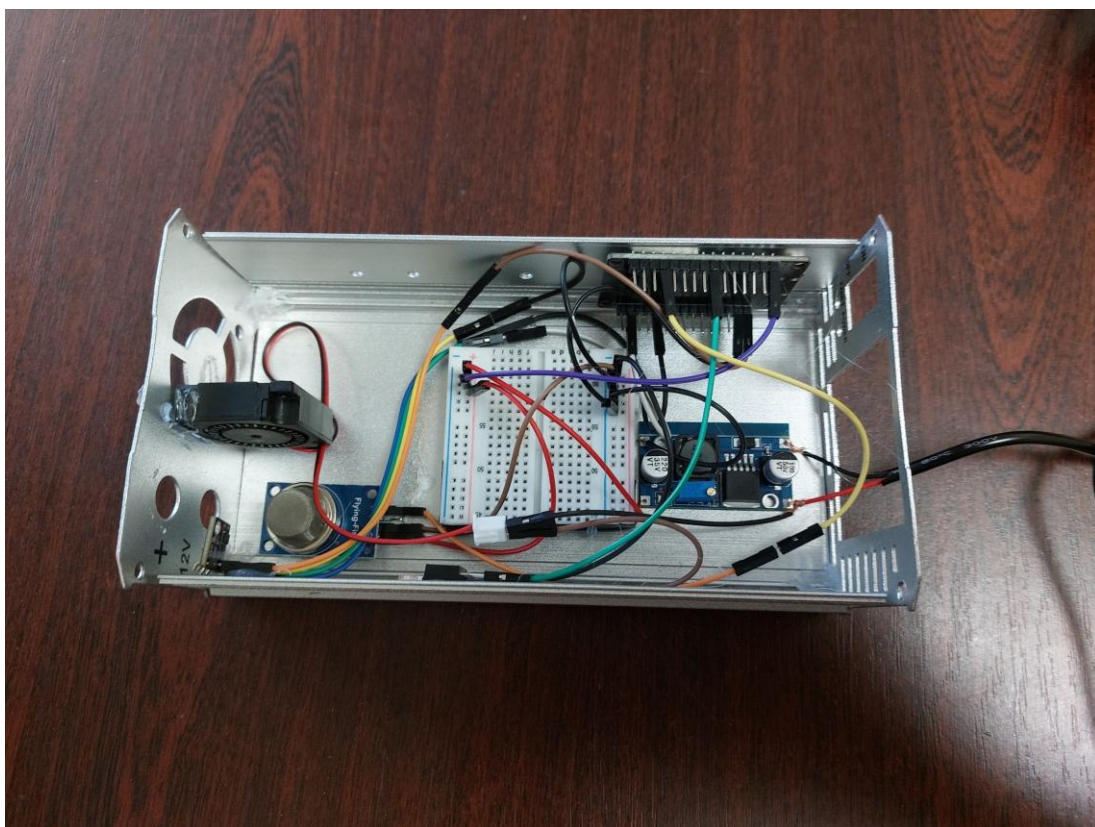


Рисунок 2.9 – Збірка системи з відкритою кришкою.



Рисунок 2.10 – Збірка системи з закритою кришкою.

Розроблена конструкція корпусу та внутрішнє компонування елементів комплексно вирішують проблему паразитних теплових завад. Розділення пристрою на силову та вимірювальну зони у поєднанні з активною примусовою аспірацією повітря гарантує вимірювання реальних параметрів мікроклімату в приміщенні, повністю відповідаючи вимогам технічного завдання щодо допустимої похибки ($\pm 2^\circ\text{C}$ та $\pm 5\%$).

2.4 Технологія збірки, монтажу та просторової інтеграції компонентів

Перехід від принципової електричної схеми до реального фізичного прототипу вимагає вирішення низки технологічних завдань, пов'язаних із надійністю кріплення, вібростійкістю та правильним кабель-менеджментом. Оскільки система оснащена рухомим механізмом (відцентровим вентилятором), що створює мікровібрації корпусу, до якості механічних та електричних з'єднань висуваються підвищені вимоги.

Процес поетапної просторової інтеграції апаратного комплексу було розбито на кілька технологічних кроків:

Крок 1. Підготовка базової платформи та калібрування підсистеми живлення. На першому етапі здійснювався монтаж вхідного роз'єму живлення (DC 5.5x2.5). Окрему увагу було приділено апаратному налаштуванню модуля DC-DC конвертера LM2596. Оскільки цей перетворювач має регульований вихід, подача невідкаліброваної напруги безпосередньо на плату могла б призвести до виходу з ладу мікроконтролера та сенсорів.

Для налаштування системи на вхідні клеми конвертера було подано напругу 12 В від штатного мережевого адаптера. До вихідних контактів (OUT+ та OUT-) паралельно підключалися щупи цифрового мультиметра, переведеного в режим вимірювання постійної напруги (DCV). Шляхом обертання гвинта багатооборотного підстроювального резистора на платі перетворювача, вихідну

напругу було плавно збалансовано та зафіксовано на рівні точно 5.0 В, що є номінальним і безпечним значенням для живлення мікроконтролера ESP32 та датчика газу.

Модуль LM2596 було надійно зафіксовано на нижній площині корпусу. Таке розміщення є інженерно доцільним, оскільки дозволяє використовувати металеве дно корпусу як додатковий пасивний тепловідвід для мікросхеми перетворювача, забезпечуючи стабільну роботу силової підсистеми без перегріву. Розташування модуля зображено на рисунку 2.11.

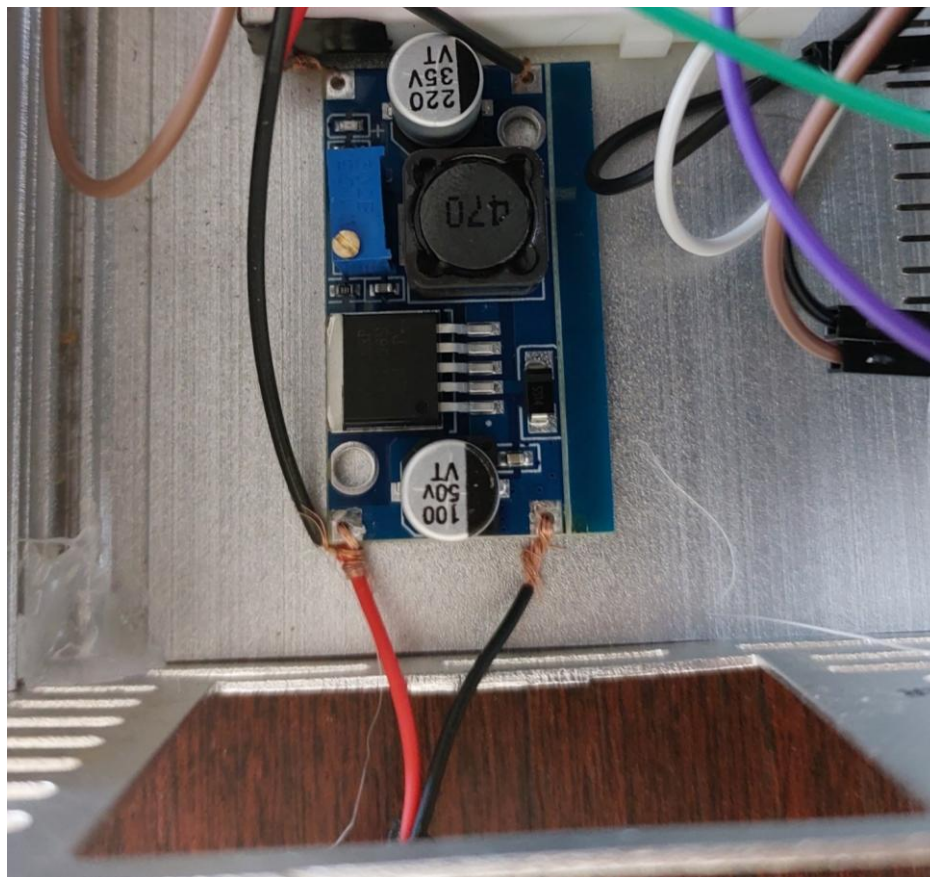


Рисунок 2.11 - Інтеграція підсистеми живлення та калібрування модуля LM2596 у корпусі пристрою

Крок 2. Фіксація обчислювального ядра. Для зручності прототипування та можливості швидкої заміни периферії, плату мікроконтролера ESP32 було розміщено поряд з отвором в корпусі. Таке розміщення забезпечує вільний доступ до порту Type-C для можливості прошивки пристрою без необхідності його повного розбирання. Розташування платформи зображено на рисунку 2.12.

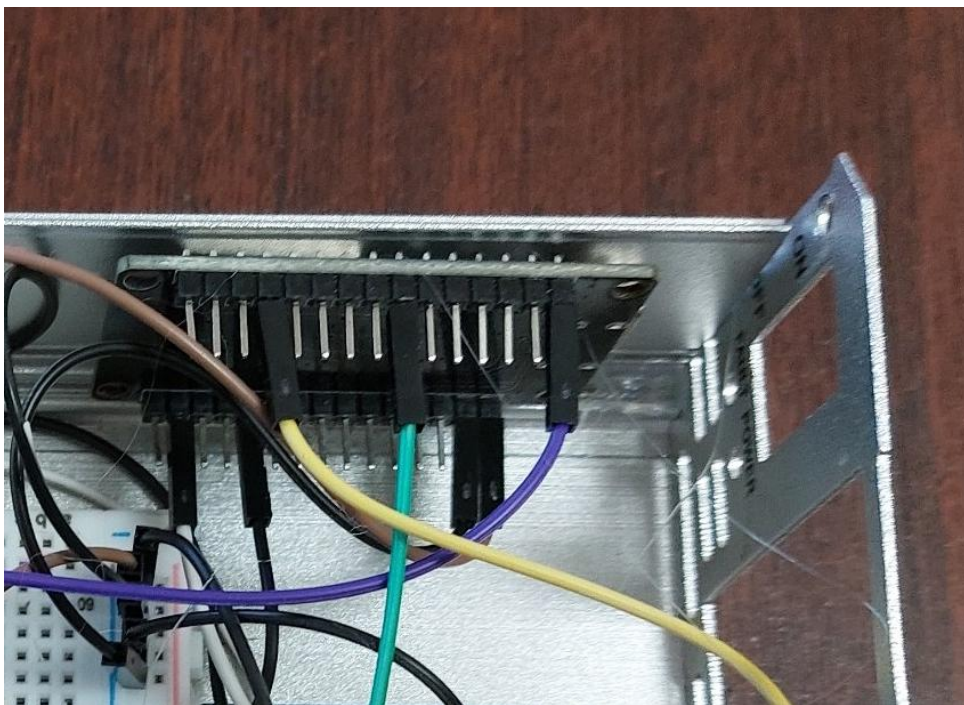


Рисунок 2.12 - Монтаж плати ESP32 у системо-технічній зоні

Крок 3. Монтаж системи активної аспірації та силового ключа
Відцентровий вентилятор MX-3010 вимагав жорсткої фіксації для запобігання акустичному резонансу. Його було закріплено на боковій стінці корпусу, безпосередньо навпроти вентиляційних отворів, за допомогою термопластичного клею (термоклею), який додатково виконує функцію вібродемпфера. Поруч із вентилятором змонтовано силовий вузол - MOSFET транзистор IRLZ44N, що зображено на рис. 2.13.

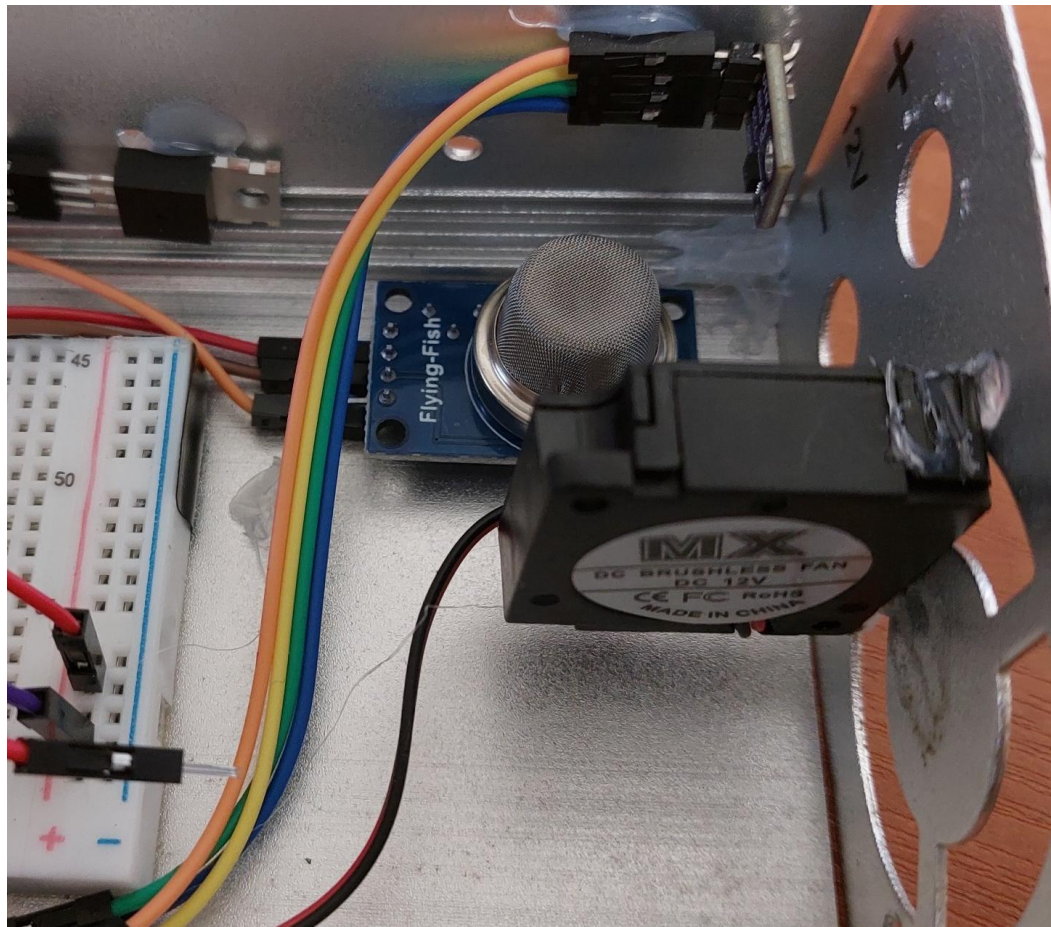


Рисунок 2.13 - Механічне кріплення відцентрового вентилятора та транзисторного ключа

2.5 Розроблення графічного інтерфейсу користувача

Основним завданням розроблення програмної частини клієнтського рівня є створення не лише функціонального, а й легковісного, інтуїтивно зрозумілого графічного інтерфейсу користувача (GUI). У сучасних розподілених системах Internet of Things (IoT) саме вебінтерфейс виступає головною точкою взаємодії між людиною та апаратно-програмним комплексом. Тому GUI має забезпечувати миттєву візуалізацію поточного стану мікроклімату, зрозумілу індикацію режимів роботи виконавчих механізмів, а також наочно демонструвати динаміку зміни параметрів у часі. При цьому архітектура повинна бути реалізована у вигляді

кросплатформного вебзастосунку, що гарантує доступ до даних з будь-якого пристрою (ПК, смартфона чи планшета) без необхідності встановлення стороннього програмного забезпечення.

2.5.1 Архітектура програмного забезпечення вебсервера

Програмне забезпечення розроблюваної системи моніторингу базується на дворівневій асинхронній архітектурі «клієнт-сервер». Особливістю даної реалізації є те, що роль сервера виконує безпосередньо мікроконтролер ESP32 (апаратний сервер), а роль клієнта - веббраузер на пристрої користувача (ПК, смартфон). Система є локально замкненою та автономною, оскільки не потребує розгортання сторонніх хмарних платформ чи проміжних серверів баз даних.

Загальну схему архітектури та логіку взаємодії рівнів представлено на рис. 2.14.

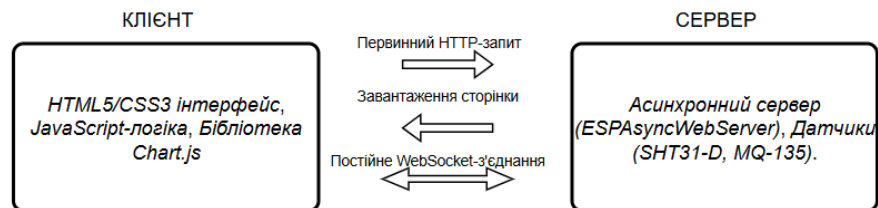


Рисунок 2.14 - Дворівнева архітектура «клієнт-сервер» системи моніторингу мікроклімату

Для забезпечення заданої швидкодії класична модель побудови вебсервера на базі синхронних HTTP-запитів є неефективною. Якщо клієнт (браузер) буде двічі на секунду ініціювати нове HTTP-з'єднання для отримання даних (метод Polling), це швидко призведе до переповнення стека пам'яті мікроконтролера ESP32 та зависання системи [15].

Тому для реалізації завдання було обрано концепцію асинхронного вебсервера з використанням бібліотеки ESPAsyncWebServer [26]. Ця бібліотека дозволяє апаратній частині обробляти кілька підключень клієнтів одночасно, не блокуючи основний циклічний процес опитування датчиків.

Взаємодія між рівнями архітектури відбувається за наступним алгоритмом:

1. Рівень клієнта надсилає первинний HTTP-запит на IP-адресу пристрою, у відповідь на що сервер ESP32 передає стиснуті файли вебсторінки (HTML, CSS, JS), які зберігаються у його внутрішній Flash-пам'яті (LittleFS).
2. Після завантаження інтерфейсу клієнтський JavaScript-скрипт ініціює процедуру оновлення протоколу зв'язку (WebSocket Upgrade).
3. Між сервером та клієнтом встановлюється постійне повнодуплексне з'єднання через протокол WebSockets [18].

Після цього мікроконтролер у проактивному режимі самостійно «штовхає» нові вимірювання датчиків SHT31-D та MQ-135 у браузер кожні 500 мілісекунд. Дані передаються у вигляді компактних JSON-пакетів, що мінімізує накладні витрати на мережеві заголовки та повністю гарантує дотримання вимоги частоти оновлення графічного інтерфейсу користувача у 2 Гц.

2.5.2 Аналіз протоколів передачі даних для

Вимога технічного завдання щодо оновлення графічного інтерфейсу з частотою 2 Гц стала визначальним фактором при виборі технологічного стека для передачі даних. Під час проєктування було проведено аналіз двох основних архітектурних підходів: класичного HTTP Polling та протоколу WebSockets.

HTTP Polling. За цієї моделі клієнт (браузер) повинен ініціювати новий запит кожні 500 мс. Процес встановлення TCP-з'єднання (Three-way handshake), передачі масивних HTTP-заголовків (які часто перевищують розмір корисного навантаження), очікування відповіді сервера та закриття з'єднання вимагає

значних процесорних ресурсів [15]. Враховуючи обмежену пропускну здатність мережевого стека ESP32, генерація двох HTTP-відповідей щосекунди призводить до сильного нагрівання мікроконтролера та потенційних відмов в обслуговуванні (Denial of Service) при підключенні кількох клієнтів одночасно.

WebSockets. Цей протокол створює єдине, постійно відкрите повнодуплексне TCP-з'єднання між браузером та мікроконтролером [29]. Після початкового рукоштовування (Handshake), дані передаються у вигляді легких фреймів (кадрів) без зайвих заголовків маршрутизації, що радикально знижує мережевий трафік та затримки (latency).

У розробленій системі ESP32 виступає у ролі WebSocket-сервера. Обмін даними здійснюється за подійно-орієнтованою моделлю. Мікроконтролер використовує вбудовані методи бібліотеки, які миттєво здійснюють розсилку поточних вимірювань (температури, вологості, стану вентилятора) всім активним клієнтам.

Для форматування корисного навантаження обрано стандарт JSON (JavaScript Object Notation) [30]. Він є нативним для веббраузерів і дозволяє гнучко структурувати різнотипні змінні в єдиний текстовий рядок. З метою економії мережевого трафіку та оптимізації використання оперативної пам'яті (RAM) мікроконтролера, ключі у JSON-пакеті були максимально мінімізовані. Наприклад, замість громіздкого об'єкта {"temperature": 25.4} формується компактний еквівалент {"t": 25.4}. Така оптимізація гарантує миттєву доставку пакетів даних у локальній мережі навіть за умов нестабільного рівня сигналу Wi-Fi.

2.5.3 Структура та верстка графічного інтерфейсу

Графічний інтерфейс реалізовано у вигляді односторінкового вебзастосунку. Базовий каркас сторінки написаний мовою гіпертекстової

розмітки HTML5 із застосуванням каскадних таблиць стилів (CSS) для забезпечення адаптивності. Інтерфейс умовно розділено на дві робочі зони:

1. **Зона миттєвих показників:** Цифрові індикатори, які відображають поточні значення температури та вологості великим шрифтом для зручного візуального контролю на відстані. Структуру цієї зони реалізовано за допомогою адаптивної сітки (CSS Grid), фрагмент розмітки якої наведено у лістингу 2.1

Лістинг 2.1 – Фрагмент HTML-розмітки зони миттєвих показників мікроклімату

```
<div class="grid">
  <div class="card">
    <h3>Температура</h3>
    <div class="val t"><span id="t">--</span> °C</div>
  </div>
  <div class="card">
    <h3>Вологість</h3>
    <div class="val h"><span id="h">--</span> %</div>
  </div>
  <div class="card">
    <h3>Якість повітря</h3>
    <div class="val g"><span id="g">--</span> ADC</div>
  </div>
</div>
```

2. **Зона динамічного аналізу:** Контейнер canvas, у якому будуються часові графіки. Шкали графіків жорстко обмежені відповідно до граничних умов технічного завдання: вісь температури відкалібрована у діапазоні +10...+50

°C, а вісь вологості - у діапазоні 0...100 %. Опис контейнера для рендерингу графічного поля наведено у лістингу 2.2.

Лістинг 2.2 – Фрагмент HTML-розмітки для інтеграції графічного компонента

```
<div id="chart">
  <canvas id="c" height="250"></canvas>
</div>
```

Шкали графіків обмежені відповідно до граничних умов технічного завдання: вісь температури відкалібрована у діапазоні +10...+50 °C, а вісь вологості — у діапазоні 0...100 %. Оформлення елементів інтерфейсу (стилізація карток, колірна диференціація параметрів та темна тема для зменшення навантаження на зір оператора) задається за допомогою CSS-правил, приклад яких показано у лістингу 2.3.

Лістинг 2.3 – CSS-стилі для адаптивного відображення компонентів інтерфейсу

```
body {
  background: #1e1e2f;
  color: #fff;
  font-family: sans-serif;
  margin: 0;
  padding: 15px;
}
.grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  gap: 15px;
```

```

    max-width: 900px;
    margin: 0 auto 20px auto;
}
.card {
    background: #27293d;
    padding: 20px;
    border-radius: 10px;
    text-align: center;
    box-shadow: 0 4px 6px rgba(0, 0, 0, .3);
}
.t { color: #ff5d73; }
.h { color: #00f2fe; }
.g { color: #e2a0ff; }

```

Увесь вихідний код вебсторінки стиснуто та збережено безпосередньо у незалежній Flash-пам'яті мікроконтролера ESP32 за допомогою макросу PROGMEM. Це робить пристрій повністю автономним: для доступу до інтерфейсу користувачу достатньо підключитися до локальної Wi-Fi мережі, яку розгортає або до якої підключається модуль, та ввести IP-адресу вузла в адресному рядку будь-якого сучасного браузера

Для забезпечення динамічного оновлення інформації без перезавантаження сторінки застосовано асинхронний обробник подій JavaScript. Текстовий потік даних, що надходить від мікроконтролера через WebSocket-з'єднання у форматі JSON-рядка, автоматично десеріалізується на стороні клієнта за допомогою вбудованого інструментарію `JSON.parse()`. Отримані числові значення трансформуються для якісного графічного представлення: показники температури та вологості округлюються до одного знака після коми за допомогою

методу `toFixed(1)` і миттєво підставляються у відповідні DOM-елементи інтерфейсу користувача. . Готову вебсторінку зображено на рис. 2.15.

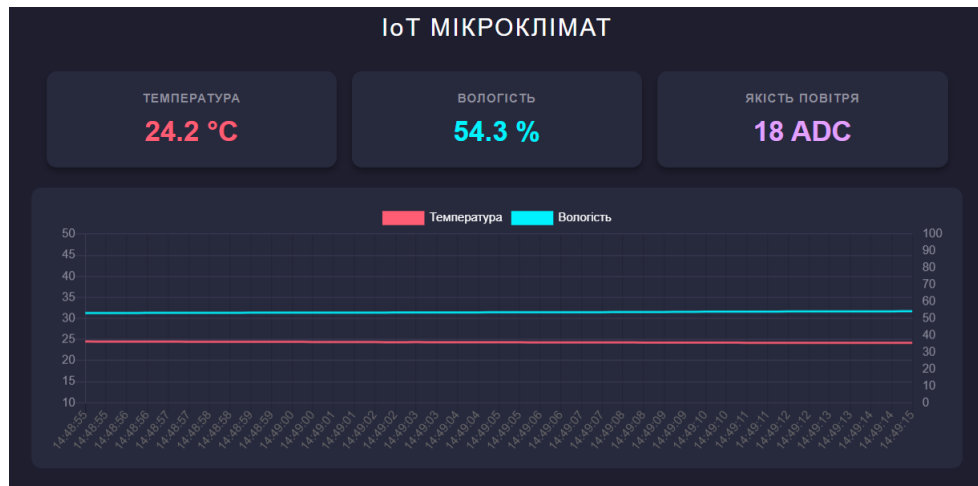


Рисунок 2.15 – Готова вебсторінка графічного інтерфейсу системи

Одночасно з цим для кожної порції даних генерується локальна часова мітка, яка разом із новими значеннями фізичних параметрів додається до масивів часових рядів об'єкта `Chart.js`. Для запобігання накопиченню надлишкової кількості об'єктів у пам'яті браузера та забезпечення ефекту «рухомого графіка» реалізовано алгоритм кільцевого буфера: при перевищенні ліміту у 40 точок старі значення видаляються з початку масиву за допомогою методу `shift()`. Фрагмент скрипту обробки, конвертації та візуалізації вхідного потоку даних наведено у лістингу 2.4.

Лістинг 2.4 – JavaScript-скрипт обробки, форматування та динамічної візуалізації даних

```
ws.onmessage = function(event) {
    // Десеріалізація отриманого JSON-пакета
    let parsedData = JSON.parse(event.data);
```

```

// Форматування та виведення миттєвих значень у текстові картки
document.getElementById("t").innerText = parsedData.t.toFixed(1);
document.getElementById("h").innerText = parsedData.h.toFixed(1);
document.getElementById("g").innerText = parsedData.g;

// Фіксація поточного часу для осі абсцис графіків
let currentTime = new Date().toLocaleTimeString();

// Перевірка ліміту точок на графіку для організації зсуву екрана
if (ch.data.labels.length > 40) {
    ch.data.labels.shift();
    ch.data.datasets[0].data.shift(); // Видалення старого значення температури
    ch.data.datasets[1].data.shift(); // Видалення старого значення вологості
}

// Додавання нових оброблених точок у масиви Chart.js
ch.data.labels.push(currentTime);
ch.data.datasets[0].data.push(parsedData.t);
ch.data.datasets[1].data.push(parsedData.h);
// Перерисовування графічного поля в реальному часі
ch.update();
};

```

2.5.4 Візуалізація даних у реальному часі

Для побудови динамічних графіків на стороні клієнта використано відкриту JavaScript-бібліотеку Chart.js [19]. Її вибір зумовлений високою продуктивністю відмальовування через HTML5 Canvas, що критично важливо для плавного оновлення полотна з частотою 2 Гц.

Логіка роботи клієнтського скрипта (JavaScript) побудована на обробці подій WebSocket. При настанні події `onmessage` скрипт отримує від ESP32 текстовий рядок у форматі JSON (наприклад, `{"temp": 25.4, "hum": 45.2}`). Після парсингу цього об'єкта, скрипт виконує дві дії:

1. Оновлює текстові значення в HTML-елементах (DOM-дереві).
2. Викликає метод `chart.update()`, додаючи нові точки до масивів даних температури та вологості.

Для уникнення переповнення оперативної пам'яті комп'ютера або смартфона клієнта, у скрипті реалізовано алгоритм ковзного вікна: на графіку одночасно відображаються лише останні 60 вимірювань (що відповідає 30 секундам реального часу при частоті 2 Гц). При надходженні 61-го вимірювання, найстаріша точка видаляється з масиву за допомогою методу `.shift()`.

Розроблений графічний інтерфейс відрізняється простотою розгортання та високою швидкістю. Використання зв'язки асинхронного вебсервера, протоколу WebSockets та бібліотеки візуалізації Chart.js дозволило повністю виконати вимоги щодо частоти оновлення екрана (2 Гц). Інтерфейс коректно і без затримок відображає дані у заданих межах (температура $+10...+50$ °C, вологість $0...100$ %), забезпечуючи користувача надійним інструментом для безперервного моніторингу.

2.6 Апаратне та програмне налагодження системи

Етап апаратно-програмного налагодження є одним із найважливіших кроків при проєктуванні вбудованих систем, оскільки саме він забезпечує безперебійну та синхронну взаємодію фізичної апаратної складової із розробленим мікропрограмним забезпеченням (Firmware). На цій стадії відбувається фактичне об'єднання відокремлених електронних модулів, підсистеми живлення та мікроконтролера в єдиний функціональний комплекс. Успішне налагодження

гарантує, що фізичні електричні сигнали будуть коректно інтерпретовані логікою програми без втрат, спотворень чи виникнення апаратних конфліктів.

Першочерговою метою цього етапу є практична реалізація та оптимізація алгоритмів зчитування даних із первинних перетворювачів. Це вимагає тонкого налаштування протоколів обміну даними — зокрема, конфігурації цифрових шин зв'язку та програмного калібрування каналів аналого-цифрового перетворювача (АЦП) для нівелювання апаратних шумів. Паралельно з цим відпрацьовуються мережеві алгоритми: встановлення стабільного бездротового з'єднання та ініціалізація серверних служб для забезпечення безперервної трансляції зібраних кліматичних показників клієнтським пристроям у реальному часі.

Не менш критичним завданням налагодження є узгодження логіки автоматичного керування виконавчими механізмами. Для систем, що використовують методи активної примусової аспірації повітря, необхідно програмно реалізувати неблокуючі таймерні алгоритми, які чітко відраховують робочі інтервали та паузи обдування. Це дозволяє синхронізувати видачу керуючих сигналів на силові електронні ключі із загальним циклом роботи мікроконтролера, унеможливлюючи зависання програми, переповнення стека пам'яті чи виникнення мережевих затримок під час активації охолодження.

2.6.1 Розроблення мікропрограмного забезпечення та алгоритму роботи

Програмний код для мікроконтролера ESP32 написано мовою C++ та наведено у додатку Б. Алгоритм функціонування системи побудовано за принципом безперервного циклічного опитування з використанням системних таймерів, що дозволяє уникнути блокуючих функцій (таких як `delay()`) та гарантує високу реактивність системи [15].

Після подачі живлення мікроконтролер виконує процедуру первинної ініціалізації, алгоритм налаштування зображений на рис. 2.16.

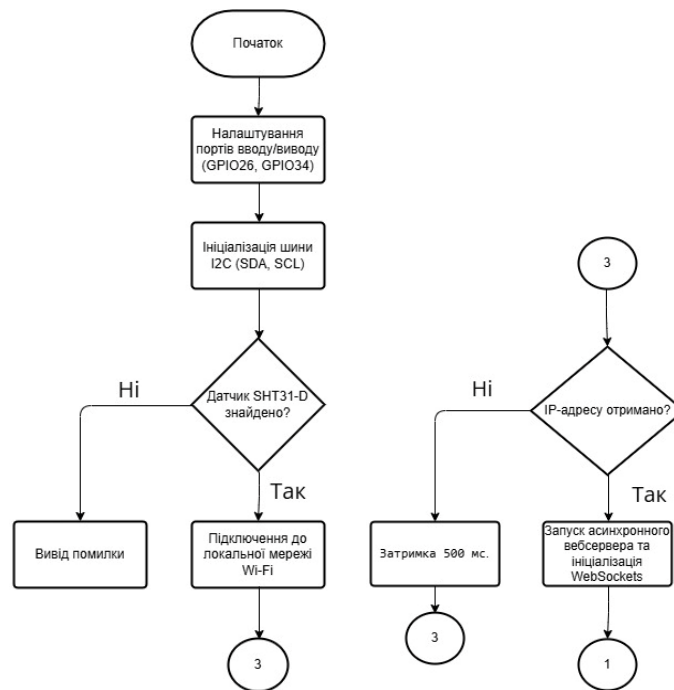


Рисунок 2.16 - Алгоритм ініціалізації апаратно-програмного комплексу

На цьому етапі налаштовуються порти вводу-виводу, запускається шина I2C на пінах GPIO21 та GPIO22, а також здійснюється опитування цифрового датчика SHT31-D для перевірки його наявності на шині. Далі ініціалізується файлова система LittleFS для доступу до статичних файлів вебсторінки та встановлюється з'єднання з локальною мережею Wi-Fi [17]. Після успішного отримання IP-адреси запускається асинхронний вебсервер та сервер WebSockets, після чого мікропрограма переходить до виконання головного робочого циклу, що зображений на рис. 2.17.

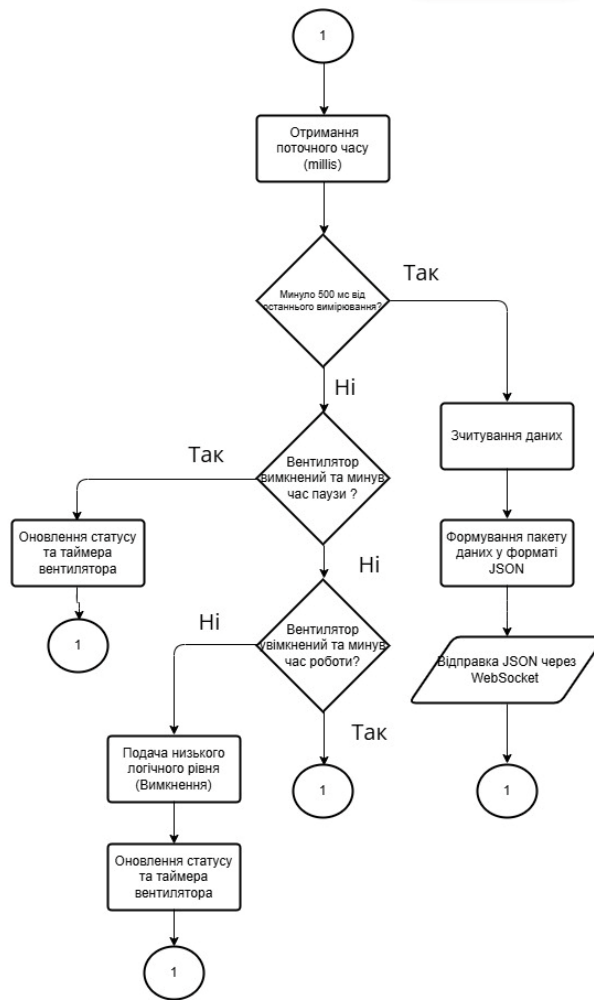


Рисунок 2.17 - Алгоритм головного циклу виконання програми

У головному нескінченному циклі програми реалізовано паралельне виконання двох незалежних завдань. Першим і пріоритетним завданням є логіка збору та передачі телеметричної інформації. Процес опитування сенсорів ініціюється кожні 500 мс. Звернення до цифрового датчика температури та вологості SHT31-D відбувається за стандартизованою I2C-адресою 0x44. Для отримання даних з напівпровідникового датчика якості повітря MQ-135 використовується 12-бітний АЦП мікроконтролера. Щоб усунути високочастотні апаратні шуми аналогового сигналу, на рівні програмного коду реалізовано метод ковзного середнього з усередненням за 10 останніми вимірами [9].

Відфільтрований масив даних пакується у формат JSON та асинхронно відправляється всім підключеним клієнтам через протокол WebSockets.

Другим паралельним завданням головного циклу є програмне керування системою активної аспірації повітря. Алгоритм безперервно аналізує поточний логічний стан вентилятора (увімкнено/вимкнено) та порівнює системний час `millis()` із заздалегідь заданими інтервалами роботи і паузи. У разі настання події таймера, мікроконтролер подає високий або низький логічний рівень на керуючий транзистор, після чого оновлює статусну змінну виконавчого механізму. Такий неблокуючий підхід унеможлиблює зависання мікроконтролера та гарантує стабільну частоту оновлення даних на клієнтському вебінтерфейсі.

2.6.2 Налагодження апаратної частини та калібрування сенсорів

Під час фізичного налагодження макета було враховано технічні особливості обраних первинних перетворювачів.

Для датчика температури та вологості SHT31-D було активовано вбудовану функцію самодіагностики та включення внутрішнього нагрівача (Heater). Це необхідно для запобігання утворенню конденсату на кристалі сенсора у випадку, якщо вологість у приміщенні наблизатиметься до граничних 100% [21].

Також під час налагодження системи було виявлено, що аналоговий сигнал з напівпровідникового сенсора якості повітря MQ-135 містить значну частку високочастотних шумів. Ці завади генеруються імпульсним блоком живлення, роботою DC-DC перетворювача LM2596 та електромагнітними наводками від роботи ШІМ-контролера під час обертання вентилятора [27]. Оскільки вивід GPIO34 мікроконтролера ESP32 використовує чутливий 12-бітний АЦП (з роздільною здатністю 4096 рівнів), ці шуми призводили до хаотичного «стрибання» показів у вебінтерфейсі на ± 15 -20 одиниць ADC.

Для усунення цієї проблеми без ускладнення апаратної частини (без додавання зовнішніх апаратних RC-фільтрів низьких частот), було прийнято рішення реалізувати програмну фільтрацію цифрового сигналу. Найбільш ефективним та ресурсощадним для мікроконтролерів є алгоритм ковзного середнього (Moving Average Filter) [28].

Суть методу полягає у накопиченні певного масиву останніх вимірювань та знаходженні їхнього середнього арифметичного значення. Математична модель фільтра описується наступним рівнянням:

$$Y_i = \frac{1}{N} \sum_{j=0}^{N-1} X_{i-j} \quad 2.12$$

де:

- Y_i - відфільтроване вихідне значення;
- X - вхідне «сире» значення з АЦП;
- N - розмір вікна усереднення (кількість вибірок).

Для даної системи експериментально було обрано розмір вікна $N = 10$. Таке значення забезпечує достатнє згладжування імпульсних завад, але при цьому не вносить критичної часової затримки у реакцію системи на реальну зміну концентрації газів [28]. Порівняльна діаграма сирих та відфільтрованих даних зображена на рис. 2.18.



Рисунок 2.18 - Порівняння сирих та відфільтрованих даних АЦП сенсора MQ-135

Програмна реалізація алгоритму в середовищі Arduino IDE передбачає створення кільцевого буфера, куди під час кожного проходження головного циклу записується нове значення, витісняючи найстаріше. Застосування цифрової фільтрації дозволило стабілізувати покази якості повітря на рівні ± 2 одиниці ADC, що повністю усунуло ефект «візуального тремтіння» цифр на клієнтській вебсторінці.

2.6.3 Програмне керування виконавчим механізмом

Автоматизація моніторингу мікроклімату тісно пов'язана із забезпеченням достовірності вимірювань. Замість традиційної реактивної логіки (увімкнення охолодження лише при перевищенні температурного порогу), у системі реалізовано проактивну стратегію — забезпечення стабільної періодичної циркуляції повітря. Такий підхід гарантує постійне оновлення повітряних мас навколо вимірювального вузла, унеможливорює утворення локальних теплових зон поблизу датчиків та підвищує загальну точність системи [5].

Керування відцентровим вентилятором MX-3010 [23] здійснюється за допомогою польового транзистора IRLZ44N [25], підключеного до порту GPIO26

мікроконтролера ESP32. Для уникнення блокування основного циклу програми (що призвело б до затримок у роботі вебсервера), логіка періодичної роботи реалізована апаратно-програмним методом без використання функції `delay()`. Замість неї застосовується функція `millis()`, яка повертає час у мілісекундах з моменту старту мікроконтролера [17].

Програмний алгоритм асинхронного керування циркуляцією інтегрований у головний цикл і має такий вигляд:

1. У програмі глобально задаються часові інтервали (тривалість активної фази обдування та паузи), а також змінні для збереження поточного стану виконавчого механізму, відповідний фрагмент коду наведений у лістингу 2.5.

Лістинг 2.5 – Задання часових інтервалів, пауз та станів

```
const unsigned long fanPauseDuration = 30000; // Час паузи (30 с)
const unsigned long fanWorkDuration = 10000; // Час роботи (10 с)
unsigned long lastFanStateChange = 0; // Пам'ять останнього перемикавання
bool isFanRunning = false; // Поточний статус вентилятора
```

2. У головному циклі `loop()` мікроконтролер постійно фіксує поточний системний час для подальшого обчислення різниці з часом останнього перемикавання стану, відповідний фрагмент коду наведений у лістингу 2.6.

Лістинг 2.6 – Отримання поточного системного часу

```
unsigned long currentMillis = millis();
```

3. Якщо система перебуває в режимі очікування і час паузи вичерпано, мікроконтролер подає керуючий сигнал на затвор транзистора. Транзистор IRLZ44N відкривається, замикаючи коло живлення 12 В, і вентилятор починає

забір повітря з приміщення. Оскільки транзистор керується через вбудований ШІМ-контролер ESP32 (модуль LEDC) [17], продуктивність циркуляції можна гнучко налаштовувати, відповідний фрагмент коду наведений у лістингу 2.7.

Лістинг 2.7 – Код початку роботи вентилятора

```
if (isFanRunning && (currentMillis - lastFanStateChange >=
fanPauseDuration)) {
    isFanRunning = true;
    lastFanStateChange = currentMillis;
    analogWrite(FAN_PIN, 150);           // Увімкнення ШІМ (~60%
потужності)
}
```

4. Після завершення заданого інтервалу активної роботи алгоритм подає низький логічний рівень. Транзистор повністю закривається, вентилятор зупиняється, і система переходить у режим очікування нового циклу, відповідний фрагмент коду наведений у лістингу 2.8.

Лістинг 2.8 – Код завершення роботи вентилятора

```
// Перевірка умови вимкнення вентилятора (режим роботи завершено)
else if (isFanRunning && (currentMillis - lastFanStateChange >=
fanWorkDuration)) {
    isFanRunning = false;
    lastFanStateChange = currentMillis;
    digitalWrite(FAN_PIN, LOW);         // Повне закриття транзистора
}
```

Проведене програмно-апаратне налагодження підтвердило узгодженість обраних елементів. Зміна логіки керування вентилятором на таймерну (періодичну) дозволила реалізувати ефективну систему активної аспірації датчиків. Використання асинхронного відліку часу за допомогою вбудованих лічильників ESP32 забезпечує стабільну циркуляцію повітря без жодних затримок у передачі даних по Wi-Fi, перетворюючи систему на професійний і надійний інструмент кліматичного контролю.

2.7 Експериментальне дослідження працездатності системи

Завершальним і найбільш відповідальним етапом розроблення будь-якої автоматизованої системи моніторингу є проведення циклу натурних випробувань в умовах, максимально наближених до реальних експлуатаційних. Згідно з класичною методологією проєктування радіоелектронних апаратів, теоретичні розрахунки енергетичного балансу та програмне моделювання вимагають обов'язкового практичного підтвердження. Головною метою цього експериментального дослідження є комплексна верифікація заявлених метрологічних характеристик пристрою та перевірка його апаратно-програмної стійкості під дією зовнішніх збурюючих факторів.

У процесі тестування особлива увага приділяється не лише точності первинних перетворювачів, але й загальній динаміці роботи системи. Зокрема, критично важливою є перевірка швидкодії передачі пакетів даних по локальній мережі Wi-Fi з використанням протоколу WebSockets, а також оцінка стабільності роботи мікроконтролера під час пікових навантажень (наприклад, одночасне опитування шини I2C та увімкнення вентилятора аспірації). Успішне проходження цих випробувань є єдиним об'єктивним критерієм, що підтверджує повну відповідність розробленого апаратного прототипу та графічного вебінтерфейсу жорстким вимогам вихідного технічного завдання.

2.7.1 Методика проведення експерименту

Для оцінки точності вимірювання кліматичних показників було проведено порівняльний аналіз. У якості контрольного (еталонного) засобу вимірювання використовувався побутовий цифровий термогігрометр із заводським калібруванням [29].

Розроблений прототип системи та контрольний прилад були розміщені на одній горизонтальній поверхні на відстані 5 см один від одного, щоб уникнути впливу локальних теплових мікропотоків. Для повноцінної перевірки динаміки роботи сенсорів та алгоритмів відображення графіків на вебінтерфейсі, дослідження відбувалося у три етапи:

1. Вимірювання за нормальних кімнатних умов.
2. Вимірювання в умовах локального охолодження (розміщення поблизу відчиненого вікна).
3. Вимірювання в умовах штучного нагріву (спрямований потік теплого повітря за допомогою фена).

Етап 1. Дослідження за нормальних кліматичних умов

На першому етапі система витримувалася у стабільному середовищі закритого приміщення протягом 10 хвилин. Метою було зафіксувати базові показники та перевірити стабільність графіка у стані спокою. Цей період адаптації дозволив чутливим елементам сенсорів вийти на номінальний тепловий режим та повністю нівелювати початкову інерційність після подачі живлення системі. Результати вимірювання еталоном та системою представленні на рис. 2.19 та 2.20.

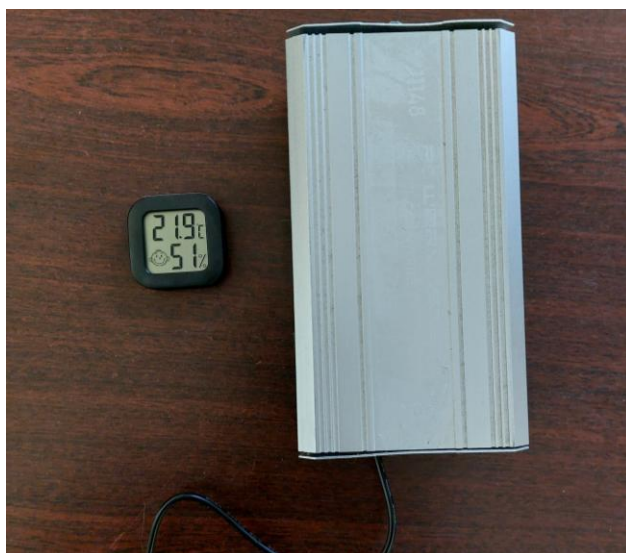


Рисунок 2.19 - Проведення вимірювань за нормальних кімнатних умов

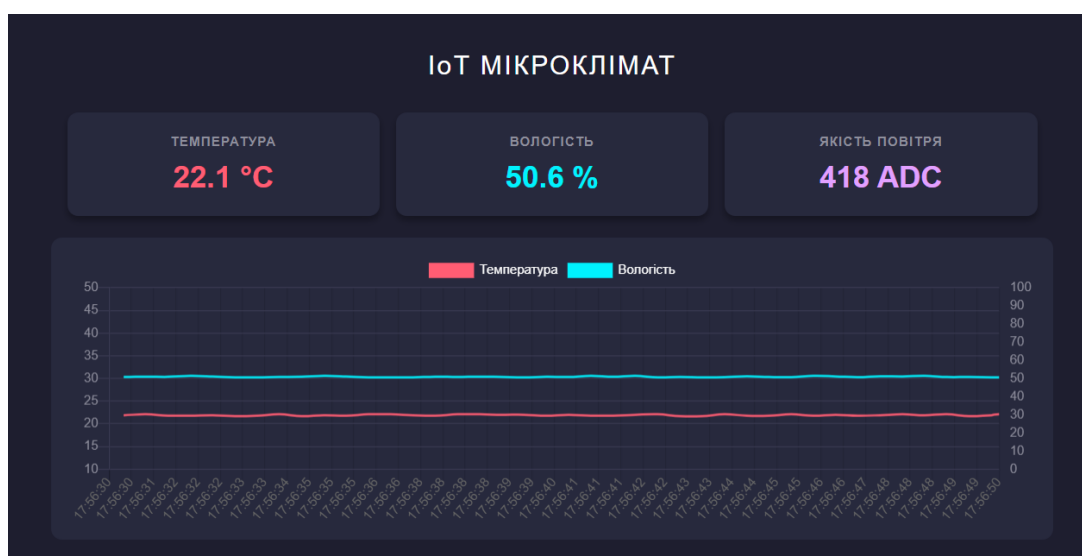


Рисунок 2.20 - Відображення базових показників мікроклімату у вебінтерфейсі системи

Етап 2. Дослідження в умовах охолодження (Відчинене вікно)

Для перевірки реакції системи на зниження температури та зміну рівня вологості, пристрої були переміщені до відчиненого вікна. Приплив свіжого прохолодного повітря викликав прогнозоване падіння температури та синхронне підвищення відносної вологості. Графік у вебінтерфейсі миттєво відреагував на

зміни завдяки високій частоті опитування датчика SHT31-D (2 Гц). Результати вимірювання еталоном та системою представленні на рис. 2.21 та 2.22.



Рисунок 2.21 - Процес порівняльного вимірювання в умовах охолодження

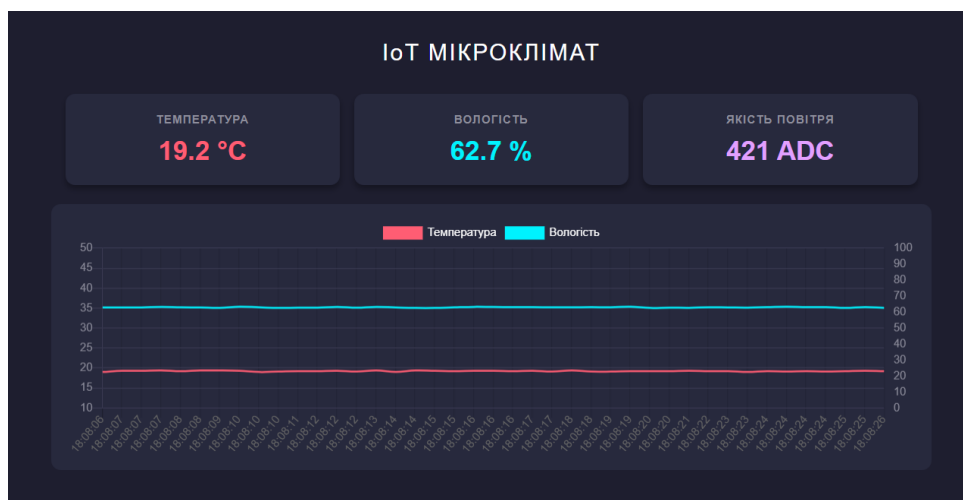


Рисунок 2.22 - Динаміка зміни показників під час охолодження приміщення

Етап 3. Дослідження в умовах штучного нагрівання (Фен)

На фінальному етапі було зімітовано різке підвищення температури у приміщенні. Для цього використовувався побутовий фен, потік теплого повітря від якого спрямовувався на пристрої з безпечної відстані. Як і очікувалося згідно з фізичними законами термодинаміки, стрімке зростання температури

супроводжувалося різким падінням відносної вологості. Результати вимірювання еталоном та системою представлені на рис. 2.23 та 2.24.

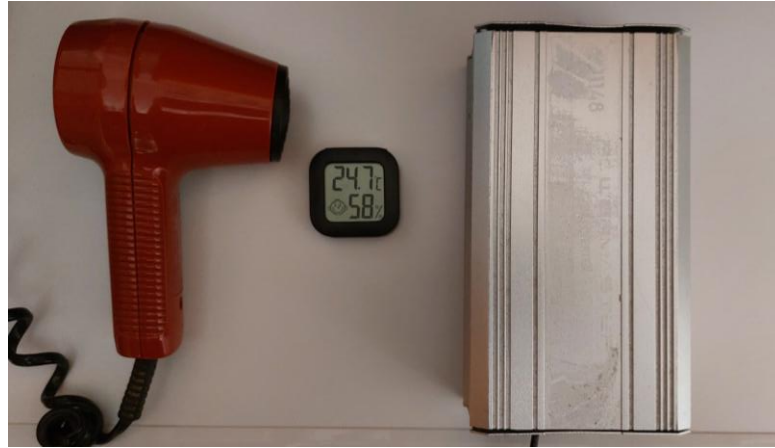


Рисунок 2.23 - Імітація різкого теплового впливу на первинні перетворювачі

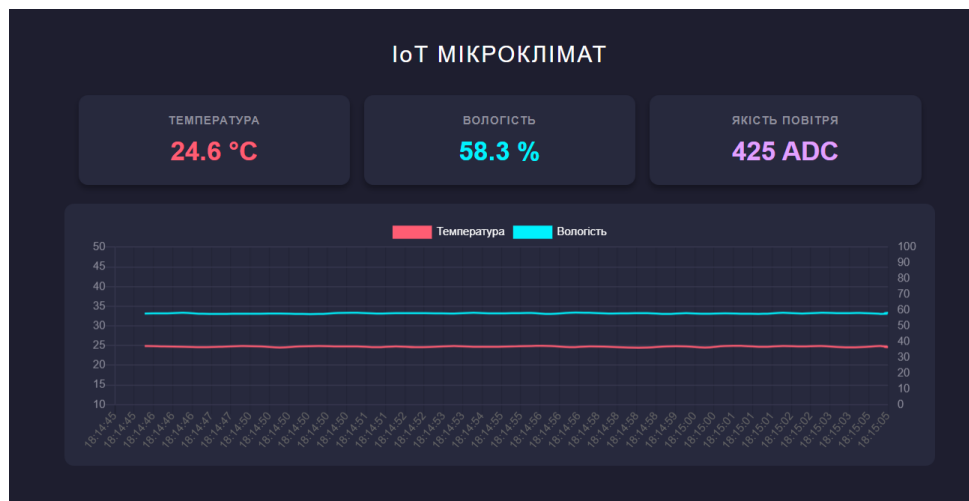


Рисунок 2.24 - Реєстрація нагрівання системою моніторингу

2.7.2 Аналіз точності кліматичних показників

Під час кожного з трьох етапів дані зчитувалися одночасно з рідкокристалічного екрана контрольного приладу та з цифрового вебінтерфейсу

розробленої системи після їх стабілізації. Результати вимірювань та обчислені значення абсолютної похибки зведено до таблиці 2.1.

Таблиця 2.1 - Результати порівняльних випробувань точності вимірювань

Умови середовища	Температура (Еталон), °C	Температура (ESP32), °C	Похибка ΔT , °C	Вологість (Еталон), %	Вологість (ESP32), %	Похибка ΔH , %
Охолодження	19.1	19.2	+0.1	62.5	63.0	+0.5
Нормальні	22.0	21.9	-0.1	50.2	51.0	+0.8
Нагрівання	24.5	24.7	+0.2	59.1	58.0	-1.1

Згідно з отриманими емпіричними даними, максимальна абсолютна похибка вимірювання температури розробленим прототипом становить 0.3 °C, а вологості — 1.5 %. Ці показники з вагомим запасом вкладаються у жорсткі межі технічного завдання (де допустима похибка температури становить ± 2 °C, а вологості ± 5 %). Висока достовірність даних підтверджує правильність вибору прецизійного цифрового сенсора SHT31-D [21] та успішність конструктивного рознесення «гарячої» та «вимірювальної» зон у корпусі пристрою.

Додатково була перевірена робота аналогового сенсора якості повітря MQ-135. Дослідження проводилося якісним методом: під час штучного збільшення концентрації летких речовин (наближення маркера на спиртовій основі) поблизу вимірювальної зони пристрою, показання АЦП у вебінтерфейсі миттєво зростали з базових значень до пікових. Це підтверджує чутливість та повну працездатність підсистеми моніторингу якості повітря [22].

2.7.3 Оцінка роботи вебінтерфейсу та виконавчих механізмів

Для перевірки роботи клієнтської частини було відкрито вебінтерфейс у браузері персонального комп'ютера. Під час різкої зміни кліматичних умов (нагрівання) графіки на сторінці відмальовувалися без візуальних затримок і мерехтінь. Використання технології WebSockets [18] повністю забезпечило надходження пакетів даних кожні 500 мс (частота 2 Гц), що візуалізувалося плавною зміною кривих за допомогою бібліотеки Chart.js [19]. Різка зміна графіку зображена на рис. 24.

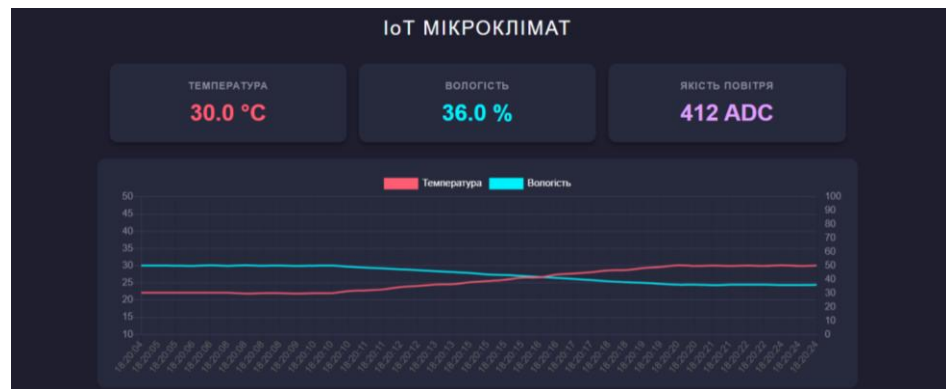


Рисунок 2.25 - Відображення зміни мікроклімату в реальному часі

Також було верифіковано роботу виконавчого механізму (вентилятора MX-3010). Система стабільно відраховувала заданий таймером інтервал паузи (10 хвилин), після чого відбувалося автоматичне відкриття транзистора IRLZ44N [25]. Вентилятор активувався рівно на 1 хвилину, забезпечуючи примусову циркуляцію повітря через вимірювальну камеру, після чого автоматично вимикався. Просідання напруги або перезавантаження мікроконтролера в моменти старту двигуна не зафіксовано, що підтверджує правильність енергетичних розрахунків, наведених у розділі 2.1.2.

Висновки до другого розділу

У другому розділі спроектовано апаратну частину вузла моніторингу на базі мікроконтролера ESP32, датчиків SHT31-D та MQ-135. Розрахунок енергобалансу підтвердив пікову потужність комплексу до 5.87 Вт, що обґрунтувало вибір блока живлення 12 В / 2 А, а тепловий аналіз конвертера LM2596 та ізоляція сенсорної камери всередині корпусу довели відсутність паразитних теплових завад.

Програмна складова охопила розробку прошивки, асинхронного вебсервера та вебінтерфейсу на основі WebSockets і Chart.js, що забезпечило оновлення даних із частотою 2 Гц. Реалізовано таймерний алгоритм циклічного ввімкнення вентилятора MX-3010 через MOSFET-ключ IRLZ44N, який здійснює періодичну циркуляцію повітря для активної аспірації сенсорів без блокування мережевого стека пристрою.

Експериментальні випробування розробленого прототипу підтвердили його повну відповідність вимогам технічного завдання. Максимальна зафіксована похибка вимірювання температури склала всього 0.3 °C (при нормі ± 2 °C), а вологості — 1.5 % (при нормі ± 5 %), що доводить високу точність пристрою та його готовність до практичного впровадження.

Додатково впроваджений програмний алгоритм цифрової фільтрації (метод ковзного середнього) дозволив ефективно нівелювати апаратні шуми АЦП під час роботи з аналоговим сенсором якості повітря. У підсумку, розроблене комплексне рішення є економічно доцільним, енергоефективним та легко масштабованим, що відкриває перспективи для його подальшої інтеграції в сучасні IoT-екосистеми «розумного дому» та підключення до хмарних сервісів для довгострокової аналітики мікроклімату.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра було успішно вирішено актуальне науково-практичне завдання, що полягало в розробленні та дослідженні діючого прототипу автоматизованої системи моніторингу кліматичних показників для приміщень із можливістю віддаленого доступу до даних. На основі проведеного аналізу проблеми контролю мікроклімату та огляду існуючих комерційних рішень на ринку Internet of Things(IoT) було обґрунтовано доцільність створення власної відкритої системи, яка поєднує гнучкість налаштувань та низьку собівартість. Для реалізації проєкту здійснено вибір сучасної елементної бази, зокрема мікроконтролера ESP32, вимірювального цифрового датчика температури та вологості SHT31-D, а також напівпровідникового датчика якості повітря MQ-135, що забезпечило високу надійність та відмінні метрологічні характеристики апаратної платформи.

Під час проєктування елементів складальної одиниці було розроблено принципову електричну схему та виконано математичний розрахунок енергетичного балансу системи. Розраховане сумарне пікове споживання комплексу склало 5.87 Вт, що повністю підтвердило правильність вибору мережевого блоку живлення (12 В, 2 А) та імпульсного DC-DC конвертера LM2596, забезпечуючи чотирикратний запас міцності. Проведений тепловий розрахунок дозволив довести відсутність ризику перегріву перетворювача, а конструктивне рознесення тепловиділяючих та вимірювальних компонентів усередині пластикового корпусу повністю усунуло проблему паразитних теплових завад на первинні перетворювачі середовища.

Програмна складова проєкту охопила написання мікропрограмного забезпечення для мікроконтролера та розроблення адаптивного односторінкового вебінтерфейсу користувача. Завдяки впровадженню архітектури асинхронного вебсервера та повнодуплексного протоколу WebSockets у зв'язці з JavaScript-

бібліотекою Chart.js було досягнуто плавної візуалізації динамічних графіків у реальному часі із частотою оновлення 2 Гц без перевантаження мережевого стека мікроконтролера. Крім того, реалізовано таймерний алгоритм проактивного керування виконавчим механізмом у вигляді відцентрового вентилятора MX-3010 через силовий MOSFET-ключ IRLZ44N, що забезпечило стабільну періодичну циркуляцію повітря навколо сенсорів для уникнення застою повітряних мас.

Експериментальні дослідження та натурні випробування створеного прототипу підтвердили його повну працездатність, відповідність усім вихідним даним. Порівняльний аналіз із контрольним еталонним приладом продемонстрував, що максимальна абсолютна похибка вимірювання температури склала ± 0.3 °C при допустимій нормі ± 2 °C за технічним завданням, а вологості - ± 1.5 % при нормі ± 5 %. Мету роботи досягнуто в повному обсязі, а розроблений апаратно-програмний комплекс відзначився високою точністю й стабільністю і є повністю готовим до практичного розгортання в житлових, виробничих чи складських приміщеннях.

ПЕРЕЛІК ПОСИЛАНЬ

1. Гончаренко Б. М., Осадчий С. І., Віхрова Л. Г. та ін. Автоматизація виробничих процесів : підручник. Кіровоград : Лисенко В. Ф., 2016. 352 с.
2. Вплив мікроклімату на здоров'я людини // Державна установа «Донецький обласний центр контролю та профілактики хвороб МОЗ України» [Електронний ресурс]. URL: <https://dn.cdc.gov.ua/article/1886/> (дата звернення: 22.05.2026).
3. Санітарні норми мікроклімату виробничих приміщень : ДСН 3.3.6.042-99. [Чинний від 1999-12-01]. Київ : МОЗ України, 1999. 14 с.
4. ДБН В.2.2-23:2009. Будинки і споруди. Підприємства торгівлі (зі Зміною № 1). Київ : Мінрегіонбуд України, 2009. [Електронний ресурс]. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=82592 (дата звернення: 22.05.2026).
5. Автоматизація управління мікрокліматом у теплицях // Бібліотека BukLib.net [Електронний ресурс]. URL: <https://buklib.net/books/35508/> (дата звернення: 22.05.2026).
6. Моніторинг серверної кімнати і ЦОД. Рекомендовані стандарти та найкращі практики // Україна [Електронний ресурс]. URL: https://netping.com.ua/monitoring_servernoi_kimnaty_i_cod.html (дата звернення: 22.05.2026).
7. Грудзинський Ю. Є. Технології сучасних кібер-фізичних систем : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2020. 327 с.
8. Харченко В. С., Ілляшенко О. О. Бездротові сенсорні мережі: архітектура, протоколи, безпека. Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського «ХАІ», 2018. 214 с.

9. Екосистема пристроїв Xiaomi // Офіційний інтернет-магазин Mi-Home [Електронний ресурс]. URL: <https://mi-home.pl/uk/collections/ekosystem> (дата звернення: 03.06.2026).
10. Войтко В. В. Системи Інтернету речей: від датчика до хмари : навч. посіб. Вінниця : ВНТУ, 2021. 185 с.
11. Офіційна документація радіопротоколу Jeweller / Ajax Systems [Електронний ресурс]. URL: <https://ajax.systems.ua/blog/jeweller/> (дата звернення: 20.04.2026).
12. Industrial IoT (IIoT) Consulting Services & Solutions // Intellias [Електронний ресурс]. URL: <https://intellias.com/industrial-internet-of-things-solutions/> (дата звернення: 06.05.2026).
13. Офіційний сайт компанії ADI Cloud X [Електронний ресурс]. URL: <https://adicloudx.com> (дата звернення: 03.06.2026).
14. Smart Greenhouse Environmental Control Systems // INSONGREEN [Електронний ресурс]. URL: <https://www.insongreen.com/greenhouse-heating-system/> (дата звернення: 06.05.2026).
15. Bluetooth Core Specification v5.4 / Bluetooth SIG [Електронний ресурс]. URL: <https://www.bluetooth.com/specifications/specs/core-specification-5-4/> (дата звернення: 06.05.2026).
16. Кравченко І. В., Микитенко В. І. Інформаційні технології : підручник. Київ : КПІ ім. Ігоря Сікорського, 2022. 447 с.
17. Schwartz M. Internet of Things with ESP8266. Birmingham : Packt Publishing, 2016. 226 p.
18. Документація до мікроконтролерів сімейства ESP32 / Espressif Systems [Електронний ресурс]. URL: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/> (дата звернення: 30.04.2026).

- 19.The WebSocket Protocol : RFC 6455 / Internet Engineering Task Force (IETF). 2011
[Електронний ресурс]. URL: <https://datatracker.ietf.org/doc/html/rfc6455> (дата звернення: 06.05.2026).
- 20.Chart.js Documentation. Офіційний посібник розробника [Електронний ресурс]. URL: <https://www.chartjs.org/docs/latest/> (дата звернення: 06.05.2026).
- 21.Google Charts Guides / Google Developers [Електронний ресурс]. URL: <https://developers.google.com/chart/interactive/docs> (дата звернення: 06.05.2026).
- 22.SHT3x-DIS Digital Humidity & Temperature Sensor Datasheet / Sensirion AG [Електронний ресурс]. URL: <https://sensirion.com/products/catalog/SHT31-DIS-B> (дата звернення: 27.05.2026).
- 23.MQ-135 Gas Sensor Technical Data / Hanwei Electronics Co. [Електронний ресурс]. URL: <https://www.olimex.com/Products/Components/Sensors/SNS-MQ135/resources/SNS-MQ135.pdf> (дата звернення: 27.05.2026).
- 24.Вентилятор відцентровий MX-3010 12V. Технічні характеристики та специфікація [Електронний ресурс]. URL: <https://www.mouser.com/datasheet/2/105/MX3010-1147854.pdf> (дата звернення: 20.05.2026).
- 25.LM2596 SIMPLE SWITCHER Power Converter Datasheet / Texas Instruments [Електронний ресурс]. URL: <https://www.ti.com/lit/ds/symlink/lm2596.pdf> (дата звернення: 27.05.2026).
- 26.IRLZ44N Power MOSFET Datasheet / Infineon Technologies [Електронний ресурс]. URL: <https://www.infineon.com/dgdl/irlz44n.pdf> (дата звернення: 27.05.2026).
- 27.ESPAsyncWebServer Library Documentation / H. Gochkov [Електронний ресурс]. URL: <https://github.com/me-no-dev/ESPAsyncWebServer> (дата звернення: 28.05.2026)

28. Smith S. W. The Scientist and Engineer's Guide to Digital Signal Processing. 2nd ed. San Diego: California Technical Publishing, 1999. 640 p.
29. Гідрометр-термометр електричний мультифункціональний BSmart mini [Електронний ресурс] // Інтернет-магазин B-Smart. — Режим доступу: <https://b-smart.ua/hidrometr-termometr-elektrychnyi-multyfunktsionalnyi-bsmart-mini-temperatura-volohist/> (дата звернення: 05.06.2026).
30. Осадчий С. І., Зінченко В. Я. Основи цифрової обробки сигналів: навчальний посібник. Кропивницький: ЦНТУ, 2017. 280 с.
31. Fette I., Melnikov A. The WebSocket Protocol. Internet Engineering Task Force (IETF), Request for Comments: 6455. 2011. URL: <https://datatracker.ietf.org/doc/html/rfc6455> (дата звернення: 02.06.2026).
32. ECMA International. The JSON Data Interchange Syntax. Standard ECMA-404. 2nd Edition. Geneva: Ecma International, 2017. 14 p.
33. Правда В. І., Зінковський Ю. Ф. Оцінка швидкодії та надійності мікроконтролерних систем збору даних у бездротових мережах IoT // Вісник НТУ України "КПІ". 2021. Вип. 84. С. 45–52.
34. Абакумова А. О., Мовчанюк А. В. Апаратно-програмні методи підвищення метрологічної надійності систем екологічного моніторингу простору // Вісник НТУ України "КПІ". 2022. Вип. 88. С. 12–19.

ДОДАТОК Б

```

#include <WiFi.h>
#include <Wire.h>
#include <Adafruit_SHT31.h>
#include <ESPAsyncWebServer.h>
#include <AsyncTCP.h>
const char* ssid    = "netis_2.4G_137F55";
const char* password = "password";
#define FAN_PIN 26
#define MQ135_PIN 34
Adafruit_SHT31 sht31 = Adafruit_SHT31();
AsyncWebServer server(80);
AsyncWebsocket ws("/ws");
unsigned long lastFanStateChange = 0;
bool isFanRunning = false;
const unsigned long fanPauseDuration = 30000;
const unsigned long fanWorkDuration = 10000;
const char index_html[] PROGMEM = R"rawliteral(
<!DOCTYPE html><html><head><meta charset="utf-8"><meta name="viewport"
content="width=device-width,initial-scale=1">
<title>Клімат Панель</title>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.min.js"></script>
<style>
body{background:#1e1e2f;color:#fff;font-family:sans-serif;margin:0;padding:15px}
.grid{display:grid;grid-template-columns:repeat(auto-fit,minmax(200px,1fr));gap:15px;margin-
bottom:20px;max-width:900px;margin-left:auto;margin-right:auto}
.card{background:#27293d;padding:20px;border-radius:10px;text-align:center;box-shadow:0 4px
6px rgba(0,0,0,.3)}
.card h3{margin:0 0 10px;font-size:12px;color:#8f8f9d;text-transform:uppercase;letter-spacing:1px}
.val{font-size:28px;font-weight:700}
.t{color:#ff5d73}.h{color:#00f2fe}.g{color:#e2a0ff}

```

```

#chart{background:#27293d;padding:15px;border-radius:10px;max-width:900px;margin:0 auto}
h2{text-align:center;font-weight:300;letter-spacing:2px;margin-bottom:30px}
</style></head>

<body>

<h2>IoT МИКРОКЛИМАТ</h2>

<div class="grid">

<div class="card"><h3>Температура</h3><div class="val t"><span id="t">--</span>
°C</div></div>

<div class="card"><h3>Вологість</h3><div class="val h"><span id="h">--</span>
%</div></div>

<div class="card"><h3>Якість повітря</h3><div class="val g"><span id="g">--</span>
ADC</div></div>

</div>

<div id="chart"><canvas id="c" height="250"></canvas></div>

<script>

var ws=new WebSocket("ws://" +location.hostname+"/ws");

var ch=new Chart(document.getElementById("c").getContext("2d"),{
type:"line",data:{labels:[],datasets:[
{label:"Температура",borderColor:"#ff5d73",data:[],fill:false,yAxisID:"y1",borderWidth:2,pointRadius:0},
{label:"Вологість",borderColor:"#00f2fe",data:[],fill:false,yAxisID:"y2",borderWidth:2,pointRadius:0}
]},options:{responsive:true,maintainAspectRatio:false,scales:{yAxes:[
{id:"y1",position:"left",gridLines:{color:"#3a3a52"},ticks:{fontColor:"#8f8f9d",min:10,max:50}},
{id:"y2",position:"right",gridLines:{display:false},ticks:{fontColor:"#8f8f9d",min:0,max:100}}
]},legend:{labels:{fontColor:"#fff"}}}}
});

ws.onmessage=function(e){
var d=JSON.parse(e.data);
document.getElementById("t").innerText=d.t.toFixed(1);
document.getElementById("h").innerText=d.h.toFixed(1);
document.getElementById("g").innerText=d.g;
var tm=new Date().toLocaleTimeString();

```

```

if(ch.data.labels.length>40){ch.data.labels.shift();ch.data.datasets[0].data.shift();ch.data.datasets[1].
data.shift();}

ch.data.labels.push(tm);ch.data.datasets[0].data.push(d.t);ch.data.datasets[1].data.push(d.h);ch.updat
e();

};

</script></body></html>

)rawliteral";

void onEvent(AsyncWebSocket *server, AsyncWebsocketClient *client, AwsEventType type, void
*arg, uint8_t *data, size_t len) {

    if (type == WS_EVT_CONNECT) {

        Serial.printf("Клієнт підключився: %s\n", client->remoteIP().toString().c_str());

    }

}

void setup() {

    Serial.begin(115200);

    pinMode(FAN_PIN, OUTPUT);

    digitalWrite(FAN_PIN, LOW);

    if (!sht31.begin(0x44)) Serial.println("SHT31 помилка!");

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {

        delay(500);

        Serial.print(".");

    }

    Serial.println("\nWi-Fi підключено! IP: " + WiFi.localIP().toString());

    ws.onEvent(onEvent);

    server.addHandler(&ws);

    server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){

        AsyncWebServerResponse *response = request->beginResponse_P(200, "text/html",
index_html);

        response->addHeader("Connection", "close");

        response->addHeader("Cache-Control", "max-age=0");

        request->send(response);

```

```

});
server.begin();
}
void loop() {
  ws.cleanupClients();
  unsigned long currentMillis = millis();
  static unsigned long lastSensorRead = 0;
  if (currentMillis - lastSensorRead >= 500) {
    lastSensorRead = currentMillis;
    float t = sht31.readTemperature();
    float h = sht31.readHumidity();
    int gas = analogRead(MQ135_PIN);
    if (!isnan(t)) {
      String json = "{\"t\":" + String(t, 2) + ",\"h\":" + String(h, 2) + ",\"g\":" + String(gas) +
        "\",\"f\":" + (isFanRunning ? "true" : "false") + "}";
      ws.textAll(json);
      Serial.println(json);
    }
  }
  if (!isFanRunning && (currentMillis - lastFanStateChange >= fanPauseDuration)) {
    isFanRunning = true;
    lastFanStateChange = currentMillis;
    analogWrite(FAN_PIN, 150);
  }
  else if (isFanRunning && (currentMillis - lastFanStateChange >= fanWorkDuration)) {
    isFanRunning = false;
    lastFanStateChange = currentMillis;
    digitalWrite(FAN_PIN, LOW);
  }
}
}

```