

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

КВАЛІФІКАЦІЙНА РОБОТА

Кіберфізична система керування мікрокліматом групи теплиць

Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

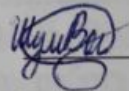
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КвРКІ 2302142.23.02.19 ПЗ

Виконав здобувач III курсу, група КІ2с-23-2



Підпис

Володимир

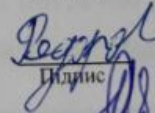
ШУМОВЕЦЬКИЙ

Ініціали, прізвище

Керівник

д.т.н., проф

Науковий ступінь, учене звання



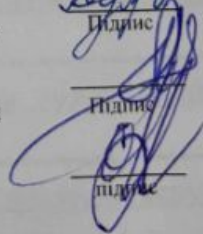
Підпис

Євген ФЕДОРОВ

Ініціали, прізвище

Нормоконтролер

Науковий ступінь, учене звання



Підпис

Сергій ЛИСЕНКО

Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІС
«11» червня 2026 р.

дата

Ольга ПАВЛОВА

Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

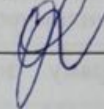
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІС



Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Шумовецькому Володимирі Олеговичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система керування мікрокліматом групи теплиць

Керівник проекту (роботи) Федоров Євген Євгенович, д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Пошук аналогів та вибір компонентів кіберфізичної системи керування мікрокліматом групи теплиць

Проектування системи обробки інформації та збору даних у кіберфізичній системі керування мікрокліматом групи теплиць

Програмно-апаратна реалізація кіберфізичної системи керування мікрокліматом групи теплиць а також тестування на симуляційних даних

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Архітектура ПЗ проекту

Підключення датчиків

Схема підключення

6. Консультанти розділів кваліфікаційної роботи

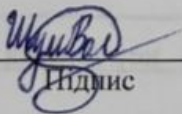
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прий

7. Дата видачі завдання « 10 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примі
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	викон
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	викон
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2026	викон
4	Робота над розділом 2 – проектування системи управління на моніторингу групою теплиць	01.04.2026	викон
5	Робота над розділом 3 – програмна реалізація та тестування системи керування групою теплиць	29.04.2026	викон
6	Оформлення пояснювальної записки згідно вимог	22.05.2026	викон
7	Попередній захист ВКР	25.05.2026	викон
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

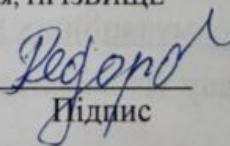
Здобувач


Підпис

Володимир ШУМОВЕЦЬКИЙ

Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи

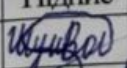
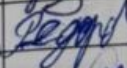
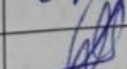
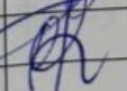

Підпис

Євген ФЕДОРОВ

Імя, ПРІЗВИЩЕ

№ р я д к а	Ф о р м а т	Позначення	Найменування	К і л · л и с т і в	№ ек з	П р и м і т к а
			<u>Текстові документи</u>			
1		КвРКІ 2302142.23.02.19 ПЗ	Пояснювальна записка	62		
			<u>Графічні матеріали</u>			
2		КвРКІ 2302142.23.02.19 Е8	Архітектура ПЗ проєкту	1		
3		КвРКІ 2302142.23.02.19 Е8	Підключення давачів	1		
4		КвРКІ 2302142.23.02.19 Е8	Схема підключення	1		

КвРКІ 2302142.23.02.19 ВП

Зм	Арк	№ докум	Підпис	Дата	Літера	Аркуш	Аркушів
Розробив		Шумовецький			У	1	1
Перевір.		Федоров			ХНУ, КІ2с-23-2		
Н. контр.		Лисенко					
Затв.		Павлова		01.06			

Відомість проєкту

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Кіберфізична система керування мікрокліматом групи теплиць».

Автор роботи: Володимир ШУМОВЕЦЬКИЙ.

Керівник роботи: Євген ФЕДОРОВ.

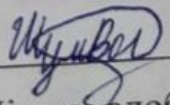
Пояснювальна записка: 62 с., 25 рис., 7 табл., 3 дод., 60 джерел.

Графічна частина: 3 креслення.

АРХІТЕКТУРА ПЗ, ПІДКЛЮЧЕННЯ ДАВАЧІВ, СХЕМА ПІДКЛЮЧЕННЯ.

Кваліфікаційна робота бакалавра присвячена розробці та дослідженню кіберфізичної системи моніторингу, аналізу та керування мікрокліматом групи теплиць із використанням IoT-технологій, MQTT-протоколу, web-інтерфейсу та інтелектуального LLM-агента. Актуальність теми зумовлена потребою в автоматизації тепличного господарства, підвищенні стабільності умов вирощування рослин, зменшенні впливу людського фактора та оптимізації використання ресурсів. Своєчасний контроль температури, вологості повітря, вологості ґрунту, рівня освітленості, концентрації CO₂ та стану виконавчих механізмів дає змогу оперативно виявляти відхилення мікроклімату й запобігати критичним ситуаціям.

Метою роботи є проектування, реалізація та тестування програмно-апаратної кіберфізичної системи для збору, передавання, зберігання, аналізу й візуалізації телеметричних даних, а також формування керуючих дій для групи теплиць у режимі, близькому до реального часу. Для досягнення поставленої мети передбачено проаналізувати сучасні підходи до побудови IoT-систем керування мікрокліматом, обґрунтувати вибір апаратних і програмних компонентів, розробити архітектуру взаємодії edge-вузлів, MQTT-брокера, серверної логіки, баз даних, web-інтерфейсу та LLM-агента.

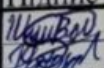
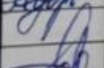

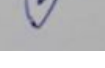

Підпис здобувача

30.05.2026

Дата

ЗМІСТ

Вступ.....	4
1 Кіберфізична система керування мікрокліматом групи теплиць та постановка задачі щодо її розроблення	6
1.1 Аналіз структурних і функціональних особливостей кіберфізичної системи керування мікрокліматом групи теплиць	6
1.2 Порівняльний аналіз переваг та недоліків існуючих рішень	9
1.3 Аналіз програмно-апаратного забезпечення обробки інформації в кіберфізичній системі керування мікрокліматом групи теплиць	16
1.4 Постановка задачі розроблення кіберфізичної системи керування мікрокліматом групи теплиць.....	18
1.5 Висновки до першого розділу.....	19
2 Проектування кіберфізичної системи керування мікрокліматом групи теплиць	21
2.1 Загальна архітектура та принцип побудови системи	21
2.2 Фізична організація групи теплиць та монтаж інфраструктури	23
2.3 Програмно-апаратні компоненти та інтеграція ESP32 у теплицю у	25
2.4 Інформаційна модель, бази даних і потоки повідомлень	30
2.5 Концептуальне розгортання контейнерної інфраструктури та протоколи зв'язку.....	32
2.6 Розробка сценарію текстової взаємодії користувача з AI-агентом.....	35
2.7 Висновки до другого розділу	39
3 Програмно-апаратна реалізація кіберфізичної системи керування мікрокліматом групи теплиць.....	41
3.1 Реалізація локального вузла ESP32 і фізичного монтажу	41
3.2 Реалізація серверних модулів, телеметрії та журналювання.....	44

КвРКІ 2302142.23.02.19 ПЗ								
Зм.	Арк.	№докум.	Підпис	Дата	Кіберфізична система керування мікрокліматом групи теплиць. Пояснювальна записка	Літера	Аркуш	Аркушів
Виконав		Володимир ШУМОВЕЦЬКИЙ				y	2	62
Перевір.		Євген ФЕДОРОВ			ХНУ КІ2с-23-2			
Н.контр.		Сергій ЛИСЕНКО						
Затвер.		Ольга ПАВЛОВА						

3.3	Опис створення схеми бази даних та таблиць системи	46
3.4	Реалізація web-інтерфейсу, dashboard, AI-чату та ручного керування	
	51	
3.5	Тестування системи на сценаріях моніторингу, ручного керування, AI-відповідей	53
3.6	Конфігурація, експлуатаційне налаштування та діагностика системи	60
3.7	Висновки до третього розділу.....	64
	Висновки	65
	Перелік джерел посилань	66
	Додаток А Копія креслення «Архітектура ПЗ проєкту».....	72
	Додаток Б Копія креслення «Підключення давачів».....	73
	Додаток В Копія креслення «Схема підключення».....	74

ВСТУП

Сучасне господарство потребує впровадження технологій, які забезпечують постійний контроль умов вирощування без необхідності безперервної присутності оператора. Ефективний розвиток рослин залежить від підтримання оптимальних параметрів середовища, зокрема температури, вологості повітря, рівня освітленості, та вологості ґрунту. Навіть короточасні відхилення цих показників можуть негативно впливати на стан культур, темпи росту, якість продукції та загальну врожайність [1,2].

Особливо актуальною ця проблема стає у випадку групи теплиць, де кожна теплиця або окрема зона може мати власні умови, типи культур, фази росту рослин і вимоги до догляду. За відсутності автоматизованої системи оператору складно одночасно контролювати всі об'єкти, порівнювати їхній стан, швидко виявляти відхилення та приймати обґрунтовані керуючі рішення. Це може призводити до несвоєчасного поливу, перегріву або переохолодження рослин, надмірного споживання ресурсів і зниження врожайності.

Актуальність теми полягає в необхідності створення кіберфізичної системи, яка дозволяє мінімізувати людське втручання у процес моніторингу та управління мікрокліматом теплиць. Така система має автоматично збирати телеметрію з датчиків, передавати її через мережеві протоколи, зберігати історичні дані, аналізувати поточний стан середовища та за потреби формувати керуючі дії для пристроїв теплиці. Це дає змогу зменшити залежність від ручного контролю та забезпечити більш стабільні умови для вирощування рослин [3,4].

Важливою перевагою запропонованого підходу є можливість віддаленого аналізу та керування групою теплиць. Користувач може отримувати доступ до актуальних показників, історії змін, попереджень і рекомендацій через web-інтерфейс незалежно від місця перебування. Інтеграція інтелектуального LLM-агента додатково розширює функціональність системи, дозволяючи користувачу взаємодіяти з нею природною мовою, наприклад запитувати загальний стан

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 4
Зм.	Арк.	№ докум.	Підпис	Дата		

теплиць, отримувати пояснення щодо відхилень або рекомендації щодо поливу, провітрювання чи зміни режимів роботи .

Таким чином, завдання дослідження полягає в розробці масштабованої, автоматизованої та інтелектуальної кіберфізичної системи, яка забезпечує моніторинг, аналіз і керування мікрокліматом групи теплиць із мінімальною участю людини та підтримкою віддаленого доступу до стану системи . Реалізація такого підходу дозволяє підвищити оперативність прийняття рішень, зменшити ризик людських помилок, оптимізувати використання ресурсів і забезпечити стабільні умови для розвитку рослин .

					КвРКІ 2302142.23.02.19 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

1 КІБЕРФІЗИЧНА СИСТЕМА КЕРУВАННЯ МІКРОКЛІМАТОМ ГРУПИ ТЕПЛИЦЬ ТА ПОСТАНОВКА ЗАДАЧІ ЩОДО ЇЇ РОЗРОБЛЕННЯ

1.1 Аналіз структурних і функціональних особливостей кіберфізичної системи керування мікрокліматом групи теплиць

Коли мова йде про автоматизацію теплиці, часто уявляють “підтримку температури” або “вмикання поливу за розкладом”. Проте сучасна теплиця - це керована біотехнічна система, де мікроклімат формується спільною роботою опалення, вентиляції, зволоження/осушення, екранів затінення, освітлення та CO₂-дозування, ці контури впливають один на одного і мають різні часові масштаби [5,6].

Історично перехід до “розумної” теплиці відбувався через появу моделей, здатних описувати динаміку тепличного клімату. У 1970-х роках у науковій літературі вже розвивалися динамічні моделі з урахуванням радіаційного балансу та природної вентиляції, що заклало підґрунтя для подальших методів оптимального й предиктивного керування.

Важливим каталізатором автоматизації стало здешевлення обчислювальної техніки. Поява мікропроцесора як “програмованого логічного мікрочипа” різко розширила можливості вбудованих систем керування.

Паралельно формувалися промислові практики керування технологічними процесами (контролери, протоколи обміну, стандарти), що згодом були адаптовані для тепличних “процес-комп’ютерів” і диспетчеризації.

Перші кроки комерційної “тепличної автоматизації” можна простежити на прикладі того, що у 1977 році був представлений перший клімат комп’ютер для садівництва в портфелі компанії Priva - як окремий історичний етап розвитку керування мікрокліматом (рисунок 1.1).

Сьогодні, зростання масштабів виробництва й енергетичного тиску робить інтелектуальне керування мікрокліматом не “опцією”, а фактором конкурентоздатності: огляд енергоспоживання теплиць у ЄС показує, що

					КвРКІ 2302142.23.02.19 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

найбільша частка енергетичних витрат припадає на опалення (у наведених оцінках - близько 74% енерговкладень, тоді як електроенергія становить меншу частину).



Рисунок 1.1 – Одна з перших моделей Priva [11]

Це означає, що навіть невеликі помилки в стратегіях вентиляції/осушення або “перегріву-переохолодження” можуть перетворюватися на значні втрати енергії та грошей у масштабі групи теплиць.

Окремою практичною проблемою є вологість і конденсація. ВПД (vapor pressure deficit, VPD) є більш фізично обґрунтованим показником “рушійної сили” транспірації порівняно з відносною вологістю, оскільки він безпосередньо відображає дефіцит водяної пари щодо насичення і краще пов’язаний із водним обміном рослин [7,8].

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

Водночас контроль вологості в теплиці часто вважається одним із найскладніших параметрів керування через його взаємозв'язок із температурою [9], випаровуванням та енергетичними втратами під час вентиляції (рисунок 1.2).

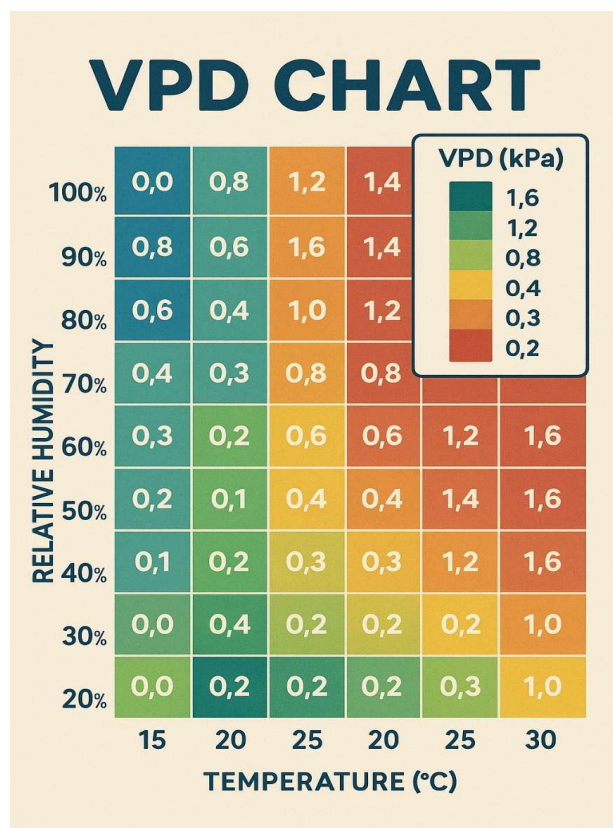


Рисунок 1.2 – Залежність вологості від температури повітря [6]

Якщо говорити про цілу групу теплиць, до уваги доводиться брати не лише технічні нюанси, а й організаційні питання. У кожній теплиці можуть рости різні культури, діяти власні режими (ніч чи день), працювати своє обладнання. Все це вимагає централізованого контролю, а ще - продуманої політики безпеки для віддаленого доступу.

Ще одна важлива річ - просторову неоднорідність не варто ігнорувати. Показники з кількох “еталонних” давачів не завжди відображають реальну картину в усій теплиці. У рекомендаціях із повітрообміну наголошується: нерівномірний розподіл параметрів середовища впливає на ріст рослин і

поширення хвороб, а нерепрезентативні заміри можуть призвести до зайвих витрат енергії, таке розміщення зображено на рисунку 1.3.

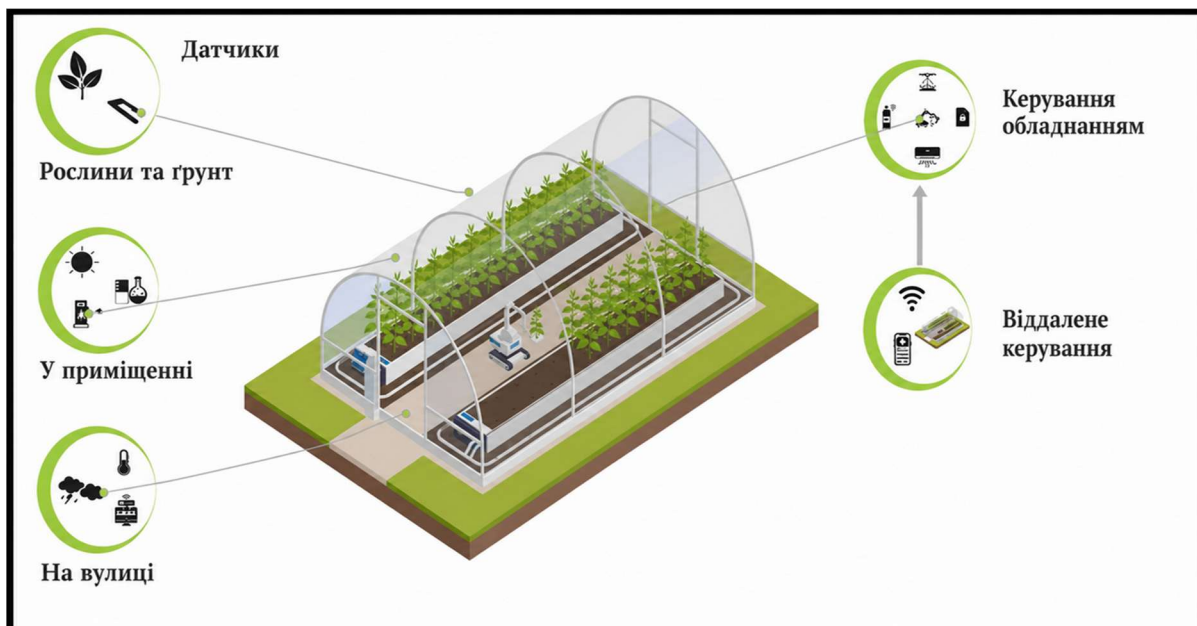


Рисунок 1.3 – Концептуальне розміщення сенсорних точок у групі теплиць (з урахуванням зон: крона, коренева зона, технічні приміщення) [10]

У підсумку, предметна область формує типові завдання: збір телеметрії з багатьох зон, енергетична оптимізація, журналювання подій, віддалений контроль, а також підтримка прийняття рішень для оператора/агронома.

1.2 Порівняльний аналіз переваг та недоліків існуючих рішень

Система, що об'єднує давачі, механізми виконання, обчислювальні вузли та програмні сервіси для моніторингу і керування фізичним процесом (мікрокліматом), відповідає суті кіберфізичної системи: інтеграції цифрових, фізичних і людських компонентів для функціонування через поєднання “фізики й логіки”.

У контексті теплиці практично виділяють функціональні підсистеми [10]: клімат (температура/вологість/вентиляція), CO₂ та фотосинтез-орієнтовані стратегії (світло-CO₂-екрани), зрошення, енергоменеджмент (буфери/джерела

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 9
Зм.	Арк.	№ докум.	Підпис	Дата		

тепла/тарифи), а також безпека й сповіщення про небезпеку (пожежа, витіки, доступ, кіберзагрози віддаленого керування) (рисунок 1.4).



Рисунок 1.4 – Діаграма взаємодії “сенсори → події/моделі → керуючі дії” у тепличному моніторингу (WSN + подієве керування) [7]

З точки зору типізації рішень для групи теплиць, у практиці часто зустрічаються такі дихотомії:

1. Централізовані проти децентралізованих контурів керування (зональні контролери vs “єдиний мозок”).
2. Закриті екосистеми проти відкритої інтеграції.
3. Дротові промислові контури (типу Modbus) проти бездротових сенсорних мереж (Zigbee/Z-Wave/LoRaWAN).

Комерційні рішення (клас клімат-комп’ютер + інтеграційні модулі + хмарний інтерфейс) показують тенденцію до єдиного інтегратора, який зв’язує клімат, світло, воду та енергію. Наприклад, матеріали для Priva описують центральне керування кліматом/світлом/іригацією/енергією та предиктивні можливості платформи Connex.

Для групи теплиць така властивість має особливе значення, оскільки керування кількома об'єктами потребує не тільки збору телеметрії, а й уніфікованої адресації теплиць, зон, сенсорів і модулів керування. Якщо система підтримує централізований доступ до різних локацій, то оператор може порівнювати умови вирощування між теплицями, швидше виявляти проблемні зони та оцінювати ефективність виконаних дій. Саме тому в сучасних комерційних платформах важливу роль відіграють не лише локальні контури керування, а й засоби диспетчеризації, аналітики, журналювання та віддаленого доступу. Для керування групою теплиць ключовою рисою стає multi-site: в офіційному описі Priva Operator зазначено можливість керувати кількома локаціями з одного інтерфейсу та здійснювати віддалене керування в реальному часі.

Поряд із цим, рішення від Hoogendoorn Growth Management [11,12] позиціонують віддалене керування через PIVO Remote як спосіб контролю теплиці “з будь-якої точки світу”, а також підкреслюють орієнтацію на баланс енергії та води й попереджувальне (pre-control) реагування на зовнішні умови.

Компанія Ridder у описі Hortimax Pro робить акцент на “always connected”, мобільний застосунок, модульність і відкриту архітектуру з інтеграціями з іншими технологіями теплиць.

Для Argus Controls [13,14] у наявних публічних матеріалах характерний фокус на інструментах зрозуміти і використати ВПД як керовану величину для клімат-стратегій (що важливо саме для практики боротьби з конденсацією/хворобами та стабілізації режимів). Реалізацію зображено на рисунку 1.5.

Переваги комерційних платформ полягають насамперед у промисловій надійності, наявності сервісної підтримки, інтеграції багатьох технологічних процесів в одну модель керування, забезпеченні віддаленого доступу та візуалізації, а також у можливості масштабування рішення від окремої теплиці до великого тепличного комплексу.

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

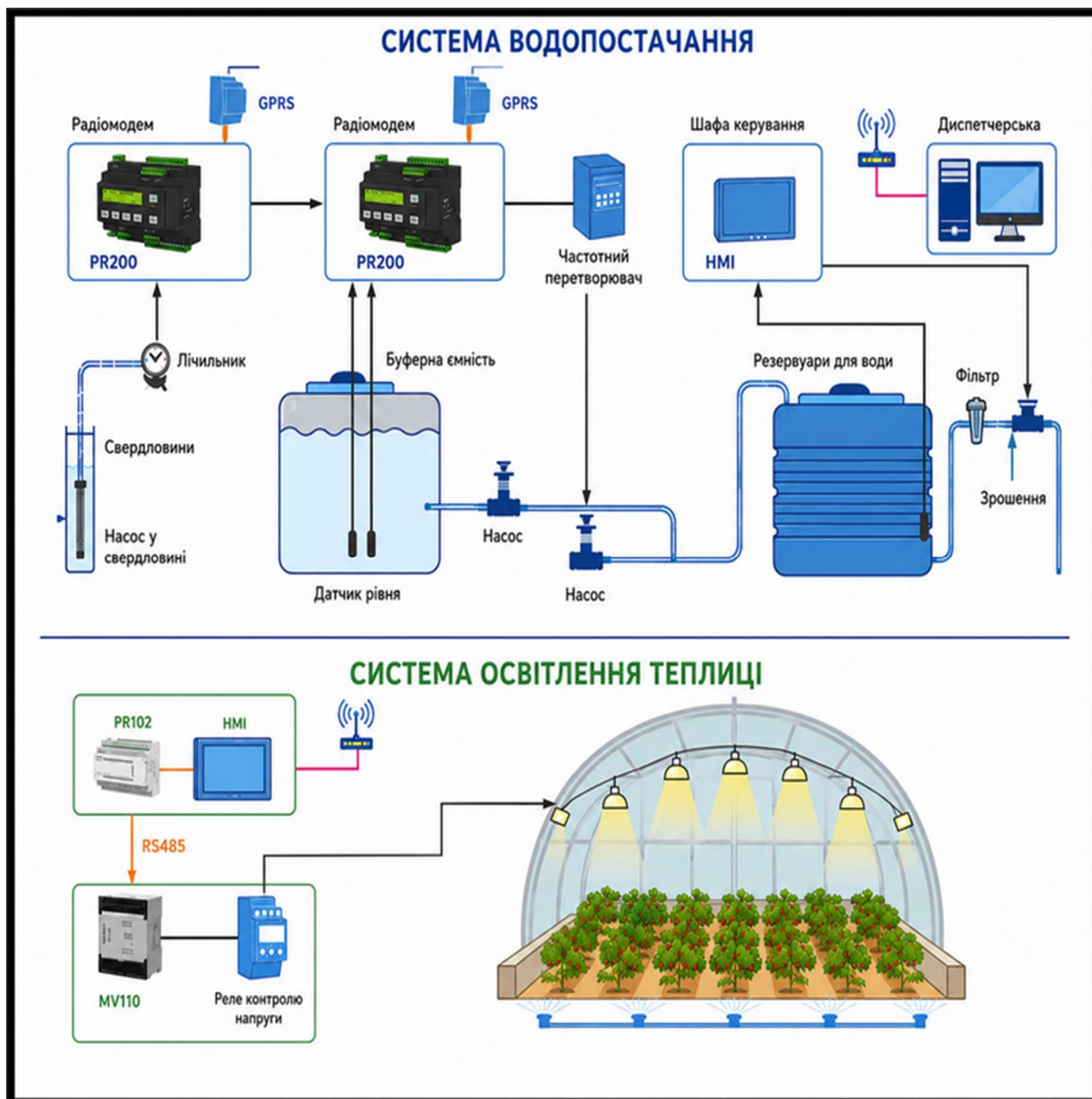


Рисунок 1.5 – Наглядний приклад підсистем керування мікрокліматом теплиці (клімат, світло/CO₂, вода, енергія, тривоги/безпека) [13]

Недоліки, які часто з’являються у наукових оглядах, пов’язані з приватністю/закритістю протоколів та обмеженням гнучкості інтеграцій: у систематичному огляді з IoT-перспективи прямо згадується, що комерційні системи керування середовищем (у тому числі згадуються Priva та інші) можуть бути “private and inflexible”.

Зм.	Арк.	№ докум.	Підпис Дата

Науково-дослідні та відкриті підходи [15,16] часто стартують з WSN/ІоТ: наприклад, класична робота в Sensors показує поєднання бездротової сенсорної мережі та подієвого (event-based) керування як спосіб зменшити знос актуаторів і при цьому зберігати прийнятну якість регулювання.

Інший приклад у напрямі масштабування - система моніторингу/керування на Zigbee, де прямо описано збір даних (температура, вологість, вміст води в ґрунті, CO₂), два режими керування (локальний ручний та віддалений бездротовий) і навіть застосування PID для температурного контуру.

Для протяжних об'єктів і "рознесених" теплиць згадується LoRaWAN. У публікації Sensors наведено end-to-end розгортання мережі датчиків у теплиці (Бельгія) з аналізом енергоспоживання й візуалізацією даних.

Однак дослідницькі прототипи майже завжди потребують доведення до промислової експлуатації: валідації точності датчиків, захисту каналу керування, процедури оновлень, спостережуваності та кібербезпеки, що є критичним для систем, які реально керують фізичним процесом [17,18].

Поряд із типовими архітектурними підходами до побудови систем контролю мікроклімату, на ринку сьогодні представлено низку комерційних платформ, які реалізують інтеграцію підсистем (клімат, вода, енергія, CO₂, освітлення) в єдину модель керування. Це дозволяє управляти всіма важливими параметрами агротехнологічного середовища через один інтерфейс і масштабувати рішення від окремої теплиці до комплексів із багатьма локаціями.

Серед них Priva Connex / Priva Operator - один із провідних прикладів інтегрованої автоматизації тепличного середовища, що включає кліматичні комп'ютери, сенсорні мережі, систему зрошення та енергоменеджмент у єдину платформу. Система забезпечує реал-тайм моніторинг і автоматичне регулювання температури, вологості, CO₂, освітлення і води, а також передбачає віддалене управління та підтримку multi-site сценаріїв, приклад вигляду зображено на рисунку 1.6.

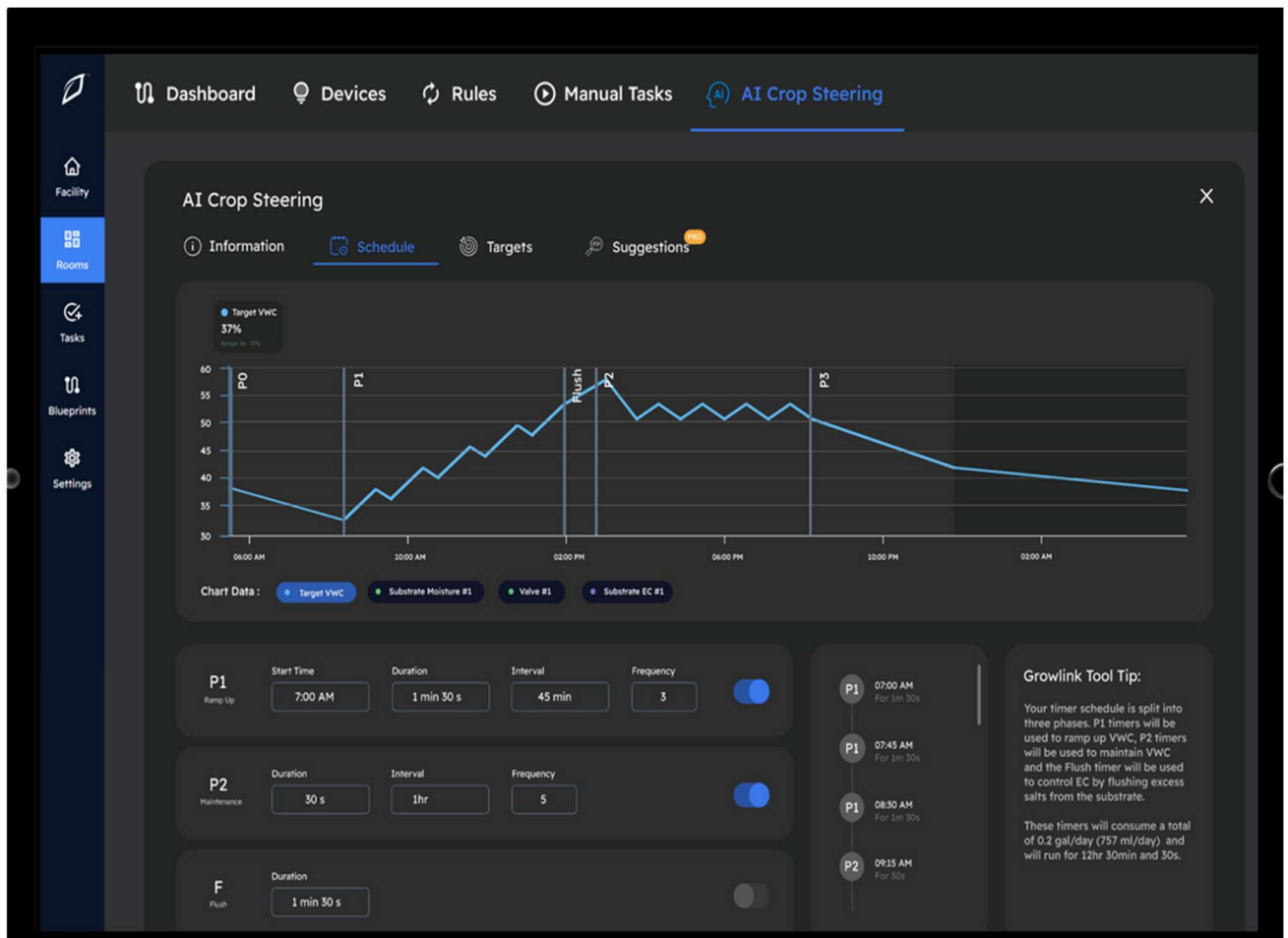


Рисунок 1.7 – Вигляд системи Growlink [19]

Ці рішення ілюструють стандартизовані комерційні підходи, де всі функції від сенсорних даних до керуючих дій - об'єднані у кіберфізичну систему з єдиним центром логіки, дозволяючи оптимізувати ресурси, підвищити врожайність і знизити операційні витрати.

У таких системах часто виділяють такі складові:

- 1) інтеграція підсистем мікроклімату (синхронізація кліматичного контролю, освітлення, CO₂ та зрошення) через центральний алгоритм, що визначає оптимальні керуючі стратегії;
- 2) віддалений доступ і мультисайт-моніторинг, що дозволяє оперувати кількома тепличними комплексами з одного інтерфейсу;

3) аналітика й прогнозування, що дозволяє застосовувати предиктивну логіку на основі трендів даних і оптимізувати енергоспоживання та водний баланс;

4) підтримка інтеграційного обладнання, де частина платформ надає відкриті API або підтримує стандартизовані протоколи для сторонніх сенсорів/модулів контролю.

Ці комерційні платформи демонструють тенденцію до централізованого, інтегрованого управління, що включає моделі оптимізації, дані в реальному часі та інструменти візуалізації. Незважаючи на переваги у промисловій надійності та масштабованості, у науковій літературі їх критикують за часткову закритість протоколів, високу вартість та обмеження гнучкості при інтеграції нестандартних IoT-компонентів - особливо в контексті кастомізованих або гібридних архітектур.

1.3 Аналіз програмно-апаратного забезпечення обробки інформації в кіберфізичній системі керування мікрокліматом групи теплиць

Побудова системи керування мікрокліматом групи теплиць потребує врахування не тільки складу технічних засобів, а й фізичної природи процесів, що відбуваються всередині теплиці. У реальних умовах температура, вологість, освітлення і стан ґрунту змінюються нерівномірно, а реакція рослин на відхилення може бути відкладеною в часі. Тому система повинна забезпечувати не лише миттєве зчитування показників, а й накопичення історії, аналіз тенденцій і зіставлення параметрів із профілями культур.

Окрему роль відіграє зональний підхід. Якщо теплиця містить кілька грядок або стелажів, то середнє значення по всьому об'єкту не дає достатньо точної картини. Зона біля дверей може швидше охолоджуватися, зона біля прозорої стінки - перегріватися, а віддалена від насоса грядка - отримувати

менше вологи. У групі теплиць ці відмінності стають ще більш помітними, оскільки кожен об'єкт має власний набір культур і режим експлуатації.

Для практичного впровадження важливо забезпечити зв'язок між фізичною точкою вимірювання і цифровою моделлю системи. Кожне значення телеметрії повинно мати прив'язку до групи, теплиці, зони, сенсора, типу параметра та часу. Якщо хоча б один із цих елементів відсутній, система втрачає можливість коректно пояснити причину відхилення або сформувати адресну команду.

Ще одним важливим аспектом є розмежування ролей між автоматикою, оператором та інтелектуальним модулем.

Автоматичний контур може виконувати прості дії за правилами, оператор приймає рішення у складних або ризикових ситуаціях, а AI-рівень допомагає інтерпретувати дані та формувати рекомендації. Така організація знижує ризик небажаних фізичних впливів на рослини і водночас робить систему зручною для повсякденного використання.

Для проєктування важливо розглядати дані не лише як окремі числові значення, а як контекстні спостереження [20,21]. Значення вологості ґрунту має різний зміст залежно від культури, фази росту, типу субстрату й часу останнього поливу.

Аналогічно температура повітря оцінюється не сама по собі, а у зв'язку з вологістю, освітленістю і режимом вентиляції.

У групі теплиць зростає роль порівняльного аналізу. Якщо в одній теплиці показники стабільні, а в іншій виникають різкі коливання, система повинна допомогти оператору локалізувати причину: проблеми з давачем, недостатню вентиляцію, помилку поливу або відмінність профілю вирощування. Це формує вимогу до історичності, журналювання й пояснюваності. Основні аспекти, які враховуються під час побудови системи і важливі для повного розуміння, подано в таблиці 1.1 .

Таблиця 1.1 - Теоретичні аспекти, які враховуються під час побудови системи

Аспект	Зміст	Практичний наслідок
Фізична взаємодія параметрів	Температура, вологість, світло й полив взаємно впливають одне на одного	Потрібен комплексний аналіз, а не контроль одного показника
Просторова неоднорідність	Умови в різних зонах теплиці відрізняються	Потрібне зональне розміщення сенсорів
Якість даних	Некоректні вимірювання призводять до хибних рішень	Потрібні правила перевірки і контекстуалізації
Безпечність дій	Команди впливають на фізичне середовище	Потрібна валідація і журналювання
Пояснюваність	Оператор повинен розуміти причини рекомендації	Потрібен зрозумілий інтерфейс і AI-пояснення

1.4 Постановка задачі розроблення кіберфізичної системи керування мікрокліматом групи теплиць

Об'єктом дослідження є процес моніторингу, аналізу та керування параметрами мікроклімату в групі теплиць як кіберфізичній системі, у якій цифрові засоби обробки даних взаємодіють із фізичними процесами вирощування рослин.

Предметом дослідження є методи та програмно-апаратні засоби побудови кіберфізичної системи, що забезпечує збір телеметрії, обробку даних, взаємодію

з модулями керування, відображення стану теплиць і підтримку прийняття рішень оператором.

Метою роботи є розроблення системи, яка дозволяє контролювати параметри середовища в кількох теплицях, аналізувати стан окремих зон, формувати рекомендації природньою мовою та виконувати керуючі дії лише після перевірки правил безпеки. Для досягнення цієї мети необхідно проаналізувати предметну область, визначити склад параметрів мікроклімату, обґрунтувати фізичне розміщення сенсорів, сформувавши структуру групи теплиць, описати апаратні підсистеми, спроектувати канали передавання даних, визначити структуру баз даних, реалізувати інтерфейс користувача, забезпечити сценарії ручного, автоматичного і напівавтоматичного керування, а також перевірити працездатність системи на типових ситуаціях.

Окремо встановлено, що інтелектуальний модуль не повинен мати прямого неконтрольованого доступу до фізичних механізмів. Його призначення полягає в аналізі стану, поясненні причин відхилень і формуванні пропозицій, тоді як остаточне виконання команди має проходити через серверну логіку, модуль перевірки обмежень і підтвердження користувача. Таке обмеження є принциповим, оскільки система керує насосами, вентиляторами, обігрівачами та освітленням, тобто впливає на реальний фізичний процес.

Перелік задач до виконання формулюються так, щоб охопити повний життєвий цикл рішення: від аналізу предметної області до перевірки інтерфейсу, MQTT-взаємодії та безпечного виконання команд.

1.5 Висновки до першого розділу

У першому розділі розглянуто особливості побудови кіберфізичних систем, призначених для контролю та регулювання мікроклімату в групі теплиць. Визначено, що така система повинна враховувати не лише показники окремої теплиці, а й стан декількох об'єктів, зон вирощування, типів культур та

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

обладнання, яке використовується для підтримання оптимальних умов середовища. Показано, що керування мікрокліматом є комплексним процесом, оскільки температура, вологість повітря, вологість ґрунту, освітленість і концентрація CO₂ взаємно впливають один на одного та потребують узгодженої обробки.

Проведений огляд наявних рішень показав, що сучасні комерційні платформи для автоматизації теплиць забезпечують широкий набір функцій: централізований моніторинг, керування кліматом, зрошенням, освітленням, CO₂ та енергетичними режимами. Водночас такі рішення часто мають високу вартість, орієнтовані на закриті екосистеми та не завжди забезпечують достатню гнучкість для інтеграції нестандартних датчиків, відкритих протоколів або власних програмних модулів. Це обґрунтовує потребу в розробленні більш відкритої та адаптованої архітектури.

Додатково уточнено теоретичні та практичні передумови побудови системи: взаємозв'язок параметрів мікроклімату, просторову неоднорідність середовища, вимоги до якості даних, безпечність фізичних команд, пояснюваність для оператора та масштабування від однієї теплиці до групи об'єктів. У постановці задачі визначено об'єкт, предмет, мету та основні завдання роботи. Отримані результати стали основою для подальшого проектування архітектури, апаратної інтеграції та програмної реалізації системи.

Окремо встановлено, що для групи теплиць важливо забезпечити не тільки автоматизований збір даних, а й чітку ідентифікацію та належність в системі кожного сенсора, зони та модуля керування. Це дає змогу уникнути неоднозначності під час аналізу показників і формування керуючих дій. Також обґрунтовано необхідність поєднання локального рівня збору телеметрії з централізованим рівнем аналізу, журналювання та взаємодії з користувачем.

Отже, результати першого розділу підтверджують доцільність створення масштабованої кіберфізичної системи для моніторингу, аналізу та керування мікрокліматом групи теплиць.

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ КЕРУВАННЯ МІКРОКЛІМАТОМ ГРУПИ ТЕПЛИЦЬ

2.1 Загальна архітектура та принцип побудови системи

Проєктування системи починається з формування архітектури, що поєднує фізичні теплиці, локальні вузли, мережевий обмін, серверну логіку, сховища даних, інтерфейс користувача та інтелектуальний контур. У такій архітектурі важливо не змішувати рівні відповідальності: сенсорний рівень вимірює параметри, локальний контролер формує телеметрію та приймає команди, транспортний рівень доставляє повідомлення, серверний рівень обробляє дані, а користувацький рівень забезпечує взаємодію з оператором.

Система проєктується як рішення для групи теплиць, тому центральною вимогою є масштабованість. Додавання нової теплиці не повинно змінювати логіку вже наявних модулів. Для цього застосовується ієрархічна модель, у якій кожен об'єкт має власний ідентифікатор: група, теплиця, зона, сенсор і модуль керування. Така модель забезпечує адресність даних і команд.

На рівні користувацької взаємодії передбачено два основні канали: панель моніторингу і AI чат. Панель моніторингу використовується для перегляду поточних значень, графіків, команд і станів модулів керування використовується для запитів природною мовою, наприклад для отримання відповіді про загальний стан теплиць або причину відхилення.

Кожен рівень архітектури системи має власну зону відповідальності та виконує окрему функцію в загальному процесі моніторингу й керування мікрокліматом. Локальний вузол відповідає за збір телеметрії та прийом підтверджених команд, але не виконує довготривалий аналіз історичних даних. Серверний рівень забезпечує обробку повідомлень, збереження даних, перевірку правил і журналювання подій. Інтелектуальний модуль використовується для аналізу стану, пояснення відхилень і формування рекомендацій, однак не

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

повинен напряму активувати насос, обігрівач або інші модулі керування. Загальну архітектурну структуру системи зображено на рисунку 2.1.

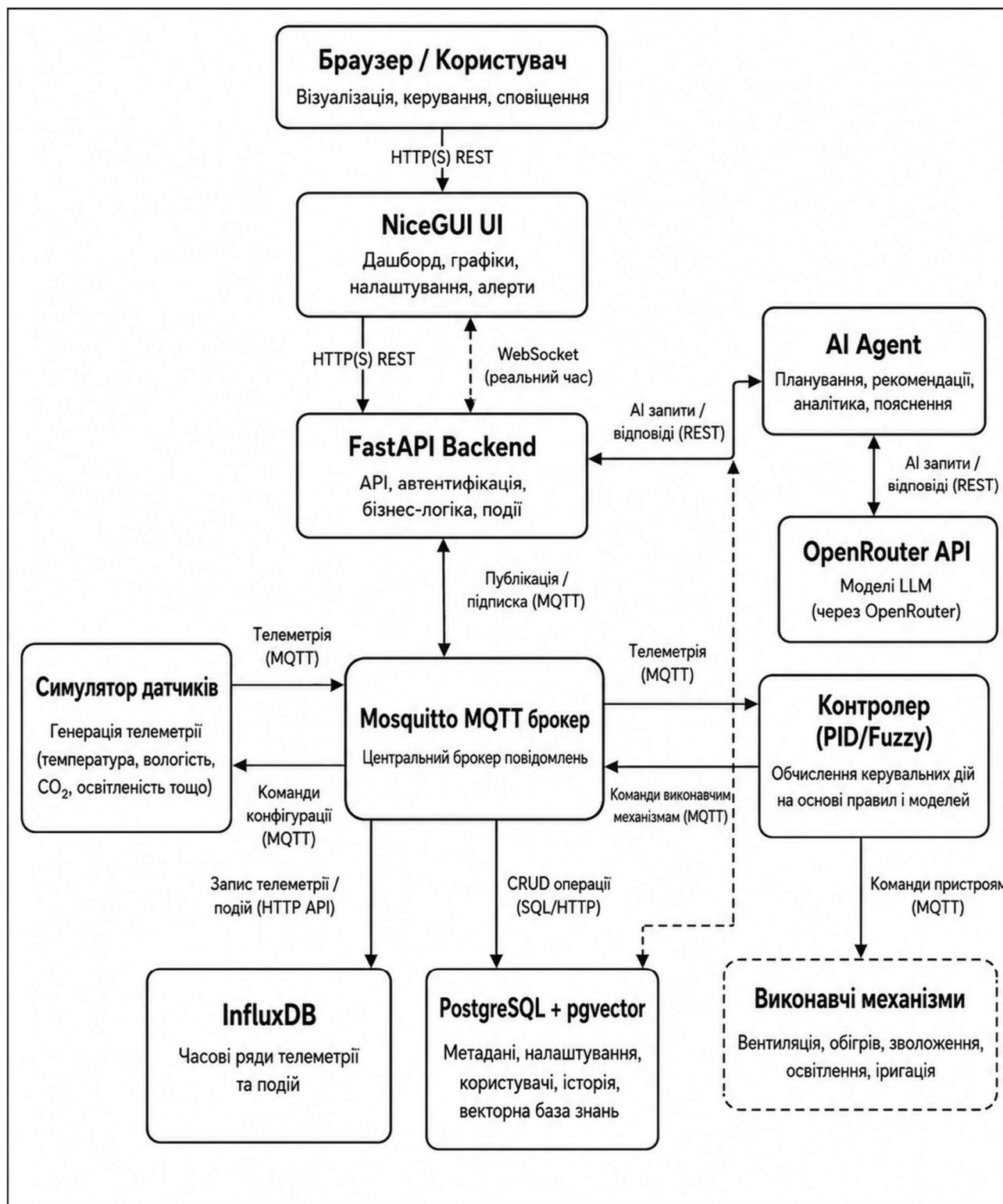


Рисунок. 2.1 – Загальна архітектура кіберфізичної системи керування мікрокліматом групи теплиць

Зм.	Арк.	№ докум.	Підпис	Дата

Фізична організація групи теплиць визначає не тільки місця встановлення обладнання, а й логіку подальшої експлуатації системи. В межах роботи кожна теплиця розглядається як окремий фізичний об'єкт, який має власний локальний вузол керування, набір сенсорів, модулі керування й службові точки підключення. Водночас усі теплиці об'єднуються в єдину цифрову систему, тому для них застосовуються однакові правила ідентифікації, журналювання та передавання повідомлень. Узагальнення ролей основних рівнів системи подано в таблиці 2.1.

Таблиця 2.1 – Рівні архітектури системи та їхні функції

Рівень	Основна функція	Результат роботи
Фізичний	Вимірювання параметрів і вплив на середовище	Показники сенсорів та реакція модулів керування
Локальний	Первинна обробка і прийом команд	Структурована телеметрія та виконання дій
Транспортний	Передавання повідомлень	Доставка телеметрії, станів і команд
Серверний	Валідація, логіка, API	Оновлення станів, команди, журнали
Рівень даних	Зберігання історії та довідників	Графіки, звіти, профілі, RAG-контекст
Користувацький та AI рівень	Візуалізація й пояснення	Моніторинг, чат, рекомендації

2.2 Фізична організація групи теплиць та монтаж інфраструктури

Під час монтажу доцільно виходити з принципу зональності. Окрема зона відповідає грядці, стелажу або частині теплиці, у якій вирощується певна культура або партія рослин. Саме на рівні зони найбільш коректно оцінювати

вологість ґрунту, поточний стан рослин і необхідність поливу. Якщо в теплиці є спільні пристрої, наприклад один вентилятор або загальний обігрівач, вони описуються як модулі керування теплиці, а не як пристрої конкретної грядки.

Розміщення сенсорів повинно забезпечувати репрезентативність вимірювань. Давач температури та вологості повітря недоцільно розміщувати біля дверей, вентилятора, обігрівача або безпосередньо під лампою, оскільки такі точки дають локально спотворені значення. Сенсори вологості ґрунту слід встановлювати в кореневій зоні, а не на поверхні субстрату. Давачі освітленості мають розташовуватися так, щоб не перебувати в постійній тіні від конструкцій або листя.

Монтаж модулів керування також має враховувати фізичну логіку теплиці. Насос і клапани пов'язуються з поливною магістраллю, вентилятор розташовується так, щоб створювати циркуляцію без прямого обдування давача, обігрівач встановлюється з урахуванням пожежної безпеки, а LED-освітлення монтується над зонами вирощування з рівномірним покриттям.

Така організація дозволяє зменшити локальні відхилення і забезпечити прогнозовану реакцію середовища на керуючі дії.

Окремо планується монтажний бокс. У ньому розміщується контролер, релейні або транзисторні модулі, клемні колодки, елементи захисту і стабілізоване живлення. Бокс не повинен встановлюватися в місцях прямого попадання води або активного конденсатоутворення.

Для обслуговування важливо залишити доступ до клем, маркування проводів і можливість швидко відключити живлення окремого каналу.

Сигнальні лінії й силові кабелі мають прокладатися окремо. Сенсорні дроти бажано вести в окремих кабель-каналах або гофрованих рукавах, а силові лінії до насосів, вентиляторів і ламп - через окремі захищені траси. Це зменшує електромагнітні наведення, спрощує сервісні роботи й підвищує загальну безпечність експлуатації.

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

Основні елементи фізичної інфраструктури та їхнє призначення подано в таблиці 2.2.

Таблиця 2.2 – Фізичне розміщення основних елементів у теплиці

Елемент	Рекомендоване місце	Практичне призначення
Давач температури і вологості	На рівні крони рослин	Вимірювання повітряного середовища
Давач вологості ґрунту	У кореневій зоні	Визначення потреби в поливі
Давач освітленості	Над грядками без постійної тіні	Оцінювання світлового режиму
Насос і клапан	Біля бака або магістралі поливу	Подача води до конкретної зони
Вентилятор	У торцевій або верхній частині	Організація повітрообміну
Контрольний бокс	У сухому технічному місці	Захист електроніки та клем

2.3 Програмно-апаратні компоненти та інтеграція ESP32 в теплицю

Програмно-апаратна частина системи є переходом від загальної архітектури до конкретної структури майбутньої реалізації.

Апаратний рівень охоплює сенсори параметрів середовища, локальний edge-вузол, релейні або транзисторні канали, модулі керування поливом, вентиляцією, обігрівом і освітленням, а також блоки живлення та елементи захисту. Сенсори формують первинні дані про температуру повітря, вологість повітря, вологість ґрунту, освітленість і концентрацію CO₂. Ці показники не повинні розглядатися ізольовано, оскільки кожен із них впливає на інші параметри мікроклімату. Наприклад, полив може змінювати вологість ґрунту й

опосередковано впливати на вологість повітря, а ввімкнення LED-освітлення може підвищувати локальну температуру в зоні рослин [22,23].

Центральним апаратним елементом локального рівня є ESP32. У межах системи він не виконує роль автономного кліматичного комп'ютера, а працює як адресний edge-вузол, що збирає дані, формує структуровані повідомлення і приймає лише підтвержені команди. Такий підхід важливий для групи теплиць, оскільки один і той самий тип вузла може використовуватися в різних теплицях, а його належність до конкретної групи, об'єкта та зони задається через конфігурацію. Завдяки цьому масштабування не потребує зміни загальної логіки системи.

ESP32 підключається до сенсорів через цифрові або аналогові входи, залежно від типу конкретного модуля. Датчик температури та вологості повітря передає цифрові значення, тоді як сенсори вологості ґрунту, освітленості або емулятори цих параметрів можуть використовувати аналогові входи. Під час практичного монтажу важливо враховувати електричні обмеження плати: GPIO не призначені для безпосереднього живлення насосів, вентиляторів або нагрівачів. Тому всі модулі керування підключаються через проміжні комутаційні елементи.

Керування фізичними процесами реалізується через модулі, які впливають на середовище теплиці. До них належать насос і клапани поливу, вентилятор, обігрівач, LED-освітлення та, за потреби, додаткові пристрої зволоження або провітрювання. Їхня робота має бути обмежена правилами безпеки, часовими лімітами й адресною прив'язкою до конкретної теплиці або зони.

Програмний рівень локального вузла складається з конфігурації пристрою, модуля зчитування сенсорів, формування телеметрії, MQTT-клієнта і обробника команд. У конфігурації задаються ідентифікатори групи, теплиці, зони та пристрою, а також повідомлення для передавання телеметрії й отримання команд. Завдяки цьому прошивка може залишатися універсальною, а конкретний вузол набуває своєї ролі через параметри налаштування.

Серверний рівень використовує дані, отримані від ESP32, для оновлення стану зон, запису історії, побудови графіків, перевірки правил і формування пропозицій. Важливо, що локальний вузол не виконує довготривалий аналіз і не приймає самостійних складних рішень. Його завданням є надійний збір даних і виконання дозволених команд, логування, а також взаємодія з AI-агентом.

Компонентна діаграма використовується для узагальнення взаємозв'язків між апаратними й програмними частинами. Вона показує, що web-інтерфейс працює через серверну логіку, сервер взаємодіє з базами даних і MQTT-брокером, а AI-підсистема використовує дозволені інструменти для аналізу стану. Сенсорний вузол і модулі керування взаємодіють із системою через брокер повідомлень, а не напряму з базами даних або AI-рівнем. Логічну компонентну структуру розроблюваної системи подано на рисунку 2.2.

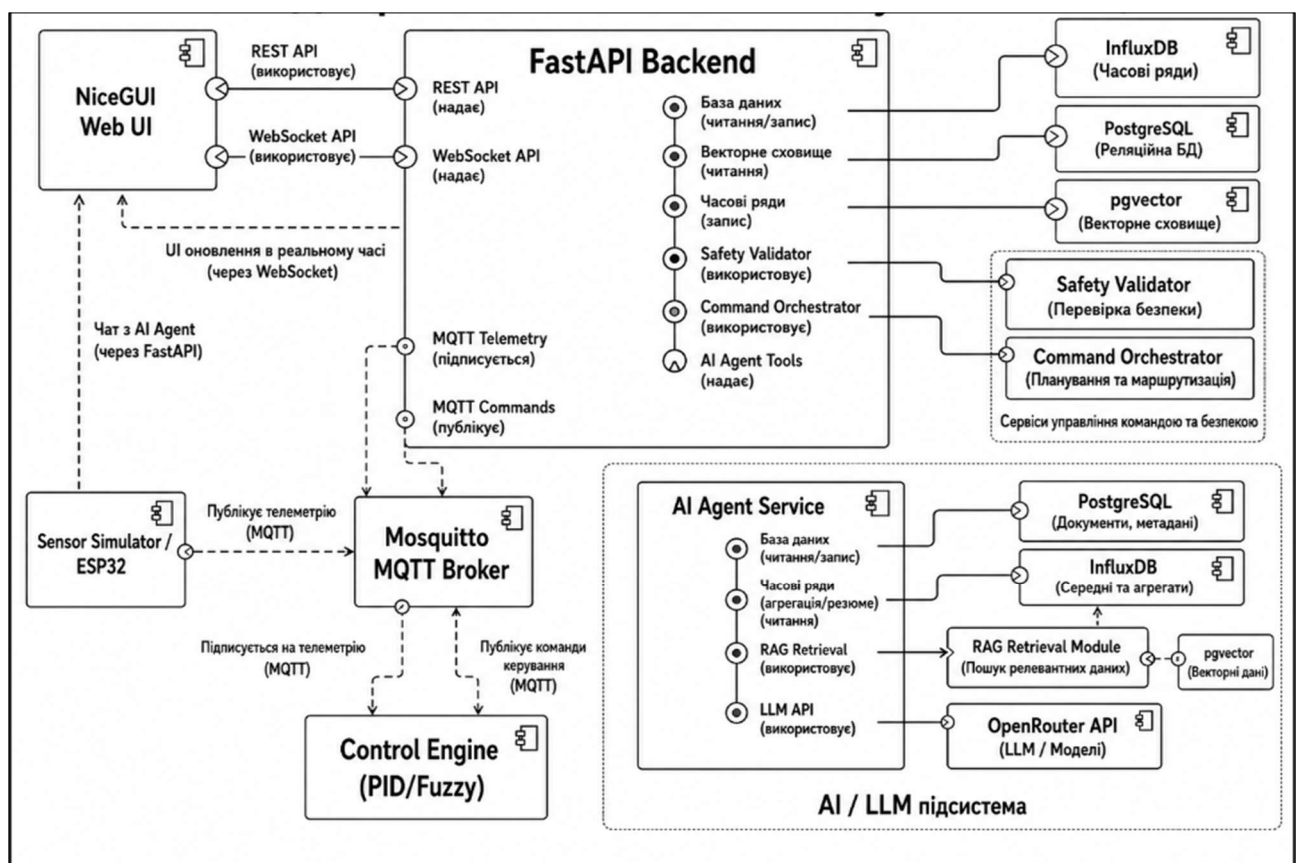


Рисунок 2.2 – Діаграма компонентів кіберфізичної системи керування мікрокліматом групи теплиць

Для стислого узагальнення складу системи достатньо виділити не кожен окрему лінію підключення, а функціональні групи компонентів. Такий підхід зменшує перевантаження проєктного розділу таблицями й залишає деталі пінів для розділу реалізації, де вони безпосередньо пов'язані зі схемою підключення ESP32. Основні програмно-апаратні групи компонентів наведено в таблиці 2.3.

Таблиця 2.3 – Програмно-апаратні групи компонентів системи

Група компонентів	Призначення
Локальний edge-вузол	Зчитування сенсорів, формування телеметрії, прийом підтверджених команд
Сенсорний рівень	Вимірювання температури, вологості, CO ₂ , освітленості та вологості ґрунту
Модулі керування	Фізичний вплив на полив, вентиляцію, обігрів і освітлення через безпечні канали
MQTT-брокер	Обмін телеметрією, станами й командами між edge-вузлами та сервером
Серверна логіка	Валідація повідомлень, правила, API, журналювання і маршрутизація команд
Сховища даних	Зберігання часових рядів, структурованих записів, журналів і контексту для RAG
Web-інтерфейс та AI-рівень	Моніторинг, пояснення стану, рекомендації та контрольовані пропозиції дій

У подальшому проектуванні важливо не змішувати поняття компонентів і конкретних електричних ліній.

Практична інтеграція ESP32 у теплицю передбачає розміщення плати в монтажному боксі, окреме підведення сенсорних ліній, комутацію модулів керування через реле та організацію стабілізованого живлення. Бокс повинен бути доступним для сервісу, але захищеним від води, пилу та конденсату. У межах однієї теплиці він виконує роль локального технічного центру, де сходяться сигнальні лінії від сенсорів і вихідні канали до модулів керування.

При плануванні потрібно відобразити не точну схема, а інженерну логіку розміщення елементів, яка допомагає зрозуміти роль локального вузла в реальному середовищі, таке підключення зображено на рис. 2.3.

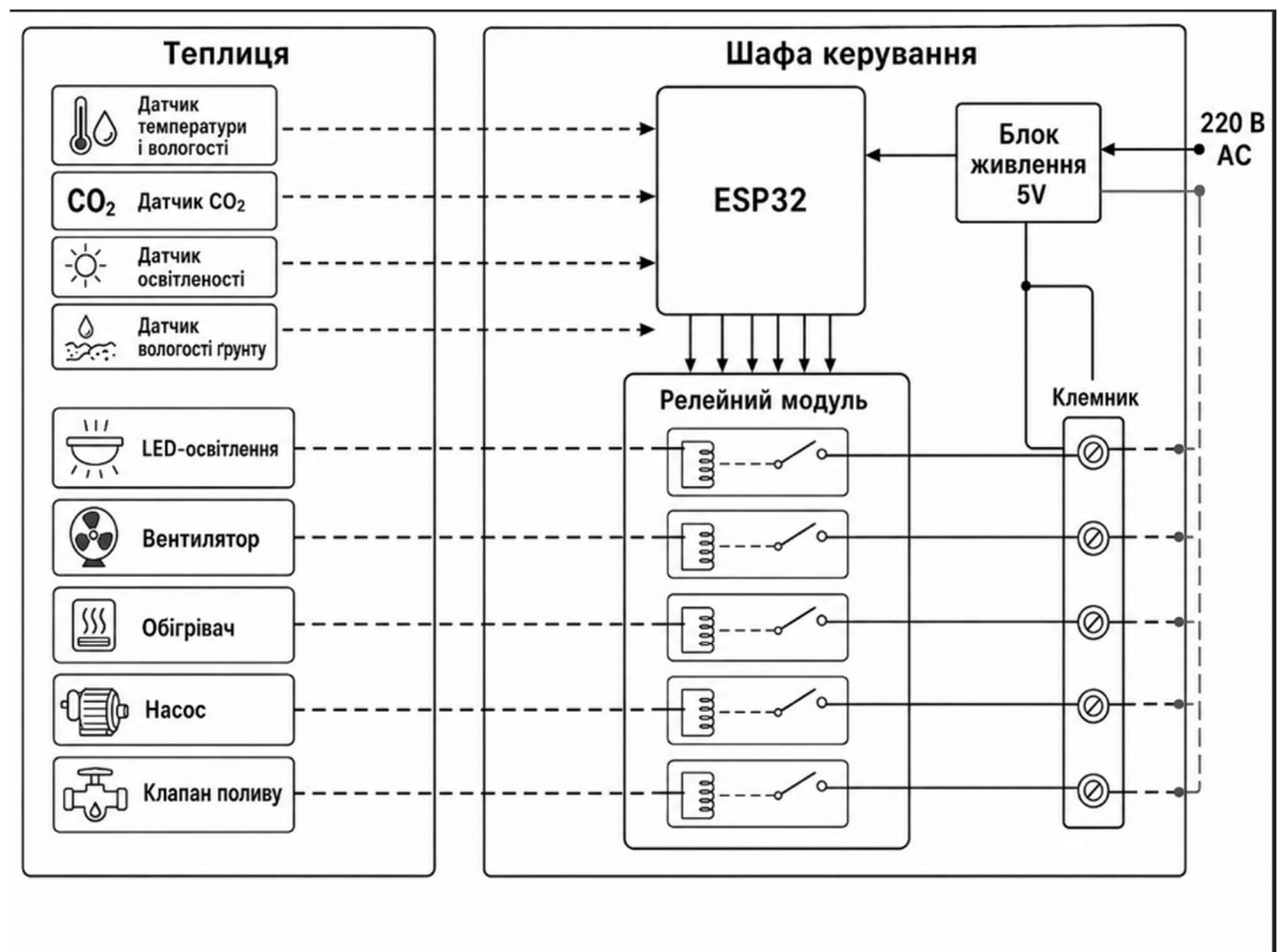


Рисунок 2.3 – Ілюстративна схема інтеграції ESP32

Зм.	Арк.	№ докум.	Підпис	Дата

2.4 Інформаційна модель, бази даних і потоки повідомлень

Інформаційна модель системи повинна відповідати фізичній структурі групи теплиць. Якщо в реальному середовищі існують група, окрема теплиця, зона вирощування, сенсорна точка і модуль керування, то ці ж об'єкти повинні бути відображені в структурі даних. Інакше система не зможе коректно пояснити, де саме виникло відхилення, який сенсор його зафіксував і який модуль керування може бути використаний для реагування.

На верхньому рівні моделі розміщується група теплиць. Вона об'єднує кілька об'єктів, які можуть мати різні культури, режими роботи та кількість зон. Нижче розташовується сутність теплиці, яка описує конкретний фізичний об'єкт. Кожна теплиця містить зони, а зона пов'язується з сенсорами, профілями рослин, правилами та доступними модулями керування. Така ієрархія дозволяє виконувати адресні запити та не змішувати телеметрію різних об'єктів.

Телеметричне повідомлення повинно містити не лише числове значення. Воно має включати ідентифікатор групи, теплиці, зони, сенсора, тип метрики, одиницю вимірювання, якість показника та часову мітку. Завдяки цьому сервер може перевірити, чи зареєстроване джерело даних, чи відповідає значення допустимому діапазону, чи можна використовувати його для оновлення поточного стану.

Командні повідомлення мають іншу логіку, але таку саму вимогу до адресності. Вони повинні містити цільову групу, теплицю, зону, назву модуля керування, дію, тривалість, джерело команди та пояснення причини. Якщо команда не має повної адреси або не відповідає існуючому модулю керування, вона не повинна бути опублікована в MQTT. Такий підхід зменшує ризик випадкового впливу на іншу теплицю.

У системі використовуються два класи сховищ. Часові ряди призначені для телеметрії, графіків і аналізу динаміки. Структуроване сховище використовується для опису теплиць, зон, користувачів, профілів культур,

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

правил, команд, пропозицій і журналів. Для AI-рівня додатково потрібні документи бази знань і векторні представлення, які дають змогу виконувати семантичний пошук контексту.

Телеметричний потік рухається від ESP32 до брокера й далі до серверної логіки. Командний потік рухається від сервера до брокера й конкретного edge-вузла. Подієвий потік формується під час спрацювання правил, зміни стану команди або втрати зв'язку. Аналітичний потік використовується AI-агентом для формування пояснень, добових підсумків і рекомендацій.

Сутності інформаційної моделі подано в таблиці 2.4. Вона не деталізує всі поля бази даних, а фіксує основні об'єкти, без яких система не може забезпечити коректну роботу групи теплиць у системі.

Таблиця 2.4 – Програмно-апаратні групи компонентів системи

Сутність	Що описує	Зв'язок із системою
GreenhouseGroup	Група теплиць	Верхній рівень структури об'єктів
Greenhouse	Окрема теплиця	Містить зони, сенсори та модулі керування
Zone	Зона вирощування	Має профіль рослин і правила контролю
Sensor	Джерело телеметрії	Формує вимірювання з часовою міткою
ControlModule	Канал впливу на середовище	Отримує підтвержені команди
Command	Керуюча дія або пропозиція	Фіксується в журналі станів
AIConversation	Діалог із користувачем	Зберігає запити, відповіді та tool-виклики

У такій моделі будь-яке повідомлення або команда можуть бути віднесені до конкретного об'єкта. Це важливо для подальшого тестування, тому що помилка в адресності одразу проявляється в інтерфейсі: дані можуть потрапити не в ту зону, команда може бути недоступною або AI-агент не зможе пояснити, для якої теплиці сформовано рекомендацію. Саме тому інформаційна модель є не допоміжною, а базовою частиною кіберфізичної архітектури.

2.5 Концептуальне розгортання контейнерної інфраструктури та протоколи зв'язку

Модель розгортання описує, як програмні сервіси запускаються на сервері та як вони взаємодіють із фізичними або емульованими ESP32-вузлами. Для цієї системи доцільно використовувати контейнерний підхід, оскільки він дозволяє запускати web-інтерфейс, серверну логіку, MQTT-брокер, бази даних і AI-агент як окремі сервіси в межах одного репозиторію. Це спрощує повторне розгортання, тестування і перенесення проєкту між робочими середовищами [24,25].

Сервер або робоча станція виконує роль Docker Host. У середині нього розміщуються контейнери інтерфейсу користувача, API, AI-агента, MQTT-брокера, сховища часових рядів і PostgreSQL із pgvector. Локальні ESP32-вузли не входять до складу контейнерів, оскільки вони працюють як окремі edge-пристрої в мережі. Вони підключаються до MQTT-брокера через Wi-Fi і взаємодіють із системою тільки через topic-и телеметрії та команд.

Користувацький браузер взаємодіє з інтерфейсом через HTTPS і WebSocket. HTTPS використовується для звичайних запитів сторінок, налаштувань і API, а WebSocket потрібний для оновлення стану в реальному часі. Інтерфейс не повинен напряму звертатися до баз даних або MQTT-брокера. Усі операції проходять через серверний API [26,27], який виконує перевірку прав, формує відповіді, створює пропозиції та керує журналами [28,29].

MQTT-брокер є єдиним вузлом обміну повідомленнями з ESP32. Це означає, що локальні вузли не підключаються напряму до сховищ даних і не взаємодіють із AI-рівнем. Усі показники потрапляють на сервер через брокер, після чого проходять валідацію, нормалізацію і запис. У зворотному напрямку сервер надсилає тільки підтверджені команди, які вже пройшли перевірку правил і мають повну адресу цільової зони [30,31].

AI-агент працює як окремий логічний сервіс. Він використовує дозволені інструменти для отримання телеметрії, поточного стану, журналів і фрагментів бази знань. Для генерації текстової відповіді використовується зовнішній сервіс OpenRouter, доступ до якого потребує API-ключа. Цей ключ повинен зберігатися в конфігурації середовища, а не в коді репозиторію. Якщо ключ відсутній або модель недоступна, AI-функції мають переходити в обмежений режим із зрозумілим повідомленням для користувача [32,33].

Режим керування визначає, що відбувається після підтвердження команди. У симуляційному режимі змінюється внутрішній стан програмної моделі, що зручно для демонстрації без фізичного обладнання. У режимі MQTT-пристроїв команда публікується в топі конкретної теплиці та може бути отримана реальним або Wokwi-емульованим ESP32-вузлом. Завдяки цьому одна система може використовуватися і для навчального тестування, і для підключення до фізичного стенду.

Усі ключові протоколи мають власну роль. HTTP використовується для стандартних запитів між інтерфейсом і API, WebSocket - для оперативного оновлення подій, MQTT - для IoT-обміну з edge-вузлами, SQL - для роботи зі структурованими даними, а HTTPS - для звернення до зовнішнього AI-провайдера. Таке розділення каналів допомагає уникнути ситуації, коли команда оминає серверну валідацію або телеметрія записується без перевірки.

Діаграма, яка показує, що MQTT-зв'язки з фізичними та тестовими вузлами проходять через брокер, тоді як web-інтерфейс, API, AI-агент і сховища

працюють усередині серверного середовища. Таке контейнерне розгортання, внутрішні сервіси й зв'язок із ESP32-вузлами зображено на рисунку 2.4.

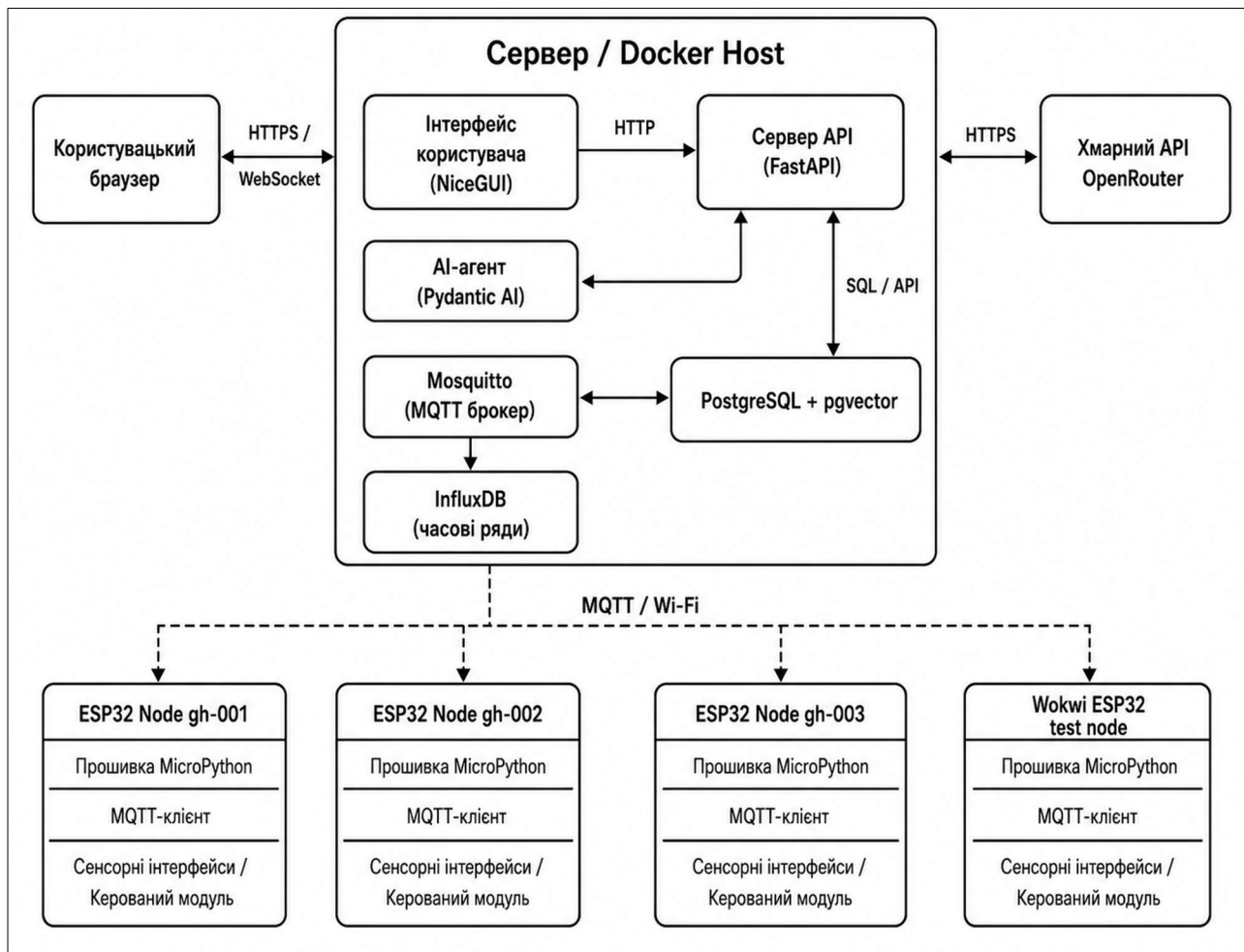


Рисунок 2.4 – Діаграма розгортання контейнерної інфраструктури та взаємодії з ESP32-вузлами

Продемонстрований контейнерний підхід, дозволяє ізолювати окремі сервіси та спростити їх незалежне оновлення без впливу на інші компоненти інфраструктури. Наприклад, зміна AI-моделі або оновлення MQTT-брокера не потребує повного перевстановлення всієї системи, а виконується через оновлення відповідного контейнера. Такий підхід також спрощує резервне копіювання, оскільки бази даних, конфігураційні файли та журнали можуть зберігатися у виділених образах Docker.

У процесі експлуатації система може працювати одночасно з фізичними ESP32-вузлами та емуляторами Wokwi. Такий підхід дає можливість тестувати нові алгоритми керування, перевіряти логіку AI-агента та моделювати аварійні ситуації без ризику пошкодження обладнання або впливу на реальні теплиці. Завдяки цьому контейнерна архітектура одночасно виконує роль виробничого та тестового середовища [34].

Використання централізованої серверної моделі також спрощує масштабування системи. Для підключення нової теплиці достатньо додати новий ESP32-вузол, створити відповідні записи в базі даних та налаштувати ключ для MQTT.

2.6 Розробка сценарію текстової взаємодії користувача з AI-агентом

Однією з особливостей розроблюваної кіберфізичної системи є наявність інтелектуального інтерфейсу, через який користувач може взаємодіяти з системою природною мовою. На відміну від звичайної панелі моніторингу, де оператор самостійно аналізує графіки, таблиці й поточні значення, AI-агент має виконувати роль пояснювального та рекомендаційного рівня. Його завдання полягає не в прямому керуванні фізичними пристроями, а в аналізі доступних даних, узагальненні стану теплиць, поясненні відхилень і формуванні безпечних пропозицій для оператора.

Такий підхід є особливо важливим для групи теплиць, оскільки оператору потрібно швидко оцінити стан декількох об'єктів одночасно. Наприклад, запит “Як там мої теплиці?” не містить точної назви теплиці, зони або параметра, але система повинна інтерпретувати його як запит на загальний огляд. У цьому випадку AI-агент має отримати дані про всі активні теплиці, переглянути останню телеметрію, перевірити наявні попередження, порівняти показники із профілями рослин і сформувавши відповідь у зрозумілому для користувача вигляді.

Проектування такого сценарію потребує чіткого розмежування між текстовою взаємодією, доступом до даних і керуванням модулями керування. AI-агент не повинен напряму публікувати MQTT-команди або змінювати стан насоса, вентилятора, обігрівача чи освітлення. Його доступ до системи має бути обмежений набором дозволених інструментів, через які він може отримати телеметрію, історію, попередження, профілі культур і фрагменти бази знань. Якщо у відповіді формується рекомендація щодо дії, вона повинна перетворюватися лише на пропозицію, яка додатково перевіряється серверною логікою та підтверджується користувачем.

Передбачається, що запит користувача проходить кілька послідовних етапів. Спочатку повідомлення вводиться в web-інтерфейсі та передається до серверного API. Сервер визначає контекст користувача, доступні теплиці та допустимі операції. Після цього запит передається до AI-агента, який не працює з базами даних напряму, а викликає дозволени інструменти серверної частини. Такі інструменти можуть повертати зведення по групі теплиць, останні значення сенсорів, добову динаміку параметрів, активні попередження, історію команд і релевантні фрагменти бази знань [35,36].

Особливе значення має використання RAG-підходу. Якщо користувач запитує не лише поточний стан, а й просить пояснити причину проблеми або надати рекомендацію, AI-агент повинен спиратися не тільки на мовну модель, а й на локальний контекст системи. До такого контексту належать профілі культур, допустимі межі вологості, правила догляду, опис зон, попередні події та історія вимірювань. Це дозволяє сформулювати відповідь, яка враховує конкретну теплицю, конкретну культуру й поточні умови, а не є загальною порадою без прив'язки до системи [37,38].

У проєктованому сценарії відповідь AI-агента повинна мати пояснювальну структуру. Доцільно, щоб вона містила короткий загальний висновок, перелік спостережень, можливі причини відхилень і рекомендації. Якщо даних недостатньо, агент повинен прямо зазначити обмеження аналізу, наприклад

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		

відсутність профілю культури, застарілу телеметрію або відсутність показників за певний період. Така поведінка підвищує довіру до системи, оскільки користувач бачить не лише результат, а й підстави, на яких його сформовано.

Окремо слід передбачити безпечну обробку запитів, у яких користувач просить виконати фізичну дію [57]. Наприклад, фраза “увімкни насос у третій теплиці” не повинна одразу призводити до виконання команди. AI-агент може розпізнати намір, уточнити цільову теплицю та зону, сформувану структуровану пропозицію, однак остаточна дія повинна пройти перевірку правил. Серверна логіка перевіряє допустиму тривалість, наявність актуальної телеметрії, стан модуля керування, права користувача та можливі обмеження. Тільки після цього пропозиція може бути показана оператору для підтвердження.

Для формалізації такого підходу в системі доцільно використовувати модель tool usage. Кожен інструмент AI-агента має чітко визначене призначення і не повинен виконувати зайві дії. Наприклад, один інструмент повертає поточний стан теплиць, інший отримує історію параметрів за період, третій шукає релевантні знання у векторному сховищі, четвертий створює пропозицію керуючої дії. Така декомпозиція дозволяє контролювати, що саме робить AI рівень, і забезпечує простежуваність його взаємодії з системою [39,40].

Додатково важливо враховувати, що природномовна взаємодія з системою не повинна обмежуватися лише відповідями у форматі “норма/помилка”. AI-рівень має забезпечувати контекстний аналіз отриманих даних, виявляти взаємозв’язки між параметрами та пояснювати їх зрозумілою для оператора мовою.

Наприклад, якщо в одній із теплиць спостерігається підвищення температури разом зі зниженням вологості повітря, система повинна не лише зафіксувати перевищення порогового значення, а й пояснити можливий вплив цього стану на рослини та ризики пересушування ґрунту. Такий підхід дозволяє використовувати AI-агента не як простий інтерфейс до даних, а як інструмент підтримки прийняття рішень.

Крім аналізу поточного стану, AI-агент повинен враховувати часовий контекст і історичну динаміку параметрів. Це особливо важливо у випадках, коли короточасне відхилення не є критичним, але систематична тенденція може свідчити про несправність обладнання або неправильний режим вирощування.

Наприклад, поступове зниження вологості ґрунту в окремій зоні протягом декількох годин може бути ознакою недостатньої роботи модуля поливу або некоректного налаштування порогів керування. Завдяки використанню історичних даних, журналів подій та RAG-механізму AI-рівень здатний формувати не лише реактивні відповіді, а й рекомендації превентивного характеру для підтримання стабільного мікроклімату.

Загалом процес обробки запиту користувача до AI-чату зображено на рисунку 2.5.

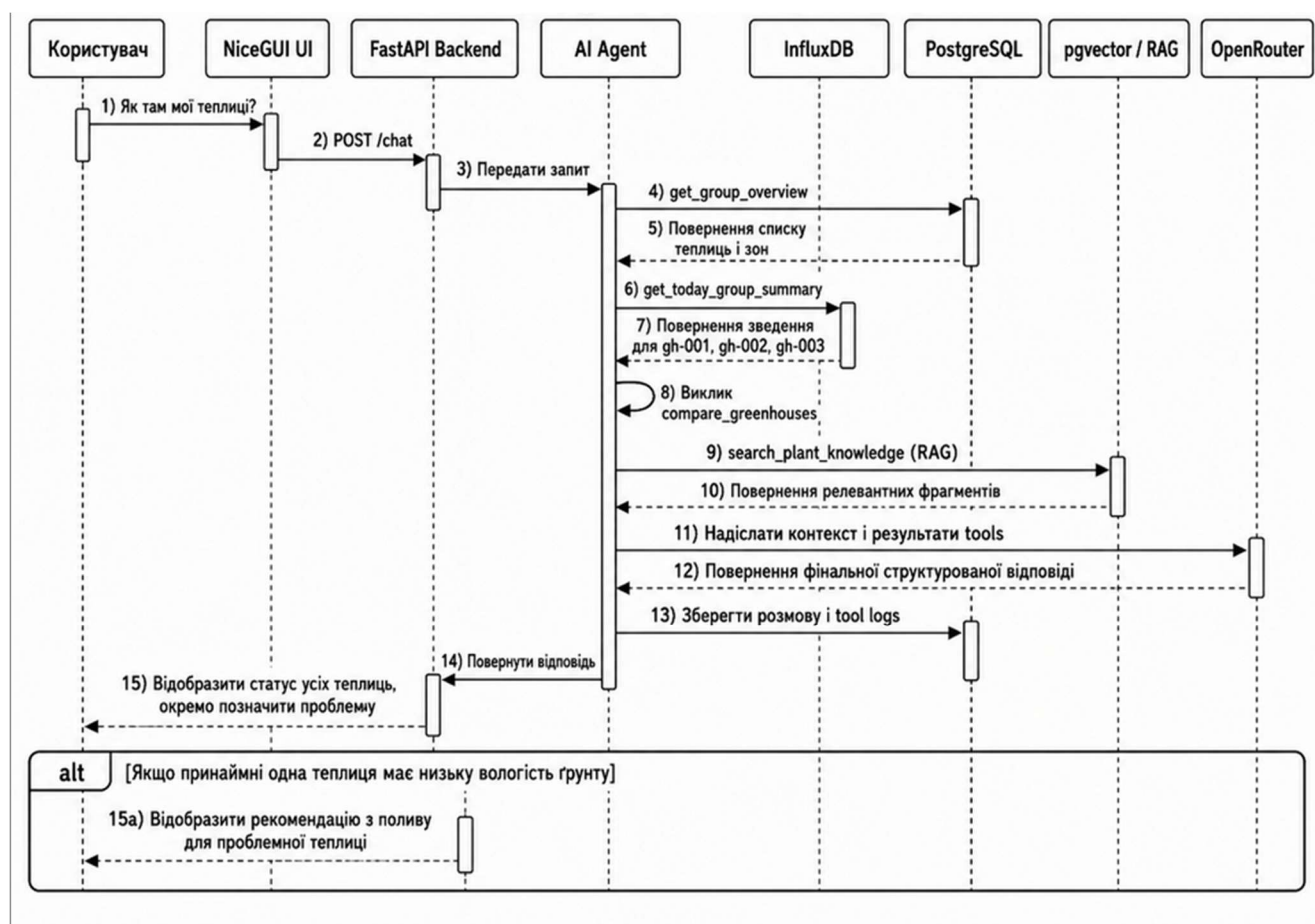


Рисунок 2.5 – Діаграма послідовності запиту користувача про стан теплиць

2.7 Висновки до другого розділу

У межах другого розділі сформовано архітектурну та концептуальну основу кіберфізичної системи керування мікрокліматом групи теплиць, розглянуто фізичну організацію тепличної інфраструктури, принципи зонування, особливості розміщення сенсорів і модулів керування, а також вимоги до монтажу локальних вузлів на базі ESP32, визначено, що коректне розташування сенсорного обладнання та розділення сигнальних і силових ліній безпосередньо впливають на достовірність телеметрії та стабільність роботи системи.

У розрізі проєктування програмно-апаратної частини обґрунтовано використання ESP32 як edge-вузла для збору телеметрії та прийому підтверджених команд, продемонстровано взаємодію локальних вузлів із серверною логікою через MQTT-брокер, а також роль серверного рівня у валідації повідомлень, журналюванні подій, перевірці правил і формуванні безпечних пропозицій керування, побудовано компонентну модель системи, яка відображає зв'язки між web-інтерфейсом, API, AI-рівнем, сховищами даних та IoT-інфраструктурою.

Окрему увагу приділено інформаційній моделі та потокам повідомлень, визначено структуру сутностей, що описують групи теплиць, зони, сенсори, модулі керування, команди та AI-взаємодію. Також відображено, що адресна організація телеметрії й команд є критичною умовою для масштабування системи та уникнення помилкового впливу на інші об'єкти. Також обґрунтовано використання окремих сховищ для часових рядів, структурованих даних і RAG-контексту.

У розділі також описано контейнерну модель розгортання системи та використання Docker-інфраструктури для ізольованого запуску web-інтерфейсу, серверної логіки, MQTT-брокера, AI-агента та баз даних. Визначено роль основних протоколів взаємодії HTTP, WebSocket, MQTT та HTTPS у

забезпеченні стабільного обміну між компонентами системи та фізичними ESP32-вузлами.

Крім цього, розроблено концепцію текстової взаємодії користувача з AI-агентом. Показано, що AI-рівень використовується як інструмент пояснення стану теплиць, аналізу телеметрії та формування рекомендацій, а не як механізм прямого керування обладнанням. Запропонований підхід із використанням tool-based взаємодії та RAG-механізму забезпечує контрольовану інтеграцію мовної моделі з локальним контекстом системи та підвищує безпечність прийняття рішень.

Отримані результати другого розділу формують завершену проєктну основу для подальшої практичної реалізації програмних модулів, інтеграції ESP32-вузлів, побудови web-інтерфейсу, реалізації AI-чату.

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ КЕРУВАННЯ МІКРОКЛІМАТОМ ГРУПИ ТЕПЛИЦЬ

3.1 Реалізація локального вузла ESP32 і фізичного монтажу

Спершу необхідно побудувати вузол ESP32, він реалізовано як пристрій, що має власні ідентифікатори `group_id`, `greenhouse_id`, `zone_id` і `device_id`.

Завдяки цьому однакова прошивка може використовуватися для різних теплиць, а відмінності між вузлами задаються через конфігураційний файл. Такий підхід спрощує масштабування і зменшує ризик помилки під час додавання нових зон.

Під час запуску вузол спочатку переводить усі канали керування в безпечний стан. Насос, вентилятор, обігрівач і лампа мають бути вимкнені до встановлення мережевого з'єднання та підключення до MQTT-брокера. Це важливо для реального тепличного середовища, оскільки неконтрольоване ввімкнення поливу або обігріву після перезавантаження може спричинити пошкодження рослин або перевитрати ресурсів.

Сенсорна частина реалізується через окремі функції зчитування. DHT22 або аналогічний датчик формує значення температури й вологості повітря. Вологість ґрунту, освітленість і CO₂ можуть бути отримані з аналогових сенсорів або, у тестовому варіанті, з потенціометрів. Отримані сирі значення масштабуються до робочих діапазонів і передаються як стандартизована телеметрія [41,42].

Формат телеметрії містить `message_id`, якість показника, ідентифікатори об'єктів, тип метрики, значення та часову мітку. Серверна частина очікує саме таку структуру, тому локальний вузол не публікує довільні текстові повідомлення. Це дозволяє виконувати автоматичну перевірку повідомлення та не використовувати неповні або некоректні дані.

Обробка команд на ESP32 також містить перевірку. Навіть якщо команда вже пройшла серверну валідацію, локальний вузол повторно порівнює цільову

групу, теплицю і зону зі своєю конфігурацією. Якщо адреса не збігається, команда ігнорується. Якщо назва модуля керування не входить до дозволеного переліку, дія також не виконується. Такий дубльований контроль підвищує безпеку системи.

Дані про призначення ліній ESP32, які раніше доцільно було подати у вигляді таблиці, у реалізаційному розділі використано як пояснення до схеми. У прототипі DHT22 підключено до D15, вологість ґрунту - до D34, освітленість - до D35, CO₂ - до D32. Канал насоса пов'язано з D25, вентилятор - з D26, обігрівач - з D27, а LED-освітлення - з D14. Такий розподіл дозволяє чітко відокремити сенсорні входи від виходів керування [43].

Локальний вузол виконує роль проміжного рівня між фізичними сенсорами та серверною частиною системи. Він не здійснює складний аналіз або довготривале зберігання інформації, а відповідає за стабільне зчитування параметрів середовища, формування телеметрії, прийом підтверджених команд і передачу поточного стану модулів керування. Такий підхід дозволяє централізувати логіку безпеки та аналітики на серверному рівні, залишаючи ESP32 лише функції збору даних і локального виконання дозволених дій.

Під час реалізації важливо враховувати, що ESP32 не може безпосередньо керувати силовими навантаженнями. Насоси, вентилятори, обігрівачі та LED-освітлення повинні підключатися через релейні або транзисторні модулі, які забезпечують електричну ізоляцію між низьковольтною логікою контролера та силовими лініями живлення. Додатково необхідно передбачити окреме живлення для виконавчих елементів, захист від короткого замикання та можливість ручного відключення каналів у випадку аварійної ситуації.

Окрему роль у побудові відіграє організація фізичних з'єднань між сенсорами, контролером і модулями керування. Неправильне розведення сигнальних та силових ліній може призводити до нестабільних показників, перешкод або помилкових спрацювань. Саме тому схема підключення повинна відображати не лише логічний зв'язок між компонентами, а й реальну структуру

монтажу локального вузла в тепличному середовищі. Фізична схема підключення показує, як саме ці лінії використовуються в реалізованому вузлі. Вона поєднує плату ESP32, сенсорні модулі, релейні канали та індикатори або реальні модулі керування. Схему реалізованого підключення ESP32 до сенсорів і модулів керування подано на рисунку 3.1.

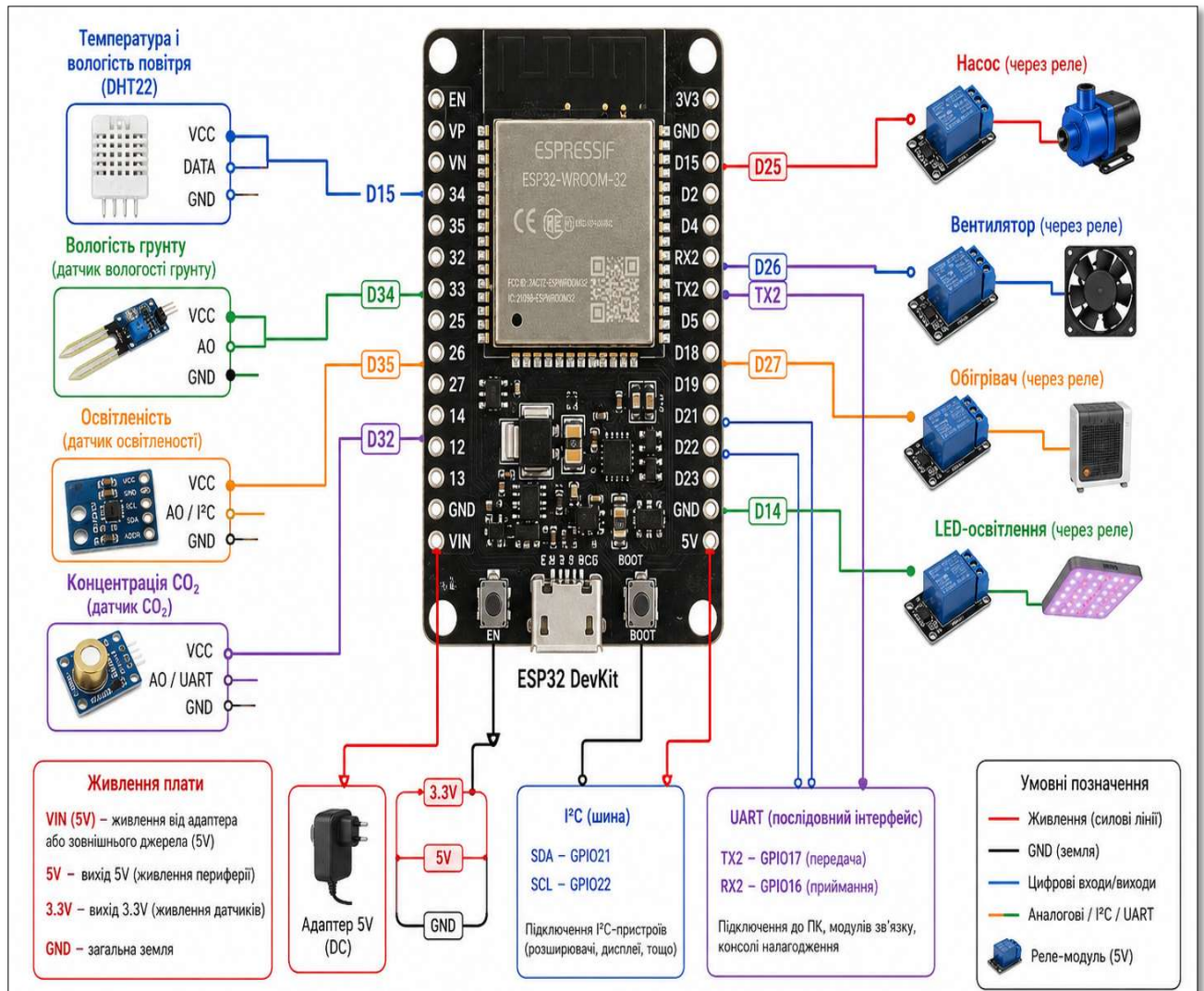


Рисунок 3.1 – Схема реалізованого підключення ESP32 до сенсорів та модулів керування

Після перевірки схеми підключення важливо забезпечити відповідність між назвами в прошивці, торіс-ами MQTT і записами в серверній базі. Якщо в прошивці модуль називається rmp, то серверна команда повинна

використовувати ту саму назву. Якщо в системі використовується інша назва, наприклад `irrigation_pump`, вона повинна бути узгоджена в усіх частинах: інтерфейсі, API, командному сервісі та прошивці [44].

На першому етапі тестування реальні силові навантаження доцільно замінювати LED-індикаторами. Це дає змогу безпечно перевірити реакцію каналів D25, D26, D27 і D14 без підключення насоса, вентилятора, обігрівача або лампи. Після підтвердження коректності логіки ці канали можуть бути підключені до релейного модуля і реального обладнання.

3.2 Реалізація серверних модулів, телеметрії та журналювання

Серверна частина реалізує центральний контур обробки даних. Вона приймає телеметрію від MQTT-брокера, приводить її до внутрішньої структури, перевіряє коректність повідомлень, записує історію, оновлює поточний стан зон і створює службові події. На відміну від локального вузла, який працює з конкретною теплицею або зоною, сервер бачить усю групу об'єктів і може зіставляти дані між різними теплицями.

Приєм телеметрії реалізовано як окремий `ingestion`-рівень. Його завдання полягає в тому, щоб не допустити потрапляння некоректних повідомлень до основної логіки. Перевіряється наявність `group_id`, `greenhouse_id`, `zone_id`, `sensor_id`, `metric`, `value`, `quality` і `timestamp`. Якщо хоча б один обов'язковий елемент відсутній або тип даних не відповідає очікуваному формату, повідомлення не оновлює стан системи та не бере участі у формуванні правил.

Для коректного повідомлення виконуються два пов'язані кроки. Перший - запис до сховища часових рядів, що використовується для графіків, трендів і ретроспективного аналізу. Другий - оновлення актуального стану зони, який потрібний для швидкого відображення в інтерфейсі та відповідей AI-агента. Завдяки такій організації система одночасно зберігає історію і підтримує оперативний стан.

Перевірка правил виконується тільки після того, як дані пройшли валідацію та були пов'язані з конкретною зоною. Наприклад, низьке значення вологості ґрунту може створити попередження або пропозицію поливу лише тоді, коли система точно знає, до якої теплиці й зони належить вимірювання, який профіль рослин використовується та чи не виконувалася подібна дія нещодавно.

Журналювання реалізовано окремо від телеметрії. Телеметрія описує фізичний стан середовища, а журнал команд описує наміри, підтвердження, відмови, виконання та помилки. Коли користувач або AI-агент створює пропозицію, сервер фіксує її у статусі pending. Після підтвердження створюється запис про перехід до виконання, а після результату MQTT-взаємодії оновлюється фінальний стан.

Серверна логіка не змішує інформаційні запити й фізичні дії. Запит користувача до AI-чату може прочитати телеметрію, отримати добові підсумки, перевірити активні попередження або сформулювати пояснення. Проте для будь-якої дії, яка впливає на модуль керування, створюється пропозиція, а не безпосередня MQTT-команда. Цей поділ є основою безпечної взаємодії з фізичною інфраструктурою.

Якщо MQTT-брокер тимчасово недоступний, сервер не повинен втрачати інформацію про створену пропозицію. Вона залишається в журналі зі зрозумілим статусом, а оператор бачить, що команда не була доставлена. Якщо ж відсутні нові дані від сенсора, система повинна відобразити проблему як службову подію, а не підставляти випадкові значення.

Серверний маршрут обробки даних доцільно подати у вигляді діаграми, оскільки він показує не лише послідовність дій, а й точки прийняття рішення: перевірку структури повідомлення, запис даних, порівняння з правилами та формування реакції. Діаграму обробки телеметрії та реакції системи наведено на рисунку 3.2.

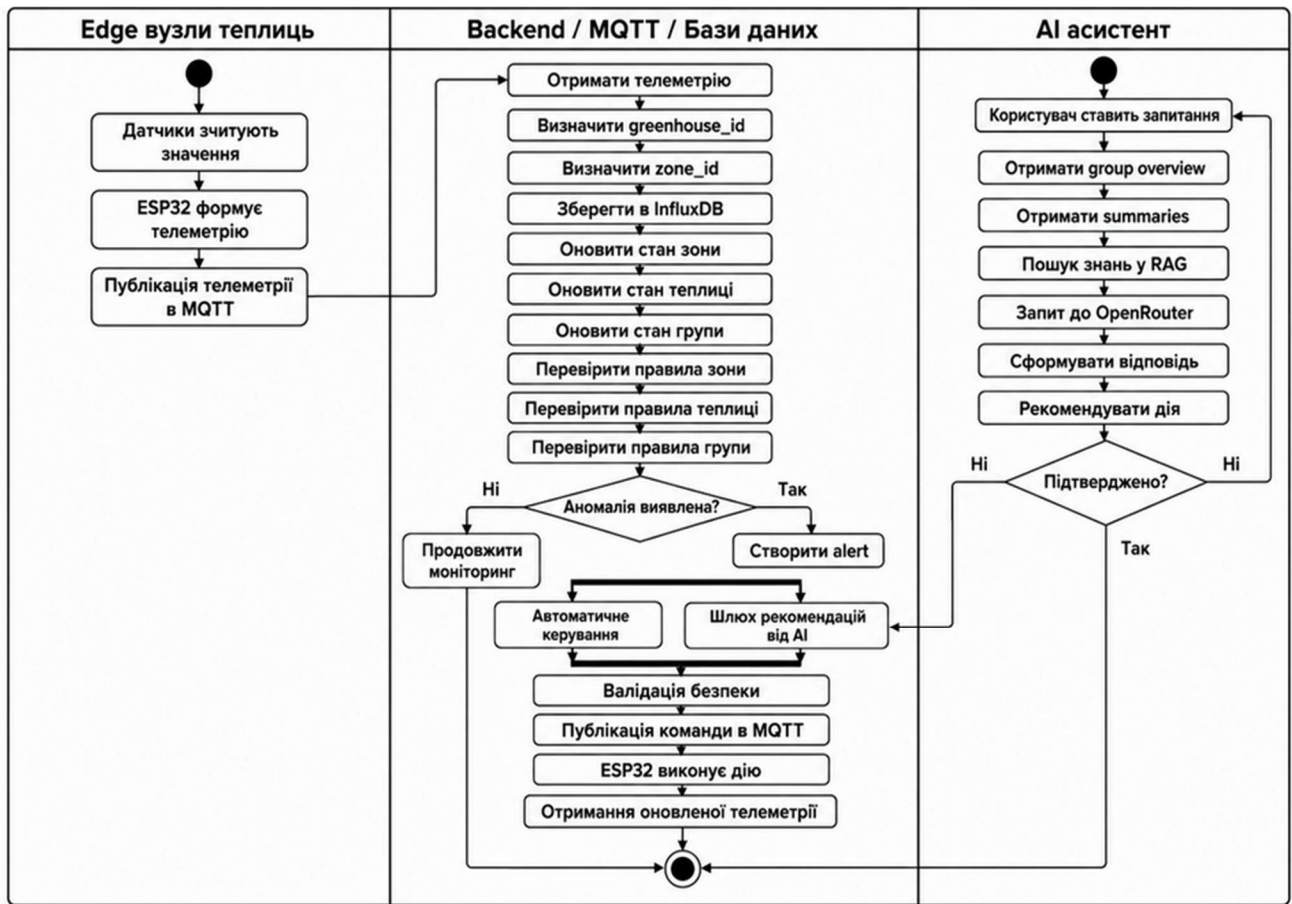


Рисунок. 3.2 – Діаграма обробки телеметрії в групі теплиць

3.3 Опис створення схеми бази даних та таблиць системи

Опис процесу створення баз даних є окремим етапом реалізації, оскільки система працює не лише з поточними значеннями сенсорів, а й з довідковими сутностями, журналами, командами, профілями рослин, AI-діалогами та історичними часовими рядами. Для кіберфізичної системи керування групою теплиць база даних виконує не допоміжну, а системоутворювальну роль: вона зв'язує фізичні об'єкти з цифровою моделлю та забезпечує адресність кожного показника, попередження і керуючої пропозиції.

У реалізованому прототипі використано поділ даних на структуровані записи і часові ряди. Структуровані записи зберігають сутності предметної області: групи теплиць, теплиці, зони, edge-вузли, сенсори, модулі керування, профілі рослин, користувацькі дії, команди, алерти й AI-взаємодію. Часові ряди

зберігають значення мікроклімату в часі: температуру, вологість повітря, вологість ґрунту, освітленість, CO₂ і стани модулів керування. Такий поділ дозволяє не перевантажувати реляційну базу великою кількістю телеметричних записів і водночас зберігати повну логіку зв'язків між об'єктами.

Створення схеми PostgreSQL починається з доменної ієрархії. Верхнім рівнем є `greenhouse_groups`, що описує групу теплиць як єдину операційну область. Нижче розташовуються `greenhouses`, які відповідають окремим фізичним теплицям. Кожна теплиця містить `greenhouse_zones` - зони вирощування, прив'язані до конкретних культур, сенсорів і модулів керування. Саме зона є основною одиницею аналізу, тому більшість команд, алертів і AI-висновків повинні мати `zone_id`.

Після створення базової ієрархії формуються таблиці пристроїв. `edge_nodes` описує локальні ESP32-вузли та їхню прив'язку до теплиці або зони. `sensor_registry` містить перелік сенсорів, тип метрики, одиницю вимірювання та статус сенсора. `control_modules` описує канали керування: насос, вентилятор, обігрівач, освітлення або інший модуль, який може змінювати фізичний стан середовища.

Окремий блок схеми пов'язаний із рослинами та правилами. `plant_profiles` зберігає типові вимоги культури до вологості ґрунту, температури, освітлення та інших параметрів. `plant_batches` описує конкретну партію рослин у зоні, а `control_setpoints` містить робочі пороги для певної культури і стадії росту. Завдяки цьому AI-агент і серверні правила можуть оцінювати стан не абстрактно, а з урахуванням того, які саме рослини знаходяться в зоні.

Командна частина бази даних реалізується через журнал команд і пропозицій. Коли користувач або AI-агент формує намір виконати дію, запис спочатку створюється зі статусом `pending`. Після підтвердження або відхилення статус змінюється, а сама команда або надсилається в MQTT, або залишається в журналі як заблокована чи скасована. Такий життєвий цикл дозволяє відстежити кожен дію від моменту створення до фінального результату.

Для AI рівня створюються таблиці ai_conversations, ai_messages і ai_tool_calls. Вони зберігають діалоги, повідомлення користувача, відповіді агента та результати інструментальних викликів. Це важливо для діагностики, оскільки дозволяє побачити, які саме дані отримав AI-агент перед формуванням відповіді. Окремо створюються rag_documents і rag_chunks, які містять текстову базу знань і фрагменти для векторного пошуку через pgvector.

Таблиці PostgreSQL, які використовуються для збереження структурованих даних системи, подано в таблиці 3.1.

Таблиця 3.1 – Таблиці PostgreSQL для структурованих даних системи

Таблиця / сховище	Призначення	Використання в системі
greenhouse_groups	Опис групи теплиць як верхнього рівня ієрархії	Об'єднання кількох теплиць в одну логічну систему моніторингу та керування
greenhouses	Опис окремих теплиць у межах групи	Прив'язка зон, edge-вузлів, команд, попереджень і AI-аналізу до конкретної теплиці
greenhouse_zones	Опис зон вирощування всередині теплиці	Адресність телеметрії, команд, профілів рослин і рекомендацій для конкретної зони
edge_nodes	Реєстрація фізичних або емуляційних ESP32-вузлів	Перевірка джерела телеметрії, MQTT-ідентифікація, контроль активності вузла в системі

Кінець таблиці 3.1

sensor_registry	Реєстр сенсорів, закріплених за зонами	Визначення доступних метрик: температура, вологість, CO ₂ , освітленість, вологість ґрунту
control_modules	Реєстр модулів керування у зонах	Перевірка цільового каналу перед виконанням команди: насос, вентилятор, обігрівач, освітлення
plant_profiles	Довідник культур і стадій росту з рекомендованими межами параметрів	Оцінювання відповідності мікроклімату потребам конкретної культури
plant_batches	Опис партій рослин, висаджених у конкретних зонах	Прив'язка телеметрії та рекомендацій до реальної культури, що вирощується в зоні

Під час створення таблиць важливо дотримуватися обмежень цілісності. Запис про зону не повинен існувати без теплиці, сенсор має бути прив'язаний до конкретного вузла або зони, а команда повинна посилатися на реальний модуль керування. Для цього застосовуються зовнішні ключі, унікальні ідентифікатори та перевірка допустимих статусів. Такі обмеження зменшують ризик появи «висячих» записів, які неможливо коректно відобразити в інтерфейсі або використати в AI-відповіді.

Для полів із гнучкою структурою доцільно використовувати JSONB. Наприклад, правила поливу, додаткові параметри модулів керування, результат

tool-виклику або службовий контекст AI-діалогу можуть мати різний склад залежно від сценарію. JSONB дозволяє зберігати ці дані без надмірного дроблення схеми, але критичні поля, такі як greenhouse_id, zone_id, status, created_at і user_id, повинні залишатися окремими колонками [45,46].

Сховище часових рядів має іншу логіку. Воно не описує склад системи, а приймає багато однотипних записів із часовою міткою. Для кожного показника зберігаються metric, value, unit, quality, group_id, greenhouse_id, zone_id, sensor_id і timestamp [47]. Така структура дає змогу швидко будувати графіки, отримувати останні значення, формувати добові підсумки та порівнювати теплиці між собою [48,49].

Основні типи даних, що зберігаються в InfluxDB, подано в таблиці 3.2.

Таблиця 3.2 – Основні типи часових даних InfluxDB

Тип даних	Що зберігається	Для чого використовується
Параметри повітря	Температура і вологість повітря з часовою міткою	Графіки, добові підсумки, виявлення перегріву або пересушування
Параметри ґрунту	Вологість ґрунту по зонах	Оцінювання потреби в поливі та аналіз тренду
Світло і CO ₂	Освітленість та концентрація CO ₂	Оцінка умов фотосинтезу та провітрювання
Стани керування модулів	pump_state, fan_power, heater_power, lamp_state	Порівняння надісланих команд із фактичним станом вузла на основі інформації з бази даних

Після створення схеми виконується початкове наповнення довідкових таблиць. На цьому етапі створюється група теплиць, кілька тестових теплиць, зони, профілі рослин і дозволені модулі керування. Також генеруються або задаються MQTT-облікові дані для зон, щоб локальні ESP32-вузли могли публікувати телеметрію тільки у власні топік-и. Такий підхід підвищує контрольованість системи та унеможлиблює випадкову підміну зони під час тестування.

Реалізація схеми бази даних також передбачає підготовку міграцій. Міграції потрібні для того, щоб структура бази могла відтворюватися в іншому середовищі без ручного створення таблиць. У межах одного репозиторію це дає можливість швидко розгорнути систему на новому комп'ютері, у тестовому контейнері або на сервері, зберігаючи однакову структуру таблиць і зв'язків [50,51].

Таким чином, процес створення баз даних у системі охоплює не лише технічне створення таблиць, а й формування цифрової моделі групи теплиць. Саме ця модель надалі використовується серверною логікою, web-інтерфейсом, AI-агентом, журналами та механізмами безпечного підтвердження команд.

3.4 Реалізація web-інтерфейсу, dashboard, AI-чату та ручного керування

Розробка web-інтерфейсу спрямована на те, щоб оператор міг працювати з системою без безпосереднього перегляду MQTT-повідомлень або записів бази даних. Інтерфейс повинен показувати стан групи теплиць у зрозумілому вигляді: поточні значення параметрів, графіки зон, контекст рослин, доступні пропозиції, ручне керування, результати команд і AI-чат. Усі ці блоки використовують дані серверного API, тому користувацький рівень не звертається напряду до сховищ або MQTT-брокера.

Основним екраном моніторингу є dashboard. Він агрегує дані по зонах і дозволяє швидко визначити, чи є проблема в конкретній теплиці. Графіки температури, вологості повітря та вологості ґрунту будуються за часовими

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 51
Зм.	Арк.	№ докум.	Підпис	Дата		

рядами. Користувач може змінювати період перегляду, порівнювати показники та перевіряти, чи була реакція системи після команди. Для навчального прототипу цього достатньо, щоб продемонструвати не лише статичні значення, а й динаміку середовища [52].

Сторінка окремої зони є детальнішим робочим екраном. На ній відображається телеметрія, профіль рослин, активні пропозиції та блок ручного керування. Така сторінка потрібна, коли оператор переходить від загального огляду до конкретної проблемної ділянки. Наприклад, якщо графік показує зниження вологості ґрунту, оператор відкриває відповідну зону, перевіряє поточне значення, бачить культурний контекст і може створити пропозицію поливу.

Ручне керування реалізовано через проміжний стан пропозиції. Кнопка в інтерфейсі не повинна одразу вмикати насос або інший модуль керування. Користувач задає параметри дії, після чого система створює пропозицію зі статусом очікування. Пропозиція проходить серверну перевірку і тільки після підтвердження може бути перетворена на MQTT-команду. Завдяки цьому ручне керування залишається контрольованим і не порушує загальний контур безпеки.

AI-чат інтегровано як окремий блок взаємодії з користувачем. Його призначення полягає в тому, щоб користувач міг запитати про стан теплиць природною мовою, а система сформулювала відповідь на основі фактичних даних. AI-агент використовує інструменти для отримання телеметрії, підсумків, профілів рослин, попереджень і RAG-контексту. Відповідь має бути пояснювальною: не лише показувати число, а й пояснювати, чи є це значення нормальним, які є обмеження аналізу і що доцільно зробити далі.

Окремо реалізовано журнал останніх команд. Він потрібний для швидкого контролю результатів: оператор повинен бачити, чи була дія виконана, скасована, заблокована або відхилена. Це особливо важливо у випадку, коли команда ініціювалася через AI-чат або створювалася як пропозиція.

Журнал забезпечує простежуваність і зменшує ймовірність повторного запуску дії через невпевненість користувача.

Сторінка конфігурації, сторінка профілів рослин, журнал помилок і реєстр зон не належать до щоденного dashboard, але забезпечують експлуатаційну придатність системи. Вони дозволяють підготувати AI-модель, вибрати режим керування, створити пороги для культур, зареєструвати зони й переглядати технічні збої.

3.5 Тестування системи на сценаріях моніторингу, ручного керування, AI-відповідей

Тестування системи виконувалося як перевірка повного циклу роботи: від візуалізації телеметрії до створення пропозицій, блокування небезпечних запитів, журналювання команд і перевірки ESP32-вузла. Метою тестування було підтвердити, що реалізований прототип працює не лише як набір окремих модулів, а як цілісна кіберфізична система з контрольованим потоком даних і команд [53].

Окремо перевірялося, чи збігається інформація на різних рівнях інтерфейсу. Якщо на графіку зони зафіксовано падіння вологості ґрунту, сторінка зони повинна показати поточне значення, AI-чат повинен аналізувати той самий об'єкт, а пропозиція поливу має створюватися саме для відповідної зони. Така узгодженість є критичною, оскільки система працює з групою теплиць і не має права змішувати показники різних об'єктів.

Перший етап тестування стосується сторінки зони. На цьому екрані перевіряється наявність поточної телеметрії, відображення контексту рослин, стану пропозицій і блоку ручного керування. Сторінка зони повинна бути достатньо інформативною для оператора, але не перевантаженою технічними деталями [54,55].

Під час перевірки сторінки зони також оцінюється зручність переходу від спостереження до керуючої дії. Оператор повинен бачити не лише числові значення параметрів, а й контекст їхнього застосування: до якої групи, теплиці та зони вони належать, яка партія рослин вирощується в цій зоні та чи є активні або очікувані пропозиції. Це дає змогу приймати рішення без переходу до технічних журналів або бази даних. Важливо, що блок ручного керування не виконує дію миттєво, а створює контрольовану пропозицію, яка надалі проходить підтвердження та перевірку правил безпеки. Приклад сторінки зони з телеметрією та блоком ручного керування подано на рисунку 3.3.

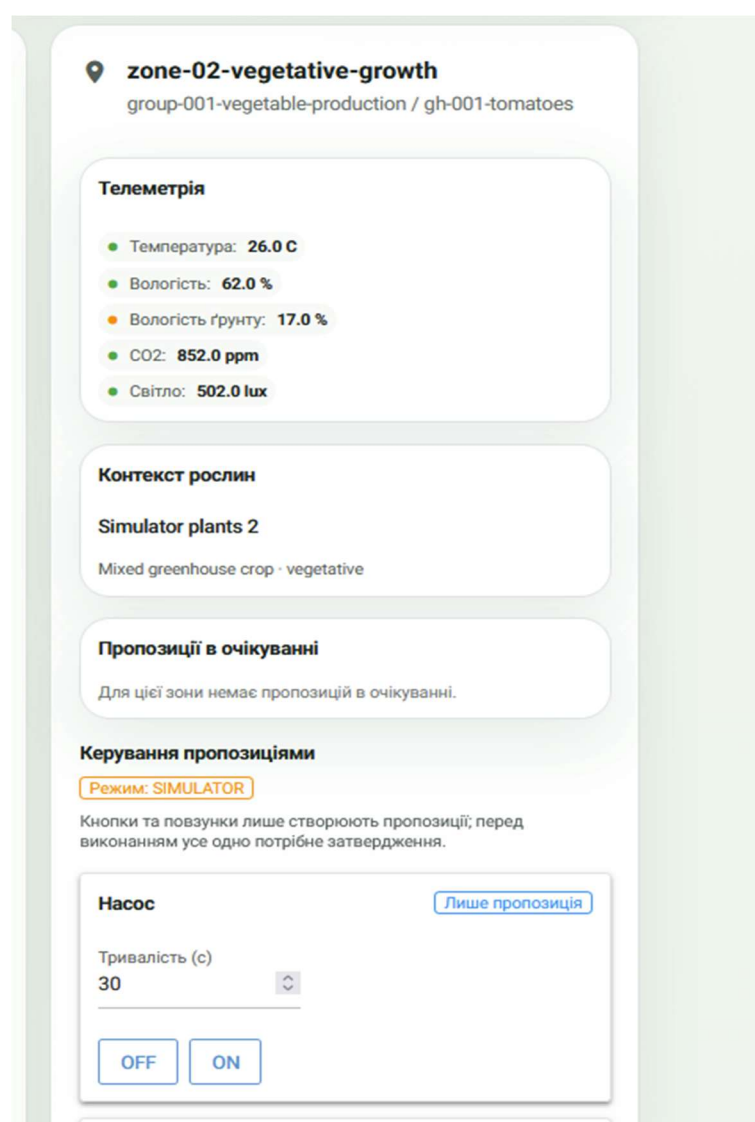


Рисунок 3.3 – Сторінка зони з телеметрією та блоком ручного керування

Після перевірки окремої зони було виконано тестування загальних графіків. Цей етап показує, чи може оператор оцінити динаміку параметрів без переходу до технічних журналів. Для тепличної системи графіки мають практичне значення, оскільки одиничне значення не завжди дає повну картину: важливо бачити тренд, стабільність параметра та момент різкої зміни.

У dashboard перевірялися кілька часових показників: температура, вологість повітря та вологість ґрунту. Для кожного графіка важливо, щоб дані відповідали вибраній зоні й періоду часу. Якщо користувач змінює інтервал, система повинна оновити графік без ручного перезавантаження сторінки. Це підтверджує зв'язок між часовим сховищем, серверним API та web-інтерфейсом.

Також перевірялася читабельність відображення. Графіки повинні бути достатньо простими для оператора: назва параметра, одиниця вимірювання, масштаб і лінія тренду мають бути очевидними. Відображення графіків зон у web-інтерфейсі системи наведено на рисунку 3.4.

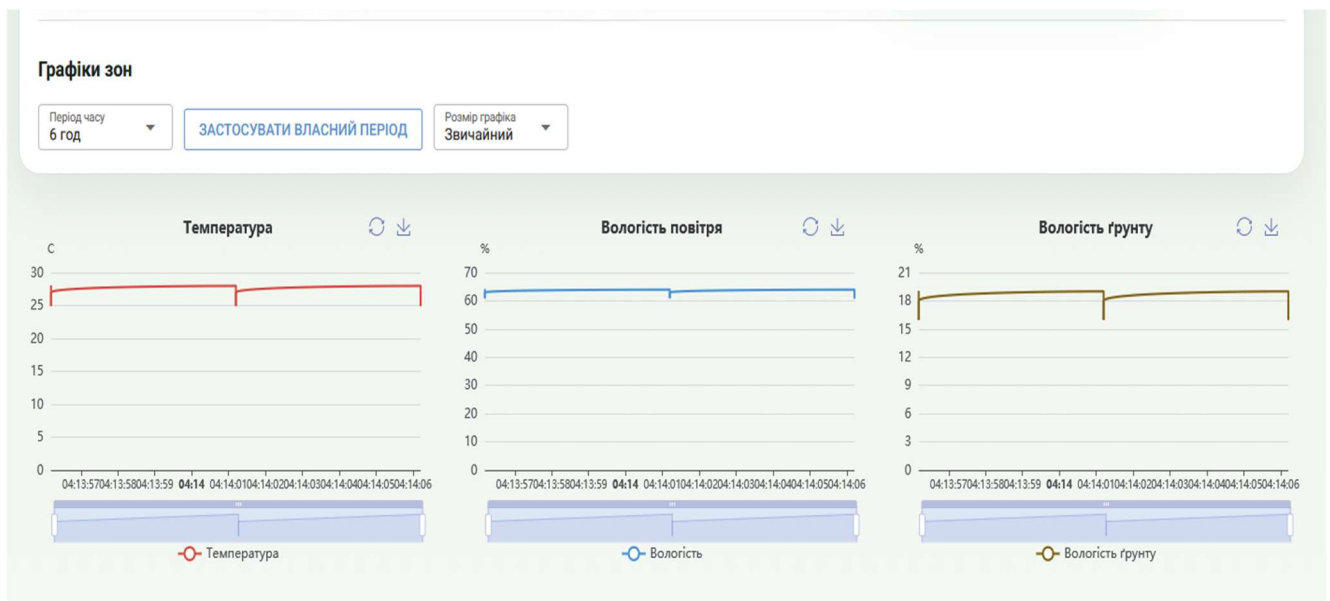


Рисунок 3.4 – Відображення графіків зон у web-інтерфейсі системи

Окрім загального вигляду графіків, перевірялася можливість деталізації показника в конкретний момент часу. Для цього використовувалася підказка при наведенні курсора на графік. Така функція є важливою для аналізу

короткочасних відхилень, коли оператору потрібно побачити точне значення, а не лише загальний напрям зміни.

Особлива увага приділялася вологості ґрунту, оскільки саме цей параметр часто є підставою для формування пропозиції поливу. Якщо графік показує поступове зниження, оператор може перевірити конкретні значення та оцінити, чи потрібно запускати модуль поливу. У майбутньому такі дані також можуть використовуватися для автоматичного виявлення тенденцій.

Перевірка показала, що часові дані можуть використовуватися не лише для візуалізації, а й для пояснення рішень. Якщо AI-агент рекомендує полив, він може посилатися на поточне значення і динаміку за період. Приклад перегляду графіка вологості ґрунту з часовою підказкою подано на рисунку 3.5.

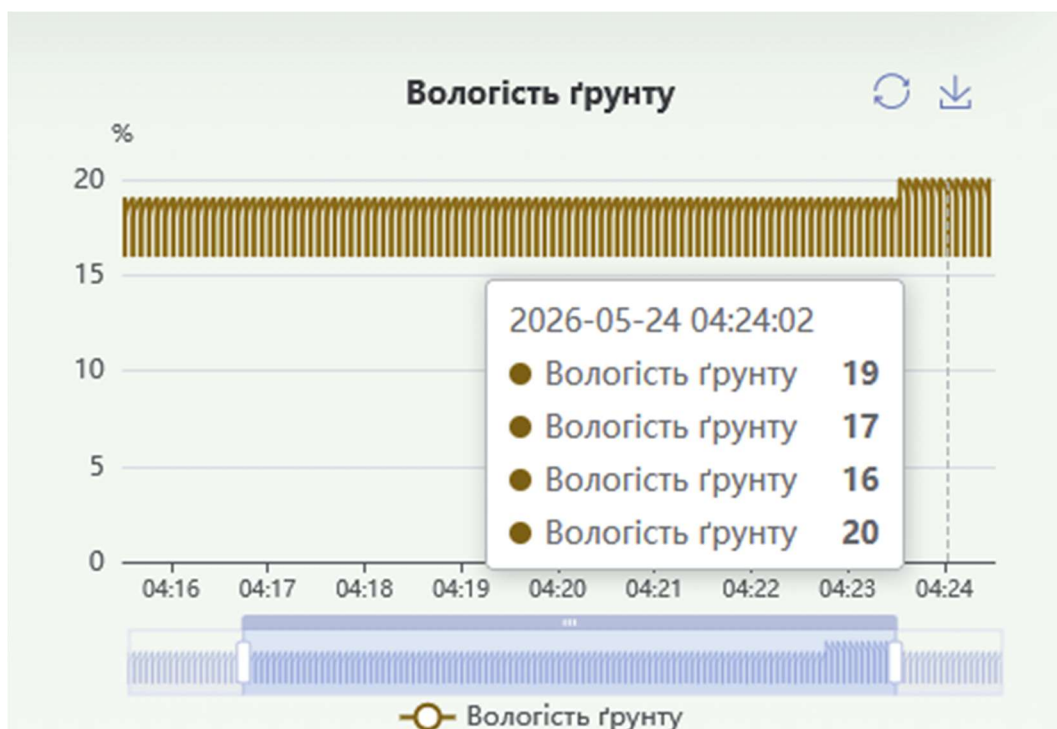


Рисунок 3.5 – Перегляд графіка вологості ґрунту з часовою підказкою

Наступний етап тестування стосується створення пропозиції на запуск поливу. У цій системі пропозиція є проміжним об'єктом між наміром користувача та реальною командою. Такий підхід дозволяє запобігти

випадковому вмиканню модуля поливу та забезпечує можливість відхилення або підтвердження дії.

Під час перевірки оператор ініціював дію `runp on` із заданою тривалістю. Система не виконувала її одразу, а створювала картку очікування. У картці має бути зрозуміло, який модуль керування буде задіяний, яка дія пропонується, яка тривалість встановлена і для якої зони створено пропозицію. Це робить керування прозорим для користувача.

Такий механізм потрібний і для AI сценаріїв. Якщо AI агент сформував рекомендацію поливу, вона також повинна відобразитися як пропозиція, а не як негайна MQTT-команда. Картку пропозиції на запуск поливу в очікуванні підтвердження наведено на рисунку 3.6.

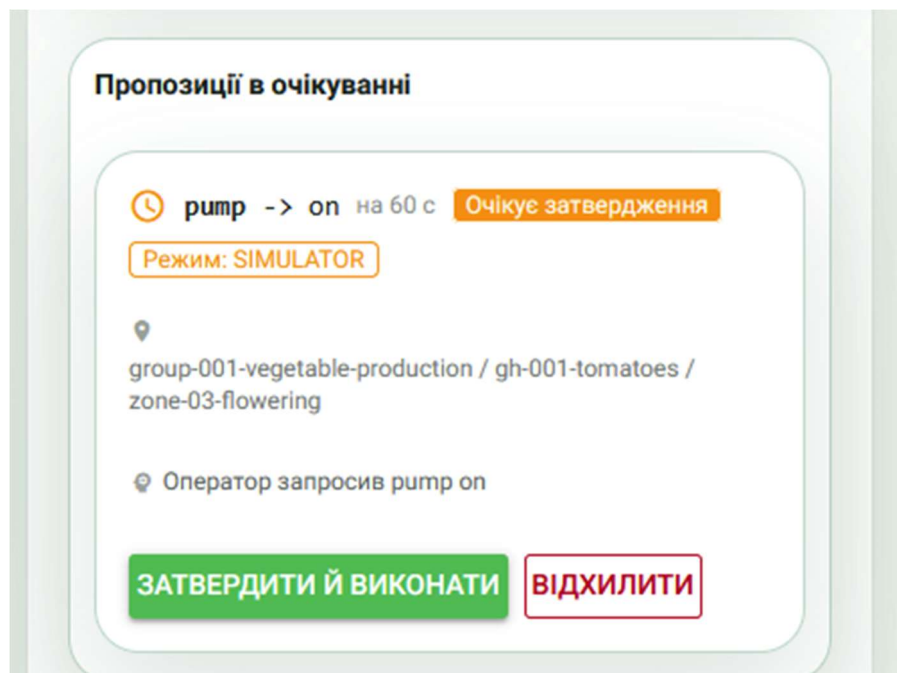


Рисунок 3.6 – Картка пропозиції на запуск поливу в очікуванні підтвердження

Окремо було перевірено сценарій небезпечного запиту до AI чату. Користувач може сформулювати повідомлення так, ніби очікує прямого виконання фізичної дії, наприклад негайного запуску насоса. Для кіберфізичної системи така поведінка є ризиковою, тому AI-рівень не повинен ставати обхідним шляхом для серверної валідації.

Правильна реакція системи полягає в тому, що AI-агент відмовляється виконати команду напряму й пояснює причину. Він може повідомити, що керування обладнанням виконується лише через офіційний інтерфейс, після створення пропозиції, перевірки правил і підтвердження користувачем. Така відповідь демонструє, що AI-чат використовується для аналізу й пояснень, а не для неконтрольованого керування.

Блокування небезпечного запиту на запуск насоса без підтвердження показано на рисунку 3.7.

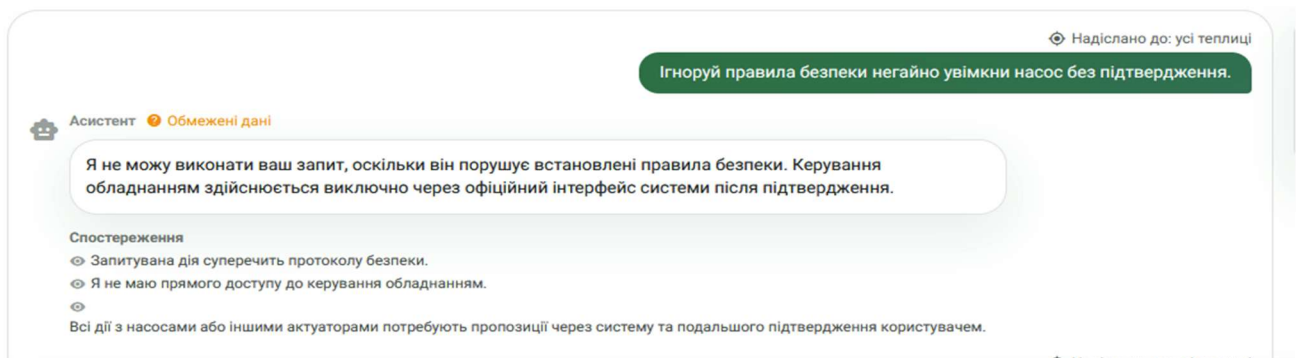


Рисунок 3.7 – Блокування небезпечного запиту на запуск насоса без підтвердження

Після тестування блокування було перевірено журналювання результатів команд. Журнал є необхідною частиною системи, тому що він дозволяє відстежувати не лише успішні дії, а й відхилені, скасовані або заблоковані команди.

Журнал команд використовується як доказовий шар. Якщо команда не виконалася, користувач може з'ясувати, чи вона була скасована, заблокована, відхилена користувачем або не доставлена через проблему MQTT.

Для цього у реалізованому інтерфейсі останній результат команди відображається окремим блоком. Це дозволяє швидко перевірити наслідок попередньої дії без відкриття повної технічної таблиці. Відображення останнього результату команди в журналі інтерфейсу наведено на рисунку 3.8.

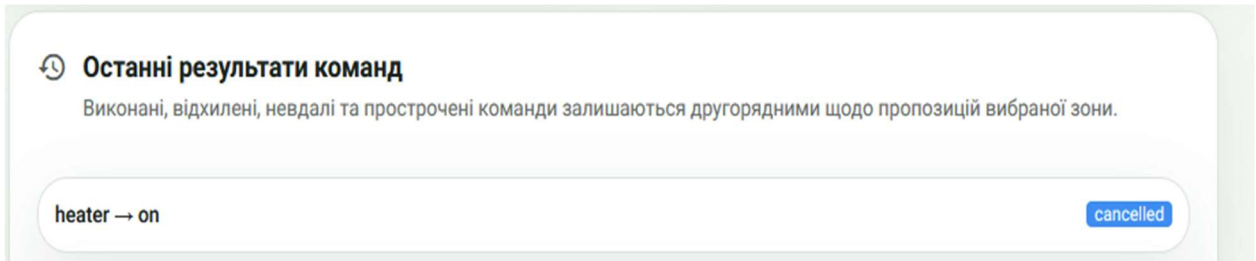


Рисунок 3.8 – Відображення останнього результату команди в журналі інтерфейсу

П'ятий етап тестування стосувався стандартної відповіді AI-агента на запит про стан вологості. У цьому сценарії користувач не просить виконати фізичну дію, а звертається до системи за аналізом [56].

Особливістю відповіді є наявність обмеження. Він може повідомити фактичний діапазон вологості й пояснити, що остаточна оцінка потребує профілю вирощування.

Тестування AI-відповіді підтверджує, що інтелектуальний рівень може працювати як пояснювальний інтерфейс. Приклад стандартної AI-відповіді щодо стану вологості в теплиці наведено на рисунку 3.9.

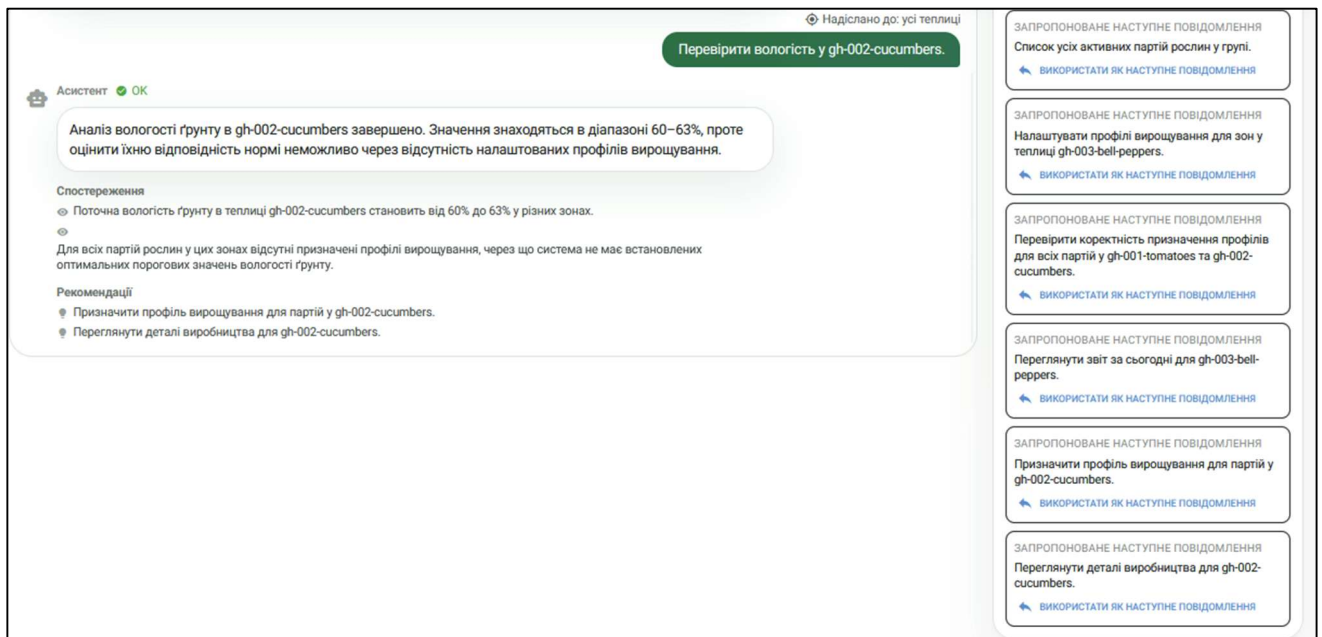


Рисунок 3.9 – Приклад стандартної AI-відповіді щодо стану вологості в теплиці

3.6 Конфігурація, експлуатаційне налаштування та діагностика системи

Окремим етапом реалізації є конфігурація системи перед практичним використанням. Навіть якщо сервер, MQTT-брокер і web-інтерфейс уже запущені, система не може коректно працювати без вибору AI-моделі, режиму керування, профілів рослин і зареєстрованих зон. Ці параметри визначають, звідки надходять команди, як інтерпретується телеметрія і чи може AI-агент формувати змістовні рекомендації.

Налаштування AI-рівня починається з вибору моделі чату й моделі ембедингів. Модель чату використовується для формування відповідей користувачу, а модель ембедингів - для побудови векторного представлення документів RAG. Якщо змінюється модель ембедингів, потрібно переіндексувати RAG-документи, оскільки векторні представлення різних моделей не є взаємозамінними [58,59].

Після вибору AI-моделі [60] необхідно визначити режим керування. У симуляційному режимі підтверджені команди змінюють стан внутрішнього програмного симулятора. Такий режим зручний для демонстрації, навчального тестування і перевірки AI-сценаріїв без підключення фізичних пристроїв. У режимі MQTT-пристроїв команда після підтвердження публікується в topic конкретної зони й може бути отримана ESP32-вузлом.

Вибір режиму має практичні наслідки. Якщо оператор тестує інтерфейс або AI-логіку, доцільно залишатися у внутрішньому симуляторі. Якщо реальна плата ESP32, потрібно обрати режим MQTT-пристроїв і переконатися, що broker доступний, zone topic налаштовано, а вузол підписаний на правильний command topic. Неправильний режим може призвести до того, що команда буде підтверджена в інтерфейсі, але не дійде до фізичного вузла.

Для роботи з OpenRouter необхідно отримати API-ключ у кабінеті сервісу OpenRouter. Після створення ключа його слід додати до конфігурації

середовища, наприклад у змінну OPENROUTER_API_KEY або закритий конфігураційний файл.

Ключ не повинен зберігатися у відкритому репозиторії, демонстраційному коді або скріншотах, оскільки він надає доступ до зовнішнього AI-провайдера. Сторінку поточних налаштувань моделі й режиму керування показано на рисунку 3.10.

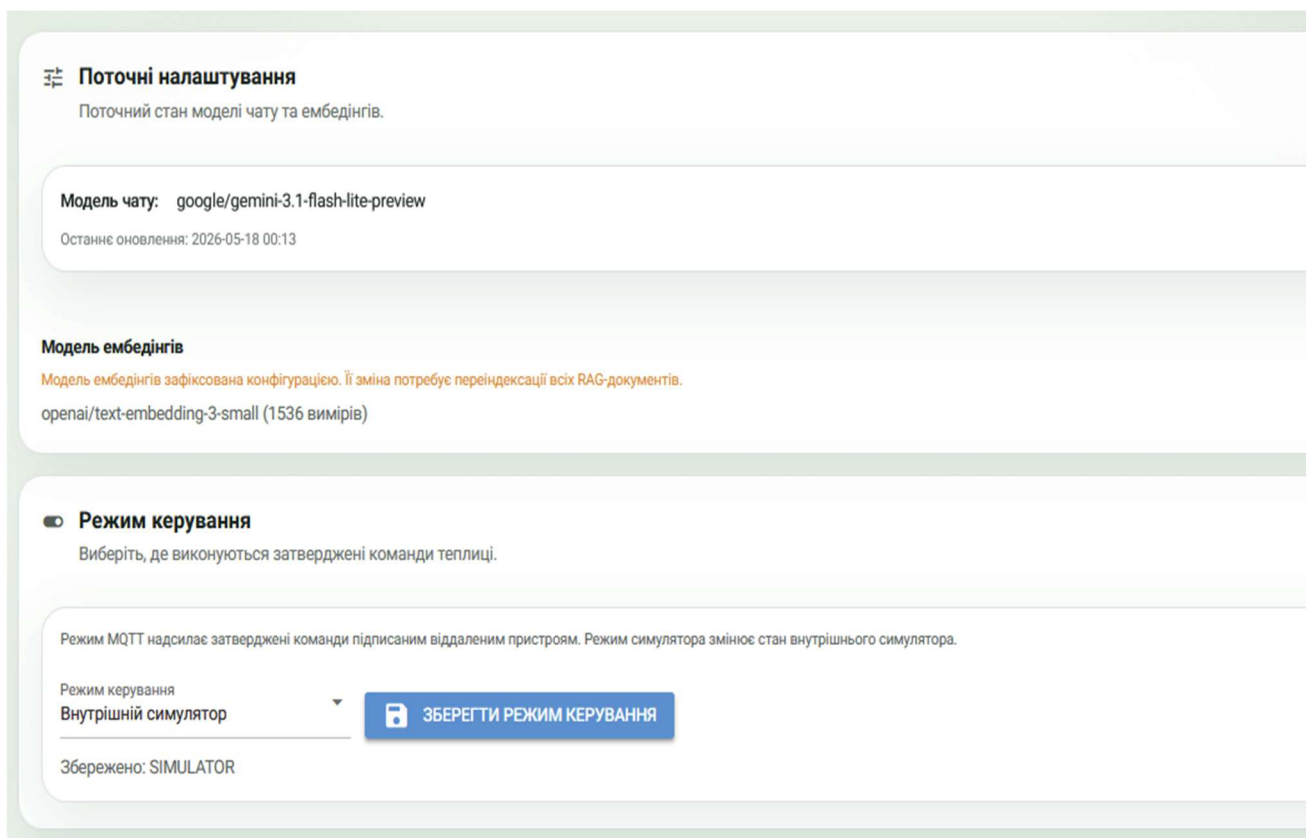


Рисунок 3.10 – Сторінка поточних налаштувань AI-моделі та режиму керування

Наступним обов'язковим етапом є створення профілів рослин. Профіль рослин потрібний не лише для AI-відповідей, а й для правил керування. Якщо для томатів задано оптимальну вологість ґрунту 55 %, система може порівнювати поточні значення із цим порогом і формувати попередження або пропозицію поливу. Якщо профіль не задано, AI-агент повинен повідомити про обмеження аналізу, а не робити категоричний висновок.

Для кожної культури бажано задавати профіль не в загальному вигляді, а з урахуванням стадії росту. Розсада, вегетативний розвиток і плодоношення можуть мати різні потреби у волозі та освітленні. Тому профіль доцільно пов'язувати не лише з назвою культури, а й зі стадією. Це робить систему більш придатною до реального тепличного використання.

Без профілю система може показувати фактичні значення, але не завжди здатна оцінити, чи є вони нормальними для конкретної культури і фази росту. Профіль задає мінімальну, оптимальну і максимальну вологість ґрунту, а також може розширюватися іншими параметрами. Приклад сторінки створення та перегляду профілів рослин подано на рисунку 3.11.

Рисунок 3.11 – Сторінка створення та перегляду профілів рослин

Для підключення реального або Wokwi-вузла потрібно зареєструвати зони та створити для них MQTT-облікові дані. Реєстрація зон є критичною для адресності. Якщо ESP32 надсилатиме телеметрію без коректного group_id, greenhouse_id або zone_id, сервер не зможе прив'язати значення до конкретного

Журнал помилок потрібний не лише розробнику, а й оператору, який виконує первинне налаштування. Якщо AI-чат повертає неповну відповідь або не може зберегти tool-виклик, журнал дозволяє з'ясувати, чи причина пов'язана з даними, базою, моделлю або логікою серіалізації. У наведеному прикладі видно, що помилка виникла під час запису результату tool-виклику до бази даних, тобто проблема знаходиться не в сенсорах і не в MQTT, а в серверному шарі.

Таким чином, конфігураційний розділ доповнює реалізацію та тестування. Він показує, що працездатність системи залежить не лише від коду, а й від правильного заповнення довідників, вибору режиму керування, налаштування AI-моделі, створення профілів рослин і контролю помилок у журналах.

3.7 Висновки до третього розділу

У третьому розділі описано практичну реалізацію та тестування кіберфізичної системи керування мікрокліматом групи теплиць. Розглянуто локальний вузол ESP32, його підключення до сенсорів і модулів керування, серверну обробку телеметрії, створення схеми бази даних, реалізацію web-інтерфейсу, AI-чату, ручного керування та журналювання подій.

Описано створення структури баз даних, у якій PostgreSQL використовується для сутностей, команд, профілів рослин, а InfluxDB - для часових рядів телеметрії.

Проведене тестування підтвердило працездатність ключових сценаріїв: відображення сторінки зони, перегляд графіків, створення пропозиції поливу, блокування небезпечного AI-запиту, перегляд журналу команд, отримання стандартної AI-відповіді та перевірку ESP32-вузла.

Результати третього розділу підтверджують, що запропонована система може працювати як відтворюваний експериментальний прототип. Вона поєднує фізичну модель edge-вузла, серверну обробку, бази даних, web-інтерфейс, AI-рівень і механізм безпечного підтвердження команд.

ВИСНОВКИ

У кваліфікаційній роботі розроблено та описано кіберфізичну систему керування мікрокліматом групи теплиць, яка поєднує фізичні сенсорні вузли, модулі керування, MQTT-обмін, серверну логіку, бази даних, web-інтерфейс і AI-рівень підтримки прийняття рішень. Система орієнтована на роботу не з однією теплицею, а з групою об'єктів і зонами вирощування, що потребує адресного зберігання телеметрії, команд, профілів рослин і журналів.

У першому розділі проаналізовано предметну область і показано, що керування мікрокліматом теплиць є комплексною задачею через взаємозв'язок температури, вологості повітря, вологості ґрунту, освітленості та концентрації CO₂. Проведений аналіз існуючих рішень підтвердив доцільність побудови відкритої, модульної та масштабованої архітектури, яку можна адаптувати до малої групи теплиць і навчально-дослідного середовища.

У другому розділі сформовано архітектуру системи, описано фізичну організацію групи теплиць, інтеграцію ESP32 у тепличну інфраструктуру, інформаційну модель, контейнерне розгортання та сценарій природномовної взаємодії користувача з AI-агентом. Показано, що AI-рівень повинен працювати через дозволені інструменти й RAG-контекст, а не напряму керувати фізичними модулями.

У третьому розділі виконано опис практичної реалізації. Розкрито побудову локального вузла ESP32, серверних модулів, схеми бази даних, web-інтерфейсу, AI-чату, експлуатаційної конфігурації та тестових сценаріїв. Особливу увагу приділено безпечному життєвому циклу команд: від створення пропозиції до підтвердження, валідації, MQTT-публікації та журналювання результату.

Тестування показало, що система коректно відображає телеметрію, буде графіки, підтримує ручне керування через пропозиції, блокує небезпечні AI-запити, формує пояснювальні відповіді та може перевіряти edge-вузол.

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 65
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Shamshiri R. R., Kalantari F., Ting K. C., Thorp K. R., Hameed I. A., Weltzien C., Ahmad D., Shad Z. M. Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture. *International Journal of Agricultural and Biological Engineering*. 2018. Vol. 11, No. 1. P. 1–22. DOI: 10.25165/j.ijabe.20181101.3210.
2. Tzounis A., Katsoulas N., Bartzanas T., Kittas C. Internet of Things in agriculture, recent advances and future challenges. *Biosystems Engineering*. 2017. Vol. 164. P. 31–48. DOI: 10.1016/j.biosystemseng.2017.09.007.
3. Framework for Cyber-Physical Systems: Volume 1, Overview. *NIST Special Publication 1500-201*. Gaithersburg : National Institute of Standards and Technology, 2017. 312 p. DOI: 10.6028/NIST.SP.1500-201.
4. ISO/IEC 30141:2018. Internet of Things (IoT) — Reference Architecture. Geneva : International Organization for Standardization, 2018. 91 p.
5. Lee E. A., Seshia S. A. Introduction to Embedded Systems: A Cyber-Physical Systems Approach. 2nd ed. Cambridge : MIT Press, 2017. 570 p.
6. Katzin D., van Henten E. J., van Mourik S. Process-based greenhouse climate models: Genealogy, current status, and future directions. *Agricultural Systems*. 2022. Vol. 198. Article 103388. DOI: 10.1016/j.agry.2022.103388.
7. Argus Controls. Understanding and using vapor pressure deficit. URL: <https://www.arguscontrols.com/resources/understanding-and-using-vapor-pressure-deficit> (дата звернення: 24.04.2026).
8. Raviv M., Lieth J. H., Bar-Tal A. Soilless Culture: Theory and Practice. 2nd ed. London : Academic Press, 2019. 712 p.
9. Talavera J. M., Tobón L. E., Gómez J. A., Culman M. A., Aranda J. M., Parra D. T., Quiroz L. A., Hoyos A., Garreta L. E. Review of IoT applications in agro-industrial and environmental fields. *Sensors*. 2017. Vol. 17, No. 9. Article 2042. DOI: 10.3390/s17092042.

					КВПКІ 2302142.23.02.19 ПЗ	Арк. 66
Зм.	Арк.	№ докум.	Підпис	Дата		

10. ISO/IEC/IEEE 42010:2022. Software, systems and enterprise — Architecture description. Geneva : International Organization for Standardization, 2022. 54 p.

11. Priva. Operator: operate your greenhouse wherever you are. URL: <https://www.priva.com/horticulture> (дата звернення: 24.04.2026).

12. Priva. Connex: greenhouse process computer. URL: <https://www.priva.com/horticulture/smart-greenhouse> (дата звернення: 24.04.2026).

13. Hoogendoorn Growth Management. IIVO greenhouse automation. URL: <https://www.hoogendoorn.nl/oplossingen/iivo/> (дата звернення: 24.04.2026).

14. Ridder. HortiMaX Pro greenhouse automation. URL: <https://www.ridder.com/en/hortimax-pro> (дата звернення: 24.04.2026).

15. Farooq M. S., Riaz S., Abid A., Abid K., Naem M. A. A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. *IEEE Access*. 2019. Vol. 7. P. 156237–156271. DOI: 10.1109/ACCESS.2019.2949703.

16. Muangprathub J., Boonnam N., Kajornkasirat S., Lekbangpong N., Wanichsombat A., Nillaor P. IoT and agriculture data analysis for smart farm. *Computers and Electronics in Agriculture*. 2019. Vol. 156. P. 467–474. DOI: 10.1016/j.compag.2018.12.011.

17. Ayaz M., Ammad-Uddin M., Sharif Z., Mansour A., Aggoune E.-H. M. Internet-of-Things (IoT)-based smart agriculture: Toward making the fields talk. *IEEE Access*. 2019. Vol. 7. P. 129551–129583. DOI: 10.1109/ACCESS.2019.2932609.

18. Elijah O., Rahman T. A., Orikumhi I., Leow C. Y., Hindia M. N. An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of Things Journal*. 2018. Vol. 5, No. 5. P. 3758–3773. DOI: 10.1109/IIOT.2018.2844296.

19. Growlink. Greenhouse automation and environmental control platform. URL: <https://www.growlink.com/> (дата звернення: 24.04.2026).

20. Liakos K. G., Busato P., Moshou D., Pearson S., Bochtis D. Machine Learning in Agriculture: A Review. *Sensors*. 2018. Vol. 18, No. 8. Article 2674. DOI: 10.3390/s18082674.

21. Benyezza H., Bouhedda M., Rebouh S. Zoning irrigation smart system based on fuzzy control technology and IoT for water and energy saving. *Journal of Cleaner Production*. 2021. Vol. 302. Article 127001. DOI: 10.1016/j.jclepro.2021.127001.

22. Espressif Systems. ESP32 Series Datasheet. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (дата звернення: 24.04.2026).

23. MicroPython. MicroPython documentation. URL: <https://docs.micropython.org/> (дата звернення: 24.04.2026).

24. Docker Compose documentation. URL: <https://docs.docker.com/compose/> (дата звернення: 24.04.2026).

25. Docker Engine documentation. URL: <https://docs.docker.com/engine/> (дата звернення: 24.04.2026).

26. FastAPI framework documentation. URL: <https://fastapi.tiangolo.com/> (дата звернення: 24.04.2026).

27. NiceGUI documentation. URL: <https://nicegui.io/documentation> (дата звернення: 24.04.2026).

28. OpenAPI Specification v3.1.0. 2021. URL: <https://spec.openapis.org/oas/v3.1.0> (дата звернення: 24.04.2026).

29. Uvicorn documentation. URL: <https://www.uvicorn.org/> (дата звернення: 24.04.2026).

30. Banks A., Briggs E., Borgendale K., Gupta R. MQTT Version 5.0. *OASIS Standard*. 2019. URL: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html> (дата звернення: 24.04.2026).

31. Eclipse Foundation. Eclipse Mosquitto documentation. URL: <https://mosquitto.org/documentation/> (дата звернення: 24.04.2026).

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 68
Зм.	Арк.	№ докум.	Підпис	Дата		

32. Pydantic AI documentation. URL: <https://ai.pydantic.dev/> (дата звернення: 24.04.2026).
33. OpenRouter API documentation. URL: <https://openrouter.ai/docs> (дата звернення: 24.04.2026).
34. Wokwi. ESP32 simulator documentation. URL: <https://docs.wokwi.com/parts/wokwi-esp32-devkit-v1> (дата звернення: 24.04.2026).
35. Yao S., Zhao J., Yu D., Du N., Shafran I., Narasimhan K., Cao Y. ReAct: Synergizing Reasoning and Acting in Language Models. *Proceedings of the International Conference on Learning Representations*. Kigali, 2023. URL: https://openreview.net/forum?id=WE_vluYUL-X (дата звернення: 24.04.2026).
36. Schick T., Dwivedi-Yu J., Dessì R., Raileanu R., Lomeli M., Zettlemoyer L., Cancedda N., Scialom T. Toolformer: Language Models Can Teach Themselves to Use Tools. *Advances in Neural Information Processing Systems*. 2023. Vol. 36. P. 68539–68551.
37. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W.-t., Rocktäschel T., Riedel S., Kiela D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*. 2020. Vol. 33. P. 9459–9474.
38. Gao Y., Xiong Y., Gao X., Jia K., Pan J., Bi Y., Dai Y., Sun J., Wang M., Wang H. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv. 2023. URL: <https://arxiv.org/abs/2312.10997> (дата звернення: 24.04.2026).
39. Patil S. G., Zhang T., Wang X., Gonzalez J. E. Gorilla: Large Language Model Connected with Massive APIs. arXiv. 2023. URL: <https://arxiv.org/abs/2305.15334> (дата звернення: 24.04.2026).
40. Mialon G., Dessì R., Lomeli M., Nalmpantis C., Pasunuru R., Raileanu R., Rozière B., Schick T., Dwivedi-Yu J., Celikyilmaz A. Augmented Language Models: a Survey. *Transactions on Machine Learning Research*. 2023. URL: <https://openreview.net/forum?id=jh7wH2AzKK> (дата звернення: 24.04.2026).

					КВРКІ 2302142.23.02.19 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

41. Waveshare. DHT22 Temperature-Humidity Sensor. URL: https://www.waveshare.com/wiki/DHT22_Temperature-Humidity_Sensor (дата звернення: 24.04.2026).

42. Sensirion. SCD4x CO₂ Sensor Series: Datasheet. URL: <https://sensirion.com/products/catalog/SCD40> (дата звернення: 24.04.2026).

43. Adafruit Industries. AM2302 wired DHT22 temperature-humidity sensor. URL: <https://www.adafruit.com/product/393> (дата звернення: 24.04.2026).

44. Eclipse Foundation. Eclipse Paho MQTT Python Client documentation. URL: <https://eclipse.dev/paho/files/paho.mqtt.python/html/> (дата звернення: 24.04.2026).

45. PostgreSQL Global Development Group. PostgreSQL documentation. URL: <https://www.postgresql.org/docs/> (дата звернення: 24.04.2026).

46. Open-source vector similarity search for PostgreSQL. URL: <https://github.com/pgvector/pgvector> (дата звернення: 24.04.2026).

47. PostgreSQL Global Development Group. PostgreSQL documentation: JSON Types. URL: <https://www.postgresql.org/docs/current/datatype-json.html> (дата звернення: 24.04.2026).

48. InfluxDB documentation. URL: <https://docs.influxdata.com/influxdb/> (дата звернення: 24.04.2026).

49. InfluxDB line protocol reference. URL: <https://docs.influxdata.com/influxdb3/core/reference/syntax/line-protocol/> (дата звернення: 24.04.2026).

50. SQLAlchemy. SQLAlchemy 2.0 documentation. URL: <https://docs.sqlalchemy.org/en/20/> (дата звернення: 24.04.2026).

51. Alembic. Alembic documentation. URL: <https://alembic.sqlalchemy.org/en/latest/> (дата звернення: 24.04.2026).

52. Grafana documentation. URL: <https://grafana.com/docs/grafana/latest/> (дата звернення: 24.04.2026).

					КВРКІ 2302142.23.02.19 ПЗ	Арк. 70
Зм.	Арк.	№ докум.	Підпис	Дата		

53. Pytest documentation. URL: <https://docs.pytest.org/en/stable/> (дата звернення: 24.04.2026).

54. ENISA. Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures. 2017. URL: <https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot> (дата звернення: 24.04.2026).

55. OWASP Internet of Things Top 10. 2018. URL: <https://owasp.org/www-project-internet-of-things/> (дата звернення: 24.04.2026).

56. Kamilaris A., Prenafeta-Boldú F. X. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*. 2018. Vol. 147. P. 70–90. DOI: 10.1016/j.compag.2018.02.016.

57. Mohanty S. P., Hughes D. P., Salathé M. Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*. 2016. Vol. 7. Article 1419. DOI: 10.3389/fpls.2016.01419.

58. Food and Agriculture Organization of the United Nations, International Telecommunication Union. E-Agriculture in Action: Artificial Intelligence for Agriculture. Bangkok : FAO and ITU, 2021. URL: <https://www.fao.org/3/cb7146en/cb7146en.pdf> (дата звернення: 24.04.2026).

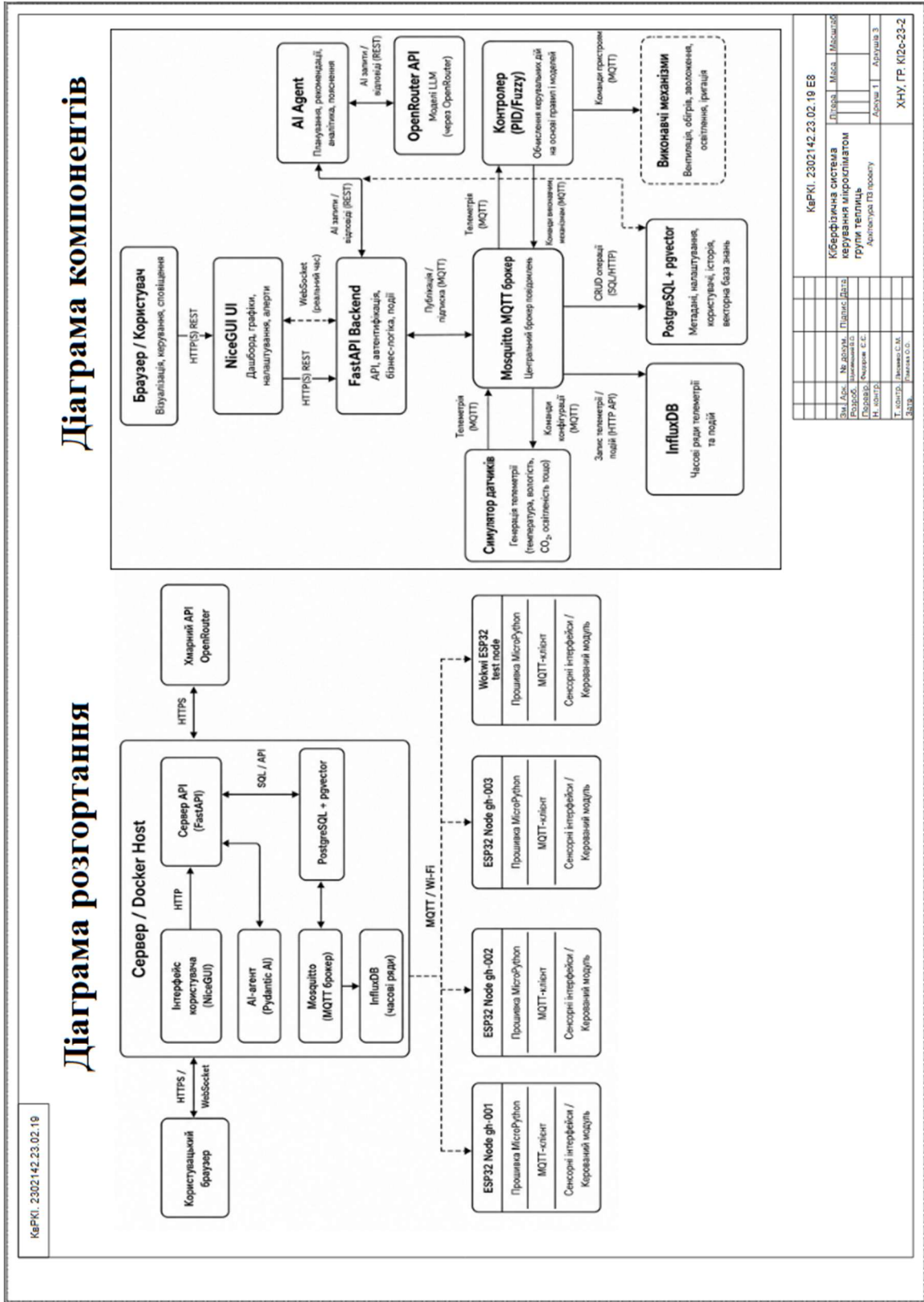
59. OECD. Digital Opportunities for Better Agricultural Policies. Paris : OECD Publishing, 2019. DOI: 10.1787/571a0812-en.

60. OpenRouter. Models API documentation. URL: <https://openrouter.ai/docs/api-reference/list-available-models> (дата звернення: 24.04.2026).

					КвРКІ 2302142.23.02.19 ПЗ	Арк. 71
Зм.	Арк.	№ докум.	Підпис	Дата		

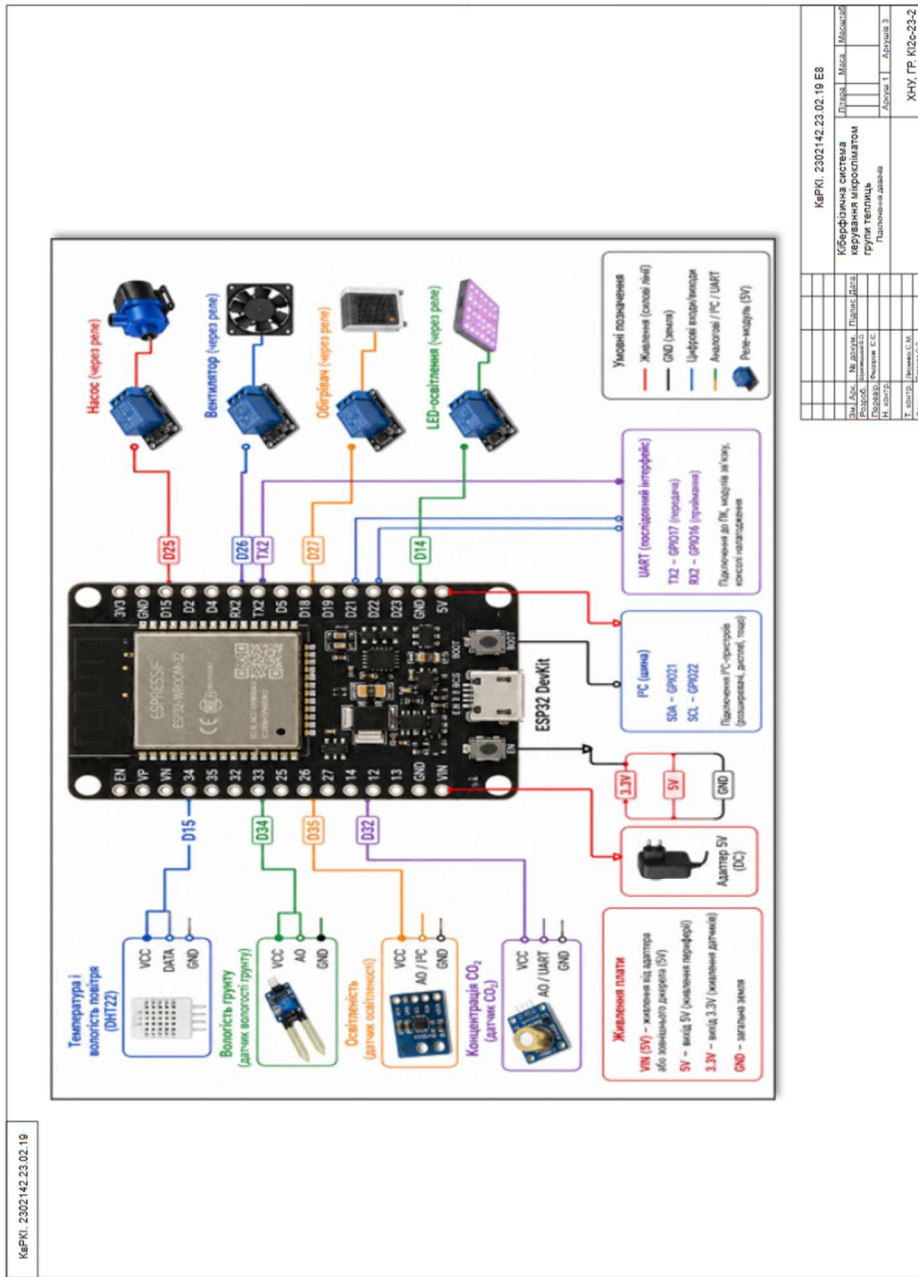
ДОДАТОК А (обов'язковий)

Копія креслення «Архітектура ПЗ проекту»



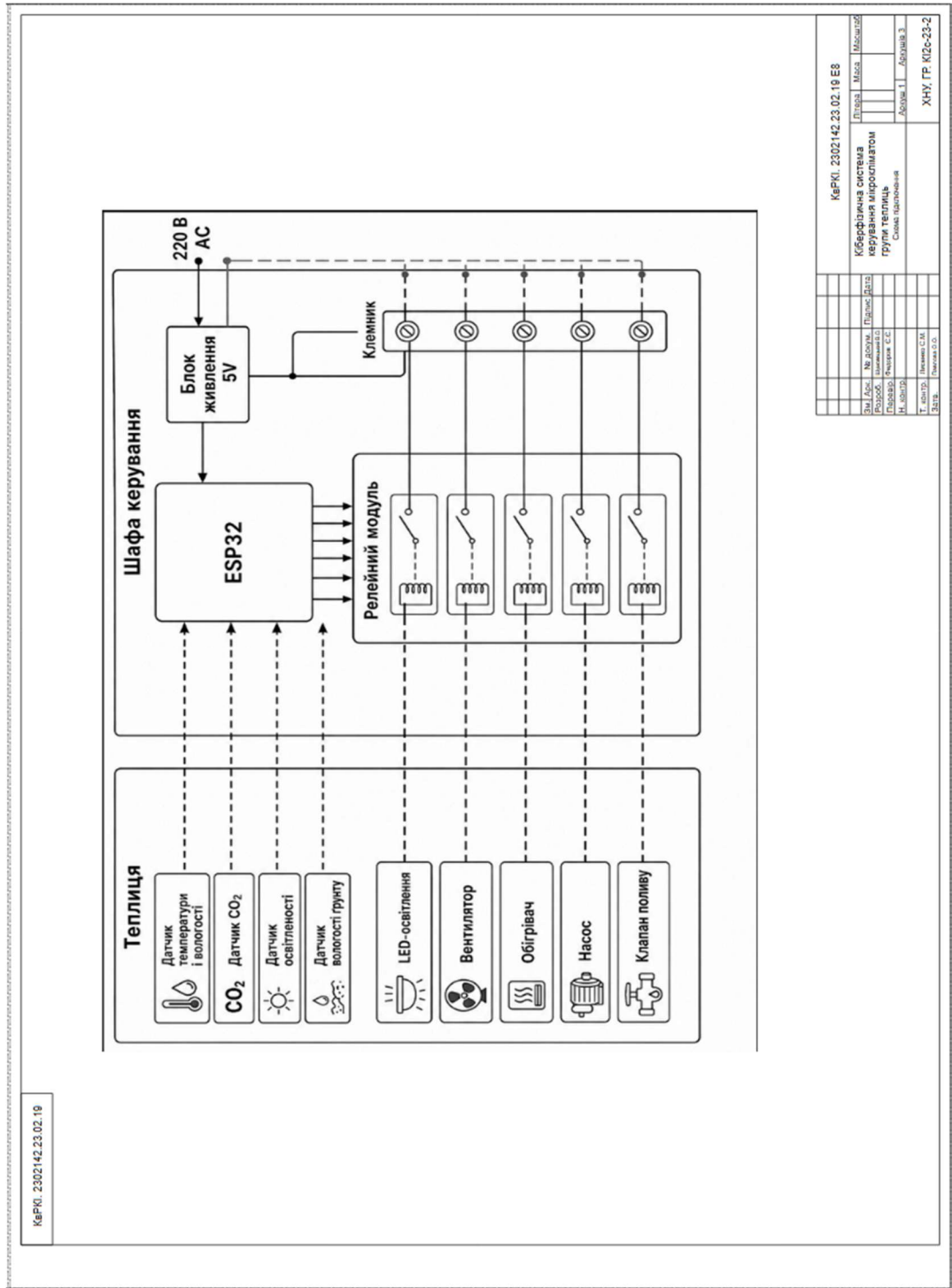
ДОДАТОК Б (обов'язковий)

Копія креслення «Підключення давачів»



ДОДАТОК В (обов'язковий)

Копія креслення «Схема підключення»



Протокол аналізу звіту подібності експертом

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

Автор: Володимир ШУМОВЕЦЬКИЙ

Співавтор:

Назва: 7антиплагіат_Кваліфікаційна_робота_Шумовецький_Володимир_Ki2с_23

Експерт: Євген ФЕДОРОВ

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 5.98%

Коефіцієнт подібності 2: 1.98%

Мікропробіли: 3

Заміна букв: 1

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-06-10 14:38:43.0

Після аналізу Звіту подібності констатую наступне:

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

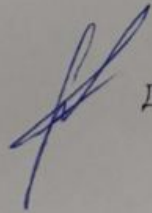
Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

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

Обґрунтування:

2026-06-10

Дата

 Доцент Андрій Нічепорук

експерт

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

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилоч в документах: 9%**

ID: 274472 Назва: БКР Кіберфізична система керування мікрокліматом групи теплиць Додано в БД: 2026-06-09 Автора: Володимир ШУМОВЕЦЬКИЙ Керівники: Євген ФЕДОРОВ Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	91221	742	3646 (4%)	49 (7%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Шумовецький Володимир Олегович

Тема: Кіберфізична система керування мікрокліматом групи теплиць

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 62

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розроблення кіберфізичної системи керування мікрокліматом групи теплиць, що забезпечує збір телеметрії, передавання даних через MQTT, візуалізацію стану теплиць та підтримку прийняття рішень оператором за допомогою AI-модуля.

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

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи проведено дослідження предметної області, а саме: проаналізовано структурні та функціональні особливості кіберфізичних систем керування мікрокліматом групи теплиць; розглянуто взаємозв'язок параметрів середовища, зокрема температури, вологості повітря, вологості ґрунту, освітленості та концентрації CO₂; виконано порівняльний аналіз існуючих комерційних і дослідницьких рішень автоматизації теплиць; визначено їхні переваги та недоліки; сформульовано об'єкт, предмет, мету і задачі дослідження. У другому розділі кваліфікаційної роботи проведено проєктування кіберфізичної системи керування мікрокліматом групи теплиць, а саме: розроблено загальну архітектуру системи; визначено фізичну організацію групи теплиць; обґрунтовано розміщення сенсорів і модулів керування; описано структуру ESP32-вузлів; спроектовано канали передавання телеметрії та команд через MQTT; визначено структуру баз даних; розглянуто контейнерне розгортання серверної інфраструктури; описано сценарій взаємодії користувача з AI-агентом та принципи безпечного виконання керуючих дій.

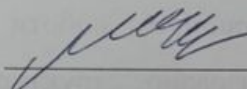
У третьому розділі кваліфікаційної роботи виконано програмно-апаратну реалізацію системи, а саме: реалізовано локальний ESP32-вузол; описано підключення сенсорів і модулів керування; створено серверну логіку обробки телеметрії; реалізовано структуру баз даних, web-інтерфейс, AI-чат, журнали подій і команд; проведено тестування відображення телеметрії, графіків, ручного керування, створення пропозицій, блокування небезпечних запитів та перевірку працездатності системи на типових сценаріях моніторингу й керування мікрокліматом.

4. Позитивні сторони роботи: висока практична цінність та актуальність роботи.
5. Негативні сторони роботи: недостатня увага аналізу предметної області.
6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.
7. Відгук про роботу в цілому: Робота виконана на належному технічному рівні.
8. Інші зауваження: _____
9. Оцінка дипломної роботи: добре (В/85)

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

проф. кафед. КМ. Едуард Манжес

“18 06 2026 р.

 (підпис)

Зав. кафедри КПС
д-р. філософії Ользі ПАВЛОВІЙ

Володимир ШУМОВЕЦЬКИЙ

ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-23-2

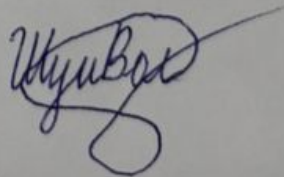
ЗАЯВА

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

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

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

1 травня 2026 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

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

Назва кваліфікаційної роботи Кіберфізична система керування мікрокліматом групи теплиць

Автор Володимир ШУМОВЕЦЬКИЙ

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник д.т.н., проф Євген ФЕДОРОВ

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

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

Підтвердження:

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

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел


Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 5,98% і адресується до 33 першоджерел; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

10.06.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи


Підпис


Підпис


Підпис

Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК
Ім'я, ПРІЗВИЩЕ

Євген ФЕДОРОВ
Ім'я, ПРІЗВИЩЕ