Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технології, автоматизації та робототехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розроблення програмного забезпечення для автоматизованого керування освітленням та кліматом у виробничих приміщеннях
(тема)

Виконав:
здобувач 4 року навчання,
групи АКТАКІТ-21-1

Кирило МОЛОКАНОВ
(власне ім'я, прізвище)

Спеціальність 151 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Автоматизація та комп'ютерно-інтегровані технології
(повна назва освітньої програми)

Керівник асистент Ганна САМОЙЛЕНКО
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри КІТАР _____ Ігор НЕВЛЮДОВ
(підпис) (власне ім'я, прізвище)

2025 р.

Я, Молоканов Кирило Андрійович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на віповідне джерело.

"16" червня 2025 р. Кирило МОЛОКАНОВ

(підпис)

Харківський національний університет радіоелектроніки

Факультет_____Автоматики і комп'ютеризованих технологій_____

Кафедра__Комп'ютерно інтегрованих технологій, автоматизації та робототехніки_

Рівень вищої освіти_____перший (бакалаврський)_____

Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології____
(код і повна назва)

Тип програми _____освітньо-професійна_____

Освітня програма____Автоматизація та комп'ютерно-інтегровані технології__
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«30» квітня 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві_____Молоканову Кирилу Андрійовичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи _Розроблення програмного забезпечення для автоматизованого керування освітленням та кліматом у виробничих приміщеннях
затверджена наказом університету від ____19.05_____2025 р. № ___390 Ст___
2. Термін подання здобувачем роботи до екзаменаційної комісії __26.05_ 2025 р.
3. Вихідні дані до роботи Програмне середовище PyCharm для розробки програмного забезпечення на мові Python; симуляційне середовище Wokwi для моделювання підключення вузлів ESP32; специфікації сенсорів, димера та реле; протоколи взаємодії між компонентами; мессенджер-середовище для реалізації інтерфейсу взаємодії з користувачем;
_____
_____
_____
_____

4. Перелік питань, що потрібно опрацювати в роботі Аналіз сучасного стану автоматизованих систем керування мікрокліматом та освітленням, порівняння апаратних і програмних засобів, розробка структури системи з використанням ESP32, розроблення програмного забезпечення на основі FastAPI та Telegram-бота, створення симуляторів вузлів у середовищі Python, проєктування логіки автоматичного й ручного керування, реалізація візуалізації даних та збереження в SQLite, моделювання динаміки підсистеми освітлення в MATLAB_____
_____
_____
_____
_____
_____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) __Слайди у форматі PowerPoint у кількості 12 слайдів з розширенням .pptx_____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз технічного завдання	30.04.2025	Виконано
2	Аналіз літератури за темою роботи	04.05.2025	Виконано
3	Виконання розділу 1 Аналіз сучасного стану автоматизованого керування освітленням та кліматом на виробництвах	12.05.2025	Виконано
4	Виконання розділу 2 Розробка структури макету автоматизованої системи віддаленого керування освітленням та кліматом у виробничих приміщення	14.05.2025	Виконано
5	Виконання розділу 3 Розрахунки та моделювання динаміки системи	15.05.2025	Виконано
6	Оформлення пояснювальної записки	15.06.2025	Виконано
7	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism	17.06.2025	Виконано
8	Подання роботи на рецензію	18.06.2025	Виконано
9	Подання роботи на підпис зав.кафедри	20.06.2025	Виконано
10	Подання атестаційної роботи до ЕК	26.06.2025	Виконано

Дата видачі завдання __30__ ___квітня_____2025 р.

Здобувач _____ _____Кирило МОЛОКАНОВ_____
(підпис)

Керівник роботи _____ ___ас. Ганна САМОЙЛЕНКО_
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 85 с., 12 табл., 51 рис., 40 джерел.

АВТОМАТИЗАЦІЯ, МІКРОКЛІМАТ, ОСВІТЛЕННЯ, ESP32, FASTAPI, ЧАТ-БОТ, СЕНСОРИ, ДИМЕР, РЕЛЕ, IoT.

Об’єкт роботи – процес автоматизованого керування мікрокліматом та освітленням у виробничих приміщеннях, що передбачає збір даних з датчиків, обробку інформації та вплив на виконавчі пристрої з можливістю ручного втручання користувача.

Мета роботи – покращення умов виробничого середовища шляхом розробки програмного забезпечення, що дозволяє автоматизовано підтримувати належні параметри середовища з можливістю ручного керування.

Предмет роботи – методи та програмні засоби автоматичного та дистанційного керування виконавчими пристроями на основі сенсорних даних.

У першому розділі проаналізовано сучасні підходи до автоматизації виробничого середовища, розглянуто вимоги до параметрів комфорту та енергоефективності, обґрунтовано доцільність впровадження локальних автономних систем на базі відкритих технологій.

У другому розділі розроблено структуру макету системи, реалізовано симулятори вузлів ESP32 у середовищі Wokwi, серверну частину FastAPI, месседжер-бот, базу даних SQLite та механізми автоматичного й ручного керування. Описано логіку обробки сенсорних даних, override-команд, перевірку зв’язку та fallback-інтерфейс.

У третьому розділі проведено моделювання підсистеми освітлення в середовищі MATLAB, побудовано динамічну модель, розраховано перехідні

характеристики, сталу складову та перевірено стійкість системи за критерієм Найквіста.

У результаті розроблено симульовану систему, що може бути адаптована до фізичного обладнання, забезпечує автоматизоване керування параметрами мікроклімату й освітлення з можливістю віддаленого доступу та відповідає сучасним вимогам до виробничих IT-рішень.

ABSTRACT

Explanatory note: 85 pages, 12 table, 51 figures, 40 references.

AUTOMATION, MICROCLIMATE, LIGHTING, ESP32, FASTAPI, CHAT BOT, SENSORS, DIMMER, RELAY, IoT.

The object of development is the process of automated microclimate and lighting control in production facilities, which involves collecting data from sensors, processing information and influencing executive devices with the possibility of manual user intervention..

The purpose of the development is to increase the efficiency of the production process by developing software that allows automated maintenance of appropriate environmental parameters with the possibility of manual control.

The subject of development is methods and software tools for automatic and remote control of executive devices based on sensor data.

The first chapter analyses modern approaches to the automation of the production environment, considers the requirements for comfort and energy efficiency parameters, and justifies the feasibility of implementing local autonomous systems based on open technologies.

The second chapter develops the structure of the system layout, implements ESP32 node simulators in the Wokwi environment, the FastAPI server part, a messenger bot, an SQLite database, and automatic and manual control mechanisms. The logic of sensor data processing, override commands, connection verification, and fallback interface are described.

In the third section, the lighting subsystem is modelled in the MATLAB environment, a dynamic model is built, transient characteristics and steady-state

components are calculated, and the stability of the system is verified according to the Nyquist criterion.

As a result, a simulated system has been developed that can be adapted to physical equipment, provides automated control of microclimate and lighting parameters with the possibility of remote access, and meets modern requirements for industrial IT solutions.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

КІТАР – комп’ютерно-інтегровані технологій, автоматизації та робототехніки;

СУБД – система управління базами даних;

ТАВР – технології та автоматизація виробництва радіоелектронних засобів;

API (Application Programming Interface) – інтерфейс прикладного програмування;

BT (Bluetooth) – бездротова технологія короткого радіусу дії;

HMI (Human-Machine Interface) – інтерфейс «людина – машина»;

HTTP (HyperText Transfer Protocol) – протокол передачі гіпертексту;

IoT (Internet of Things) – інтернет речей;

JSON (JavaScript Object Notation) – текстовий формат обміну даними;

MQTT (Message Queuing Telemetry Transport) – протокол передачі повідомлень;

MP – MicroPython;

PLC (Programmable Logic Controller) – програмований логічний контролер;

PWM (Pulse Width Modulation) – широтно-імпульсна модуляція;

SCADA (Supervisory Control and Data Acquisition) – система диспетчерського управління та збору даних;

UART (Universal Asynchronous Receiver-Transmitter) – асинхронний послідовний інтерфейс.

ВСТУП

Ефективність виробничих процесів значною мірою залежить від стабільності параметрів мікроклімату та рівня освітлення в приміщеннях. Ці фактори впливають не лише на якість продукції, але й на енерговитрати, працездатність персоналу, відповідність нормативам та безпеку. Попри наявність сучасного обладнання, багато підприємств досі використовують ручне або частково автоматизоване управління системами вентиляції, освітлення та кондиціонування, що не дозволяє оперативно реагувати на зміну умов.

В умовах підвищених вимог до гнучкості та енергоефективності виробництва доцільним є впровадження автоматизованої системи, яка здійснює моніторинг параметрів середовища, приймає рішення щодо вмикання або регулювання виконавчих пристроїв (вентиляторів, димерів, реле тощо) та надає можливість віддаленого контролю.

Об'єктом дослідження виступає система автоматизованого керування мікрокліматом і освітленням у виробничих приміщеннях.

Предмет роботи – методи та програмні засоби автоматичного та дистанційного керування виконавчими пристроями (освітлення, вентиляція, кондиціонування) на основі сенсорних даних.

Метою роботи є покращення умов виробничого середовища шляхом розробки програмного забезпечення, що дозволяє автоматизовано підтримувати належні параметри середовища з можливістю ручного керування.

Система має забезпечити автоматичну реакцію на зміну параметрів (температура, вологість, CO_2, освітленість) на основі даних, отриманих від сенсорів, і приймати керуючі рішення щодо підтримання цих параметрів у допустимих межах.

Для досягнення мети застосовано такі методи та інструменти:

– аналіз аналогів і вибір мікроконтролера ESP32 як базової платформи;

– моделювання апаратної частини у віртуальному середовищі WokWi;

– реалізація серверної частини з використанням FastAPI (Python);

– збереження даних у локальній базі SQLite;

– розробка мессенжджер-бота як інтерфейсу користувача;

– тестування логіки керування у симуляційному середовищі.

Сформоване рішення демонструє підхід до проєктування локальних автоматизованих систем керування, які можуть масштабуватись і адаптуватись до потреб малого та середнього виробництва. Комбінація автономного реагування та можливості ручного контролю забезпечує сучасний рівень гнучкості, надійності та зручності використання.

Пояснювальну записку оформлено згідно ДСТУ 3008:2015 [1], а також з методичними вказівками з підготовки й оформлення кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти спеціальності 151 Автоматизація та комп’ютерно-інтегровані технології освітньої програми «Автоматизація та комп’ютерно-інтегровані технології» [2].

1 АНАЛІЗ СУЧАСНОГО СТАНУ АВТОМАТИЗОВАНОГО КЕРУВАННЯ ОСВІЛЕННЯМ ТА КЛІМАТОМ НА ВИРОБНИЦТВАХ

1.1 Визначення актуальності, мети, об’єкта та предмета розробки

У сучасних умовах розвитку виробничих технологій та зростання вимог до енергоефективності й безпеки праці автоматизоване управління освітленням і мікрокліматом набуває особливого значення. На багатьох підприємствах досі використовуються застарілі системи з ручним або частковим керуванням, які не адаптуються до змін навантаження, режиму роботи чи умов середовища. Це призводить до перевитрати енергії, зниження комфорту персоналу та підвищеного навантаження на інженерну інфраструктуру.

Доцільним є впровадження автономних інтелектуальних систем, здатних реагувати на зміни параметрів у реальному часі та адаптуватися до виробничих умов. Вони дозволяють оптимізувати енерговитрати, покращити умови праці та підвищити ефективність управління.

Крім того, важливою є підтримка дистанційного доступу, що дає змогу швидко реагувати на нестандартні ситуації без безпосередньої присутності персоналу.

Метою роботи є покращення умов виробничого середовища шляхом розробки програмного забезпечення, що дозволяє автоматизовано підтримувати належні параметри середовища з можливістю ручного керування.

Об’єкт дослідження – процес автоматизованого управління в умовах виробництва з урахуванням технічних і експлуатаційних обмежень. Предметом розробки є програмна система, що реалізує логіку керування світловими й кліматичними параметрами на основі внутрішніх і зовнішніх факторів.

Особливу увагу приділено потенціалу впровадження таких систем у форматі DIY на малих та середніх підприємствах. Використання відкритих апаратних і програмних компонентів – таких як ESP32, FastAPI і MySQL – дає змогу побудувати ефективну систему без значних фінансових витрат. Її архітектура дозволяє підлаштовуватися під різні виробничі потреби: як за кількістю датчиків, так і за масштабом розгортання. Система не залежить від закритого комерційного програмного забезпечення, тому її можна доповнювати, змінювати та обслуговувати навіть без залучення сторонніх рішень [3].

1.2 Огляд технічних та програмних рішень для автоматизованого керування освітленням і мікрокліматом

У сучасних виробничих умовах усе більше підприємств переходять на автоматизовані системи керування мікрокліматом та освітленням. Це не лише дозволяє зменшити енергоспоживання, а й покращує умови праці персоналу та стабілізує технологічні процеси. Завдяки розвитку IoT, широкій доступності мікроконтролерів, різноманіттю сенсорів, хмарних платформ і мобільних інтерфейсів стало можливим створення адаптивних систем, що працюють автономно, орієнтуючись на задані параметри та змінні зовнішнього середовища.

До апаратної основи таких систем зазвичай входять:

– мікроконтролери (ESP32 [4], Arduino Uno [5], Raspberry Pi [6] тощо) – слугують ядром системи, забезпечуючи зчитування даних із датчиків і керування виконавчими пристроями;

– датчики освітленості (наприклад, BH1750), температури та вологості (DHT22, BME280), рівня CO_2 (MQ-135, SCD30) – передають інформацію про стан середовища;

– виконавчі пристрої – світлодіоди, реле, вентилятори, кондиціонери, які вмикаються/вимикаються за командою системи;

– комунікаційні модулі – модулі Wi-Fi, Bluetooth або ZigBee забезпечують зв'язок між системою та користувачем або хмарним сервером.

З боку ПЗ системи керування часто реалізуються за допомогою:

– arduino ide/platformio – основні середовища для програмування контролерів типу Arduino;

– python у pycharm або vscode – для систем на базі raspberry pi або esp32 з micropython;

– mqtt/http – легкі протоколи для обміну даними з інтерфейсами або хмарою;

– бази даних (MySQL, SQLite, Firebase, MongoDB) – бази даних для збереження логів, станів і історії роботи системи;

– графічні або мобільні інтерфейси – чат-боти, веб-додатки, мобільні застосунки для дистанційного керування.

У період з 2020 по 2025 рік в автоматизації освітлення й мікроклімату чітко простежуються кілька ключових тенденцій.

Перший напрям – інтеграція таких систем у загальні енергоменеджмент-платформи. Це не лише зручно, а й стратегічно виправдано: оптимізується витрата енергії, дотримуються політики сталого розвитку, а підприємства знижують експлуатаційні витрати [4].

Другий напрям – впровадження алгоритмів штучного інтелекту. Системи на основі машинного навчання здатні прогнозувати зміну температури, вологості чи рівня освітлення й автоматично адаптувати свої параметри. Замість фіксованих налаштувань реалізується живе реагування.

Третій напрям – розвиток сенсорних мереж із підтримкою самоконфігурації. Це істотно спрощує процес монтажу й подальшого обслуговування: система здатна автоматично виявляти нові датчики, синхронізувати їх та використовувати у логіці керування без ручного втручання.

В умовах активного застосування хмарних сервісів і віддаленого керування зростає кількість кіберзагроз. Саме тому сучасні автоматизовані системи (табл. 1.1) дедалі частіше доповнюються шифруванням, захищеними протоколами передавання даних і багаторівневою автентифікацією доступу.

Таблиця 1.1 – Порівняння популярних рішень

Параметр	Arduino Uno [5]	Arduino Mega [5]	ESP8266 [6]	ESP32 [6]	Raspberry Pi Pico [7]	Raspberry Pi Zero [7]	Raspberry Pi 4 [7]
Мова програмування	C/C++	C/C++	C++, MP	C/C++, MP	C/C++, MP	Python C++	Python C++
Кількість пінів	14 Digital / 6 Analog	54 Digital / 16 Analog	17 GPIO	34 GPIO	26 GPIO	40 GPIO	40 GPIO
Flash-пам’ять	32 КБ	256 КБ	4 МБ	4 МБ+	2 МБ	4-32 ГБ	32 ГБ+
Wi-Fi	Немає	Немає	Є	Є	Немає	Є	Є
Bluetooth	Немає	Немає	Нема	Є	Немає	Є	Є
Переваги	Простий, дешевий, багато бібліотек	Багато пінів	Wi-Fi, компактність, дешевий	Wi-Fi + BT, потужний, багато периферії	Хороша підтримка в MP, дешевий	Компактний, під Linux	Потужний, для складних задач

1.3 Аналіз існуючих рішень керування освітленням та кліматом на промислових підприємствах

Промислові підприємства сьогодні, дедалі частіше зіштовхуються з потребою оптимізувати споживання енергії. І тут одразу виходять на передній план два ключові напрямки – самоконфігурація освітлення та клімат-контроль.

Обидва напрями критично важливі для безпечної, ефективної й енергоощадної роботи виробничого об’єкта.

Рівень автоматизації систем може бути дуже різним – від класичних релейних схем до складних IoT-рішень із централізованим управлінням і аналітикою.

На деяких підприємствах досі застосовуються релейні щитки (рис. 1.1). Це прості системи з ручним або автоматичним керуванням освітленням і вентиляторами. Вони були поширеними в минулому столітті через простоту та дешевизну, проте в сучасних умовах мають суттєві обмеження.

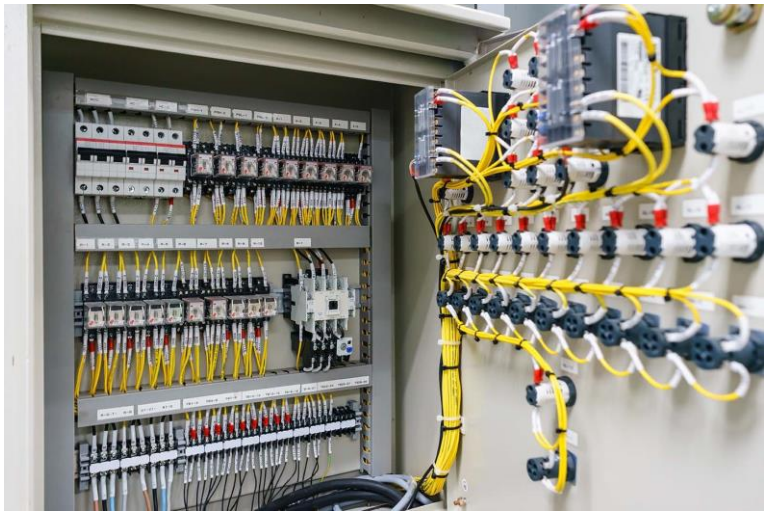


Рисунок 1.1 – Приклад релейного щитка для автоматизованого керування електроспоживанням на виробництві [8]

Їхній головний недолік – відсутність адаптації до змін середовища: такі системи не враховують наявність персоналу, рівень освітленості, температуру повітря, що призводить до перевитрати енергії.

З появою енергоощадних LED-систем і сенсорів руху стали доступними більш розумні рішення. Освітлення може вимикатися автоматично за відсутності людей або змінювати яскравість, якщо в приміщенні є природне освітлення. Це дозволяє знизити споживання без шкоди для функціональності.

Схожі зміни спостерігаються і в кліматичних системах. Опалення чи вентиляція активуються тільки за потреби – наприклад, якщо в приміщенні присутні працівники або температура виходить за допустимі межі. Хоча такі рішення не завжди мають централізоване керування, вони добре працюють на об’єктах середнього масштабу. Приклад подібної системи Digital Lumens наведено на рисунку 1.2.

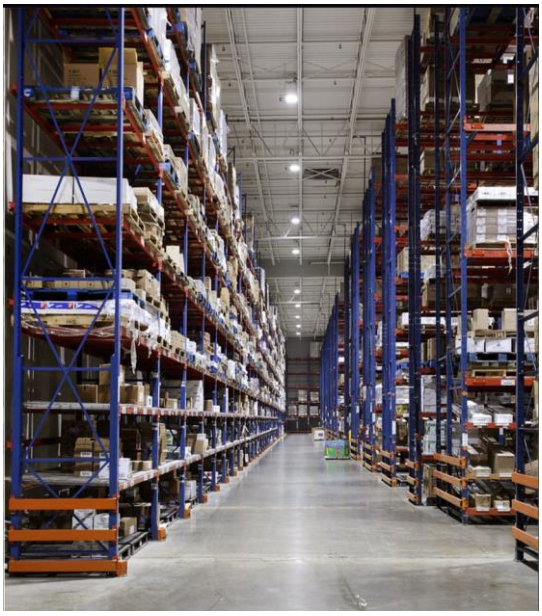


Рисунок 1.2 – Інтелектуальна LED-система освітлення Digital Lumens з вбудованими сенсорами та централізованим керуванням [9]

На сучасному етапі лідери галузі впроваджують інтегровані системи освітлення та клімату, що працюють на базі програмованих логічних контролерів (PLC) або IoT-платформ.

Ці системи здатні не лише автоматизовано вмикати світло чи вентилятори, але й регулювати температуру, керувати вентиляцією, контролювати вологість. Більше того – вони синхронізуються з пожежною сигналізацією, охоронною системою, енергомоніторингом та системами доступу.

Такі рішення підтримують інтеграцію в SCADA або HMI, забезпечують сценарне керування («економний режим уночі», «повне освітлення в зоні

оператора», «посилене охолодження при перегріві») та дозволяють отримати повний контроль у зручному візуальному інтерфейсі. Проте їх вартість є вищою, і потребується участь кваліфікованих фахівців для налаштування і обслуговування.

На рисунку 1.3 представлено простий графічний інтерфейс, реалізований у середовищі SCADA-системи Trace Mode. Основним завданням інтерфейсу є візуалізація та регулювання температурного режиму виробничого середовища.

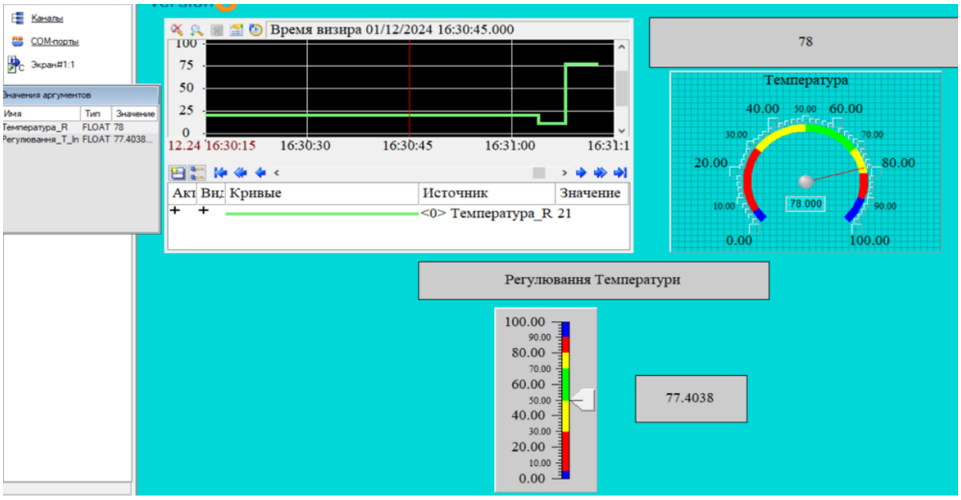


Рисунок 1.3 – Приклад простого SCADA-інтерфейсу у середовищі Trace Mode створений з метою регулювання температури

Попри високу ефективність, системи цього рівня мають вищу вартість та вимагають участі кваліфікованих фахівців для налаштування й обслуговування.

Нині, особливої популярності набувають гнучкі системи на базі мікроконтролерів, що дозволяють реалізувати інтелектуальну логіку з дистанційним керуванням при мінімальних витратах. Вони забезпечують повний контроль – як автоматичний, так і ручний – із локального або мобільного інтерфейсу.

Найбільш перспективним виглядає підхід, що поєднує сучасні технології автоматизації з простотою впровадження та експлуатації.

1.4 Програмні та апаратні засоби, що використовуються в системах автоматизації

Сучасні системи автоматизації функціонують на основі тісної інтеграції апаратного забезпечення та програмного забезпечення, де перше відповідає за збирання та передачу даних, а друге – за їх обробку, аналіз і реалізацію логіки керування. Саме через цей зв’язкок, система здатна зчитувати дані, приймати рішення й керувати технікою. Усе це – з можливістю візуалізації поточного стану.

При проєктуванні таких систем особливо важливо правильно підібрати апаратні й програмні компоненти. Надійність, масштабованість і зручність інтеграції напряму залежать від їхньої взаємодії.

Одним із перевірених індустріальних рішень є програмовані логічні контролери (PLC). Вони забезпечують стабільну роботу навіть в умовах сильних електромагнітних завад, мають високий ступінь точності керування та використовуються на об'єктах, де потрібна безперервність процесів. Проте для невеликих систем таке рішення часто є занадто складним і дорогим.

У малих і середніх проєктах доцільніше використовувати мікроконтролери загального призначення – Arduino, ESP32, Raspberry Pi. Вони мають достатню обчислювальну потужність, прості у програмуванні та мають велику спільноту підтримки.

Одним із доцільних рішень для реалізації такої системи є використання ESP32 для зчитування показників навколишнього середовища та керування виконавчими пристроями. Такий вибір дозволяє реалізувати автономну логіку, просте налаштування та дистанційне керування через інтерфейс віддаленого доступу.

Для моніторингу параметрів застосовуються цифрові й аналогові сенсори, такі як: температура, вологість, освітленість, якість повітря. Вони дозволяють системі динамічно адаптувати свою поведінку, наприклад: зменшити яскравість

ламп при яскравому природному світлі або активувати вентиляцію при перевищенні температури.

З боку ПЗ важливу роль відіграють SCADA-системи (Siemens WinCC, Trace Mode, Ignition тощо), які дозволяють:

– будувати складні сценарії керування;

– забезпечувати візуалізацію через HMI-інтерфейси;

– організовувати моніторинг у реальному часі [10].

Такі рішення вдало підходять для масштабних об’єктів, таких як Siemens WinCC (рис. 1.4)

Рисунок 1.4 – Приклад інтерфейсу SCADA-системи Siemens WinCC [11]

У менш складних системах часто використовуються веб-інтерфейси або графічні середовища, такі як Node-RED, що працює у поєднанні з Raspberry Pi. Це середовище дозволяє створювати функціональні панелі моніторингу та логіку керування, використовуючи протоколи MQTT, REST API або локальні підключення через USB чи Wi-Fi [12]. На рисунку 1.5 показано приклад інтеграції погодних даних у Node-RED.

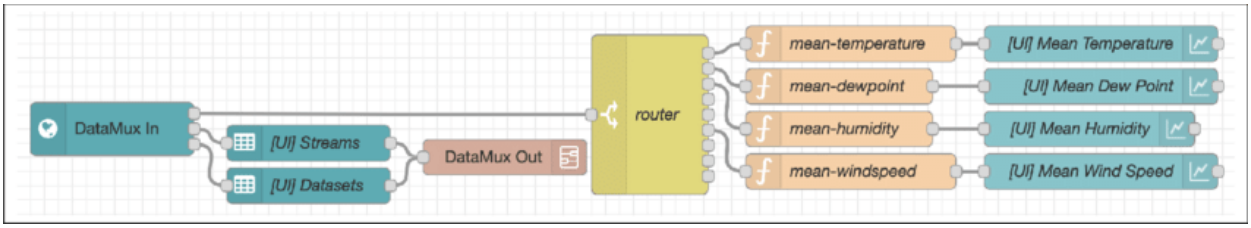


Рисунок 1.5 – Візуалізація погодних даних у Node-RED [13]

Такі рішення дуже прийнятні для проєктів з обмеженим бюджетом, навчальних стендів або локальних автоматизованих систем.

Щоб краще зрозуміти особливості застосування, наведено порівняння SCADA-системи Siemens WinCC та open-source Node-RED у табл. 1.2.

Таблиця 1.2 – Порівняння SCADA-систем Siemens WinCC та Node-RED

Характеристика	Siemens WinCC [10]	Node-RED+Raspberry Pi [12]
Тип ліцензії	Пропрієтарна, платна. WinCC є комерційним продуктом Siemens з ліцензійною політикою.	Відкрите програмне забезпечення, безкоштовне. Node-RED є open-source проєктом, який можна вільно використовувати та модифікувати.
Складність впровадження	Висока. Потребує досвіду роботи з Siemens TIA Portal та знань у галузі промислової автоматизації.	Низька. Підходить навіть для новачків завдяки простому інтерфейсу та великій кількості навчальних матеріалів.

Продовження таблиці 1.2

Характеристика	Siemens WinCC [10]	Node-RED+Raspberry Pi [12]
Платформи	Windows. WinCC працює на операційній системі Microsoft Windows.	Windows, Linux, Raspberry Pi. Node-RED є кросплатформеним і може працювати на різних операційних системах.
Інтеграція з обладнанням	Широка підтримка промислових PLC, особливо від Siemens, з використанням стандартних протоколів, таких як OPC UA.	Підключення через MQTT, HTTP, USB, GPIO. Node-RED легко інтегрується з різноманітними пристроями та протоколами.
Масштабованість	Висока. Підходить для великих виробництв з можливістю розширення та інтеграції в комплексні системи.	Середня. Обмежена потужністю Raspberry Pi, але достатня для малих та середніх систем.
Візуалізація	Потужна, гнучка, професійні екрани з можливістю створення складних графічних інтерфейсів.	Простий веб-інтерфейс з можливістю створення панелей керування за допомогою drag-and-drop.

Продовження таблиці 1.2

Характеристика	Siemens WinCC [10]	Node-RED+Raspberry Pi [12]
Сфера використання	Промисловість, великі підприємства, де потрібна висока надійність та масштабованість.	Smart home, навчальні проєкти, малі системи автоматизації.

Ще один важливий напрям – системи дистанційного керування. Завдяки стрімкому розвитку веб-технологій, месенджер-ботів і мобільних панелей, сьогодні можливо з будь-якої точки світу:

– переглянути стан кліматичної установки;

– дізнатись рівень освітленості;

– увімкнути чи вимкнути вентилятор або лампу;

– змінити режим роботи автоматично чи вручну.

Такий підхід підвищує зручність керування, адаптивність до змін і загальну ефективність системи.

Правильно підібране поєднання контролера, сенсорів і програмного інструменту – це невід’ємна частина будь-якої системи автоматизації. Саме тут визначається, наскільки легко систему буде масштабувати, адаптувати або підтримувати в майбутньому. Вибір між PLC, SCADA або легкими open-source інструментами залежить від масштабу завдань і бюджету, але саме баланс між функціональністю й простотою – ключовий у сучасних рішеннях.

1.5 Інтерфейси користувача для віддаленого контролю інженерних систем

Ключем до ефективної системи управління є не лише визначення об’єкта керування, а й методу керування, який буде інтиїтивно зрозумілим до людини, яка не брала участь у його розробці. Саме інтерфейс – це інструмент коммунікації між користувачем і технічною частиною, який забезпечує взаємодію в реальному часі. Особливо це стосується об'єктів із розосередженою інфраструктурою, де фізичний доступ до обладнання не завжди можливий.

Серед основних рішень для реалізації віддаленого доступу вирізняють:

– чат-інтерфейси – реалізуються як боти у месенджерах. Дозволяють отримувати повідомлення, переглядати стан системи, змінювати параметри за допомогою команд або кнопок;

– веб-панелі – надають повноцінну графічну оболонку для моніторингу та керування. Містять індикатори, графіки, елементи взаємодії (кнопки, повзунки);

– мобільні застосунки – забезпечують повний контроль зі смартфона. Зазвичай включають синхронізацію з хмарними платформами або шлюзами;

– hmi-дисплеї – сенсорні панелі, вбудовані в шафи керування або розміщені безпосередньо в цеху. Працюють автономно, навіть без підключення до Інтернету.

Чат-боти у месенджерах (рис. 1.6) стали одним із найпростіших способів взаємодії з системами автоматизації. Вони не потребують складної інфраструктури, працюють через API і забезпечують базовий функціонал – отримання поточних даних, зміна режимів, надсилання повідомлень.

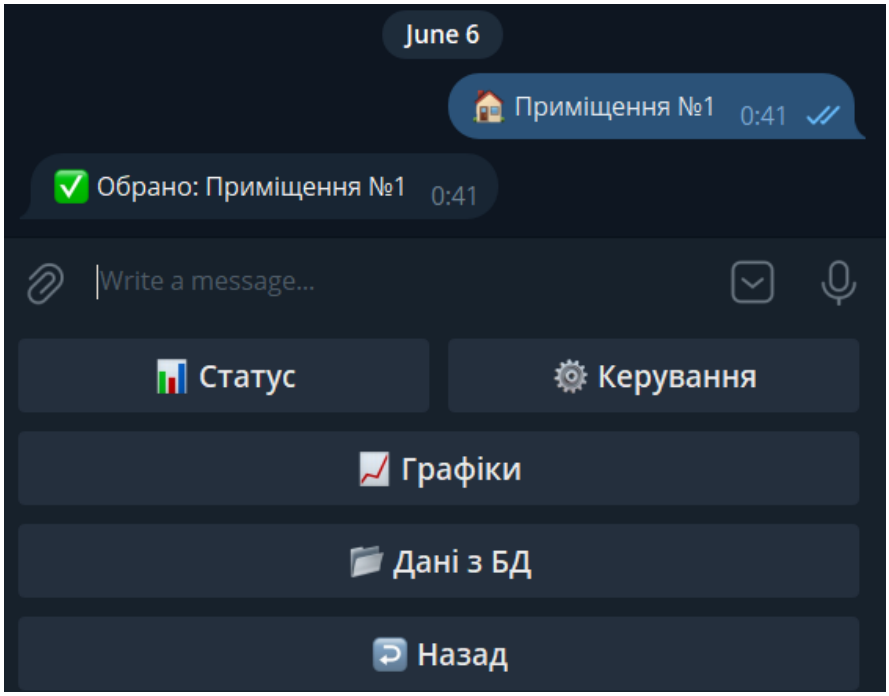


Рисунок 1.6 – Панель управління та моніторингу за допомогою чат-інтерфейса

Для складніших задач використовуються веб-інтерфейси, які дозволяють створювати повноцінні панелі керування з графікою, сенсорними елементами, живими графіками та системами логування. Node-RED [12], Home Assistant, OpenHAB – всі вони мають вбудовані редактори для створення подібних інтерфейсів без потреби писати код з нуля. Приклад інтерфейсу для моніторингу Home Assistant наведено на рисунку 1.7.

Рисунок 1.7 – Веб-панель Home Assistant зі значеннями температури/вологи [14]

На підприємствах, де важлива стабільність і контроль, веб-панелі часто інтегрують із SCADA-системами, забезпечуючи одночасно і зручність, і надійність промислового рівня.

Окремий клас – це мобільні застосунки, які синхронізуються з мікроконтролерами, шлюзами або хмарними платформами. Тут є як готові рішення, наприклад, Blynk (рис. 1.8), так і можливість створення власного застосунку під конкретну систему.

Завдяки мобільним додаткам керування освітленням, кліматом або вентиляцією стає можливим з будь-якої точки світу. Є доступ – є контроль. Наприклад, користувач може встановити оптимальний температурний режим перед початком робочої зміни, знизити інтенсивність вентиляції у неробочий час або отримати сповіщення при відхиленні параметрів середовища за межі норми.

Рисунок 1.8 – Приклад мобільного інтерфейсу для контролю мікроклімату на виробництві [15]

Так само залишаються доволі розповсюдженими HMI-панелі, які монтуються прямо в шафи управління або на стіни цехів. Їх основна перевага – локальність і надійність. Навіть якщо зникне інтернет або зв'язок з хмарою – оператор все одно зможе взаємодіяти з системою через сенсорний екран.

Такі панелі часто використовуються як дублюючий інтерфейс для аварійного чи ручного режиму роботи.

Отже, інтерфейси для віддаленого контролю вже давно вийшли за межі традиційних SCADA та кнопок на релейному щиті. Сьогодні це гнучкі, адаптивні і навіть персоналізовані рішення, які можуть бути частиною загального стилю підприємства або працювати у формі зручного месенджера у нашому смартфоні.

Сучасний підхід до проєктування інтерфейсів – є вимогою сучасних стандартів, де необхідність у динамічному виробничому середовищі, де

параметри змінюються швидко, а рішення потрібно приймати ще швидше. Це особливо актуально для систем освітлення та мікроклімату, які постійно реагують на зміну зовнішніх умов і потребують адаптації під задачі конкретного користувача.

1.6 Способи збереження, обробки та візуалізації даних у системах автоматизації

У системах автоматизації, дані – це основа для прийняття рішень, аналітики, прогнозування та вдосконалення роботи всієї системи. Тому способи збереження, обробки та візуалізації інформації мають вагоме значення для ефективності управління.

Одним із найпростіших підходів є зберігання локально – наприклад, у форматі CSV-файлів або у вбудованій пам’яті мікроконтролера (EEPROM). Такий варіант підходить для прототипів, тестів, або коли обсяг даних невеликий.

Для більш структурованого зберігання та подальшої обробки використовуються реляційні бази даних, такі як MySQL (рис. 1.9) або PostgreSQL. Вони дозволяють будувати запити SQL, фільтрувати, сортувати й обробляти великі обсяги даних. Перевагою є чітка таблична структура та робота з різними типами даних.

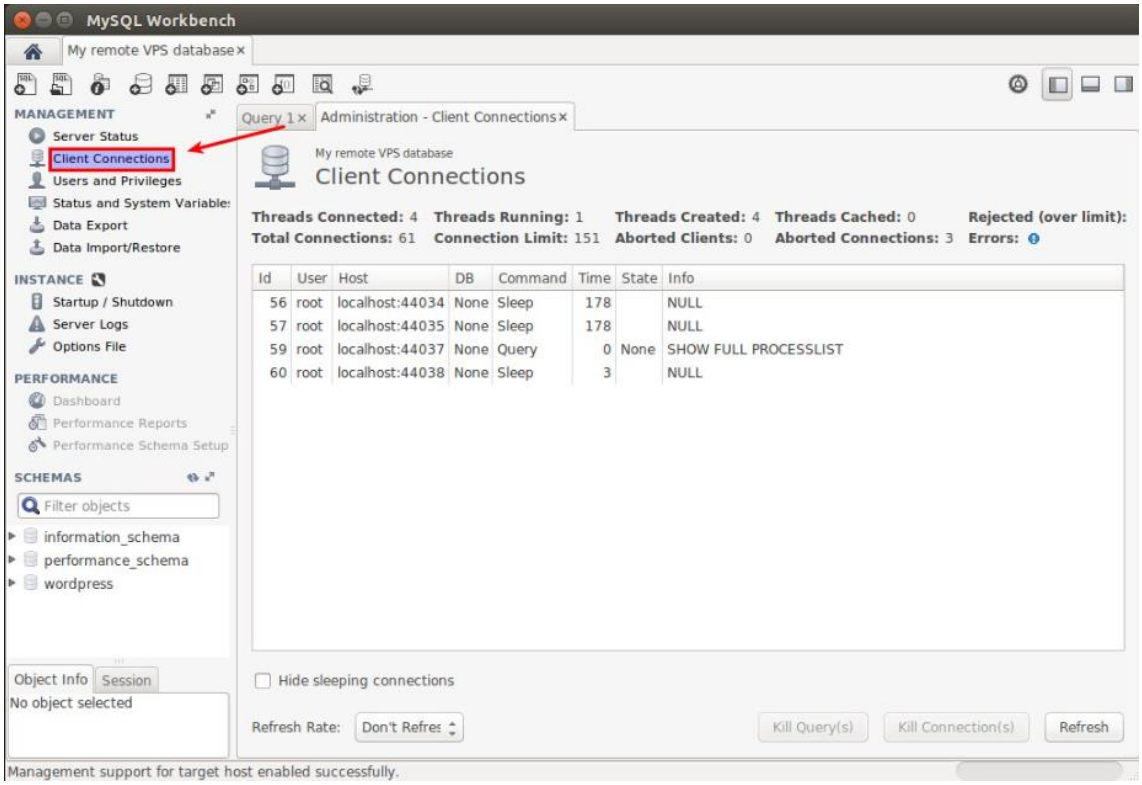


Рисунок 1.9 – Приклад інтерфейсу MySQL для моніторингу користувачів [16]

Проте реляційні СУБД не завжди ефективні при роботі з великими потоками часових рядів. У таких випадках доцільно використовувати спеціалізовані системи, зокрема InfluxDB – базу даних, створену саме для time-series data. Вона дозволяє:

– зберігати дані у вигляді «мітка часу → значення» [17];

– здійснювати агрегацію, фільтрацію, виведення даних за проміжками;

– інтегруватися з візуалізаторами типу Grafana або з Node-RED.

Інтерфейс InfluxDB з вікном Data Explorer показано на рисунку 1.10.

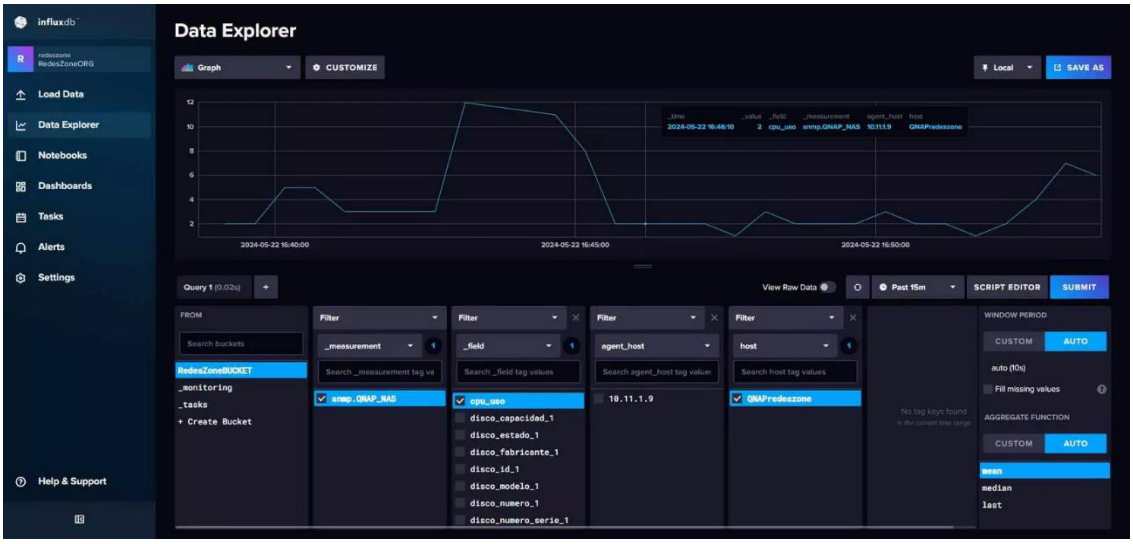


Рисунок 1.10 – Інтерфейс середовища InfluxDB [18]

У випадках, коли потрібно лише мінімальне збереження без складної структури, використовують SQLite – компактну, файл-орієнтовану СУБД, яка чудово підходить для вбудованих рішень, де є обмеження по ресурсах. На рисунку 1.11 наведено приклад створення БД для збереження інформації про покупців.

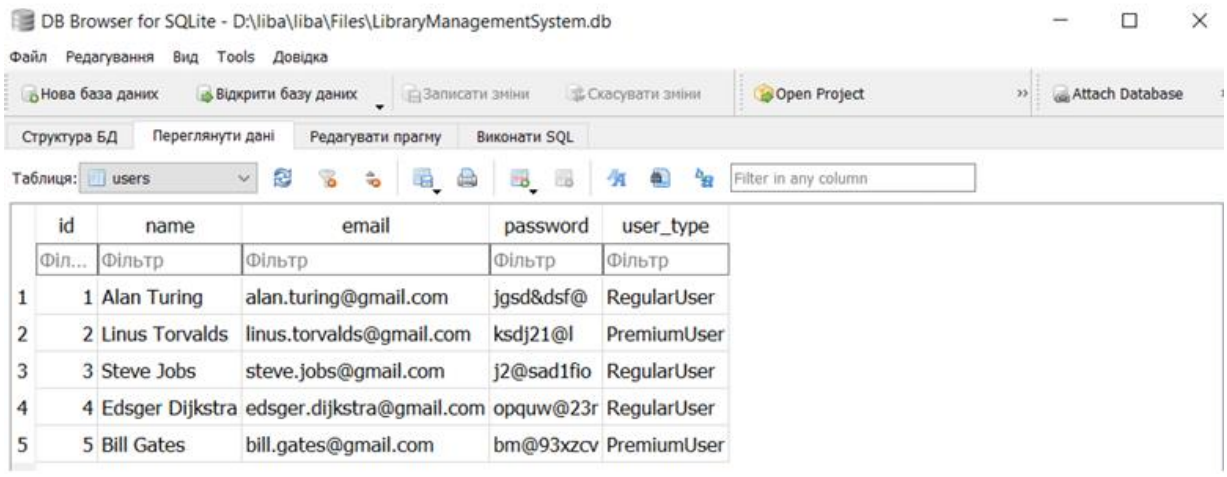


Рисунок 1.11 – Інтерфейс БД SQLite (власна ілюстрація)

З такої кількості варіантів з’ясовується, що кожна має свої переваги й обмеження, і вибір залежить від обсягу даних, частоти запису, потреб у структурованості та вимог до інтеграції з іншими сервісами (таблиця 1.3).

На відміну від реляційних баз даних, NoSQL-системи, зокрема Firebase Realtime Database або MongoDB, не використовують жорстку табличну структуру. Замість цього вони оперують об'єктами або документами, які можуть мати гнучку ієрархію. Це особливо зручно для систем, де структура даних може змінюватися або бути нестандартною.

У випадку Firebase, дані зберігаються у форматі JSON-дерева [19]. Кожен вузол дерева – це ключ-значення, і зміни можуть відбуватись у реальному часі. Наприклад, якщо система керування змінює стан реле або оновлює температуру – це миттєво синхронізується з базою й усі клієнти бачать зміни одразу. Такий підхід дозволяє реалізовувати push-нотифікації, динамічне оновлення інтерфейсу та мінімізувати час очікування.

У MongoDB дані представлені у вигляді документів BSON (розширення JSON), які зберігаються у колекціях. Це дозволяє легко зберігати, наприклад, показники температури з часовою міткою, додатковими мітками чи контекстом (зона, режим роботи, помилки). Такі бази ідеальні для журналювання подій або збереження історії, де кожен запис може бути унікальним за структурою.

Таблиця 1.3 – Порівняння методів збереження даних у системах автоматизації

Метод збереження	Тип системи	Переваги	Обмеження	Де застосовується
CSV-файл / EEPROM	Локальне, просте	Простота, мінімум ресурсів	Відсутність структури, складність обробки	Прототипи, тестові стенди, Arduino

Продовження таблиці 1.3

Метод збереження	Тип системи	Переваги	Обмеження	Де застосовується
SQLite	Вбудоване, офлайн	Легка СУБД, не потребує сервера	Обмежена масштабованість, немає багатокористувачності	ESP32, Raspberry Pi
MySQL / PostgreSQL	Реляційне	Потужні SQL-запити, структурованість	Повільна робота з великим потоком часових даних	SCADA, сервери збору даних
InfluxDB	Time-series	Агрегація, аналіз, інтеграція з Grafana	Не підходить для складних реляційних зв’язків	Клімат-контроль, енергомоніторинг
Firebase / MongoDB	Хмарне, NoSQL	Масштабованість, синхронізація, JSON-структура	Залежність від інтернету, платна модель при великих обсягах	Мобільні застосунки, IoT, чат-боти

Ще один важливий компонент – візуалізація. Без неї немає практичної доцільності збирати гігабайти значень, якщо оператор не може швидко оцінити ситуацію. Тут можливі кілька підходів:

– grafana – один із найпотужніших інструментів для візуалізації time-series даних, часто використовується з InfluxDB або Prometheus [20];

– node-red dashboard – дозволяє швидко створювати панелі з графіками, кнопками та індикаторами без написання коду [12];

– мессенджер-бот – можуть відправляти дані у зручному форматі прямо у смартфон;

– hmi-дисплеї – застосовуються для локального контролю без залежності від інтернету. Часто дублюють функції мобільних або веб-інтерфейсів.

1.7 Постановка завдання на розробку автоматизованої системи керування

У межах кваліфікаційної роботи ставиться завдання розробити автоматизовану систему керування освітленням та мікрокліматом для виробничого приміщення з можливістю віддаленого доступу, адаптації до робочого графіку підприємства та мінімізації енергоспоживання. Система має бути побудована з використанням доступних апаратних та програмних засобів, мати просту архітектуру, підтримку масштабування і модульного розширення.

Основна ідея полягає в створенні гнучкого середовища, здатного:

– зчитувати параметри навколишнього середовища (температура, вологість, CO2, освітленість);

– приймати рішення щодо ввімкнення або вимкнення виконавчих пристроїв (ламп підтримуючих димування, вентиляції, кондиціонування);

– зберігати історію станів та змін для подальшого аналізу;

– надавати користувачу інтуїтивний інтерфейс для моніторингу й керування.

Ядром системи є мікроконтролерна плата Master ESP32 підключена до виконавчих пристроїв, яка поєднана у єдину мережу із усіма Slave ESP32, які зчитують дані з цифрових сенсорів. Управління здійснюється як у повністю автоматичному режимі (на основі порогів і часу), так і з участю користувача через систему віддаленого доступу (інтерфейс реалізовано у вигляді чат-бота з логікою контекстного спілкування, кнопками та відповідями у реальному часі).

Особливості реалізації включають:

– часове керування – врахування робочого графіка підприємства. У позаробочий час світло та вентиляція переходять у енергозберігаючий режим або вимикаються;

– логіка адаптації – реагування на зміну температури та рівня освітленості відповідно до заданих порогів;

– інтерфейсна взаємодія – користувач може отримати поточні дані, дати команду або змінити поріг спрацювання прямо з мобільного пристрою;

– збереження даних – усі події фіксуються в базі даних, що дозволяє не лише відстежувати історію, а й виявляти шаблони або аномалії в роботі системи.

З метою обмеження залежності від хмарних сервісів та підвищення автономності, система має працювати локально, з можливістю розгортання на локальному сервері або віртуальній машині. Дані зберігаються в реляційній базі даних (наприклад, SQLite), яка забезпечує легкість інтеграції з іншими програмними модулями та простоту обслуговування.

Очікуваний результат – створення прототипу системи, який продемонструє можливість автоматичного керування кліматом та освітленням, з опцією ручного втручання через віддалений інтерфейс. Такий підхід формує основу для подальшої розробки прототипу системи, що відповідає сучасним вимогам до енергоефективного управління виробничим середовищем.

2 РОЗРОБКА СТРУКТУРИ МАКЕТУ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ВІДДАЛЕНОГО КЕРУВАННЯ ОСВІТЛЕННЯМ ТА КЛІМАТОМ У ВИРОБНИЧИХ ПРИМІЩЕННЯХ

2.1 Обґрунтування вибору архітектури системи

Автоматизована система управління мікрокліматом і освітленням повинна відповідати умовам обраного виробництва, в нашому випадку ми націлилися на харчове виробництво, але система є доволі універсальною, де до навколишнього середовища пред'являється підвищений набір вимог, таких як температурний режим, вологість, освітлення, вентиляція, заощадження електроенергії без втрати функціоналу. Від стабільності параметрів мікроклімату безпосередньо залежать робочі процеси, що відбуваються у цих приміщеннях. Крім того, у виробничих зонах часто використовується обладнання, чутливе до пилу, перегріву або вологи, тому необхідна ефективна система кондиціонування і контролю повітря.

Система повинна бути досить точною, щоб підтримувати температуру і вологість в заданих межах і не вимагати постійного ручного регулювання. Необхідно мати гнучку роботу вентиляції, контрольоване освітлення за розкладом, а також функцію аварійного оповіщення в деяких зонах, враховуючи санітарні норми та енергетичну політику, у разі виявлення витоку або перевищення рівня CO_2.

Крім того, оскільки ми маємо справу з малим або середнім підприємством, простота системи дозволяє уникнути громіздких серверних рішень, підтримує автономну роботу локально і наближається до мінімуму в установці, проводці та експлуатаційному бюджеті.

З огляду на масштабовану точність контролю параметрів середовища, від енергоефективності та простоти інсталяції залежить рішення застосовувати розподілену модульну архітектуру, де кожне приміщення або окрема зона обслуговуються пов’язаними вузлами на базі ESP32 з набором сенсорів і виконавчих пристроїв в межах своєї зони.

Даний принцип дозволяє запобігти централізації і прив'язці до поточного сервера або головного контролера, гарантувати автономність і зручність управління в межах кожної зони, просте масштабування системи з розвитком виробничих площ.

Всередині вузла розподілена логіка, при якій один контролер здійснює керування виконавчими елементами, а інші вузли, якщо вони є, функціонують у його сенсорному підпорядкуванні. Перед подачею команди на включення або виключення вентиляції та освітлення контролер перевіряє наявність реле в необхідному положенні, що дозволяє уникнути двократного виконання одного й того ж дії та збереження логічної послідовності процесів.

Інтерфейс віддаленого доступу розроблений для взаємодії з користувачем, він забезпечує перегляд стану системи, отримання сповіщень та коригування порогів реагування.

Передача даних між пристроями в межах вузла може здійснюватись окремо взятої локальної Wi-Fi мережею. Як розширений варіант може застосовуватись легкий протокол MQTT для обміну даними між зонами або із зовнішнім сервером статистичного збору.

Отже, такий тип архітектури поєднує локальну автономність в окремих приміщеннях із можливістю централізованого моніторингу через систему віддаленого доступу, що забезпечує надійність та простоту при розгортанні.

Оскільки для виконання задач системи потрібні мікроконтролери, ESP32 має вбудований Wi-Fi модуль, достатньо GPIO, підтримуються основні інтерфейси (I2C, UART, SPI) та має доволі високу обчислювальну потужність (до 240 МГц, два ядра), це дає йому змогу обробляти показники з кількох

датчиків, управляти виконавчими механізмами з прийняттям рішенняй та підтримувати зв'язок із системою віддаленого доступу.

Якщо порівняти з Arduino Uno та Mega, для яких потрібні додаткові модулі для підключення до мережі, ESP32 такої необхідності не має. Він також споживає значно менше енергії, ніж Raspberry Pi, і не потребує інсталяції операційної системи.

Отже, вважаю, що ESP32 найкраще підходить для створення автономних бездротових вузлів, які здійснюють локальне керування та передають дані до бази даних.

Для того щоб забезпечити точний контроль параметрів середовища, система розбивається на зони, при цьому кожна масштабується під окремий вузол ESP32. У цьому разі можна адаптувати конфігурацію під особливості приміщення та надати їй локальне керування без надмірного навантаження на мережу.

На рівні зони:

– один вузол виконує функцію керування (реле та димер підключено, логіка вмикання/вимикання);

– інші працюють як сенсорні підлеглі (вимірювання температури, вологості, CO_2 тощо).

Кількість вузлів визначається залежно від:

– площі приміщення (орієнтовно один вузол на 30–40 м²);

– фізичного розташування обладнання (наприклад, мийні зони, камери зберігання, теплі/холодні зони);

– необхідної точності контролю (окремі параметри можуть істотно коливатися в різних кутах приміщення).

Датчики встановлюються на висоті двох метрів у пластиковій коробці захищеною від зовнішніх пошкоджень сітчастими стінками, розташовуються датчики таким чином:

– CO_2 – всередині коробки;

– температура/вологість – всередині, подалі від техніки;

– датчики протікання – поблизу підлоги, по точках можливих витоків, не вважаю важливим використання саме DIY рішень, краще встановити окремий пристрій, типу Neptun, бо його ціна не є великою;

– датчики диму або газу – на стелі, теж краще встановити сертифіковане обладнання, замість DIY рішень;

– датчик освітленяя – у примішеннях з вікнами для адаптації з природнім освітленням, на стіні, паралельно до підлоги та стелі.

Комунікація між вузлами та інтерфейсом здійснюється за допомогою Wi-Fi із можливістю джерела локального зберігання або передачі даних до центральної бази. За потребою кожен з вузлів може працювати автономно, що забезпечить надійність системи.

Передача даних у системі організована шляхом локальної Wi-Fi-мережі, до якої приєднуються всі вузли на базі ESP32. Це сприяє бездротовій взаємодії в системі та мінімізує кількість кабелів, що у свою чергу, спрощує масштабування нових застосувань системи.

Обміном інформації між вузлами та інтерфейсом віддаленого доступу здійснюють через HTTP-запити або ж через MQTT.

Стандартна дія вузла:

– зчитування сенсорних значень;

– прийняття рішень щодо керування виконавчими пристроями;

– передача даних через локальний сервер до бази даних;

– отримання запитів через сервер на зміни станів реле та димерів від користувача.

Користувач має змогу взаємодіяти із системою через зручний чат-інтерфейс, де він може переглядати поточні показники, отримувати повідомлення про події, а також регулювати режими роботи. Усі команди користувача надходять у вигляді простих запитів, а внутрішня логіка керуватиме та здійснюватиме усе, що їй потрібно.

У випадку втрати зв'язку вузол продовжуватиме працювати автономно за внутрішнім алгоритмом. Це дозволяє зберегти функціональність навіть при тимчасових перебоях у мережі.

При тимчасових перебоях з чат-ботом, реалізовано альтернативний інтерфейс у вигляді сайту зі спрощеним функціоналом, який може працювати локально.

Електроживлення системи може бути організованим централізовано, за допомогою одного імпульсного блоку живлення 12 В, 60 Вт (5 А), встановленого біля панелі управління.Такий вибір обумовлений тим, що ESP32 працює на 3.3–5 В, а подача живлення 12 В по кабелю дозволяє зменшити втрати напруги на великих відстанях. За умови використання типового вузла ESP32 з кількома сенсорами (до 300–350 мА на один вузол), такий блок живлення здатен паралельно забезпечити стабільну роботу до 12–14 вузлів ESP32. Такий підхід забезпечує стабільну подачу енергії до всіх вузлів системи без необхідності використання окремих адаптерів або батарей.

Двожильний мідний кабель прокладається від джерела живлення по периметру об'єкта в кабельних каналах. До кожного підлеглого вузла ESP32 підводяться відгалуження, які монтуються на висоті близько 2 метрів у вентильованих пластикових корпусах. Кожен вузол отримує стабільне живлення 5 В через понижуючий перетворювач постійного струму (з 12 В до 5 В).

Master ESP32 також може живитися від тієї ж лінії 12 В, або він живився окремо від стандартного адаптера 5 В, підключеного до мережі 220 В змінного струму, але це не зовсім доцільно враховуючи вже готовий концепт через блок живлення.

Така централізована схема живлення допомагає зменшити кількість окремих джерел живлення, спрощує монтаж і гарантує безпеку завдяки використанню низьковольтної проводки. Це також дозволяє легко масштабувати систему для додаткових зон, якщо це необхідно. Для швидкого, надійного та зручного з'єднання кабелів використовуються клемні колодки типу WAGO.

Теоретична вартість реалізації такої системи розподілу електроенергії для 7 вузлів ESP32 на площі приблизно 250 квадратних метрів оцінюється приблизно від 3500 до 4500 грн, залежно від конкретних матеріалів і компонентів, що використовуються.

Таким чином, обрана комунікаційна модель забезпечує стабільну взаємодію між вузлами, а також дає змогу віддалено контролювати систему без складного налаштування або додаткового ручного втручання у технічні параметри.

2.2 Вибір основних компонентів системи

Особливістю цієї системи є те, що вона використовує розподілені вузли, кожен з яких забезпечує локальний контроль параметрів середовища та керування виконавчими пристроями. Отже, все має бути компактним, енергоефективним та інтегрованим у загальну архітектуру.

По кожній категорії компонентів проводилося порівняння технічних характеристик з урахуванням їхнього практичного застосування за умов малого або середнього виробництва. На основі проведеного аналізу були сформовані порівняльні таблиці й здійснено вибір оптимальних рішень для реалізації.

У рамках реалізації системи було обрано мікроконтролер з родини ESP (рис. 2.1) як базовий елемент вузла, оскільки ці контролери поєднують вбудовану підтримку Wi-Fi, достатню обчислювальну потужність, невисоку вартість та широку підтримку середовищ розробки.

а) б)

а) ESP8266 [6]; б) ESP32 [6]

Рисунок 2.1 – Найпоширеніші мікроконтролери родини ESP

Серед доступних рішень (див. таблицю 1.1) перевагу надано саме ESP32, який:

– має двоядерний процесор до 240 МГц, що дозволяє одночасно обробляти дані, виконувати логіку та підтримувати зв’язок;

– підтримує до 34 GPIO, що забезпечує підключення кількох сенсорів і виконавчих елементів;

– оснащений вбудованими модулями Wi-Fi та Bluetooth, чого достатньо для бездротової роботи системи без потреби в додаткових адаптерах;

– забезпечує підтримку протоколів I2C, SPI, UART, необхідних для підключення сенсорів різного типу.

У порівнянні з ESP8266 або іншими мікроконтролерами, ESP32 демонструє кращу продуктивність і більшу кількість доступних портів, що є важливим для розширення системи або роботи з кількома сенсорами в одному вузлі.

Для контролю мікроклімату на виробництві критично важливо мати стабільні показники температури та вологості. Нижче наведено аналіз трьох поширених сенсорів (рис 2.2, табл. 2.1), які можуть бути інтегровані у вузол системи автоматизації.

а) б) в)

а) DHT22 [21]; б) BME280 [22]; в) AM2320 [23]

Рисунок 2.2 – Датчики температури та вологості

Таблиця 2.1 – Порівняння сенсорів температури та вологості

Модель сенсору	DHT22	BME280	AM2320
Тип сигналу	Цифровий (1-Wire)	I2C / SPI	I2C
Точність температури (°C)	±0.5	±0.5	±0.5
Точність вологості (%)	±2–5	±3	±3
Діапазон температури (°C)	-40…80	-40…85	-40…80
Напруга живлення (В)	3.3 / 5	3.3	3.1–5.5
Особливості	Дешевий, але повільний	Стабільний, міряє тиск	Схожий на DHT22, але I2C
Ціна	~ 155 грн	~ 160 грн	~ 235 грн

Аналіз технічних характеристик наведених сенсорів (рис 2.3, табл. 2.2) показав, що найдоцільнішим вибором для розроблюваної системи є BME280,

оскільки він забезпечує необхідну точність, стабільність показників, працює з протоколами I2C та SPI і додатково дозволяє вимірювати атмосферний тиск.

Для контролю повітряного середовища в закритих приміщеннях виробничого типу важливо точно відстежувати концентрацію вуглекислого газу. Підвищений рівень CO_2 негативно впливає на самопочуття персоналу.

а) б) в)

а) MQ-135 [24]; б) MH-Z19B [25]; в) SCD30 [26]

Рисунок 2.3 – Датчики CO_2

Таблиця 2.2 – Порівняння сенсорів CO_2

Модель сенсору	MQ-135 [24]	MH-Z19B [25]	SCD30 [26]
Тип сигналу	Аналоговий	UART / PWM	I2C
Діапазон вимірювання (ppm)	100–10000	0–5000	400–10000
Точність	Низька, потребує калібрування	±50 ppm + 5%	±30 ppm + 3%
Калібрування	Ручне	Автоматичне	Автоматичне

Продовження таблиці 2.2

Модель сенсору	MQ-135 [24]	MH-Z19B [25]	SCD30 [26]
Напруга живлення (В)	5	4.5–5.5	3.3/5
Особливості	Дешевий, але нестабільний	Надійний, простий у підключенні	Прецизійний, додатково міряє температуру і вологість, дорогий
Ціна	~ 70 грн	~ 980 грн	~ 2600 грн

У межах цієї розробляємої системи краще використовувати MH-Z19B як більш доступну та перевірену альтернативу, та SCD30, якщо це узгоджено із підприємством згідно бюджету і йде як альтернатива вже обраному нами датчику температури та вологості BME280.

У випадку, коли виробниче приміщення має доступ до природного освітлення, доцільним є використання сенсора освітленості (рис 2.4, табл. 2.3) для адаптивного керування штучним світлом. Це дозволяє зменшити витрати електроенергії та уникнути перевищення допустимого рівня освітлення. Навіть у повністю закритих зонах сенсор може виконувати функцію контролю справності освітлення або реагування на аварійне затемнення.

а) б)

а) TEMT6000 [27]; б) GY-302 BH1750FVI [28]

Рисунок 2.4 – Датчики освітлення

Таблиця 2.3 – Порівняння датчиків освітлення

Параметр	TEMT6000 [27]	BH1750 [28]
Тип сигналу	Аналоговий	Цифровий
Тип виходу	Напруга, пропорційна освітленості	Готове значення в люксах
Діапазон освітленості (люкс)	1–1000	1–65535
Чутливість	Середня, пікова чутливість на 570 нм	Висока, адаптована до спектру людського ока
Інтерфейс	ADC	I2C
Напруга живлення (В)	5	3–5
Особливості	Простий у використанні, лінійна характеристика, не реагує на ІЧ та УФ світло	Точний, стабільний, вбудований АЦП, низьке енергоспоживання
Ціна	~ 50 грн	~ 50 грн

Обидва сенсори мають свої переваги. TEMT6000 – простий у використанні, з лінійною характеристикою та середньою чутливістю, що робить його

придатним для базових застосувань. BH1750 забезпечує високу точність вимірювань, широкий діапазон освітленості та цифровий вихід, що спрощує інтеграцію та обробку даних.

Але використанням одних датчиків освітлення все не завершується, якщо виробниче приміщення обладнане освітлювальними приладами змінного струму (наприклад, світлодіодними або галогенними лампами на 220 В), доцільним є не лише їх вмикання/вимикання, а й регулювання рівня яскравості відповідно до змін природного освітлення.

Для реалізації плавного керування яскравістю було обрано використання AC Dimmer Module (рис. 2.5) на основі симістора. Цей модуль дозволяє програмно змінювати фазу включення в кожному циклі змінного струму. Завдяки цьому можна керувати потужністю, що подається на навантаження, і, відповідно, регулювати яскравість ламп.

Важливо зазначити, що для правильної роботи диммера змінного струму в промислових умовах слід використовувати лише лампи з підтримкою димування (dimmable LED). Стандартні світлодіодні лампи без регулювання яскравості або люмінесцентні лампи не дозволяють плавно регулювати яскравість і можуть мерехтіти, працювати неправильно або не вмикатися при низькій напрузі фаз.

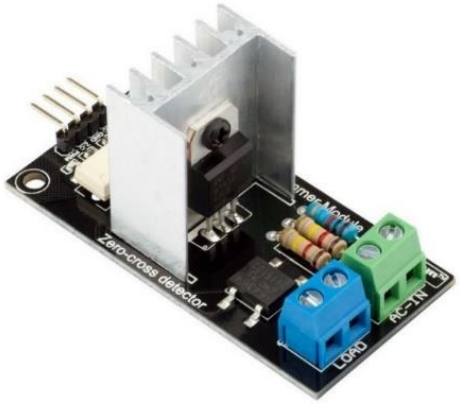


Рисунок 2.5 – AC Dimmer Module [29]

Керування модулем здійснюється безпосередньо з ESP32 за допомогою цифрового піну. Для реалізації алгоритму фазового керування використовується спеціалізована бібліотека RBDDimmer, яка синхронізується з точками переходу через нуль у синусоїді змінного струму. На основі даних з сенсора освітленості BH1750, мікроконтролер визначає рівень поточного освітлення в приміщенні та передає відповідний сигнал диммеру.

Тому для реальної реалізації системи необхідно або

– використовувати спеціалізовані світлодіодні лампи з можливістю регулювання яскравості, які чітко позначені як такі на упаковці;

– або застосувати альтернативне релейне зональне керування освітленням, де яскравість імітується шляхом ступеневого ввімкнення окремих груп ламп (наприклад, 33%, 66%, 100%).

Таким чином, у системі реалізується адаптивне керування штучним освітленням, при якому яскравість ламп змінюється в залежності від природного світла та може повністю замінити реле для керування освітленням, за умови використання ламп, сумісних з димуванням. Це не тільки підвищує енергоефективність, а й дозволяє гнучко налаштовувати сценарії роботи освітлення відповідно до часу доби або зовнішніх умов.

У виробничих приміщеннях, де є мийні зони, трубопроводи, резервуари або обладнання з водяним охолодженням, необхідно забезпечити своєчасне виявлення протікання (рис. 2.6, табл. 2.4). Навіть незначне скупчення вологи біля обладнання може призвести до короткого замикання або зупинки процесу.

а) б)

а) Neptun SW007 [30]; б) SEN18 [31]

Рисунок 2.6 – Датчик та модуль протікання

Таблиця 2.4 – Порівняння датчиків протікання

Параметр	Neptun SW007 [30]	SEN18 [31]
Тип	Професійний дротовий датчик	Бюджетний аналоговий сенсор
Призначення	Інтеграція в системи контролю протікання води	Виявлення наявності води на поверхні
Інтерфейс	Два дроти (сухий контакт), потребує підключення до контролера або модуля	Аналоговий вихід (0–5 В), можна підключати безпосередньо до аналогового входу ESP32
Живлення	Не потребує живлення	3.3–5 В

Продовження таблиці 2.4

Параметр	Neptun SW007 [30]	SEN18 [31]
Сумісність з ESP32	Потребує додаткового інтерфейсу або реле для підключення	Пряма сумісність через аналоговий пін
Монтаж	Кругла форма, призначена для вбудовування в плитку або поверхню	Плоский сенсор, розміщується на поверхні
Особливості	Висока надійність, захист IP67, сумісність з системами Neptun	Простота використання, низька вартість, можливість швидкої інтеграції в DIY-проєкти
Рекомендовано для	Професійних систем захисту від протікань у житлових та комерційних об'єктах	Прототипування, навчальні проєкти, невеликі системи моніторингу
Ціна	~ 700 грн	~ 50 грн

В залежності від вимог та бюджету виробництва може бути обраний любий з цих датчиків, але для професійних інсталяцій, тип паче для підприємства краще обрати Neptun SW007 як датчик чіткого реагування, аварійного сповіщення та перекривання постачання води.

У виробничих приміщеннях, де можливе скупчення парів, диму або горючих газів (наприклад, у зонах з паянням, роботою нагрівальних елементів або поблизу хімічних речовин), доцільним є впровадження газового сенсора (рис. 2.7).

Рисунок 2.7 – Датчик MQ-2 [32]

Сенсор MQ-2 хоч і позиціонується як універсальний для диму та газів, на практиці демонструє нестабільну чутливість до диму, особливо у слабко концентрованих чи розсіяних джерелах. Його аналоговий вихід часто шумний, потребує прогріву та точного калібрування, а реакція може бути запізнілою або некоректною. У виробничих умовах, де важлива швидка й надійна реакція на задимлення, MQ-2 не забезпечує належного рівня безпеки. Тому доцільніше використовувати окремий сертифікований пожежний датчик, який має гарантовану швидкість спрацювання, тестування на чутливість до диму, а також може бути інтегрований у систему як окремий модуль.

Для керування вентиляцією доцільно використовувати механічне реле (рис. 2.8 а) SRD-05VDC-SL-C (~ 40 грн), яке витримує напругу до 250 В і навантаження до 10 А. Воно підходить для вмикання вентиляторів змінного струму, має оптопару для гальванічної розв’язки та легко керується з ESP32. У разі потреби в безшумній роботі або при частих перемиканнях рекомендується твердотільне реле (SSR) (рис. 2.8 б) (~160 грн), яке має вищу швидкодію та ресурс. Обидва варіанти забезпечують надійне комутування навантаження і повну інтеграцію з логікою системи.

а) б)

а) SRD-05VDC-SL-C [33]; б) SSR-40 DA [34]

Рисунок 2.8 – Механічне та твердотільне реле

2.3 Розробка структурної схеми

За результатами архітектурного аналізу системи була складена логічна структурна схема, яка зображена на рисунку 2.9.

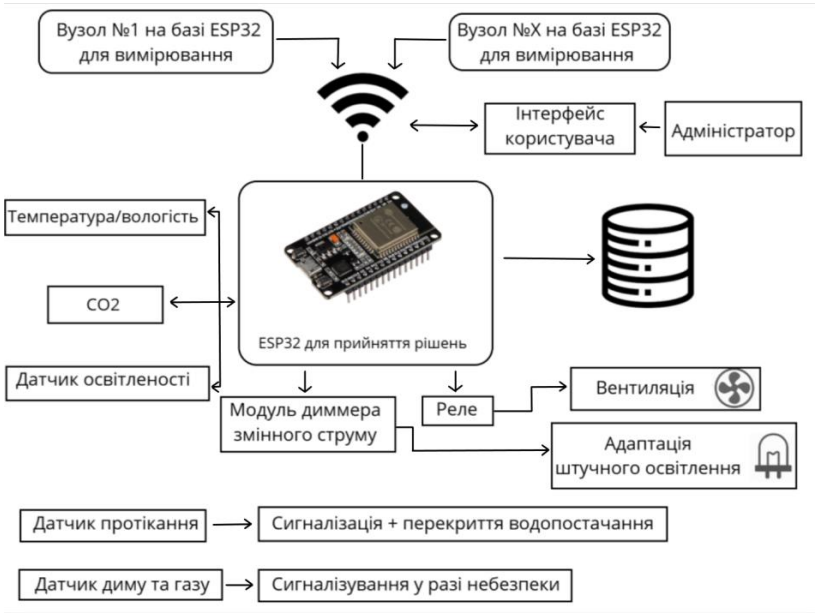


Рисунок 2.9 – Структурна схема автоматизованої системи управління температурою та кліматом на підприємстві

Схема показує основні компоненти автоматизованого рішення та взаємозв'язки між ними. ESP32 є ядром системи, що приймає рішення на основі даних, отриманих від датчиків температури, вологості, CO_2 та освітленості. Паралельно можуть працювати вузли на базі ESP32, передаючи інформацію через бездротовий канал.

Контролер, залежно від проаналізованих даних, активує модуль, що складається з реле і диммерів для змінного струму. За допомогою цих приводів вмикається вентиляція або приглушується освітлення відповідно до поточних умов у приміщенні. Система також передбачає обробку сигналів тривоги від датчиків протікання води і виявлення диму, що тягне за собою реакцію, яка полягає в активації сигналізації або перекритті водопостачання.

Вся інформація зберігається в базі даних, а взаємодія з оператором здійснюється через інтерфейс віддаленого доступу. Крім того, над ним реалізовано адміністративний рівень, що дозволяє керувати конфігураціями системи та політиками реагування.

2.4 Розробка алгоритму роботи системи

Алгоритм функціонування розробленої системи автоматизованого керування (рис. 2.10) побудовано з урахуванням розподіленої архітектури та використання хмарного середовища як логічного ядра.

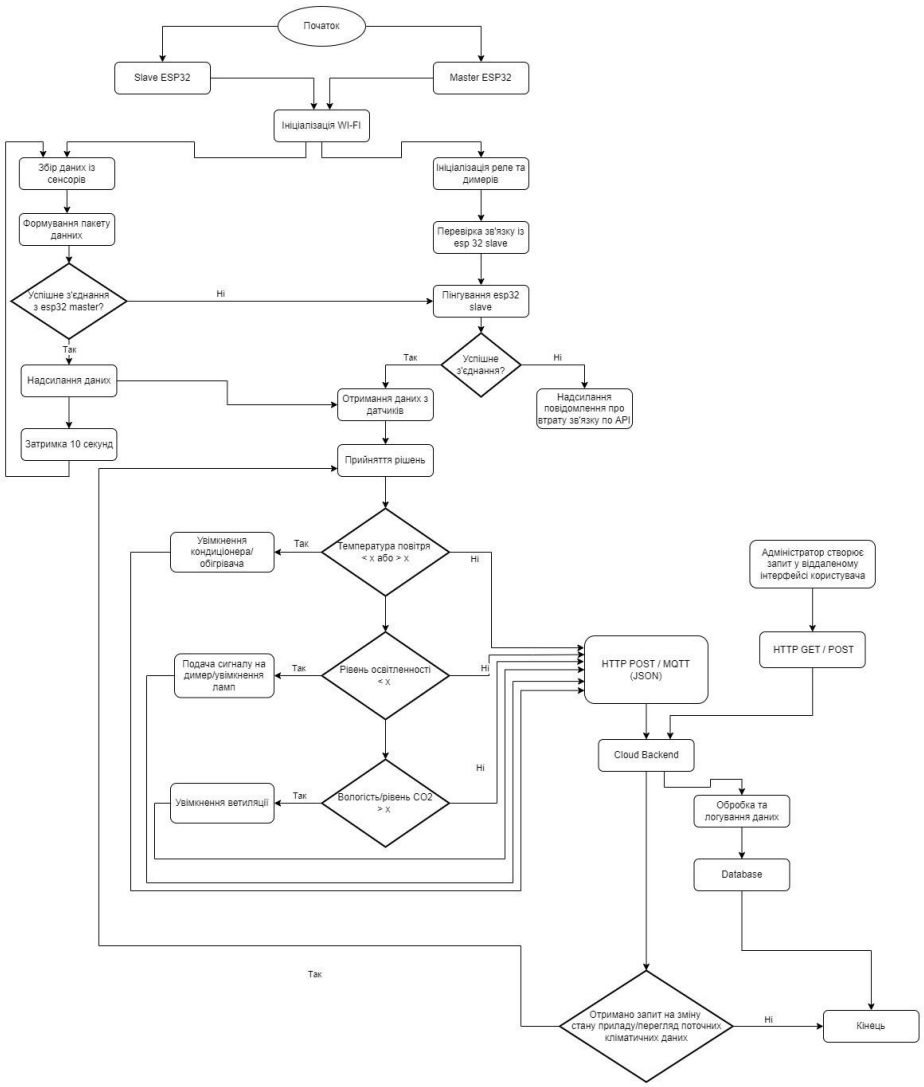


Рисунок 2.10 – Алгоритм роботи автоматизваної системи керування освітленням та кліматом

Робота системи починається з запуску мікроконтролера Master ESP32 та усіх Slave ESP32, Master ESP32 після ініціалізації здійснює збирання та попередню обробку даних від підключених сенсорів. Отримані дані структуруються у форматі JSON та передаються в хмарну частину системи – Cloud Backend – за допомогою HTTP-запитів або MQTT-протоколу. У хмарному середовищі ці дані приймаються через відкритий API, верифікуються, обробляються та зберігаються у базі даних для подальшого аналізу або звернень з боку користувача.

Паралельно з цим до Cloud Backend підключено віддалений інтерфейс користувача у вигляді бота. Користувач може звернутися до системи через бот, запитуючи поточний стан параметрів середовища або надсилаючи команди на зміну стану керованих пристроїв. Бот обробляє вхідні повідомлення та здійснює відповідні звернення до Cloud Backend через REST API. У випадку запиту даних, бот передає GET-запит, а у випадку надсилання команди, виконується POST-запит. Хмарний сервер обробляє такі запити, звертається до бази даних (для читання або оновлення значень), формує відповідь або керуючий сигнал, і, за необхідності, надсилає цю команду назад до Master ESP32.

Зворотній зв’язок з Master ESP32 забезпечує завершення циклу обміну інформацією. Якщо користувач надсилає команду через бота, хмара передає її у вигляді HTTP GET або іншого керуючого запиту на відповідний API-ендпойнт мікроконтролера. Master ESP32 приймає цю команду, інтерпретує її та активує необхідний виконавчий елемент (реле, димер тощо). Таким чином реалізується повний цикл замкненого управління, де користувач не лише може спостерігати за станом системи, але й дистанційно керувати її роботою.

2.5 Створення макету підсистем

Для перевірки працездатності системи було створено два функціональних макети: один виконує роль головного пристрою (Master ESP32), другий – підлеглого (Slave ESP32), який відповідає за зчитування параметрів середовища. Реалізацію обох макетів виконано у віртуальному середовищі моделювання Wokwi з урахуванням обмежень платформи, що накладає певні технічні компроміси – зокрема, використання OLED-дисплею замість окремого інтерфейсу, використання датчиків для Master ESP32, хоч у реальній системі це може залишатися опційним, або LED-індикаторів замість реле з реальним навантаженням. Незважаючи на це, вдалося реалізувати базову логіку обміну

даними, керування виконавчими пристроями та візуального зворотного зв’язку (див. додаток Ж).

На рисунку 2.11 зображено макет Master ESP32, до якого підключено:

– два релейні модулі для керування системами освітлення та вентиляції;

– світлодіоди як індикатори стану освітлення та вентиляції;

– датчик температури й вологості (DHT22);

– газовий датчик, для вимірювання CO2 (MQ2);

– димер (потенціометр), що імітує регулювання освітленням за вхідним значенням у люксах;

– OLED-дисплей для виводу інформації.

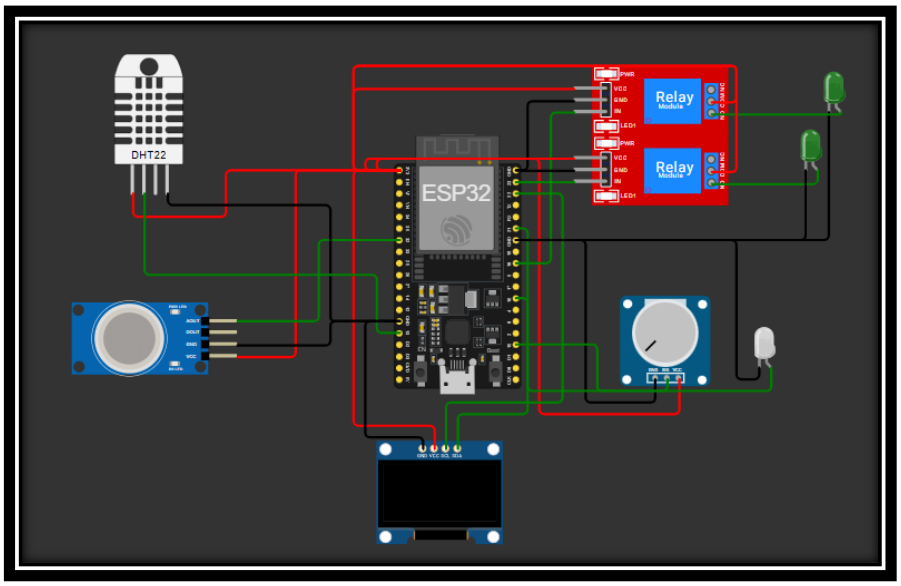


Рисунок 2.11 – Макет головного вузла системи на базі Master ESP32

На рисунку 2.12 представлено макет Slave ESP32, призначений для збирання даних з навколишнього середовища (див. додаток Ж). У ньому реалізовано:

– зчитування температури та вологості за допомогою DHT22;

– аналіз стану повітря за MQ2;

– фоторезистор для визначення рівня освітленності у приміщенні;

– передачу стану у вигляді світлодіодної індикації.

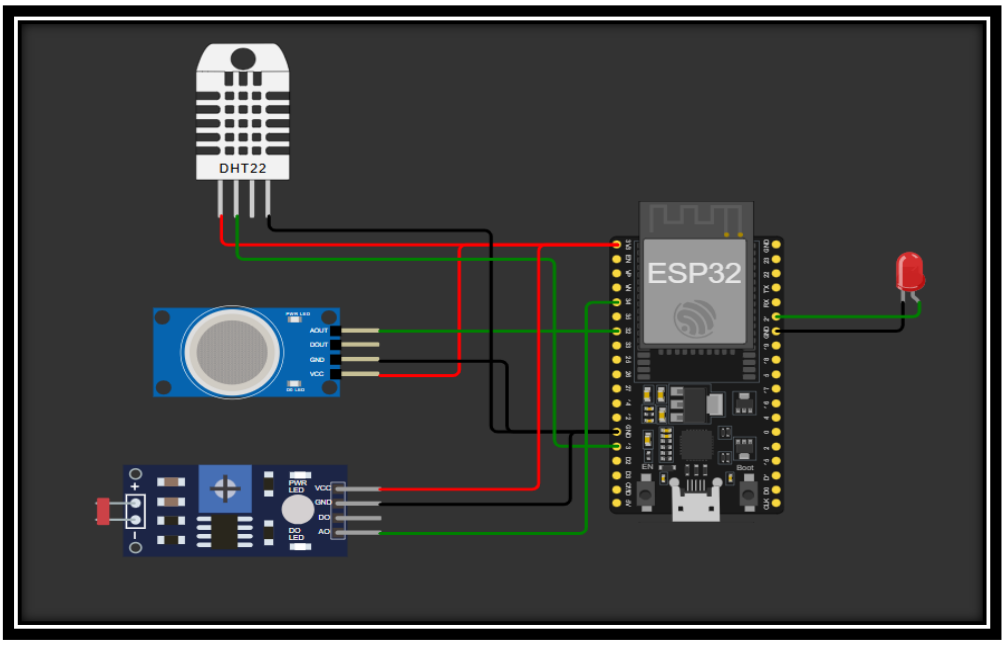


Рисунок 2.12 – Макет підлеглого вузла системи на базі Slave ESP32

Макетування проводилось задля тестування на працездатність основної логіки системи, алгоритму обміну даними та реакції на зміну вхідних параметрів. Попри обмеження, платформою були реалізовані основні функції: сенсорне зчитування, передача умовного сигналу, керування реле й візуалізація стану.

2.6 Опис використаних бібліотек

У процесі реалізації проєкту було використано низку бібліотек, які забезпечили зручну побудову серверної частини, взаємодію з зовнішніми пристроями, зберігання та обробку даних, а також створення графічного інтерфейсу. Частина з них є вбудованими у Python, інші – сторонні, встановлені через менеджер пакетів pip.

Застосовані бібліотеки умовно поділяються за функціональним призначенням:

Бібліотеки для створення API та інтерфейсів користувача:

– fastapi – основа серверної частини системи. Забезпечує обробку HTTP-запитів, реалізацію маршрутизаторів, обробку форм (у модулі fallback_ui.py) та асинхронну взаємодію з симуляторами;

– uvicorn – сервер для запуску FastAPI-додатка, підтримує "гаряче" оновлення та асинхронність. Використовується для запуску master і slave API.

– requests – бібліотека для виконання HTTP-запитів між модулями (бот, fallback UI, симулятори);

– htmlresponse, form (fastapi) – використовуються для створення fallback вебінтерфейсу з HTML-форми керування.

Бібліотеки для структурування та обробки даних:

– pydantic – валідація вхідних даних для API-запитів (класи SensorData, ZoneData, PingData у routes.py);

– json – використовується для обробки конфігураційного файлу work_schedule.json із робочим графіком;

– datetime, pytz – для обробки часу, переведення до часових зон, перевірки, чи зараз робочий час;

– os – доступ до змінних середовища, шляхів та керування файлами конфігурацій.

Бібліотеки для роботи з базою даних, це sqlite3, яка є вбудованою базою даних, яка використовується для зберігання історії сенсорних даних, журналу override-дій та рівнів освітлення по зонах. Робота з базою реалізована у модулі db.py.

Бібліотеки для мессенджер-бота, це python-telegram-bot, який використовується для створення бот-інтерфейсу керування. Дозволяє реалізувати обробку команд, кнопок та надсилання графіків або поточних статусів системи (модуль bot.py).

Бібліотеки для побудови графіків, це matplotlib, яка використовується для побудови графіків температури або інших параметрів за день. Зображення формуються на основі вибірки з бази даних та надсилаються через бот.

Інші допоміжні бібліотеки:

– dotenv – для зчитування конфігураційного файлу .env (наприклад, API-токен бота, адреса API);

– logging – для реєстрації дій, помилок, втрати зв’язку з ESP тощо. Забезпечує зручне налагодження під час тестування та розробки;

– random, time – використовуються у симуляторах (slave_sim.py) для генерації випадкових даних і затримок.

Окрім сторонніх бібліотек, проєкт включає низку власних модулів:

– bot.py – реалізує інтерфейс користувача для віддаленого керування освітленням, вентиляцією та кондиціонером. Через нього користувач може отримувати поточні параметри з бази даних, переглядати статус системи, надсилати команди та будувати графіки (див. додаток Б);

– main.py – точка входу FastAPI-застосунку. Ініціалізує сервер, реєструє маршрути, запускає роботу API (див. додаток В);

– routes.py – реалізує API-ендпоїнти для взаємодії з ESP-симуляторами, базою даних і ботом. Представлено логіку обробки вхідних запитів, формування відповідей і механізми керування порогами (див. додаток В);

– master_sim.py, slave_sim.py – симулятори пристроїв ESP32. Представлено механізм зчитування даних, періодичної генерації показників навколишнього середовища, обміну запитами між вузлами та сервером, а також логіку імітації реле й сенсорів. (див. додаток Г);

– db.py – доступ до бази даних, запис та отримання даних (див. додаток Д);

– fallbackui.py – альтернатива боту при його тимчасовій дисфункції, реалізованого через FastAPI + HTML (див. додаток Е);

– threshold.py – збереження граничних значень (пороги температури, вологості, CO_2) (див. додаток Є).;

– config.py, work_schedule.json, api.env – конфігураційні файли (див. додаток Є).

2.7 Реалізація та тестування системи

У ході розробки програмного забезпечення була реалізована повноцінна система тестування з використанням емуляторів та графічних інтерфейсів. Тестування проводилося у середовищі PyCharm на платформі Windows з використанням віртуального середовища Python 3.9.

Симулятори master_sim.py та slave_sim.py імітують поведінку пристроїв ESP32. Slave-симулятор генерує дані з сенсорів (температура, вологість, рівень CO_2, освітлення) та періодично надсилає їх через HTTP-запити до master-сервера (рис. 2.13). Master-симулятор приймає ці запити через FastAPI та зберігає у локальну базу даних sensors.db.

Результат запуску slave-симулятора з випадковими даними:

“ Slave ESP запущено. Передача всіх даних до Master ESP...

Дані відправлено: {'timestamp': '2025-06-05T22:32:27.494233+00:00', 'temperature': 25.3, 'humidity': 85, 'co2': 607, 'zones': [{'zone': 1, 'light_level': 58}, {'zone': 2, 'light_level': 278}, {'zone': 3, 'light_level': 54}]} | Статус: 200”

Рисунок 2.13 – Запуск та прийняті master-сервером

Дані зберігаються у таблицях sensor_data та zone_data. Також створюється журнал override_log при ручному керуванні та загальна кількість записів зберігається у sqlite_sequence, що зображено на рисунку 2.14. Весь код для взаємодії з базою міститься у модулі db.py (функції insert_data, log_user_override). Дані які зчитуються з датчиків та дії які відбуваються автоматично відносно виконавчих пристроїв містяться у таблицях 2.5 та 2.6.

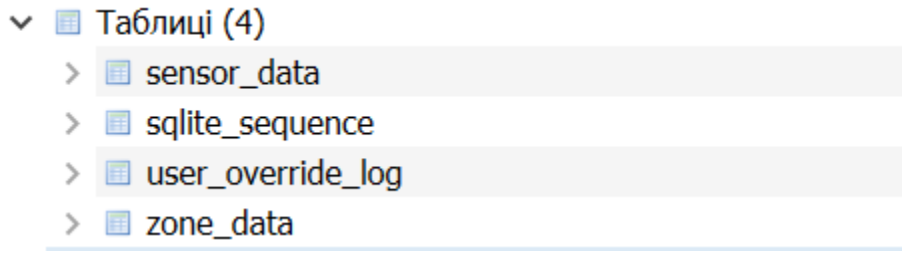


Рисунок 2.14 – Таблиці які створюються та поповнюються у SQLite

Таблиця 2.5 – Значення які зберігаються у sensor_data

Поле	Тип / Значення	Опис
id	Ціле число (int)	Унікальний ідентифікатор запису
timestamp	Дата й час ISO 8601	Час отримання даних від ESP
temperature	Float	Температура, °C
humidity	Float	Вологість повітря, %
co2	Ціле число	Рівень CO_2, ppm
relay_ventilation	Булеве (0 або 1)	Стан вентиляції: 1 – увімкнено, 0 – вимкнено
relay_aircon	Булеве (0 або 1)	Стан кондиціонера

Таблиця 2.6 – Значення які зберігаються у zone_data

Поле	Тип / Значення	Опис
id	Ціле число (int)	Унікальний ID запису
zone	Ціле число (1–3)	Номер зони освітлення
light_level	Ціле число (lx)	Поточний рівень освітлення з датчика
dimming_level	Ціле число (%)	Яскравість штучного освітлення (0–100%)
timestamp	Дата й час ISO 8601	Час фіксації даних

Дані які були реалізовані для зберігання ручних змін та величини бази даних, містяться у таблицях 2.7 та 2.8.

Таблиця 2.7 – Значення які зберігаються у user_override_log

Поле	Тип / Значення	Опис
id	Ціле число (int)	Унікальний ID запису
user_id	Ціле число (id)	Ідентифікатор користувача, що ініціював зміну
target	Текст	Назва цільового елемента (zone_1, aircon тощо)
action	Текст	Дія (наприклад, on, 50%)
duration_sec	Ціле число (секунди)	Тривалість впливу
timestamp	Дата й час ISO 8601	Час застосування команди

Таблиця 2.8 – Значення які зберігаються у sqlite_sequence

Поле	Пояснення
sensor_data	Поточне автоінкрементне значення id
zone_data	Номер останнього запису в zone_data
user_override_log	Номер останнього запису в override

Мессенджер-бот, який реалізовано у модулі bot.py, дозволяє протестувати віддалене керування. Через команди типу /start або кнопки меню (наприклад, «Дані з БД», «Статус») можна перевірити реальний стан системи. Бот надсилає запити до API, запуск якого продемонстровано на рисунку 2.15 (шляхом функцій fetch_full_status, handle_control_message) і формує відповіді для користувача (рис. 2.16).

```
PS C:\Users\User\PycharmProjects\diplom_project> python -m uvicorn main:app --reload
INFO:     Will watch for changes in these directories: ['C:\\Users\\User\\PycharmProjects\\diplom_project']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [3700] using StatReload
INFO:     Started server process [18520]
INFO:     Waiting for application startup.
INFO:root:🛠 Ініціалізація бази даних...
INFO:root:⛑ Запуск Telegram-бота...
INFO:     Application startup complete.
INFO:     127.0.0.1:60906 - "GET /api/status HTTP/1.1" 200 OK
INFO:     127.0.0.1:60917 - "POST /api/update HTTP/1.1" 200 OK
INFO:     127.0.0.1:60918 - "POST /api/update-zone HTTP/1.1" 200 OK
```

Рисунок 2.15 – Запуск головного FastApi який забезпечує зв’язок між ESP та користувачем

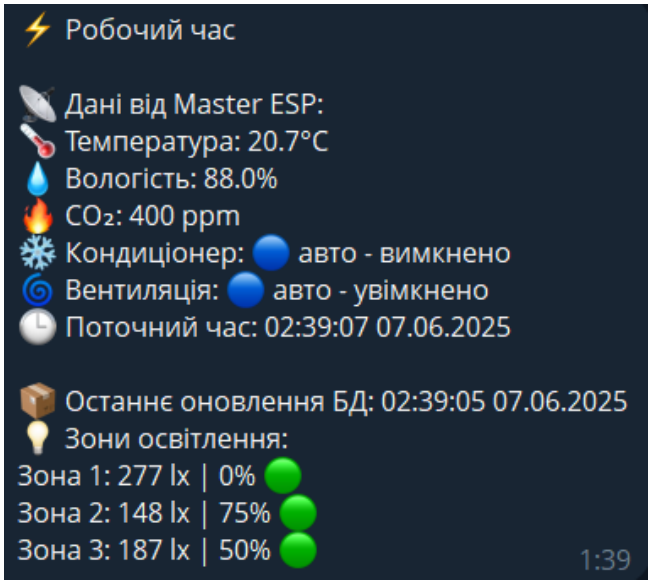


Рисунок 2.16 – Відображення статусу через Telegram-бота

Реалізовано перевірку зв’язку з ESP і автоматичне повідомлення про втрату з’єднання (рис. 2.17).

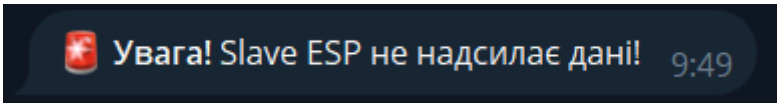
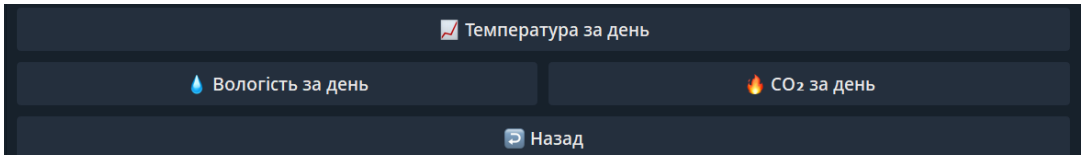


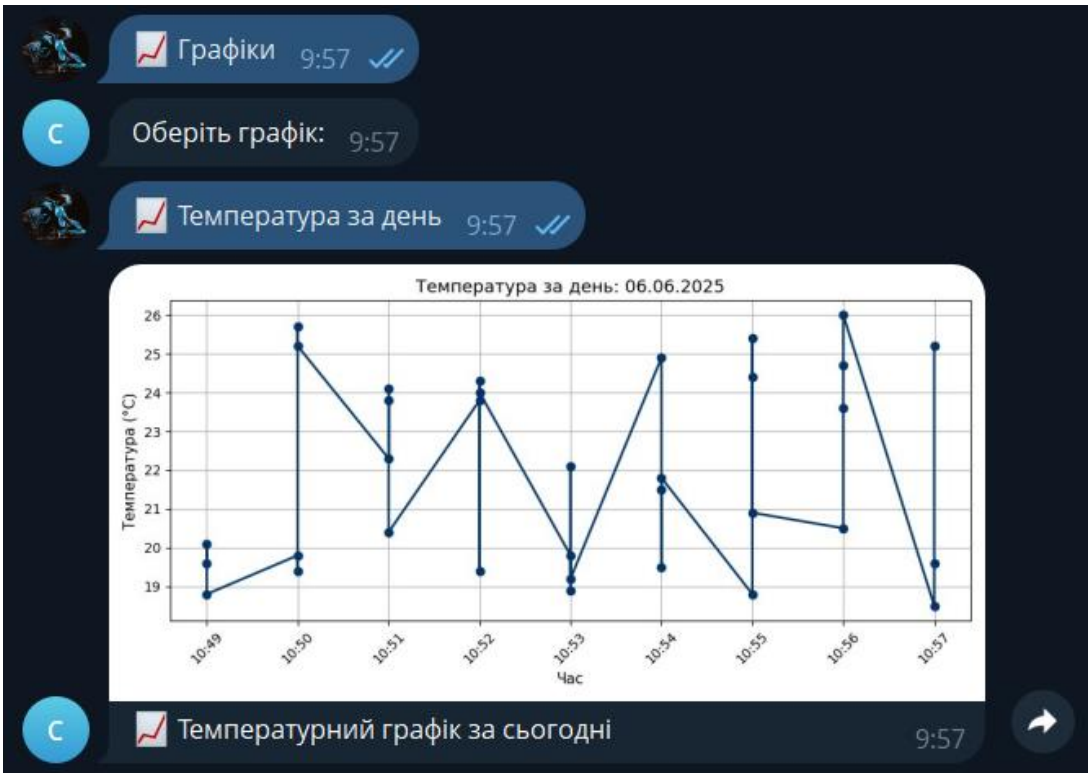
Рисунок 2.17 – Повідомлення про втрату з’єднання

На рисунку 2.18 показано обробку команди «Температура за день», яка формує запит до бази даних, обирає записи з таблиці sensor_data та виводить графік зміни температури протягом останньої доби.

Формування графіка відбувається за допомогою бібліотеки matplotlib, а функція temp_graph() автоматично фільтрує осі часу, форматує вивід та надсилає результат у вигляді зображення.



а)



б)

а) контексне меню графіків; б) реалізація графіків

Рисунок 2.18 – Меню вибору графіків та їх реалізація у вигляді діаграми

Контексне меню “Дані з БД” (рис. 2.19), має 3 параметри які може отримати користувач, рисунок 2.20 демонструє приклад роботи функції handle_db_message, яка читає останні 5 записів із таблиці sensor_data та форматує їх у зручний для користувача вигляд, видає максимальні та середні значення за добу (рис. 2.21).

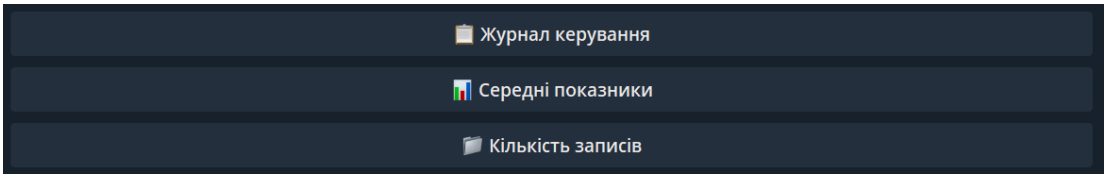


Рисунок 2.19 – Кнопка “Дані з БД”

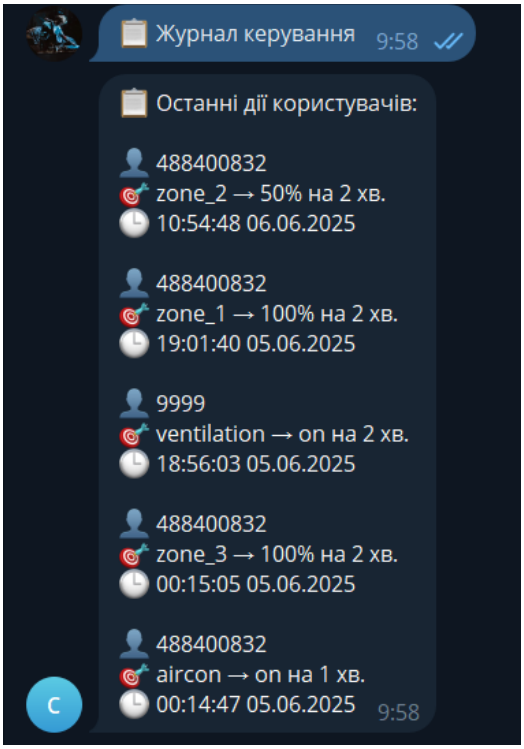


Рисунок 2.20 – Звіт журналу керування

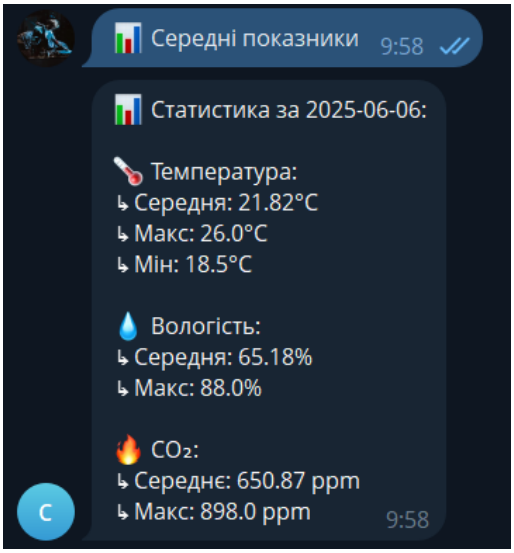


Рисунок 2.21 – Звіт середніх та максимальних показників

Реалізація відображення загальної кількості записів у БД наведено на рисунку 2.22.

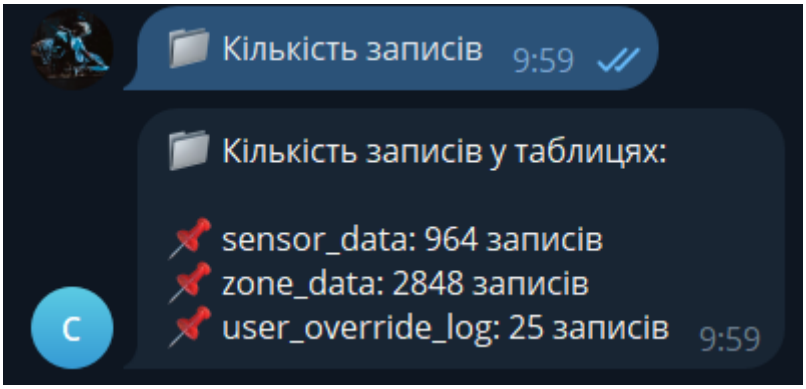


Рисунок 2.22 – Звіт з кількості записів у БД

Це підтверджує, що бот має доступ до бази даних, може читати свіжі значення та надсилати їх у відповідь на запит користувача.

Fallback UI (модуль fallback_ui.py) реалізує альтернативний графічний інтерфейс користувача через браузер. Його можна запустити на іншому порту (наприклад, 8010) для тестування керування без бота (рис. 2.23). Інтерфейс дозволяє вручну увімкнути вентиляцію або кондиціонер на певний час, а також задати рівень освітлення по кожній зоні (рис. 2.24).

```
PS C:\Users\User\PycharmProjects\diplom_project> python -m uvicorn fallback_ui:app --reload --port 8010
INFO:     Will watch for changes in these directories: ['C:\\Users\\User\\PycharmProjects\\diplom_project']
INFO:     Uvicorn running on http://127.0.0.1:8010 (Press CTRL+C to quit)
INFO:     Started reloader process [20456] using StatReload
INFO:     Started server process [24332]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

Рисунок 2.23 – Запуск FastApi для альтернативного керування у вигляді сайту

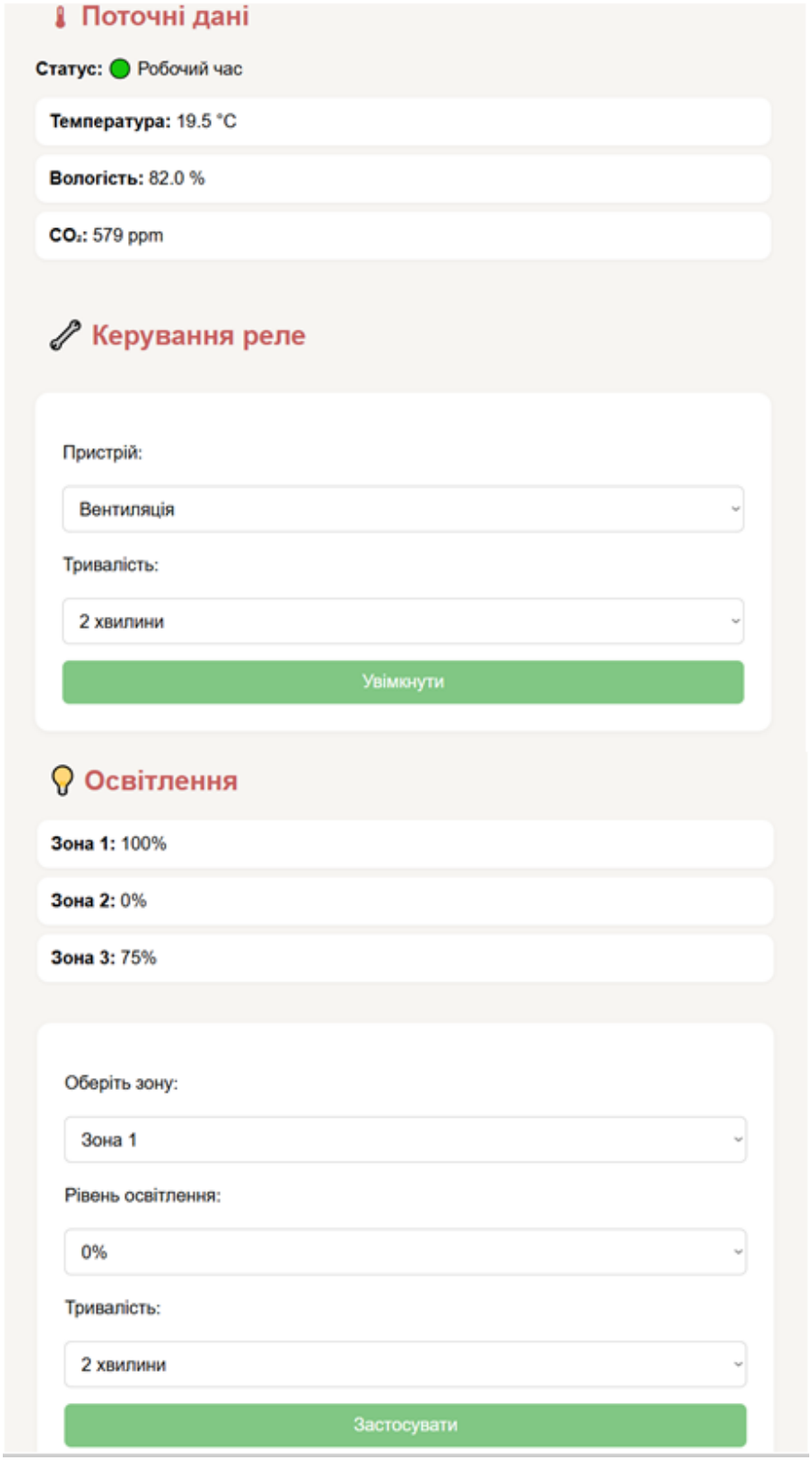


Рисунок 2.24 – Альтернативний локальний веб-інтерфейс, реалізований на основі FastAPI та HTML-відповідей з вбудованим CSS

Кожна дія з UI фіксується у базі даних. Наприклад, при надсиланні запиту через форму вебінтерфейсу, функція /light-control у fallback_ui.py звертається до API /set-light-level, що викликає log_user_override для збереження інформації.

У системі використано зовнішній конфігураційний файл .env, у якому зберігаються чутливі налаштування середовища, зокрема токен Telegram-бота. Це рішення дозволяє розділити код і конфігурацію, що сприяє підвищенню безпеки, зручності розгортання та модифікації системи. Зчитування змінних середовища відбувається автоматично через модуль config.py за допомогою бібліотеки python-dotenv.

Окремий конфігураційний файл work_schedule.json містить інформацію про робочий графік системи. У ньому задаються години початку й завершення робочого дня, а також перелік робочих днів тижня. Цей файл використовується у логіці функції is_work_time, яка визначає, чи дозволено автоматичне увімкнення освітлення та реле в поточний момент, рідше надсилаються дані з датчиків. За вимогами пори року, можно також додати реалізацію вентиляції кожні, наприклад, 30 хвилин, у літній сезон, та вмикання теплого режиму кондиціонування за годину до початку робочого дня, у зимовий період. Таким чином, реалізовано режим енергозбереження (рис. 2.25) шляхом блокування автоматичних сценаріїв у неробочий час, з можливістю ручного керування в обхід та збільшенням часового проміжку зчитування оновлень із мікроконтролерів.

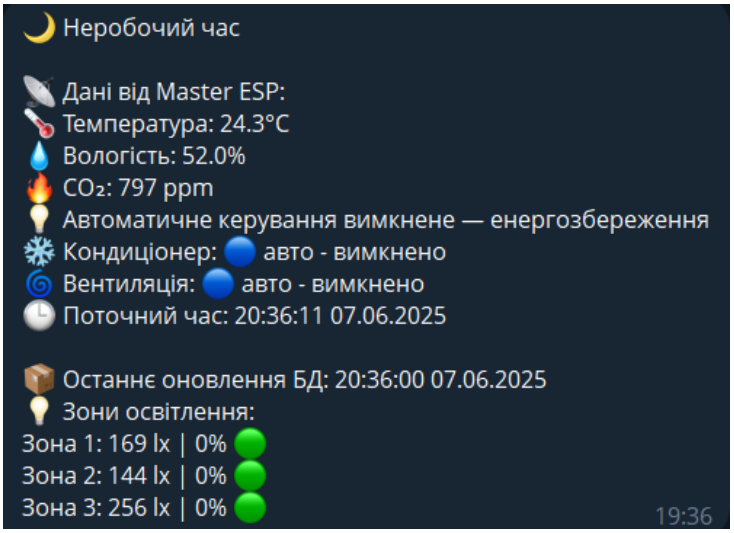


Рисунок 2.25 – Результат реалізації системи у неробочий час

Отже, проведене тестування дозволило перевірити коректність роботи всіх компонентів системи в інтегрованому середовищі, забезпечити відповідність логіки обробки даних, ручного керування та збереження історії. Наявність fallback інтерфейсу та симуляторів дозволяє демонструвати повну функціональність без підключення фізичних пристроїв.

3 РОЗРАХУНКИ ТА МОДЕЛЮВАННЯ ДИНАМІКИ СИСТЕМИ

3.1 Динамічна модель підсистеми освітлення та її перехідна характеристика

Для аналізу реакції освітлення на керуючий сигнал у вигляді широтно-імпульсної модуляції (ШІМ) було прийнято спрощену математичну модель, що описує поведінку освітлювального елемента (LED-лампи з димером) у вигляді аперіодичної ланки першого порядку:

$$G(s) = \frac{K}{(Ts + 1)}, \tag{3.1}$$

де K = 350 – статичний коефіцієнт підсилення (відображає максимальну освітленість, lux);

T = 2 с – стала часу, що визначає інерційність реакції лампи.

На вхід цієї ланки подається сигнал керування у вигляді ШІМ з амплітудою 1 (еквівалентно 100% яскравості). Для побудови перехідної характеристики в середовищі MATLAB була використана функція step(G).

Як видно з рисунку 3.1 та рисунку 3.2, при подачі одиничного імпульсу ШІМ (100% яскравості), освітленість досягає стаціонарного значення близько 350 lux із типовою експоненціальною затримкою. Час виходу на 95% рівня становить близько 6 секунд, що відповідає інерційній поведінці системи з T = 2 с.

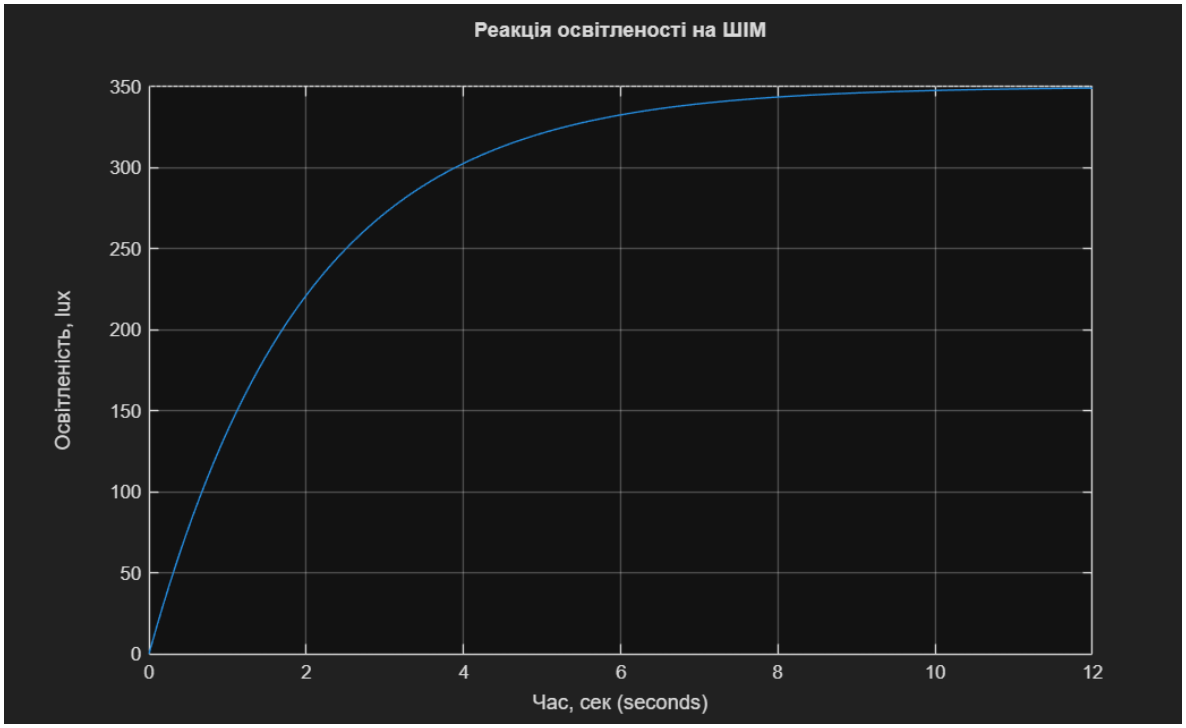


Рисунок 3.1 – Реакція освітленості на одиничний ШІМ-сигнал при моделюванні у фізичних одиницях (lux)

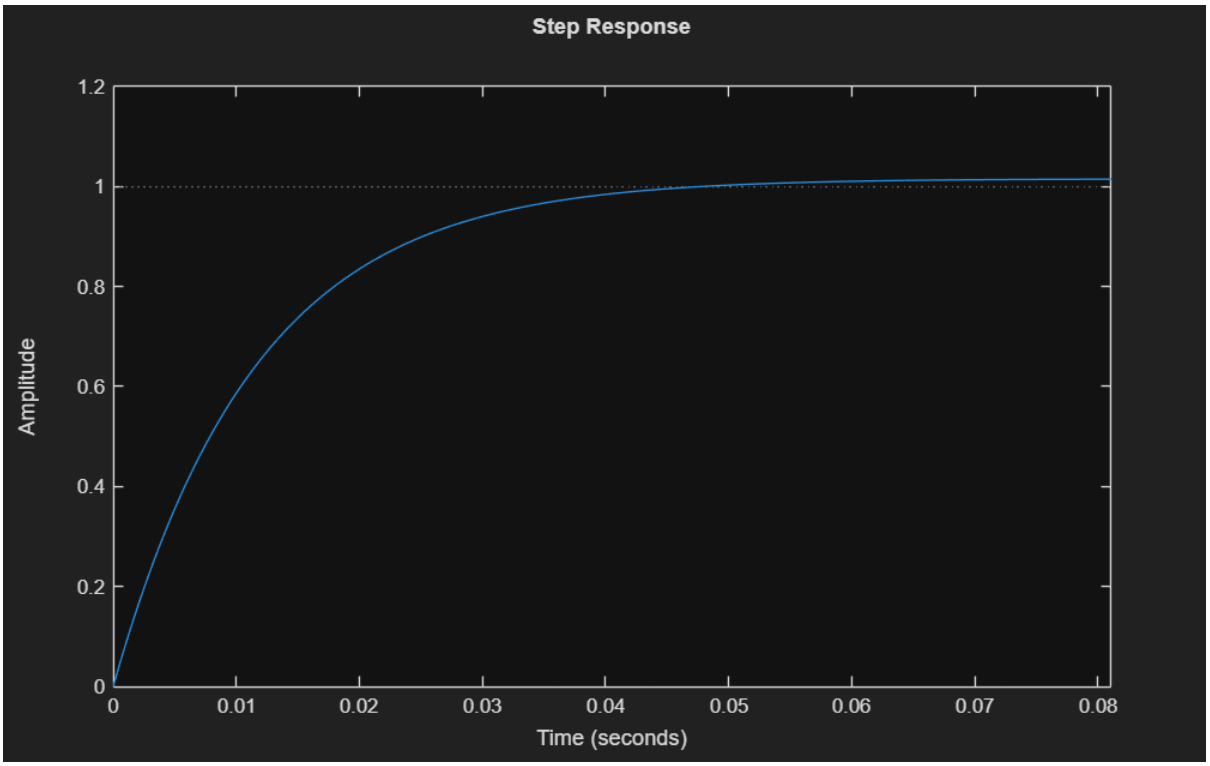


Рисунок 3.2 – Нормована перехідна характеристика освітлювальної ланки

Для перевірки стійкості передавальної функції G(s), яка має вигляд:

$$G(s) = \frac{350}{2s+1}, \quad (3.2)$$

Було проведено аналіз розташування її полюсів. Обчислення у середовищі MATLAB (pole(G)) показало, що полюс розташований у лівій півплощині комплексної площини:

pole(G) = -0.5,

Це означає, що система є стійкою. Було виконано логічну перевірку умов:

all(real(pole(G)) < 0)

ans = logical 1

Таким чином, математична модель освітлення є коректною для подальшого проєктування регулятора і включення до загальної моделі автоматизованої системи.

3.2 Аналіз динамічної відповіді та розрахунок сталої часу системи освітлення

З метою визначення параметрів динаміки системи освітлення була проведена симуляція перехідної характеристики в середовищі MATLAB. Модель системи, як і раніше, задається передавальною функцією (3.2). Для отримання числових показників динаміки було використано функцію stepinfo(G), яка надала наступні характеристики системи (табл 3.1):

Таблиця 3.1 – Характеристики перехідного процесу

Параметр	Значення	Опис
Час наростання	4.39 с	Час досягнення від 10% до 90% від сталого значення
Час перехідного процесу	7.82 с	Час досягнення стабільного рівня без суттєвих коливань
Час встановлення	7.82 с	Час, за який сигнал потрапляє і залишається в межах 2%
Мінімальне значення	316.58 lux	Нижня межа в період коливань
Максимальне значення	349.77 lux	Пік сигналу (наближено до сталого значення)
Перерегулювання	0%	Відсутнє – типово для аперіодичних систем
Час до максимуму	14.64 с	Час досягнення найвищого значення освітленості

На основі отриманих параметрів зроблено висновок, що система освітлення є інерційною, з достатньо швидкою реакцією на керуючий сигнал (вихід на 95% протягом 8 секунд), без перерегулювання.

3.3 Моделювання замкненої системи регулювання освітлення з ПІ-регулятором

З метою підвищення точності регулювання освітленості, у систему було впроваджено ПІ-регулятор. Такий тип регулятора поєднує у собі переваги пропорційного та інтегрального впливу, дозволяючи зменшити стійку похибку та забезпечити швидку реакцію системи без значного перерегулювання.

Передавальна функція ПІ-регулятора має вигляд:

$$C(s) = K_p + \frac{K_i}{s}, \tag{3.3}$$

де K_p – коефіцієнт пропорційної дії,

K_i – коефіцієнт інтегральної дії.

Разом із передавальною функцією освітлювальної ланки утворюється замкнена система, що описується наступним виразом:

$$W_{\text{замк}}(s) = \frac{C(s)*G(s)}{1+C(s)*G(s)}, \tag{3.4}$$

На рисунку 3.3 подано структурну модель замкненої системи регулювання освітлення, реалізовану в середовищі MATLAB Simulink. Вхідним сигналом є одиничний стрибок (Step), що відповідає бажаному рівню освітленості. Вихідна освітленість порівнюється із заданим значенням, і сформована похибка надходить на вхід ПІ-регулятора. Після цього сигнал передається на об'єкт регулювання – димер-лампа, що змодельована передавальною функцією першого порядку (3.2).

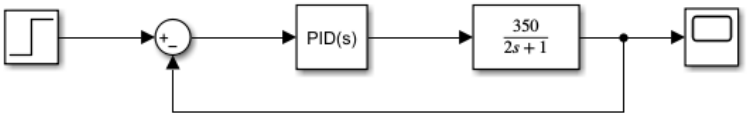


Рисунок 3.3 – Структурна схема моделі замкненої системи регулювання освітленості з ПІ-регулятором

Для регулятора обрано наступні коефіцієнти:

Kp = 0.6

Ki = 0.8

У моделі використовувались стандартні блоки: Step, Sum, PID Controller, Transfer Function, Scope.

На рисунку 3.4 показано реакцію системи згідно з результатами моделювання. Система характеризується високою швидкодією, відсутністю коливань та стійкою стабільністю виходу.

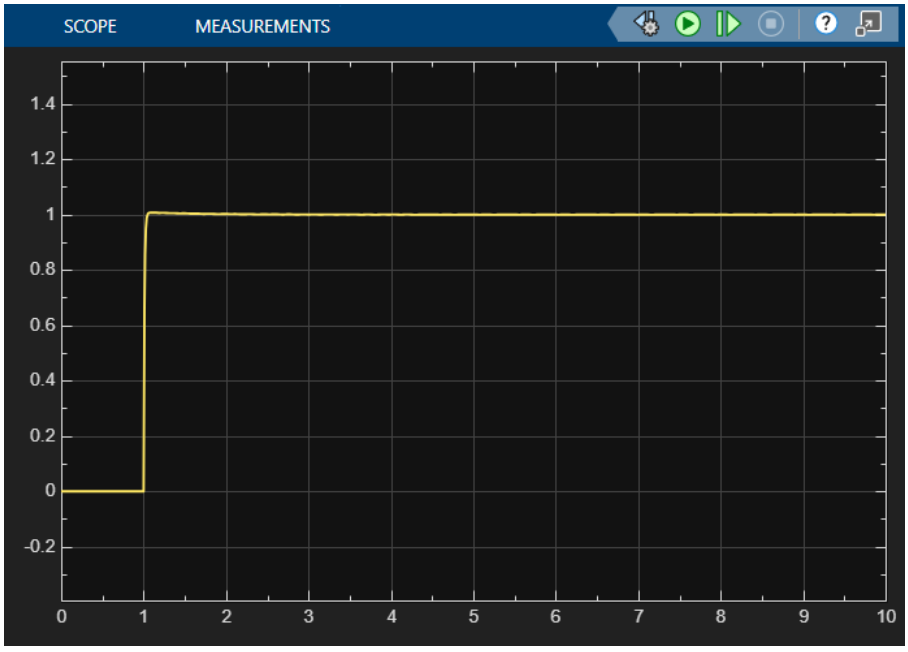


Рисунок 3.4 – Графік вихідного сигналу освітленості на ділянці регулювання

Згідно з аналізом динамічних характеристик реакції системи (рис. 3.5), отримано наступні числові параметри:

– час наростання: 0.01570.0157 с;
– перехідний час: 0.02770.0277 с;
– час встановлення: 0.02770.0277 с;
– мінімум у встановленому режимі: 0.90080.9008;
– максимум у встановленому режимі: 1.00021.0002;
– перерегулювання: 0.0211%;
– пік реакції: 1.00021.0002;
– час досягнення піку: 0.05230.0523 с.

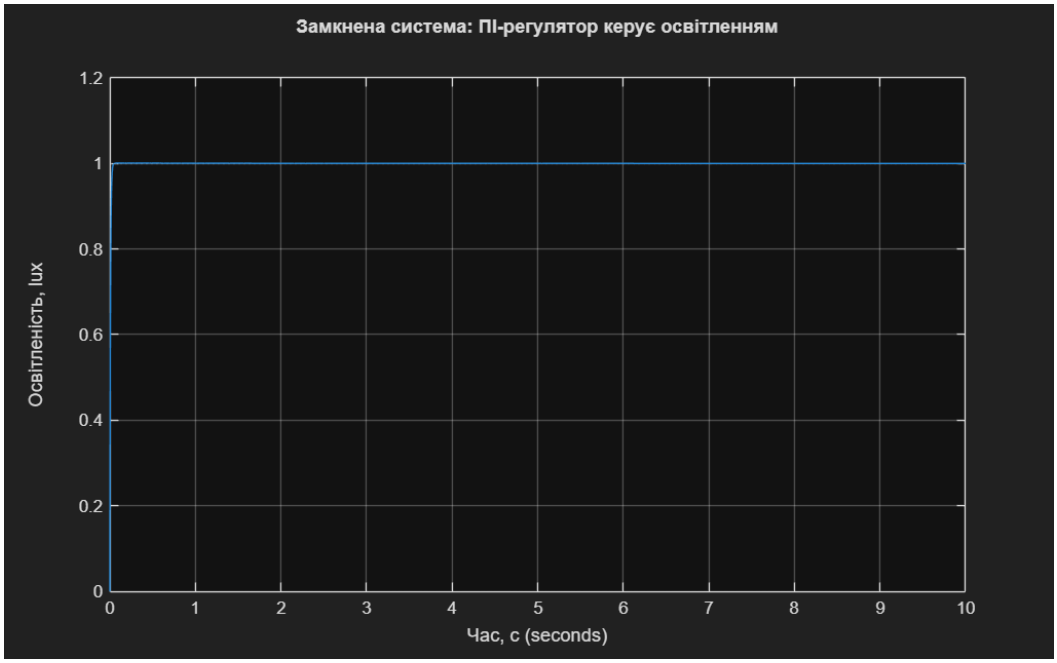


Рисунок 3.5 – Реакція освітленості замкненої системи на одиничний вхід з ПІ-регулятором

Таким чином, впровадження ПІ-регулятора забезпечило точне та швидке регулювання освітлення. Система демонструє ефективність та відповідає вимогам до інерційності у виробничих приміщеннях.

3.4 Перевірка системи на стійкість

Для перевірки системи на стійкість було використано критерій Найквіста. Згідно із рисунком 3.6, отриманим у програмному засобі MATLAB, видно, що система є стійкою, оскільки вона не перетинає точку $(-1;j0)$.

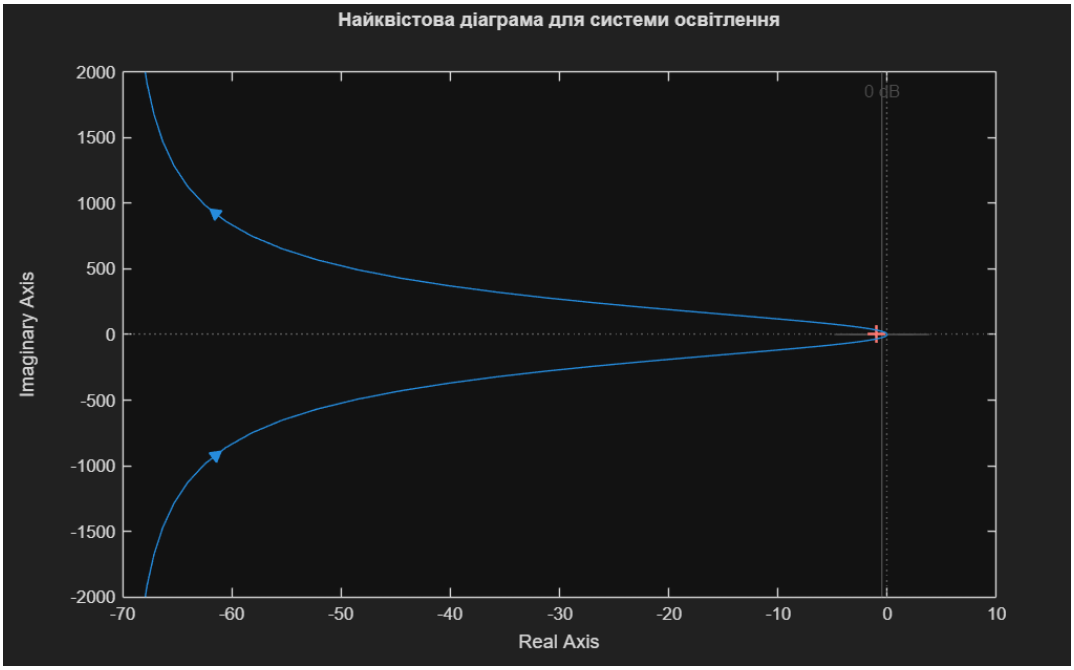


Рисунок 3.6 – Діаграма Найквіста

Таким чином, у програмному засобі MATLAB було отримано такі результати:

– передавальна функція освітлення змодельована як аперіодична ланка 1-го порядку з коефіцієнтом підсилення 350 та сталою часу 2 с. Реакція на одиничний вплив показала адекватну поведінку, що відповідає інерційним об'єктам;

– характеристики реакції системи без регулятора засвідчили стабільність, однак наявність великої сталої часу (до 7.8 с) свідчить про потребу в оптимізації регулювання;

– додавання ПІ-регулятора значно покращило динамічні показники: час наростання скоротився до 0.0157 с, а перерегулювання не перевищувало 0.02%. Система демонструє високу точність та швидкодію;

– аналіз стійкості за допомогою метода Найквіста, виявив що система є стійкою.

3.5 Охорона праці

Забезпечення охорони праці при проєктуванні та впровадженні систем автоматизованого керування мікрокліматом і освітленням є важливою складовою безпечної експлуатації виробничих приміщень.

Відповідно до Закону України «Про охорону праці» №2694-XII від 14.10.1992 (зі змінами) [35], роботодавець зобов’язаний забезпечити безпечні умови праці, зокрема через застосування засобів автоматизації, які дозволяють знизити вплив шкідливих факторів на працівників.

Розроблена система спрямована на усунення негативних впливів мікрокліматичних порушень, зокрема перегріву, надмірної вологості, недостатньої вентиляції та слабкого освітлення. Згідно з ДСанПіН 3.3.6.042–99 [36], температура повітря у виробничих зонах має становити 18–26 °C, вологість – 40–60 %, а концентрація вуглекислого газу – не перевищувати 1000 ppm. ДБН В.2.5-28:2018 [37] встановлює норматив освітленості для загального освітлення на рівні 300–500 лк.

У системі реалізовано автоматичний контроль вищевказаних параметрів, а також керування виконавчими пристроями на основі заданих порогів. Це знижує навантаження на персонал, мінімізує людський фактор і сприяє дотриманню санітарних норм у постійному режимі.

З боку електробезпеки, система містить елементи, що працюють під напругою 220 В (зокрема, реле, що комутують освітлення та вентиляцію).

Відповідно до НПАОП 40.1-1.32-01 «Правила безпечної експлуатації електроустановок споживачів» [38] передбачено:

– застосування понижувальних перетворювачів для живлення низьковольтних вузлів;
– захист струмопровідних частин;
– ізоляцію дротів;
– використання автоматичних вимикачів при перевантаженні;
– монтаж у вогнестійкому корпусі.

У частині пожежної безпеки передбачено встановлення захисних елементів (запобіжників, іскрозахищених реле), регулярну перевірку з’єднань та недопущення перевантаження ліній живлення. Всі дії узгоджено з вимогами НПАОП 0.00-5.18-07 «Пожежна безпека» [39].

Завдяки реалізації цих заходів система не лише забезпечує енергоефективність і автоматизацію, але й сприяє створенню безпечного виробничого середовища, зменшенню ризику професійних захворювань, отруєнь або електротравм.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було реалізовано програмно-апаратну модель системи автоматизованого керування освітленням та мікрокліматом у виробничих приміщеннях. Система орієнтована на роботу в умовах підприємств малого та середнього масштабу, де важливими є автономність, енергоефективність і можливість дистанційного контролю.

Проведено аналіз сучасних підходів до автоматизації виробничих процесів та обґрунтовано вибір відкритої платформи ESP32 як базової для реалізації. Для тестування системи без фізичного обладнання було використано онлайн-середовище Wokwi, яке дозволило спроєктувати логіку роботи мікроконтролерів, взаємодію з сенсорами та реле.

Розроблено симулятори вузлів Master і Slave ESP32 у середовищі PyCharm. Вони імітують зчитування температури, вологості, рівня CO2 та освітленості, а також передають ці дані на сервер. Серверна частина, реалізована на основі FastAPI, обробляє вхідні запити, формує рішення про керування пристроями та забезпечує взаємодію з мессенджер-ботом.

Мессенджер-бот реалізує зручний інтерфейс для користувача, дозволяє переглядати поточні показники, перемикати режими керування, будувати графіки та виконувати ручні команди. У випадку недоступності бота передбачено fallback веб-інтерфейс. Система підтримує логування дій користувача та контроль зв’язку між компонентами, включно з автоматичним сповіщенням про втрату з’єднання з вузлами.

Збереження всіх даних реалізовано через локальну базу SQLite. Алгоритми керування враховують не лише сенсорні показники, а й робочий графік підприємства. Це дозволяє обмежувати автоматичні дії в неробочий час, при цьому зберігаючи можливість ручного керування.

Проведено моделювання динаміки підсистеми освітлення в MATLAB, що підтвердило стійкість і контрольованість об'єкта за допомогою ПІ-регулятора. Аналіз за методом Найквіста вказує на коректність обраної структури керування.

Результати роботи демонструють, що створена система є функціонально завершеною, придатною до подальшої реалізації на реальному обладнанні та відповідає вимогам до гнучкості, масштабованості й простоти інтеграції в сучасне виробниче середовище.

Крім того, розроблена система прямо підтримує Ціль сталого розвитку 8 «Гідна праця та економічне зростання», а саме пункт 8.5, що передбачає сприяння забезпеченню надійних та безпечних умов праці, зокрема шляхом впровадження інноваційних технологій у сфері охорони праці та промислової безпеки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.

2. Методичні вказівки з підготовки кваліфікаційної роботи для здобувачів першого (бакалаврського) рівня вищої освіти денної і заочної форми навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І.Ш. Невлюдов, О.І. Филипенко, О.В. Токарєва, С.П. Новоселов, О.В Сичова. Харків: ХНУРЕ, 2023. 64 с.

3. Розроблення програмного забезпечення для автоматизованого керування освітленням та кліматом у виробничих приміщеннях. Формування сучасної науки: методика та практика: матеріали VII Всеукраїнської студентської наукової конференції. Молоканов К. А. Вінниця : ТОВ «УКРЛОГОС Груп», 2025. 907 с.

4. SmartDim. Розумне освітлення: як заощадити на електриці та створити затишок [Електронний ресурс] : Режим доступу: https://smartdim.com.ua/rozumne-osvitlennya-yak-zaoshhadyty-na-elektryczi-ta-stvoryty-zatyshok . (дата звернення: 13.04.2025).

5. Arduino Documentation [Електронний ресурс] : Режим доступу: https://docs.arduino.cc/. (дата звернення: 13.04.2025).

6. ESP8266 і ESP32 [Електронний ресурс] : Режим доступу: https://botland.com.pl/blog/esp8266-i-esp32-niepozorne-uklady-o-olbrzymich-mozliwosciach/. (дата звернення: 13.04.2025).

7. Raspberry Pi Documentation – Computers [Електронний ресурс] : Режим доступу: https://www.raspberrypi.com/documentation/computers/raspberry-pi.html. (дата звернення: 13.04.2025).

8. Control Panel Building and Machinery Wiring [Електронний ресурс] : Режим доступу: https://peganelectric.com/services/control-panel-building-and-machinery-wiring-industrial-electrical-solutions/. (дата звернення: 14.04.2025).

9. Energy-Efficient Electrical Solutions – LED Lighting [Електронний ресурс] : Режим доступу: https://peganelectric.com/services/energy-efficient-electrical-solutions-led-lighting-power-factor-upgrades/. (дата звернення: 14.04.2025).

10. Siemens AG. SIMATIC WinCC V7 – System Description. – Siemens, 2018. – 296 с.

11. Siemens – SIMATIC S7-1500 [Електронний ресурс] : Режим доступу: https://mall.industry.siemens.com/mall/sv/se/Catalog/Products/10007460. (дата звернення: 15.04.2025).

12. Ibrahim D. Programming with Node-RED. – Elektor International Media, 2021. – 325 с.

13. Weather analysis workflow on Node-RED [Електронний ресурс] : Режим доступу: https://www.researchgate.net/figure/Weather-analysis-workflow-on-Node-RED-Weather-data-from-the-DataMux-In-node-is-passed_fig2_361422645.
(дата звернення: 15.04.2025).

14. Home Assistant Temperature/Humidity [Електронний ресурс] : Режим доступу: https://www.youtube.com/watch?v=RUWpeTXsQ_U. (дата звернення: 17.04.2025).

15. Blynk Android App [Електронний ресурс] : Режим доступу: https://github.com/BlynkMobile/Blynk-Android-App?tab=readme-ov-file. (дата звернення: 17.04.2025).

16. Візуалізація систем моніторингу [Електронний ресурс] : Режим доступу: https://www.flickr.com/photos/xmodulo/16909119337. (дата звернення: 18.04.2025).

17. InfluxData Documentation – InfluxDB [Електронний ресурс] : Режим доступу: https://docs.influxdata.com/influxdb/v2.7/ (дата звернення: 18.04.2025).

18. Моніторинг NAS через Telegraf, InfluxDB та Grafana [Електронний ресурс] : Режим доступу: https://www.redeszone.net/marcas/qnap/instalar-configurar-telegraf-influxdbv2-grafana-monitorizar-nas-qnap/. (дата звернення: 18.04.2025).

19. Firebase Realtime Database – Documentation [Електронний ресурс] : Режим доступу: https://firebase.google.com/docs/database . (дата звернення: 18.04.2025).

20. Що таке Grafana [Електронний ресурс] : Режим доступу: https://itfb.com.ua/uk/shho-take-grafana/ . (дата звернення: 18.04.2025).

21. Датчик температури і вологості DHT22 [Електронний ресурс]: Режим доступу: https://www.mini-tech.com.ua/datchik-temperatury-i-vlazhnosti-dht22. (дата звернення: 07.05.2025).

22. Датчик температури, вологості та тиску BME280 [Електронний ресурс]: Режим доступу: https://www.mini-tech.com.ua/ua/barometr-termometr-gigrometr-bme280. (дата звернення: 07.05.2025).

23. Датчик температури і вологості AM2320 [Електронний ресурс]: Режим доступу: https://3v3.com.ua/product_7894.html. (дата звернення: 08.05.2025).

24. Датчик газу MQ-135 [Електронний ресурс]: Режим доступу: https://www.mini-tech.com.ua/ua/datchik-gaza-mq-135-modul. (дата звернення: 08.05.2025).

25. CO_2 сенсор MH-Z19B [Електронний ресурс]: Режим доступу: https://www.mini-tech.com.ua/ua/datchik-co2. (дата звернення: 09.05.2025).

26. CO_2 модуль SCD30 [Електронний ресурс]: Режим доступу: https://www.rcscomponents.kiev.ua/product/scd30_171519.html. (дата звернення: 09.05.2025).

27. Датчик освітленості TEMT6000 [Електронний ресурс]: Режим доступу: https://arduino.ua/prod2560-datchik-osveshhennosti-temt6000-dlya ardyino?srsltid=AfmBOopROqolfC3uLCosfXOQZjw1MoxQx-lsnpCujP9X7h4jBNptIxCl. (дата звернення: 09.05.2025).

28. Датчик освітленості BH1750FVI [Електронний ресурс]: Режим доступу: https://arduino.ua/prod1116-datchik-osveshhennosti-cifrovoi-bh1750fvi. (дата звернення: 09.05.2025).

29. AC Dimmer Module RobotDyn [Електронний ресурс]: Режим доступу: https://miniboard.com.ua/modules/722-dimmer-peremennogo-toka-robotdyn.html?srsltid=AfmBOorAUxOXXi06cyyBtIqJyzkaBTOJsuVQtM_65MkTuPyaCIPGmAkn. (дата звернення: 10.05.2025).

30. Датчик протікання води Neptun SW007 [Електронний ресурс]: Режим доступу: https://teploradost.com.ua/datchik-kontrolya-protechki-vody-neptun-sw007-provodnoj. (дата звернення: 10.05.2025).

31. SEN18 Water Detection Sensor [Електронний ресурс]: Режим доступу: https://www.amazon.in/Robodo-Electronics-SEN18-Detection-Arduino/dp/B0787HGY19?th=1. (дата звернення: 10.05.2025).

32. Датчик диму MQ-2 [Електронний ресурс]: Режим доступу: https://arduino.ua/prod298-modyl-datchika-dima-mq2?srsltid=AfmBOoqyACn2Kb9gnFt0gqXfApEIZN3qF7rKbNlhoO9E8_hO33ojTA2-. (дата звернення: 11.05.2025).

33. Механічне реле SRD-05VDC-SL-C [Електронний ресурс]: Режим доступу: https://www.mini-tech.com.ua/srd-05vdc-sl-c. (дата звернення: 11.05.2025).

34. Твердотільне реле SSR-40 DA [Електронний ресурс]: Режим доступу: https://www.mini-tech.com.ua/ua/ssr-40-da-tverdotelnoe-rele. (дата звернення: 11.05.2025).

35. Закон України «Про охорону праці» №2694-XII від 14.10.1992 (зі змінами) [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/2694-12#Text (дата звернення: 14.06.2025).

36. ДСанПіН 3.3.6.042–99 Санітарні норми мікроклімату виробничих приміщень [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/rada/show/va042282-99#Text, (дата звернення:

14.06.2025).

37. ДБН В.2.5-28:2018 Природне і штучне освітлення [Електронний ресурс]. – Режим доступу: https://ledeffect.com.ua/images/__branding/dbn2018.pdf, (дата звернення: 14.06.2025).

38. НПАОП 40.1-1.32-01. Правила безпечної експлуатації електроустановок споживачів [Електронний ресурс]. – Режим доступу: https://dbn.co.ua/load/normativy/npaop/npaop_40_1_1_21_98/23-1-0-821 (дата звернення: 14.06.2025).

39. НПАОП 0.00-5.18-07. Пожежна безпека. Загальні вимоги [Електронний ресурс]. – Режим доступу: https://dnaop.com/html/55207/doc-derzhavnij-rejestrnormativnih-aktiv-z-pitany-pozhezhnoji-bezpeki (дата звернення: 14.06.2025).

40. Кабінет Міністрів України. Національна доповідь з імплементації Цілей сталого розвитку в Україні [Електронний ресурс]. – Режим доступу: https://www.kmu.gov.ua/storage/app/sites/1/natsionalna-dopovid-csr-Ukrainy.pdf (дата звернення: 15.06.2025).