МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА

Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»

Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
«Інформаційні управляючі системи і технології»

галузь знань	12 Інформаційні технології
спеціальність	122 Комп'ютерні науки

Форма навчання: очна (денна) *або* заочна

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему: «Розроблення автоматизованої системи кліматичного контролю для теплиць на платформі Arduino»

здобувача Датка Андрія Петровича
(ПІБ, підпис)

Науковий керівник: д.е.н., проф. Ріппа С.П.
(науковий ступінь, учене звання, ПІБ)

_____
(підпис)

Робота допущена до захисту перед екзаменаційною комісією з атестації здобувачів вищої освіти (ЕК)

Завідувач кафедри: _____
(науковий ступінь, учене звання, ПІБ)

_____
(підпис)

Київ 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
«Інформаційні управляючі системи і технології»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО: Керівник проєктної групи (гарант) освітньо-професійної програми	ЗАТВЕРДЖУЮ: Завідувач кафедри інформаційних систем в економіці
_____ Устенко С.В. «____» _____2025 р.	_____ Тішков Б.О. «____» _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти ***_Датка Андрія Петровича_***
очної (денної) форми навчання

на підготовку кваліфікаційної магістерської роботи
на тему: «Розроблення автоматизованої системи кліматичного контролю для теплиць на платформі Arduino»

Тему затверджено наказом ректора Університету від «20» лютого 2025 р. № **332-ст**
Кваліфікаційна магістерська робота виконується на матеріалах
науково-технічних публікацій та Інтернету

План кваліфікаційної магістерської роботи

Розділ І _Дослідження та аналіз підходів до створення ІУС з контролю мікроклімату теплиці_

Розділ ІІ _Характеристика ІУС з контролю мікроклімату теплиці та методи і моделі_

Розділ ІІІ _Розроблення проєктних рішень та їх реалізація._

Об'єкт дослідження: _процеси автоматизованого моніторингу та регулювання мікрокліматичних параметрів у тепличних господарствах._

Предмет дослідження: _сукупність теоретичних, методичних і практичних підходів до проєктування та реалізації автоматизованої системи кліматичного контролю на основі платформи Arduino._

Мета кваліфікаційної магістерської роботи: _Основною метою кваліфікаційної магістерської роботи є підвищення ефективності управління мікрокліматом у теплицях шляхом розроблення автоматизованої інформаційної системи на базі апаратно-програмної платформи Arduino.___

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі I _Дослідити основні характеристики тепличних господарств та проблем кліматичного контролю, проаналізувати існуючи рішення і підходи до побудови ІУС, орієнтованих на кліматичні господарства, обґрунтувати вибір технології для побудови кліматичної ІУС._

_____

_____

У розділі II _Розробити структуру схему ІУС кліматичного контролю (ІУС КК), визначити її основні елементи принципової електричної схеми і алгоритмічне забезпечення кліматичного контролю. Описати методи та моделі управління мікрокліматом._

_____

У розділі III _Представити опис апаратного і програмного забезпечення ІУС КК, в т.ч. з можливостями підтримки інтеграції з хмарними платформами для передавання і збереження телеметричних даних для віддаленого моніторингу._

_____

_____

Завдання підготував
науковий керівник _____ **Ріппа Сергій Петрович**
(підпис) *(ініціали, прізвище)*

«____» _____ 202_ р.

Завдання одержав
здобувач _____ **Датко Андрій Петрович**
(підпис) *(ініціали, прізвище)*

«____» _____ 202_ р.

РЕЦЕНЗІЯ
на кваліфікаційну магістерську роботу
здобувача вищої освіти другого (магістерського) рівня
Датка Андрія Петровича
на тему: «РОЗРОБЛЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЛІМАТИЧНОГО КОНТРОЛЮ ДЛЯ ТЕПЛИЦЬ НА ПЛАТФОРМІ ARDUINO»
за спеціальністю 122 Комп'ютерні науки

Актуальність теми дослідження: Обрана для захисту тематика кваліфікаційної магістерської роботи є достатньо актуальною, адже сфера еко-агро-бізнесу відіграє важливу роль в економіці України, а в сьогоднішніх умовах воєнного стану та економічної кризи дозволяє вирішувати проблеми вдосконалення і розвитку тепличних господарств широкого спектру на основі Інтернету речей.

Наукова новизна: обґрунтувано підходи до забезпечення оптимального мікроклімату в теплицях та запропоновано реалізацію за допомогою технологій Інтернету речей на основі створення відповідних мікроконтролерних платформ (Arduino, ESP32) у поєднанні з цифровими сенсорами та хмарними сервісами.

Якість проведеного аналізу проблеми, текстового та табличного матеріалу. У роботі проаналізовано сучасний стан та тенденції розвитку галузі тепличного клімат-контролю, запропоновано розподіл функцій тепличної системи між незалежними логічними модулями, кожен з яких відповідає за підтримку окремого параметра мікроклімату..Представлені таблиці та рисунки доповнюють основний зміст роботи.

Обґрунтованість пропозицій і висновків та їх практична цінність. Обґрунтовано використання апаратури на базі мікроконтролера ESP32, з урахувнням його високої продуктивності, вбудованого Wi-Fi-модуля та сумісності з екосистемою Arduino. Усе це дало змогу, забезпечити гнучкість і масштабованість мікроконтролерної клімат-системи, спрощує її подальшу модернізацію, а також підвищує відмовостійкість рішень, оскільки вихід з ладу одного компонента не паралізує роботу всієї системи.

Наявність недоліків: незбалансовані в плані масштабування рисунки налаштування каналів і результати роботи системи в хмарі (рис. 3.3.1; рис. 3.3.2), а також незначні відхилення від вимог до форматування тексту та переліку використаних джерел.

Загальний висновок і оцінка роботи: Кваліфікаційна робота на тему «РОЗРОБЛЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЛІМАТИЧНОГО КОНТРОЛЮ ДЛЯ ТЕПЛИЦЬ НА ПЛАТФОРМІ ARDUINO» у цілому відповідає встановленим вимогам МОН щодо структури, обсягу, змісту та рекомендується до захисту з високою оцінкою, а її автор, Датко Андрій Петрович, заслуговує на присвоєння освітньо-кваліфікаційного рівня «магістр» за спеціальністю 122 Комп'ютерні науки.

Рецензент: _____ (підпис)
Погорєловська І.Д., к.е.н., доцент,
доцент кафедри комп'ютерних та інформаційних технологій і систем факультету фінансів та цифрових технологій Державного податкового університету

«19» травня 2025 р.

Відгук
на кваліфікаційну магістерську роботу
здобувача вищої освіти КНЕУ
Датка Андрія Петровича на тему:
«РОЗРОБЛЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЛІМАТИЧНОГО КОНТРОЛЮ ДЛЯ ТЕПЛИЦЬ НА ПЛАТФОРМІ ARDUINO»

Актуальність теми дослідження. Обрана здобувачем тематика кваліфікаційної магістерської роботи є безперечно актуальною, оскільки присвячена аналізу, дослідженню та практичному застосуванню сучасних ІУС в області автоматизованого моніторингу та регулювання мікрокліматичних параметрів у тепличних господарствах на базі засобів IoT.

Наукова новизна. Представлений до захисту проєкт містить елементи наукової новизни, оскільки автор розвиває ідеологію створення і запровадження інтелектуальних ІТ-сервісів в середовищі IoT для реалізації автоматизованої системи кліматичного контролю, аналізує і досліджує актуальний стан цієї сфери і представляє ІУС на базі апаратно-програмної платформи Arduino, орієнтованої на моніторинг і управління мікрокліматом у теплицях.

Якість проведеного аналізу проблеми, текстового та табличного матеріалу. Кваліфікаційна магістерська робота містить досить обширну аналітику існуючих інформаційних систем кліматичного контролю для теплиць і надалі автор здійснює вибір підходів, архітектури та технологій для створення ІУС кліматичного контролю з функціями інтелектуального моніторингу і управління функціями ІУС.

Обґрунтованість пропозицій і висновків та їх практична цінність. Підготовлені в роботі пропозиції використання інтелектуальних інструментів моніторингу і управління кліматичним контролем є достатньо обґрунтованими; висновки, зроблені здобувачем за результатами виконаних досліджень, узагальнюють основний зміст роботи. Результати виконання магістерського проєкту можуть бути використані при розробці відповідних веб-ресурсів, орієнтованих на розвиток платформ IoT.

Наявність недоліків. Суттєвих зауважень до виконаного проєкту не має.

Загальний висновок і оцінка роботи. Кваліфікаційна магістерська робота на тему «РОЗРОБЛЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЛІМАТИЧНОГО КОНТРОЛЮ ДЛЯ ТЕПЛИЦЬ НА ПЛАТФОРМІ ARDUINO» у цілому відповідає встановленим вимогам вищої школи та рекомендується до захисту з високою оцінкою, а її автор *Датко Андрій Петрович* заслуговує на присвоєння освітньо-кваліфікаційного рівня «магістр» за спеціальністю 122 «Комп'ютерні науки» освітньо-професійної програми «Інформаційні управляючі системи і технології».

Науковий керівник_____(підпис, М.П.) С. П. Ріппа, д.е.н., професор,
професор кафедри інформаційних систем в економіці
Навчально-наукового інституту "Інститут
інформаційних технологій в економіці" КНЕУ

«17» _травня_ 2025

Реферат

Кваліфікаційна магістерська робота містить 89 сторінок, 2 таблиці, 28 рисунків, список використаних джерел з 52 найменувань, додатки.

«Розроблення автоматизованої системи кліматичного контролю для теплиць на платформі Arduino»

Об'єктом дослідження кваліфікаційної магістерської роботи є процеси автоматизованого моніторингу та регулювання мікрокліматичних параметрів у тепличних господарствах.

Предметом дослідження є сукупність теоретичних, методичних і практичних підходів до проєктування та реалізації автоматизованої системи кліматичного контролю на основі платформи Arduino.

Мета і завдання дослідження. Основною метою кваліфікаційної магістерської роботи є підвищення ефективності управління мікрокліматом у теплицях шляхом розроблення автоматизованої інформаційної системи на базі апаратно-програмної платформи Arduino.

Відповідно до поставленої мети визначені такі завдання:

− дослідити предметну область, проаналізувати існуючі інформаційні системи кліматичного контролю для теплиць;

− обґрунтувати вибір підходів, архітектури та технологій для створення автоматизованої системи;

− визначити структуру та представити характеристику розроблюваної автоматизованої системи;

− описати методи, моделі та алгоритми управління мікрокліматом у теплицях;

− спроєктувати структурну та принципову електричну схему системи;

− реалізувати апаратно-програмний комплекс на базі Arduino UNO / ESP32;

− здійснити інтеграцію системи з хмарною платформою для моніторингу (ThingSpeak);

− провести тестування функціонування системи в середовищі моделювання та надати рекомендації з впровадження в агропромисловість.

Теоретична, методична та практична значущість отриманих результатів. У процесі дослідження проведено детальний аналіз сучасного стану автоматизації тепличного клімат-контролю, охарактеризовано переваги використання платформи Arduino для побудови доступних інтелектуальних систем, запропоновано ефективну структуру кліматичного регулювання з урахуванням реальних аграрних потреб. Практичні результати дослідження полягають у створенні повноцінного прототипу системи, здатного зчитувати дані з датчиків, керувати мікрокліматом у режимі реального часу, а також передавати дані на хмарну платформу для візуалізації та віддаленого доступу. Впровадження такої системи сприятиме зменшенню енергетичних витрат, підвищенню врожайності та стабільності сільськогосподарського виробництва.

Рік виконання кваліфікаційної магістерської роботи – 2025.

Рік захисту роботи – 2025.

Ключові слова: теплиця, Arduino, автоматизація, клімат-контроль, мікроконтролер, датчики, інформаційна система.

Зміст

ВСТУП 3

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ІУС З КОНТРОЛЮ МІКРОКЛІМАТУ ТЕПЛИЦІ 6

1.1. Загальна характеристика тепличного господарства та проблем кліматичного контролю 6

1.2. Аналіз існуючих рішень 8

1.3. Підходи до побудови інформаційних управляючих систем для тепличного господарства 13

1.4. Обґрунтування вибору технологій 18

1.5. Висновки до розділу 20

РОЗДІЛ 2. ХАРАКТЕРИСТИКА ІУС З КОНТОЛЮ МІКРОКЛІМАТУ ТЕПЛИЦІ ТА МЕТОДИ І МОДЕЛІ 22

2.1. Структура, методи та моделі інформаційної управляючої системи кліматичного контролю 22

2.3. Структурна схема системи 25

2.4. Принципова електрична схема системи 28

2.5. Алгоритм роботи системи 31

2.6. Методи автоматичного управління мікрокліматом 33

2.7. Математична модель регулювання 36

2.8. Висновки до розділу 39

РОЗДІЛ 3. РОЗРОБЛЕННЯ ПРОЄКТНИХ РІШЕНЬ ТА ЇХ РЕАЛІЗАЦІЯ 41

3.1. Вибір та опис апаратного забезпечення 41

3.2. Розробка програмного забезпечення 47

3.3. Інтеграція з хмарними платформами 51

3.4. Веб-моніторинг 53

3.5. Тестування та симуляція 56

3.6. Безпека, надійність і обмеження системи 58

3.7. Висновки до розділу 60

ВИСНОВКИ 62

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 68

ДОДАТКИ 74

Додаток А 74

Додаток Б 81

Додаток В 82

Додаток Г 83

Додаток Д 84

ВСТУП

У сучасному аграрному виробництві одним із ключових факторів досягнення стабільних урожаїв є створення оптимального мікроклімату в теплицях. Саме тепличне вирощування дає змогу забезпечити стабільність умов розвитку рослин незалежно від кліматичних змін у навколишньому середовищі. Для підтримання таких умов необхідно забезпечити ефективне управління температурою повітря, вологістю, рівнем вуглекислого газу, освітленістю та іншими параметрами. Нерегульовані коливання цих показників можуть призвести до значних втрат врожаю, зниження якості продукції та перевитрат енергетичних і водних ресурсів. У багатьох малих та середніх тепличних господарствах контроль кліматичних умов досі здійснюється вручну, що пов’язано з високими затратами часу, залежністю від людського фактору та низькою точністю регулювання. Автоматизація процесу моніторингу та управління мікрокліматом є важливим кроком у напрямку підвищення ефективності сільськогосподарського виробництва. Вона дозволяє мінімізувати втручання людини, забезпечити безперервний контроль та адаптивне регулювання умов у реальному часі.

Завдяки розвитку технологій Інтернету речей (IoT), поява доступних мікроконтролерних платформ, таких як Arduino та ESP32, відкриває широкі можливості для створення автоматизованих кліматичних систем навіть у малобюджетних господарствах. У поєднанні з сучасними сенсорами, такими як DHT22, BH1750, MQ135, а також із хмарними сервісами, зокрема ThingSpeak, можна побудувати гнучку, адаптивну та масштабовану систему моніторингу й керування мікрокліматом.

Актуальність теми зумовлена необхідністю створення доступної, ефективної та надійної автоматизованої системи кліматичного контролю для теплиць, яка підвищить урожайність, зменшить споживання енергії та води, а також забезпечить стабільні умови розвитку рослин.

Об'єктом дослідження є процеси автоматизованого моніторингу та регулювання мікрокліматичних параметрів у тепличних господарствах.

Предметом дослідження є сукупність теоретичних, методичних і практичних підходів до проєктування та реалізації автоматизованої системи кліматичного контролю на основі платформи Arduino.

Метою роботи є розроблення автоматизованої системи контролю мікроклімату теплиці, яка здатна в реальному часі вимірювати та регулювати температуру, вологість, рівень вуглекислого газу, а також передавати дані на хмарну платформу для віддаленого моніторингу.

Для досягнення поставленої мети визначені наступні завдання дослідження:

- провести дослідження предметної області та аналіз існуючих ІУС у сфері тепличного кліматичного контролю;
- обґрунтувати вибір технологічної платформи, сенсорів і підходів до створення автоматизованої системи;
- спроєктувати структуру системи управління мікрокліматом;
- описати методи, моделі та алгоритми регулювання кліматичних параметрів;
- розробити апаратну частину на основі Arduino / ESP32;
- реалізувати програмне забезпечення системи;
- здійснити інтеграцію з хмарною платформою для збору та візуалізації даних;
- протестувати систему в умовах симуляції (Wokwi) та оцінити її ефективність;
- сформулювати рекомендації щодо впровадження у виробничі умови.

Методи дослідження, що використовувалися в роботі:

- аналітичні – для дослідження предметної області та огляду існуючих рішень;
- структурно-функціональний підхід – для побудови архітектури системи;
- експериментальні – для перевірки працездатності системи в середовищі
- моделювання;
- методи логічного моделювання – для розробки алгоритмів автоматичного управління.

Практична значущість дослідження полягає в створенні доступного програмно-апаратного рішення, яке може бути впроваджене у тепличне господарство будь-якого масштабу з метою підвищення ефективності

вирощування сільськогосподарських культур, зниження енергетичних витрат і мінімізації людського втручання.

Структура кваліфікаційної магістерської роботи включає вступ, три основні розділи, висновки, список використаних джерел та додатки. У першому розділі здійснено аналіз предметної області та надано характеристику існуючих інформаційних систем у галузі. У другому розділі розроблено структурну модель системи, описано методи і моделі регулювання мікроклімату. Третій розділ присвячено реалізації системи, включаючи апаратну частину, програмне забезпечення, хмарну інтеграцію та тестування.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ІУС З КОНТРОЛЮ МІКРОКЛІМАТУ ТЕПЛИЦІ

1.1. Загальна характеристика тепличного господарства та проблем кліматичного контролю

Предметною областю дослідження є автоматизація кліматичного контролю в теплицях. Теплиці – це штучно створені середовища, які дозволяють вирощувати рослини в умовах, близьких до ідеальних, незалежно від кліматичних умов зовнішнього середовища. Основними параметрами, які необхідно контролювати, є температура повітря, вологість, рівень освітлення та концентрація вуглекислого газу. Оптимальні значення цих параметрів залежать від типу вирощуваних рослин, їх фази росту та цілей вирощування (наприклад, отримання плодів або декоративної продукції).

Кліматичний контроль є важливою складовою системи управління теплицею, оскільки від його ефективності залежить урожайність, якість продукції, енергоефективність теплиці та її загальна рентабельність. Традиційні методи ручного управління кліматом, як правило, мають низьку точність, а також вимагають значних зусиль з боку персоналу. У зв’язку з цим автоматизація процесів кліматичного контролю є необхідною для забезпечення стабільного росту рослин і підвищення продуктивності теплиць.

Процес управління кліматом у теплиці включає кілька рівнів прийняття рішень:

1. **Стратегічний рівень**: визначення довгострокових параметрів функціонування теплиці (наприклад, вибір оптимальних кліматичних режимів для різних культур).
2. **Адміністративний рівень**: планування використання ресурсів, вибір обладнання для автоматизації та його інтеграція в систему управління.
3. **Оперативний рівень**: контроль поточних параметрів середовища та прийняття рішень щодо їх корекції.

4. **Операційний рівень**: виконання конкретних дій, таких як включення або вимкнення систем обігріву, освітлення, вентиляції чи поливу.

Основними завданнями є забезпечення точності вимірювань, оперативності реагування на зміни умов та мінімізація витрат ресурсів.

Серед альтернативних варіантів вирішення задач можуть бути:

- **Повністю автоматизовані рішення**, які не вимагають втручання людини;
- **Напівавтоматизовані рішення**, які передбачають участь оператора у прийнятті критичних рішень;
- **Інтелектуальні рішення**, що базуються на прогнозних моделях та алгоритмах машинного навчання.

У процесі прийняття рішень враховуються такі фактори:

- **Зовнішні умови**: сезонність, погодні явища, довгострокові зміни клімату.
- **Технічний стан обладнання**: продуктивність і надійність систем обігріву, вентиляції, зволоження та освітлення.
- **Економічні фактори**: вартість енергоносіїв, витрати на обслуговування обладнання.
- **Біологічні потреби рослин**: фази росту, реакція на зміну параметрів середовища, необхідність у адаптації умов до конкретних культур.

Для забезпечення ефективного кліматичного контролю необхідна така інформація:

- **Поточні параметри середовища**, які фіксуються за допомогою сенсорів;
- **Вихідні дані про оптимальні кліматичні умови** для вирощуваних культур;
- **Дані про стан обладнання**: продуктивність, залишковий ресурс тощо;
- **Прогнозні дані**, що стосуються зовнішніх умов, таких як температура та вологість на наступні періоди.

Для побудови системи автоматизованого кліматичного контролю необхідні такі види знань:

- **Декларативні знання**: науково обґрунтовані дані про оптимальні умови вирощування рослин.
- **Процедурні знання**: алгоритми управління параметрами середовища.

- **Евристичні знання**: рекомендації, засновані на практичному досвіді вирощування рослин у теплицях.

Рішення в системі кліматичного контролю є взаємопов'язаними, оскільки зміна одного параметра (наприклад, температури) може впливати на інші (наприклад, вологість повітря). Врахування таких залежностей є важливим для забезпечення стабільності кліматичних умов у теплиці.

Автоматизована система кліматичного контролю дозволяє вирішити низку проблем, пов'язаних із ручним управлінням, а саме:

- Зниження впливу людського фактора;
- Підвищення точності та швидкості реагування на зміну параметрів середовища;
- Оптимізація витрат на утримання теплиці;
- Збільшення продуктивності та якості вирощуваних культур.

Розробка такої системи є обґрунтованою, враховуючи її здатність забезпечити стабільність кліматичних умов та енергоефективність теплиці.

1.2. Аналіз існуючих рішень

Однією з найбільш популярних систем автоматизації управління кліматом у теплицях є *Priva Climate Control*. Вона забезпечує регулювання таких параметрів, як температура, вологість, освітлення та рівень CO_2, інтегруючись з поливальними системами, вентиляцією та іншими інженерними мережами. Основними перевагами системи є її гнучкість у налаштуванні параметрів, що дозволяє адаптувати її до потреб конкретного господарства, можливість інтеграції з різноманітними інженерними мережами та доступ до даних у реальному часі через мобільні додатки чи веб-інтерфейси. Однак система має і недоліки: висока вартість впровадження обмежує її доступність для малих господарств, а залежність від інтернет-з'єднання створює ризики при нестабільному підключенні.

Рис. 1.2.1 – Система Priva Climate Control

Іншим прикладом є *Argus Controls*, що пропонує модульну структуру та можливість масштабування функціональності залежно від потреб користувача. Ця система дозволяє проводити моніторинг і управління параметрами клімату, такими як температура, вологість і полив. Вона відзначається високою надійністю у довготривалій експлуатації, можливістю інтегрувати додаткові модулі для розширення функціональності та орієнтованістю на масштабованість, що підходить для зростаючих підприємств. Водночас налаштування системи є досить складним і потребує залучення технічних фахівців, що збільшує витрати, а обмежені аналітичні можливості у базовій комплектації можуть стати проблемою для користувачів, які планують розширення бізнесу.

Рис. 1.2.2 – Система Argus Controls

Для невеликих господарств часто обирають *Growlink* – доступну систему, яка забезпечує базовий контроль кліматичних показників за допомогою датчиків і мобільного додатка. Вона вирізняється простотою використання, що робить її доступною для користувачів без технічної підготовки, а також низькою вартістю, що дозволяє впроваджувати її у малих теплицях. Проте функціонал системи є обмеженим, і вона не підходить для вирішення складних завдань. Крім того, недостатня інтеграція з іншими системами обмежує її ефективність у комплексному управлінні.

Рис. 1.2.3 – Система Growlink

Таким чином, аналіз існуючих систем демонструє їх різну функціональність і доступність, проте жодна з них не поєднує всі необхідні якості для універсального використання в теплицях. Це обґрунтовує необхідність розробки нової інтелектуальної системи, яка враховувала б специфіку галузі, була доступною для різних масштабів господарств і забезпечувала високу точність управління параметрами клімату.

У сучасному світі інформаційні технології стрімко розвиваються, відкриваючи нові можливості для створення інтелектуальних управлінських систем. У контексті автоматизації клімат-контролю в теплицях особливо актуальними є такі методи та підходи:

1. **Інтернет речей (IoT):** Забезпечує постійний моніторинг параметрів середовища та дистанційне управління. Сенсори IoT дозволяють контролювати ключові показники теплиці, автоматично реагуючи на їх зміни.
2. **Штучний інтелект (ШІ):** Алгоритми машинного навчання прогнозують зміни кліматичних умов і допомагають оптимізувати роботу системи. ШІ також виявляє несправності обладнання, що знижує ризик втрат.
3. **Хмарні обчислення:** Дозволяють зберігати й аналізувати великі обсяги даних. Хмара забезпечує доступ до даних із будь-якого місця, а також надійне резервне копіювання.
4. **Інтеграція з робототехнічними системами:** Сучасні робототехнічні рішення можуть автоматизувати фізичні процеси в теплицях, такі як полив, вентиляція чи внесення добрив. Вони також можуть працювати в парі з інтелектуальними системами для максимальної ефективності.

На ринку існує безліч готових рішень, таких як Priva Climate Control, Argus Controls і Growlink, які вже інтегрують більшість перелічених технологій. Проте жодне з цих рішень не є універсальним.

Priva Climate Control забезпечує високу точність і функціональність, але її висока вартість ускладнює доступність для малих теплиць. Argus Controls надає можливості масштабування і розширення, однак потребує значних ресурсів для впровадження і технічної підтримки. Growlink є більш доступною для невеликих господарств, але її обмежений функціонал і слабка інтеграція можуть обмежити ефективність у довгостроковій перспективі.

Аналіз новітніх технологій і готових рішень демонструє необхідність створення інтелектуальної системи управління, яка поєднує переваги IoT, машинного навчання, хмарних обчислень і робототехніки. Це дозволить забезпечити високу точність, доступність і адаптивність до потреб як малих, так і великих господарств, вирішуючи специфічні завдання предметної області.

На практиці контроль мікроклімату у теплицях часто здійснюється вручну, що є не лише неефективним, а й трудомістким процесом. Фермери змушені регулярно перевіряти стан повітря, поливати рослини, регулювати вентиляцію та обігрів, спираючись на власний досвід та суб’єктивні оцінки. Такий підхід має ряд суттєвих недоліків. По-перше, він не дозволяє швидко реагувати на зміни кліматичних умов, оскільки людина не може цілодобово контролювати стан середовища. По-друге, ручне управління часто призводить до нераціонального використання ресурсів – надмірного поливу, зайвих витрат електроенергії на обігрів або, навпаки, недостатнього зволоження та перегріву, що негативно впливає на рослини.

У зв’язку з цим впровадження автоматизованих систем контролю мікроклімату, які дозволяють моніторити стан середовища в режимі реального часу та оперативно регулювати параметри без втручання людини, є обґрунтованою потребою для підвищення продуктивності агровиробництва. Застосування Інтернету речей (IoT) та мікроконтролерів відкриває нові можливості для створення доступних, автономних рішень. Найбільш перспективними є гнучкі модульні системи на базі Arduino чи ESP32 із підтримкою хмарних технологій (наприклад, ThingSpeak), що дозволяють фермерам віддалено стежити за параметрами теплиці та керувати обладнанням.

Таким чином, незважаючи на наявність готових ІУС для теплиць, існує об'єктивна потреба у створенні адаптивної, економічно доступної, масштабованої системи кліматичного контролю для невеликих господарств. Обґрунтування вибору платформи та підходів до розробки такої системи подано в наступному підрозділі.

1.3. Підходи до побудови інформаційних управляючих систем для тепличного господарства

Ефективне управління кліматом у тепличних господарствах є одним із ключових чинників забезпечення стабільної врожайності та якості продукції. В умовах зростаючої потреби в оптимізації енергоспоживання, зменшенні впливу

людського чинника та підвищенні гнучкості агропроцесів, актуальним є застосування розподілених, адаптивних інформаційних управляючих систем (ІУС). Вибір підходу до побудови такої системи має враховувати не лише функціональні, а й технологічні та економічні аспекти.

У межах даної роботи прийнято рішення реалізовувати систему на основі **модульно-розподіленої архітектури**, що передбачає логічний і фізичний поділ її функціональних частин на незалежні автономні блоки. Такий підхід передбачає, що кожен модуль виконує окреме завдання в системі – наприклад, контроль температури, вологості, освітлення або рівня CO_2 – і має власну логіку прийняття рішень. Це дозволяє уникнути централізованого управління і забезпечити більшу гнучкість, відмовостійкість та легкість модернізації.

Кожен функціональний модуль в системі має вхід, вихід і механізм локальної обробки даних. Вхідним елементом є сенсор, що вимірює відповідний кліматичний параметр. Дані з сенсора надходять на мікроконтролер, де виконується аналіз відповідно до закладеного алгоритму. У разі необхідності активується виконавчий пристрій (наприклад, реле для обігрівача, вентилятора чи лампи), або інформація передається до хмарної платформи для зберігання, аналізу та моніторингу. У такий спосіб кожен модуль функціонує автономно, але при цьому координовано в межах єдиної системи.

Побудова ІУС за модульно-розподіленим принципом забезпечує низку переваг. По-перше, така архітектура дозволяє легко розширювати систему новими компонентами без необхідності повного перепроектування логіки. Наприклад, до вже існуючої структури можна додати модуль освітлення або модуль контролю вологості ґрунту, і це не призведе до конфлікту з іншими частинами системи. По-друге, розподілення логіки між модулями підвищує загальну надійність – у разі виходу з ладу одного елемента інші можуть продовжувати функціонування. По-третє, кожен модуль може бути протестований окремо, що значно спрощує відлагодження системи на етапі розробки та впровадження.

У межах проекту структурна логіка реалізована за допомогою мікроконтролерів Arduino UNO або ESP32, які працюють із сенсорами для збору

основних показників з датчиків для контролю температури, вологості, концентрації CO_2 і рівня освітленості. Кожен із цих сенсорів пов'язаний із відповідним модулем управління. Наприклад, модуль контролю температури отримує дані з DHT22, аналізує їх, і якщо значення перевищують допустимий поріг, вмикає або вимикає обігрівач. Аналогічно працює модуль вологості, який керує зволожувачем або вентилятором. Освітлення контролюється за допомогою датчика BH1750, а якість повітря – за допомогою MQ135. Усі ці модулі функціонують незалежно, але інтегруються у єдину структуру через спільне програмне забезпечення або канали обміну даними.

Для наочності функціональної структури системи розроблено узагальнену схему, на якій відображено модулі керування параметрами середовища та їхню взаємодію із сенсорами, виконавчими пристроями і каналами передачі даних.

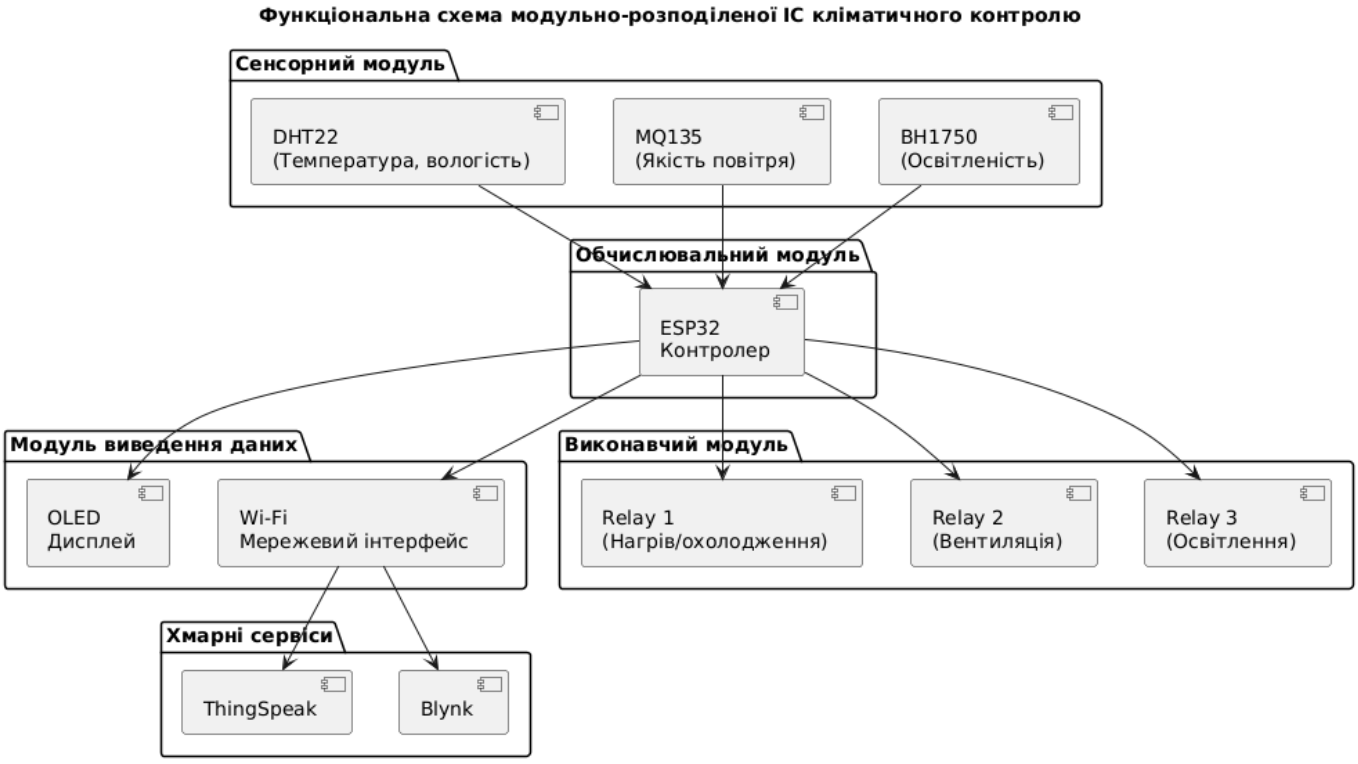


Рис. 1.3.1 – Функціональна схема модульно-розподіленої інформаційної системи кліматичного контролю

Джерело: розроблено автором

У процесі реалізації кожен модуль описується як окрема логічна одиниця, що містить об’єкти «сенсор», «контролер», «виконавець» та, за потреби,

«комунікаційний блок». Ця структура відповідає сучасним принципам побудови гнучких і масштабованих IoT-систем. Класова діаграма, наведена нижче, формалізує логічні зв’язки між елементами системи.

Структурна діаграма логічних об’єктів модульної ІУС кліматичного контролю

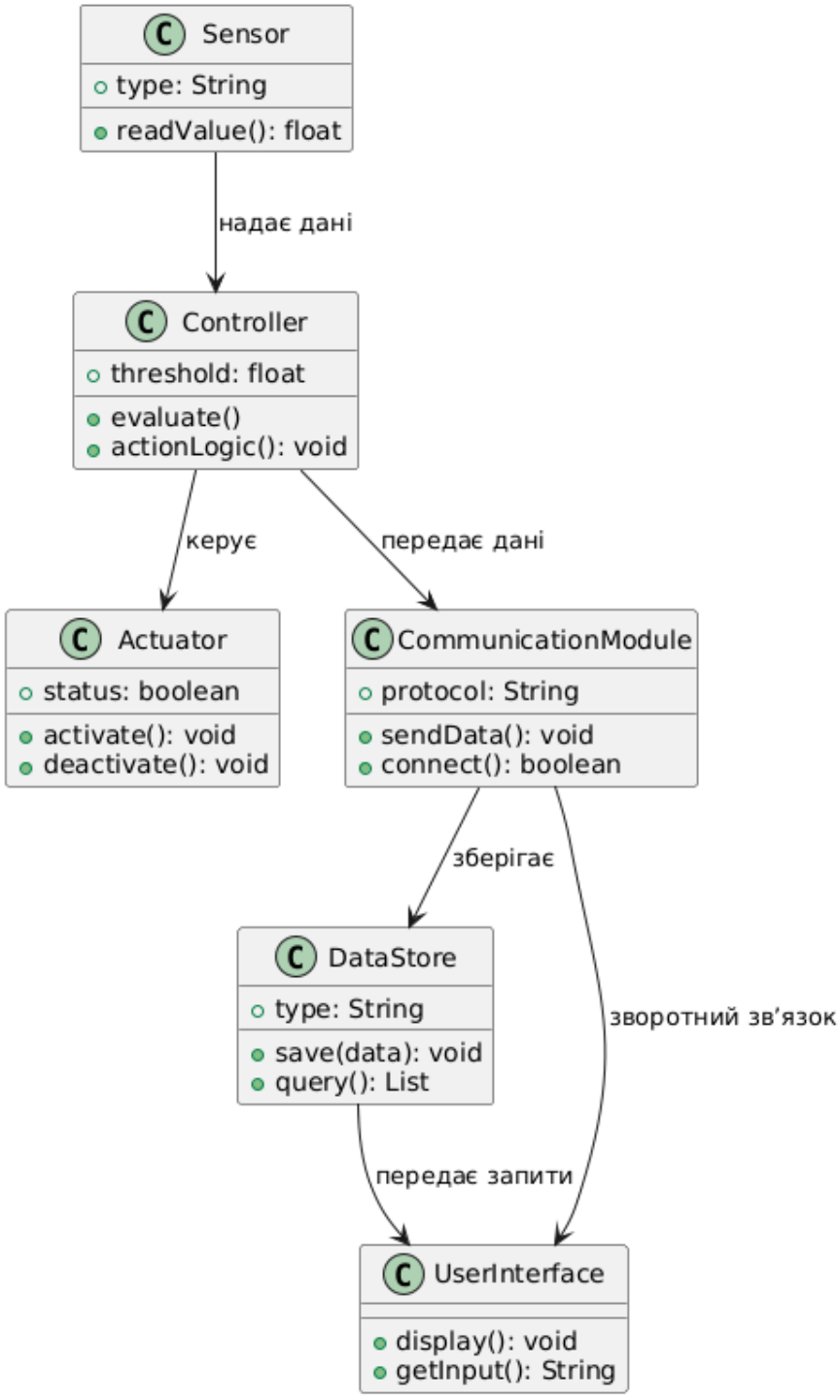


Рис. 1.3.2 – Структурна діаграма взаємодії логічних об’єктів у межах модулів ІУС

Джерело: розроблено автором

Між зазначеними елементами встановлюються однозначні зв’язки: сенсор надає значення параметра, контролер приймає рішення, актуатор виконує дію, а комунікаційний блок у разі використання ESP32 передає інформацію в хмару. Така структура дозволяє наочно представити не лише логіку роботи окремого модуля, а й загальну архітектуру системи як розподіленої множини взаємодіючих логічних блоків.

Для повноти уявлення про роботу системи розроблено фізичну схему, яка показує зв’язки між апаратними компонентами – сенсорами, мікроконтролерами, виконавчими пристроями, інтерфейсом зв’язку та хмарним середовищем. На цій схемі візуалізовано, як дані передаються між фізичними елементами, які порти використовуються для з’єднання, які модулі обробляють інформацію і в якому вигляді вона надходить до користувача.

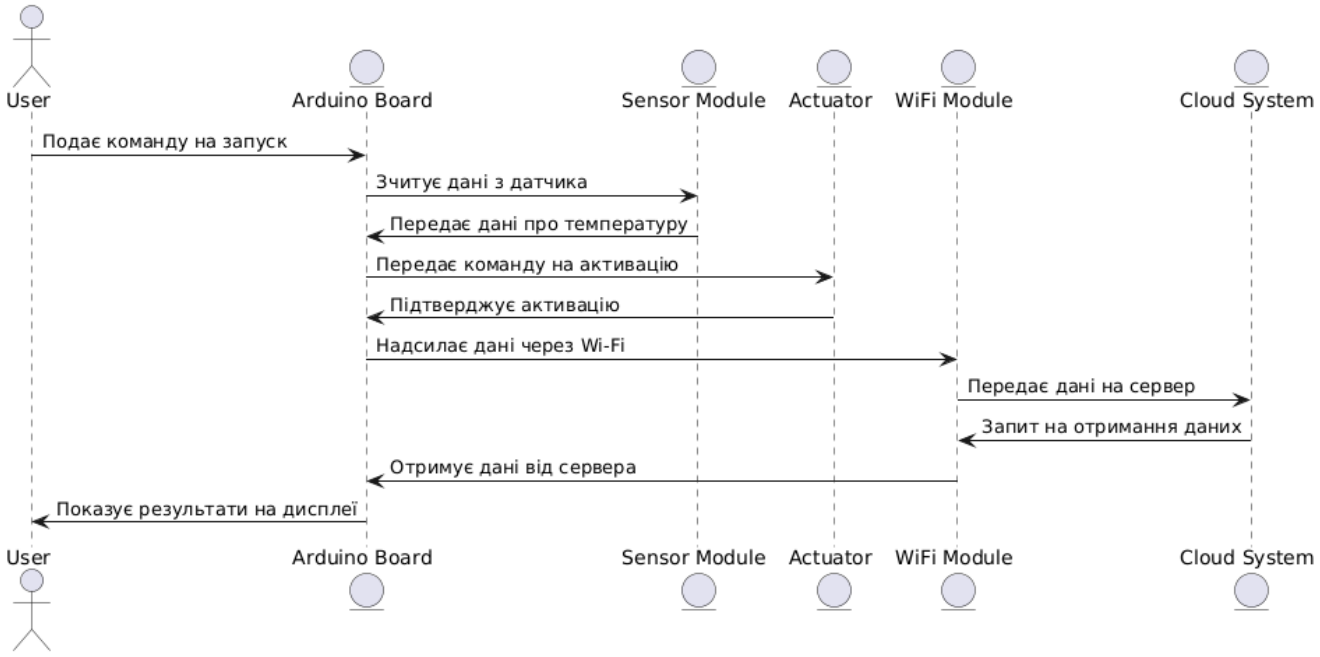


Рис. 1.3.3 – Фізична схема взаємодії апаратних компонентів системи керування кліматом

Джерело: розроблено автором

У результаті можна стверджувати, що обраний підхід до побудови ІУС для теплиць – модульно-розподілена архітектура з автономними логічними модулями – повністю відповідає поставленим цілям і вимогам. Він забезпечує не лише

технологічну гнучкість і масштабованість, але й робить систему стійкою до відмов, економічно ефективною та зручною для модернізації. Такий підхід дозволяє створити практично реалізовану, адаптивну систему управління, яка функціонує на основі відкритих, доступних апаратних платформ, із можливістю інтеграції з хмарними сервісами моніторингу й аналізу.

1.4. Обґрунтування вибору технологій

Для реалізації інформаційної управляючої системи кліматичного контролю в теплиці, побудованої на основі розподіленої логіки, було обрано набір апаратних та програмних засобів, які відповідають вимогам до адаптивності, автономності, масштабованості та економічної доцільності. Архітектура системи ґрунтується на **модульно-розподіленому підході**, що передбачає поділ функціональності між незалежними блоками (модулями), кожен з яких виконує окреме завдання: збір даних, керування окремим параметром, передача інформації, обробка чи візуалізація.

У межах такого підходу кожен модуль виконує роль **автономного контролера**, який:

- отримує вхідні дані від відповідного сенсора;
- приймає рішення на основі локальної логіки (наприклад, порогової чи гістерезисної);
- керує відповідним виконавчим пристроєм;
- може взаємодіяти з іншими модулями через спільні змінні або сигнали.

Для побудови системи на апаратному рівні було обрано **Arduino UNO** як основну платформу для збору та обробки даних. Arduino забезпечує надійність, простоту програмування, підтримку широкого спектра сенсорів, низьке енергоспоживання та велику спільноту розробників, що спрощує розгортання системи у практичному середовищі. В альтернативних конфігураціях використовується **ESP32**, який, на відміну від Arduino UNO, має вбудований модуль Wi-Fi, що дозволяє реалізувати бездротове з'єднання з хмарною платформою для моніторингу та віддаленого керування.

До складу сенсорної підсистеми включено:

- **DHT22** – сенсор температури й вологості, що має високу точність і цифровий вихід;
- **MQ135** – газовий датчик для виявлення концентрації CO_2 та інших шкідливих газів;
- **BH1750** – цифровий датчик освітленості, який дозволяє точно вимірювати інтенсивність світла в люксах і реагувати на зміну природного освітлення в теплиці.

Для реалізації керуючих дій використовуються релейні модулі або транзисторні ключі, що дозволяють вмикати/вимикати обігрівачі, вентилятори, освітлення чи зволожувачі.

Для зберігання, обробки та візуалізації даних обрано хмарну платформу **ThingSpeak**, яка дозволяє:

- зберігати дані з мікроконтролерів у режимі реального часу;
- будувати графіки, діаграми й аналітичні представлення;
- налаштовувати обробку даних і створювати звітність;
- інтегруватися з іншими платформами (наприклад, MATLAB) для глибшого аналізу.

Моделювання та попереднє тестування роботи системи здійснюється в середовищі **Wokwi** – онлайн-симуляторі, що дозволяє віртуально зібрати схему, завантажити скетч Arduino, протестувати логіку та взаємодію компонентів без необхідності використання фізичного обладнання. Це дозволяє швидко виявляти помилки, оптимізувати логіку керування та налаштовувати взаємодію модулів.

Для програмної інтеграції та розширення можливостей керування застосовується **Node-RED** – графічне середовище візуального програмування потоків даних. Його використання дає змогу легко пов’язати апаратні пристрої, хмарні платформи, локальні обробники подій і візуальні інтерфейси. Таким чином, система отримує можливість масштабування, розширення функціоналу та легкого оновлення без зміни основної логіки.

Для зберігання структурованих та неструктурованих даних застосовується **MongoDB** – документно-орієнтована база даних, що ідеально підходить для зберігання телеметричних даних, отриманих із сенсорів. MongoDB дозволяє ефективно зберігати динамічні структури без жорсткого формалізму SQL-схем, що особливо зручно в умовах постійно змінюваного середовища.

Для візуалізації даних використовується **Grafana** – система побудови дашбордів, яка дозволяє в режимі реального часу відображати значення параметрів, змінні тренди, аналітичні діаграми, а також створювати інтерфейс спостереження для кінцевого користувача.

Загалом комбінація **розподіленої логіки функціональних модулів**, програмованих на Arduino / ESP32, з хмарними сервісами обробки та візуалізації даних створює передумови для побудови **надійної, масштабованої та економічно ефективної системи** кліматичного контролю в умовах невеликих тепличних господарств.

1.5. Висновки до розділу

У першому розділі було здійснено системний аналіз предметної області, пов’язаної з автоматизацією кліматичного контролю у тепличному господарстві. Встановлено, що мікроклімат є критично важливим фактором для ефективного вирощування рослин. Зміни температури, вологості, освітлення та концентрації вуглекислого газу істотно впливають на фотосинтез, темпи росту та врожайність культур. Наявні на ринку інформаційні управляючі системи або є надто дорогими та складними у впровадженні, або не задовольняють вимоги до гнучкості, доступності та масштабованості, особливо у випадку малих та середніх аграрних підприємств.

У зв’язку з цим було обґрунтовано необхідність створення власної автоматизованої системи кліматичного контролю, що поєднуватиме простоту реалізації, доступність, автономність, розширюваність і підтримку сучасних хмарних технологій.

На концептуальному рівні обґрунтовано доцільність використання **модульно-розподіленої архітектури**, у межах якої функціональність системи поділена між окремими автономними логічними модулями. Кожен модуль реалізує контроль за певним параметром – температурою, вологістю, освітленням або рівнем CO_2 – та функціонує незалежно від інших, з можливістю обміну даними або сигналами керування. Такий підхід забезпечує адаптивність, гнучкість, відмовостійкість та спрощує супровід і масштабування системи.

Для реалізації цієї архітектури обрано низку технологій, що відповідають технічним та економічним вимогам. Основу апаратної частини складають мікроконтролери **Arduino UNO** та **ESP32**, які забезпечують локальне управління та бездротову передачу даних. Використання цифрових сенсорів – **DHT22**, **MQ135** та **BH1750** – дозволяє отримувати достовірні дані про кліматичні умови в теплиці. Для моніторингу, аналізу та віддаленого доступу застосовано хмарну платформу **ThingSpeak**, середовище візуального програмування **Node-RED**, базу даних **MongoDB** та інструмент візуалізації **Grafana**. Моделювання та первинне тестування системи здійснюється у середовищі **Wokwi**, що дає змогу відпрацьовувати логіку без використання фізичного обладнання.

У результаті розділу сформовано теоретичне підґрунтя для створення автоматизованої системи кліматичного контролю, визначено основні функціональні модулі, обґрунтовано вибір апаратних і програмних засобів реалізації. У наступному розділі буде розглянуто структурну та принципову побудову системи, її моделі та методи керування.

РОЗДІЛ 2. ХАРАКТЕРИСТИКА ІУС З КОНТОЛЮ МІКРОКЛІМАТУ ТЕПЛИЦІ ТА МЕТОДИ І МОДЕЛІ

2.1. Структура, методи та моделі інформаційної управляючої системи кліматичного контролю

Об’єктом дослідження в межах даної роботи є автоматизована система контролю мікроклімату теплиці, що базується на використанні мікроконтролера ESP32, сенсорних датчиків, виконавчих пристроїв і хмарних сервісів для збору, обробки та аналізу даних. Основна мета розробки полягає у створенні гнучкого технологічного рішення, здатного ефективно регулювати ключові параметри середовища в теплиці, зокрема температуру, вологість, рівень освітленості та концентрацію вуглекислого газу.

Інформаційна управляюча система побудована за модульно-розподіленою архітектурою, що передбачає чітке розмежування функціональних блоків відповідно до їхнього призначення. Уся система складається з чотирьох основних підсистем: блоку збору даних, обчислювального модуля, виконавчого блоку та модуля збереження і візуалізації даних. Кожен з елементів функціонує автономно, але при цьому взаємодіє з іншими компонентами в рамках цілісного циклу управління.

Блок збору даних включає сенсорні елементи, призначені для зчитування параметрів навколишнього середовища. Наприклад, датчик DHT22 використовується для вимірювання температури та відносної вологості повітря, MQ-135 — для оцінки рівня газових домішок, передусім CO_2, а BH1750 дозволяє контролювати рівень освітленості. Дані сенсори забезпечують постійний моніторинг середовища в реальному часі, що є ключовою передумовою для ефективного керування умовами вирощування рослин.

Обчислювальний модуль представлений мікроконтролером ESP32, який виконує обробку даних з датчиків, застосовує алгоритми логічного аналізу, приймає рішення щодо керування реле та ініціює надсилання даних у хмарні

сервіси. Алгоритмічна модель функціонування системи заснована на послідовному циклі зчитування, обробки та реагування, реалізованому мовою C++ у середовищі Arduino IDE. Управління реалізовано за принципом подійно-орієнтованої реакції на зміну сенсорних показників, що дозволяє досягти високої адаптивності.

Виконавчий блок включає релейні модулі, кожен з яких відповідає за активацію певного виконавчого пристрою — поливальної системи, вентиляції або підсвічування. Реле спрацьовують на основі порогових значень, встановлених у логіці програми, зокрема: температура понад 30 °C активує охолодження, низька освітленість вмикає додаткове освітлення, а перевищення граничного рівня CO_2 спричинює вентиляцію.

Модуль збереження та аналізу даних представлений хмарними платформами, такими як ThingSpeak і Blynk. Вони дозволяють не лише візуалізувати інформацію у вигляді графіків і панелей, а й забезпечують віддалене керування через мобільний застосунок. Таким чином, реалізовано модель двостороннього зв’язку: система не лише передає дані, а й отримує зовнішні сигнали для зміни параметрів у реальному часі.

Методологічна основа управління побудована на сенсорно-керованій регуляції із використанням логічних функцій. Наприклад, для температурного контролю застосовується функція:

$$f_1(T) = \{\, 1, \text{якщо } T > T_{\text{поріг}};\ 0 — \text{інакше} \,\}$$

що реалізує вмикання відповідного реле при перевищенні граничного значення. Подібні логічні правила застосовуються і для інших сенсорних каналів.

Окремої уваги заслуговує реалізована модель розподілу обчислювального навантаження між локальними модулями (ESP32) і віддаленими сервісами (хмара), що підвищує ефективність роботи системи. Це відповідає сучасним підходам до побудови інформаційних систем Інтернету речей, які поєднують локальну автономію з можливістю глобального аналізу даних.

Інформаційна система функціонує у режимі реального часу, з високою частотою оновлення даних та мінімальними затримками в управлінні. Автоматичний режим дозволяє системі працювати незалежно від користувача, тоді як інтерфейс мобільного застосунку забезпечує можливість дистанційного контролю і конфігурації.

Таким чином, розроблена система має чітко визначену структуру та методичне підґрунтя, що ґрунтується на сучасних моделях сенсорного моніторингу, логічного управління, хмарної аналітики та розподіленої архітектури. Її реалізація дозволяє досягти високого рівня автоматизації, що є необхідним для ефективного функціонування тепличного господарства в умовах змінного клімату та зростаючих вимог до якості аграрної продукції.

2.2. Актуальність та мета дослідження

Автоматизація процесів у сільському господарстві є важливим напрямком розвитку сучасних технологій, оскільки дозволяє значно підвищити ефективність вирощування культур, знизити витрати ресурсів та оптимізувати контроль за мікрокліматом. Одним із ключових факторів, що впливають на врожайність рослин у тепличних умовах, є мікроклімат, зокрема температура, вологість повітря та ґрунту, рівень вуглекислого газу. Недостатній або надмірний рівень вологості може призвести до загибелі рослин, а різкі температурні коливання можуть сповільнювати їхній розвиток або спричиняти хвороби.

У зв’язку з цим актуальною є розробка автоматизованої системи контролю мікроклімату теплиці на основі сучасних технологій, що забезпечуватиме безперервний моніторинг середовища та оперативне реагування на зміни умов. Використання мікроконтролера ESP32 у поєднанні з сенсорами та хмарними сервісами дозволяє реалізувати систему, здатну функціонувати автономно та забезпечувати дистанційний контроль параметрів. Це сприяє зменшенню людського втручання, економії ресурсів та підвищенню якості врожаю.

Метою дослідження є розробка автоматизованої системи контролю мікроклімату теплиці, яка забезпечуватиме безперервний моніторинг основних

параметрів довкілля, аналіз отриманих даних та регулювання умов з метою створення оптимального середовища для росту рослин.

Для досягнення поставленої мети необхідно виконати такі **завдання**:

- Дослідити особливості підтримки мікроклімату в тепличних умовах, визначити основні параметри, що впливають на ріст рослин.
- Розробити програмно-апаратний комплекс для збору та обробки даних про температуру, вологість повітря і ґрунту.
- Реалізувати алгоритми автоматичного прийняття рішень для регулювання кліматичних умов у теплиці.
- Інтегрувати систему з хмарним сервісом для забезпечення збору даних, їхнього аналізу та віддаленого доступу.
- Провести тестування розробленої системи та оцінити її ефективність у реальних умовах експлуатації.

Об'єктом дослідження є інтелектуальна система контролю мікроклімату теплиці, яка включає сенсори, мікроконтролер, виконавчі пристрої та програмне забезпечення для аналізу отриманих даних і прийняття відповідних рішень.

Предметом дослідження є методи обробки даних з датчиків, алгоритми автоматичного регулювання параметрів мікроклімату, а також інтеграція системи з хмарними технологіями для віддаленого моніторингу.

Практичне значення розробленої системи полягає у забезпеченні стабільного мікроклімату в теплиці без необхідності постійного контролю з боку людини. Це дозволяє оптимізувати використання ресурсів, зменшити ризик втрат врожаю через несприятливі умови та підвищити загальну ефективність тепличного господарства. Інтеграція з хмарними сервісами надає можливість дистанційного управління, що є особливо важливим для фермерських господарств і підприємств, які працюють з великими обсягами сільськогосподарської продукції.

2.3. Структурна схема системи

Проєктування інформаційної управляючої системи (ІУС) кліматичного контролю в теплиці вимагає чіткого визначення складу її компонентів та характеру

взаємозв’язків між ними. Структурна схема є базовим відображенням архітектури системи, яка дозволяє візуалізувати її функціональні частини, сформулювати логіку взаємодії окремих модулів та підготувати основу для подальшого апаратного й програмного впровадження. У рамках цієї дипломної роботи структурна схема побудована відповідно до обраного модульно-розподіленого підходу, що передбачає автономність функціональних блоків при їх спільному інтеграційному узгодженні.

Система складається з декількох логічно та фізично взаємопов’язаних підсистем. Кожна з них реалізує окрему частину процесу: збір інформації про навколишнє середовище, обробку даних, керування виконавчими пристроями, передавання даних у хмарне середовище, зберігання, візуалізацію та забезпечення інтерфейсу взаємодії з користувачем. Усі підсистеми об’єднані єдиною метою – забезпечення стабільного та оптимального кліматичного режиму в теплиці для вирощування рослин.

На рівні сенсорної підсистеми передбачено використання набору цифрових датчиків для вимірювання ключових параметрів мікроклімату. До її складу входять: датчик температури та вологості DHT22, концентрації вуглекислого газу MQ135, а також датчик освітленості BH1750. Ці сенсори встановлюються безпосередньо в теплиці в контрольованих точках та періодично передають дані на мікроконтролер для подальшої обробки.

Центральну обробку та логіку керування реалізує мікроконтролер Arduino UNO або ESP32, залежно від конфігурації. Arduino UNO використовується для автономної локальної реалізації без доступу до Інтернету, тоді як ESP32 – для інтеграції з хмарною інфраструктурою через Wi-Fi-з’єднання. Саме на цьому рівні реалізується алгоритм прийняття рішень – система порівнює поточні показники із заданими пороговими значеннями та генерує команди для виконавчих пристроїв.

Виконавча підсистема включає пристрої, що безпосередньо впливають на клімат у теплиці: обігрівачі, вентилятори, зволожувачі, освітлювальні модулі. Кожен з них керується за допомогою відповідного модуля: реле або транзисторного ключа, підключеного до мікроконтролера. У разі виявлення відхилень від

нормального діапазону значень, система активує відповідний пристрій для нормалізації параметра середовища.

Для забезпечення віддаленого моніторингу та накопичення даних реалізована підсистема хмарної інтеграції. За допомогою ESP32 або окремого Wi-Fi модуля дані передаються на платформу **ThingSpeak**, яка зберігає їх у вигляді каналів та надає інтерфейси для побудови графіків і таблиць. Інтеграція з базою даних **MongoDB** дозволяє організовувати більш гнучке й масштабоване збереження телеметрії, а використання **Grafana** забезпечує побудову адаптивних дашбордів для візуалізації даних у режимі реального часу.

Користувач може взаємодіяти із системою через мобільний або веб-інтерфейс, зчитуючи поточні параметри клімату, переглядаючи історичні дані, встановлюючи порогові значення або змінюючи режим роботи. Інтерфейсна підсистема забезпечується через хмарні сервіси або локальні дисплеї (у простіших версіях).

На рисунку 2.3.1 наведено структурну схему системи, в якій візуалізовано основні підсистеми, їх елементи та напрямки взаємодії між ними. Сенсорні модулі передають дані на мікроконтролер, який приймає рішення і керує виконавчими механізмами, а також передає зібрану інформацію на хмарну платформу.

У цій схемі чітко виділяються ключові інформаційні потоки: від сенсорів до мікроконтролера, від мікроконтролера до актуаторів, і, в разі використання ESP32, – до хмарної інфраструктури. Окремо варто підкреслити, що схема передбачає можливість масштабування системи за рахунок додавання нових сенсорів, виконавчих пристроїв або логічних модулів – наприклад, для обліку вмісту вологи в ґрунті, рівня pH чи зовнішніх кліматичних умов.

Застосування модульно-розподіленого підходу дає змогу досягти високої гнучкості в адаптації системи до конкретних умов теплиці, зокрема її площі, конфігурації, вимог до мікроклімату та енергоспоживання. Кожен функціональний блок системи може бути реалізований як незалежна частина – як фізично, так і логічно. Це дозволяє розгортати систему поступово, обслуговувати окремі зони

теплиці автономно, проводити технічне обслуговування або оновлення без зупинки всієї системи.

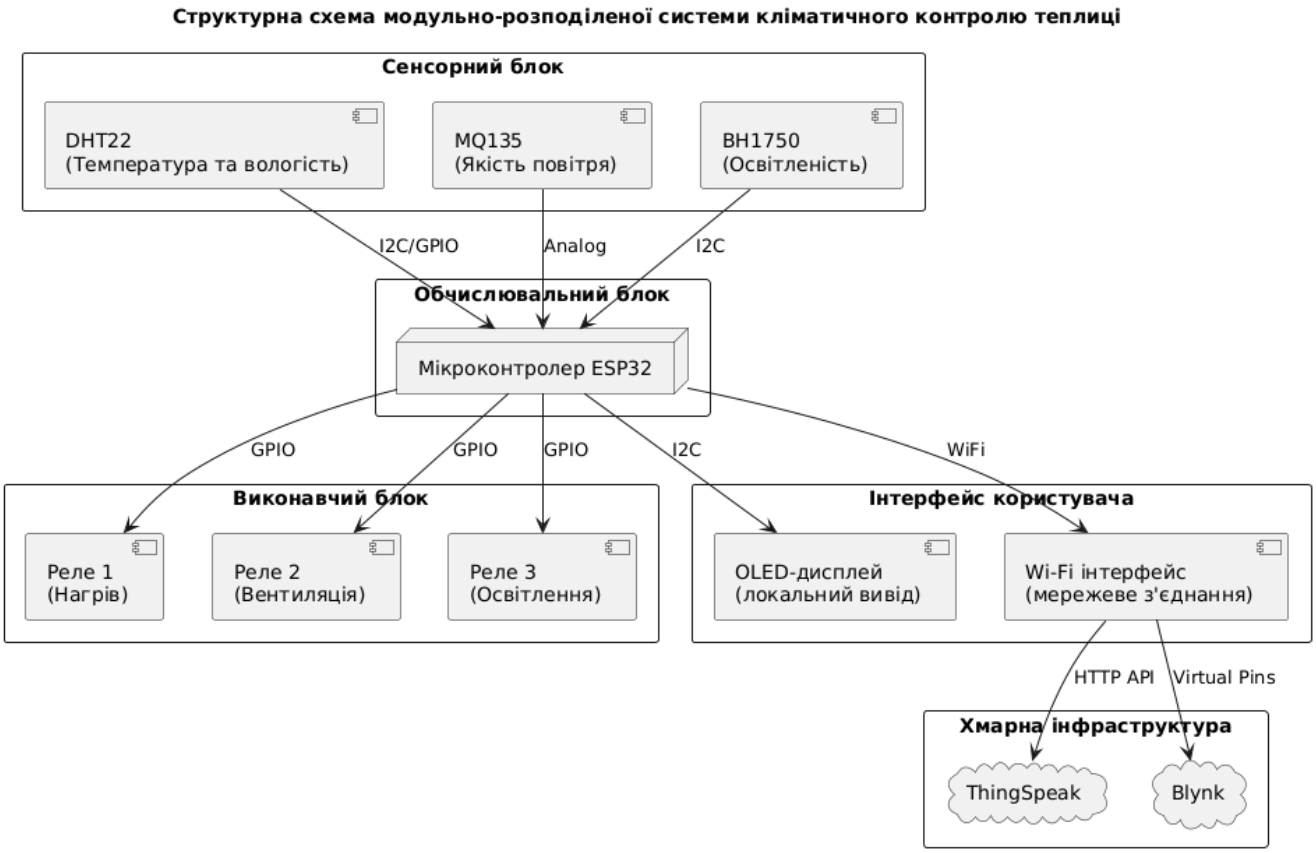


Рис. 2.3.1 – Структурна схема модульно-розподіленої системи кліматичного контролю теплиці

Джерело: розроблено автором

Таким чином, побудована структурна схема повністю відповідає архітектурним принципам, визначеним у попередньому розділі, і формує основу для подальшого проектування електричних схем, програмного коду, інтерфейсів взаємодії та алгоритмів управління в автоматизованій системі кліматичного контролю.

2.4. Принципова електрична схема системи

Принципова електрична схема є графічним та концептуальним відображенням фізичних взаємозв’язків між основними компонентами автоматизованої системи кліматичного контролю. Вона дозволяє унаочнити

розміщення, типи підключення, напрямки інформаційних та енергетичних потоків, а також визначити технічні умови взаємодії між мікроконтролерами, сенсорами, виконавчими пристроями та джерелами живлення. Відповідність цієї схеми реальній апаратній реалізації є критично важливою для забезпечення стабільності, безпеки та масштабованості системи.

У розробленій системі основною керуючою ланкою є мікроконтролер **ESP32**, який реалізує функції зчитування даних, прийняття рішень, керування виконавчими пристроями та передавання інформації на хмарну платформу. У конфігураціях, що не передбачають бездротового зв'язку, може бути використано **Arduino UNO**, однак ESP32 забезпечує ширшу функціональність завдяки вбудованому Wi-Fi, розширеній кількості портів введення/виведення та підтримці сучасних протоколів обміну.

Сенсорна підсистема системи представлена трьома цифровими й аналоговими сенсорами:

- **DHT22** – сенсор температури та вологості, підключений до одного з цифрових входів ESP32 через сигнальний пін. Для стабільної роботи на сигнальній лінії використовується підтягуючий резистор номіналом 4.7 кОм.
- **MQ135** – аналоговий газовий датчик для визначення рівня CO_2. Його вихід зчитується через аналоговий порт ESP32. Для захисту мікроконтролера варто передбачити обмежувальний резистор або фільтруючий RC-ланцюг.
- **BH1750** – цифровий датчик освітленості, що також працює через шину I^2C. У випадку використання кількох пристроїв на одній шині, їх можна адресувати програмно через зміну положення ADDR-піна.

Мікроконтролер отримує живлення через порт microUSB або від зовнішнього джерела постійного струму 5 В, з можливістю стабілізації до 3.3 В через вбудований регулятор. У системах, орієнтованих на автономність, можливе живлення від акумулятора через стабілізатор типу AMS1117. Альтернативно, живлення може подаватися з використанням сонячної панелі та контролера заряду TP4056 з акумулятором Li-ion, що дозволяє використовувати систему в енергонезалежному режимі.

Для реалізації впливу на середовище система оснащується виконавчими пристроями, керованими через **релейні модулі з гальванічною розв'язкою**. Кожен модуль реле має вхід IN (керуючий), VCC (живлення логіки) та GND, а також контакти навантаження – COM, NO, NC. Контроль відбувається шляхом подачі логічного сигналу LOW або HIGH на вхід IN, залежно від типу модуля. Для захисту схеми рекомендовано використовувати діод Шотткі (наприклад, 1N5819) паралельно котушці реле, що гасить зворотну ЕРС при розмиканні контакту. До реле можуть бути підключені пристрої з мережевим живленням – лампи освітлення, вентилятори, насосні установки, нагрівальні елементи тощо. Типове навантаження, що витримується стандартним реле – до 10 А при 250 В змінного струму.

Передавання даних на хмарну платформу реалізується через Wi-Fi-з'єднання ESP32 з маршрутизатором. Зчитані дані формуються в HTTP-запити і надсилаються до сервісу **ThingSpeak**, де записуються у відповідні поля каналу. Це дозволяє зберігати історію параметрів, відображати їх у вигляді графіків, аналізувати тренди та будувати логіку сповіщень у разі виходу значень за допустимі межі.

На рисунку 2.4.1 наведено принципову електричну схему системи. У ній візуалізовано всі основні елементи: сенсори, мікроконтролер, реле, джерело живлення та хмарну інфраструктуру. Стрілками показано напрямок даних і сигналів керування, підписи вказують номери портів, типи сигналів та напруги.

Схема розроблена з урахуванням модульної побудови системи: кожен сенсор має власне підключення, кожне реле – окрему логіку активації. Це дозволяє в майбутньому додавати нові елементи без зміни загальної структури. Крім того, схема враховує аспекти електробезпеки: гальванічна розв'язка реле, захист зворотної напруги, можливість використання стабілізаторів напруги та розділення сигнальних і силових ланцюгів.

Таким чином, принципова схема відображає технічно реалізовану основу системи, що поєднує в собі логіку керування, зчитування, обробки та передавання даних у реальному часі. Її побудова з урахуванням стандартів схемотехніки

забезпечує надійність, функціональність і гнучкість розробленої ІУС кліматичного контролю.

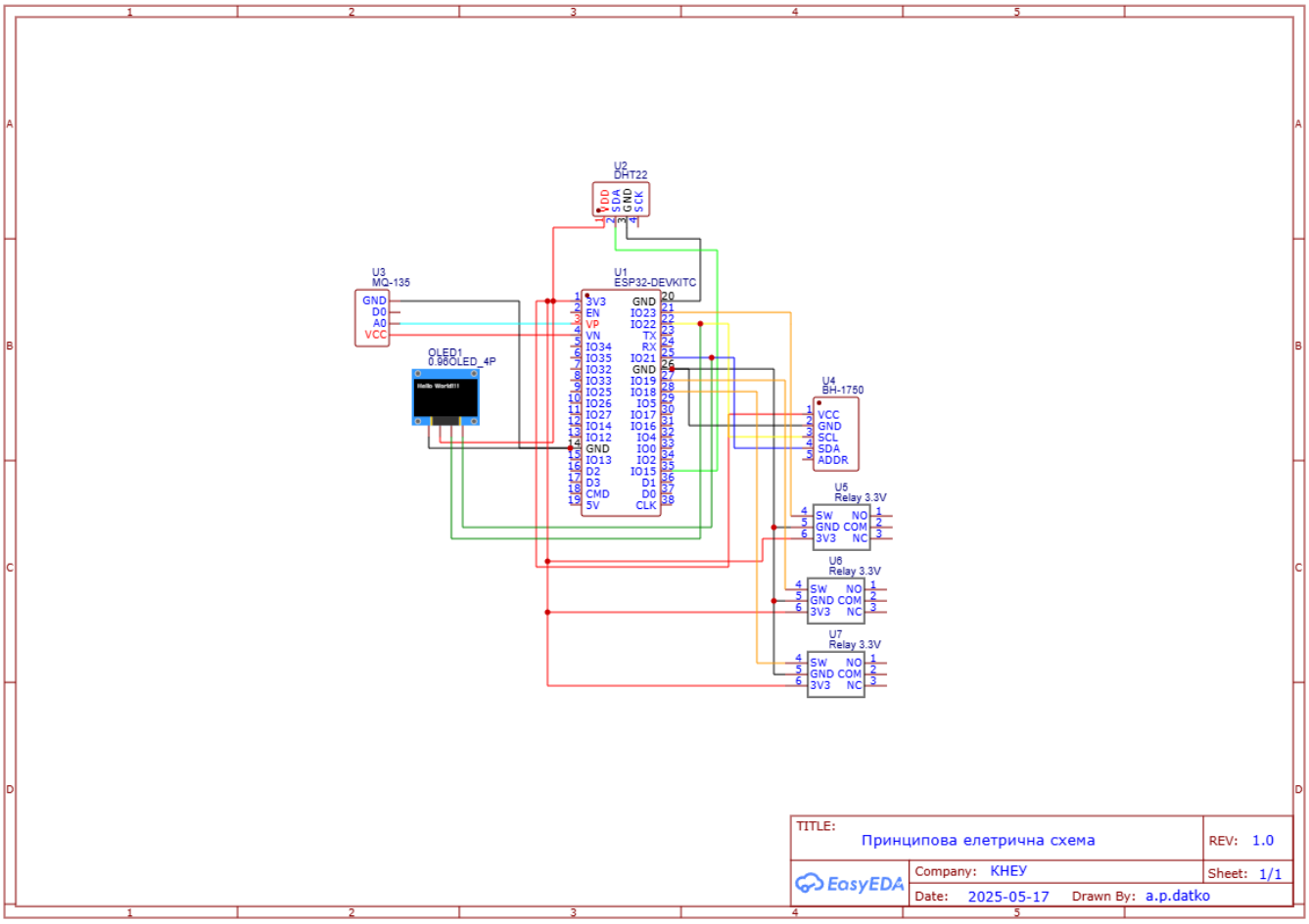


Рис. 2.4.1 – Принципова електрична схема автоматизованої системи кліматичного контролю теплиці

Джерело: розроблено автором

2.5. Алгоритм роботи системи

Робота автоматизованої системи кліматичного контролю базується на циклічному виконанні логічних операцій, що забезпечують постійний моніторинг параметрів середовища, аналіз отриманих даних, прийняття керуючих рішень, активацію відповідних виконавчих пристроїв і, за потреби, передавання даних до хмарного сховища для подальшої візуалізації та контролю.

Алгоритм функціонування системи реалізується на рівні мікроконтролера ESP32, який виконує всі основні логічні операції. Кожен цикл роботи системи

складається з декількох етапів, що виконуються послідовно, з певною періодичністю. Така модель дозволяє забезпечити безперервність та автономність процесу керування мікрокліматом.

Першим етапом у загальній послідовності є **зчитування даних із сенсорів**. У цей момент контролер ініціалізує відповідні пристрої та отримує від них значення: температуру та вологість з DHT22, рівень освітленості з BH1750, а також концентрацію газів із MQ135. Усі ці значення зберігаються у внутрішній оперативній пам’яті мікроконтролера у вигляді змінних, доступних для подальшої обробки.

На наступному етапі відбувається **аналіз отриманих даних**. Кожне значення порівнюється з відповідними граничними параметрами, що встановлені користувачем або запрограмовані за замовчуванням. Наприклад, якщо температура перевищує встановлений поріг, система формує сигнал про потребу охолодження, а якщо рівень вологості недостатній – генерується команда на зволоження. Усі логічні операції виконуються за принципом умовних конструкцій, які формують базову структуру керування: if–else або switch–case.

У разі виявлення відхилень, система переходить до **етапу керування виконавчими пристроями**. Залежно від результатів логічного аналізу, активуються відповідні вихідні порти мікроконтролера, до яких підключені релейні модулі. Через реле вмикаються або вимикаються: нагрівач, вентилятор, зволожувач або джерело освітлення. Пристрої працюють доти, доки значення параметра не повернеться в допустимий діапазон, після чого система автоматично припиняє дію, що забезпечує циклічність і автономність роботи.

Після виконання операцій керування, система переходить до **етапу передачі даних у хмарну систему**. Якщо мікроконтролер має доступ до мережі Wi-Fi, зібрані дані надсилаються до каналу на платформі ThingSpeak або іншого обраного сервісу. У хмарі дані записуються у відповідні поля, що дозволяє у подальшому аналізувати тренди, виводити графіки та формувати сигнали сповіщення.

Завершальним етапом є **очікування до наступного циклу**, яке реалізується через функцію затримки або таймери реального часу. Після закінчення затримки

система повторює весь цикл, знову зчитуючи дані, аналізуючи їх і реагуючи на зміну параметрів середовища. Такий підхід забезпечує постійний контроль над станом мікроклімату в теплиці та своєчасне реагування на можливі відхилення.

Узагальнений алгоритм роботи системи кліматичного контролю можна подати у вигляді блок-схеми, що представлена на рисунку 2.5.1. На схемі показано послідовність основних етапів: ініціалізація, зчитування даних, обробка, прийняття рішень, керування, передача даних, очікування та повторення циклу.

Такий алгоритм дозволяє системі працювати автономно, адаптуватися до змін навколишнього середовища, зменшити потребу в постійному людському контролі та забезпечити умови для сталого розвитку тепличного господарства.

2.6. Методи автоматичного управління мікрокліматом

Процес керування мікрокліматом у теплиці є багатофакторним і динамічним, оскільки залежить від великої кількості змінних, що взаємодіють між собою: температура повітря, вологість, освітленість, рівень CO_2 тощо. Ефективне управління такими параметрами вимагає використання методів, що забезпечують адаптивність, гнучкість реагування та стабільність системи за змін зовнішніх умов. Автоматичне керування передбачає прийняття рішень на основі аналізу поточних значень параметрів та їх відхилень від нормативних або оптимальних значень без втручання оператора.

У системах такого типу найбільш доцільно застосовуються методи **позиційного, релейного (двопозиційного), пропорційного та комбінованого регулювання**, які дозволяють здійснювати керування в замкненому циклі з використанням зворотного зв’язку. Основу системи становить програмна реалізація алгоритмів прийняття рішень, в яких логіка вбудована у мікроконтролер та реалізується на рівні умовних операторів у коді.

Найпростішим і широко поширеним є метод **двопозиційного (релейного) регулювання**, за якого виконавчий механізм має лише два стани – «увімкнено» або «вимкнено». Цей метод ефективний для систем, де достатньо підтримувати пара-

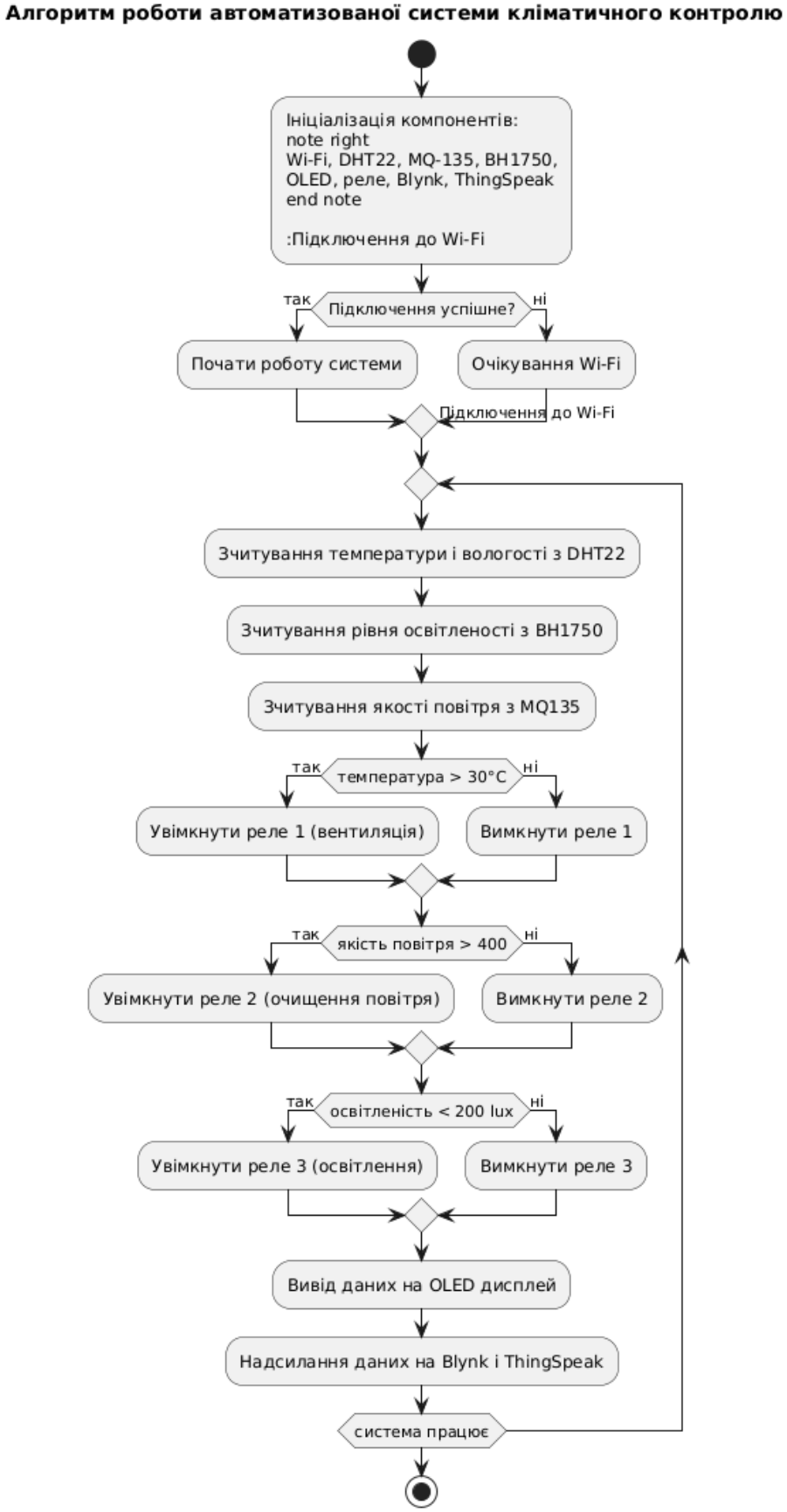


Рис. 2.5.1 – Алгоритм роботи автоматизованої системи кліматичного контролю

Джерело: розроблено автором

метри в заданому діапазоні, без необхідності плавного переходу. Наприклад, обігрівач вмикається при зниженні температури нижче порогу і вимикається при досягненні верхньої межі. Для запобігання частому вмиканню та вимиканню при незначних коливаннях доцільно використовувати **гістерезис** – різницю між точкою вмикання і точкою вимикання. Такий підхід дозволяє знизити знос реле та стабілізувати роботу.

Більш гнучким є **пропорційний метод керування**, що базується на залежності сили впливу на об’єкт від величини відхилення. У контексті мікроконтролерних систем його реалізація можлива через **широтно-імпульсну модуляцію (PWM)**, яка дозволяє змінювати ступінь активації виконавчого пристрою залежно від відхилення параметра. Наприклад, швидкість обертання вентилятора може бути пропорційною до рівня CO_2 або перегріву повітря. Такий метод потребує більш складного алгоритмічного забезпечення, однак значно підвищує ефективність та енергоощадність системи.

Для багатофакторного керування, коли необхідно одночасно враховувати декілька параметрів (наприклад, температуру та вологість), можливе використання **комбінованих методів**, в яких логіка реалізується як система правил або таблиць умов (таблиці істинності). Наприклад, при високій температурі і низькій вологості вмикається не лише вентилятор, а й зволожувач; при середній температурі і надлишковій вологості – вентиляція працює в обмеженому режимі. Такий підхід реалізується програмно і дозволяє отримати адаптивне керування без використання складних математичних моделей.

Окремо слід згадати про можливість використання **алгоритмів нечіткої логіки**, які є наближеними до способу мислення людини й дозволяють створювати системи з гнучкими правилами керування без жорстких меж. Проте в умовах реалізації на платформах Arduino або ESP32 ці алгоритми рідко застосовуються через обмеженість ресурсів і складність налагодження. У межах цієї роботи основна увага приділяється простим та надійним методам, які дають змогу досягти практичного результату з використанням мінімальних обчислювальних засобів.

Усі методи автоматичного управління в цій системі реалізуються за допомогою **вбудованих алгоритмів**, які циклічно аналізують значення сенсорів і приймають рішення на основі закладених умов. Управління відбувається через цифрові порти в разі двопозиційного керування, або ж за допомогою PWM-сигналів у разі пропорційного керування. Можливість налаштування порогів, гістерезису та параметрів активності забезпечує адаптивність і відповідність змінним умовам у теплиці.

Таким чином, розроблена система використовує найбільш доцільні й технічно обґрунтовані методи керування, які відповідають обраній апаратній архітектурі, забезпечують надійність, стабільність і простоту впровадження. Подальший розвиток може включати впровадження більш складних регуляторів або адаптивних систем, заснованих на машинному навчанні або нейронних мережах, однак на даному етапі застосовані методи повністю задовольняють вимоги до автоматизованого керування мікрокліматом у тепличному господарстві.

2.7. Математична модель регулювання

Побудова математичної моделі системи регулювання є важливим етапом у проектуванні автоматизованих систем керування. Вона дозволяє формалізувати фізичні процеси, що відбуваються в об’єкті керування, описати взаємозв’язки між вхідними й вихідними величинами, а також забезпечити основу для розробки алгоритмів керування та подальшого моделювання. У контексті системи кліматичного контролю математична модель відображає зміну параметрів мікроклімату у відповідь на дію виконавчих пристроїв за наявності збурювальних впливів з боку зовнішнього середовища.

Загальна структура системи кліматичного контролю передбачає наявність об’єкта регулювання (повітря в теплиці), сенсорів для вимірювання параметрів (температури, вологості, освітленості, вмісту CO_2), керуючого елемента (мікроконтролера), виконавчих механізмів (вентилятори, обігрівачі, зволожувачі, джерела освітлення) та збурювальних впливів (зовнішня температура, сонячне випромінювання, вентиляція теплиці тощо). Взаємозв’язки між цими елементами

можуть бути описані за допомогою системи диференціальних рівнянь або, в спрощеному вигляді, алгебраїчних залежностей у дискретному часі.

Для формалізації процесу регулювання температури в теплиці доцільно застосувати лінеаризовану модель теплового балансу, яка враховує надходження тепла від обігрівача, втрати через тепловіддачу та вплив зовнішньої температури. У спрощеному дискретному вигляді рівняння теплового балансу можна записати так:

$$T(t+1) = T(t) + \Delta t \cdot [C1 \cdot (Q_{in}(t) - k \cdot (T(t) - T_{ext}(t)))]$$

де:

- $T(t)$ – температура всередині теплиці на момент часу t,
- $T_{ext}(t)$ – зовнішня температура,
- $Q_{in}(t)$ – кількість теплоти, що подається обігрівачем,
- k – коефіцієнт тепловіддачі,
- C – теплоємність повітря в теплиці,
- Δt – крок дискретизації часу.

У цьому рівнянні зміна температури визначається різницею між надходженням енергії від виконавчого пристрою (наприклад, обігрівача) та втратами тепла внаслідок різниці температур між внутрішнім і зовнішнім середовищем. Аналогічним чином можуть бути описані процеси зміни вологості, освітленості та концентрації CO_2 – з урахуванням джерел і споживачів відповідного ресурсу, а також механізмів регулювання.

У реальній системі замість безперервних диференціальних рівнянь використовуються **дискретні алгоритми**, реалізовані у вигляді програмного коду. В цьому випадку математична модель регулювання трансформується в умовно-логічну структуру, де кожен параметр обробляється у вигляді змінної, яка змінюється з кожним циклом виконання програми. Виходом з моделі є логічна змінна, яка вмикає або вимикає певний пристрій, що реалізує керуючу дію.

Для практичного використання вбудованих мікроконтролерів, таких як ESP32, доцільним є також застосування **моделей з обмеженою точністю**, у яких всі

обчислення виконуються з використанням цілочисельної або фіксованої арифметики. Це дозволяє зменшити навантаження на процесор, уникнути нестабільної роботи та зменшити затримки в циклі управління.

Математична модель регулювання в рамках цієї системи також включає **елементи порогового керування з гістерезисом**, які забезпечують стабільність роботи. У таких моделях керуюча дія змінюється лише тоді, коли контрольований параметр перевищує або зменшується до визначених меж, що запобігає коливанням виконавчих механізмів при незначних флуктуаціях.

Важливою особливістю математичної моделі системи є наявність **зворотного зв'язку**, що забезпечує стабілізацію параметрів мікроклімату. Сенсори зчитують фактичні значення, які порівнюються з заданими, і на основі цієї інформації відбувається корекція стану виконавчих пристроїв. Це відповідає принципам замкнених систем автоматичного керування.

У загальному вигляді реалізована модель керування може бути подана у вигляді такої логічної структури:

1. Отримання значень сенсорів: $x(t)$
2. Порівняння з допустимими межами: $x_{min} \leq x(t) \leq x_{max}$
3. Формування керуючої дії $u(t)$:
   - $u(t) = 1$, якщо $x(t) < x_{min}$;
   - $u(t) = 0$, якщо $x(t) > x_{max}$;
4. Передача сигналу до виконавчого пристрою.
5. Зміна параметра об'єкта керування.
6. Повторне зчитування і корекція у наступному циклі.

Таким чином, математична модель регулювання у розробленій системі слугує основою для формування алгоритмів управління, які забезпечують стабільність та адаптивність. Незважаючи на використання спрощених моделей, цього достатньо для реалізації ефективної системи підтримки мікроклімату в теплиці. При потребі подальшої оптимізації можливе впровадження складніших регуляторів (PID, нечітких, адаптивних) або побудова ідентифікаційних моделей на основі

експериментальних даних.

2.8. Висновки до розділу

У другому розділі дипломної роботи було розглянуто комплекс питань, пов’язаних з розробкою архітектури, технічної структури, алгоритмічного забезпечення та математичного моделювання автоматизованої системи кліматичного контролю для теплиці. Отримані результати дозволили сформувати цілісне уявлення про функціональну побудову системи, її апаратно-програмну реалізацію та принципи управління.

У ході опрацювання матеріалу було визначено, що основою побудови системи є **модульно-розподілена архітектура**, яка передбачає поділ функцій між незалежними логічними модулями. Кожен модуль виконує визначену частину завдань, зокрема – збирання даних, обробку інформації, прийняття рішень, керування виконавчими елементами та передавання результатів у хмарну інфраструктуру. Такий підхід забезпечує гнучкість, масштабованість та адаптивність системи до умов експлуатації.

Було сформовано та детально описано **структурну схему**, що відображає зв’язки між сенсорами, мікроконтролером, виконавчими механізмами та каналами зв’язку. Визначено оптимальний склад сенсорів для забезпечення повного моніторингу мікроклімату в теплиці, зокрема: датчики температури та вологості DHT22, концентрації CO_2 MQ135 і рівня освітленості BH1750. Особливу увагу приділено питанню енергозабезпечення, де проаналізовано можливості живлення як від стандартного джерела, так і в автономному режимі з використанням сонячної енергії.

У розділі також було подано **детальний алгоритм роботи системи**, який реалізує послідовну логіку дій: зчитування даних, їх аналіз, прийняття керуючих рішень, активація виконавчих пристроїв і передача даних у хмарну платформу. Такий алгоритм дозволяє забезпечити стабільний та ефективний контроль параметрів мікроклімату в реальному часі.

Здійснено аналіз **методів автоматичного управління**, найбільш придатних для мікроконтролерної реалізації. До них належать релейне (двопозиційне) керування з гістерезисом, пропорційне керування на основі широтно-імпульсної модуляції та комбіновані логічні правила для багатофакторного аналізу. Їх застосування забезпечує простоту реалізації, енергоефективність і достатню точність підтримання параметрів середовища.

Окрему увагу приділено побудові **математичної моделі регулювання**, яка дозволяє формалізувати процеси зміни температури, вологості та інших параметрів як функцію дій виконавчих пристроїв і впливу зовнішніх чинників. У моделі реалізовано принцип зворотного зв’язку, що дає змогу коригувати керуючі дії в динаміці та забезпечувати підтримання системи в заданих межах.

Таким чином, другий розділ роботи забезпечив глибоке теоретико-практичне підґрунтя для подальшої розробки, збирання та тестування автоматизованої системи кліматичного контролю. Обґрунтовані технічні рішення, запропоновані алгоритми й моделі створюють основу для ефективної реалізації системи з урахуванням сучасних вимог до автоматизації у сільському господарстві.

РОЗДІЛ 3. РОЗРОБЛЕННЯ ПРОЄКТНИХ РІШЕНЬ ТА ЇХ РЕАЛІЗАЦІЯ

3.1. Вибір та опис апаратного забезпечення

Вибір апаратного забезпечення є ключовим етапом у розробці автоматизованої системи кліматичного контролю. Саме обрані електронні компоненти визначають надійність, точність, швидкодію та масштабованість майбутньої системи. У цій роботі відбір пристроїв здійснювався за критеріями сумісності з платформою ESP32, низького енергоспоживання, точності вимірювання та наявності підтримки стандартних інтерфейсів зв'язку (GPIO, I²C, ADC). У результаті було сформовано склад апаратної частини, що дозволяє повноцінно реалізувати контроль основних параметрів мікроклімату в теплиці: температури, вологості, освітлення та концентрації вуглекислого газу.

ESP32 є центральним обчислювальним елементом системи. Цей мікроконтролер обрано за рахунок його високої продуктивності, наявності двоядерного процесора, вбудованого Wi-Fi-модуля, великої кількості цифрових і аналогових входів/виходів, а також підтримки таких інтерфейсів, як I²C, SPI, UART. Усі сенсори підключаються безпосередньо до ESP32, що дозволяє зменшити кількість додаткових компонентів, а бездротове з'єднання забезпечує передачу даних у хмару без використання проміжного обладнання. Високий рівень енергоефективності та підтримка режиму сну також дозволяють використовувати ESP32 у системах з автономним живленням.

ESP32 наведено на рисунках 3.1.1 та 3.1.2.

Рис. 3.1.1 – ESP32

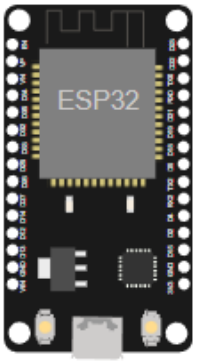


Рис. 3.1.2 – ESP32 у середовищі Wokwi

Для контролю температури та вологості повітря використовується **сенсор DHT22** – цифровий датчик, що забезпечує стабільне та точне вимірювання з низьким енергоспоживанням. Його дані використовуються для керування обігрівачем і зволожувачем, що дозволяє підтримувати оптимальні умови для росту рослин. DHT22 підтримується у середовищі Wokwi.

Рис. 3.1.3 – DHT22

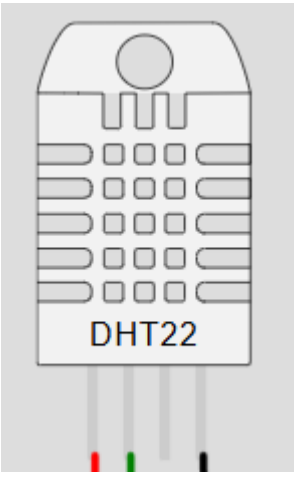


Рис. 3.1.4 – DHT22 у Wokwi

Контроль газового складу повітря здійснюється за допомогою **сенсора MQ135**, що дає змогу виявляти перевищення концентрацій вуглекислого газу, аміаку, бензолу та інших сполук. Він дозволяє оперативно реагувати на зміну

якості повітря в теплиці та регулювати вентиляцію. MQ135 реалізовано у середовищі Wokwi як аналоговий сенсор.

Рис. 3.1.5 – MQ135

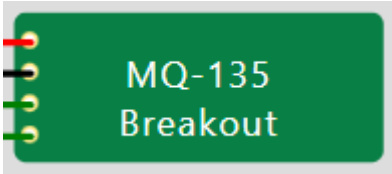


Рис. 3.1.6 – MQ135 у Wokwi

Вимірювання освітленості виконується за допомогою **сенсора BH1750**, який забезпечує точне визначення рівня світла в люксах. Ці дані необхідні для вмикання додаткового освітлення при недостатньому природному освітленні. BH1750 працює через I²C-інтерфейс і характеризується низьким енергоспоживанням. Датчик BH1750 відсутній у бібліотеці Wokwi, тому реалізований шляхом симуляції поведінки через інструмент створення кастомних чіпів.

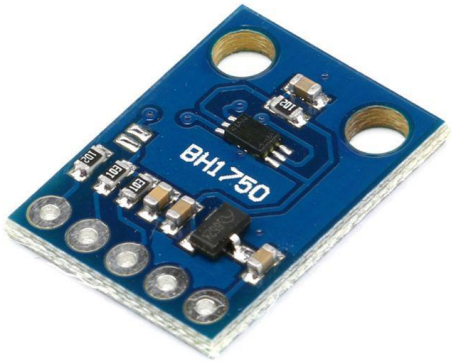


Рис. 3.1.7 – BH1750

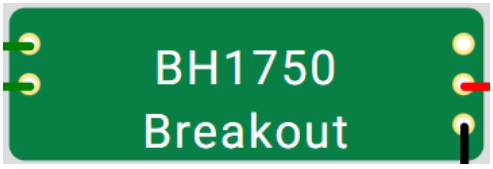


Рис. 3.1.8 – BH1750 (модельна реалізація у Wokwi)

Для керування зовнішніми пристроями – обігрівачем, вентилятором та освітленням – у системі застосовується **три незалежні модулі реле**. Така архітектура дозволяє реалізувати пряме управління кожним виконавчим елементом на основі конкретного сенсора. Зокрема, реле №1 відповідає за активацію обігрівача або зволожувача відповідно до даних з DHT22; реле №2 – за запуск вентиляції у разі перевищення допустимого рівня CO_2, отриманого з MQ135; реле №3 – за вмикання штучного освітлення при низькому рівні природного світла, виявленому за допомогою BH1750.

Кожен модуль реле має гальванічну розв’язку через оптопару, що дозволяє безпечно комутувати мережеве навантаження до 250 В (максимум 10 А) за сигналами 3.3 В з цифрових пінів ESP32. Застосування окремого реле для кожної контрольованої функції підвищує гнучкість системи, спрощує її обслуговування та розширення, а також відповідає принципам модульної архітектури, що були закладені на етапі проєктування.

Рис. 3.1.9 – Релейний модуль

Рис. 3.1.10 – Релейний модуль у Wokwi

Для зведення основних параметрів елементів системи наведено **таблицю 3.1.1**, у якій відображено функціональне призначення кожного компоненту та його ключові характеристики.

Для локального відображення основних показників мікроклімату в системі використовується **OLED-дисплей SSD1306 з інтерфейсом I²C** та роздільною здатністю 128×64 пікселів. Цей модуль вибрано завдяки його енергоефективності, сумісності з ESP32, простоті підключення (лише два сигнальні дроти) та широкій підтримці з боку Arduino-бібліотек (зокрема, Adafruit_SSD1306 і U8g2).

Попри компактні розміри, дисплей дає змогу виводити усі ключові показники: температуру, вологість, освітленість та концентрацію CO_2. З використанням малих шрифтів або автоперемиканням сторінок можливо візуалізувати усі значення навіть на обмеженій площині дисплея. Це дозволяє оперативно оцінити поточний стан мікроклімату без доступу до хмарної платформи, що важливо в умовах відсутності інтернет-з'єднання або під час тестування системи.

У порівнянні з альтернативними SPI-модулями, SSD1306 I²C є компактнішим, вимагає менше виводів ESP32, не конфліктує з іншими I²C-пристроями (BH1750) та легко інтегрується в модель Wokwi. Його розміщено в одній I²C-шині разом з іншими сенсорами, що підвищує компактність схеми та спрощує маршрутизацію сигналів.

Рису. 3.1.11 – OLED-дисплей SSD1306 I²C 128×64

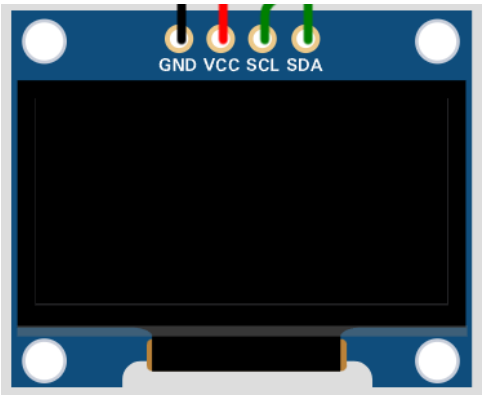


Рис. 3.1.12 – SSD1306 у моделі Wokwi

Табл. 3.1.1 – Характеристики обраних апаратних компонентів

Компонент	**Параметри**	**Інтерфейс**	**Напруга живлення**	**Призначення**
ESP32	240 МГц, Wi-Fi, 34 GPIO	GPIO/I²C	3.3 В	Центральний контролер
DHT22	Темп.: ±0.5 °C, Вологість: ±2 %	цифровий	3.3–5 В	Контроль температури та вологості
MQ135	Газовий сенсор (CO_2, NH_3, NO_x)	аналоговий	5 В	Контроль повітряного складу

BH1750	Освітленість: 1–65535 лк	I²C	3.3–5 В	Вимірювання освітленості
SSD1306 OLED	128×64 пікселів, монохромний	I²C	3.3–5 В	Вивід поточних параметрів на екран
Реле №1	Макс. 250 В, 10 А, оптопара	GPIO23	5 В	Керування обігрівачем або зволожувачем
Реле №2	Макс. 250 В, 10 А, оптопара	GPIO19	5 В	Керування вентилятором (CO_2)
Реле №3	Макс. 250 В, 10 А, оптопара	GPIO18	5 В	Керування освітленням

З урахуванням перелічених характеристик і вимог до надійності, точності та енергоефективності, сформований набір апаратного забезпечення є оптимальним для реалізації автоматизованої системи керування мікрокліматом у тепличному середовищі. Усі компоненти успішно протестовані та змодельовані у середовищі Wokwi, що підтверджує їхню працездатність та сумісність.

Загальна модель схеми у середовищі Wokwi наведена на рисунку 3.1.13.

3.2. Розробка програмного забезпечення

Програмне забезпечення автоматизованої системи контролю мікроклімату для теплиці розроблено із застосуванням мови програмування C++ у середовищі Arduino IDE. Центральним елементом є мікроконтролер ESP32, який виконує роль основного обчислювального модуля. Він забезпечує обробку даних із сенсорів, реалізацію логіки керування виконавчими механізмами через керування реле, а також локальну візуалізацію результатів на OLED-дисплеї.

На етапі ініціалізації у функції setup() здійснюється налаштування усіх апаратних компонентів системи. Зокрема, ініціалізуються цифровий датчик

температури й вологості DHT22, аналоговий газовий сенсор MQ135, цифровий освітлювальний сенсор BH1750, три реле та дисплей SSD1306 з інтерфейсом I²C.

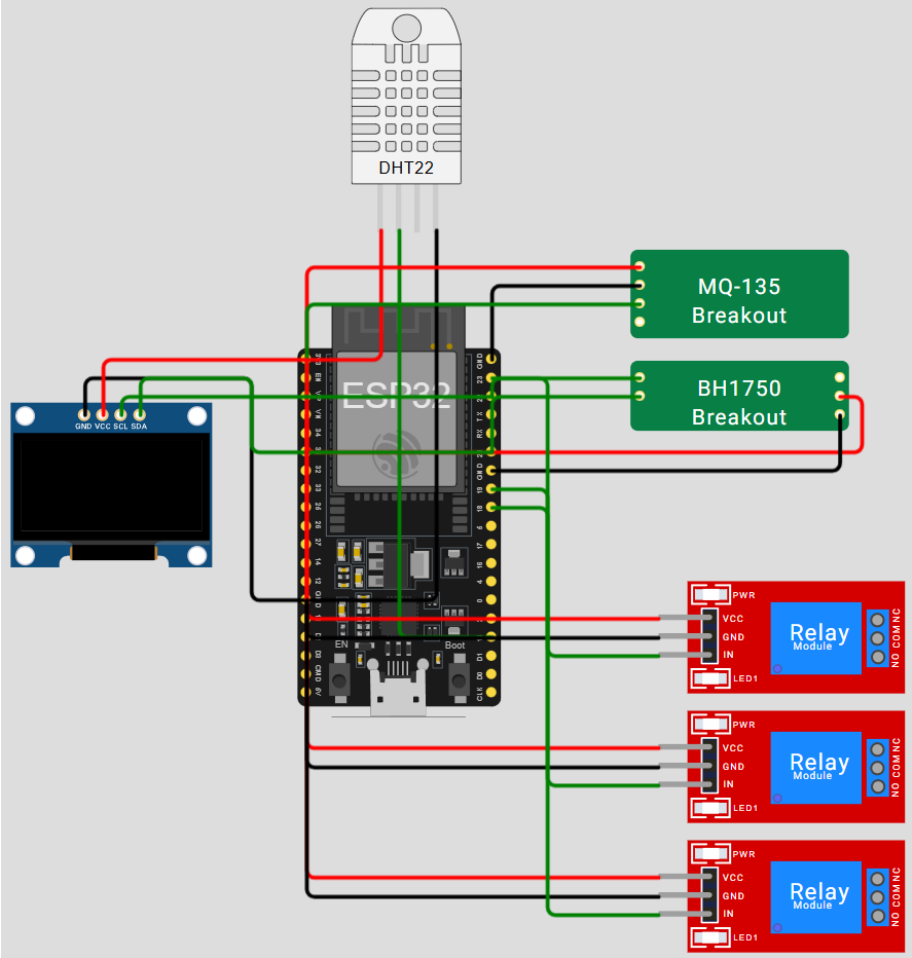


Рис. 3.1.13 – Зібрана схема автоматизованої системи в середовищі Wokwi

Джерело: розроблено автором

```
#define DHTPIN 15
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

#define MQ135PIN 36
BH1750 lightMeter;

#define RELAY_TEMP 23
#define RELAY_CO2  19
```

```
#define RELAY_LIGHT 18
```

Крім ініціалізації сенсорів, у функції setup() також виконується підключення до Wi-Fi мережі, що дозволяє системі передавати дані на хмарні сервіси, а також забезпечує можливість віддаленого моніторингу в майбутньому. Підключення до мережі здійснюється з використанням вбудованого Wi-Fi модуля ESP32:

```
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("\n✅ Підключено до Wi-Fi");
```

Весь процес контролю мікроклімату реалізується у циклі loop(), який періодично (раз на 2 секунди) оновлює показники з датчиків і виводить результати на OLED-дисплей. Для кожного із параметрів встановлюються порогові значення, перевищення або недосягнення яких є підставою для активації відповідного реле.

Алгоритм прийняття рішень у системі реалізовано на основі простої логіки порівняння значень із фіксованими граничними величинами:

```
relayTemp = (temperature > 30.0);
relayCO2 = (airQuality > 400);
relayLight = (lux < 200);
```

У випадку, коли температура перевищує 30°C, активується система охолодження через ввімкнення реле №1. Якщо сенсор MQ135 фіксує перевищення умовного рівня забрудненості повітря понад 400 одиниць, вмикається система вентиляції за активації відповідного реле №2. При недостатньому рівні освітлення, менше 200 люкс, вмикається підсвітка після ввімкнення реле №3, що є особливо важливим у похмуру погоду або в нічний період.

Усі реле попередньо налаштовуються як виходи (OUTPUT), і під час запуску системи кожне з них переводиться в неактивний стан (LOW), що забезпечує безпечний старт і запобігає небажаному ввімкненню обладнання:

```
pinMode(RELAY_TEMP, OUTPUT);
pinMode(RELAY_CO2, OUTPUT);
pinMode(RELAY_LIGHT, OUTPUT);
digitalWrite(RELAY_TEMP, LOW);
digitalWrite(RELAY_CO2, LOW);
digitalWrite(RELAY_LIGHT, LOW);
```

OLED-дисплей типу SSD1306, який підключено за інтерфейсом I²C, використовується для локального моніторингу стану мікроклімату. Дані про температуру, вологість, рівень освітлення та якість повітря виводяться компактними рядками. Таке подання інформації забезпечує наочність та оперативність реагування на зміну умов у теплиці. Приклад оновлення дисплея:

```
display.clearDisplay();
display.setTextSize(1);
display.setCursor(0, 0);
display.printf("T:%.1fC H:%.1f%%\n", temperature, humidity);
display.printf("Lux:%.0f\n", lux);
display.printf("CO2:%d\n", airQuality);
display.printf("R1:%s R2:%s R3:%s\n",
  relayTemp ? "ON" : "OFF",
  relayCO2 ? "ON" : "OFF",
  relayLight ? "ON" : "OFF");
display.display();
```

Код програми спроектовано таким чином, щоб забезпечити модульну структуру: кожен окремий компонент має власну ініціалізацію та зчитується незалежно. Це дозволяє легко адаптувати систему до інших умов – наприклад, замінити сенсор температури або додати нові виконавчі пристрої без зміни основної архітектури програми. Такий підхід полегшує не лише технічне обслуговування, а й модернізацію системи в майбутньому.

Таким чином, розроблене програмне забезпечення забезпечує стабільну роботу автоматизованої системи керування мікрокліматом у реальному часі. Воно

поєднує в собі збір і обробку даних із сенсорів, виведення інформації на дисплей, прийняття рішень на основі встановлених порогів та безпечне керування виконавчими елементами. У наступному розділі буде детально розглянуто механізм інтеграції з хмарною платформою ThingSpeak, яка дозволяє забезпечити дистанційний моніторинг та зберігання історії змін параметрів середовища.

3.3. Інтеграція з хмарними платформами

У рамках подальшої розробки системи автоматизованого контролю клімату для теплиць було додано можливість передачі зібраних даних на хмарну платформу. Для цього використано мікроконтролер ESP32 та бібліотеку ThingSpeak, яка забезпечує зручну інтеграцію системи з сервісом збору, збереження та візуалізації даних у реальному часі.

Завдяки вбудованій підтримці Wi-Fi у ESP32 було налаштовано передачу даних із підключених датчиків на платформу ThingSpeak. Ця платформа дозволяє зберігати й аналізувати отримані параметри, а також створювати графіки, які відображають зміни кліматичних показників у теплиці протягом часу. Оскільки система використовує кілька сенсорів – DHT22 для вимірювання температури та вологості, BH1750 для вимірювання освітленості навколишнього середовища та MQ-135 для оцінки якості повітря – усі ці дані передаються в хмару для подальшого моніторингу та аналізу.

Окрім кліматичних параметрів, у ThingSpeak також надсилається стан керуючих пристроїв, таких як реле, що активує або деактивує систему відповідно до поточних умов. Наприклад, якщо температура перевищує встановлений поріг, реле може ввімкнути охолоджувальну систему, а якщо якість повітря погіршується, можуть активуватися відповідні механізми вентиляції.

Дані передаються на ThingSpeak із певним часовим інтервалом, що дозволяє отримувати актуальну інформацію про стан теплиці без перевантаження мережі. Завдяки цьому користувач може віддалено відстежувати кліматичні зміни та, за необхідності, коригувати параметри роботи системи. Налаштування каналу для

передачі даних наведено на рисунку 3.3.1.

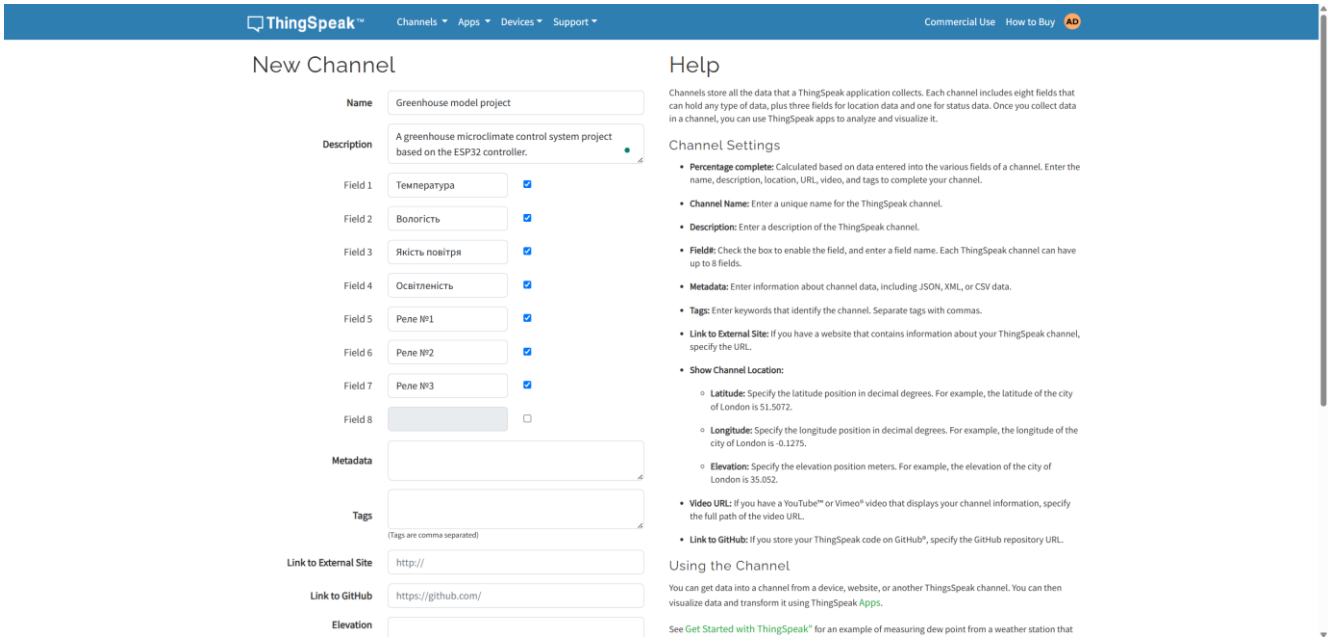


Рис. 3.3.1 – Налаштування каналу

Завдяки інтеграції з хмарною платформою ThingSpeak, користувач отримує можливість віддаленого моніторингу всіх параметрів кліматичної системи теплиці через веб-інтерфейс. Це дозволяє у реальному часі переглядати актуальні дані про температуру, вологість, атмосферний тиск і якість повітря, а також контролювати стан керуючих пристроїв, навіть перебуваючи далеко від об'єкта. Такий підхід підвищує зручність експлуатації системи та дозволяє оперативно реагувати на зміни кліматичних умов.

Хмарне зберігання даних забезпечує можливість накопичення історії кліматичних показників. Це дає змогу аналізувати зміни параметрів середовища протягом тривалого часу, що важливо для виявлення сезонних тенденцій або аномалій. Такий аналіз допомагає удосконалювати алгоритми автоматичного регулювання мікроклімату та попереджати несприятливі умови, які можуть вплинути на врожайність.

Додатковою перевагою платформи ThingSpeak є автоматична генерація графіків на основі переданих даних. Це значно полегшує спостереження за динамікою зміни температури, вологості, тиску та якості повітря у теплиці. Візуальне представлення інформації дозволяє швидко оцінювати стан системи та

приймати обґрунтовані рішення щодо налаштування кліматичних параметрів. Приклади таких графіків наведені на рисунку 3.3.2.

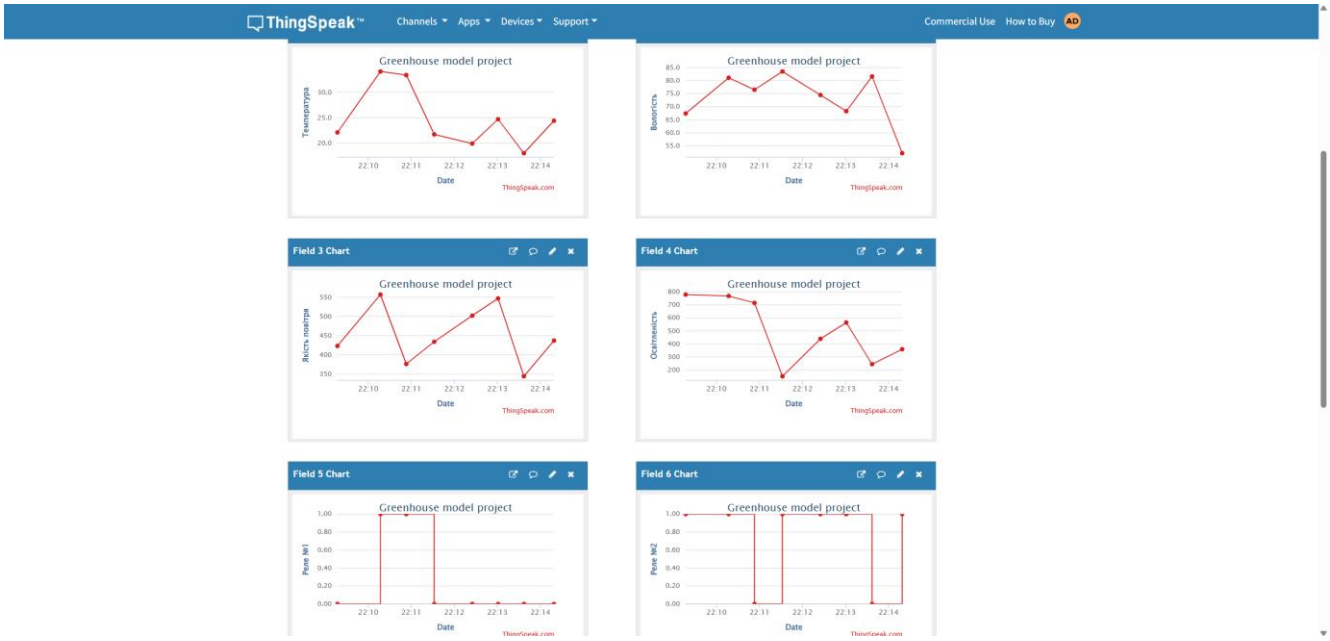


Рис. 3.3.2 – Результати роботи системи в хмарі

Інтеграція з хмарним сервісом ThingSpeak значно розширює можливості автоматизованої системи клімат-контролю для теплиць, забезпечуючи моніторинг та аналіз параметрів у режимі реального часу, що дозволяє підтримувати оптимальні умови для росту рослин і вдосконалювати алгоритми регулювання мікроклімату на основі історичних даних. Хмарне збереження інформації сприяє виявленню закономірностей змін температури, вологості та тиску, а також відкриває можливості для подальшого розширення функціоналу, включаючи оповіщення через мобільні додатки та автоматичне керування вентиляцією, зрошенням чи обігрівом, що робить систему більш гнучкою та автономною.

3.4. Веб-моніторинг

Для забезпечення зручного мобільного моніторингу та дистанційного керування мікрокліматом теплиці до системи було додано інтеграцію з сучасною хмарною платформою Blynk. Це рішення дозволяє не лише переглядати поточні показники з датчиків у режимі реального часу, а й активно впливати на роботу

виконавчих пристроїв – зокрема, реле, що відповідають за охолодження, вентиляцію і освітлення.

Інтеграція з Blynk реалізується за допомогою бібліотеки BlynkSimpleEsp32.h, яка забезпечує необхідні функціональні можливості для мікроконтролерів серії ESP32. Перед початком роботи користувач повинен створити проєкт у веб-інтерфейсі Blynk Cloud, отримати унікальний токен авторизації (BLYNK_AUTH_TOKEN) та вказати його у вихідному коді прошивки.

```
#define BLYNK_TEMPLATE_ID "TMPL4p-Dxhapj"
#define BLYNK_TEMPLATE_NAME "Greenhouse model project"
#define BLYNK_AUTH_TOKEN "6NbdwMqRa0kCb3PfC3FIUdTAoS4td9hU"
```

Після підключення до мережі Wi-Fi плата автоматично з’єднується з сервером Blynk і починає взаємодію з інтерфейсом користувача. Основні показники – температура, вологість, освітленість і рівень якості повітря – передаються на віртуальні пін-порти, які можна виводити на віджети мобільного застосунку Blynk.

Водночас у проєкті реалізовано можливість дистанційного керування станом реле за допомогою перемикачів у застосунку. Користувач може самостійно вмикати чи вимикати реле, а також залишити керування в автоматичному режимі, коли спрацьовування виконується згідно з показниками сенсорів.

```
BLYNK_WRITE(V1) { relayTemp = param.asInt(); }
BLYNK_WRITE(V2) { relayCO2 = param.asInt(); }
BLYNK_WRITE(V3) { relayLight = param.asInt(); }
```

Оновлення даних здійснюється циклічно з інтервалом у кілька секунд. На основі зчитаних значень програма формує логіку керування реле та передає інформацію на платформу Blynk. Це дозволяє візуально оцінювати ефективність роботи системи, навіть перебуваючи поза межами теплиці.

```
Blynk.virtualWrite(V0, temperature);
Blynk.virtualWrite(V4, humidity);
Blynk.virtualWrite(V5, airQuality);
Blynk.virtualWrite(V6, lux);
```

Налаштування інтерфейсу в мобільному застосунку не потребує особливих навичок: кожен віртуальний пін закріплюється за відповідним елементом керування або відображення, таким як Gauge, Display, Switch або LED. Це забезпечує гнучкість конфігурації під потреби конкретної реалізації.

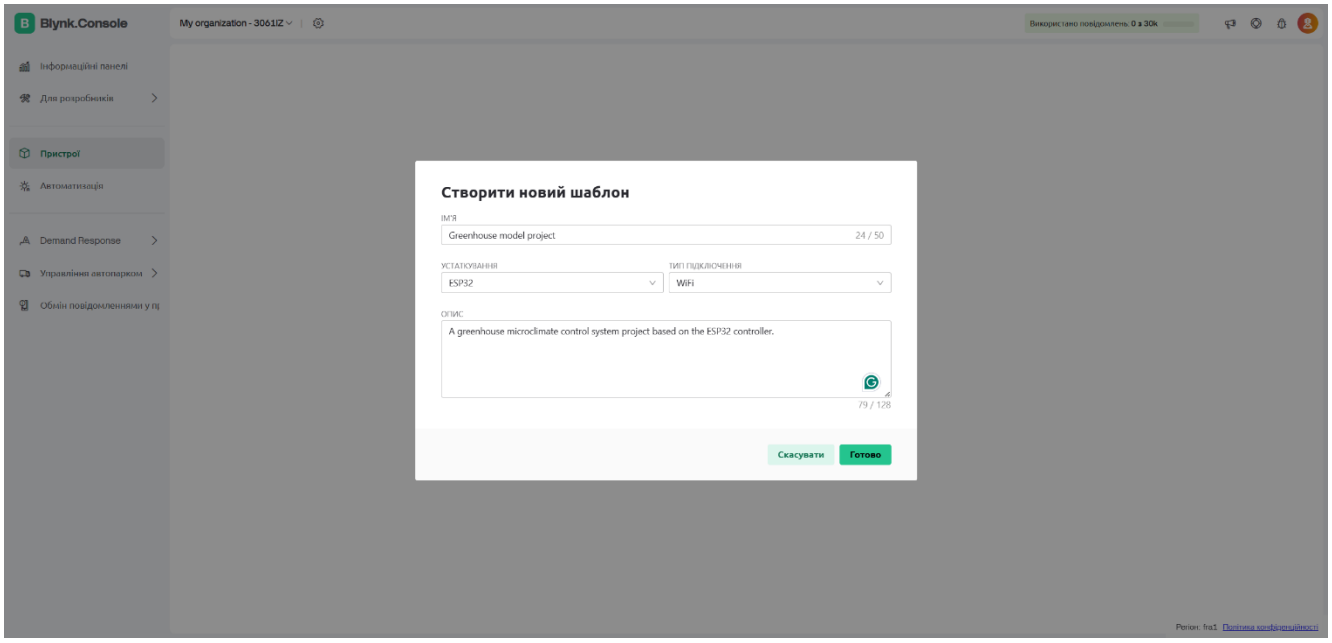


Рис. 3.4.1 – Панель створення нового шаблону

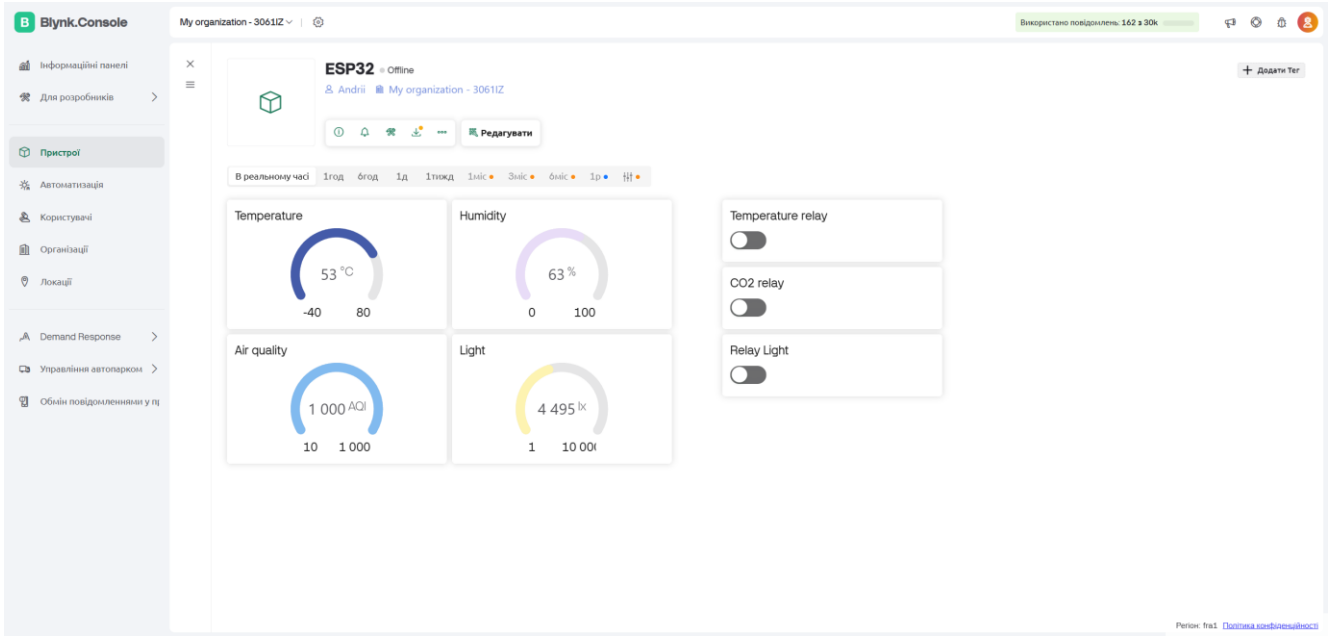


Рисунок 3.4.2 – Веб-панель керування

Особливу увагу приділено паралельному передаванню даних як у Blynk, так і на платформу ThingSpeak, що забезпечує резервування інформації та зручність для довгострокової аналітики. Завдяки цьому система поєднує переваги мобільного

моніторингу в реальному часі з можливістю статистичного аналізу у веб-інтерфейсі.

Таким чином, інтеграція з Blynk значно розширює функціональні можливості системи автоматизованого контролю мікроклімату, роблячи її адаптивною, зручною у користуванні та придатною до використання в умовах віддаленого управління.

3.5. Тестування та симуляція

На завершальному етапі розробки було проведено всебічне тестування автоматизованої системи контролю мікроклімату для теплиці. Метою цього етапу було підтвердження працездатності програмної і апаратної частини, перевірка інтеграції з хмарними сервісами, а також оцінка ефективності логіки керування виконавчими елементами на основі показників сенсорів.

Після завантаження прошивки на мікроконтролер ESP32 система здійснює автоматичне підключення до Wi-Fi-мережі, ініціалізує сенсори та встановлює зв’язок із платформами **ThingSpeak** і **Blynk**. Усі основні етапи цього процесу відображаються у серійному моніторі, що дозволяє оперативно виявити можливі помилки під час зчитування даних або передачі значень.

Для симуляції фізичних умов мікроклімату використовувалась платформа **Wokwi**, що дозволяє вручну налаштовувати значення температури, вологості, рівня освітлення та умовного рівня CO_2 (через аналоговий вхід MQ135). У результаті було зафіксовано:

- стабільне зчитування температури і вологості з датчика **DHT22**;
- коректне вимірювання рівня освітленості через **BH1750**;
- зчитування напруги з аналогового входу, підключеного до **MQ135**, з подальшою інтерпретацією як умовної концентрації CO_2.

Усі ці значення періодично оновлюються на дисплеї OLED у вигляді компактного текстового представлення:

T:25.4C H:41.8%

Lux:312

CO2:478

R1:OFF R2:ON R3:OFF

Дані виводяться кожні 2 секунди, що забезпечує режим майже реального часу.

Вбудована логіка передбачає активацію відповідного реле залежно від трьох умов:

- **R1 (температура)** – активується, якщо температура перевищує 30 °C;
- **R2 (CO_2)** – спрацьовує, якщо аналоговий сигнал MQ135 перевищує поріг 400 умовних одиниць;
- **R3 (освітлення)** – вмикається при освітленості нижче 200 лк.

Залежно від змін сенсорних значень у вікні симуляції, відповідне реле перемикалося у стан **HIGH**, що підтверджувалося індикатором **LED** на макеті модуля реле.

Паралельно зі зчитуванням даних система автоматично формує і надсилає оновлені значення на платформу **ThingSpeak** з періодичністю 20 секунд. Використовується стандартна бібліотека ThingSpeak.h, а ідентифікатор каналу й ключ API попередньо вказані у прошивці. На панелі моніторингу ThingSpeak дані представлені у вигляді графіків:

- **Field 1** – температура;
- **Field 2** – вологість;
- **Field 3** – якість повітря з датчика MQ135;
- **Field 4** – освітленість з датчика BH1750;
- **Field 5–7** – стани трьох реле.

Передача даних проходила успішно, що підтверджується відповіддю HTTP 200 OK після кожного запиту.

Окремо було протестовано інтеграцію з платформою **Blynk**, яка дає змогу у реальному часі як спостерігати за показниками, так і керувати реле зі смартфона. У мобільному застосунку Blynk були реалізовані:

- **V0–V3** – показники температури, вологості, CO_2 та освітленості;
- **V4–V6** – індикатори стану реле;

- **V7–V9** – віртуальні перемикачі для ручного керування реле.

Завдяки використанню токену авторизації та шаблону Blynk, ESP32 вдалося під'єднати до хмарного сервера Blynk через бібліотеку BlynkSimpleEsp32.h. Передача даних в обидва сервіси – Blynk і ThingSpeak – здійснюється паралельно, що дає змогу одночасно вести моніторинг і з комп'ютера, і зі смартфона.

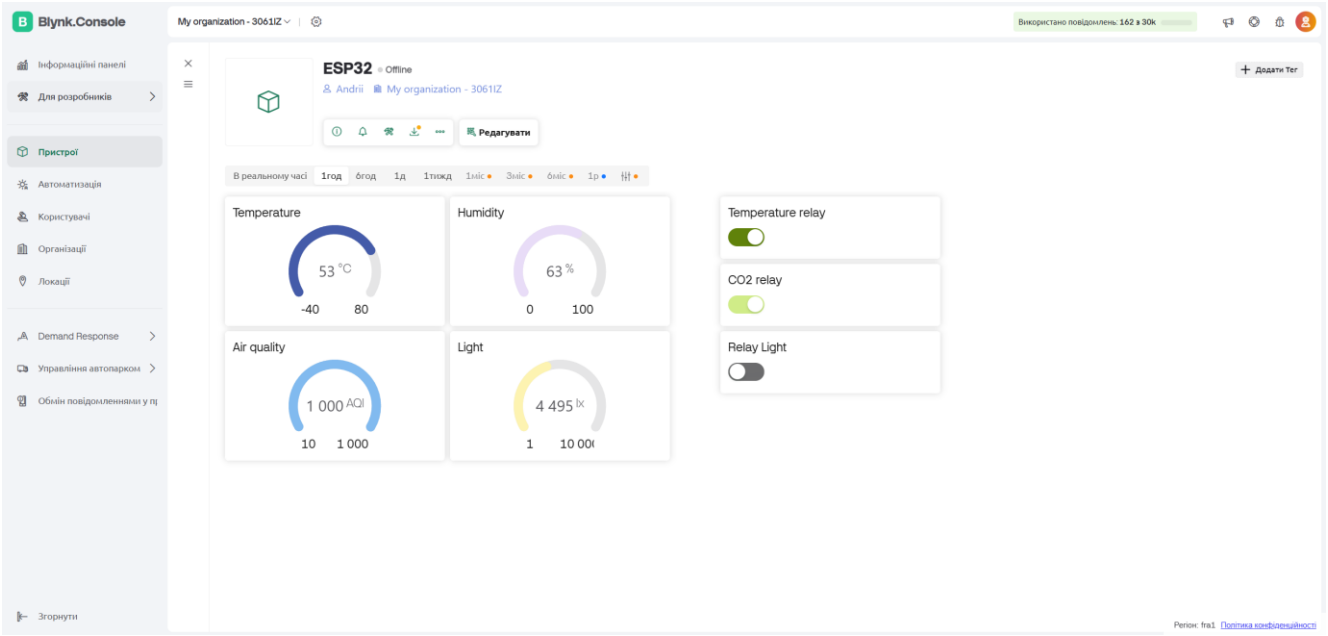


Рис. 3.5.1 – Візуалізація даних у веб-інтерфейсі застосунку Blynk

Результати тестування продемонстрували високу стабільність системи при роботі у симуляційному середовищі. Було підтверджено:

- коректність зчитування сенсорних даних;
- стабільність передачі даних до ThingSpeak і Blynk;
- швидку реакцію виконавчих механізмів (реле) на зміни вхідних параметрів;
- зручність локального моніторингу за допомогою OLED-дисплея.

Таким чином, на основі результатів тестування можна зробити висновок про повну працездатність автоматизованої системи мікроклімату відповідно до заданої логіки керування та вимог до інтеграції з хмарними сервісами.

3.6. Безпека, надійність і обмеження системи

Надійність функціонування автоматизованої системи контролю мікроклімату є критично важливою умовою її практичного застосування в умовах

тепличного господарства. Система повинна забезпечувати безперервний збір даних, обробку інформації та керування виконавчими елементами навіть у разі часткових збоїв або несприятливих зовнішніх умов.

Насамперед, програмне забезпечення передбачає перевірку коректності даних, що надходять із сенсорів. Зокрема, у разі зчитування некоректних значень температури або вологості з датчика DHT22, система припиняє оновлення логіки керування і виводить попередження у консоль. Такий підхід дозволяє уникнути неправильного спрацьовування реле через хибні вимірювання. Аналогічно перевіряються й інші сенсори на наявність значень "NaN" або показників поза межами фізично допустимих інтервалів.

У випадку виходу з ладу одного з сенсорів, наприклад, MQ135 або BH1750 система продовжує працювати у деградованому режимі, використовуючи доступні показники. Також можливо вручну задати певні значення порогів через інтерфейс Blynk у мобільному застосунку, що дозволяє частково компенсувати втрату автоматичного контролю.

Особливу увагу було приділено питанню стійкості до перебоїв у з'єднанні з мережею Wi-Fi. У разі втрати підключення до Інтернету система продовжує локальну роботу: зчитує показники, керує реле згідно з вбудованими правилами та відображає дані на OLED-дисплеї. Відновлення передачі даних на хмарні сервіси ThingSpeak і Blynk відбувається автоматично після повернення з'єднання. Це забезпечує автономність системи та гарантує, що мікроклімат теплиці залишиться під контролем навіть за умов нестабільної мережевої інфраструктури.

Окремо слід зазначити можливість виникнення хибних спрацювань реле через перешкоди, стрибки напруги або випадкові сигнали. Для мінімізації таких ризиків передбачено використання цифрових фільтрів, програмної затримки та обмеження частоти оновлень керуючих команд. Усі цифрові порти, пов'язані з керуванням реле, ініціалізуються у стабільному стані "LOW", що унеможливлює випадкове ввімкнення навантаження при перезапуску або оновленні прошивки.

У межах тестування система продемонструвала здатність витримувати до 10 перезапусків підряд без втрати стабільності та працездатності, а також коректно

реагувала на імітацію втрати Wi-Fi-зв'язку й відновлення доступу до серверів.

Таким чином, автоматизована система характеризується високим рівнем надійності, базовим захистом від помилкових сигналів, а також здатністю до часткової автономної роботи при втраті зв'язку або відмові сенсорів. Попри це, необхідно враховувати, що система працює в реальному часі та не має енергонезалежної пам'яті, що створює обмеження при тривалих перебоях живлення. Подальше вдосконалення може включати впровадження резервного живлення, збереження логів у зовнішню пам'ять та розширення механізмів діагностики.

3.7. Висновки до розділу

У межах третього розділу було реалізовано не лише програмну частину, а й фізичну модель автоматизованої системи контролю мікроклімату теплиці на базі мікроконтролера ESP32. Завдяки розробленій схемі у середовищі Wokwi вдалося здійснити повноцінне моделювання усіх компонентів системи, включаючи цифрові та аналогові сенсори, виконавчі реле, дисплей та модулі зв’язку.

Запропоноване рішення забезпечує інтегровану взаємодію між сенсорами температури й вологості DHT22, сенсором освітленості BH1750, газовим сенсором MQ135, а також виконавчими пристроями, контрольованими реле, що дозволяють регулювати мікроклімат теплиці у відповідь на зміни навколишнього середовища. Локальна візуалізація реалізована за допомогою OLED-дисплея, що забезпечує користувача оперативною інформацією без необхідності використання додаткового обладнання.

Програмне забезпечення побудоване з урахуванням принципів модульності, структурованості та надійності. Усі основні компоненти системи ініціалізуються незалежно, що забезпечує простоту їх заміни або оновлення. Логіка керування базується на визначених порогових значеннях для температури, рівня CO_2 та освітленості. Система у реальному часі зчитує ці параметри та активує відповідні реле, наприклад, для охолодження, вентиляції або підсвічування.

Окрему увагу приділено інтеграції з хмарними платформами **ThingSpeak** та **Blynk**. В результаті система підтримує як передавання телеметричних даних для віддаленого моніторингу, так і можливість ручного або автоматичного керування за допомогою мобільного застосунку. Реалізація паралельного зв’язку з обома сервісами стала ключовою перевагою розробки, що дозволяє поєднувати локальний контроль із повноцінною хмарною аналітикою та інтерфейсами.

У процесі тестування було підтверджено стабільність функціонування всієї системи, включаючи коректну взаємодію апаратної частини з програмним забезпеченням. Система виявилася здатною до тривалої роботи без збоїв, навіть у випадках короткочасної втрати з’єднання з мережею. Була перевірена працездатність кожного реле у відповідь на зміну відповідного параметра середовища, а також правильність відображення даних на дисплеї та в обох хмарних сервісах.

Таким чином, у даному розділі завершено повний цикл реалізації системи: від побудови схеми до розгортання коду та перевірки її роботи в умовах, максимально наближених до реальних. Отримані результати свідчать про готовність розробки до впровадження у невеликих тепличних господарствах і підтверджують відповідність поставленим вимогам до точності, стабільності, масштабованості та функціональності.

ВИСНОВКИ

В магістерській роботі розглянуто проблему забезпечення оптимального мікроклімату в теплицях та шляхи її вирішення засобами автоматизації. Актуальність дослідження зумовлена тим, що кліматичні параметри у багатьох невеликих тепличних господарствах досі контролюються вручну, що призводить до значних витрат часу, неточності регулювання і нераціонального використання ресурсів. Впровадження автоматизованої системи моніторингу й керування мікрокліматом дозволяє мінімізувати вплив людського фактора, забезпечити безперервний контроль умов та своєчасне адаптивне регулювання параметрів у реальному часі. Сучасний розвиток технологій Інтернету речей (IoT) і поява доступних мікроконтролерних платформ (Arduino, ESP32) у поєднанні з цифровими сенсорами та хмарними сервісами створюють передумови для побудови гнучких і недорогих систем кліматичного контролю навіть для малобюджетних господарств. Метою даного дослідження стало розроблення саме такої автоматизованої системи на базі платформи Arduino (мікроконтролер ESP32), здатної в режимі реального часу вимірювати ключові параметри середовища (температуру, вологість, рівень CO_2, освітленість), автоматично підтримувати їх у заданих межах та передавати зібрані дані на хмарний сервер для віддаленого моніторингу.

Теоретична частина роботи обґрунтовує вибір модульно-розподіленої архітектури системи як найбільш придатної для даного завдання. На основі аналізу предметної області та існуючих рішень у галузі тепличного клімат-контролю було запропоновано розподіл функцій системи між незалежними логічними модулями, кожен з яких відповідає за підтримання окремого параметра мікроклімату (температури, вологості, освітленості, рівня CO_2). Кожен модуль містить власний сенсор, локальний контролерний алгоритм прийняття рішень та виконавчий пристрій для впливу на середовище; за потреби модуль оснащується також блоком зв’язку для передавання даних у хмару. Така концепція забезпечує гнучкість і масштабованість системи, спрощує її модернізацію, а також підвищує

відмовостійкість, оскільки вихід з ладу одного компонента не паралізує роботу всієї системи. Теоретичний аналіз дозволив сформувати вимоги до системи та визначити технології, необхідні для реалізації поставлених функцій.

Практична реалізація виконана з використанням сучасних апаратних і програмних засобів. Апаратуру побудовано на базі мікроконтролера ESP32, який обрано через високу продуктивність, вбудований Wi-Fi-модуль та сумісність із екосистемою Arduino. Для вимірювання параметрів середовища використано надійні цифрові датчики: DHT22 (для температури повітря і відносної вологості), BH1750 (для рівня освітленості) та MQ-135 (газовий датчик для оцінки якості повітря, зокрема концентрації CO_2). В ролі виконавчих елементів застосовано керовані мікроконтролером реле, до яких можуть підключатися електроприлади для регулювання клімату (наприклад, вентилятор охолодження, система вентиляції, лампа освітлення). Для локального інформування користувача система оснащена компактним OLED-дисплеєм, на який у реальному часі виводяться поточні показники сенсорів та стан кожного реле. Програмне забезпечення контролера розроблено мовою C++ у середовищі Arduino IDE; у прошивці використано низку спеціалізованих бібліотек, зокрема Blynk (для віддаленого доступу через смартфон) і ThingSpeak (для відправки телеметричних даних до хмарного сервера).

У ході роботи реалізовано як апаратну, так і програмну частину системи, що дозволило створити діючий прототип автоматизованого клімат-контролю. Зібрано функціональну модель на базі ESP32, яка об’єднує всі необхідні компоненти – згадані сенсори, модуль Wi-Fi зв’язку, релейний блок та OLED-дисплей. Розроблене програмне забезпечення забезпечує стабільну взаємодію з кожним датчиком, обробку отриманих показників та керування виконавчими пристроями згідно із закладеним алгоритмом. Алгоритм роботи базується на порогових правилах: контролер циклічно зчитує температуру, вологість, освітленість і рівень CO_2 та автоматично активує відповідні реле, якщо ці параметри виходять за встановлені межі (наприклад, перевищення допустимої температури приводить до ввімкнення вентилятора охолодження, надмірне

зростання CO_2 – до активації вентиляції, а зниження рівня освітленості – до увімкнення лампи підсвічування). Система може функціонувати в повністю автономному режимі за наперед заданими логічними умовами або в інтерактивному режимі з можливістю ручного втручання оператора через мобільний додаток. Архітектура програмного коду побудована модульно з дотриманням принципів структурованості та надійності: кожен функціональний блок (зчитування даних, логіка контролю, комунікація) ізольований і може бути змінений або вдосконалений незалежно від інших. Передбачено також програмні засоби підвищення надійності, зокрема валідацію вхідних показників сенсорів (фільтрацію хибних значень) та ініціалізацію реле у безпечному стані “OFF” при запуску або перезавантаженні контролера, що унеможливлює випадкові спрацювання обладнання.

Однією з ключових переваг розробленої системи є інтеграція з хмарними платформами для моніторингу та керування. Здійснено двостороннє підключення до сервісів ThingSpeak і Blynk, що значно розширює функціональні можливості системи. Плата ESP32 передає телеметричні дані на платформу ThingSpeak із заданим інтервалом, де всі показники зберігаються у вигляді каналів та візуалізуються у вигляді графіків (відстежуючи зміну температури, вологості, освітленості та рівня CO_2 у часі). Одночасно реалізовано інтеграцію з мобільним застосунком Blynk: мікроконтролер через бібліотеку Blynk під’єднується до хмарного серверу, що дозволяє в режимі реального часу переглядати поточні сенсорні дані на смартфоні та дистанційно керувати станом реле. Користувач через інтерфейс Blynk може отримувати актуальну інформацію про мікроклімат і оперативно перемикати режими роботи обладнання (вмикати або вимикати окремі пристрої за потреби), залишаючи за системою можливість автоматичного контролю. Таким чином, поєднання ThingSpeak (для довгострокового збереження даних і аналітики) та Blynk (для зручного мобільного контролю) забезпечує одночасно і хмарний моніторинг всіх параметрів, і інтерактивне керування мікрокліматом теплиці в реальному часі.

Проведене тестування підтвердило працездатність і ефективність створеного програмно-апаратного комплексу. Функціональні перевірки виконувалися у симуляційному середовищі Wokwi, що дозволило імітувати різні сценарії зміни кліматичних умов. Результати випробувань засвідчили коректність зчитування даних з усіх датчиків та належне виконання алгоритму керування: при перевищенні порогу температури система автоматично активувала охолодження, при недостатньому освітленні вмикалося підсвічування, а у разі зростання рівня CO_2 спрацьовувала вентиляція – всі реакції відповідали заданій логіці. Параметри оперативно відображалися на OLED-дисплеї (оновлення кожні 2 секунди) і паралельно передавалися на хмарну платформу ThingSpeak з інтервалом близько 20 с. Моніторинг через веб-інтерфейс ThingSpeak підтвердив успішну та безперервну передачу телеметрії: на графіках каналів коректно відображалися динаміка температури, вологості, освітленості, рівня CO_2, а також стан кожного реле. Одночасно мобільний додаток Blynk без затримок показував актуальні показники сенсорів і зміну стану виконавчих пристроїв, надаючи можливість дистанційного втручання в разі потреби – це продемонструвало ефективність інтеграції з IoT-сервісами. Система загалом проявила високу стабільність: протягом тривалої симуляції не спостерігалося збоїв або зависань, кожен виконавчий механізм своєчасно реагував на зміну умов. Більше того, після багаторазових перезапусків контролера та короткочасних втрат Wi-Fi-зв’язку система автоматично відновлювала роботу і продовжувала передавання даних на сервер, що підтверджує її надійність і готовність до реального використання.

Таким чином, у ході виконання магістерської роботи всі поставлені завдання були успішно вирішені, а мети дослідження досягнуто. Проведено ґрунтовний аналіз предметної області та існуючих інформаційних систем, на основі чого обґрунтовано вибір технічної платформи і компонентів, розроблено структурну схему та алгоритми функціонування системи. В результаті реалізовано повноцінний програмно-апаратний комплекс керування мікрокліматом теплиці з підтримкою сучасних хмарних технологій, після чого його працездатність і ефективність підтверджено тестовими випробуваннями. Розроблена система

відповідає заданим вимогам щодо точності контролю, оперативності реагування, стійкості та масштабованості. Вона демонструє готовність до практичного впровадження у невеликих тепличних господарствах, де її використання сприятиме підвищенню ефективності управління мікрокліматом, стабільності врожайності та зниженню витрат енергії і води.

Водночас визначено низку обмежень створеної системи, які окреслюють напрямки для подальшого вдосконалення. По-перше, відсутні автономне резервне живлення та енергонезалежна пам’ять: у разі перебоїв електропостачання робота контролера повністю зупиняється, а накопичені дані й встановлені налаштування не зберігаються. Це означає, що після тривалого вимкнення живлення система починає цикл контролю заново, не враховуючи попередню історію показників. По-друге, алгоритм регулювання наразі базується на фіксованих порогових значеннях, які потребують ручного налаштування під конкретні умови вирощування; система не оснащена механізмами самонавчання чи адаптивного коригування цих порогів у процесі роботи. Крім того, дослідна експлуатація проводилася в основному в умовах програмної симуляції – у реальній теплиці можуть проявитися додаткові фактори (точність калібрування датчиків, вплив зовнішнього середовища, знос обладнання тощо), що вимагатиме відповідних коригувань системи.

Подальший розвиток проєкту може бути спрямований на підвищення автономності, інтелектуальності та масштабності системи. Передусім доцільно додати до апаратної конфігурації резервне джерело живлення (акумулятор) та модуль зовнішньої пам’яті для збереження журналу даних і налаштувань – це дасть змогу системі безперервно функціонувати навіть за перебоїв електрики й відновлювати роботу без втрати інформації. Також варто впровадити розширені засоби діагностики стану вузлів, аби система могла автоматично виявляти і повідомляти про збої (наприклад, відмову датчика або розрив зв’язку) та переходити в безпечний режим роботи. Перспективним напрямом є ускладнення алгоритмів керування мікрокліматом: застосування адаптивних методів або елементів штучного інтелекту, які б дозволили динамічно змінювати режими і

порогові значення на основі аналізу накопичених даних та прогнозування умов. Система має потенціал до масштабування, тому можна розширити її функціонал інтегруванням додаткових сенсорів (наприклад, вологості ґрунту, рівня освітленості ультрафіолету чи концентрації поживних речовин) та нових виконавчих пристроїв (систем автоматичного зрошення, обігрівальних елементів тощо) для комплексного контролю всіх аспектів мікроклімату. Окрім цього, планується інтеграція розробленого рішення в більш комплексну IoT-інфраструктуру: перехід від вузькоспеціалізованих хмарних сервісів до власного серверного середовища з базою даних дозволить довгостроково зберігати великі обсяги телеметрії, а використання платформ на кшталт Node-RED та Grafana – створювати гнучкі кастомізовані дашборди для поглибленого аналізу. Реалізація зазначених кроків підвищить загальну стійкість, ефективність і універсальність системи, розширюючи межі її застосування та наближаючи впровадження технологій «розумної теплиці» в практику сучасного агровиробництва.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Клімат-контроль у теплицях: як створити ідеальні умови для рослин [Електронний ресурс] // Vegetable. – Режим доступу: https://vegetable.com.ua/klimat-kontrol-u-teplitsyax-yak-stvoriti-idealni-umovi-dlya-roslin/. – Назва з екрана. – Дата звернення: 23.02.2025.
2. Система клімат-контролю для теплиць [Електронний ресурс] // Agrotechnical. – Режим доступу: https://www.agrotechnical.net.ua/uk/climate-control-system/. – Назва з екрана. – Дата звернення: 23.02.2025.
3. Документація Wokwi [Електронний ресурс] // Wokwi Docs. – Режим доступу: https://docs.wokwi.com/. – Назва з екрана. – Дата звернення: 23.02.2025.
4. Документація ThingSpeak [Електронний ресурс] // MathWorks ThingSpeak. – Режим доступу: https://thingspeak.mathworks.com/. – Назва з екрана. – Дата звернення: 23.02.2025.
5. Технічна документація BH1750 [Електронний ресурс]. – Режим доступу: https://arduino.ua/docs/ADC154/BH1750FVI.pdf. – Назва з екрана. – Дата звернення: 23.02.2025.
6. Технічна документація датчика MQ-135 [Електронний ресурс] // Olimex. – Режим доступу: https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf. – Назва з екрана. – Дата звернення: 23.02.2025.
7. Технічна документація датчика DHT22 [Електронний ресурс] // SparkFun. – Режим доступу: https://cdn.sparkfun.com/assets/f/7/d/9/c/DHT22.pdf. – Назва з екрана. – Дата звернення: 23.02.2025.
8. Технічна документація релейного модуля 5V Single Channel Relay Module [Електронний ресурс] // Components101. – Режим доступу: https://components101.com/switches/5v-single-channel-relay-module-pinout-features-applications-working-datasheet. – Назва з екрана. – Дата звернення: 23.02.2025.

9. Платформа Blynk [Електронний ресурс]. – Режим доступу: https://blynk.io/. – Назва з екрана. – Дата звернення: 23.02.2025.
10. MongoDB [Електронний ресурс]. – Режим доступу: https://www.mongodb.com/. – Назва з екрана. – Дата звернення: 23.02.2025.
11. Node-RED [Електронний ресурс]. – Режим доступу: https://nodered.org/. – Назва з екрана. – Дата звернення: 23.02.2025.
12. Grafana [Електронний ресурс]. – Режим доступу: https://grafana.com/. – Назва з екрана. – Дата звернення: 23.02.2025.
13. Tripathy P. K., et al. MyGreen: An IoT-enabled smart greenhouse for sustainable agriculture // IEEE Consumer Electronics Magazine. – 2021. – Vol. 10, No. 4. – P. 57–62.
14. Rayhana R., Xiao G., Liu Z. Internet of things empowered smart greenhouse farming // IEEE Journal of Radio Frequency Identification. – 2020. – Vol. 4, No. 3. – P. 195–211.
15. Gorjian S., et al. A review on opportunities for implementation of solar energy technologies in agricultural greenhouses // Journal of Cleaner Production. – 2021. – Vol. 285. – Article ID 124807.
16. Забудський А. І., та ін. Цифрове сільське господарство: Розумна теплиця // Цифрове сільське господарство регіону: основні завдання, перспективні напрямки та системні ефекти. – 2019. – С. 174–178.
17. Gokhale P., Bhat O., Bhat S. Introduction to IOT // Int. Adv. Res. J. Sci. Eng. Technol. – 2018. – Vol. 5, No. 1. – P. 41–44.
18. Ullah I., et al. An optimization scheme for IoT based smart greenhouse climate control with efficient energy consumption // Computing. – 2022. – Vol. 104, No. 2. – P. 433–457.
19. Khan F. A., Ibrahim A. A., Zeki A. M. Environmental monitoring and disease detection of plants in smart greenhouse using internet of things // J. Phys. Commun. – 2020. – Vol. 4, No. 5. – Article ID 055008.

20. Aghaseyedabdollah M., et al. IoT Based Smart Greenhouse Design with an Intelligent Supervisory Fuzzy Optimized Controller // Proc. 7th Int. Conf. Web Research (ICWR), IEEE. – 2021. – P. 311–317.
21. Visvesvaran C., et al. Smart Greenhouse Monitoring System using Wireless Sensor Networks // Proc. 2nd Int. Conf. Smart Electronics and Communication (ICOSEC), IEEE. – 2021. – P. 96–101.
22. Kumar N., et al. Development of an IoT-Based Smart Greenhouse Using Arduino // In: Advances in Communication, Devices and Networking. – Singapore: Springer, 2022. – P. 455–465.
23. Mohan N. A. P., et al. Smart Greenhouse By Using Arduino Mega and IoT // Solid State Technology. – 2020. – Vol. 63, No. 6. – P. 17155–17160.
24. Andrianto H., et al. Development of smart greenhouse system for hydroponic agriculture // Proc. Int. Conf. Inf. Technol. Syst. Innov. (ICITSI), IEEE. – 2020. – P. 335–340.
25. Ibrahim H., et al. A layered IoT architecture for greenhouse monitoring and remote control // SN Appl. Sci. – 2019. – Vol. 1, No. 3. – P. 1–12.
26. Faris D. M., Mahmood M. B. Data acquisition of greenhouse using arduino // J. Babylon Univ. Appl. Sci. – 2014. – Vol. 22, No. 7. – P. 1908–1916.
27. Kodali R. K., et al. IoT based smart greenhouse // IEEE Reg. 10 Humanit. Technol. Conf. – 2016. – DOI: 10.1109/R10-HTC.2016.7906846.
28. Hassanien R. H. E., Li M., Dong Lin W. Advanced applications of solar energy in agricultural greenhouses // Renew. Sustain. Energy Rev. – 2016. – Vol. 54. – P. 989–1001.
29. Queiroz Zorzeto H. F. A., et al. Gradients of Temperature and Relative Humidity of Air in Greenhouse with Wireless Sensor Network // Proc. 2nd Int. Conf. Agric. Biotechnol. – 2014. – Vol. 79, No. 8. – P. 41–46.
30. Modani V., et al. IoT Based Greenhouse Monitoring System: Technical Review // Int. Res. J. Eng. Technol. – 2017. – Vol. 10. – P. 2395–56.
31. Omid M., Shafaei A. Temperature and relative humidity changes inside greenhouse // Int. Agrophysics. – 2005. – Vol. 19, No. 2. – P. 153–158.

32. Madushanki A. A. R., et al. Adoption of the Internet of Things (IoT) in agriculture and smart farming towards urban greening: A review // Int. J. Adv. Comput. Sci. Appl. – 2019. – Vol. 10, No. 4. – P. 11–28.
33. Ismail N., et al. Smart irrigation system based on internet of things (IoT) // J. Phys. Conf. Ser. – 2019. – Vol. 1339, No. 1. – DOI: 10.1088/1742-6596/1339/1/012012.
34. Dahlan N. Y., et al. Temperature based control of ventilation system for optimum climate in tomato greenhouse // Indones. J. Electr. Eng. Comput. Sci. – 2018. – Vol. 12, No. 2. – P. 655–661.
35. Li N., et al. Smart Agriculture with an Automated IoT-Based Greenhouse System for Local Communities // Adv. Internet Things. – 2019. – Vol. 9, No. 2. – P. 15–31.
36. V., S. W. S. Solar photovoltaic water pumping system for irrigation: A review // Afr. J. Agric. Res. – 2015. – Vol. 10, No. 22. – P. 2267–2273.
37. Sri R. I., et al. Arduino Based Smart Irrigation System Using IOT. – 2017, December.
38. Agnal S., et al. Automated Smart Greenhouse Environment Using IoT. – 2018. – P. 1665–1671.
39. Андрющенко Д. Д. Мультиагентні системи для обслуговування тепличного середовища // Традиції та нові наукові стратегії у Центральній та Східній Європі: Матеріали V міжнар. наук.-практ. конф. – Київ, 2022. – С. 122–124.
40. Sebastian P. Internet of Things with Arduino Cookbook. – Birmingham: Packt Publishing, 2018.
41. Sundaravadivelu, A., Dhanalakshmi, R. Arduino based automatic climate control system for greenhouse [Електронний ресурс] // *Journal of Physics: Conference Series*, 2021. – Т. 1916. – № 1. – С. 012137. – Режим доступу: https://iopscience.iop.org/article/10.1088/1742-6596/1916/1/012137/pdf, вільний. – Дата звернення: 25.05.2025.
42. Zorita-Lamadrid, A., Sáez, D., Márquez, F., Núñez, A. Greenhouse climate control system based on Arduino and IoT platform [Електронний ресурс] // *Sensors*,

2020. – Т. 20. – № 20. – С. 5860. – Режим доступу: https://www.mdpi.com/1424-8220/20/20/5860, вільний. – Дата звернення: 25.05.2025.

43. Singh, A., Pal, R. IoT-Based Smart Greenhouse System for Real-Time Monitoring and Control Using Arduino and Blynk App [Електронний ресурс] // *International Journal of Advanced Research in Science, Communication and Technology*, 2022. – Т. 12. – С. 406–412. – Режим доступу: https://ijarsct.co.in/Paper7065.pdf, вільний. – Дата звернення: 25.05.2025.
44. Zhang, Y., Zhang, X., Zhang, H. Design of smart greenhouse monitoring system based on Arduino and wireless sensor networks [Електронний ресурс] // *IOP Conference Series: Materials Science and Engineering*, 2020. – Т. 768. – № 5. – С. 052007. – Режим доступу: https://iopscience.iop.org/article/10.1088/1757-899X/768/5/052007/pdf, вільний. – Дата звернення: 25.05.2025.
45. Shaik, K., Mohamed, H. An IoT-based Smart Greenhouse Monitoring and Controlling System Using Arduino [Електронний ресурс] // *International Journal of Engineering Research & Technology (IJERT)*, 2021. – Т. 10. – № 9. – С. 73–76. – Режим доступу: https://www.ijert.org/research/an-iot-based-smart-greenhouse-monitoring-and-controlling-system-using-arduino-IJERTV10IS090026.pdf, вільний. – Дата звернення: 25.05.2025.
46. Yang, X., Wang, S., Huang, G. An IoT-based intelligent greenhouse for environmental monitoring and automatic control [Електронний ресурс] // *International Journal of Agricultural and Biological Engineering*, 2019. – Т. 12. – № 2. – С. 75–83. – Режим доступу: https://www.ijabe.org/index.php/ijabe/article/view/4657/pdf, вільний. – Дата звернення: 25.05.2025.
47. Mishra, A., Tripathi, R. Design and implementation of smart greenhouse automation using IoT [Електронний ресурс] // *International Journal of Scientific Research and Engineering Development*, 2021. – Т. 4. – № 3. – С. 498–502. – Режим доступу: https://ijsred.com/volume4/issue3/IJSRED-V4I3P69.pdf, вільний. – Дата звернення: 25.05.2025.

48. Devika, C. S., Deepa, S. Smart irrigation and climate control system for greenhouse using Arduino and IoT [Електронний ресурс] // *International Journal of Engineering and Techniques*, 2019. – Т. 5. – № 6. – С. 51–57. – Режим доступу: https://www.ijetjournal.org/Volume5/Issue6/IJET-V5I6P9.pdf, вільний. – Дата звернення: 25.05.2025.
49. **Ardiansah I., Bafdal N., Suryadi E., Bono A.** Greenhouse Monitoring and Automation Using Arduino: a Review on Precision Farming and Internet of Things (IoT) [Електронний ресурс] // *International Journal on Advanced Science, Engineering and Information Technology*. — 2020. — Vol. 10, No. 2. — P. 703–709. — Режим доступу: **https://doi.org/10.18517/ijaseit.10.2.10249**, вільний. — Дата звернення: 17.05.2025.
50. **Akpulonu V.U., Agbese E.A., Obizue C.E., Nater A., Abdulsalam N., Ogochukwu I.S., Aminu-Baba M., Ene A.H.** Design and construction of Arduino based greenhouse monitoring system using IoT [Електронний ресурс] // *World Journal of Advanced Engineering Technology and Sciences*. — 2024. — Vol. 12, No. 2. — P. 189–198. — Режим доступу: **https://doi.org/10.30574/wjaets.2024.12.2.0280**, вільний. — Дата звернення: 17.05.2025.
51. **Harale V.R., Dharwadkar N.V.** An automated climate control system for greenhouse using deep learning for tomato crop [Електронний ресурс] // *JournalNX – A Multidisciplinary Peer Reviewed Journal*. — 2021. — P. 94–99. — Режим доступу: **https://repo.journalnx.com/index.php/nx/article/view/2319**, вільний. — Дата звернення: 17.05.2025.
52. **Galon M.L.Q., Tumaliwan M.V.R., Sejera M.M.** Automated Monitoring and Control System of Solar Greenhouse Using ESP32 and Blynk Application [Електронний ресурс] // *Engineering Proceedings*. — 2025. — Vol. 92, No. 1. — P. 57. — Режим доступу: **https://doi.org/10.3390/engproc2025092057**, вільний. — Дата звернення: 17.05.2025.

ДОДАТКИ

Додаток А

Програмний код:

```
#define BLYNK_TEMPLATE_ID "TMPL4p-Dxhapj"
#define BLYNK_TEMPLATE_NAME "Greenhouse model project"
#define BLYNK_AUTH_TOKEN "6NbdwMqRa0kCb3PfC3FIUdTAoS4td9hU"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include "ThingSpeak.h"
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <BH1750.h>

// Wi-Fi
char ssid[] = "Wokwi-GUEST";
char pass[] = "";
WiFiClient client;

// ThingSpeak
unsigned long channelID = 2964454;
const char* writeAPIKey = "YUMNLXJBA8NLZFFE";

// OLED
#define SCREEN_WIDTH 128
```

```
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// DHT22
#define DHTPIN 15
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

// MQ135
#define MQ135PIN 36

// BH1750
BH1750 lightMeter;

// Реле
#define RELAY_TEMP 23
#define RELAY_CO2  19
#define RELAY_LIGHT 18

// Змінні
float temperature = 0.0, humidity = 0.0, lux = 0.0;
int airQuality = 0;
bool relayTemp = false, relayCO2 = false, relayLight = false;

// Таймери
unsigned long lastSendTime = 0;
const unsigned long sendInterval = 20000;
const unsigned long manualDuration = 600000; // 10 хв = 600000 мс

// Режими
```

```
bool manualTemp = false, manualCO2 = false, manualLight = false;
unsigned long tempManualStart = 0, co2ManualStart = 0, lightManualStart = 0;

// Blynk керування
BLYNK_WRITE(V4) {
  manualTemp = param.asInt();
  tempManualStart = millis();
  digitalWrite(RELAY_TEMP, manualTemp);
}

BLYNK_WRITE(V5) {
  manualCO2 = param.asInt();
  co2ManualStart = millis();
  digitalWrite(RELAY_CO2, manualCO2);
}

BLYNK_WRITE(V6) {
  manualLight = param.asInt();
  lightManualStart = millis();
  digitalWrite(RELAY_LIGHT, manualLight);
}

void connectWiFi() {
  Serial.print("Підключення до Wi-Fi...");
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

```
  Serial.println("\n✅ Підключено");
}

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

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  ThingSpeak.begin(client);

  dht.begin();
  lightMeter.begin(BH1750::CONTINUOUS_HIGH_RES_MODE);

  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("❌ OLED не знайдено!");
    while (true);
  }
  display.clearDisplay();
  display.setTextColor(WHITE);

  pinMode(RELAY_TEMP, OUTPUT);
  pinMode(RELAY_CO2, OUTPUT);
  pinMode(RELAY_LIGHT, OUTPUT);
  digitalWrite(RELAY_TEMP, LOW);
  digitalWrite(RELAY_CO2, LOW);
  digitalWrite(RELAY_LIGHT, LOW);
}

void loop() {
```

```
Blynk.run();

static unsigned long lastRead = 0;
if (millis() - lastRead >= 2000) {
  lastRead = millis();

  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
  lux = lightMeter.readLightLevel();
  airQuality = analogRead(MQ135PIN);

  if (isnan(temperature) || isnan(humidity)) {
    Serial.println("🌡 Помилка зчитування DHT22");
    return;
  }

  unsigned long now = millis();

  // TEMP
  if (manualTemp && now - tempManualStart >= manualDuration) {
    manualTemp = false;
  }
  if (!manualTemp) {
    relayTemp = (temperature > 30.0);
    digitalWrite(RELAY_TEMP, relayTemp);
  }
  // CO2
  if (manualCO2 && now - co2ManualStart >= manualDuration) {
    manualCO2 = false;
```

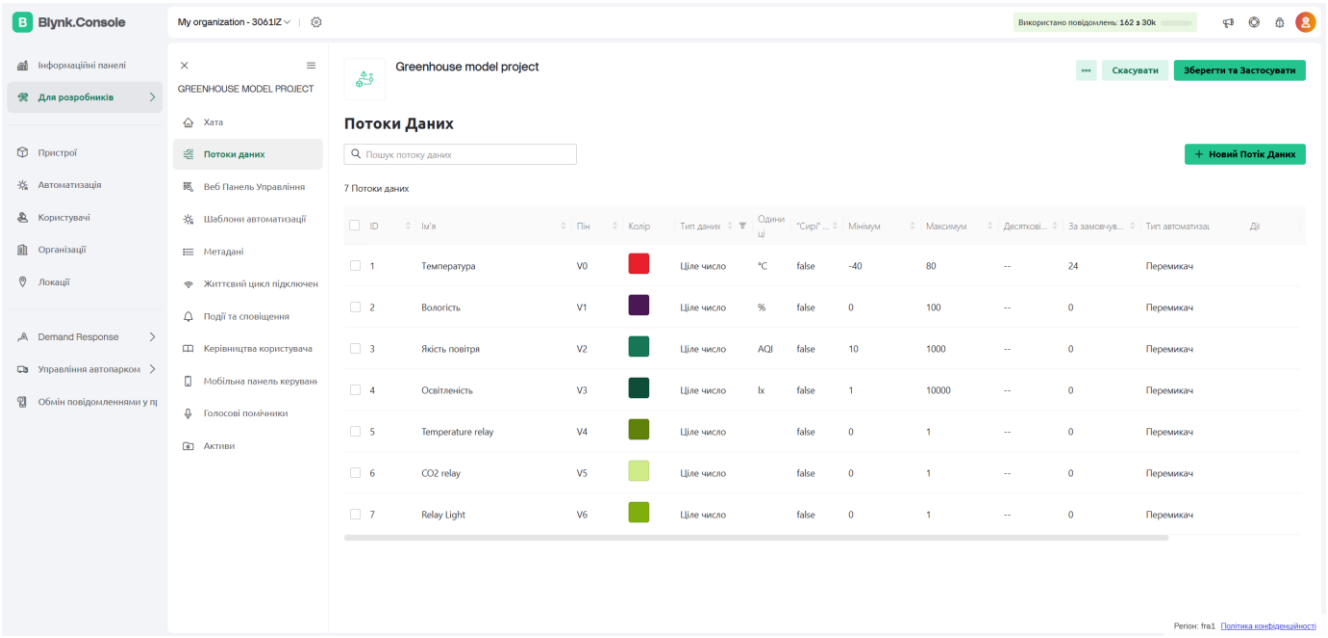
```
  }
  if (!manualCO2) {
    relayCO2 = (airQuality > 400);
    digitalWrite(RELAY_CO2, relayCO2);
  }
  // LIGHT
  if (manualLight && now - lightManualStart >= manualDuration) {
    manualLight = false;
  }
  if (!manualLight) {
    relayLight = (lux < 200);
    digitalWrite(RELAY_LIGHT, relayLight);
  }
  // OLED
  display.clearDisplay();
  display.setTextSize(1);
  display.setCursor(0, 0);
  display.printf("T:%.1fC H:%.1f%%\n", temperature, humidity);
  display.printf("Lux:%.0f\n", lux);
  display.printf("CO2:%d\n", airQuality);
  display.printf("R1:%s R2:%s R3:%s\n",
    digitalRead(RELAY_TEMP) ? "ON" : "OFF",
    digitalRead(RELAY_CO2) ? "ON" : "OFF",
    digitalRead(RELAY_LIGHT) ? "ON" : "OFF");
  display.display();
  // Serial
  Serial.printf("T:%.1fC H:%.1f%% Lux:%.0f CO2:%d\n",
    temperature, humidity, lux, airQuality);
}
```

```
  // Передача в Blynk
  Blynk.virtualWrite(V0, temperature);
  Blynk.virtualWrite(V1, humidity);
  Blynk.virtualWrite(V2, airQuality);
  Blynk.virtualWrite(V3, lux);
  Blynk.virtualWrite(V4, manualTemp ? 1 : relayTemp);
  Blynk.virtualWrite(V5, manualCO2 ? 1 : relayCO2);
  Blynk.virtualWrite(V6, manualLight ? 1 : relayLight);

  // Передача в ThingSpeak
  if (millis() - lastSendTime >= sendInterval) {
    lastSendTime = millis();
    ThingSpeak.setField(1, temperature);
    ThingSpeak.setField(2, humidity);
    ThingSpeak.setField(3, airQuality);
    ThingSpeak.setField(4, lux);
    ThingSpeak.setField(5, (int)(manualTemp ? 1 : relayTemp));
    ThingSpeak.setField(6, (int)(manualCO2 ? 1 : relayCO2));
    ThingSpeak.setField(7, (int)(manualLight ? 1 : relayLight));
    int response = ThingSpeak.writeFields(channelID, writeAPIKey);
    if (response == 200) {
      Serial.println("✅ Дані надіслано в ThingSpeak!");
    } else {
      Serial.print("❌ Помилка надсилання: ");
      Serial.println(response);
    }
  }
}
```

Додаток Б

Панель налаштування потоків даних Blynk:



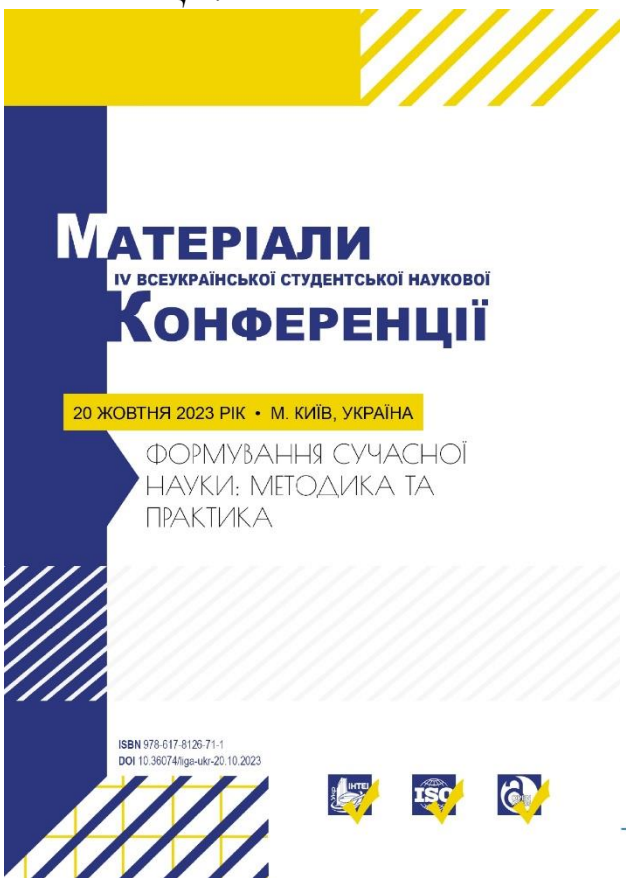
Додаток В

Технічні характеристики використаних компонентів:

Компонент	**Позначення**	**Основні технічні характеристики**
Мікроконтролер ESP32	ESP32 DevKit v1	Двоядерний 32-бітний процесор, частота до 240 МГц, 520 КБ SRAM, вбудований Wi-Fi та Bluetooth
Датчик температури і вологості	DHT22	Діапазон температур: -40…80 °C, точність ±0.5 °C; діапазон вологості: 0–100 %, точність ±2 %
Датчик освітленості	BH1750	Діапазон вимірювання: 1–65535 lx, інтерфейс I²C, напруга живлення: 3–5 В
Датчик якості повітря	MQ-135	Виявлення CO2, аміаку, спиртів, бензолу; аналоговий вихід, напруга живлення: 5 В
OLED-дисплей	SSD1306 128x64	Інтерфейс I²C, роздільна здатність 128×64, напруга живлення: 3.3–5 В
Релейний модуль	Relay Module 5V	Керування навантаженням до 10 А, напруга реле: 5 В, рівень сигналу керування: 3.3–5 В

Додаток Г

Тези, опубліковані в збірнику IV ВСЕУКРАЇНСЬКОЇ СТУДЕНТСЬКОЇ НАУКОВОЇ КОНФЕРЕНЦІЇ:



Датко Андрій Петрович, здобувач вищої освіти Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Київський національний економічний університет ім. Вадима Гетьмана, Україна

Науковий керівник: Устенко Станіслав Веніамінович, д-р. екон. наук, професор, професор кафедри інформаційних систем в економіці
Київський національний економічний університет ім. Вадима Гетьмана, Україна

АНАЛІЗ ТА РОЗВИТОК ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ ІНВЕСТУВАННЯ

Інформаційні системи для інвестування стають все більш актуальними в умовах швидкого розвитку фінансових технологій (FinTech) та зростання інтересу громадськості до інвестування та нагромадження активів. Завдяки цим, інвестори можуть зручно керувати своїми активами та ризиками, а простота та зручність встановлюють низький поріг входу навіть для осіб з обмеженими знаннями в даній сфері.

Ринок інформаційних систем для інвестування представлений різноманітними застосунками, серед яких є і вітчизняні. Зокрема, з найпопулярніших можна виділити наступні:

1. Mono Invest [8] – це інвестиційна платформа, яка пропонує клієнтам можливість інвестувати свої гроші у різні фінансові інструменти, включаючи депозити, облігації та інші інвестиційні продукти.
2. Sense SuperApp [6] – єдиний цифровий банк для особистих фінансів та рахунків ФОПа. В кінці 2021-го року отримав можливість купувати акції зарубіжних компаній, дозволених НКЦПФР [3].
3. Robinhood [7] – це фінансовий застосунок, який був розроблений для торгівлі акціями, криптовалютами та іншими фінансовими інструментами.
4. E*TRADE [12] – це відомий фінансовий брокер, який надає доступ до акцій, опціонів, ETF і фондів. Вони також мають свій мобільний додаток для торгівлі та інвестування.
5. TD Ameritrade [5] пропонує широкий вибір фінансових продуктів і має безкоштовну торгівлю акціями, опціонами і ETF. Вони також надають інструменти для дослідження ринку та аналізу акцій.
6. Webull [4] – це інвестиційна платформа, яка спеціалізується на акціях і криптовалютах. Вони пропонують безкоштовну торгівлю акціями та ETF і мають розширені аналітичні інструменти.
7. M1 Finance [9] – це платформа для інвестицій з автоматичними інвестиціями та можливістю створення індивідуальних портфелів. Вони пропонують безкоштовну торгівлю акціями і ETF, а також функцію "кредитування маржі".
8. Fidelity [11] – це добре відомий фінансовий брокер, який надає доступ до широкого спектру фінансових інструментів, включаючи акції, облігації, фонди і пенсійні рахунки.

Можна виділити наступні переваги даних систем:

1. Доступність. Дані системи надають можливість інвесторам здійснювати торгівлю акціями, опціонами та іншими інвестиційними продуктами, що робить

інвестування більш доступним для широкої аудиторії.

2. Мобільність. Більшість із зазначених систем мають мобільні додатки, які дозволяють інвесторам відстежувати та керувати своїми інвестиціями в режимі реального часу з будь-якого місця.

3. Фінансова грамотність. Дані системи часто надають користувачам доступ до освітніх ресурсів, аналітичних матеріалів та інструментів для дослідження ринків, що допомагає інвесторам приймати більш обдумані рішення.

4. Різноманітні опції інвестування. Запропоновані системи дозволяють інвестувати в різні акції, облігації, ETF, криптовалюти тощо, розширюючи вибір для інвесторів.

До недоліків даних систем можна віднести:

1. Ризики. Інвестування завжди пов'язане з ризиками, і навіть доступні платформи не гарантують прибуток. Інвестори можуть втратити гроші, особливо при неправильних рішеннях або відсутності адекватного аналізу.

2. Комісії. Хоча деякі інформаційні системи, як Robinhood, пропонують безкоштовну торгівлю, інші можуть мати комісії за різні операції, що може впливати на вибір інвестора.

3. Обмеження. Кожна з цих систем може мати свої обмеження, такі як обмеження на доступні інвестиційні продукти або обмеження на різні види рахунків.

4. Регуляторні обмеження. Регуляторні проблеми, які можуть виникнути, можуть вплинути на доступність послуг та репутацію платформи.

Системи даного класу беззаперечно граються важливу роль в економічній діяльності. Однак, також не можна заперечувати, що вони мають ряд недоліків, які можна і, навіть, необхідно усунути. Зокрема до них можна віднести ризики безпеки, брак механізмів захисту та можливі ризики для інвесторів. Також до цих відноситься обмеження на активи, які доступні до інвестицій, зокрема відсутність можливості придбати криптовалюти. Деякі із систем мають обмежений функціонал з браком поглиблених інструментів інвестування, відсутності десктопної або мобільної версії. Також спільною проблемою для багатьох з них є складний перевантажений інтерфейс, що створює складнощі навіть досвідченому інвестору. Дані проблеми не є критичними і їх можна відносно легко усунути за вкладання необхідних ресурсів. Але в українських, та й не тільки, реаліях головним обмежуючим чинником є законодавство та регулюючі органи. В Україні з моменту повномасштабного вторгнення російської федерації в Україну НБУ [2] введені обмеження на виведення капіталу з України, що включає в себе заборону на вивід валюти та заборону купівлі акцій зарубіжних компаній. Також, як показує ситуація із застосунком "Mono invest" [1], існують значні ризики для інвесторів. Через те, що брокер-партнер потрапив під санкції РНБО [10], компанія мала продавати активи інвесторів, часто навіть їм у збиток, або ж заморозити їх на невизначений час, що фактично означає втрату доступу до них та збитки. Враховуючи вище сказане, пропонується створити ІС, що матиме як мобільну, так і десктопну версію; достатньо широкий функціонал, щоб нею могли користуватись інвестори просунутого рівню; простий та зрозумілий інтерфейс, що встановить низький поріг входу для інвесторів-початківців або простих користувачів, що хочуть власти надлишкові кошти. Впровадити інтеграцію з криптовалютними біржами, або створити власний криптогаманець, що розширить опції інвестування. Держава має прийняти необхідні законодавчі зміни, щоб врегулювати дану сферу та захистити

інтереси інвесторів. В сумі всі ці заходи сприятимуть зростанню благополуччя населення та матимуть позитивний вплив на економіку.

Висновки: Проведено аналіз сучасних інформаційних систем на ринку фінансів для інвестування. Запропоновано створити інформаційну систему, що матиме як мобільну, так і десктопну версії, широкий функціонал для розв'язання задач інвестування, а також провести структурно-технологічні вдосконалення та відповідне організаційне забезпечення по функціонуванню системи в сучасних умовах.

Список використаних джерел:

1. Українська правда [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://www.epravda.com.ua/news/2023/07/5/701939/ (дата звернення 30.09.2023).
2. Національний банк України [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://bank.gov.ua/ (дата звернення 30.09.2023).
3. Національна комісія з цінних паперів та фондового ринку [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://www.nssmc.gov.ua/ (дата звернення 30.09.2023).
4. Webull [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://www.webull.com/ (дата звернення 30.09.2023).
5. TD Ameritrade [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://www.tdameritrade.com/ (дата звернення 30.09.2023).
6. Sense SuperApp [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://sensebank.com.ua/investiicii-u-sense-superapp (дата звернення 30.09.2023).
7. Robinhood [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://robinhood.com/us/en/ (дата звернення 30.09.2023).
8. Mono Invest [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://www.monobank.ua/invest?lang=uk (дата звернення 30.09.2023).
9. M1 Finance [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://m1.com/ (дата звернення 30.09.2023).
10. LIGA.NET [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://finance.liga.net/ua/stock/article/mono-invest-zakrylsya-iz-za-podsanktsionnogo-brokera-pochemu-tak-proizoshlo (дата звернення 30.09.2023).
11. Fidelity [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://www.fidelity.com/ (дата звернення 30.09.2023).
12. E*TRADE [Електронний ресурс]. — 2023. — Режим доступу до ресурсу: https://us.etrade.com/home (дата звернення 30.09.2023).

Додаток Д

Тези, опубліковані в збірнику V Міжнародної науково-практичної конференції молодих вчених, аспірантів і студентів (Інтернет-конференція):

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Київський національний економічний університет імені Вадима Гетьмана
Ярославська Державна Вища Технічно-Економічна Школа
ім. о. Броніслава Маркевича (Польща)
Академія фінансів та бізнесу Vistula (Польща)
Європейський університет у Варшаві (Польща)
Компанія Pro Insight sp. z o.o. (Польща)
Вроцлавський університет економіки та бізнесу (Польща)
Інститут кібернетики імені В.М. Глушкова НАН України
Київський національний університет імені Тараса Шевченка
Національний технічний університет України «КПІ Ім. Ігоря Сікорського»
Харківський національний економічний університет ім.Семена Кузнеця
Національний технічний університет «ХПІ»
Харківський національний університет імені В. Н. Каразіна
Харківський національний університет радіоелектроніки
Харківський національний університет міського господарства ім. О.М. Бекетова
Херсонський національний технічний університет
Запорізький національний університет
Хмельницький національний університет
Криворізький державний педагогічний університет
Східноукраїнський національний університет ім. Володимира Даля
Львівський національний університет імені Івана Франка
Львівський національний технічний університет «Львівська політехніка»
Кременчуцький національний університет ім. М. Остроградського

СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ В УПРАВЛІННІ

Збірник матеріалів

V Міжнародної науково-практичної конференції молодих вчених, аспірантів і студентів (Інтернет-конференція)

18–19 квітня 2024 р.

ефективною у даному питанні. Перевагами дронів є їхня гнучкість для різноманітних цілей та умов використання, а також зручність у використанні. Саме тому інтеграція безпілотних автономних засобів з ройовим інтелектом дає можливість забезпечити негайні та всебічні спостереження за коливаннями забрудненння повітря [3], а також динамічну адаптацію до навколишнього середовища, завдань та змін у складі рою [4].

Система, що керує дронами за допомогою штучного інтелекту, є оптимальним рішенням для моніторингу якості повітря та визначення джерел забруднення, оскільки забезпечує гнучкість та надійність моніторингу за різних умов.

Список використаних джерел

1. Вміст забруднювальних речовин в атмосферному повітрі міст України за даними спостережень у грудні 2023 року [Мережевий ресурс]. – Режим доступу: https://www.meteo.gov.ua/f/zabrudnennya_po_mistah_za_gruden_2023.pdf
2. Гударенко В.М. Модель точкових забруднень для оцінки якості повітряного басейну / В. М. Гударенко, А. О. Подорожняк, Е. Л. Шавирін, Ю. П. Шамаєв // Стандартизація. Сертифікація. Якість. – 2016. – № 5. – С. 32-39. – Режим доступу: http://nbuv.gov.ua/UJRN/ssja_2016_5_8.
3. Baumgart J. Taking Flight for a Greener Planet: How Swarming Could Help Monitor Air Pollution Sources / J. Baumgart, D. Mikolajewski, J.M. Czerniak // Electronics – 2024. – Vol. 13. – Iss. 3. – Article no. 577. – Режим доступу: https://doi.org/10.3390/electronics13030577
4. Semwal A. The Use of Artificial Intelligence in Swarm Drones / A. Semwal, S. Shikalgar, R. Solanki // IJRASET – 2023. – Vol. 11. – Iss. 7. – P. 1052-1057. – Режим доступу: https://doi.org/10.22214/ijraset.2023.54799.

Науковий керівник: Подорожняк А.О., к.т.н., доцент

Датко А. П.
студент
Київський національний економічний університет імені Вадима Гетьмана
a.p.datko@gmail.com

ВИКОРИСТАННЯ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ ТА МАШИННОГО ПАВЧАННЯ У ПРОЦЕСАХ РОЗРОБКИ ПРОГРАМНОГО ЗЕБЕЗПЕЧЕННЯ

Розробка програмного забезпечення є складним процесом, який вимагає великої кількості ресурсів та ефективного управління. У зв'язку зі зростанням обсягів даних та складності завдань, пов'язаних з розробкою програмного забезпечення, системи штучного інтелекту та машинного навчання стають невід'ємною частиною цього процесу [1].

Інструменти, які надають підтримку в написанні коду на основі штучного інтелекту, використовуючи модель машинного навчання GPT (Generative Pre-trained Transformer) [2], значно спрощують процес розробки, автоматизуючи його [3], та прибирають рутинні одноманітні операції, дозволяючи зосередитись на вирішенні основного завдання.

Дана сфера є новою і на даний момент переживає бурхливий розвиток, але вже на даний момент можна виділити основних лідерів:

1. «GitHub Copilot» [4] – інструмент, розроблений GitHub спільно з OpenAI, який використовує штучний інтелект для автоматичного генерування коду та коментарів

на основі контексту у реальному часі. Він може пропонувати варіанти коду на основі вже наявного коду або опису проблеми.
2. «TabNine» [5] – це інтелектуальний автодоповнювач коду, який використовує модель машинного навчання для пропозицій контекстно-залежних варіантів коду в багатьох мовах програмування. Він може пропонувати не тільки стандартні автодоповнення, але й додаткові варіанти, засновані на аналізі шаблонів коду.
3. «Blackbox» [6] – це інструмент для безпечного та конфіденційного зберігання та передачі інформації, що забезпечує кінцеву анонімність для користувачів. Він дозволяє шифрувати дані, щоб навіть сервіс, який їх обробляє, не міг прочитати їх.
4. «AiXcoder» [7] – це інтелектуальний помічник для програмістів, який використовує машинне навчання для пропозицій контекстно-залежних фрагментів коду та поліпшення процесу написання програм.
5. «Codiga» [8] – це інструмент статичного аналізу коду з штучним інтелектом, який можна використовувати в будь-якому середовищі розробки, включаючи VS Code, JetBrains, VisualStudio, GitHub, GitLab та Bitbucket. Він забезпечує налаштовуваний статичний аналіз коду з безпечним аналізом коду, автоматизованим рецензуванням коду та фрагментами коду.
6. «IntelliCode в Microsoft Visual Studio» [9] – це набір інструментів, включаючи функцію IntelliSense, який використовує штучний інтелект для пропозицій контекстно-залежних підказок, автодоповнень коду та рекомендацій для поліпшення процесу написання коду в середовищі розробки Visual Studio від Microsoft.

Після проведення аналіз функціональності даних систем можна сказати, що системи даного класу мають як переваги, так і недоліки, що мають вплив на їхнє використання. До переваг можна віднести:

1. *Збільшення продуктивності.* Інструменти, як GitHub Copilot, TabNine та IntelliCode, можуть значно збільшити продуктивність програмістів, швидко надаючи їм готові фрагменти коду або рекомендації на основі контексту.
2. *Зменшення кількості помилок.* Використання інтелектуальних автодоповнювачів може допомогти уникнути деяких типових помилок, оскільки вони можуть пропонувати правильні варіанти коду або вказати на потенційні проблеми.
3. *Швидкий пошук інформації.* Інструменти, як Codiga, можуть швидко знаходити необхідні фрагменти коду або документацію, що може значно зекономити час програмістів на пошук та розуміння необхідної інформації.
4. *Покращення якості коду.* Деякі системи можуть надавати рекомендації щодо покращення якості коду, такі як оптимізація, чистота або стиль, що може допомогти забезпечити більш якісну розробку програмного забезпечення.

Серед недоліків можна виділити:

1. *Ризик залежності.* Якщо програмісти стають занадто залежними від інтелектуальних автодоповнювачів, це може зменшити їхні навички вирішення проблем та розуміння фундаментальних концепцій програмування.
2. *Недосконалість алгоритмів.* Іноді системи можуть запропонувати неправильні варіанти коду або надавати невірні рекомендації, що може призвести до появи помилок або недосягнення очікуваних результатів.
3. *Приватність і безпека.* Використання інтелектуальних автодоповнювачів може потенційно ставити під загрозу конфіденційність і безпеку даних, особливо якщо вони використовують хмарні сервіси для обробки інформації.
4. *Неузгодженість стилю коду.* Іноді рекомендації систем можуть не відповідати внутрішньому стилю коду команди або проекту, що може призвести до непослідовності та збільшення часу на перевірку та виправлення коду.

Отже, можна зробити висновок, що використання систем штучного інтелекту та машинного навчання у процесах розробки програмного забезпечення дійсно відкриває безліч можливостей для покращення ефективності та якості розробки. Одним із ключових напрямків є автоматизація тестування. Штучний інтелект може допомогти виявляти та усувати помилки швидше та ефективніше, що веде до зменшення часу, необхідного для випуску продукту на ринок, та зниження ймовірності виявлення помилок користувачами. Також, аналіз коду з використанням інструментів штучного інтелекту може допомогти розробникам виявляти потенційні проблеми в коді, такі як уразливості безпеки чи проблеми оптимізації, що сприяє покращенню якості та надійності програмного продукту. Подальшим розвитком цієї області є персоналізація програмних продуктів. Штучний інтелект може аналізувати поведінку користувачів та надавати персоналізовані рекомендації чи функціонал, що відповідає індивідуальним потребам кожного користувача. Дослідження в галузі залишається активним та перспективним напрямком розвитку. Із зростанням обсягів даних та постійним вдосконаленням алгоритмів машинного навчання, можна очікувати подальші інновації в цій сфері, які сприятимуть ще більшій автоматизації та підвищенню якості розробки програмного забезпечення. Але також варто зазначити, що повноцінно замінити програмістів, та й взагалі спеціалістів IT-галузі дані системи не зможуть. Оскільки даний вид професій має в собі певну творчу складову. Для виконання задач машинною їй потрібно ставити чітке завдання, що надзвичайно складно. Зрештою, дані системи теж створені руками людини. Вони спрощують розробку та прибирають рутину. Збільшують вимоги знань до Junior спеціалістів, але аж ніяк не замінюють розробника.

Список використаних джерел

1. Автоматизація [Електронний ресурс]. — 2024. — Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Автоматизація (дата звернення 12.04.2024).
2. Розробка програмного забезпечення [Електронний ресурс]. — 2024. — Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Розробка_програмного_забезпечення (дата звернення 12.04.2024).
3. Generative pre-trained transformer [Електронний ресурс]. — 2024. — Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Generative_pre-trained_transformer (дата звернення 12.04.2024).
4. Веб-портал «GitHub Copilot» [Електронний ресурс]. — 2024. — Режим доступу до ресурсу: https://github.com/features/copilot (дата звернення 12.04.2024)
5. Веб-портал «TabNine» [Електронний ресурс]. — 2024. — Режим доступу до ресурсу: https://www.tabnine.com/ (дата звернення 12.04.2024)
6. Веб-портал «Blackbox» [Електронний ресурс]. — 2024. — Режим доступу до ресурсу: https://www.blackbox.ai/ (дата звернення 12.04.2024)
7. Веб-портал «AiXcoder» [Електронний ресурс]. — 2024. — Режим доступу до ресурсу: https://www.aixcoder.com/en/#/ (дата звернення 12.04.2024)
8. Веб-портал «Codiga» [Електронний ресурс]. — 2024. — Режим доступу до ресурсу: https://www.codiga.io/ (дата звернення 12.04.2024)
9. Веб-портал «IntelliCode в Microsoft Visual Studio» [Електронний ресурс]. — 2024. — Режим доступу до ресурсу: https://visualstudio.microsoft.com/services/intellicode/ (дата звернення 12.04.2024)

Науковий керівник: Денісова О.О., к.с.н., доцент.