

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Волков Олексій Олексійович

(прізвище, ім'я, по-батькові здобувача освіти)

УДК 004.46:631.23

КВАЛІФІКАЦІЙНА РОБОТА

Розробка системи управління «Розумною теплицею»

(тема роботи)

122 «Комп'ютерні науки»

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Волков Олексій Олексійович

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи

Євсєєв Сергій Петрович

(прізвище, ім'я, по батькові)

доктор технічних наук, професор

(науковий ступінь, вчене звання)

Житомир – 2025

Висновок кафедри _____
за результатами попереднього захисту: _____

Протокол засідання кафедри _____
№ _____ від «_____» _____ 20____ р.

Завідувач кафедри _____

(науковий ступінь, вчене звання)

(підпис)

(прізвище, ім'я, по батькові)

«_____» _____ 20____ р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

(науковий ступінь, вчене звання)

(підпис)

(прізвище, ім'я, по батькові)

АНОТАЦІЯ

Волков О.О. Розробка системи управління «Розумною теплицею». – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 122 – комп'ютерні науки. – Поліський національний університет, Житомир, 2025.

У роботі представлено процес розробки автоматизованої системи управління «Розумною теплицею», що базується на сучасних смарт-технологіях. Досліджено основні принципи функціонування теплиць та фактори, які впливають на ріст і розвиток рослин. Проведено аналіз існуючих інтелектуальних рішень у галузі агротехнологій та визначено доцільність їх впровадження для автоматичного регулювання мікроклімату. Запропоновано архітектуру системи, що включає апаратну та програмну складові, з обґрунтуванням вибору відповідних технологій. Описано процес реалізації системи, включно з функціональними модулями та прикладами сценаріїв використання. Розроблене рішення дозволяє підвищити ефективність агровиробництва, оптимізувати використання ресурсів та забезпечити стабільні умови для вирощування рослин.

Ключові слова: розумна теплиця, сенсори, Arduino, мікроклімат, автоматизація.

SUMMARY

Volkov O.O. Development of a “Smart Greenhouse” management system. – Qualification work in the form of a manuscript.

Qualification work for obtaining a bachelor’s degree in specialty 122 – computer science. – Polissia National University, Zhytomyr, 2025.

The work presents the process of developing an automated “Smart Greenhouse” management system based on modern smart technologies. The basic principles of greenhouse operation and factors that influence plant growth and development are studied. An analysis of existing intelligent solutions in the field of agrotechnology is carried out and the feasibility of their implementation for automatic microclimate regulation is determined. The system architecture is proposed, including hardware and software components, with a justification for the choice of appropriate technologies. The process of implementing the system is described, including functional modules and examples of usage scenarios. The developed solution allows you to increase the efficiency of agricultural production, optimize resource use and ensure stable conditions for growing plants.

Keywords: smart greenhouse, sensors, Arduino, microclimate, automation.

ЗМІСТ

АНОТАЦІЯ.....	3
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
1 ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1. Загальні принципи функціонування теплиць та фактори, що впливають на ріст рослин.....	9
1.2. Аналіз смарт-технологій.....	11
Висновки до першого розділу.....	13
2 ПЛАНУВАННЯ АРХІТЕКТУРИ ТА КОМПОНЕНТІВ СИСТЕМИ УПРАВЛІННЯ РОЗУМНОЮ ТЕПЛИЦЕЮ.....	15
2.1 Архітектура системи управління розумною теплицею.....	15
2.2 Вибір та обґрунтування технологій для розробки системи.....	20
Висновки до другого розділу.....	21
3 РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ РОЗУМНОЮ ТЕПЛИЦЕЮ.....	23
3.1 Програмна реалізація роботи функцій та розробка апаратної частини.....	23
3.2 Сценарії використання системи.....	25
Висновки до третього розділу.....	26
ВИСНОВКИ.....	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	29
ДОДАТКИ.....	34

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

ОС – операційна система

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

СКБД – система керування базою даних

ШІ – штучний інтелект

HTTP – HyperText Transfer Protocol

IoT – Internet of Things (Інтернет речей)

ВСТУП

У сучасних умовах зростання населення планети та необхідності забезпечення продовольчою безпекою, питання підвищення ефективності сільськогосподарського виробництва набуває особливої важливості. Традиційні методи вирощування культур у закритому ґрунті характеризуються високими витратами людських ресурсів, нестабільністю умов вирощування та значними втратами через несвоєчасне реагування на зміни мікроклімату. Автоматизація процесів управління тепличним господарством дозволяє вирішити ці проблеми, забезпечуючи оптимальні умови для росту рослин, мінімізуючи людський фактор та підвищуючи якість продукції.

Мета дослідження. Розробити та реалізувати автоматизовану систему управління тепличним господарством з можливістю дистанційного моніторингу та контролю параметрів мікроклімату.

Завдання дослідження:

- проаналізувати існуючі рішення автоматизації тепличних господарств та визначити їх переваги й недоліки;
- дослідити сучасні технології IoT та сенсорні системи для моніторингу параметрів мікроклімату;
- спроектувати архітектуру системи управління теплицею з урахуванням вимог масштабованості та надійності;
- реалізувати прототип системи з використанням мікроконтролерних платформ та датчиків;
- створити інтерфейс для дистанційного моніторингу та управління системою;

Методи дослідження. У роботі використовувались методи системного аналізу для дослідження предметної області, методи проєктування програмно-апаратних комплексів, алгоритми автоматичного регулювання, методи розробки веб-додатків.

Об'єктом дослідження є процес функціонування та управління тепличним господарством.

Предметом дослідження виступає система управління «розумною теплицею» – її архітектура, технологічна складова, алгоритми роботи та взаємодія програмно-апаратних компонентів.

За темою дослідження автором підготовлено два тези опубліковані у наступних збірниках:

- Безпека, технології, інновації: нові горизонти : збірник праць учасників міжфакультетської науково-практичної інтернет-конференції здобувачів вищої освіти і молодих вчених, 12 листопада 2024 р. Житомир : Поліський національний університет, 2024. 102 с.
- Інформаційні технології та моделювання систем : збірник праць учасників Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих вчених, 15 травня 2025 р. Житомир : Поліський національний університет, 2025. 91 с.

Практичне значення роботи полягає у створенні доступної системи управління теплицею, що може бути впроваджена у малих або середніх агропідприємствах, дослідницьких проєктах або навчальних закладах. Розроблена система дозволяє автоматизувати процеси догляду за рослинами, зменшити витрати ресурсів та покращити якість продукції.

Структура роботи. Кваліфікаційна робота складається з трьох розділів. У першому розглянуто теоретичні основи функціонування теплиць і смарт-технологій. У другому – описано архітектуру системи та обґрунтовано вибір технологій. У третьому – наведено реалізацію системи та сценарії її використання.

1 ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Загальні принципи функціонування теплиць та фактори, що впливають на ріст рослин

Розумні теплиці базуються на інтеграції IoT, сенсорних систем та алгоритмів штучного інтелекту для автоматизації контролю параметрів середовища. Ключові компоненти включають мережу датчиків (температура, вологість, CO₂, освітлення), актуатори (вентиляція, полив, опалення) та централізовану систему аналізу даних [1, 4]. Наприклад, система Serragrow 4.0 (Ofylia) використовує зовнішні та внутрішні сенсори для моніторингу ґрунтової вологи, сонячної радіації та вологості листя, автоматично коригуючи провітрювання та зрошення [2].

Основними факторами впливу на ріст рослин є:

- температура – оптимізація за допомогою термостатів та систем обігріву/охолодження, що підтримують діапазон 20–30°C [1, 6];
- вологість – датчики вологості повітря та ґрунту керують поливом, запобігаючи грибковим захворюванням або дефіциту води [3, 5];
- CO₂ – концентрація вуглекислого газу регулюється через вентиляцію або генератори, підвищуючи ефективність фотосинтезу на 12–14% [7];
- освітлення – LED-системи з динамічним спектром компенсують недостатнє природне світло, імітуючи оптимальний фотоперіод [4, 5].

Сучасні системи, такі як EcoGrow, використовують предиктивні алгоритми, які аналізують історичні дані та прогнози погоди для проактивного управління. Це дозволяє зменшити витрати енергії на 15–20% та підвищити врожайність на 25–30% порівняно з традиційними теплицями [2, 3]. Однак ключовим залишається ризик технологічних збоїв – наприклад, неправильні показання датчиків вологості можуть призвести до перезволоження ґрунту та гниття коренів [6]. Дослідження підкреслюють [36], що ефективність розумних теплиць залежить від синхронізації всіх компонентів. Системи на основі ШІ автоматично балансують параметри,

враховуючи взаємодію факторів – наприклад, підвищення CO₂ потребує паралельного збільшення освітлення для максимізації фотосинтезу [3, 7].

У сучасних IoT-системах для розумних теплиць найчастіше використовуються такі протоколи комунікації:

- MQTT (Message Queueing Telemetry Transport) – легковаговий протокол з архітектурою publish/subscribe, ідеальний для передачі даних від сенсорів у середовищах з обмеженою пропускнуою здатністю та енергоспоживанням. MQTT забезпечує надійну, асинхронну передачу даних із низькими накладними витратами, що робить його популярним у сільському господарстві для моніторингу температури, вологості, освітлення тощо [36].
- HTTP (HyperText Transfer Protocol) – широко використовується для обміну даними через REST-API, зручно інтегрується з веб-додатками. Однак HTTP є протоколом із запит-відповідь, що потребує більших ресурсів і не завжди оптимальний для енергоефективних IoT-пристроїв, але підходить для систем з меншими обмеженнями по енергоспоживанню і там, де потрібна інтеграція з існуючими веб-сервісами [37].
- WebSocket – протокол, що забезпечує двонаправлену (full-duplex) комунікацію між клієнтом і сервером у реальному часі. Використовується для передачі даних сенсорів і команд управління в режимі реального часу, що особливо корисно для інтерактивних веб-інтерфейсів розумних теплиць і систем автоматизації. WebSocket дозволяє серверу надсилати оновлення клієнтам без необхідності їхнього запиту, що підвищує оперативність реакції системи [38].

Таким чином, вибір IoT-протоколу у смарт-технологіях теплиць базується на балансі між дальністю передачі, енергоспоживанням, пропускнуою здатністю та вимогами до затримок. Часто застосовується комбінований підхід, де MQTT використовується для сенсорних мереж, HTTP

– для інтеграції з веб-сервісами, а WebSocket - для забезпечення реального часу у взаємодії користувача з системою.

1.2. Аналіз смарт-технологій

Аналіз смарт-технологій у розумних теплицях свідчить про те, що сучасні системи ґрунтуються на трьох ключових технологіях: Інтернеті речей (IoT) із сенсорними мережами, автоматизованих системах управління та штучному інтелекті (ШІ) з машинним навчанням [8]. Датчики IoT забезпечують збір даних у реальному часі про температуру, вологість ґрунту, рівень поживних речовин та якість повітря, що дозволяє створювати оптимальні умови для росту рослин і зменшувати споживання ресурсів. Автоматизовані системи виконують завдання зрошення, клімат-контролю та освітлення, знижуючи людський фактор і підвищуючи ефективність [8].

Штучний інтелект аналізує великі обсяги даних, прогнозує потреби рослин, виявляє ранні ознаки хвороб і шкідників, а також оптимізує використання води та добрив. Це дозволяє підвищити врожайність і зменшити витрати [8, 9]. Окрім того, у розумних теплицях застосовують гідропонні системи, що забезпечують безґрунтове вирощування культур із точним контролем поживного розчину, що сприяє сталому землеробству та цілорічному виробництву [9]. Водночас висока вартість впровадження та складність інтеграції таких технологій залишаються викликами, особливо для малих і середніх фермерів [8, 10].

Для підвищення точності контролю застосовують тензодатчики та вагові системи, які допомагають оптимізувати полив і запобігати захворюванням рослин, що підвищує продуктивність при мінімальному використанні ресурсів [11]. Ринок охолодження теплиць також демонструє значне зростання, що свідчить про важливість клімат-контролю в системах розумних теплиць [12]. Загалом, розвиток смарт-технологій у тепличному господарстві сприяє підвищенню ефективності, сталості та екологічності

сільського господарства, хоча потребує подолання фінансових і технічних бар'єрів [13]. Сучасні смарт-технології значно розширюють можливості автоматизації тепличного господарства, забезпечуючи точний контроль кліматичних параметрів, поливу та освітлення. Для кращого розуміння їх практичного застосування доцільно порівняти існуючі комерційні рішення з розробленою системою (табл. 1.1), враховуючи ключові характеристики, такі як автономність, гнучкість управління, вартість та освітній потенціал.

Таблиця 1.1 – Порівняння існуючих рішень

Характеристика	(Growlink)	(Autogrow)	(Розумна теплиця на Arduino)
Тип контролера	Growlink	Autogrow	Arduino Nano
Типи сенсорів	Температура, вологість, CO ₂	Температура, ЕС, вологість, освітлення	BME280 (темп, вологість), освітленість, волога ґрунту
Інтерфейс	Мобільний застосунок	Хмара + ПК-інтерфейс	Nextion дисплей + Веб-інтерфейс
Зв'язок	Wi-Fi (через хмару)	Ethernet / Wi-Fi	UART + Wi-Fi
Можливість ручного керування	Частково (тільки через застосунок)	Обмежено (віддалено)	Повноцінне ручне керування через дисплей
Автоматичне керування	Так, на основі сценаріїв	Так	Так, адаптується під культуру
Модульність / розширення	Низька (закрита система)	Середня	Висока (відкрите ПЗ / можна додати датчики, реле)
Ціна	від 400–800 \$	від 700–1500 \$	≈ 60–80 \$ (Arduino, датчики, реле, дисплей)

Проведений порівняльний аналіз різних систем керування агрокультурами демонструє суттєві відмінності у функціональності, архітектурі, можливостях керування та вартості. Комерційні рішення, такі як Growlink та Autogrow, пропонують переважно закриту або частково розширювану архітектуру, що обмежує користувача у можливостях модифікації та розширення системи. Вони забезпечують базові функції автоматичного контролю, проте мають обмежені можливості для ручного втручання, що може бути критичним у складних або нестандартних умовах вирощування. Крім того, висока вартість цих систем (від 400 до 1500 доларів

США) робить їх менш доступними для широкого кола користувачів, особливо в умовах обмеженого бюджету.

Натомість рішення на базі Arduino, представлене як «Розумна теплиця», вирізняється високою гнучкістю та відкритістю. Завдяки можливості підключення додаткових датчиків і реле, а також повноцінному ручному керуванню через інтерактивний дисплей, ця система дозволяє користувачам адаптувати її під індивідуальні потреби і особливості конкретних культур. Вартість реалізації такої системи є значно нижчою (близько 60–80 доларів США), що робить її привабливою для аматорів, невеликих фермерських господарств та дослідників.

Таким чином, Arduino-рішення є оптимальним вибором для тих, хто цінує доступність, масштабованість і можливість гнучкого налаштування системи. Воно сприяє розвитку інноваційних підходів у агротехнологіях, дозволяючи поєднувати автоматизацію з індивідуальним контролем, що особливо важливо для інклюзивних та адаптивних агрокультурних практик. У перспективі це відкриває широкі можливості для подальшого розвитку і впровадження web-орієнтованих систем інклюзивної підтримки в агросфері.

Висновки до першого розділу

У рамках дослідження було розглянуто загальні принципи функціонування розумних теплиць та фактори, що впливають на ріст рослин, а також проведено аналіз сучасних смарт-технологій, які застосовуються у тепличному господарстві. Визначено, що автоматизація контролю кліматичних параметрів, поливу, освітлення та внесення добрив за допомогою сенсорів, систем Інтернету речей і штучного інтелекту суттєво підвищує ефективність вирощування рослин і оптимізує використання ресурсів.

Також було проаналізовано переваги і виклики впровадження таких технологій, зокрема питання вартості, складності інтеграції та необхідності

підвищення надійності систем. Отримані результати підкреслюють важливість комплексного підходу до проєктування і експлуатації розумних теплиць для досягнення сталого і високоефективного сільського господарства. Створено порівняльну таблицю існуючих рішень.

2 ПЛАНУВАННЯ АРХІТЕКТУРИ ТА КОМПОНЕНТІВ СИСТЕМИ УПРАВЛІННЯ РОЗУМНОЮ ТЕПЛИЦЕЮ

2.1 Архітектура системи управління розумною теплицею

В основі архітектури системи управління розумною теплицею лежить Arduino Nano, до якого підключені всі основні сенсори, виконавчі пристрої та модулі зв'язку. Для збору та обробки даних про мікроклімат використовується сенсор BME280, який вимірює температуру, вологість і атмосферний тиск, а також датчик вологості ґрунту FC-28, що дозволяє контролювати стан поливу.

Освітленість у теплиці відстежує фотодатчик KY-026. Для керування потужними навантаженнями, такими як електроклапан для поливу, LED-стрічка для підсвітки та серво-привід або електропоршень для відкривання віконець, використовується модуль реле на 4 виходи, який дозволяє підключати до трьох лампочок на 12В або інші пристрої, що працюють від постійного струму. Вся система живиться від комп'ютерного блоку живлення на 250 Вт, який забезпечує необхідні рівні напруги для різних компонентів.

Для організації інтерфейсу користувача застосовується сенсорний дисплей Nextion, на якому виводиться інформація про стан системи та дозволяється налаштовувати параметри керування. Для розширення можливостей віддаленого моніторингу та керування в системі присутній модуль ESP8285, який забезпечує бездротовий зв'язок через Wi-Fi і може передавати дані на хмарний сервер або мобільний додаток. Зчитування та обробка сигналів від усіх сенсорів, а також керування реле та серво-приводом здійснюється Arduino Nano, який отримує живлення та керуючі сигнали через відповідні лінії, що чітко розділені за кольорами на схемі (рис. 2.1):

- чорний – GND;
- червоний – 5V;
- жовтий – 12V;
- помаранчевий – 3.3V;

- зелений – сигнальні дроти.

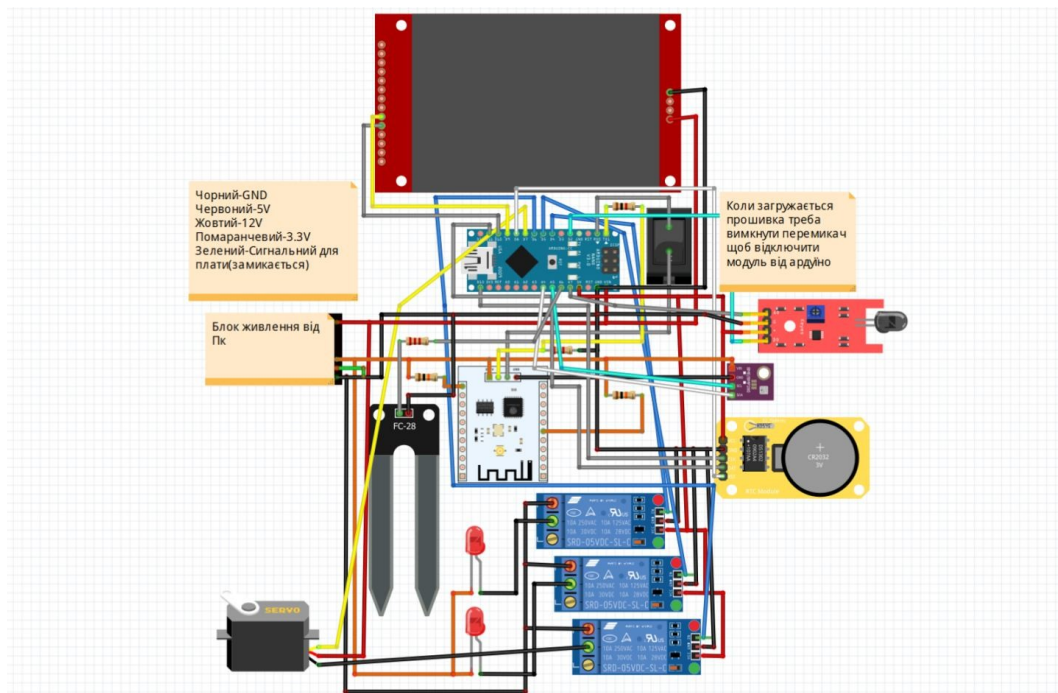


Рисунок 2.1 – Схема системи

Уся система побудована таким чином, щоб забезпечити автоматичне підтримання оптимальних умов для рослин: при недостатній вологості ґрунту вмикається полив через електроклапан, при низькому рівні освітленості активується LED-стрічка, а при підвищенні температури чи вологості відкриваються віконця за допомогою серво-приводу або поршня. Всі ці процеси можуть відображатися та налаштовуватися через Nxtion, а також контролюватися дистанційно через Wi-Fi завдяки ESP8285.

Також система є локальною і працює в межах домашньої Wi-Fi мережі або безпосередньо через сенсорний дисплей. Відсутність персональних кабінетів, веб-інтерфейсів та хмарних сервісів із авторизацією зводить до мінімуму потребу в складних заходах кібербезпеки. Віддалений доступ обмежений локальною мережею, що суттєво знижує ризики несанкціонованого втручання і забезпечує достатній рівень захисту для навчальних та демонстраційних цілей.

Архітектура застосунку, описана на UML-діаграмі послідовностей (рис. 2.2 – 2.4) детально візуалізує взаємодію ключових компонентів системи для

управління процесами вирощування, ймовірно, рослин або інших культур, де користувач взаємодіє з дисплеєм, який, у свою чергу, обмінюється даними з контролером Arduino, базою даних та сенсорами і пристроями.

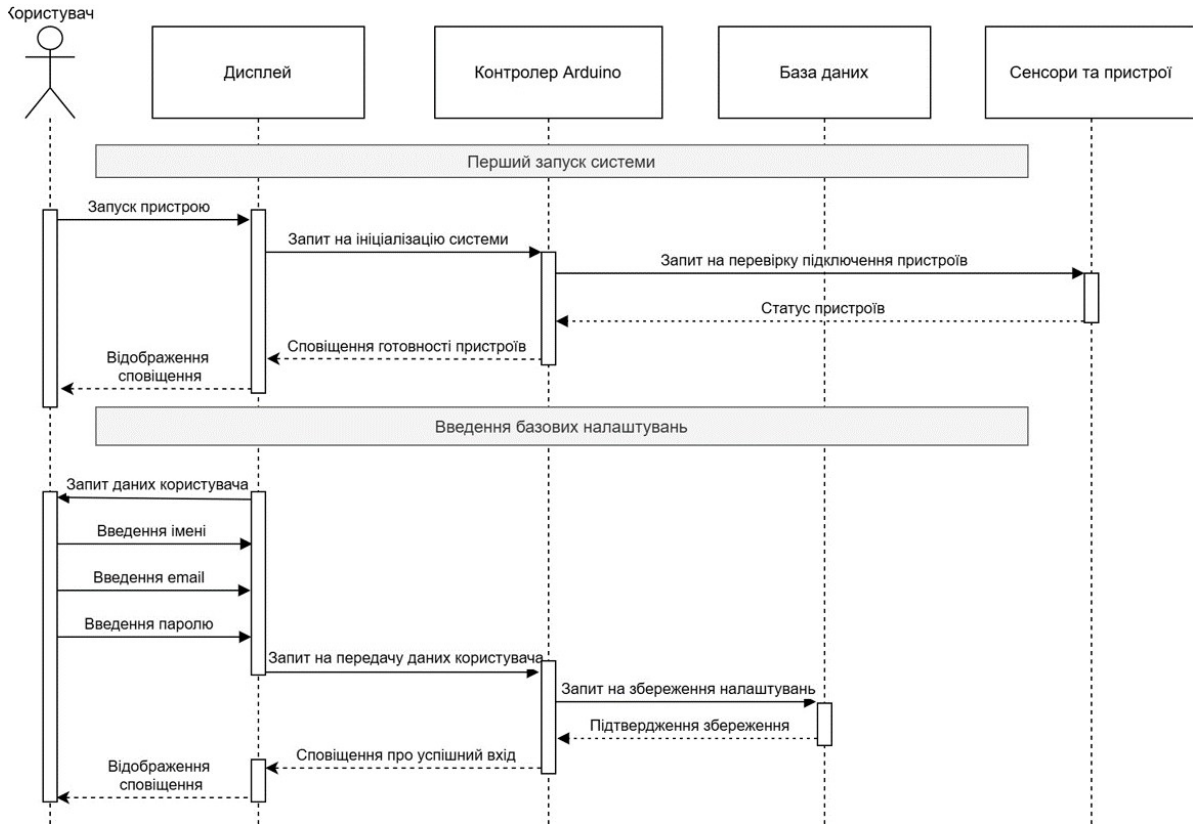


Рисунок 2.2 – UML-діаграма полідовностей (етап першого запуску та введення базових налаштувань)

При першому запуску системи користувач ініціює запуск пристрою, що викликає запит на ініціалізацію системи до дисплея. Дисплей відправляє запит на перевірку підключення пристроїв до контролера Arduino, який надсилає цей запит до сенсорів та пристроїв. Сенсори та пристрої відповідають контролеру Arduino, надаючи статус пристроїв, який потім передається дисплею як сповіщення готовності пристроїв. Після цього дисплей відображає сповіщення користувачу. Далі відбувається введення базових налаштувань. Користувач здійснює запит даних користувача, вводячи своє ім'я, email та пароль через дисплей. Дисплей відправляє запит на передачу даних користувача до контролера Arduino, який надсилає запит на збереження налаштувань до бази даних. База даних підтверджує збереження,

контролер Arduino передає сповіщення про успішний вхід дисплею, а дисплей відображає сповіщення користувачу.

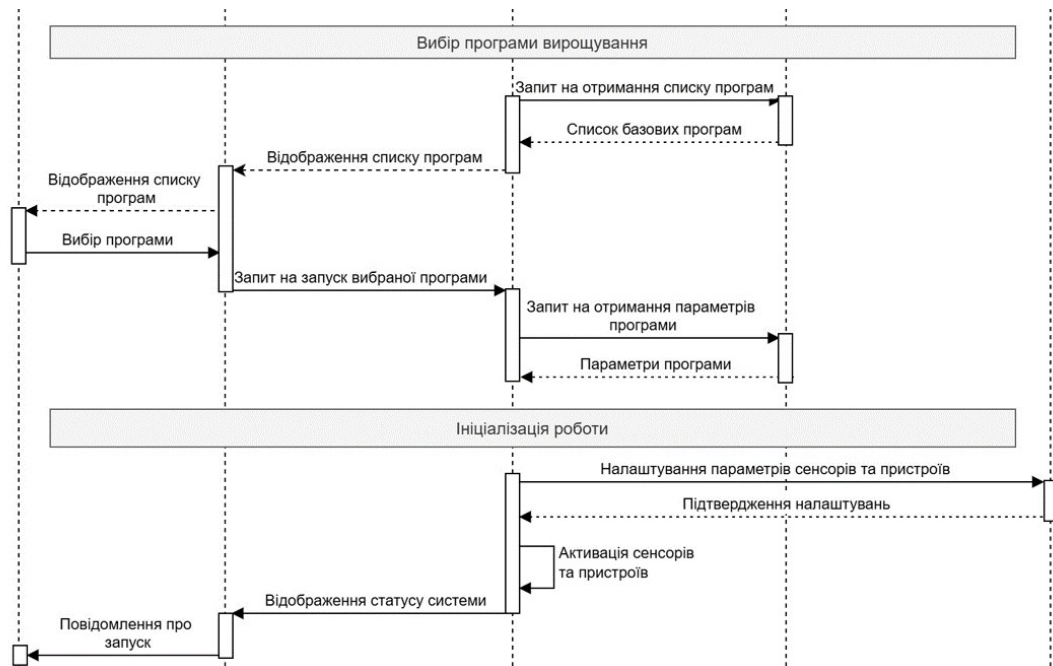


Рисунок 2.3 – UML-діаграма полідовностей (етап вибору програм вирощування та ініціалізації роботи)

На етапі вибору програми вирощування користувач робить запит на отримання списку програм до дисплея, який надсилає його контролеру Arduino. Контролер Arduino запитує список базових програм у бази даних. База даних надає список базових програм контролеру Arduino, який передає його дисплею для відображення списку програм. Користувач вибирає програму, після чого дисплей відправляє запит на запуск вибраної програми до контролера Arduino. Контролер Arduino робить запит на отримання параметрів програми до бази даних, яка повертає параметри програми. Далі йде ініціалізація роботи, де контролер Arduino налаштовує параметри сенсорів та пристроїв, передаючи їх сенсорам та пристроям. Сенсори та пристрої підтверджують налаштування. Контролер Arduino активує сенсори та пристрої, а дисплей відображає статус системи, після чого користувач отримує повідомлення про запуск.

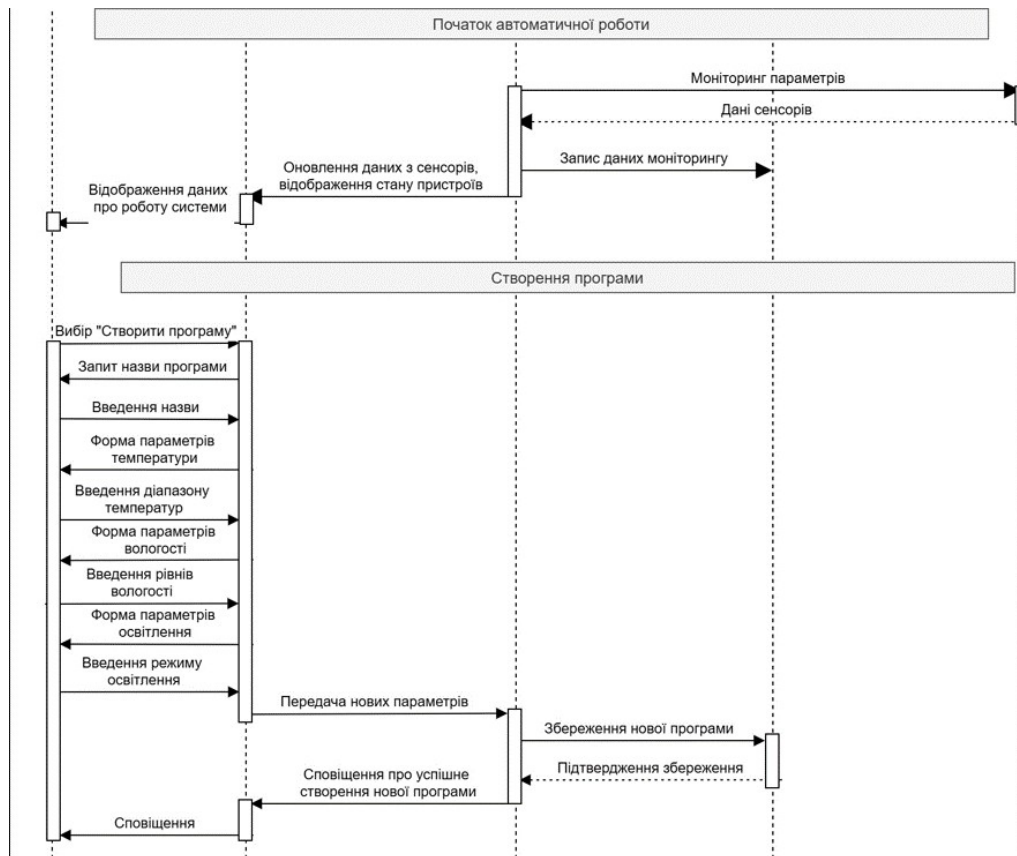


Рисунок 2.4 – UML-діаграма полідовностей (етап автоматичної роботи та створення програми)

Під час автоматичної роботи контролер Arduino здійснює моніторинг параметрів, отримуючи дані сенсорів від сенсорів та пристроїв, і записує дані моніторингу до бази даних. Дисплей оновлює дані з сенсорів та відображає стан пристроїв, показуючи дані про роботу системи користувачу. Крім того, передбачена можливість створення програми. Користувач робить вибір «Створити програму», потім через дисплей відбувається запит назви, введення назви, введення діапазону температури через форму параметрів температури, введення рівнів вологості через форму параметрів вологості, та введення режиму освітлення через форму параметрів освітлення. Всі ці нові параметри передаються дисплеєм контролеру Arduino. Контролер Arduino надсилає запит на збереження нової програми до бази даних, яка підтверджує збереження. Контролер Arduino надсилає сповіщення про успішне створення нової програми дисплею, який відображає сповіщення користувачу. Ця

архітектура дозволяє гнучко управляти пристроями, моніторити їхній стан та налаштовувати програми роботи.

2.2 Вибір та обґрунтування технологій для розробки системи

Для побудови автоматизованої системи управління «розумною теплицею» було обрано сучасні апаратні та програмні засоби, що забезпечують надійність, енергоефективність і можливість розширення функціональності. Основою апаратної частини є мікроконтролер Arduino Nano, який відповідає за зчитування показників з датчиків, керування виконавчими механізмами та взаємодію з локальними інтерфейсами. Його було обрано завдяки компактним розмірам, достатній кількості входів/виходів і сумісності з численними модулями. Для реалізації бездротового зв'язку використовується ESP8285 – мікроконтролер із вбудованим Wi-Fi, що виконує роль шлюзу між системою та користувачем. У його пам'яті зберігаються веб-сторінки (index.html, стилі, скрипти), що дозволяє реалізувати повноцінний веб-інтерфейс без потреби в зовнішньому сервері. Через цей інтерфейс користувач має можливість переглядати поточні дані, керувати режимами роботи та отримувати віддалений доступ до теплиці.

Для локального керування та візуалізації показників використовується сенсорний дисплей, який спрощує взаємодію з користувачем. У середовищі було створено графічний інтерфейс із відображенням температури, вологості, освітлення, а також кнопками керування виконавчими механізмами. Дисплей напряду обмінюється командами з Arduino, дозволяючи реалізувати як автоматичний, так і ручний режим роботи.

У якості сенсорів було застосовано BME280 для вимірювання температури, вологості повітря та атмосферного тиску, FC-28 для визначення вологості ґрунту, та фоторезистор KY-02B для контролю рівня освітленості. Дані з цих датчиків є ключовими для прийняття рішень системою щодо поливу, освітлення та вентиляції. Для керування виконавчими пристроями

(наприклад, водяним клапаном, LED-стрічкою чи вентилятором) використовується 4-канальний релейний модуль, який дозволяє безпечно комутувати потужні навантаження.

Серед виконавчих елементів планується використання LED-стрічки на 12 В для підсвітки в умовах недостатнього природного освітлення, електромагнітного клапана для автоматичного поливу, а також сервоприводу або лінійного поршня для відкривання вентиляційних вікон. Живлення всієї системи здійснюється через комп'ютерний блок живлення на 250 Вт, що дозволяє забезпечити стабільну роботу як логічної, так і силовій частини системи. Програмне забезпечення системи розроблено з використанням мови програмування C++. Прошивка Arduino відповідає за логіку опитування сенсорів, керування виконавчими механізмами та обмін даними з дисплеєм Nextion. ESP8285 також має власну прошивку, що відповідає за обробку HTTP-запитів, передачу даних та виведення веб-інтерфейсу. Веб-сторінки було створено вручну з використанням HTML, CSS та JavaScript, що дозволяє адаптувати інтерфейс під мобільні та десктопні пристрої. Завдяки розділенню функцій між мікроконтролерами, система демонструє високу стабільність та гнучкість у керуванні.

Таким чином, обрані апаратні та програмні компоненти забезпечують ефективну роботу системи управління теплицею, відкриваючи можливості для майбутньої модернізації, зокрема, інтеграції з хмарними сервісами, додаванням системи сповіщень або застосунку для смартфона.

Висновки до другого розділу

У другому розділі детально розглянуто архітектуру та вибір технологій для розробки системи управління розумною теплицею. Обрана апаратна база відзначається компактністю, енергоефективністю та гнучкістю, що дозволяє адаптувати систему під різні умови та розширювати її функції в майбутньому. Програмне забезпечення, розроблене на основі C++ та веб-технологій,

забезпечує стабільну роботу, автоматизацію ключових процесів (полив, вентиляція, освітлення) і можливість як автоматичного, так і ручного керування.

3 РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ РОЗУМНОЮ ТЕПЛИЦЕЮ

3.1 Програмна реалізація роботи функцій та розробка апаратної частини

Система керування агрокультурами базується на мікроконтролері ESP8266, який забезпечує зв'язок через Wi-Fi, веб-інтерфейс та комунікацію з Arduino для обробки даних сенсорів і керування виконавчими пристроями. Arduino зчитує дані з сенсорів температури (BME280), вологості ґрунту (аналоговий датчик) та освітленості (фоторезистор), конвертуючи аналогові значення у відсотки.

Для автоматичного керування використовуються три релейних модулі: обігріву/охолодження (реле 1), поливу (реле 2) та додаткового освітлення (реле 3). Логіка керування реле залежить від режиму autoMode: якщо він активний, стан реле змінюється на основі порівняння поточних даних з заданими значеннями (temp, humi, light).

Команди з інтерфейсу передаються через UART у форматі рядків (дод. А, рис. 1):

- TEMP=25, HUMI=60, LIGHT=500 – встановлення цільових параметрів;
- SAVE – збереження налаштувань у EEPROM;
- RESET – скидання налаштувань до стандартних;
- SLEEPMODE=ON/OFF – перемикання режиму сну.

Дані зберігаються в EEPROM за допомогою функцій saveSettingsToEEPROM() та loadSettingsFromEEPROM(), які записують або зчитують значення temp, humi, light і sleepMode. Скидання (resetEEPROM()) обнуляє ці параметри та деактивує режим сну (дод. А, рис. 2). Nextion дисплей відображає інтерфейс з вибором культури (наприклад, «Полуниця», «Огірок»), де кожна культура має власні уставки для температури, вологості та освітлення.

При зміні культури дисплей формує команди типу TEMP=25 і відправляє їх на Arduino через UART, додаючи завершальні байти FF FF FF. Для ручного введення параметрів значення з текстових полей (t1.txt, t2.txt, t3.txt) конвертуються у числовий формат і передаються разом із префіксами (наприклад, t10.txt=«TEMP=» + t1.txt). Лістинг коду наведено в додатку С.

Основні компоненти системи:

- Мікроконтролер відповідає за обробку логіки керування.
- сенсори для зчитування показників середовища (температура, вологість ґрунту, освітленість).
- реле: керування зовнішніми пристроями (обігрів, зволоження, освітлення).
- EEPROM: збереження налаштувань.
- Nextion: вивід інформації та взаємодія з користувачем.
- ESP: передача даних і керування через веб-інтерфейс.

Для збереження параметрів у EEPROM:

```
void saveSettingsToEEPROM() {
  Serial.println("Збереження в EEPROM...");
  EEPROM.put(EEPROM_TEMP, temp);
  EEPROM.put(EEPROM_HUMI, humi);
  EEPROM.put(EEPROM_LIGHT, light);
  EEPROM.put(EEPROM_SLEEP_MODE, sleepMode);
}
```

Цей фрагмент забезпечує збереження температури, вологості, освітлення, а також режиму сну після натискання кнопки SAVE на дисплеї або в інтерфейсі.

Конвертація значення з повзунка в текстове поле у Nextion – «conv h1.val,t1.txt,0,1»

Режим автоматичного керування

```
void controlDevicesAutomatically(float temp, int soil, int light) {
  digitalWrite(relay1, (temp < currentParams.temp) ? LOW : HIGH);
}
```



```
digitalWrite(relay2, (soil < currentParams.humi) ? LOW : HIGH);
digitalWrite(relay3, (light < currentParams.light) ? LOW : HIGH);
}
```

Це автоматично вмикає чи вимикає реле відповідно до заданих параметрів культури та актуальних сенсорних даних.

Обробка помилок: якщо сенсор BME280 не відповідає, у setup() виводиться повідомлення:

```
if (!bme.begin(0x76)) {
  Serial.println(" ✕ Помилка ініціалізації BME280!");
}
```

Якщо приходить невідома або пошкоджена команда, у функції handleCommand():

```
if (command.length() < 3) {
  Serial.println(" ⚠ Ігноровано шум або коротку команду.");
  return;
}
```

Якщо дисплей відключено – система продовжує працювати автономно, керування можливе через веб-інтерфейс ESP. Якщо протягом 3 хвилин немає активності, дисплей вимикається командою: «nextion.print("sleep=1");», але при дотику дисплей пробуджується і оновлює таймер: «nextion.print("sleep=0");»

3.2 Сценарії використання системи

При увімкненні системи на дисплеї Nextion відображається екран завантаження, що сигналізує про ініціалізацію всіх модулів і перевірку роботи сенсорів та виконавчих пристроїв. Після завершення завантаження користувач потрапляє в меню програм, де може обрати потрібну культуру рослин із запропонованого списку. Для кожної культури передбачені стартові налаштування, які користувач може коригувати за допомогою повзунків на

екрані, наприклад, встановлювати оптимальні параметри вологості, температури чи освітлення. Також є можливість створити власний профіль культури з унікальними параметрами або скинути всі налаштування, щоб система не виконувала автоматичне керування.

Далі користувач може перейти в головне меню і відкрити сервісне меню, де відображаються поточні дані з усіх підключених датчиків – температури, вологості ґрунту, освітленості тощо. Тут можна перевірити коректність роботи сенсорів, а також вручну вмикати або вимикати реле, що керують електроклапаном, LED-стрічкою та серво-приводом.

У меню налаштувань користувач має змогу сконфігурувати параметри мережі Wi-Fi, задаючи логін і пароль для підключення модуля ESP8285. Також у цьому розділі можна скинути всі налаштування системи до заводських та активувати режим сну дисплея для економії енергії. На головному екрані додатково розміщено QR-код, який при скануванні перенаправляє користувача на веб-інтерфейс системи. Через цей інтерфейс можна віддалено моніторити стан теплиці і керувати її режимами роботи зі смартфона або комп'ютера, що значно підвищує зручність експлуатації.

Таким чином, система забезпечує інтуїтивний і гнучкий інтерфейс для користувача, дозволяючи як автоматизувати процеси в теплиці, так і здійснювати ручне керування та віддалений моніторинг.

Висновки до третього розділу

У розділі описано програмну реалізацію функцій системи управління розумною теплицею та розробку апаратної частини, що забезпечують ефективну автоматизацію агрокультур. Розроблені сценарії використання демонструють інтуїтивно зрозумілий інтерфейс, що дозволяє користувачу легко налаштовувати параметри для різних культур, контролювати роботу сенсорів і виконавчих механізмів, а також здійснювати ручне чи автоматичне

керування. Впровадження режиму сну дисплея та можливість скидання налаштувань підвищують енергоефективність і зручність обслуговування.

ВИСНОВКИ

У процесі виконання дипломного проєкту було розроблено ефективну систему управління розумною теплицею на базі мікроконтролера Arduino Nano, сенсорного дисплея Nxtion та модуля Wi-Fi ESP8285, яка забезпечує автоматичний контроль мікроклімату для оптимізації росту рослин. Проведено детальний аналіз сучасних смарт-технологій, досліджено ключові фактори, що впливають на врожайність, та реалізовано архітектуру, яка дозволяє інтегрувати контроль освітлення, поливу, температури з можливістю локального й дистанційного керування.

У результаті розробки вдалося досягти основної мети – створити доступну, гнучку та ефективну систему для підтримання оптимальних умов у теплиці, що сприяє підвищенню продуктивності вирощування рослин і раціональному використанню ресурсів. Ця система може бути розширена за рахунок додаткових сенсорів або модулів, інтегрована з хмарними платформами для агромоніторингу та адаптована під різні типи тепличних господарств.

Отримані результати можуть слугувати основою для впровадження подібних рішень у малих фермерських господарствах, дослідницьких установах або освітніх проєктах, а також бути розвинутими в майбутніх дослідженнях, спрямованих на впровадження машинного навчання, енергозбереження й мобільної автоматизації в аграрному секторі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. GitHub - PramodPatilGS/Smart-Greenhouse-Monitoring-System-Based-On-IOT: Greenhouses have to be monitored manually whereas by making it a smart greenhouse that s as per the requirement are used to monitor greenhouse and perform the actions according to the parameters sensed. GitHub. URL: <https://github.com/PramodPatilGS/Smart-Greenhouse-Monitoring-System-Based-On-IOT> (date of access: 09.01.2025).
2. The traditional greenhouse is now smarter. Hortidaily: global greenhouse news. URL: <https://www.hortidaily.com/article/9724624/the-traditional-greenhouse-is-now-smarter/> (date of access: 15.06.2025).
3. [Smart Climate Control System for Greenhouse based on Temperature, Humidity, and CO2. Let's Save The World For The Future Generation | Smart Life.](https://fse-automation.com/smart-climate-control-system-for-greenhouse-based-on-temperature-humidity-and-co2) URL: <https://fse-automation.com/smart-climate-control-system-for-greenhouse-based-on-temperature-humidity-and-co2> (date of access: 09.01.2025).
4. The importance of sensors and data for the smart greenhouse - Light Science Technologies. Light Science Technologies. URL: <https://lightsciencetech.com/sensors-data-smart-greenhouses> (date of access: 15.01.2025).
5. IPPS | International Plant Propagator's Society. URL: https://ipps.org/uploads/docs/57_015.pdf (дата звернення: 15.01.2025).
6. Give your Plants the Best Suitable Environment for Green House. Softweb Solutions. URL: <https://www.softwebsolutions.com/resources/greenhouse-environment-control-system.html> (date of access: 15.01.2025).
7. Cho R. How Climate Change Will Affect Plants. State of the Planet. URL: <https://news.climate.columbia.edu/2022/01/27/how-climate-change-will-affect-plants> (date of access: 18.01.2025).
8. Які нові технології в розумних теплицях 2025?. Китайська оранжерея, Glass Prighous. URL: <https://ua.yt-greenhouse.com/news/what-is-the-new-technologies-in-smart-greenhou-83867050.html> (дата звернення: 18.01.2025).

9. Smart Agriculture: How Advanced Greenhouses Use Technology to Achieve Year-Round Hydroponic Growth and Environmental Harmony - Miilkiia. Smart Agriculture: How Advanced Greenhouses Use Technology to Achieve Year-Round Hydroponic Growth and Environmental Harmony - Miilkiia. URL: <https://www.miilkiiagrow.com/uk/news/smart-agriculture-how-advanced-greenhouses-use-technology-to-achieve-year-round-hydroponic-growth-an-1> (date of access: 18.01.2025).
10. Казанцев Д. AgriTech у 2025 році: як розвиваються агротехнології в Україні та світі. Mezha.Media. URL: <https://mezha.media/articles/yak-rozvivayetsya-rinok-agritech-v-ukrajini-ta-sviti-300226> (дата звернення: 18.01.2025).
11. Smart- теплиці з тензодатчиками ZEMIC: розумні технології для максимально корисних овочів і фруктів!. Zemic - виробництво і впровадження комплектуючих для вагових систем. URL: <https://zemic.com.ua/smart-teplytsi-z-tenzodatchyky-zemic-rozumni-tekhnologii-dlia-maksymalno-korysnykh-ovochiv-i-fruktiv> (дата звернення: 18.01.2025).
12. Ринок охолодження теплиць зростає до 3 мільярдів доларів. AGRIRADAR. URL: <https://agriradar.news/охолодження-теплиць/6807> (дата звернення: 25.01.2025).
13. Екологічні виклики сучасних теплиць. Vegetable - Садова Магія. URL: <https://vegetable.com.ua/ekologichni-aspekti-vedennya-teplichnogo-gospodarstva/> (дата звернення: 15.06.2025).
14. Смартрішення крокують агросвітом. iFarming. URL: <https://ifarming.ua/itehnologii/selektsiya/smartrishennya-krokuyut-agrosvitom> (дата звернення: 25.01.2025).
15. IoT у сільському господарстві: розумна теплиця, ферма. TotalAPI. URL: <https://totalapi.io/silske-gospodarstvo/> (дата звернення: 28.01.2025).
16. Китайська фабрика виробників розумних теплиць - Оптова торгівля розумними теплицями для продажу. Китайська оранжерея, Glass Prighous.

URL: <https://ua.yt-greenhouse.com/smart-greenhouses/> (дата звернення: 07.02.2025).

17. Сільськогосподарський IoT Smart Greenhouse – керування мобільним телефоном. Китайське сільськогосподарське іригаційне обладнання, машини для фертигації, виробники обладнання для контролю температури, завод - LVZHIXIN. URL: <https://ua.lzxgreenhouse.com/the-internet-of-things-control-system/agricultural-iot-smart-greenhouse-mobile.html> (date of access: 07.02.2025).

18. Розробка Smart Рішень: Які Знання та Навички Необхідні IoT Спеціалісту. Eastern Peak. URL: <https://careers.easternpeak.com/blog/iot-engineers-skills-to-develop-smart-solutions/> (дата звернення: 07.02.2025).

19. How to Build an Intelligent Greenhouse from Scratch - Miilkiia. How to Build an Intelligent Greenhouse from Scratch - Miilkiia. URL: <https://www.miilkiiaGrow.com/uk/news/how-to-build-an-intelligent-greenhouse-from-scratch/> (date of access: 08.02.2025).

20. Automated Greenhouse Management with IoT: Monitoring and Controlling. webbylab. URL: <https://webbylab.com/blog/smart-greenhouse-solutions-iot-based-environmental-monitoring-and-control/> (date of access: 09.02.2025).

21. GitHub - tabaraei/Smart-Greenhouse: This is an IOT project which aims to track greenhouse status, in addition to "Automatic" & "Manual" irrigation. GitHub. URL: <https://github.com/tabaraei/Smart-Greenhouse> (date of access: 11.02.2025).

22. Smart Greenhouse System, IoT Monitoring & Automation – Autogrow. Autogrow. URL: <https://autogrow.com/our-products-solutions/smart-greenhouse-monitoring-systems> (date of access: 11.02.2025).

23. Development of IoT Smart Greenhouse System for Hydroponic Gardens. arXiv.org. URL: <https://arxiv.org/abs/2305.01189> (date of access: 15.06.2025).

24. Monitoring of Smart Greenhouse Using Internet of Things (IoT) / S. A. Bharathi et al. SpringerLink. URL: https://link.springer.com/chapter/10.1007/978-981-99-9610-0_19 (date of access: 11.02.2025).

25. The Basics Of Microcontroller Programming. Build Electronic Circuits. URL: <https://www.build-electronic-circuits.com/microcontroller-programming/> (date of access: 15.02.2025).
26. Kohli V. Microcontroller Programming: Mastering the Foundation of Embedded Systems. Wevolver. URL: <https://www.wevolver.com/article/microcontroller-programming-mastering-the-foundation-of-embedded-systems> (date of access: 15.02.2025).
27. Microcontroller Programming with source code. Microcontroller Programming. URL: <https://www.microcontrollerprogramming.com/> (date of access: 28.03.2025).
28. Microcontroller Programming: from logic to final product | Detus. Detus. URL: <https://www.detus.pt/microcontroller-programming/> (date of access: 28.03.2025).
29. Instructables. Programming PIC Microcontrollers. Instructables. URL: <https://www.instructables.com/Programming-PIC-Microcontrollers/> (date of access: 28.03.2025).
30. Essentials of Microcontroller Use Learning about Peripherals: Programming Part 2. Renesas. URL: <https://www.renesas.com/en/support/engineer-school/mcu-programming-peripherals-06?srsId=AfmBOoqWzQscFuENYnMdik-TovU5bSpoM5kFpIOyNtr5MuGI1aZGCvqu> (date of access: 29.03.2025).
31. Development of a Microcontroller Based Automated Greenhouse Cultivation System for Mushrooms - Hardware. IEEE Xplore. URL: <https://ieeexplore.ieee.org/document/10339014/> (date of access: 29.03.2025).
32. GitHub - malmardi/Greenhouse-control-system-using-Arduino-microcontrollers: Course project: development of a distributed greenhouse monitoring system using Arduino microcontrollers and C#. GitHub. URL: <https://github.com/malmardi/Greenhouse-control-system-using-Arduino-microcontrollers> (date of access: 01.04.2025).

33. Greenhouse Monitoring using PIC MICROCONTROLLER. Hackster.io. URL: <https://www.hackster.io/Embedded-DIY-Tech/greenhouse-monitoring-using-pic-microcontroller-c76520> (date of access: 01.04.2025).
34. My code for an Automated Greenhouse. Arduino Forum. URL: <https://forum.arduino.cc/t/my-code-for-an-automated-greenhouse/968596> (date of access: 01.04.2025).
35. Instructables. Automation of Greenhouse Using Temperature and Humidity Sensor. Instructables. URL: <https://www.instructables.com/Automation-of-Greenhouse-Using-Temperature-and-Hum/> (date of access: 10.04.2025).
36. Zhang, Y., Li, X., & Wang, J. (2020). Integrated IoT and AI-based smart greenhouse system for precision agriculture. *Computers and Electronics in Agriculture*, 175, 105592. URL: <https://doi.org/10.1016/j.compag.2020.105592> (date of access: 04.01.2025).
37. Deep Convolutional and LSTM Networks on Multi-Channel Time Series Data for Gait Phase Recognition. MDPI. URL: <https://www.mdpi.com/1424-8220/21/3/789> (date of access: 15.04.2025).
38. WebSockets Standard. WebSockets Standard. URL: <https://websockets.spec.whatwg.org/> (date of access: 11.04.2025).
39. Волков О. О. ІНТЕЛЕКТУАЛЬНА АВТОМАТИЗАЦІЯ МІКРОКЛІМАТУ ТЕПЛИЦІ НА БАЗІ ARDUINO ТА ESP. ЗБІРНИК ПРАЦЬ УЧАСНИКІВ ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ І МОЛОДИХ ВЧЕНИХ : " ІНФОРМ. ТЕХНОЛОГІЇ ТА МОДЕЛЮВАННЯ СИСТЕМ", м. Житомир, 15 трав. 2025 р. (дата звернення: 15.05.2025)
40. Волков О. О. РОЗРОБКА СИСТЕМИ ДЛЯ КЕРУВАННЯ РОЗУМНОЮ ТЕПЛИЦЕЮ. ЗБІРНИК ПРАЦЬ УЧАСНИКІВ ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ І МОЛОДИХ ВЧЕНИХ : "БЕЗПЕКА, ТЕХНОЛОГІЇ, ІННОВАЦІЇ: НОВІ ГОРИЗОНТИ", м. Житомир, 12 листоп. 2024 р. (дата звернення: 12.11.2024)

ДОДАТКИ

ДОДАТОК А

```
void handleCommand(String command) {  
    command.trim();  
    if (command.length() < 2) return;  
  
    Serial.println("→ Команда: " + command);  
    lastActivityTime = millis();  
  
    if (command == "SAVE") {  
        saveSettingsToEEPROM();  
    } else if (command == "RESET") {  
        resetEEPROM();  
    } else if (command.startsWith("TEMP=")) {  
        temp = command.substring(5).toInt();  
    } else if (command.startsWith("HUMI=")) {  
        humi = command.substring(5).toInt();  
    } else if (command.startsWith("LIGHT=")) {  
        light = command.substring(6).toInt();  
    } else if (command == "SLEEPMODE=ON") {  
        sleepMode = true;  
        EEPROM.put(EEPROM_SLEEP_MODE, sleepMode);  
    } else if (command == "SLEEPMODE=OFF") {  
        sleepMode = false;  
        EEPROM.put(EEPROM_SLEEP_MODE, sleepMode);  
        wakeUpDisplay();  
    } else if (command == "SETTING_CROPS") {  
        autoMode = false;  
        sendSensors = false;  
    }  
}
```

Рисунок 1 – Розпізнання команд і запис їх

```
void saveSettingsToEEPROM() {  
    EEPROM.put(EEPROM_TEMP, temp);  
    EEPROM.put(EEPROM_HUMI, humi);  
    EEPROM.put(EEPROM_LIGHT, light);  
    EEPROM.put(EEPROM_RESET_FLAG, false);  
    EEPROM.put(EEPROM_SLEEP_MODE, sleepMode);  
}  
  
void loadSettingsFromEEPROM() {  
    EEPROM.get(EEPROM_TEMP, temp);  
    EEPROM.get(EEPROM_HUMI, humi);  
    EEPROM.get(EEPROM_LIGHT, light);  
    EEPROM.get(EEPROM_RESET_FLAG, resetFlag);  
    EEPROM.get(EEPROM_SLEEP_MODE, sleepMode);  
}  
  
void resetEEPROM() {  
    temp = 0;  
    humi = 0;  
    light = 0;  
    sleepMode = false;  
    resetFlag = false;  
    saveSettingsToEEPROM();  
}
```

Рисунок 2 – Скидання та збереження

ДОДАТОК В

```
if(currentCrop==0)
{
    t0.txt="Полуниця"
    h1.val=strawberry_temp
    h2.val=strawberry_humi
    h3.val=strawberry_light
}
if(currentCrop==1)
{
    t0.txt="Огірок"
    h1.val=cucumber_temp
    h2.val=cucumber_humi
    h3.val=cucumber_light
}
if(currentCrop==2)
{
    t0.txt="Помідор"
    h1.val=tomato_temp
    h2.val=tomato_humi
    h3.val=tomato_light
}
if(currentCrop==3)
{
    t0.txt="Перець"
    h1.val=pepper_temp
    h2.val=pepper_humi
    h3.val=pepper_light
}
```

```
}  
if(currentCrop==4)  
{  
    t0.txt="Баклажан"  
    h1.val=eggplant_temp  
    h2.val=eggplant_humi  
    h3.val=eggplant_light  
}  
if(currentCrop==5)  
{  
    t0.txt="Редиска"  
    h1.val=radish_temp  
    h2.val=radish_humi  
    h3.val=radish_light  
}  
if(currentCrop==6)  
{  
    t0.txt="Картопля"  
    h1.val=potato_temp  
    h2.val=potato_humi  
    h3.val=potato_light  
}  
if(currentCrop==7)  
{  
    t0.txt="Капуста"  
    h1.val=cabbage_temp  
    h2.val=cabbage_humi  
    h3.val=cabbage_light  
}  
if(currentCrop==8)
```

```

{
    t0.txt="Шпинат"
    h1.val=spinach_temp
    h2.val=spinach_humi
    h3.val=spinach_light
}
if(currentCrop==9)
{
    t0.txt="Крпн"
    h1.val=dill_temp
    h2.val=dill_humi
    h3.val=dill_light
}
if(currentCrop==10)
{
    t0.txt="Custom"
    h1.val=custom_temp
    h2.val=custom_humi
    h3.val=custom_light
}
// Конвертуємо вміст h1-h3 у текст ? показуємо в полях t1-t3
// Перетворюємо значення h1, h2, h3 у текст вручну через посередника
covx h1.val,t1.txt,0,1
covx h2.val,t2.txt,0,1
covx h3.val,t3.txt,0,1
print "SETTING_CROPS"
printh FF FF FF

```