

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«07» червня 2024 р

Пояснювальна записка

до кваліфікаційної роботи
бакалавра

на тему: «ІНФОРМАЦІЙНА СИСТЕМА «РОЗУМНА ТЕПЛИЦЯ»»

Захищено на засіданні
Атестаційної комісії № 42
протокол № __ від __.06.2024 р.
Оцінка _____ / _____
Голова Атестаційної комісії
_____ **СКОБ Ю. О.**

Виконав:
студент 4 курсу, групи КІ-41
Галузь знань: 12 – Інформаційні
технології
Спеціальність: 123 – Комп'ютерна
інженерія.

ГРИГОР'ЄВ Данило Романович 

Керівник: к.т.н., доцент ЗВО кафедри
ТПС

БИКОВА Тетяна Володимирівна 

Рецензент: к.ф.-м.н., доцент ЗВО, зав.
кафедри електроніки і управляючих
систем

ХРУСЛОВ Максим Михайлович 

Харків – 2024

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і двох додатків. Загальний обсяг роботи складає 78 сторінки, із яких 44 сторінок основної частини з 18 рисунками, 5 таблицями, 11 найменуваннями списку використаних джерел та двома додатками.

Метою кваліфікаційної роботи є підвищення рівня автоматизації систем управління мікрокліматом в теплицях.

Об'єкт дослідження – процеси управління мікрокліматом в теплицях.

Предмет дослідження – комп'ютерна система управління мікрокліматом в теплицях.

Проблема, яку вирішує кваліфікаційна робота, полягає у створенні інформаційної системи «Розумна теплиця», яка забезпечить автоматизацію процесів управління теплицею, включаючи іригацію, освітлення, опалення, вентиляцію та інші системи. Існуючі системи часто є недостатньо інтегрованими та вимагають значних витрат часу на налаштування і підтримку.

Область застосування – автоматизація управління теплицями. Розроблена інформаційна система «Розумна теплиця» може широко використовуватися в аграрному секторі, зокрема у фермерських господарствах та агропромислових підприємствах, які займаються вирощуванням рослин у теплицях.

Ключові слова: інформаційна система, розумна теплиця, агропромисловість, клімат, автоматизація, IoT, технічне обладнання, програмне забезпечення, Cisco Packet Tracer, Python.

ABSTRACT

An explanatory note to the bachelor's attestation work is created in the introduction, three sections, conclusions, a list of sources used and two additional substances. The total volume of work is 78 pages, of which 44 pages of the main part with 18 figures, 5 table, 11 names of the list of used sources and two additions.

The aim of the thesis is to develop an information system for automating microclimate management processes in greenhouses using IoT technologies.

Object of research – software and hardware of special-purpose computer systems.

The subject of research – an information system for automatic microclimate control in greenhouses that integrates IoT technologies to optimize growing conditions.

The problem that is solved in the diploma collaboration is to create a Smart Greenhouse information system that will automate greenhouse management processes, including irrigation, lighting, heating, ventilation and other systems. Existing systems are often not sufficiently integrated and require significant time to set up and maintain.

Scope – the automation of greenhouse management. The developed information system «Smart Greenhouse» can be widely used in the agricultural sector, in particular in farms and agro-industrial enterprises engaged in growing plants in greenhouses.

Keywords: information system, smart greenhouse, agricultural industry, climate, automation, IoT, technical equipment, software, Cisco Packet Tracer, Python.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП	7
РОЗДІЛ 1. ОГЛЯД АНАЛОГІВ СИСТЕМ ТИПУ «РОЗУМНА ТЕПЛИЦЯ».....	9
1.1 Суть концепції побудови систем типу «Розумна теплиця»	9
1.2 Компоненти для побудови систем «Розумна теплиця» та їх характеристики	9
1.3 Функції програмного забезпечення систем типу «Розумна теплиця» ..	10
1.4 Технічні характеристики систем типу «Розумна теплиця».....	11
1.5 Порівняльна характеристика існуючих систем і постановка задач проектування.....	12
Висновки за розділом 1	15
РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ СИСТЕМИ	17
2.1 Розробка структурної схеми	17
2.2 Розробка схеми теплиці.....	18
2.3 Вибір компонентів системи та технологій їх зв'язку.....	19
2.3.1 Вибір компонентів системи	19
2.3.2 Обґрунтування кількості обраних компонентів	22
2.3.3 Технологія зв'язку	30
2.3.4 Системна інтеграція та підключення сервера.....	32
Висновки за розділом 2	32
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	34
3.1. Розробка алгоритму функціонування інформаційної системи	34
3.2. Моделювання системи «Розумна теплиця»	36
3.2.1 Розробка основи для моделі.....	36
3.2.2 Налаштування мережі.....	40
3.2.3 Встановлення правил для контролерів	41
3.2.4 Моделювання мережі	45

3.3. Розробка програмного додатку управління системою	46
Висновки за розділом 3	49
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ.....	53

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

- UI – User Interface (користувацький інтерфейс);
- UX – User Experience (користувацький досвід);
- Wi-Fi – Wireless Fidelity;
- Zigbee – специфікація для набору високорівневих протоколів зв'язку з використанням малопотужних цифрових радіостанцій;
- Python – високорівнева мова програмування
- DHCP – Dynamic Host Configuration Protocol (протокол динамічної конфігурації хосту).

ВСТУП

У зв'язку зі зростанням інтересу до автоматизації процесів управління теплицями, особливо в аграрному секторі, стає актуальним розробка інформаційних систем, які б дозволяли ефективно контролювати мікроклімат та оптимізувати умови вирощування. Одним із напрямків цієї роботи є створення інформаційної системи для автоматизації управління мікрокліматом в теплицях за допомогою розумних технологій.

Актуальність роботи. Зі збільшенням чисельності населення планети та відповідним зростанням попиту на продукти харчування виникає нагальна потреба у стійких та ефективних методах ведення сільського господарства. Розумні теплиці, оснащені сучасними системами моніторингу та управління, пропонують перспективне рішення для підвищення продуктивності сільськогосподарських культур при мінімізації споживання ресурсів. Таким чином, розробка надійної інформаційної моделі для «розумних» теплиць має вирішальне значення для просування сільськогосподарських практик у напрямку сталого розвитку та ефективності. Вона має потенціал трансформувати спосіб виробництва продуктів харчування, забезпечуючи продовольчу безпеку.

Метою дослідження є розробка інформаційної системи, що використовує розумні технології для автоматизації процесів управління мікрокліматом в теплицях. Дана система спрямована на підвищення ефективності та оперативності управління умовами навколишнього середовища, критичними для росту та розвитку рослин.

Об'єкт дослідження – програмні та апаратні компоненти комп'ютерних систем спеціального призначення.

Методи дослідження: методи моделювання, методи аналізу і синтезу, методи комп'ютерного моделювання, методи автоматизованого проектування.

Предмет дослідження – інформаційна система, призначена для автоматичного контролю мікроклімату в теплицях.

Завдання дослідження

1. Розробити архітектуру інформаційної системи, яка визначає, як різні компоненти, такі як датчики, контролери і центральний блок управління, будуть взаємодіяти і спілкуватися один з одним.

2. Розробити алгоритм роботи, який визначатиме, як система реагує на дані про навколишнє середовище, зібрані датчиками, регулюючи температуру, вологість та умови освітлення, щоб підтримувати оптимальні умови для росту рослин.

3. Виконати моделювання роботи системи «Розумна теплиця».

4. Розробити програмне забезпечення для користувацького інтерфейсу для моніторингу та управління середовищем теплиці.

РОЗДІЛ 1

ОГЛЯД АНАЛОГІВ СИСТЕМ ТИПУ «РОЗУМНА ТЕПЛИЦЯ»

1.1 Суть концепції побудови систем типу «Розумна теплиця»

Концепція «Розумна теплиця» базується на впровадженні автоматизованих систем контролю та управління параметрами середовища, які є критичними для ефективного вирощування рослин. Основна мета подібних систем зводиться до забезпечення оптимальних умов для росту та розвитку рослин, включаючи контроль температури, вологості, освітлення, вентиляції та інших показників.

В основі концепції лежить інтеграція різних сенсорних технологій та автоматизованих механізмів управління. Це включає використання датчиків для вимірювання поточних умов у теплиці, а також пристроїв для регулювання цих умов. Датчики можуть вимірювати температуру, вологість, освітленість та інші параметри. Отримані дані обробляються контролерами, які приймають рішення про включення або вимкнення різних систем, таких як обігрівачі, зволожувачі, системи освітлення та вентиляції.

Використання «розумних» технологій дозволяє мінімізувати вплив людського фактору, знизити витрати на утримання теплиці та підвищити ефективність виробництва. Автоматизовані системи можуть працювати в режимі реального часу, забезпечуючи своєчасне реагування на зміни зовнішніх умов та внутрішнього середовища теплиці.

Таким чином, основна суть концепції побудови систем типу «розумна теплиця» полягає у створенні інтелектуальної системи управління, яка здатна самостійно підтримувати оптимальні умови для вирощування рослин, забезпечуючи їх здоровий ріст та високу врожайність.

1.2 Компоненти для побудови систем «Розумна теплиця» та їх характеристики

Розумна теплична система вимагає декількох ключових компонентів для ефективного моніторингу та контролю умов навколишнього середовища для

оптимального росту рослин. Ці компоненти включають датчики, приводи, систему управління, мережу зв'язку та інтерфейс користувача.

Датчики використовуються для вимірювання різних параметрів навколишнього середовища, таких як температура, вологість, інтенсивність світла, вологість ґрунту та рівень вуглекислого газу. Ці датчики надають системі керування дані в режимі реального часу, що дозволяє їй приймати обґрунтовані рішення про те, коли активувати виконавчі механізми.

Приводи - це пристрої, якими керує система для зміни умов навколишнього середовища в теплиці. Прикладами є нагрівачі, вентилятори, вентиляційні отвори та системи зрошення. Ці приводи допомагають підтримувати потрібну температуру, вологість і рівень вентиляції для росту рослин.

Система керування – відповідає за обробку даних з датчиків і надсилання команд на виконавчі механізми. Вона використовує алгоритми для аналізу умов навколишнього середовища та визначення оптимальних дій для підтримки бажаних умов вирощування.

Для з'єднання різних компонентів системи розумної теплиці необхідна комунікаційна мережа. Ця мережа дозволяє датчикам надсилати дані до системи керування, а системі керування – надсилати команди виконавчим механізмам.

Інтерфейс користувача необхідний для моніторингу та управління системою «розумної теплиці». Цей інтерфейс може бути простою інформаційною панеллю, що відображає дані датчиків в режимі реального часу і дозволяє користувачам змінювати налаштування, або ж він може бути більш просунутим, пропонуючи інструменти для аналізу даних і можливості віддаленого доступу.

1.3 Функції програмного забезпечення систем типу «Розумна теплиця»

Програмне забезпечення (ПЗ) систем типу «Розумна теплиця» є ключовим елементом, який забезпечує ефективне управління та моніторинг усіх аспектів функціонування теплиці. Основні функції ПЗ спрямовані на автоматизацію

процесів, аналіз даних та забезпечення користувачів необхідною інформацією для прийняття обґрунтованих рішень.

ПЗ системи «Розумна теплиця» забезпечує моніторинг параметрів середовища шляхом збору даних з сенсорів, розташованих у теплиці. Ці сенсори можуть вимірювати температуру повітря та ґрунту, вологість повітря та ґрунту, рівень освітленості, інші параметри. Отримані дані обробляються для виявлення трендів та аномалій у стані мікроклімату теплиці.

Функціонал ПЗ також включає автоматизацію процесів управління системами теплиці. ПЗ може автоматично регулювати роботу систем поливу, вентиляції, опалення та освітлення на основі заданих параметрів та показників сенсорів. Крім того, підтримка користувацьких сценаріїв дозволяє автоматизувати різні аспекти управління теплицею з урахуванням специфічних потреб користувача.

1.4 Технічні характеристики систем типу «Розумна теплиця»

Технічні характеристики систем типу «Розумна теплиця» включають в себе апаратне та програмне забезпечення, які забезпечують автоматизоване управління та моніторинг умов росту рослин у теплиці. Вони визначають функціональні можливості системи, її продуктивність та ефективність.

Апаратне забезпечення системи «Розумна теплиця» включає в себе сенсори, контролери, виконавчі пристрої та комунікаційне обладнання. Сенсори вимірюють параметри середовища, такі як температура, вологість, освітленість, рівень рН та інші. Контролери відповідають за збір даних від сенсорів, прийняття рішень на основі цих даних та керування виконавчими пристроями. Виконавчі пристрої, такі як насоси, вентилятори, клапани та освітлювальні пристрої, реалізують управління системами поливу, вентиляції, опалення та освітлення в теплиці. Комунікаційне обладнання забезпечує обмін даними між різними частинами системи та можливість віддаленого керування.

Програмне забезпечення системи «Розумна теплиця» включає в себе програмні модулі для збору, обробки, аналізу та відображення даних, автоматизації управління теплицею та алгоритми для аналізу та прогнозування

умов для росту рослин. Це ПЗ повинно бути надійним, ефективним та забезпечувати користувача необхідною інформацією для прийняття рішень.

1.5 Порівняльна характеристика існуючих систем і постановка задач проектування

Priva Connected відома своїми розширеними можливостями клімат-контролю, пропонуючи точне регулювання температури, вологості та рівня CO₂. Вона інтегрується з різними датчиками і виконавчими механізмами, забезпечуючи моніторинг і управління в режимі реального часу за допомогою зручного інтерфейсу.

Argus Controls підкреслює масштабованість і можливість кастомізації, що робить її придатною для малих і великомасштабних операцій. Її модульна конструкція дозволяє інтегрувати з пристроями сторонніх виробників, пропонуючи гнучкість у розширенні системи та адаптації до різних розмірів і планувань теплиць.

Certhon виділяється своєю увагою до енергоефективності, використовуючи інноваційні технології, такі як теплові насоси та енергетичні завіси для зменшення споживання енергії. Його системи розроблені таким чином, щоб максимізувати врожайність сільськогосподарських культур, зводячи до мінімуму вплив на довкілля.

Rough Brothers, Inc. (Rough Brothers) наголошує на довговічності та надійності, задовольняючи потреби комерційних виробників. Її системи міцні і здатні витримувати жорсткі умови довкілля, забезпечуючи безперебійну роботу.

Цілі проектування систем розумних теплиць повинні враховувати ці характеристики, щоб досягти оптимального росту рослин, ресурсоефективності та експлуатаційної надійності.

Основні цілі включають в себе:

- Оптимізований клімат-контроль: Системи повинні забезпечувати точне регулювання температури, вологості та рівня CO₂, щоб створити ідеальне середовище для росту рослин.

- Масштабованість та кастомізація: Системи повинні бути модульними та адаптивними, що дозволяє легко розширювати та налаштовувати їх відповідно до потреб теплиць різного розміру та планування.
- Енергоефективність: Системи повинні включати енергоефективні технології для мінімізації споживання енергії та зменшення впливу на навколишнє середовище.
- Надійність і довговічність: Системи повинні бути міцними та надійними, здатними витримувати суворі умови навколишнього середовища та забезпечувати безперебійну роботу.
- Інтеграція та сумісність: Системи повинні легко інтегруватися з різноманітними датчиками, виконавчими механізмами та пристроями сторонніх виробників, забезпечуючи комплексні можливості контролю та моніторингу.
- Зручний інтерфейс: Системи повинні мати інтуїтивно зрозумілий і зручний інтерфейс, що полегшує фермерам моніторинг і контроль умов у теплицях.
- Аналіз даних і підтримка прийняття рішень: Системи повинні надавати інструменти для аналізу даних і підтримки прийняття рішень, допомагаючи фермерам оптимізувати свої стратегії вирощування та максимізувати врожайність.

Порівняльна характеристика наведена у табл. 1.1 та табл. 1.2. У цій таблиці наведено порівняльний огляд ключових характеристик систем розумних теплиць Priva Connected, Argus Controls, Certhon і Rough Brothers, Inc. (Rough Brothers). Кожна система має свої сильні сторони і фокуси, задовольняючи різні потреби і переваги в автоматизації та управлінні теплицями.

Таблиця 1.1

Порівняльна характеристика системи

Характеристика	Priva Connected	Argus Controls	Certhon	Rough Brothers, Inc. (Rough Brothers)
Клімат-контроль	Розширений контроль температури, вологості, CO2	Масштабований контроль, що налаштовується	Фокус на енергоефективному клімат-контролі з використанням інноваційних технологій	Надійна система, призначена для роботи в суворих умовах навколишнього середовища
Масштабованість	Обмежена масштабованість	Високомасштабована, модульна конструкція	Масштабована, адаптація до різних розмірів	Масштабована
Енергоефективність	Стандартні функції енергоефективності	Орієнтація на енергоефективні технології	Інноваційні енергозберігаючі технології	Акцент на довговічність, а не на енергоефективність
Надійність	Висока надійність	Надійна	Надійна	Дуже надійна
Інтеграція	Інтегрується з різними датчиками, виконавчими механізмами	Модульна, інтегрується зі сторонніми пристроями	Сумісність з різними пристроями	Інтегрується з різними пристроями
Інтерфейс користувача	Простий та зручний	Простий та зручний	Простий та зручний	Простий та зручний

Продовження у табл. 1.2.

Таблиця 1.2

Порівняльна характеристика системи

Характеристика	Priva Connected	Argus Controls	Certhon	Rough Brothers, Inc. (Rough Brothers)
Аналіз даних	Надає інструменти для аналізу даних	Підтримує аналіз даних і підтримку прийняття рішень	Обмеженість інструментів аналізу даних	Обмежені можливості аналізу даних

Висновки за розділом 1

Вивчення аналогічних систем «Розумної теплиці» виявляє фундаментальну концепцію, що базується на інтеграції автоматизованих систем моніторингу та управління для регулювання критичних параметрів навколишнього середовища, які мають вирішальне значення для ефективного вирощування рослин. Ці системи спрямовані на створення оптимальних умов для росту та розвитку рослин шляхом управління температурою, вологістю, освітленням, рівнем CO₂, вентиляцією та іншими ключовими показниками. Центральним елементом цієї концепції є поєднання різноманітних сенсорних технологій та автоматизованих механізмів управління. Ці датчики використовуються для вимірювання умов, що переважають у теплиці, тоді як механізми управління відповідно коригують фактори навколишнього середовища. Дані, зібрані з датчиків, обробляються контролерами, які потім приймають рішення щодо активації або деактивації різних систем у теплиці, таких як обігрівачі, зволожувачі, системи освітлення та вентиляції. Використання розумних технологій дозволяє мінімізувати вплив людського втручання, тим самим знижуючи операційні витрати і підвищуючи ефективність виробництва. Працюючи в режимі реального часу, ці автоматизовані системи можуть оперативно реагувати на коливання зовнішніх умов і підтримувати оптимальне внутрішнє середовище теплиці. Отже, основною метою побудови

розумних тепличних систем є створення інтелектуальної системи управління, здатної автономно підтримувати оптимальні умови для росту рослин, забезпечуючи тим самим їх здоровий розвиток і високу врожайність.

РОЗДІЛ 2

РОЗРОБКА АРХІТЕКТУРИ СИСТЕМИ

2.1 Розробка структурної схеми

При розробці системи «Розумної теплиці» застосовано інтегрований та структурований підхід. Такий підхід забезпечує ефективну організацію функціональних блоків та їхню взаємодію в середовищі теплиці. Структурна схема відображає взаємозв'язки між центральною системою управління, датчиками, виконавчими механізмами та іншими підключеними пристроями. Вона також ілюструє інтеграцію різних технологічних рішень, спрямованих на оптимізацію роботи теплиці, таких як клімат-контроль, зрошення та освітлення тощо.

Структурну схему системи подано на рис. 2.1.

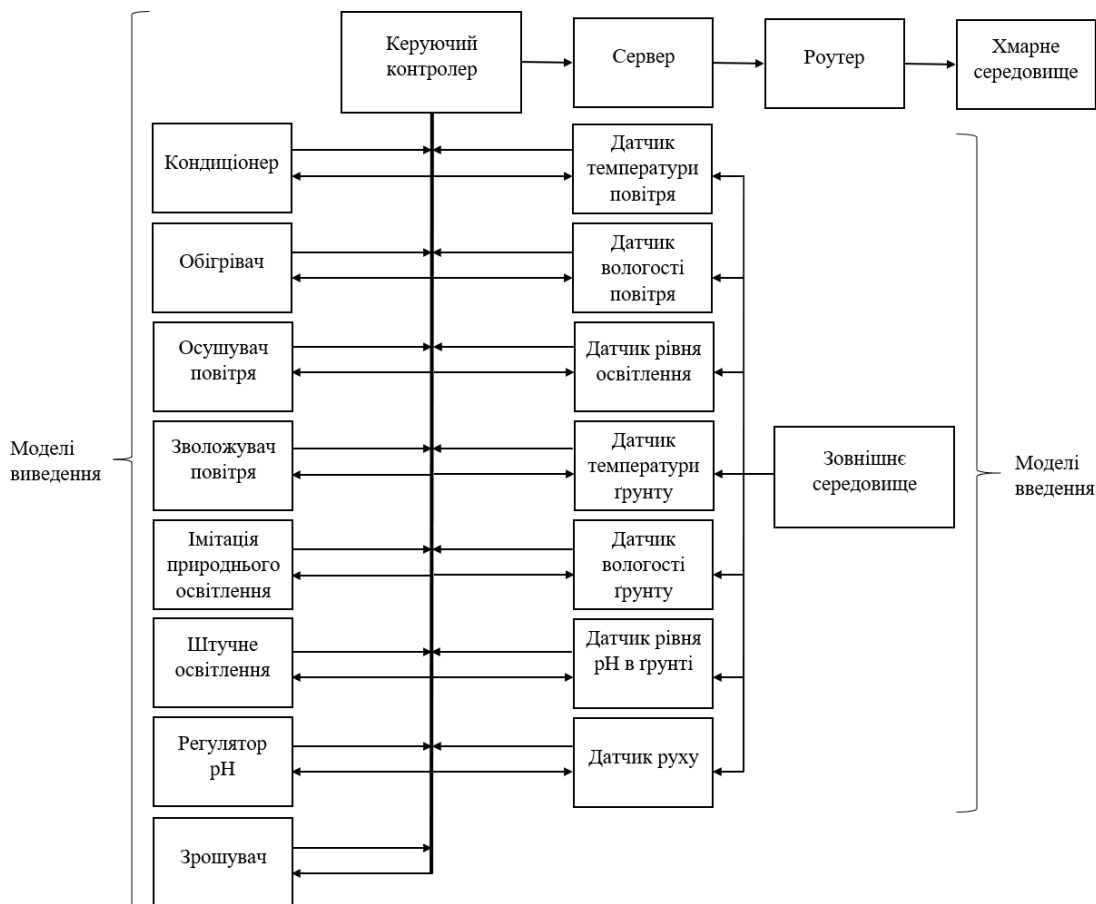


Рисунок 2.1 – Структурна схема системи

2.2 Розробка схеми теплиці

Дизайн і планування теплиці є фундаментальними аспектами, які безпосередньо впливають на ефективність і результативність системи.

Схему приміщення теплиці подано на рис. 2.1.

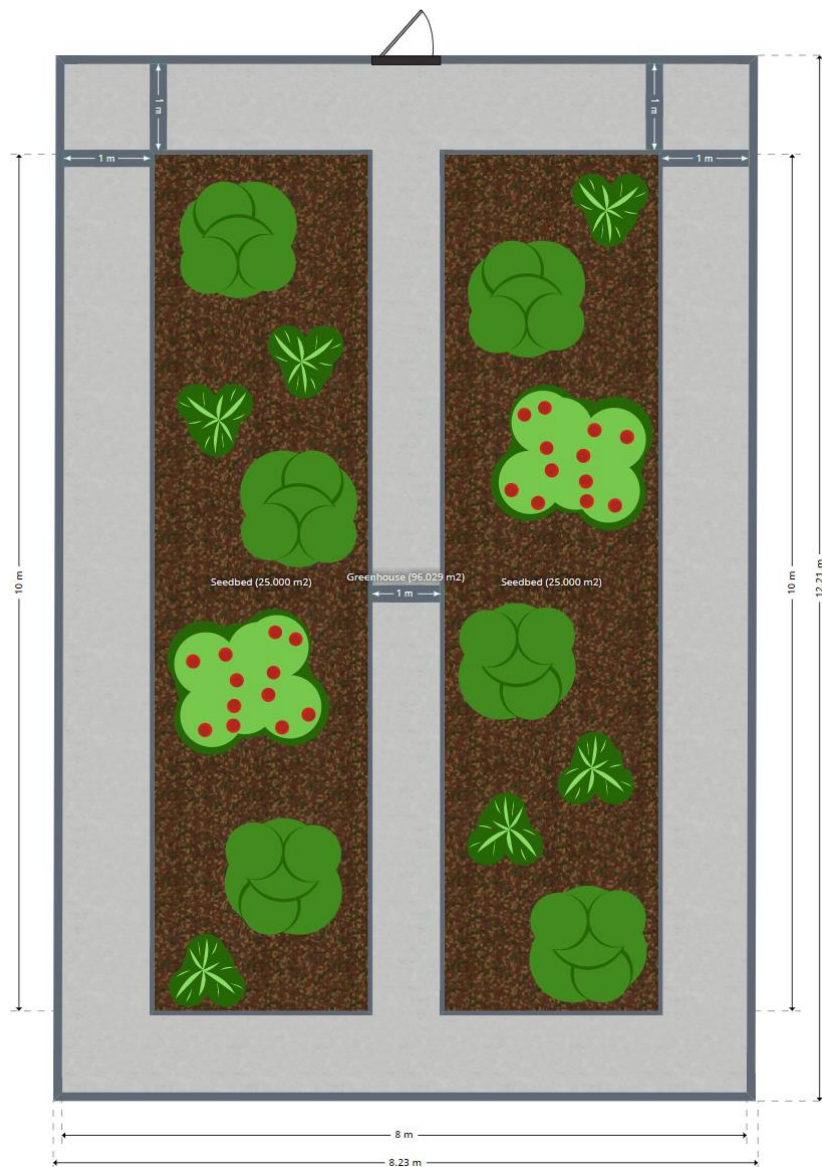


Рисунок 2.1 – Схема приміщення теплиці

Внутрішній простір теплиці організовано таким чином, щоб розмістити компоненти системи розумної теплиці та оптимізувати використання простору для вирощування рослин. Теплиця із загальною внутрішньою площею 10 x 8

метрів розділена на дві основні грядки, кожна розміром 2.5 x 8 метрів. Грядки розташовані паралельно одна одній, що забезпечить ефективне переміщення та доступ до рослин.

Компоненти системи розумної теплиці стратегічно інтегровані в конструкцію теплиці для забезпечення їх функціональності та доступності. Система включає датчики для моніторингу температури та рівня вологості ґрунту, приводи для управління системами вентиляції та зрошення, а також центральний блок управління для обробки даних та управління середовищем теплиці.

Датчики розміщені в ґрунті на кожній грядці для постійного моніторингу температури та рівня вологості, надаючи дані в режимі реального часу на центральний блок управління. Приводи встановлені для регулювання систем вентиляції та зрошення на основі даних, отриманих від датчиків, забезпечуючи оптимальні умови для вирощування рослин.

2.3 Вибір компонентів системи та технологій їх зв'язку

2.3.1 Вибір компонентів системи

Технічні характеристики кожного компонента:

- Samsung SmartThings Hub 3-го покоління

Особливості: Цей контролер підтримує пристрої Zigbee, Z-Wave та Wi-Fi. Він дозволяє здійснювати віддалений моніторинг і керування підключеними пристроями та інтегрується з додатком для автоматизації та сповіщень. Сумісний з широким спектром розумних продуктів.

- Dell XPS 13

Технічні характеристики: Dell XPS 13 оснащений процесором Intel Core i7-10710U, 16 ГБ оперативної пам'яті LPDDR3 і твердотільним накопичувачем 512 ГБ. Він має 13,3-дюймовий сенсорний дисплей InfinityEdge з роздільною здатністю 4K UHD (3840 x 2160) і працює під управлінням Windows 10 Home. Порти включають 2 x Thunderbolt 3, 1 x USB-C 3.1 і кардрідер microSD.

- TP-Link Archer C7 AC1750

Технічні характеристики: Двodiaпазонний маршрутизатор з підтримкою технології Wi-Fi 5 (802.11ac). Він забезпечує швидкість до 450 Мбіт/с у діапазоні 2,4 ГГц і 1300 Мбіт/с у діапазоні 5 ГГц, що в сумі становить 1750 Мбіт/с. Оснащений чотирма портами Gigabit Ethernet для дротового підключення, а також портами USB для спільного доступу до файлів і принтерів. Маршрутизатор підтримує гостьовий доступ до мережі та включає шифрування WPA/WPA2 для безпеки.

- Synology DiskStation DS920+

Особливості DS920+ працює на базі процесора Intel Celeron J4125 з 4 ГБ пам'яті DDR4 з можливістю розширення до 8 ГБ. Він має 4 відсіки для 3,5»/2,5» SATA HDD/SSD і можливості підключення, включаючи 2 порти 1GbE LAN, 2 порти USB 3.0 і 1 порт eSATA. Підтримує RAID 0, 1, 5, 6, 10 і JBOD.

- 3-Series Temp & Humidity Sensor

Технічні характеристики: Цей датчик вимірює температуру від -10°C до 50°C (14°F до 122°F) і рівень вологості від 0% до 100% RH. Він працює бездротово через Zigbee і має термін служби батареї до 2 років.

- Датчик денного світла EC-Link Zigbee

Особливості Цей датчик вимірює рівень освітленості і працює бездротово через Zigbee. Він живиться від батареї з тривалим терміном служби, його легко встановити та інтегрувати з системами розумної теплиці.

- Soil Temperature Humidity Tester

Технічні характеристики Цей тестер вимірює температуру від -40°C до 85°C (від -40°F до 185°F) і рівень вологості від 0% до 100% RH. Працює від батарейок.

- Датчик руху в приміщенні Philips Hue

Технічні характеристики: Цей датчик працює бездротово через Zigbee і має дальність виявлення до 5 метрів (16,4 футів). Він працює від батареї

з тривалим терміном служби та інтегрується з системою освітлення Philips Hue для автоматизації.

- Розумний портативний кондиціонер LG LP1419IVSM
Особливості Цей кондиціонер має холодопродуктивність 14 000 BTU і використовує додаток для дистанційного керування. Він оснащений технологією Dual Inverter для енергоефективності та працює з рівнем шуму 44 дБ.
- Обігрівач для теплиць Dr. Heater
Технічні характеристики: Цей обігрівач має потужність 1500 Вт і оснащений термостатом для підтримання бажаної температури. Він сконструйований таким чином, щоб бути довговічним в тепличних умовах і включає в себе такі функції безпеки, як захист від перегріву і перекидний вимикач.
- Електричний міні-осушувач Pro Breeze
Особливості Цей осушувач має продуктивність 500 мл/добу при 86°F і 80% відносній вологості, з ємністю бака 1,5 л. Він використовує надтиху технологію Пельтьє і автоматично вимикається після заповнення.
- Зволожувач повітря LEVOIT Smart WiFi
Технічні характеристики: Цей зволожувач має ємність 6 л і покриває площу до 753 кв. м. Він пропонує як теплий, так і холодний туман і може управлятися за допомогою програми VeSync.
- Світлодіодна смарт-лампочка Hue White та Color Ambiance
Особливості Ця лампа працює бездротово через Zigbee і дозволяє регулювати колірну температуру та яскравість. Вона інтегрується з системою Philips Hue для автоматизації та має термін служби до 25 000 годин.
- Світлодіодна лампа для вирощування Aspect

Технічні характеристики: Ця лампа для вирощування рослин має потужність 40 Вт і забезпечує світлодіодне світло повного спектру для росту рослин. Він покриває площу до 6 кв. футів і має термін служби 15 років за умови використання 16 годин на добу.

- Контролер рН Bluelab Connect

Особливості: Цей контролер забезпечує моніторинг і регулювання рН в режимі реального часу з бездротовим підключенням для віддаленого моніторингу. Він включає функцію дозування для автоматичного контролю рН і підходить для гідропонних систем.

- Комбінований набір VIVOSUN Seedling Heat Mat та цифровий термостат

Технічні характеристики: Розмір термокилимка 48» x 20,75» з температурним діапазоном від 68°F до 108°F (від 20°C до 42°C). Він має цифровий терморегулятор для точного контролю, а також водонепроникну та міцну конструкцію.

- Розумна система поливу Rain Bird ST8I-WIFI

Особливості Ця система контролює до 8 зон поливу та пропонує розумне планування на основі погодних даних. Системою можна дистанційно керувати за допомогою додатку Rain Bird, а також вона має функції економії води для оптимізації ефективності поливу.

2.3.2 Обґрунтування кількості обраних компонентів

Компонент: Samsung SmartThings Hub 3rd Generation.

Кількість: 1.

Обґрунтування: Контролер слугує центральним блоком управління, керуючи всіма підключеними пристроями. Одного пристрою достатньо, оскільки він може забезпечити зв'язок і керування всією системою теплиці.

Компонент: Dell XPS 13.

Кількість: 1.

Обґрунтування: Dell XPS 13 забезпечує обчислювальну потужність для обробки даних та управління системою. Одного пристрою достатньо, виходячи

з розміру та складності теплиці, оскільки він може впоратися з необхідним обчислювальним навантаженням.

Компонент: TP-Link Archer C7.

Кількість: 1.

Обґрунтування: Wi-Fi роутер забезпечує підключення до Інтернету для розумної теплиці. Одного маршрутизатора достатньо, щоб забезпечити належне покриття та зв'язок для всієї теплиці.

Компонент: Synology DiskStation.

Кількість: 1.

Обґрунтування: Synology DiskStation виконує роль сервера Інтернету речей, керуючи зберіганням і обробкою даних. Потужності одного пристрою достатньо для обробки даних, що генеруються датчиками та пристроями теплиці.

Компонент: 3-Series Temp & Humidity Sensor.

Кількість: 4.

Обґрунтування: Точний діапазон чутливості може змінюватися, але для надійних вимірювань він зазвичай становить близько 4-6 метрів. Теплиця розміром 12x8 метрів і має дві грядки, кожна з яких 10x2.5 метрів. Діапазону одного датчика достатньо для ширини грядки, але недостатньо для довжини.

Розрахунок кількості датчиків для покриття ширини однієї грядки:

$$N=10/((4+6)/2)=2 \quad (2.1)$$

Результат розрахунку зображує, що для однієї грядки необхідно два датчика, а оскільки в теплиці розташовано дві грядки, то необхідно чотири датчики, щоб забезпечити достатнє покриття і моніторинг рівня температури і вологості повітря.

Компонент: EC-Link Zigbee Daylight Sensor PZD 510P.

Кількість: 4.

Обґрунтування: Точний діапазон чутливості може змінюватися, але для надійних вимірювань він зазвичай становить близько 5-7 метрів. Теплиця розміром 12x8 метрів і має дві грядки, кожна з яких 10x2.5 метрів. Діапазону одного датчика достатньо для ширини грядки, але недостатньо для довжини.

Розрахунок кількості датчиків для покриття ширини однієї грядки:

$$N = \text{Довжина} / \text{Середній діапазон} = 10 / ((5+7)/2) = 1.67 \quad (2.2)$$

Оскільки не можливо мати дробову частину числа пристроїв, виконується округлення до більшого цілого числа. Результат розрахунку зображує, що для однієї грядки необхідно два датчика, а оскільки в теплиці розташовано дві грядки, то необхідно чотири датчики, щоб забезпечити достатнє покриття і моніторинг рівня освітлення теплиці.

Компонент: Soil Temperature Humidity Tester Soil Tester Temperature and Humidity Sensor.

Кількість: 4.

Обґрунтування: Точний діапазон чутливості може змінюватися, але для надійних вимірювань він зазвичай становить близько 5-6 метрів. Теплиця розміром 12x8 метрів і має дві грядки, кожна з яких 10x2.5 метрів. Діапазону одного датчика достатньо для ширини грядки, але недостатньо для довжини.

Розрахунок кількості датчиків для покриття ширини однієї грядки:

$$N = \text{Довжина} / \text{Середній діапазон} = 10 / ((5+6)/2) = 1.82 \quad (2.3)$$

Оскільки не можливо мати дробову частину числа пристроїв, виконується округлення до більшого цілого числа. Результат розрахунку зображує, що для однієї грядки необхідно два датчика, а оскільки в теплиці розташовано дві грядки, то необхідно чотири датчики, щоб забезпечити достатнє покриття і моніторинг рівня температури та вологості ґрунту.

Компонент: Philips Hue Indoor Motion Sensor.

Кількість: 1.

Обґрунтування: Одного датчика руху на вході достатньо, щоб виявити рух у теплиці та запустити дії, що залежать від присутності людей, наприклад, регулювання освітлення.

Компонент: LG LP1419IVSM Smart Portable Air Conditioner.

Кількість: 2.

Обґрунтування: Для регулювання температури в теплиці потрібні два кондиціонери, що забезпечують достатню потужність охолодження для обох грядок.

Компонент: Dr. Heater Greenhouse Heater.

Кількість: 2.

Обґрунтування: Два обігрівачі необхідні для підтримки оптимального рівня температури для обох грядок.

Компонент: Pro Breeze Electric Mini Dehumidifier.

Кількість: 4.

Обґрунтування: Електричний міні-осушувач призначений зазвичай покриває площу до 270 квадратних футів (приблизно 25 квадратних метрів). Враховуючи розмір теплиці 12x8 метрів (96 квадратних метрів), цей діапазон означає, що для належного покриття всього простору необхідно кілька пристроїв.

Розрахунок кількості пристроїв для покриття теплиці:

$$N = \text{Площа теплиці} / \text{Діапазон дії} = 96 / 25 = 3.84 \quad (2.4)$$

Оскільки не можливо мати дробову частину числа пристроїв, виконується округлення до більшого цілого числа. Результат розрахунку зображує, що для покриття теплиці необхідно 4 пристрої.

Компонент: LEVOIT Smart WiFi Humidifier.

Кількість: 2.

Обґрунтування: Зволожувач повітря зазвичай має ефективну площу покриття до 753 квадратних футів (приблизно 70 квадратних метрів). Це покриття може дещо змінюватися залежно від таких факторів, як планування приміщення, вентиляція та специфічні потреби у вологості навколишнього середовища. Враховуючи розмір теплиці (близько 80 квадратних метрів), цей діапазон означає, що для належного покриття всього простору необхідно два пристрої.

Компонент: Hue White and Color Ambiance LED Smart Bulb.

Кількість: 6.

Обґрунтування: Діапазон освітлення ламп близько 2-2.5 метра. Лампи розміщені між грядками для зручності їх огляду за умов низького освітлення.

Розрахунок кількості пристроїв для покриття теплиці:

$$N = \text{Довжина теплиці} / \text{Середній діапазон} = 12 / ((2 + 2.5) / 2) = 5.3 \quad (2.5)$$

Оскільки не можливо мати дробову частину числа пристроїв, виконується округлення до більшого цілого числа. Результат розрахунку зображує, що для покриття довжини проходу необхідно шість лампочок.

Компонент: Aspect LED Growlight Aspect-40W.

Кількість: 8.

Обґрунтування: Діапазон освітлення ламп близько 2.5 метрів. Лампи розміщені над грядками для імітації сонячного освітлення. Цього достатньо для ширини грядки, але недостатньо для довжини.

Розрахунок кількості пристроїв для покриття довжини грядки:

$$N = \text{Довжина грядки} / \text{Діапазон дії} = 10 / 2.5 = 4 \quad (2.6)$$

Результат розрахунку зображує, що для покриття довжини грядки необхідно чотири лампочки. Оскільки в теплиці розташовано дві грядки, то загалом необхідно вісім лампочок.

Компонент: Bluelab pH Controller Connect CONTPH.

Кількість: 10.

Обґрунтування: Ефективна площа, яку охоплює контролер pH Bluelab Connect CONTPH, зазвичай залежить від конкретної установки і способу розміщення датчиків pH в теплиці. Однак, як правило, такі контролери призначені для управління рівнем pH для певного об'єму розчину або певної ділянки в гідропонній або ґрунтовій системі вирощування.

Кожен контролер pH Bluelab pH Controller Connect здатний ефективно управляти рівнем pH на площі, еквівалентній стандартному гідропонному резервуару або аналогічній за розміром ділянці ґрунту. Зазвичай один контролер pH може ефективно покривати площу близько п'яти квадратних метрів. Враховуючи розмір грядок (25 квадратних метрів кожна), цей діапазон означає, що для належного покриття всього простору необхідно декілька пристроїв.

Розрахунок кількості пристроїв для покриття площі грядки:

$$N = \text{Площа грядки} / \text{Діапазон дії} = 20 / 5 = 5 \quad (2.7)$$

Результат розрахунку зображує, що для покриття площі грядки необхідно п'ять пристроїв. Оскільки в теплиці розташовано дві грядки, то загалом необхідно 10 контролерів.

Компонент: VIVOSUN Seedling Heat Mat and Digital Thermostat Combo Set.

Кількість: 36.

Обґрунтування: Розмір одного теплового килима становить 48x20.75 дюймів (близько 122x52.70 сантиметрів). Враховуючи розмір грядок (25 квадратних метрів кожна), цей діапазон означає, що для належного покриття всього простору необхідно декілька пристроїв.

Необхідними параметрами при розрахунках є довжина грядки $B_l = 10$ метрів, ширина грядки $B_w = 10$ метрів. Також важливими є довжина одного теплового килима $M_l = 1.219$ метра та ширина $M_w = 0.527$ метра. Важливо враховувати можливу відстань між термокилимками. По довжині: $M_{sl} = 0.4$ метра та по ширині $M_{sw} = 0.25$ метра. Також необхідно враховувати відступи від країв грядки. По довжині $B_{sl} = 0.15$ метра та по ширині $B_{sw} = 0.1$ метра.

Розрахунок кількості для покриття довжини грядки (L):

$$L = (B_l - B_{sl} * 2) / (M_l + M_{sl}) = (10 - 2 * 0.15) / (1.219 + 0.4) = 5.99 = 6 \quad (2.8)$$

Розрахунок кількості для покриття ширини грядки (W):

$$W = (B_w - B_{sw} * 2) / (M_w + M_{sw}) = (2.5 - 2 * 0.1) / (0.527 + 0.25) = 2.96 = 3 \quad (2.9)$$

Розрахунок кількості для покриття грядки (N):

$$N = L * W = 18 \quad (2.10)$$

Результат розрахунку зображує, що для покриття площі грядки необхідно вісімнадцять пристроїв. Оскільки в теплиці розташовано дві грядки, то загалом необхідно 36 термокиликів.

Компонент: Rain Bird ST8I-WIFI Smart Irrigation Sprinkler System.

Кількість: 1.

Обґрунтування: Теплиця має всього чотири зони поливу в той час як система поливу може одночасно контролювати до восьми таких зон, тому одного пристрою достатньо для підвищення рівня вологості ґрунту на обох грядках.

Компоненти для побудови системи наведені у табл. 2.1, табл. 2.2 та табл. 2.3.

Таблиця 2.1

Компоненти для побудови системи

Назва	Модель	Виробник	Функція	Тип зв'язку в системі	Ціна	Кількість
Samsung SmartThings Hub 3rd Generation	GP-U999SJV LGDA	Samsung	Центральний блок управління для керування всіма підключеними пристроями	Wi-Fi, Zigbee	\$69.99	1
Dell XPS 13	XPS7390-7681SLV-PUS	Dell	Керування системою за допомогою розробленого інтерфейсу	Wi-Fi	\$1199.99	1
TP-Link Archer C7 AC1750	AC1750	TP-Link	Доступ до Інтернету	Wi-Fi	\$60.99	1
Synology DiskStation DS920+	DS920+	Synology	Сервер IoT	Wi-Fi	\$549.99	1
3-Series Temp & Humidity Sensor	3310-G	Centralite	Датчик температури та вологості повітря	Zigbee	\$29.95	4
EC-Link Zigbee Daylight Sensor	PZD 510P	Pooko LED Light Company	Датчик світла	Zigbee	\$29.99	4

Продовження у табл. 2.2.

Таблиця 2.2

Компоненти для побудови системи

Назва	Модель	Виробник	Функція	Тип зв'язку в системі	Ціна	Кількість
Soil Temperature Humidity Tester	Soil Tester Temperature and Humidity Sensor	ІНОМЕ САМ	Датчик температури та вологості ґрунту	Zigbee	\$17.00	4
Philips Hue Indoor Motion Sensor	046677570972	Philips	Датчик руху	Zigbee	\$59.99	1
LG LP1419IVSM Smart Portable Air Conditioner	LP1419IVSM	LG	Зниження температури повітря, фільтрація повітря	Zigbee	\$499.99	2
Dr. Heater Greenhouse Heater	DR218-1500W	Dr. Heater	Підвищення температури в теплиці	Wi-Fi	\$149.99	2
Pro Breeze Electric Mini Dehumidifier	PB-02-US	Pro Breeze	Зниження вологості повітря	Wi-Fi	\$49.99	4
LEVOIT Smart WiFi Humidifier	LV600HN	LEVOIT	Підвищення вологості повітря	Wi-Fi	\$89.99	2
Hue White and Color Ambiance LED Smart Bulb	046677548590	Philips	Штучне освітлення	Zigbee	\$49.99	6
Aspect LED Growlight	Aspect-40W	Soltech	Імітація природнього освітлення	Wi-Fi	\$150.00	8
Bluelab pH Controller Connect	CONTPH	Bluelab	Аналіз та підвищення вмісту рН ґрунту	Wi-Fi	\$349.99	10

Продовження у табл. 2.3.

Таблиця 2.3

Компоненти для побудови системи

Назва	Модель	Виробник	Функція	Тип зв'язку в системі	Ціна	Кількість
VIVOSUN Seedling Heat Mat and Digital Thermostat Combo Set	4316276172	VIVOSUN	Регулювання температури ґрунту	Wi-Fi	\$59.99	36
Rain Bird ST8I-WIFI Smart Irrigation Sprinkler System	ST8I-WIFI	Rain Bird	Підвищення вологості ґрунту	Zigbee	\$159.99	1

Загальні витрати на компоненти: \$11428.06.

2.3.3 Технологія зв'язку

Ефективне функціонування інформаційної системи «Розумна теплиця» значною мірою залежить від надійних комунікаційних технологій, які забезпечують безперервний обмін даними між різними компонентами в тепличному середовищі. Вибір комунікаційних технологій має вирішальне значення, оскільки він безпосередньо впливає на загальну продуктивність, надійність і масштабованість системи.

При розробці інформаційної системи «Розумна теплиця» вибір технологій зв'язку відіграє вирішальну роль у забезпеченні ефективної передачі даних та управлінні підключеними пристроями. Для цього проекту в якості основних технологій зв'язку обрані ZigBee і Wi-Fi завдяки їх специфічним перевагам і сумісності з вимогами системи.

ZigBee - це технологія бездротового зв'язку з малим рівнем енергоспоживання та швидкістю передачі даних, спеціально розроблена для віддаленого моніторингу та управління. Вона працює за стандартом IEEE 802.15.4, який забезпечує надійний і безпечний зв'язок на невеликих відстанях.

ZigBee пропонує кілька переваг:

- Пристрої ZigBee споживають мінімальну кількість енергії, що робить їх ідеальними для датчиків і пристроїв, що працюють від батарейок, у тепличному середовищі, де необхідний безперервний моніторинг.
- ZigBee підтримує комірчасті мережі, дозволяючи пристроям взаємодіяти один з одним через проміжні вузли. Це покращує покриття мережі та надійність, що має вирішальне значення в теплицях, де сигнали можуть бути заблоковані металевими конструкціями або густою рослинністю.
- Протокол зв'язку ZigBee забезпечує надійну передачу даних навіть у шумному середовищі, яке часто зустрічається в сільському господарстві.
- Пристрої ZigBee від різних виробників можуть безперешкодно взаємодіяти, забезпечуючи гнучкість у виборі та інтеграції пристроїв.
- ZigBee пропонує вбудовані функції безпеки, такі як шифрування і аутентифікація, що забезпечують конфіденційність і цілісність даних.

Wi-Fi, або IEEE 802.11, - це широко використовувана технологія бездротового зв'язку, відома своєю високою швидкістю передачі даних і широкою доступністю.

Wi-Fi має кілька переваг:

Wi-Fi забезпечує високошвидкісну передачу даних, що підходить для потокової передачі відео з камер спостереження або передачі великих наборів даних з датчиків навколишнього середовища.

Мережі Wi-Fi можуть покривати великі території, забезпечуючи зв'язок по всьому тепличному комплексу.

Wi-Fi є стандартною функцією багатьох споживчих пристроїв, що дозволяє легко інтегруватися з існуючою інфраструктурою та пристроями.

Wi-Fi - це добре відома і широко використовувана технологія, що полегшує користувачам налаштування і управління системою.

Мережі Wi-Fi можна легко масштабувати для підключення додаткових пристроїв або розширення теплиці без значних змін в інфраструктурі.

Разом ці технології забезпечують надійну та ефективну комунікаційну основу для управління та контролю розумного тепличного середовища.

2.3.4 Системна інтеграція та підключення сервера

Спеціалізований контролер Samsung SmartThings Hub слугує центральним вузлом, з'єднуючи мережі ZigBee та WiFi і забезпечуючи безперебійну інтеграцію всіх пристроїв у теплиці. Контролер зв'язується з IoT-сервером, який зберігає дані з різних датчиків і пристроїв, забезпечуючи комплексне уявлення про середовище теплиці. Також сервер впроваджує заздалегідь визначені правила та графіки для автоматизації таких процесів, як полив, освітлення, опалення та вентиляція. Окрім цього він дозволяє користувачам здійснювати моніторинг і контроль тепличного середовища за допомогою програмного додатку. Доступ до сервера здійснюється через Dell XPS 13, який слугує локальним інтерфейсом, забезпечуючи моніторинг, управління та візуалізацію даних у режимі реального часу.

Висновки за розділом 2:

Дослідження з розробки інформаційної системи для «розумної теплиці» показує, що розробка виконується із застосуванням структурованого підходу, що забезпечує ефективну організацію функціональних блоків та їхню взаємодію в тепличному середовищі. Система інтегрує різні технологічні рішення, спрямовані на оптимізацію роботи теплиці, як-от контроль клімату, зрошення та освітлення. Архітектура системи включає центральну систему управління, датчики, виконавчі пристрої та інші підключені пристрої. Організація внутрішнього простору теплиці оптимізована для розміщення компонентів системи та оптимального використання простору для вирощування рослин. Для забезпечення безперебійного обміну даними між різними компонентами системи обрано надійні технології зв'язку, такі як Zigbee і Wi-Fi. Zigbee забезпечує низьке енергоспоживання і підтримку мереж з маршрутизацією, що важливо для забезпечення зв'язку в тепличному середовищі.

Таким чином, розроблена інформаційна система для «розумної теплиці» є комплексним рішенням, що забезпечує ефективне і стійке управління тепличним середовищем для оптимального росту рослин.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1. Розробка алгоритму функціонування інформаційної системи

Розробка алгоритму функціонування інформаційної системи «Розумна теплиця» є критично важливим аспектом забезпечення ефективної та результативної роботи всієї системи. Алгоритм призначений для управління та контролю різних компонентів розумної теплиці, включаючи клімат-контроль, системи зрошення, освітлення та моніторинг параметрів навколишнього середовища.

Алгоритм подано на рис. 3.1.

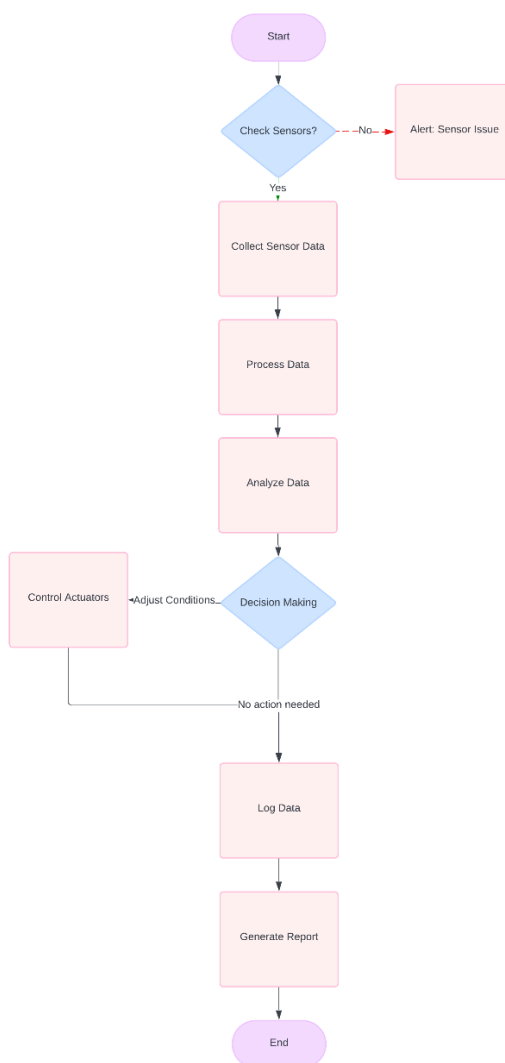


Рисунок 3.1 – Алгоритм функціонування інформаційної системи

Алгоритм дотримується набору заздалегідь визначених правил та інструкцій для регулювання середовища в теплиці на основі даних, зібраних з датчиків у реальному часі. Він враховує такі фактори, як температура, вологість, рівень освітленості та вологість ґрунту, щоб приймати рішення щодо роботи приводів і пристроїв у теплиці.

Розроблений алгоритм є адаптивним, що дозволяє йому підлаштовувати параметри відповідно до мінливих умов навколишнього середовища та потреб рослин. Наприклад, якщо температура всередині теплиці перевищує певний поріг, алгоритм може активувати систему вентиляції для зниження температури. Аналогічно, якщо рівень вологості ґрунту нижче певного порогу, алгоритм може активувати систему зрошення для поливу рослин. Це допомагає оптимізувати використання ресурсів і гарантувати, що рослини отримують необхідний догляд у потрібний момент.

Існує також можливість змінити порогові значення за допомогою програмного продукту. Після того як оператор теплиці здасть та збереже нові певні порогові значення, вони відправляються до системи теплиці.

Схему передачі даних подано на рис. 3.2.

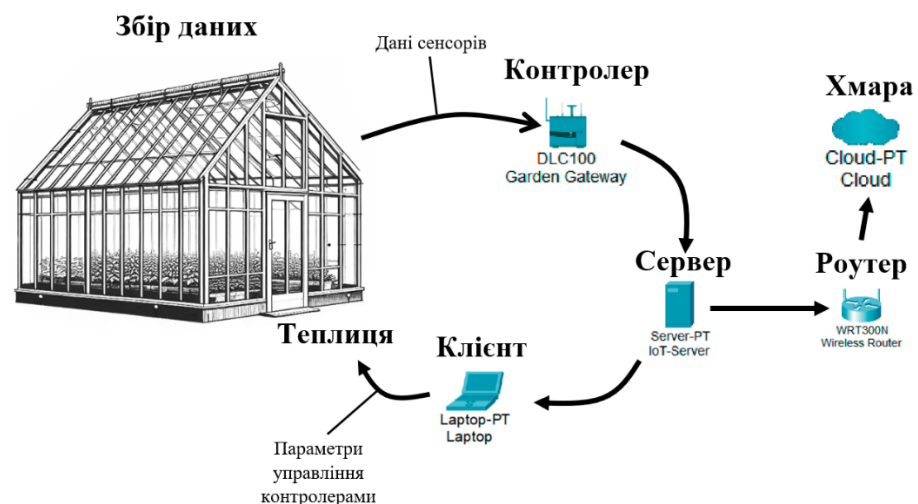


Рисунок 3.2 – Схема передачі даних в системі

Усередині теплиці розгорнута мережа датчиків, які ретельно збирають дані про різні фактори навколишнього середовища. Зібрані дані передаються на контролер, який виконує роль центрального процесора. Він інтерпретує показання датчиків і організовує подальші процеси передачі даних. Після цього приймаються рішення щодо необхідних коригувань за допомогою приводів для підтримання ідеальних умов вирощування. Приводи системи відповідають за регулювання мікроклімату в теплиці в режимі реального часу. Далі дані передаються до серверу. Останнім елементом архітектури є управління операціями через додаток, що дозволяє користувачам контролювати і наглядати за середовищем теплиці з будь-якого місця, регулюючи умови для росту рослин.

3.2. Моделювання системи «Розумна теплиця»

3.2.1 Розробка основи для моделі

Моделювання системи «Розумна теплиця» є критично важливим етапом у процесі розробки. Цей етап передбачає створення як концептуальних, так і обчислювальних моделей, які представляють фізичні та логічні компоненти тепличного середовища. Основна мета – змодельовати взаємодію між різними підсистемами та оцінити загальну продуктивність системи.

Cisco Packet Tracer надає необхідні інструменти для моделювання системи.

Модель системи «Розумна теплиця» Cisco Packet Tracer подано на рис. 3.3.

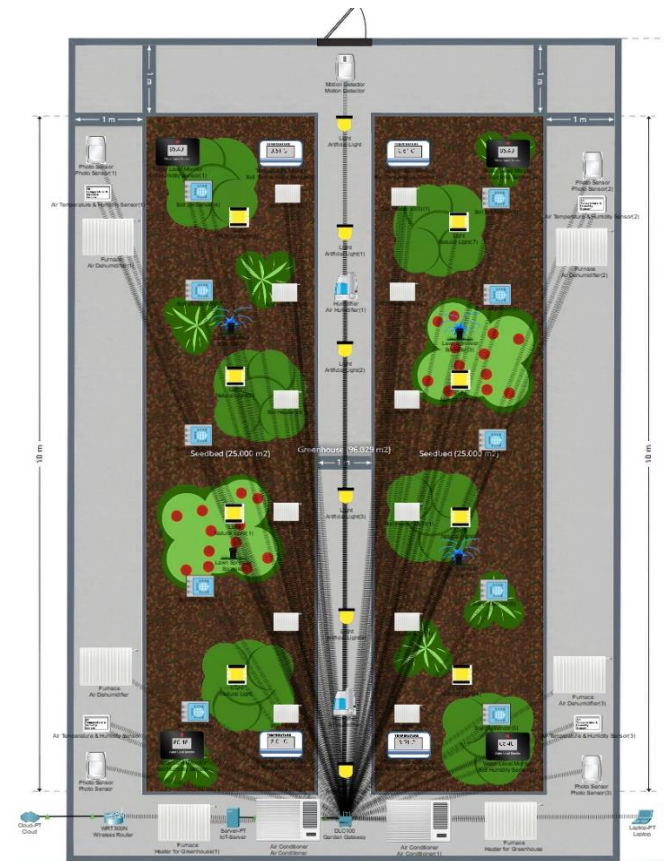


Рисунок 3.3 – Модель системи «Розумна теплиця» в Cisco Packet Tracer

За замовчуванням Cisco Packet Tracer не має деяких компонентів для розумної теплиці.

Інструмент не має пристрою, який має в своєму функціоналі можливість вимірювати вологість та температуру повітря одночасно, але досягти цього надає можливість функція об'єднання «Humidity Monitor» та «Temperature Monitor» в кластер «Air Temperature & Humidity Sensor».

Кластер «Air Temperature & Humidity Sensor» зсередини подано на рис. 3.4.

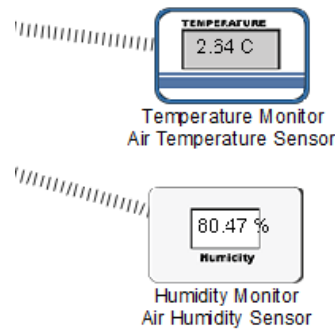


Рисунок 3.4 – Внутрішнє наповнення кластеру «Air Temperature & Humidity Sensor».

Кластер «Air Temperature & Humidity Sensor» ззовні подано на рис. 3.4.

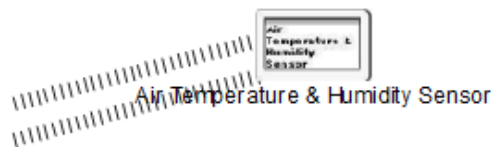


Рисунок 3.5 – Кластер «Air Temperature & Humidity Sensor»

Cisco Packet Tracer не містить в собі параметрів та пристроїв для симуляції температури та рівня рН ґрунту. За допомогою можливостей інструменту створена одиниця вимірювання «рН» та компоненти «Soil pH Monitor» для аналізу рівня рН та «pH Controller» для підвищення рівня рН в ґрунті. «Soil pH Monitor» розроблявся як новий пристрій, а «pH Controller» створений на основі пристрою «Humidifier» за допомогою зміни програмного коду. Обраний BlueLab pH Controller Connect суміщає в собі функції датчика та контролера рН, тому нові компоненти об'єднані у кластер «Soil pH Sensor»

Кластер «Soil pH Sensor» зсередини подано на рис. 3.6.

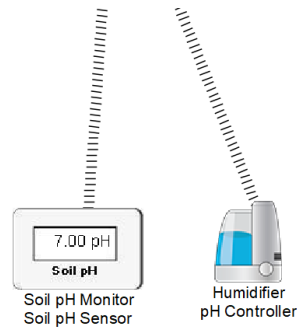


Рисунок 3.6 – Внутрішнє наповнення кластеру «Soil pH Sensor».

Кластер «Soil pH Sensor» ззовні подано на рис. 3.7.

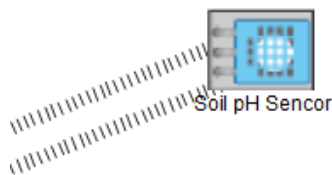


Рисунок 3.7 – Кластер «Soil pH Sensor»

«Soil Temperature Sensor» та «Soil Heater» за допомогою перепрограмування створені на основі «Temperature Monitor» та «Furnace» відповідно «Soil Heater» для зручності об'єднанні в кластери по три елементи.

Кластер «Soil Heater x3» ззовні подано на рис. 3.8.



Рисунок 3.8 – Кластер «Soil Heater x3»

У ролі Samsung SmartThings Hub використовується «DLC100 Home Gateway».

3.2.2 Налаштування мережі

Конфігурація мережі для інформаційної системи «Розумна теплиця» в Cisco Packet Tracer є важливим кроком у забезпеченні надійного зв'язку між усіма компонентами системи. Топологія мережі для «розумної» теплиці розроблена з урахуванням центрального контролера, сервера, ноутбука, маршрутизатора, хмарного середовища, сенсорних вузлів та виконавчих механізмів.

Центральний контролер виступає в ролі концентратора мережі, збираючи дані з датчиків і відправляючи команди на виконавчі пристрої. Він отримав адресу 192.168.1.1 та є шлюзом за замовчуванням для інших пристроїв. Сервер, підключений до центрального контролера, керує обробкою та зберіганням даних. Ноутбук, оснащений програмним забезпеченням, підключається до сервера за допомогою центрального контролера, який також виконує роль хаба. Це дозволяє дистанційно керувати правилами контролера. Окрім цього сервер підключений до маршрутизатора, який забезпечує зв'язок з хмарним середовищем для віддаленого доступу та резервного копіювання даних.

Налаштування мережі почалося з фізичного планування теплиці, включаючи розміщення центрального контролера, сервера, ноутбука, сенсорних вузлів і виконавчих механізмів. Кожен сенсорний вузол був налаштований на

зв'язок з центральним контролером за допомогою бездротового зв'язку. Центральний контролер був підключений до сервера за допомогою дротового з'єднання, що забезпечило стабільну та високошвидкісну передачу даних.

Безпека враховувалась під час конфігурації. У мережі Wi-Fi ввімкнено шифрування WPA2, що використовує алгоритм шифрування AES, який шифрує дані, що передаються бездротовою мережею, роблячи їх незрозумілими для будь-кого, хто перехопить їх без ключа шифрування. Це запобігає підслухуванню та фальсифікації даних, забезпечуючи конфіденційність і цілісність переданих даних.

Налаштування безпроводного зв'язку контролера подано на рис. 3.9.

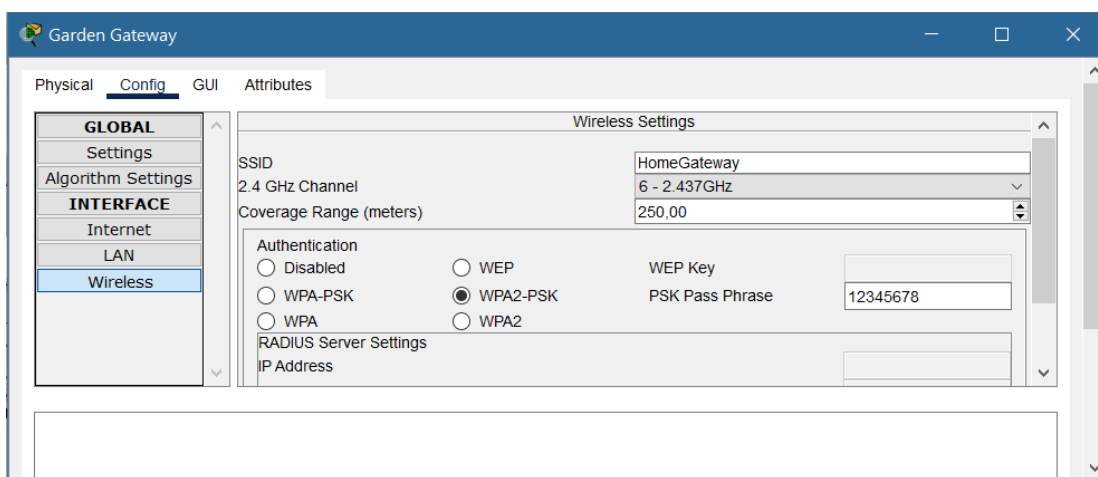


Рисунок 3.9 – Налаштування безпроводного зв'язку Garden Gateway

3.2.3 Встановлення правил для контролерів

Встановлення правил для контролерів в інформаційній системі «Розумна теплиця» передбачало визначення логіки їхньої поведінки на основі даних з датчиків та цілей управління теплицею.

Для моніторингу та управління визначені ключові параметри, такі як температура, вологість, вологість ґрунту та інтенсивність світла. Для цих параметрів були встановлені певні порогові значення, що забезпечують оптимальні умови для росту рослин.

Правила роботи пристроїв в інформаційній системі "Розумна теплиця" регулюються низкою порогових значень, які визначають активацію та деактивацію різних керуючих пристроїв на основі даних датчиків.

Керування лампами природного освітлення здійснюється шляхом порівняння показників рівня освітленості з датчиків з пороговими значеннями увімкнення та вимкнення. Кожен з датчиків пов'язаний з двома найближчими до нього лампами. Таким чином, один датчик контролює половину освітлення однієї грядки. Якщо рівень освітленості перевищує поріг вимкнення, лампи вимикаються, а як тільки рівень природного освітлення зростає до достатнього, система автоматично вмикає лампи, забезпечуючи ефективне використання енергії.

Кондиціонери та обігрівачі працюють на основі середніх показників температури повітря. Якщо температура перевищує бажаний діапазон, система вмикає кондиціонер для охолодження повітря. І навпаки, якщо температура опускається нижче бажаного діапазону, система вмикає обігрівач для підігріву повітря. Цей динамічний механізм управління гарантує, що в теплиці підтримується оптимальна температура для росту рослин при мінімальному споживанні енергії.

Зволожувач та осушувач повітря також керуються середніми показниками вологості повітря. Зволожувач повітря вмикається, якщо середня вологість перевищує поріг вимкнення, і вмикається, якщо вона опускається нижче порогу увімкнення. Осушувач повітря працює за аналогічною схемою, вмикаючись, якщо вологість падає нижче порогу вимкнення, і вмикаючись, якщо вона занадто висока, що свідчить про надлишок вологи в повітрі.

Контролери рН ґрунту активуються або деактивуються на основі показань рН ґрунту, причому кожен контролер відповідає певному датчику. Якщо рН ґрунту перевищує поріг вимкнення, контролер вмикається, а якщо опускається нижче порогу увімкнення - вмикається. Аналогічно, якщо рівень вологості ґрунту занадто високий або занадто низький, що вказує на надмірний або

недостатній полив, система керує зрошувачами ґрунту, щоб відрегулювати рівень вологості.

Ґрунтові обігрівачі керуються показаннями температури ґрунту, причому різні обігрівачі контролюються різними датчиками. Кожен датчик контролює половину обігрівачів грядки. Якщо температура ґрунту перевищує поріг вимкнення, нагрівач вимикається, а якщо опускається нижче порогу увімкнення - вмикається. Ґрунтові зрошувачі керуються показниками вологості ґрунту, вмикаючись, якщо вологість перевищує поріг вимкнення, і вмикаючись, якщо вона падає нижче порогу увімкнення.

Крім того, лампи штучного освітлення керуються на основі середнього рівня освітленості, отриманого від декількох датчиків світла, та датчику руку. Якщо середній рівень освітлення падає нижче певного порогу, що вказує на недостатнє природне освітлення, і в приміщення входить оператор теплиці, то вмикаються лампи штучного освітлення, щоб доповнити світло, необхідне для комфортного перебування в теплиці.

Визначені на сервері механізми правил вступають в силу, оскільки, центральний контролер безперервно відстежував дані з датчиків і передавав серверу, щоб визначити яким пристроям надіслати сигнали для виконання відповідних дій на основі попередньо визначених правил.

Інтерфейс для відстеження стану пристроїв подано на рис. 3.10.

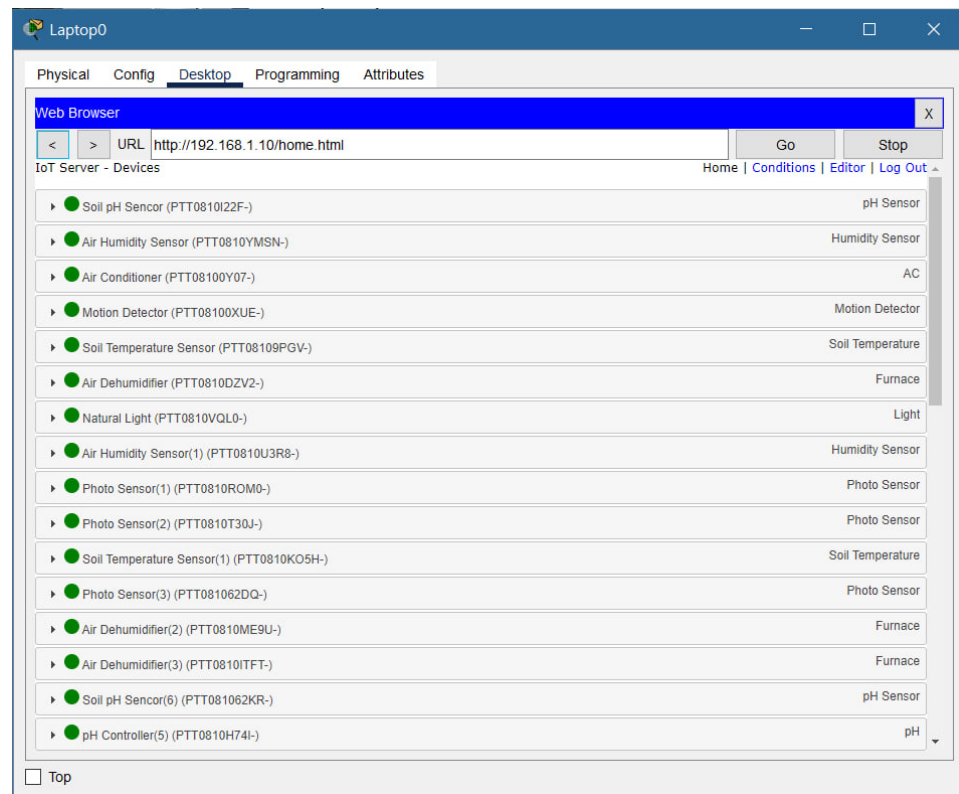


Рисунок 3.10 – Інтерфейс Cisco для відстеження стану пристроїв

Ноутбук, підключений до сервера, дозволяє легко керувати правилами. Клієнтський інтерфейс на пристрої забезпечив зручну платформу для визначення, модифікації та моніторингу правил, що регулюють поведінку контролерів. Таке налаштування уможливило ефективно та гнучке управління середовищем теплиці.

Інтерфейс зміни правил подано на рис. 3.11.

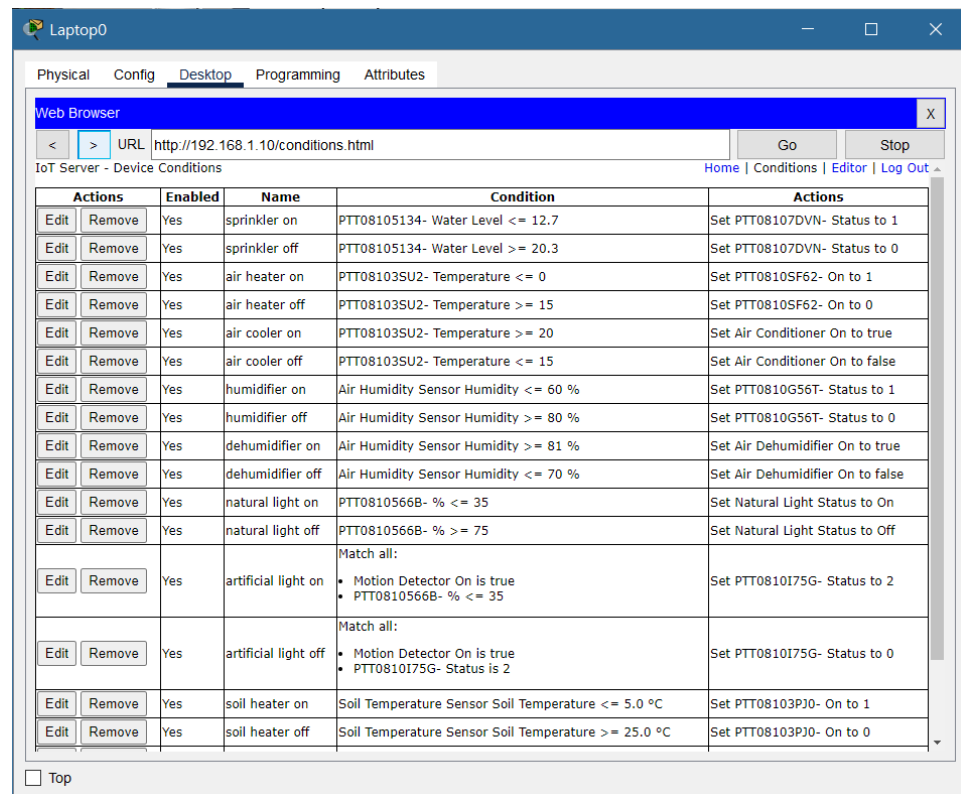


Рисунок 3.11 – Інтерфейс Cisco для зміни правил роботи контролерів

3.2.4 Моделювання мережі

У Cisco Packet Tracer мережа була змодельована шляхом створення детального плану тепличного середовища, включаючи фізичне розміщення всіх мережевих пристроїв: центрального контролера, сервера, ноутбука, маршрутизатора, сенсорних вузлів і виконавчих механізмів. Кожен пристрій було налаштовано відповідно до реального сценарію розгортання.

Потік даних в мережі був змодельований шляхом створення пакетів даних, що представляють показання датчиків і команди управління. Ці пакети були направлені через мережу, щоб спостерігати за їхнім шляхом від вузлів датчиків до центрального контролера і назад до виконавчих пристроїв. Моніторинг потоку пакетів допоміг виявити потенційні вузькі місця та точки відмови.

Моделювання потоку даних подано на рис. 3.12.

Simulation Panel				
Event List				
Vis.	Time(sec)	Last Device	At Device	Type
	0.021	Garden Gateway	pH Controller	TCP
	0.021	Garden Gateway	Laptop	TCP
	0.022	--	Artificial Light(4)	TCP
	0.023	Artificial Light(4)	Garden Gateway	TCP
	0.024	Garden Gateway	IoT-Server	TCP
	0.025	--	pH Controller(1)	TCP
	0.026	pH Controller(1)	Garden Gateway	TCP
	0.027	Garden Gateway	IoT-Server	TCP
	0.027	--	Soil pH Sencor(7)	TCP
	0.028	Soil pH Sencor(7)	Garden Gateway	TCP
	0.029	Garden Gateway	IoT-Server	TCP
	0.031	--	Garden Gateway	TCP

Рисунок 3.12 – Моделювання потоку даних в Cisco Packet Tracer

3.3. Розробка програмного додатку управління системою

Розробка програмного забезпечення для управління розумною теплицею є важливим аспектом системи, що надає користувачам зручний та інтуїтивно зрозумілий інтерфейс для віддаленого моніторингу та управління роботою теплиці. Додаток розроблений таким чином, щоб легко інтегруватися з існуючою апаратною та програмною інфраструктурою, використовуючи сучасні технології, щоб запропонувати комплексне рішення для управління.

Програму створено з використанням мови програмування Python та інструменту Tkinter з розширенням ttkbootstrap. В результаті додаток має зручний інтерфейс, розроблений з акцентом на простоту, зручність та доступність.

Головне меню надає доступ до трьох основних функцій: відображення даних отриманими від датчиків, відображення інформації про стан контролюючих пристроїв та редагування порогових значень для встановлених правил.

Головне меню додатку подано на рис. 3.13.

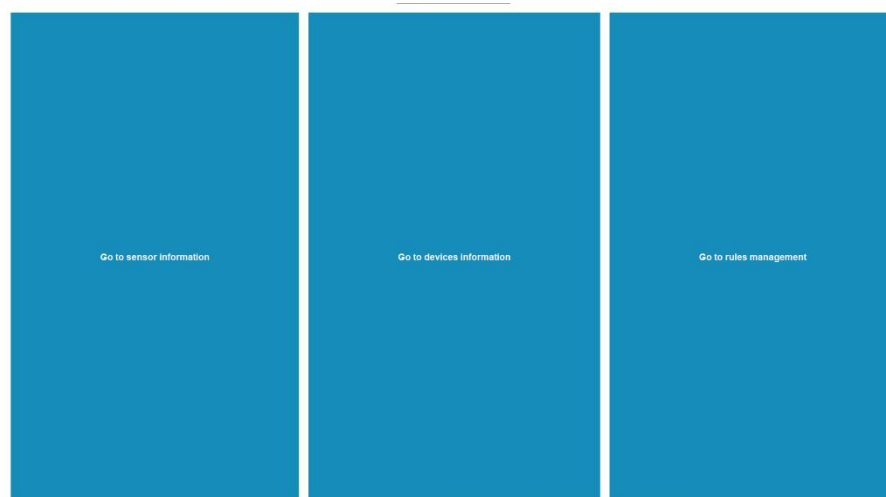


Рисунок 3.13 – Головне меню додатку

Перша функція надає доступ до інформаційної панелі для огляду поточного стану теплиці, включаючи дані в режимі реального часу з різних датчиків, таких як температура, вологість, вологість ґрунту та рівень освітленості. Ці дані представлені у візуально привабливій формі за допомогою таблиці, що дозволяє користувачам швидко оцінити умови в теплиці.

Сторінку сенсорів додатку подано на рис. 3.14.

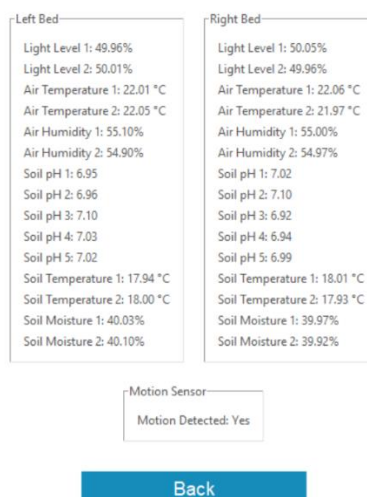


Рисунок 3.14 – Сторінка сенсорі

Аналогічним чином представлені дані для другої функції.
Сторінку контролерів додатку подано на рис. 3.15.

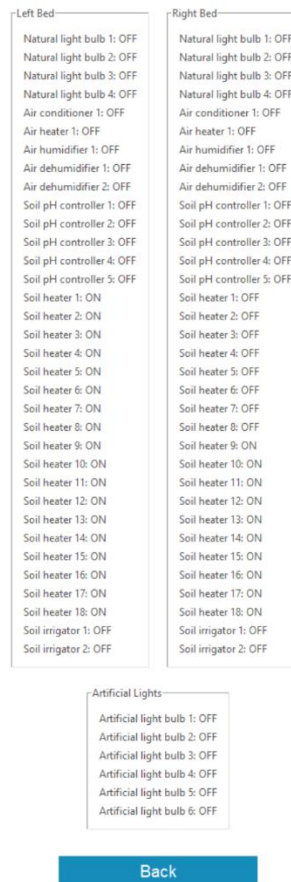


Рисунок 3.15 – Сторінка контролерів

В обох випадках оператор має можливість дізнатися про певні проблеми в роботі системи, якщо з певним пристроєм було розірвано зв'язок.

Третя функція відкриває доступ до підменю з вибором типу контролюючого компонента системи, для якого існує можливість змінити порогові значення для пов'язаного з цим пристроєм правила.

Сторінку редагування правил кондиціонера подано на рис. 3.15.

Air conditioners on threshold:

Air conditioners off threshold:

Рисунок 3.15 – Сторінка редагування правил кондиціонера

Висновки за розділом 3

У Розділі 3 виконано всебічний огляд процесу розробки програмного забезпечення для інформаційної системи «Розумна теплиця». У ньому висвітлюються найважливіші етапи створення функціональної, ефективної та зручної для користувача системи, від розробки алгоритмів до конфігурації мережі та дизайну програмного додатку. Детальне моделювання та встановлення правил забезпечують безперебійну роботу системи, а додаток покращує взаємодію системи користувачами та забезпечує її контроль.

ВИСНОВКИ

Дана робота була присвячена розробці інформаційної системи, призначеної для автоматизації процесів управління мікрокліматом в теплицях з використанням технологій Інтернету речей. Був проведений комплексний огляд існуючих систем «Розумна теплиця», в результаті якого були виявлені недоліки існуючих систем, які часто не мають достатньої інтеграції і вимагають значних витрат на налаштування і обслуговування. При розробці архітектури системи була запропонована структурна схема, а також детальний план теплиці з ретельним підбором та обґрунтуванням компонентів системи, технологій зв'язку та інтеграції з сервером. Було створено алгоритм функціонування інформаційної системи та проведено моделювання мережі за допомогою Cisco Packet Tracer для перевірки працездатності системи в реальних умовах. Це включало в себе конфігурацію мережі, встановлення правил для контролерів та розробку надійної моделі мережі. Крім того, було розроблено додаток для управління розумною теплицею, що надає користувачам інтуїтивно зрозумілий інтерфейс для віддаленого моніторингу та управління середовищем теплиці. Дослідження демонструє, що запропонована інформаційна система «Розумна теплиця» може значно покращити автоматизацію та ефективність управління теплицями, пропонуючи спрощене та ефективне рішення для оптимізації умов вирощування, тим самим підтримуючи сільськогосподарський сектор у підвищенні продуктивності та стійкості. Система підтвердила свій потенціал для широкого застосування у фермерських господарствах та агропромислових підприємствах, що займаються тепличним господарством.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bhujel, A.; Basak, J.K.; Khan, F.; Arulmozhi, E.; Jaihuni, M.; Sihalath, T.; Kim, H.T. Sensor Systems for Greenhouse Microclimate Monitoring and Control: A Review. *J. Biosyst. Eng.* 2020, 45, 341–361.
2. Howard, D.A., Ma, Z., Veje, C. et al. Greenhouse industry 4.0 – digital twin technology for commercial greenhouses. *Energy Inform* 4 (Suppl 2), 37 (2021)
3. Imran, M.; Ozcatalbas, O. Optimization of energy consumption and its effect on the energy use efficiency and greenhouse gas emissions of wheat production in Turkey. *Discov. Sustain.* 2021, 2, 28.
4. Kumar, D.C.; Adiraju, R.V.; Pasupuleti, S.; Nandan, D. A Review of Smart Greenhouse Farming by Using Sensor Network Technology. In *Proceedings of the International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications*, Hyderabad, India, 28–29 March 2020; Springer: Singapore, 2021; pp. 849–856.
5. La Notte, L.; Giordano, L.; Calabrò, E.; Bedini, R.; Colla, G.; Puglisi, G.; Reale, A. Hybrid and organic photovoltaics for greenhouse applications. *Appl. Energy* 2020, 278, 115582.
6. Li, H.; Guo, Y.; Zhao, H.; Wang, Y.; Chow, D. Towards automated greenhouse: A state of the art review on greenhouse monitoring methods and technologies based on internet of things. *Comput. Electron. Agric.* 2021, 191, 106558.
7. Packet Tracer 8.2- IoT devices configuration [Електронний ресурс]. – режим доступу: URL: <https://www.packettracernetwork.com/internet-of-things/pt7-iot-devices-configuration.html> (дата звернення – 16.01.2024).
8. Sampaio, H.; Motoyama, S. Implementation of a greenhouse monitoring system using hierarchical wireless sensor network. In *Proceedings of the 2017 IEEE 9th Latin-American conference on communications (LATINCOM)*, Guatemala City, Guatemala, 8–10 November 2017; pp. 1–5.
9. The Python Tutorial — Python 3.12.3 documentation [Електронний ресурс]. – режим доступу: URL: <https://docs.python.org/3/tutorial/index.html> (дата звернення – 11.02.2024).

10. Tutorial - ttkbootstrap - Read the Docs [Электронный ресурс]. – режим доступа:
URL: <https://ttkbootstrap.readthedocs.io/en/latest/gettingstarted/tutorial/>
(дата звернення – 28.02.2024).
11. Zhang, K.S.; Zhang, X.W.; Zhou, Y.; Tang, W. Design of agricultural greenhouse environment monitoring system based on internet of things technology. In Advanced Materials Research; Trans Tech Publications Ltd.: Beijing, China, 2013; Volume 791, pp. 1651–1655.

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **бакалавр**
Галузь знань: **12 – Інформаційні технології**
Спеціальність: **123 – Комп'ютерна інженерія.**

ЗАТВЕРДЖУЮ
Завідувач кафедри теоретичної
та прикладної системотехніки
д.т.н., проф. Шматков С. І.
«21» грудня 2024 року



З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

ГРИГОР'ЄВ ДАНИЛО РОМАНОВИЧ

(прізвище, ім'я, по батькові студента)

1. Тема роботи «**Інформаційна система “Розумна теплиця”**»

керівник роботи Бикова Тетяна Володимирівна, к.т.н., доцент кафедри ТПС
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «03» травня 2024 року № 4101-5/909

2. Строк подання студентом роботи 31 травня 2024 року

3. Перелік питань, які потрібно розробити:

- 1) Огляд сучасних підходів до проектування розумних теплицю. Формування вимог до технічних характеристик. Розробка технічного завдання.
- 2) Розробка структурної схеми розумної теплиці та вибір необхідного обладнання.
- 3) Розробка програмного забезпечення для підсистеми підтримки клімату в теплиці.
- 4) Тестування програмного забезпечення.
- 5) Оформлення пояснювальної записки.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Аналіз наукової літератури та існуючих технологій, пов'язаних з системою "Розумної теплиці".	21.12.2023 – 25.01.2024
2	Формування вимог до технічних характеристик.	19.12.2023 – 02.01.2024
3	Розробка технічного завдання.	02.01.2024 – 02.02.2024
4	Розробка структурної схеми розумної теплиці та вибір необхідного обладнання.	02.01.2024 – 02.02.2024
5	Розробка програмного забезпечення	03.02.2024 – 30.03.2024
6	Тестування та валідація роботи системи "Розумного теплиця"	03.03.2024 – 30.04.2024
7	Підготовка технічного звіту	31.03.2024 – 27.05.2024
8	Розробка пояснювальної записки.	31.03.2024 – 27.05.2024
9	Представлення кваліфікаційної роботи керівнику та рецензенту.	15.05.2024 – 31.05.2024
10	Оформлення пояснювальної записки та підготовка презентації.	31.05.2024

5. Дата видачі завдання 21.12.2023

Студент

Д. Р. Григор'єв

ініціали, прізвище


 підпис

Керівник роботи

Т. В. Бикова

ініціали, прізвище


 підпис

Затверджую

«_____» _____ 2024 р.


**Технічне завдання
на розробку системи
«ІНФОРМАЦІЙНА СИСТЕМА «РОЗУМНА ТЕПЛИЦЯ»»**

1.	Введення	1.1. Назва: Інформаційна система "Розумна теплиця". 1.2. Галузь застосування: інформаційні технології.
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп'ютерна інженерія 2.2. Завдання на кваліфікаційну роботу бакалавра № № 4101-5/909 від «03» травня 2024 (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3.	Призначення розробки	3.1. Мета розробки: забезпечення автоматизації процесів управління кліматичними умовами, оптимізуючи використання ресурсів та підвищуючи продуктивність рослинництва за допомогою впровадження інформаційної системи. 3.2. Призначення розробки надає можливість автоматизувати контроль та управління кліматичними умовами в теплиці, забезпечуючи оптимальні умови для росту рослин. Система інтегрується з розумними пристроями для регулювання температури, вологості, освітлення та поливу. 3.3. Вхідні дані розробки: дані про поточний стан клімату в теплиці (температура, вологість повітря та ґрунту, інтенсивність освітлення), команди від користувача для зміни налаштувань, дані про стан пристроїв, що контролюють стан клімату. 3.4. Вихідні дані розробки: дані про поточний стан клімату в теплиці (температура, вологість повітря та ґрунту, інтенсивність освітлення), дані про стан пристроїв, що контролюють стан клімату.
4.	Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: автоматизація процесів моніторингу та управління кліматичними умовами для оптимального росту рослин, можливість дистанційного керування правилами роботи системи теплиці через інтерфейс програмного додатку

		<p>4.2. Вимоги до нефункціональних характеристик: інтуїтивно зрозумілий інтерфейс для спрощення процесів управління та налаштування</p> <p>4.3. Вимоги до надійності: забезпечувати точність та достовірність визначення параметрів стану не гіршу за існуючі методи та програмне забезпечення</p> <p>4.4. Вимоги до умов експлуатації: немає</p> <p>4.5. Вимоги до складу і параметрів технічних засобів: датчики, керуючі пристрої, контролер, сервер, роутер, ПК, тощо</p> <p>4.6. Вимоги до інформаційної та програмної сумісності: немає</p> <p>4.7. Вимоги до маркування та упаковки: немає</p> <p>4.8. Вимоги до транспортування і зберігання: на звичайних носіях інформації</p> <p>4.9. Спеціальні вимоги: немає.</p>	
5.	Вимоги до програмної документації	<p>Програмною документацією до виробу «Інформаційна система «Розумна теплиця»» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Методику розрахунку інформативності змінних стану (у вигляді <i>глав 1.1 та 1.3</i> пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Опис виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)</p>	
6.	Вимоги до техніко-економічних показників	<p>Програмною документацією до виробу «Метод аналізу інформативності змінних стану при діагностиці систем з використанням інформаційних критеріїв» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Методику розрахунку інформативності змінних стану (у вигляді <i>глав 1.1 та 1.3</i> пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Опис виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)</p>	
7.	Стадії і етапи розробки	Дата	Назва етапу
		від 21 грудня 2023 до 25 січня 2024	Аналіз наукової літератури та існуючих технологій, пов'язаних з системою «Розумної теплиці».

		<p>від 19 грудня 2023 до 2 січня 2024</p> <p>від 2 січня 2024 до 2 лютого 2024</p> <p>від 2 січня 2024 до 2 лютого 2024</p> <p>від 3 лютого 2024 до 30 березня 2024</p> <p>від 3 березня 2024 до 30 квітня 2024</p> <p>від 31 квітня 2024 до 27 травня 2024</p> <p>від 15 травня 2024 до 31 травня 2024</p> <p>31 травня 2024</p>	<p>Формування вимог до технічних характеристик.</p> <p>Розробка технічного завдання.</p> <p>Розробка структурної схеми розумної теплиці та вибір необхідного обладнання.</p> <p>Розробка програмного забезпечення</p> <p>Тестування та валідація роботи системи «Розумного теплиця»</p> <p>Розробка пояснювальної записки.</p> <p>Представлення кваліфікаційної роботи керівнику та рецензенту</p> <p>Оформлення пояснювальної записки та підготовка презентації</p>
8.	Порядок контролю і приймання програмного продукту (моделі)	<ol style="list-style-type: none"> 1. Перевірку ходу розробки програми виконувати раз в 3 тижні. 2. Захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати в електронному вигляді в 1 примірнику. 	

Виконавець
студент групи КІ- 41
ГРИГОР'ЄВ Д. Р.



Замовник
д. т. н., проф.
БИКОВА Т. В.



Програма і методика випробувань програмного виробу
«ІНФОРМАЦІЙНА СИСТЕМА «РОЗУМНА ТЕПЛИЦЯ»»

1 Об'єкт випробувань

1. Назва програмного виробу : «Інформаційна система «Розумна теплиця»»
2. Галузь застосування : Інформаційні технології
3. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності функціональності програмної реалізації системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення

1. Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

2. Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

1. працювати на найбільш поширеній операційній системі Windows;
2. вимоги до надійності;
3. передбачити захист від некоректних дій користувача;
4. сумісність з іншими програмними продуктами;
5. зменшити об'єм програмного коду необхідного для створення настільних додатків;

6. бути легко розширюваною;
7. елементи програми повинні бути ізольовані одне від одного для зменшення їх впливу на роботи програми під час редагування програмного коду;
8. вимоги до складу і параметрів технічних засобів;
9. вимоги до маркування та упаковки (не висуваються);
10. вимоги до транспортування і зберігання (не висуваються).
11. спеціальні вимоги (не висуваються).

5. Вимоги до програмної документації

Програмною документацією щодо розроблюваного програмного продукту вважати:

1. справжнє технічне завдання на розробку програми (представити як Додаток Б до пояснювальної записки до кваліфікаційної роботи);
2. Програму і методику випробувань розробленої програми (представити як Додаток В до пояснювальної записки до кваліфікаційної роботи);
3. рекомендацій щодо застосування створеної програмної стандартизації у проєктах (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи).
4. Текст програми (представити як Додаток Г до пояснювальної записки до кваліфікаційної роботи).

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Для проведення випробувань необхідний проєкт для розробки веб-додатку на мові програмування Python з використання інструменту Tkinter з розширенням ttkbootstrap.

6.2 Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

-ознайомчий (1-й етап);

-випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1. Перевірку комплектності програмної документації.
2. Перевірка комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.
3. Перевірку комплектності складу технічних і програмних засобів.
4. Методику проведення перевірок на 1 етапі випробувань.
5. Якість програмної документації перевіряється на відповідність вимогам стандартів ЕСПД.

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

1. перевірку відповідності технічних характеристик програми вимогам технічного завдання;
 2. перевірку ступеня виконання функціональних вимог до програми;
 3. методику проведення перевірок, що входять до переліку по 2 етапу випробувань.
-
1. Програма працює відповідно до умов експлуатації операційної системи MS Windows.
 2. Для роботи необхідний компілятор мови програмування Python, версії не нижчої ніж 3.0.
 3. Для роботи необхідний додаток ttkbootstrap до мови програмування Python.
 4. Порядок проведення випробувань:
 - 4.1. Запуск програми здійснюється за допомогою активації виконавчого файлу «SmartGreenhouse.exe»;
 - 4.2. Після запуску програми необхідно обрати один з трьох пунктів меню для доступу до відповідного функціоналу, в третьому пункті також необхідно обрати вид керуючого пристрою, для якого застосовуються зміни граничних правил;
 - 4.3. Після натискання на екрані з'явиться результат відповіді програми. Для проведення випробувань пропонується тест 1, тест 2 та тест 3.

Тест 1

1. Перевірка виконання програми;
2. Перехід до сторінки інформаційної панелі, яка відображає дані, отримані з датчиків;
3. Отримання нових даних від кожного з датчиків.

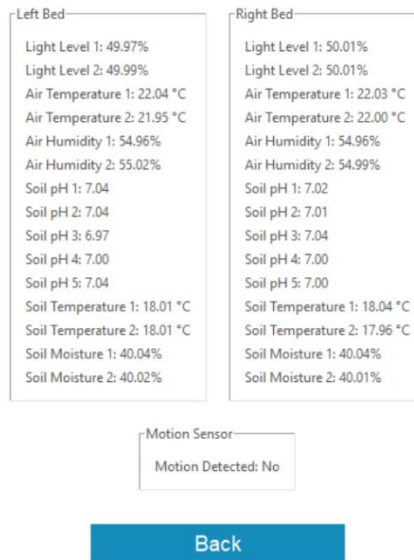


Рис. В.1 Тест 1

Тест 2

1. Перевірка виконання програми;
2. Перехід до сторінки інформаційної панелі, яка відображає дані про поточний стан керуючих пристроїв;
3. Отримання нових даних від кожного з датчиків.

Left Bed	Right Bed
Natural light bulb 1: OFF	Natural light bulb 1: OFF
Natural light bulb 2: OFF	Natural light bulb 2: OFF
Natural light bulb 3: OFF	Natural light bulb 3: OFF
Natural light bulb 4: OFF	Natural light bulb 4: OFF
Air conditioner 1: OFF	Air conditioner 1: OFF
Air heater 1: OFF	Air heater 1: OFF
Air humidifier 1: OFF	Air humidifier 1: OFF
Air dehumidifier 1: OFF	Air dehumidifier 1: OFF
Air dehumidifier 2: OFF	Air dehumidifier 2: OFF
Soil pH controller 1: OFF	Soil pH controller 1: OFF
Soil pH controller 2: OFF	Soil pH controller 2: OFF
Soil pH controller 3: OFF	Soil pH controller 3: OFF
Soil pH controller 4: OFF	Soil pH controller 4: OFF
Soil pH controller 5: OFF	Soil pH controller 5: OFF
Soil heater 1: ON	Soil heater 1: OFF
Soil heater 2: ON	Soil heater 2: OFF
Soil heater 3: ON	Soil heater 3: OFF
Soil heater 4: ON	Soil heater 4: OFF
Soil heater 5: ON	Soil heater 5: OFF
Soil heater 6: ON	Soil heater 6: OFF
Soil heater 7: ON	Soil heater 7: OFF
Soil heater 8: ON	Soil heater 8: OFF
Soil heater 9: ON	Soil heater 9: ON
Soil heater 10: ON	Soil heater 10: ON
Soil heater 11: ON	Soil heater 11: ON
Soil heater 12: ON	Soil heater 12: ON
Soil heater 13: ON	Soil heater 13: ON
Soil heater 14: ON	Soil heater 14: ON
Soil heater 15: ON	Soil heater 15: ON
Soil heater 16: ON	Soil heater 16: ON
Soil heater 17: ON	Soil heater 17: ON
Soil heater 18: ON	Soil heater 18: ON
Soil irrigator 1: OFF	Soil irrigator 1: OFF
Soil irrigator 2: OFF	Soil irrigator 2: OFF

Artificial Lights
Artificial light bulb 1: OFF
Artificial light bulb 2: OFF
Artificial light bulb 3: OFF
Artificial light bulb 4: OFF
Artificial light bulb 5: OFF
Artificial light bulb 6: OFF

[Back](#)

Рис. В.2 Тест 2

Тест 3

1. Перевірка виконання програми;
2. Перехід до сторінки зміни правил керуючих пристроїв та обрання кондиціонеру;
3. Зміна параметру та його збереження.

Air conditioners on threshold:

Air conditioners off threshold:

[Save](#)

[Back](#)

Рис. В.3 Тест 3

Тест вважається пройденим, якщо відбуваються вказані операції і їх відображення у програмному продукті.

Висновки: тест 1 успішно пройшов випробування, тест 2 успішно пройшов випробування і тест 3 успішно пройшов випробування. Випробування пройшло успішно.

Виконавець
студент групи КІ- 41
ГРИГОР'ЄВ Д. Р.



Лістинг головного класу main:

```

using SmartHome.Service;
import tkinter as tk
from smart_greenhouse_gui import SmartGreenhouseGUI
from network import SmartGreenhouseClientNetwork

if __name__ == "__main__":
    root = tk.Tk()
    gui = SmartGreenhouseGUI(root)
    network = SmartGreenhouseClientNetwork(gui, root)
    gui.network = network

    root.mainloop()

```

Лістинг графічного інтерфейсу SmartGreenhouseGUI:

```

from ttkbootstrap import Style

from devices_page import DevicesPage
from main_menu import MainMenu
from rules_management_page import RulesManagementPage
from sensor_page import SensorPage

from threshold_manager import NaturalLightBulbManager
from threshold_manager import AirConditioner
from threshold_manager import AirHeaterManager
from threshold_manager import AirHumidifierManager
from threshold_manager import AirDehumidifierManager
from threshold_manager import SoilPhControllerManager
from threshold_manager import SoilHeaterManager
from threshold_manager import IrrigatorManager

class SmartGreenhouseGUI:
    def __init__(self, main_root):
        self.network = None
        self.root = main_root
        main_root.attributes('-fullscreen', True) # Open in
full screen mode
        self.style = Style(theme='lumen') # Initialize
ttkbootstrap style

        self.root.title("Smart Greenhouse Client")

        # Set the window size to the screen size

self.root.geometry("{0}x{1}+0+0".format(self.root.winfo_screen
width(), self.root.winfo_screenheight()))

```

```

# Make the window adaptive to screen size changes
self.root.bind("<Configure>", self.on_configure)

self.main_menu = MainMenu(self)
self.sensor_page = SensorPage(self)
self.devices_page = DevicesPage(self)
self.rules_management_page = RulesManagementPage(self)

self.natural_light_bulb_management_page =
NaturalLightBulbManager(self)
self.air_conditioner_management_page =
AirConditioner(self)
self.air_heater_management_page =
AirHeaterManager(self)
self.air_humidifier_management_page =
AirHumidifierManager(self)
self.air_dehumidifier_management_page =
AirDehumidifierManager(self)
self.soil_ph_controller_management_page =
SoilPhControllerManager(self)
self.soil_heater_management_page =
SoilHeaterManager(self)
self.irrigator_management_page =
IrrigatorManager(self)

self.show_main_menu()

# noinspection PyUnusedLocal
def on_configure(self, event=None):

self.root.geometry("{0}x{1}+0+0".format(self.root.wininfo_screen
width(), self.root.wininfo_screenheight()))
    pass

def show_main_menu(self):
    self.main_menu.show()

def show_sensor_page(self):
    self.sensor_page.show()

def show_devices_page(self):
    self.devices_page.show()

def show_rules_management_page(self):
    self.rules_management_page.show()

def show_natural_light_bulb_management_page(self):
    self.natural_light_bulb_management_page.show()

def show_air_conditioner_management_page(self):
    self.air_conditioner_management_page.show()

```

```

def show_air_heater_management_page(self):
    self.air_heater_management_page.show()

def show_air_humidifier_management_page(self):
    self.air_humidifier_management_page.show()

def show_air_dehumidifier_management_page(self):
    self.air_dehumidifier_management_page.show()

def show_soil_ph_controller_management_page(self):
    self.soil_ph_controller_management_page.show()

def show_soil_heater_management_page(self):
    self.soil_heater_management_page.show()

def show_irrigator_management_page(self):
    self.irrigator_management_page.show()

```

Лістинг класу сторінки датчиків SensorPage:

```

import tkinter as tk

class SensorPage:
    def __init__(self, app):
        self.app = app

        self.sensor_frame = tk.Frame(self.app.root)
        self.left_sensor_frame =
tk.LabelFrame(self.sensor_frame, text="Left Bed", padx=10,
pady=10)
        self.right_sensor_frame =
tk.LabelFrame(self.sensor_frame, text="Right Bed", padx=10,
pady=10)
        self.motion_sensor_frame =
tk.LabelFrame(self.sensor_frame, text="Motion Sensor",
padx=10, pady=10)

        self.left_sensor_labels =
self.create_sensors_bed_labels(self.left_sensor_frame)
        self.right_sensor_labels =
self.create_sensors_bed_labels(self.right_sensor_frame)
        self.motion_sensor_label =
tk.Label(self.motion_sensor_frame, text="Motion Detected:
N/A")

        self.sensor_to_main_menu_button = tk.Button(
            self.sensor_frame,
            text="Back",
            command=self.app.show_main_menu,
            font=("Helvetica", 14),
            width=20
        )

```

```

self.create_sensor_page()

def show(self):
    self.app.main_menu.main_menu_frame.pack_forget()
    self.sensor_frame.pack(fill='both', expand=True)

    self.sensor_frame.pack_configure(
        padx=(self.app.root.winfo_width() -
self.sensor_frame.winfo_reqwidth()) // 2,
        pady=(self.app.root.winfo_height() -
self.sensor_frame.winfo_reqheight()) // 2)

def create_sensor_page(self):

    self.left_sensor_frame.grid(row=0, column=0, padx=10,
pady=10)
    self.right_sensor_frame.grid(row=0, column=1, padx=10,
pady=10)
    self.motion_sensor_frame.grid(row=1, column=0,
columnspan=2, padx=10, pady=10)

    self.motion_sensor_label.grid(row=0, column=0,
sticky="w")

    self.sensor_to_main_menu_button.grid(row=2, column=0,
columnspan=2, pady=20)

    @staticmethod
    def create_sensors_bed_labels(frame):
        labels = {
            "light_level": [tk.Label(frame, text=f"Light Level
{i + 1}: N/A") for i in range(2)],
            "air_temperature": [tk.Label(frame, text=f"Air
Temperature {i + 1}: N/A") for i in range(2)],
            "air_humidity": [tk.Label(frame, text=f"Air
Humidity {i + 1}: N/A") for i in range(2)],
            "soil_ph": [tk.Label(frame, text=f"Soil pH {i +
1}: N/A") for i in range(5)],
            "soil_temperature": [tk.Label(frame, text=f"Soil
Temperature {i + 1}: N/A") for i in range(2)],
            "soil_moisture": [tk.Label(frame, text=f"Soil
Moisture {i + 1}: N/A") for i in range(2)]
        }

        row = 0
        for key, label_list in labels.items():
            for label in label_list:
                label.grid(row=row, column=0, sticky="w")
                row += 1

    return labels

```

```

    @staticmethod
    def update_sensor_bed_labels(labels, data):
        for i, label in enumerate(labels["light_level"]):
            label.config(text=f"Light Level {i + 1}:
{data['light']['level'][i]:.2f}%")
        for i, label in enumerate(labels["air_temperature"]):
            label.config(text=f"Air Temperature {i + 1}:
{data['air']['temperature'][i]:.2f} °C")
        for i, label in enumerate(labels["air_humidity"]):
            label.config(text=f"Air Humidity {i + 1}:
{data['air']['humidity'][i]:.2f}%")
        for i, label in enumerate(labels["soil_ph"]):
            label.config(text=f"Soil pH {i + 1}:
{data['soil']['ph'][i]:.2f}")
        for i, label in enumerate(labels["soil_temperature"]):
            label.config(text=f"Soil Temperature {i + 1}:
{data['soil']['temperature'][i]:.2f} °C")
        for i, label in enumerate(labels["soil_moisture"]):
            label.config(text=f"Soil Moisture {i + 1}:
{data['soil']['moisture'][i]:.2f}%")

```

Лістинг класу сторінки керуючих пристроїв DevicesPage:

```

import tkinter as tk

class DevicesPage:
    def __init__(self, app):

        self.app = app

        self.devices_frame = tk.Frame(self.app.root)
        self.left_devices_frame =
tk.LabelFrame(self.devices_frame, text="Left Bed", padx=10,
pady=10)
        self.right_devices_frame =
tk.LabelFrame(self.devices_frame, text="Right Bed", padx=10,
pady=10)
        self.artificial_lights_frame =
tk.LabelFrame(self.devices_frame, text="Artificial Lights",
padx=10, pady=10)

        self.left_devices_labels =
self.create_devices_bed_labels(self.left_devices_frame)
        self.right_devices_labels =
self.create_devices_bed_labels(self.right_devices_frame)
        self.artificial_lights_label =
self.create_artificial_light_devices_label(self.artificial_lig
hts_frame)

        self.devices_to_main_menu_button = tk.Button(
            self.devices_frame,

```

```

        text="Back",
        command=self.app.show_main_menu,
        font=("Helvetica", 14),
        width=20
    )

    self.create_devices_page()

def show(self):
    self.app.main_menu.main_menu_frame.pack_forget()
    self.devices_frame.pack(fill='both', expand=True)

    self.devices_frame.pack_configure(
        padx=(self.app.root.winfo_width() -
self.devices_frame.winfo_reqwidth()) // 2,
        pady=(self.app.root.winfo_height() -
self.devices_frame.winfo_reqheight()) // 2)

    def create_devices_page(self):
        self.left_devices_frame.grid(row=0, column=0, padx=10,
pady=10)
        self.right_devices_frame.grid(row=0, column=1,
padx=10, pady=10)
        self.artificial_lights_frame.grid(row=1, column=0,
columnspan=2, padx=10, pady=10)

        self.devices_to_main_menu_button.grid(row=2, column=0,
columnspan=2, pady=20)

    @staticmethod
    def create_devices_bed_labels(frame):
        labels = {
            "natural_light_bulb": [tk.Label(frame,
text=f"Natural light bulb {i + 1}: N/A") for i in range(4)],
            "air_conditioner": [tk.Label(frame, text=f"Air
conditioner {i + 1}: N/A") for i in range(1)],
            "air_heater": [tk.Label(frame, text=f"Air heater
{i + 1}: N/A") for i in range(1)],
            "air_humidifier": [tk.Label(frame, text=f"Air
humidifier {i + 1}: N/A") for i in range(1)],
            "air_dehumidifier": [tk.Label(frame, text=f"Air
dehumidifier {i + 1}: N/A") for i in range(2)],
            "soil_ph_controller": [tk.Label(frame, text=f"Soil
pH controller {i + 1}: N/A") for i in range(5)],
            "soil_heater": [tk.Label(frame, text=f"Soil heater
{i + 1}: N/A") for i in range(18)],
            "soil_irrigator": [tk.Label(frame, text=f"Soil
irrigator {i + 1}: N/A") for i in range(2)]
        }

        row = 0
        for key, label_list in labels.items():

```

```

        for label in label_list:
            label.grid(row=row, column=0, sticky="w")
            row += 1

    return labels

    def update_devices_bed_labels(self, labels_1, labels_2,
data):
    status = ["ON", "OFF"]
    mean_value = 0
    tmp = 0

    natural_light_bulb_off =
self.app.natural_light_bulb_management_page.off_threshold_var.
get()
    natural_light_bulb_on =
self.app.natural_light_bulb_management_page.on_threshold_var.g
et()

    air_conditioner_off =
self.app.air_conditioner_management_page.off_threshold_var.get
()
    air_conditioner_on =
self.app.air_conditioner_management_page.on_threshold_var.get(
)

    air_heater_off =
self.app.air_heater_management_page.off_threshold_var.get()
    air_heater_on =
self.app.air_heater_management_page.on_threshold_var.get()

    air_humidifier_off =
self.app.air_humidifier_management_page.off_threshold_var.get(
)
    air_humidifier_on =
self.app.air_humidifier_management_page.on_threshold_var.get()

    air_dehumidifier_off =
self.app.air_dehumidifier_management_page.off_threshold_var.ge
t()
    air_dehumidifier_on =
self.app.air_dehumidifier_management_page.on_threshold_var.get
()

    soil_ph_controller_off =
self.app.soil_ph_controller_management_page.off_threshold_var.
get()
    soil_ph_controller_on =
self.app.soil_ph_controller_management_page.on_threshold_var.g
et()

```

```

        soil_heater_off =
self.app.soil_heater_management_page.off_threshold_var.get()
        soil_heater_on =
self.app.soil_heater_management_page.on_threshold_var.get()

        soil_irrigator_off =
self.app.irrigator_management_page.off_threshold_var.get()
        soil_irrigator_on =
self.app.irrigator_management_page.on_threshold_var.get()

        for i, device_label in
enumerate(labels_1["natural_light_bulb"]):
            for j, sensor_label in
enumerate(labels_2["light_level"]):
                if (i <= 1) and (j == 0):
                    if data['light']['level'][j] >=
natural_light_bulb_off:
                        device_label.config(text=f"Natural
light bulb {i + 1}: {status[1]}")
                    elif data['light']['level'][j] <=
natural_light_bulb_on:
                        device_label.config(text=f"Natural
light bulb {i + 1}: {status[0]}")
                    else:
                        device_label.config(text=f"Natural
light bulb {i + 1}: {status[1]}")
                    elif (i >= 2) and (j == 1):
                        if data['light']['level'][j] >=
natural_light_bulb_off:
                            device_label.config(text=f"Natural
light bulb {i + 1}: {status[1]}")
                        elif data['light']['level'][j] <=
natural_light_bulb_on:
                            device_label.config(text=f"Natural
light bulb {i + 1}: {status[0]}")
                        else:
                            device_label.config(text=f"Natural
light bulb {i + 1}: {status[1]}")

        for i, device_label in
enumerate(labels_1["air_conditioner"]):
            for j, sensor_label in
enumerate(labels_2["air_temperature"]):
                mean_value = mean_value +
data['air']['temperature'][j]
                tmp += 1

                if mean_value / tmp <= air_conditioner_off:
                    device_label.config(text=f"Air conditioner {i
+ 1}: {status[1]}")
                elif mean_value / tmp >= air_conditioner_on:

```

```

        device_label.config(text=f"Air conditioner {i
+ 1}: {status[0]}")
    else:
        device_label.config(text=f"Air conditioner {i
+ 1}: {status[1]}")

    mean_value = 0
    tmp = 0

    for i, device_label in
enumerate(labels_1["air_heater"]):
        for j, sensor_label in
enumerate(labels_2["air_temperature"]):
            mean_value = mean_value +
data['air']['temperature'][j]
            tmp += 1
            if mean_value / tmp >= air_heater_off:
                device_label.config(text=f"Air heater {i + 1}:
{status[1]}")
            elif mean_value / tmp <= air_heater_on:
                device_label.config(text=f"Air heater {i + 1}:
{status[0]}")
            else:
                device_label.config(text=f"Air heater {i + 1}:
{status[1]}")

    mean_value = 0
    tmp = 0

    for i, device_label in
enumerate(labels_1["air_humidifier"]):
        for j, sensor_label in
enumerate(labels_2["air_humidity"]):
            mean_value = mean_value +
data['air']['humidity'][j]
            tmp += 1

            if mean_value / tmp >= air_humidifier_off:
                device_label.config(text=f"Air humidifier {i +
1}: {status[1]}")
            elif mean_value / tmp <= air_humidifier_on:
                device_label.config(text=f"Air humidifier {i +
1}: {status[0]}")
            else:
                device_label.config(text=f"Air humidifier {i +
1}: {status[1]}")

    mean_value = 0
    tmp = 0

    for i, device_label in
enumerate(labels_1["air_dehumidifier"]):

```

```

        for j, sensor_label in
enumerate(labels_2["air_humidity"]):
            if (i == 0) and (j == 0):
                if data['air']['humidity'][j] <=
air_dehumidifier_off:
                    device_label.config(text=f"Air
dehumidifier {i + 1}: {status[1]}")
                elif data['air']['humidity'][j] >=
air_dehumidifier_on:
                    device_label.config(text=f"Air
dehumidifier {i + 1}: {status[0]}")
                else:
                    device_label.config(text=f"Air
dehumidifier {i + 1}: {status[1]}")
            elif (i == 1) and (j == 1):
                if data['air']['humidity'][j] <=
air_dehumidifier_off:
                    device_label.config(text=f"Air
dehumidifier {i + 1}: {status[1]}")
                elif data['air']['humidity'][j] >=
air_dehumidifier_on:
                    device_label.config(text=f"Air
dehumidifier {i + 1}: {status[0]}")
                else:
                    device_label.config(text=f"Air
dehumidifier {i + 1}: {status[1]}")

        for i, device_label in
enumerate(labels_1["soil_ph_controller"]):
            for j, sensor_label in
enumerate(labels_2["soil_ph"]):
                if i == j:
                    if data['soil']['ph'][j] >=
soil_ph_controller_off:
                        device_label.config(text=f"Soil pH
controller {i + 1}: {status[1]}")
                    elif data['soil']['ph'][j] <=
soil_ph_controller_on:
                        device_label.config(text=f"Soil pH
controller {i + 1}: {status[0]}")
                    else:
                        device_label.config(text=f"Soil pH
controller {i + 1}: {status[1]}")

        for i, device_label in
enumerate(labels_1["soil_heater"]):
            for j, sensor_label in
enumerate(labels_2["soil_temperature"]):
                if (i <= 7) and (j == 0):
                    if data['soil']['temperature'][j] >=
soil_heater_off:

```

```

        device_label.config(text=f"Soil heater
{i + 1}: {status[1]}")
        elif data['soil']['temperature'][j] <=
soil_heater_on:
        device_label.config(text=f"Soil heater
{i + 1}: {status[0]}")
        else:
            device_label.config(text=f"Soil heater
{i + 1}: {status[1]}")
            elif (i >= 8) and (j == 1):
                if data['soil']['temperature'][j] >=
soil_heater_off:
                device_label.config(text=f"Soil heater
{i + 1}: {status[1]}")
                elif data['soil']['temperature'][j] <=
soil_heater_on:
                device_label.config(text=f"Soil heater
{i + 1}: {status[0]}")
                else:
                    device_label.config(text=f"Soil heater
{i + 1}: {status[1]}")

        for i, device_label in
enumerate(labels_1["soil_irrigator"]):
            for j, sensor_label in
enumerate(labels_2["soil_moisture"]):
                if i == j:
                    if data['soil']['moisture'][j] >=
soil_irrigator_off:
                    device_label.config(text=f"Soil
irrigator {i + 1}: {status[1]}")
                    elif data['soil']['moisture'][j] <=
soil_irrigator_on:
                    device_label.config(text=f"Soil
irrigator {i + 1}: {status[0]}")
                    else:
                        device_label.config(text=f"Soil
irrigator {i + 1}: {status[1]}")

    @staticmethod
    def create_artificial_light_devices_label(frame):
        labels = {
            "artificial_light_bulb": [tk.Label(frame,
text=f"Artificial light bulb {i + 1}: OFF") for i in range(6)]
        }

        row = 0
        for key, label_list in labels.items():
            for label in label_list:
                label.grid(row=row, column=0, sticky="w")
                row += 1

```

```

        return labels

    @staticmethod
    def update_artificial_light_devices_label(labels_1,
labels_2, labels_3, data_1, data_2, data_3):
        status = ["ON", "OFF"]
        average_light_level = 0
        tmp = 0

        for i, sensor_label in
enumerate(labels_2["light_level"]):
            average_light_level = average_light_level +
data_2['light']['level'][i]
            tmp += 1

        for i, sensor_label in
enumerate(labels_3["light_level"]):
            average_light_level = average_light_level +
data_3['light']['level'][i]
            tmp += 1

        if data_1 and (average_light_level / tmp <= 30):
            for i, device_label in
enumerate(labels_1["artificial_light_bulb"]):
                current_status =
device_label.cget("text").split(": ")[1]
                new_status = status[0] if current_status ==
status[1] else status[1]
                device_label.config(text=f"natural_light_bulb
{i + 1}: {new_status}")

```

Лістинг класу граничних параметрів для правил ThresholdManager:

```

import json
import os
import tkinter as tk

class ThresholdManager:
    def __init__(self, app, device_name, threshold_type,
off_threshold_default, on_threshold_default):
        self.app = app
        self.config_file = "../rules_config.json"
        self.device_name = device_name
        self.threshold_type = threshold_type
        self.off_threshold_var = tk.DoubleVar()
        self.on_threshold_var = tk.DoubleVar()
        self.rules_frame = tk.Frame(self.app.root)
        self.off_threshold_default = off_threshold_default
        self.on_threshold_default = on_threshold_default

        self.load_thresholds()

```

```

self.save_button = tk.Button(
    self.rules_frame,
    text="Save",
    command=self.save_thresholds,
    font=("Helvetica", 18),
    width=20
)

self.to_main_menu_button = tk.Button(
    self.rules_frame,
    text="Back",
    command=self.app.show_rules_management_page,
    font=("Helvetica", 18),
    width=20
)

self.create_page()

def show(self):

self.app.rules_management_page.rules_management_frame.pack_for
get()
    self.rules_frame.pack(fill='both', expand=True)

    self.rules_frame.pack_configure(
        padx=(self.app.root.winfo_width() -
self.rules_frame.winfo_reqwidth()) // 2,
        pady=(self.app.root.winfo_height() -
self.rules_frame.winfo_reqheight()) // 3)

    def create_page(self):
        vcmd =
(self.app.root.register(self.validate_threshold), '%P')

        tk.Label(self.rules_frame, text=f"{self.device_name}
on threshold:", font=("Helvetica", 18)).pack(pady=5)
        tk.Entry(self.rules_frame,
textvariable=self.on_threshold_var, font=("Helvetica", 18),
validate="key",
            validatecommand=vcmd).pack(pady=5)

        tk.Label(self.rules_frame, text=f"{self.device_name}
off threshold:", font=("Helvetica", 18)).pack(pady=5)
        tk.Entry(self.rules_frame,
textvariable=self.off_threshold_var, font=("Helvetica", 18),
validate="key",
            validatecommand=vcmd).pack(pady=5)

        self.save_button.pack(pady=20)
        self.to_main_menu_button.pack(pady=20)

def load_thresholds(self):

```

```

    if os.path.exists(self.config_file):
        with open(self.config_file, "r") as file:
            config = json.load(file)
            threshold_type_1, threshold_type_2 =
self.threshold_type.split("_", 1)
            on_threshold = config.get(threshold_type_1,
{}).get(threshold_type_2, {}).get(
                'on', self.on_threshold_default)
            off_threshold = config.get(threshold_type_1,
{}).get(threshold_type_2, {}).get(
                'off', self.off_threshold_default)
        else:
            off_threshold = self.off_threshold_default
            on_threshold = self.on_threshold_default

        self.off_threshold_var.set(off_threshold)
        self.on_threshold_var.set(on_threshold)

    def save_thresholds(self):
        threshold_type_1, threshold_type_2 =
self.threshold_type.split("_", 1)

        if os.path.exists(self.config_file):
            with open(self.config_file, "r") as file:
                config = json.load(file)
        else:
            config = {}

        config.setdefault(threshold_type_1,
{}).setdefault(threshold_type_2, {})[ 'on' ] =
self.on_threshold_var.get()
        config.setdefault(threshold_type_1,
{}).setdefault(threshold_type_2, {})[ 'off' ] =
self.off_threshold_var.get()

        with open(self.config_file, "w") as file:
            json.dump(config, file)

        # Send the updated config to the server
        self.app.network.send_rules_config()

    @staticmethod
    def validate_threshold(value):
        try:
            val = float(value)
            return -50 <= val <= 100
        except ValueError:
            return False

class NaturalLightBulbManager(ThresholdManager):
    def __init__(self, app):

```

```

        super().__init__(app, "Natural light bulbs",
"light_level",
                    55.0, 20.0)

```

```

class AirConditioner(ThresholdManager):
    def __init__(self, app):
        super().__init__(app, "Air conditioners",
"air_temperature_c",
                    19.0, 25.0)

```

```

class AirHeaterManager(ThresholdManager):
    def __init__(self, app):
        super().__init__(app, "Air heaters",
"air_temperature_h",
                    25.0, 19.0)

```

```

class AirHumidifierManager(ThresholdManager):
    def __init__(self, app):
        super().__init__(app, "Air humidifiers",
"air_humidity_h",
                    65.0, 50.0)

```

```

class AirDehumidifierManager(ThresholdManager):
    def __init__(self, app):
        super().__init__(app, "Air dehumidifiers",
"air_humidity_d",
                    50.0, 75.0)

```

```

class SoilPhControllerManager(ThresholdManager):
    def __init__(self, app):
        super().__init__(app, "Soil pH controllers",
"soil_ph",
                    6.5, 5.5)

```

```

class SoilHeaterManager(ThresholdManager):
    def __init__(self, app):
        super().__init__(app, "Soil heaters",
"soil_temperature",
                    24.0, 18.0)

```

```

class IrrigatorManager(ThresholdManager):
    def __init__(self, app):
        super().__init__(app, "Soil irrigators",
"soil_moisture",
                    60.0, 25.0)

```