

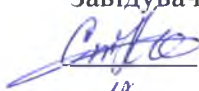
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет робототехніки та приладобудування

Кафедра комп'ютерно-інтегрованих технологій виробництва приладів

До захисту допущено:

Завідувач кафедри

 **Наталія СТЕЛЬМАХ**

«12» 06 2026 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Комп'ютерно-інтегровані системи
та технології в приладобудуванні»**

**спеціальності 151 «Автоматизація та комп'ютерно-інтегровані
технології»**

на тему: «Автоматизована система керування мікрокліматом теплиці»

Виконала:

студентка IV курсу, групи ПБ-321
Шепелюк Катерина Олександрівна



Керівник:

Професор, доктор технічних наук, професор,
Антонюк Віктор Степанович



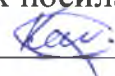
Рецензент:

Професор, доктор технічних наук, професор,
Чиж Ігор Генріхович



Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студентка



Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет робототехніки та приладобудування
Кафедра комп'ютерно-інтегрованих технологій виробництва приладів

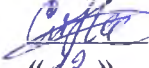
Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютерно-інтегровані системи та технології в приладобудуванні»

ЗАТВЕРДЖУЮ

Завідувач кафедри

 **Наталія СТЕЛЬМАХ**
«12» 06 2026 р.

ЗАВДАННЯ
на дипломну роботу студенту
Шепелюк Катерина Олександрівна

1. Тема роботи «Автоматизована система контролю розмірів деталі на верстатах ЧПК», керівник роботи Антонюк Віктор Степанович, професор, д.т.н, вчене професор, затверджені наказом по університету від «25» травня 2026 р. №1888-с

2. Термін подання студентом роботи 07.06.2026

3. Вихідні дані до роботи: нормативні діапазони параметрів мікроклімату: $T = 16...28\text{ }^{\circ}\text{C}$, $E = 100...1800\text{ мкмоль/(м}^2\cdot\text{с)}$, $\phi = 55...90\text{ \%}$, $T_{\text{зовн}} = -20...+40\text{ }^{\circ}\text{C}$; обрані технічні засоби — SHT35-DIS-F, SQ-521-SS, ESP32, релейний модуль.

4. РОЗДІЛ 1. Аналіз мікроклімату теплиці та засобів його автоматизованого керування. 1.1 Теплиця як об'єкт керування та параметри її мікроклімату. 1.2 Закони регулювання та підходи до автоматизації мікроклімату. 1.3 Засоби вимірювання та виконавчі механізми. РОЗДІЛ 2. Параметри мікроклімату, засоби вимірювання та нормативні вимоги до системи. 2.1 Параметри мікроклімату теплиці як об'єкти регулювання. 2.2 Температурний режим: денна і нічна уставки. 2.3 Світловий режим і фотоперіод. 2.4 Режим вологості повітря. 2.5 Засоби вимірювання та виконавчі механізми. 2.6 Нормативні діапазони та параметри регулювання. 2.7 Принцип двопозиційного керування. РОЗДІЛ 3. Проєктування автоматизованої системи керування. 3.1 Структурна схема системи. 3.2 Принцип двопозиційного регулювання з гістерезисом. 3.3 Функціональна схема автоматизації (ФСА). 3.4 Схема

інформаційних потоків (DFD). 3.5 Об'єктно-орієнтована модель системи (UML). 3.6 Координація контурів керування. РОЗДІЛ 4. Алгоритмічне та програмне забезпечення системи. 4.1 Головний алгоритм функціонування системи. 4.2 Алгоритм підпрограми СК₁ (керування температурою). 4.3 Алгоритм підпрограми СК₂ (керування освітленістю). 4.4 Алгоритм підпрограми СК₃ (керування вологістю). 4.5 Алгоритм обробки аварій і безпечного стану. 4.6 Програмна реалізація системи.

ВИСНОВКИ

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо) структурна схема, функціональна схема автоматизації, схема інформаційних потоків, UML-діаграми, схема головного алгоритму, схеми алгоритмів підпрограм СК₁–СК₃.

6. Дата видачі завдання 03.03.2026

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз літературних джерел і постановка задачі досліджень	24.04.2026	
2	Постановка задачі дипломної роботи	02.05.2026	
3	Методика контролю деталей на верстатах з ЧПК	08.05.2026	
4	Структурно-функціональна схема автоматизованої системи контролю	12.05.2026	
5	Математична модель визначення геометричних параметрів деталі	14.05.2026	
6	Розробка автоматизованої системи контролю	19.05.2026	
7	Алгоритм системи автоматизованої контролю	22.05.2026	
8	Програмне забезпечення автоматизованої системи контролю	29.05.2026	
9	Інструкція з використання програми	02.06.2026	
10	Оформлення дипломної роботи	05.06.2026	

Студент

Катерина ШЕПЕЛЮК

Керівник

Віктор АНТОНЮК

ЗМІСТ

АНОТАЦІЯ

ABSTRACT

ВСТУП

РОЗДІЛ 1. АНАЛІЗ МІКРОКЛІМАТУ ТЕПЛИЦІ ТА ЗАСОБІВ ЙОГО АВТОМАТИЗОВАНОГО КЕРУВАННЯ

- 1.1. Теплиця як об'єкт керування та параметри її мікроклімату
- 1.2. Закони регулювання та підходи до автоматизації мікроклімату
- 1.3. Засоби вимірювання та виконавчі механізми

Висновки до розділу 1

РОЗДІЛ 2. ПАРАМЕТРИ МІКРОКЛІМАТУ, ЗАСОБИ ВИМІРЮВАННЯ ТА НОРМАТИВНІ ВИМОГИ ДО СИСТЕМИ

- 2.1. Параметри мікроклімату теплиці як об'єкти регулювання
- 2.2. Температурний режим: денна і нічна уставки
- 2.3. Світловий режим і фотоперіод
- 2.4. Режим вологості повітря
- 2.5. Засоби вимірювання та виконавчі механізми
- 2.6. Нормативні діапазони та параметри регулювання
- 2.7. Принцип двопозиційного керування

Висновки до розділу 2

РОЗДІЛ 3. ПРОЄКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ

- 3.1. Структурна схема системи
- 3.2. Принцип двопозиційного регулювання з гістерезисом
- 3.3. Функціональна схема автоматизації (ФСА)
- 3.4. Схема інформаційних потоків (DFD)
- 3.5. Об'єктно-орієнтована модель системи (UML)
- 3.6. Координація контурів керування

Висновки до розділу 3

РОЗДІЛ 4. АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

- 4.1. Головний алгоритм функціонування системи
- 4.2. Алгоритм підпрограми СК₁ (керування температурою)
- 4.3. Алгоритм підпрограми СК₂ (керування освітленістю)
- 4.4. Алгоритм підпрограми СК₃ (керування вологістю)
- 4.5. Алгоритм обробки аварій і безпечного стану
- 4.6. Програмна реалізація системи

Висновки до розділу 4

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А. Ілюстративний матеріал (плакати формату А1)

ДОДАТОК Б. Специфікація приладів і засобів автоматизації до функціональної схеми

ДОДАТОК В. Лістинг програмного коду

АНОТАЦІЯ

Дипломний робота присвячений розробленню автоматизованої системи керування мікрокліматом теплиці у вигляді програмної моделі, яка підтримує температуру, освітленість і вологість повітря в заданих межах за допомогою двопозиційного регулювання та узгоджує роботу виконавчих механізмів.

У роботі проаналізовано теплицю як об'єкт автоматизованого керування, розглянуто параметри її мікроклімату та сучасні засоби й методи їх регулювання. Обґрунтовано доцільність застосування двопозиційного (релейного) закону регулювання з огляду на інерційність об'єкта та дискретний характер виконавчих механізмів. Визначено параметри регулювання трьох контурів - температури, освітленості та вологості, - для кожного з яких задано робочий діапазон, початкові цільові значення (уставки) та гістерезис. Освітленість оцінено через фотосинтетично активну радіацію як фізично коректну для рослин величину.

Для прив'язки моделі до реальних умов обрано конкретні технічні засоби – цифрові давачі температури, вологості й освітленості, мікроконтролер та релейний модуль, - характеристики яких покладено в основу параметрів моделі. Розроблено структуру системи, функціональну схему автоматизації, схему інформаційних потоків і UML-діаграми, а також алгоритмічне забезпечення у вигляді головного алгоритму та підпрограм керування контурами. Систему реалізовано у вигляді програмної моделі мовою Python із графічним інтерфейсом оператора.

Результати роботи можуть стати основою для створення реальних недорогих систем автоматизації теплиць, а також як навчальний засіб для вивчення принципів двопозиційного регулювання.

КЛЮЧОВІ СЛОВА: мікроклімат теплиці, автоматизоване керування, двопозиційне регулювання, гістерезис, координація контурів, програмна модель.

ABSTRACT

The aim of the work is the development of an automated greenhouse microclimate control system in the form of a software model, which maintains air temperature, illuminance and humidity within set limits by means of two-position control and coordinates the operation of the actuators.

The thesis analyses the greenhouse as an object of automated control and examines its microclimate parameters together with modern means and methods of their regulation. The expediency of applying the two-position (relay) control law is substantiated, given the inertia of the object and the discrete nature of the actuators. The control parameters of three loops - temperature, illuminance and humidity - are defined, each with its operating range, initial target values (setpoints) and hysteresis. Illuminance is evaluated through photosynthetically active radiation as the physically correct quantity for plants.

To relate the model to real conditions, specific technical means were selected - digital sensors of temperature, humidity and illuminance, a microcontroller and a relay module - whose characteristics form the basis of the model parameters. The system structure, the functional automation diagram, the data flow diagram and UML diagrams were developed, as well as the algorithmic support in the form of the main algorithm and the loop control subroutines. The system is implemented as a software model in the Python language with a graphical operator interface.

The results of the work can serve as a basis for creating real low-cost greenhouse automation systems, as well as a teaching aid for studying the principles of two-position control.

KEYWORDS: greenhouse microclimate, automated control, two-position control, hysteresis, control loop coordination, software model.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ

АЦП – аналого-цифровий перетворювач;

ВМ – виконавчий механізм;

ОК – об'єкт керування;

ПІД – пропорційно-інтегрально-диференціальний (закон регулювання);

ПЛК – програмований логічний контролер;

ФСА – функціональна схема автоматизації;

DFD – Data Flow Diagram;

PAR – Photosynthetically Active Radiation;

UML – уніфікатор Unified Modeling Language;

I²C – послідовний інтерфейс передавання даних;

SDI-12 – Serial Data Interface;

USB – Universal Serial Bus;

СК₁, СК₂, СК₃ – контури керування температурою, освітленістю та вологістю.

ВСТУП

Тепличне рослинництво є одним із найважливіших напрямів сучасного сільського господарства, оскільки дає змогу вирощувати продукцію незалежно від пори року та погодних умов і отримувати стабільно високі врожаї на обмеженій площі. На відміну від відкритого ґрунту, закрите середовище теплиці уможливорює штучне підтримання параметрів повітря - температури, освітленості та вологості - у межах, сприятливих для конкретної культури. Саме від точності підтримання цих параметрів значною мірою залежать урожайність, якість продукції та витрати енергії на одиницю врожаю.

Підтримання мікроклімату – ресурсомісткий процес: значна частина експлуатаційних витрат теплиці припадає на роботу систем опалення та інших виконавчих механізмів [Kokieva]. Тому стабільне й узгоджене керування цими механізмами є важливою практичною задачею. Традиційно мікрокліматом теплиць керують вручну або за допомогою найпростіших засобів автоматики, що не забезпечує стабільності параметрів і потребує постійної участі людини. Ручне керування неминуче супроводжується запізненням реакції на відхилення, суб'єктивністю оцінок та перевитратами ресурсів, а використання окремих незалежних регуляторів без узгодження їхньої роботи призводить до того, що різні виконавчі механізми діють усупереч один одному. Через це актуальною є задача створення автоматизованої системи, яка підтримувала б параметри мікроклімату в заданих межах без постійного втручання оператора й узгоджувала роботу виконавчих механізмів між собою.

Параметри мікроклімату взаємопов'язані, тож зміна одного впливає на інші, що потребує системного підходу до керування [Суворін]. Водночас для підтримання основних параметрів теплиці найчастіше застосовують двопозиційне (релейне) регулювання [Kokieva], яке вирізняється простотою реалізації та надійністю. Розвиток доступної цифрової елементної бази - недорогих мікроконтролерів і цифрових датчиків - робить таку автоматизацію

економічно виправданою навіть для невеликих господарств і дає змогу будувати системи керування на основі простих і надійних законів регулювання, не вдаючись до дорогого промислового обладнання. Перевірку таких рішень доцільно проводити спершу на програмній моделі, що дає змогу відпрацювати алгоритми керування без витрат на фізичне обладнання.

Метою роботи є розроблення автоматизованої системи керування мікрокліматом теплиці, реалізованої у вигляді програмної моделі, яка підтримує температуру, освітленість і вологість повітря в заданих межах за допомогою двопозиційного регулювання та узгоджує роботу виконавчих механізмів.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати параметри мікроклімату теплиці та сучасні засоби й методи їх автоматизованого регулювання;
- обґрунтувати вибір законів регулювання, складу давачів і виконавчих механізмів системи;
- розробити алгоритми двопозиційного керування для контурів температури, освітленості та вологості;
- реалізувати систему у вигляді програмної моделі, що відтворює роботу всіх контурів керування.

Об'єктом дослідження є процес автоматизованого контролю мікроклімату теплиці.

Предметом дослідження є методи й засоби автоматизованого двопозиційного регулювання параметрів мікроклімату теплиці на основі програмної моделі.

Методи дослідження. Для вирішення поставленої задачі застосовано аналіз і систематизацію науково-технічної літератури [Антонюк] та комп'ютерне моделювання роботи системи [Бурау]. Програмну модель реалізовано мовою Python із графічним інтерфейсом на основі бібліотеки Tkinter.

Наукова новизна полягає в розробленні структури та алгоритмів двопозиційного керування трьома взаємопов'язаними параметрами мікроклімату теплиці - температурою, освітленістю та вологістю - з координацією контурів, яка узгоджує роботу виконавчих механізмів.

Практичне значення одержаних результатів полягає у можливості впровадження автоматизованих систем контролю мікроклімату теплиць на основі сучасних мікроконтролерів і датчиків. Склад апаратної частини моделі узгоджено з реально існуючим обладнанням, що дає змогу за потреби перенести систему на фізичні засоби без зміни її логіки.

РОЗДІЛ 1. АНАЛІЗ МІКРОКЛІМАТУ ТЕПЛИЦІ ТА ЗАСОБІВ ЇЇ АВТОМАТИЗОВАНОГО КЕРУВАННЯ

1.1. Теплиця як об'єкт керування та параметри її мікроклімату

Теплиця є штучно створеним замкненим середовищем, призначеним для вирощування рослин в умовах, що не залежать від зовнішнього клімату. Теплиці пройшли тривалий шлях розвитку - від найпростіших плівкових укриттів, що лише захищають рослини від холоду, до сучасних промислових комплексів із повністю автоматизованим керуванням усіма параметрами середовища. У спеціальній літературі виділяють до семи поколінь теплиць - від простих плівкових конструкцій до інтелектуальних систем зі штучним інтелектом, характерних для голландського та ізраїльського тепличного господарства [Суворін]. Незалежно від покоління, спільним для всіх теплиць є те, що внутрішні умови не залежать від погоди повністю, а можуть підтримуватися штучно - і саме це робить теплицю об'єктом керування.

Як об'єкт керування теплиця характеризується сукупністю фізичних параметрів середовища, що впливають на життєдіяльність рослин. До основних параметрів мікроклімату належать температура повітря, відносна вологість, концентрація вуглекислого газу, освітленість (зокрема її фотосинтетично активна складова), а також, у системах із поливом, кислотність поживного розчину [Суворін]. Кожен із цих параметрів впливає на перебіг фотосинтезу, дихання й транспірацію рослин, а отже - на їхній ріст, урожайність та якість продукції.

Температура повітря визначає швидкість перебігу усіх біохімічних процесів у рослині. Для більшості тепличних культур потреба в температурі залежить від фази доби - удень, під час активного фотосинтезу, рослині потрібна вища температура, ніж уночі, що відповідає природному добовому ритму.

Відносна вологість повітря регулює інтенсивність транспірації - випаровування води листям. Як надто низька, так і надто висока вологість шкідливі для рослин, тому її потрібно підтримувати в певному діапазоні.

Інколи рівень вологості характеризують через дефіцит тиску водяної пари (VPD), оптимальні значення якого для тепличних культур лежать у межах приблизно 0,8-1,5 кПа [Суворін].

Освітленість для рослин принципово відрізняється від звичайного освітлення приміщень. Для фотосинтезу значення має не яскравість світла, а кількість фотонів у діапазоні довжин хвиль 400-700 нм - так звана фотосинтетично активна радіація (PAR), яку кількісно характеризують щільністю потоку фотонів у мікромолях на квадратний метр за секунду ($\mu\text{mol}/\text{m}^2/\text{s}$) [McCree]. Саме ця величина, а не освітленість у люксах, є фізично коректною мірою освітленості для рослинництва. Окрім рівня освітленості, для рослин важливий і фотоперіод - чергування світлої та темної фаз доби, оскільки для нормального розвитку рослинам потрібен повноцінний темновий період.

Концентрація вуглекислого газу безпосередньо впливає на інтенсивність фотосинтезу, оскільки CO_2 є його сировиною; у промислових теплицях її нерідко штучно підвищують для збільшення врожайності. Проте керування концентрацією CO_2 потребує окремого комплексу обладнання й у більшості простіших систем не реалізується.

Істотною особливістю мікроклімату теплиці є взаємозв'язок його параметрів: зміна одного параметра може впливати на інші, тому керування ними не можна розглядати як набір цілком незалежних задач [Суворін]. Наприклад, провітрювання, що знижує температуру, одночасно знижує й вологість повітря, а підвищення температури впливає на відносну вологість навіть за незмінної кількості вологи в повітрі. Через це для якісного керування мікрокліматом потрібен системний підхід, що враховує взаємний вплив параметрів і узгоджує роботу виконавчих механізмів.

Сукупність розглянутих параметрів і їхній взаємозв'язок визначають теплицю як складний багатопараметричний об'єкт керування. На практиці набір параметрів, що підлягають автоматичному регулюванню, обирають залежно від призначення й рівня системи: прості системи обмежуються

підтриманням температури, тоді як промислові комплекси керують усім переліком параметрів, включно з CO₂ та поливом. Температура, вологість і освітленість при цьому є базовим набором параметрів повітряного середовища, що піддаються прямому керуванню за допомогою порівняно простих виконавчих механізмів і вимірюються доступними засобами, тому саме вони найчастіше становлять основу систем автоматизації теплиць.

1.2. Закони регулювання та підходи до автоматизації мікроклімату

Керування будь-яким параметром мікроклімату зводиться до того, що система порівнює його виміряне значення із заданим (уставкою) і за наявності відхилення впливає на нього через виконавчий механізм. Те, за яким правилом формується цей вплив, визначається законом регулювання. Від вибору закону регулювання залежать як якість підтримання параметра, так і складність та вартість системи, тому його обґрунтування є одним із ключових питань проєктування.

У теорії автоматичного керування закони регулювання поділяють на дві великі групи - неперервні та позиційні (релейні) [Попович-Ковальчук]. За неперервного закону керувальний вплив змінюється плавно й пропорційно відхиленню параметра; найпоширенішим його різновидом є пропорційно-інтегрально-диференціальний (ПІД) регулятор, що формує вплив як суму трьох складових - пропорційної до відхилення, інтегральної (накопиченого відхилення) та диференціальної (швидкості його зміни). ПІД-регулювання забезпечує високу точність підтримання параметра й плавність роботи, проте потребує виконавчих механізмів із плавним керуванням (регульованих нагрівачів, приводів змінної потужності), точного налаштування коефіцієнтів та достатньо точної математичної моделі об'єкта.

За позиційного закону керувальний вплив набуває лише кількох фіксованих значень. Найпростішим і найпоширенішим є двопозиційне (релейне) регулювання, за якого виконавчий механізм має лише два стани - «увімкнено» і «вимкнено». Регулятор вмикає механізм при відхиленні

параметра в один бік і вимикає при відхиленні в інший, утримуючи параметр у певному діапазоні навколо уставки. Двопозиційне регулювання поступається неперервному в точності, оскільки параметр не стабілізується в одній точці, а коливається в межах зони нечутливості, проте має суттєві переваги: простоту реалізації, низьку вартість, надійність і відсутність потреби в точному налаштуванні чи в моделі об'єкта.

Вибір закону регулювання визначається властивостями об'єкта керування. Температура й вологість повітря в теплиці змінюються повільно через велику інерційність повітряного середовища. Для такого повільного об'єкта точність і швидкодія неперервного регулювання надмірні: достатньо утримувати параметр у вузькому діапазоні навколо уставки, і коливання в цих межах для рослин неістотні. Тому для теплиці природним вибором є двопозиційне регулювання [Kokieva].

Окрім закону регулювання, системи автоматизації теплиць різняться технічним рівнем реалізації - від промислових інтегрованих комплексів до простих рішень на базі мікроконтролерів. На верхньому рівні представлені промислові системи світових виробників, такі як Priva Connex (Нідерланди) та Argus TITAN (Канада). Це інтегровані комплекси, що керують усіма параметрами теплиці - температурою, вологістю, CO₂, освітленням, поливом та енергоспоживанням - і використовують складні алгоритми з прогнозуванням умов на основі математичних моделей та хмарного моніторингу [Priva; Argus]. Вони забезпечують максимальну якість керування, проте мають високу вартість (від кількох тисяч доларів), що робить їх доцільними лише для великих промислових господарств.

Протилежний полюс становлять рішення на базі доступних мікроконтролерів (Arduino, ESP32) із простими пороговими (двопозиційними) алгоритмами. Вони керують базовим набором параметрів - температурою, вологістю, освітленістю - мають низьку вартість (десятки доларів), будуються переважно на відкритих програмних та апаратних компонентах і доступні навіть невеликим господарствам [Ardiansah]. Хоч і

поступаючись промисловим системам у функціональності та якості прогнозування, такі рішення цілком задовольняють потреби автоматизації невеликих теплиць.

Окремий інтерес становлять дослідницькі розробки вітчизняних науковців. Зокрема, у дослідженнях науковців КПІ ім. Ігоря Сікорського досліджувалося автоматизоване керування мікрокліматом виробничих приміщень для складання прецизійних приладів, де ключовими параметрами є температура, вологість та чистота повітря [Мережаний; Антонюк та ін.]. Цей доробок показує, що задача автоматизованого керування мікрокліматом приміщення має на кафедрі усталену наукову традицію. Теплиця близька до таких приміщень за самою постановкою задачі, але відрізняється тим, що до температури й вологості тут додається освітленість - критичний для рослин параметр. Прикладом сучасної вітчизняної розробки саме для теплиць є мехатронна система керування температурним режимом мікроклімату на основі аналізу теплового випромінювання споруди, реалізована науковцями КПІ ім. Ігоря Сікорського [Губарєв]. Узагальнене порівняння розглянутих систем за основними характеристиками наведено в таблиці 1.1.

Таблиця 1.1 - Порівняння систем автоматизації мікроклімату теплиць

Критерій	Priva Connex	Argus Titan	Система Синіциної-Губарєва (КПІ)	Типове DIY-рішення Arduino/ESP32
Параметри керування	Температура, вологість, CO ₂ , PAR, полив, енергія	Температура, вологість, CO ₂ , PAR, полив, фертигація	Температура	Температура, вологість, освітленість
Тип алгоритму	Інтегроване керування + AI-прогнозування	Алгоритмічне прогнозування + моделі	Керування нагрівом на основі теплової моделі	Двопозиційний (пороговий)
Тип інтерфейсу	Web + Mobile (Priva Operator), хмарний моніторинг 24/7	Web (TITAN Envoy) + Mobile (Argus Live)	Промисловий контролер (SIEMENS S7-1200 / Schneider)	Web (Blynk, ThingSpeak, локальний HTTP-сервер)
Вартість	від \$10 000	від \$5 000	-	\$50-150
Відкритий код	Ні	Ні	Ні	Так

Як видно з порівняння систем автоматизації мікроклімату теплиць - промислові системи забезпечують найширшу функціональність, але є дорогими й закритими, тоді як рішення на мікроконтролерах доступні,

відкриті й достатні для базового керування мікрокліматом. Для задачі цієї роботи - створення доступної системи керування мікрокліматом невеликої теплиці - раціональним є саме мікроконтролерний підхід із двопозиційним законом регулювання, який поєднує простоту, надійність і низьку вартість за прийнятною якістю підтримання параметрів.

1.3. Засоби вимірювання та виконавчі механізми

Будь-яка автоматизована система керування мікрокліматом складається з кількох груп технічних засобів, зокрема: датчиків, що вимірюють параметри середовища, і виконавчих механізмів, що впливають на ці параметри. Від характеристик цих засобів безпосередньо залежить якість роботи системи, тому їх вибір є важливим етапом проектування.

Засоби вимірювання. Датчики параметрів мікроклімату поділяють за вимірюваною величиною (температура, вологість, освітленість) та за типом вихідного сигналу - аналогові й цифрові. Аналоговий датчик видає неперервний сигнал (зазвичай напругу), пропорційний вимірюваній величині, який потребує перетворення в цифрову форму за допомогою аналого-цифрового перетворювача. Цифровий датчик містить вбудований перетворювач і передає вже готове числове значення за цифровим інтерфейсом. Це спрощує під'єднання до мікроконтролера й підвищує завадостійкість, оскільки числовий сигнал, на відміну від аналогового, не спотворюється завадами в лінії зв'язку.

Для вимірювання температури в недорогих системах автоматизації поширені такі датчики, як аналоговий LM35 та цифровий DS18B20. LM35 простий і дешевий, але як аналоговий датчик чутливий до завад і потребує АЦП [Ardiansah]. DS18B20 передає дані цифровою шиною 1-Wire, точніший за аналогові датчики й випускається у вологозахищеному корпусі, тому придатний і для вимірювань поза приміщенням. Для одночасного вимірювання температури й вологості часто застосовують комбіновані датчики - наприклад, DHT22, що вимірює обидва параметри в одному корпусі, або

точніші цифрові давачі серій SHT (Sensirion) з інтерфейсом I²C. Порівняльні характеристики поширених давачів температури й вологості наведено в таблиці 1.2.

Таблиця 1.2 - Порівняння поширених давачів температури та вологості

Давач	Вимірювані величини	Тип сигналу / інтерфейс	Точність (температура)	Особливості
LM35	температура	аналоговий	±0,5 °C	простий, дешевий, потребує АЦП
DS18B20	температура	цифровий, 1-Wire	±0,5 °C	вологозахисний корпус, придатний для вулиці
DHT22	температура, вологість	цифровий	±0,5 °C	два параметри в одному корпусі, невисока точність
SHT35	температура, вологість	цифровий, I ² C	±0,1 °C	два параметри в одному корпусі, висока точність

Для вимірювання освітленості потрібно враховувати її особливість для рослинництва. Звичайні давачі освітленості (фоторезистори, люксметри) розраховані на чутливість людського ока й вимірюють освітленість у люксах, що не відповідає потребам рослин. Для вимірювання саме фотосинтетично активної радіації застосовують спеціальні квантові давачі (PAR-давачі), які вимірюють щільність потоку фотонів у діапазоні 400-700 нм безпосередньо в $\mu\text{mol}/\text{m}^2/\text{s}$. Такі давачі точніші й дорожчі за побутові, проте лише вони вимірюють світло так, як його насправді сприймають рослини.

Виконавчі механізми. Для впливу на параметри мікроклімату використовують виконавчі механізми, кожен з яких діє на певний параметр. Температуру підвищують нагрівачами, а знижують провітрюванням - відкриванням вентиляційних кватирок або вмиканням вентиляторів. Освітленість підвищують додатковим штучним досвічуванням за допомогою фітоламп - світильників зі спектром, придатним для фотосинтезу. Вологість підвищують зволожувачами, зокрема системами туманоутворення, а знижують тим самим провітрюванням, що й температуру. У відомих реалізаціях такі механізми - нагрівач, нагнітач повітря, насос

туманоутворення - керуються від мікроконтролера через комутаційні елементи [Ardiansah].

Більшість цих механізмів за характером дії дискретні - вони працюють у режимі «увімкнено/вимкнено», що узгоджується з двопозиційним законом регулювання, розглянутим у попередньому пункті. Оскільки мікроконтролер не може комутувати потужне мережеве навантаження безпосередньо, для керування виконавчими механізмами використовують проміжні комутаційні елементи - електромагнітні реле, зазвичай у складі релейних модулів з гальванічною опторозв'язкою, що захищає керувальну частину системи від силової.

Підхід до керування. Об'єднує давачі й виконавчі механізми в єдину систему керувальний пристрій. У промислових системах це програмовані логічні контролери (ПЛК), а в доступних рішеннях - мікроконтролери (Arduino, ESP32 тощо), що опитують давачі, обробляють їхні показання за заданим алгоритмом і керують виконавчими механізмами [Ardiansah]. Сучасні мікроконтролери мають вбудовані цифрові інтерфейси для під'єднання давачів і достатню кількість виходів для керування виконавчими механізмами, що робить їх зручною основою для систем автоматизації невеликих теплиць.

Висновки до розділу 1

У розділі проаналізовано теплицю як об'єкт автоматизованого керування, розглянуто закони регулювання та підходи до автоматизації мікроклімату, а також засоби вимірювання й виконавчі механізми, що використовуються в таких системах.

Установлено, що теплиця є складним багатопараметричним об'єктом керування, основними параметрами мікроклімату якого є температура, вологість, освітленість та концентрація вуглекислого газу. Показано, що ці параметри взаємопов'язані, тому керування ними потребує системного підходу з узгодженням роботи виконавчих механізмів. Обґрунтовано, що

базовий набір параметрів повітряного середовища - температуру, вологість і освітленість - можна регулювати порівняно простими засобами, і саме він найчастіше становить основу систем автоматизації теплиць. Окремо показано, що освітленість для рослин слід оцінювати не в люксах, а через фотосинтетично активну радіацію (PAR).

Розглянуто неперервні та позиційні закони регулювання й показано, що для теплиці, якій притаманні інерційність об'єкта та дискретний характер виконавчих механізмів, доцільним є двопозиційне (релейне) регулювання, що поєднує простоту й надійність із прийнятною для рослин якістю підтримання параметрів. Порівняння наявних систем автоматизації - від промислових комплексів (Priva, Argus) до доступних рішень на мікроконтролерах - показало, що для невеликої теплиці раціональним є мікроконтролерний підхід, який забезпечує потрібну функціональність за низької вартості.

Розглянуто засоби вимірювання та виконавчі механізми й обґрунтовано перевагу цифрових давачів над аналоговими (простота під'єднання та завадостійкість). Показано, що для комутації виконавчих механізмів у режимі «увімкнено/вимкнено» зручно використовувати релейні модулі з опторозв'язкою під керуванням мікроконтролера, що узгоджується з двопозиційним законом регулювання.

Таким чином, за результатами аналізу обґрунтовано загальний підхід до побудови системи: двопозиційне керування трьома параметрами мікроклімату (температурою, освітленістю та вологістю) на основі цифрових давачів і мікроконтролера. Це визначає постановку задачі для подальших розділів - обґрунтування конкретних параметрів регулювання та засобів вимірювання (розділ 2) і розроблення алгоритмічного й програмного забезпечення системи (розділи 3-5).

РОЗДІЛ 2. ПАРАМЕТРИ МІКРОКЛІМАТУ, ЗАСОБИ ВИМІРЮВАННЯ ТА НОРМАТИВНІ ВИМОГИ ДО СИСТЕМИ

2.1. Параметри мікроклімату теплиці як об'єкти регулювання

Мікроклімат теплиці - це сукупність фізичних параметрів повітряного середовища в зоні вирощування рослин, які безпосередньо впливають на перебіг фотосинтезу та загального розвитку культур. На відміну від відкритого ґрунту, де ці параметри визначаються лише погодними умовами, закрите середовище теплиці дає змогу штучно підтримувати їх у межах, сприятливих для конкретної культури. Підтримання мікроклімату в заданих межах і є головним завданням системи автоматизації.

У межах розробленої системи об'єктами регулювання виступають три фізичні параметри: температура повітря, освітленість та відносна вологість повітря. Вибір саме цих параметрів зумовлений тим, що вони найсильніше впливають на стан рослин, піддаються прямому керуванню за допомогою порівняно простих виконавчих механізмів і можуть бути виміряні стандартними цифровими датчиками. Інші чинники - концентрація вуглекислого газу, склад поживного розчину, температура ґрунту - у роботі не розглядаються, оскільки потребують окремого комплексу обладнання й не належать до базового набору параметрів повітряного середовища.

Кожному з трьох параметрів відповідає окремий контур керування. Контур температури (СК₁) є найскладнішим, оскільки містить два виконавчі органи протилежної дії - нагрівач, що підвищує температуру, та вентиляційну кватирку, що її знижує. Контур освітленості (СК₂) керує фітолампюю як єдиним виконавчим органом і працює лише в межах світлової фази доби. Контур вологості (СК₃) керує зволожувачем повітря. Така структура з трьох окремих контурів дає змогу регулювати кожен параметр незалежно від інших.

Параметри відрізняються за швидкістю реакції на керувальний вплив. Температура й вологість повітря є інерційними: після вмикання чи вимикання виконавчого механізму вони змінюються поступово, а не

стрибком. Освітленість, навпаки, змінюється практично миттєво - рівень фотосинтетично активної радіації підвищується або спадає одразу після вмикання чи вимикання фітолампи. Цю відмінність у динаміці об'єктів регулювання враховано при налаштуванні параметрів кожного контуру. Нижче кожен параметр розглянуто окремо з обґрунтуванням його робочого діапазону, уставок та особливостей режиму керування, а вибір конкретних засобів вимірювання винесено в підрозділ 2.5.

2.2. Температурний режим: денна і нічна уставки

Температура повітря є найважливішим параметром мікроклімату, оскільки визначає інтенсивність усіх біохімічних процесів у рослині. Відхилення температури від оптимального діапазону в той чи інший бік однаково шкідливе: за надмірно низьких значень уповільнюється розвиток рослини, тоді як перегрів спричиняє в'янення та незворотні пошкодження тканин. Тому температура потребує регулювання у двох напрямках - як на підвищення, так і на зниження.

Характерною особливістю температурного режиму теплиці є його залежність від фази доби. Удень, коли відбувається активний фотосинтез, рослині потрібна вища температура; уночі, коли фотосинтез припиняється, температуру доцільно знижувати, що відповідає природному добовому ритму рослин. З огляду на це в системі передбачено дві окремі уставки температури - денну і нічну. За замовчуванням денну уставку прийнято на рівні 22 °С, нічну - зниженою до 17 °С. Обидва значення обрані у межах оптимальних діапазонів, рекомендованих для тепличних культур [Суворін] (наведених у таблиці 2.1), причому нижча нічна уставка відповідає природному добовому ритму рослин, за якого температуру вночі знижують на кілька градусів. Перемикання між ними відбувається автоматично за сигналом таймера відповідно до заданих меж світлової фази доби.

Оскільки різні культури потребують різного температурного режиму, оператор може змінювати уставки в межах робочого діапазону від 16 до 28

°C, задаючи їх відповідно до вирощуваної культури та фази її розвитку. Ці межі охоплюють потреби більшості тепличних культур помірного клімату й водночас запобігають введенню завідомо неприйнятних значень.

Навколо кожної уставки система формує зону нечутливості завширшки в гістерезис $\Delta = 2$ °C, у межах якої стан виконавчих механізмів не змінюється. Це означає, що за денної уставки 22 °C нагрівач вмикається, коли температура опускається нижче 21 °C, а кватирка відкривається, коли температура перевищує 23 °C; у проміжку від 21 до 23 °C обидва механізми зберігають попередній стан. Загальний принцип формування зони нечутливості та її призначення детально розглянуто в пункті 2.6.

Окремо слід зазначити роль зовнішньої температури. Зниження внутрішньої температури в розробленій системі досягається пасивно - відкриванням кватирки, через яку тепле повітря теплиці заміщується зовнішнім.

Такий спосіб охолодження ефективний лише за умови, що зовні холодніше, ніж усередині. Якщо ж зовнішня температура дорівнює внутрішній або перевищує її, відкривання кватирки не дає охолоджувального ефекту. Цю ситуацію система розпізнає й формує відповідне попередження оператору, не вважаючи її аварією, оскільки обладнання залишається справним.

Для виявлення таких випадків у складі системи передбачено окремий зовнішній давач температури, який не входить до жодного контуру регулювання й використовується винятково для індикації та формування зазначеного попередження.

2.3. Світловий режим і фотоперіод

Світло є джерелом енергії для фотосинтезу, тому забезпечення достатнього рівня освітленості - одна з ключових функцій системи керування мікрокліматом. На відміну від побутового освітлення, де рівень світла оцінюють в одиницях освітленості (люксах), для потреб рослинництва

використовують поняття фотосинтетично активної радіації (PAR) - частини спектра в діапазоні довжин хвиль 400-700 нм, яку рослини здатні засвоювати [PAR-джерело - McCree]. Кількісно PAR характеризують щільністю потоку фотонів, що вимірюється в мікромолях на квадратний метр за секунду ($\mu\text{mol}/\text{m}^2/\text{s}$). Саме ця величина, а не люкси, є фізично коректною мірою світла для рослин, оскільки фотосинтез залежить від кількості фотонів, а не від енергетичної яскравості світла [PAR-джерело - McCree]; тому в усій роботі рівень освітленості подано в $\mu\text{mol}/\text{m}^2/\text{s}$.

Денну уставку освітленості в системі прийнято на рівні $1000 \mu\text{mol}/\text{m}^2/\text{s}$, що відповідає потребам світлолюбних культур у період активної вегетації [Суворін]. Робочий діапазон, доступний оператору для задання уставки, становить від 100 до $1800 \mu\text{mol}/\text{m}^2/\text{s}$, що дає змогу налаштувати систему під різні культури та фази їхнього розвитку. Гістерезис контуру освітленості прийнято рівним $\Delta = 100 \mu\text{mol}/\text{m}^2/\text{s}$. За уставки $1000 \mu\text{mol}/\text{m}^2/\text{s}$ фітолампа вмикається при зниженні рівня PAR до $950 \mu\text{mol}/\text{m}^2/\text{s}$ і вимикається при підвищенні до $1050 \mu\text{mol}/\text{m}^2/\text{s}$. Загальний принцип формування зони нечутливості та її зв'язок із похибкою давача розглянуто в пункті 2.6.

Принципова особливість світлового режиму полягає в наявності фотоперіоду - чергування світлої та темної фаз доби. Для нормального розвитку рослинам потрібен не лише достатній рівень освітленості вдень, а й повноцінний темновий період уночі. Тому контур освітленості, на відміну від контурів температури та вологості, працює не цілодобово, а лише в межах світлової фази.

Темновий період реалізовано через нічну уставку освітленості, що дорівнює нулю. За нульової уставки умова вмикання фітолампи стає недосяжною, оскільки рівень PAR не може опуститися нижче від'ємної нижньої межі зони гістерезису. Завдяки цьому вночі лампа лишається вимкненою навіть за наявності стороннього світла, і темновий період витримується незалежно від показань давача освітленості.

Межі світлової фази доби - час її початку та кінця - задаються оператором і за замовчуванням становлять з 6-ї до 22-ї години. Зміна цих меж дає змогу налаштувати тривалість фотоперіоду під вимоги конкретної культури, оскільки різні рослини потребують різної тривалості світлового дня. Перемикання між світлою й темною фазами здійснює таймер, який водночас керує й перемиканням денної та нічної уставок температури, описаним у попередньому підрозділі. Таким чином, фаза доби є спільним чинником, що узгоджено впливає одразу на два контури - температури та освітленості.

2.4. Режим вологості повітря

Відносна вологість повітря - третій параметр мікроклімату, що підлягає регулюванню. Вона визначає інтенсивність транспірації рослин - випаровування води листям. За надто низької вологості транспірація стає надмірною, рослина втрачає вологу швидше, ніж встигає поповнювати її через коріння, що призводить до в'янення. За надмірно високої вологості транспірація майже припиняється, погіршується надходження поживних речовин, а на листі та плодах створюються сприятливі умови для розвитку грибкових захворювань. Тому вологість потрібно підтримувати в певному діапазоні, уникаючи обох крайнощів.

На відміну від температури, режим вологості не залежить від фази доби: потреби рослин у вологості вдень і вночі практично однакові. Тому в системі передбачено єдину уставку вологості, спільну для світлової й темної фаз. За замовчуванням її прийнято на рівні 70 %, що є типовим робочим значенням для тепличних культур [Губарєв]. Оператор може змінювати уставку в межах робочого діапазону від 55 до 90 % відповідно до вимог вирощуваної культури; верхню межу обмежено значенням 90 %, перевищення якого пригнічує фотосинтез [Суворін].

Регулювання вологості здійснюється зволожувачем повітря, який підвищує вологість у разі її зниження. Гістерезис контуру вологості прийнято

$\Delta = 5 \%$, тобто за уставки 70% зволожувач вмикається при зниженні вологості до $67,5 \%$ і вимикається при підвищенні до $72,5 \%$. Загальний принцип формування зони нечутливості розглянуто в пункті 2.6.

Окремо слід зазначити взаємозв'язок контуру вологості з контуром температури. Відкривання вентиляційної квартирки для зниження температури одночасно знижує й вологість повітря, оскільки вологе повітря тепліше заміщується сухішим зовнішнім. Через це одночасна робота квартирки та зволожувача була б суперечливою: зволожувач намагався б підвищити вологість, тоді як відкрита квартирка її знижувала б. Для усунення цієї суперечності в системі передбачено координацію контурів, згідно з якою за відкритої квартирки зволожувач блокується. Детально це правило розглянуто в розділі 3.

2.5. Засоби вимірювання та виконавчі механізми

Розглянуті в попередніх пунктах параметри мікроклімату мають вимірюватися й регулюватися реальними технічними засобами. Хоча в межах роботи система реалізована як програмна модель, склад її апаратної частини відповідає реально існуючому обладнанню: для кожного давача та виконавчого механізму обрано конкретні промислові прилади, а їхні характеристики покладено в основу параметрів моделі. Такий підхід забезпечує відповідність моделі реальним умовам роботи й дає змогу за потреби перенести систему на фізичне обладнання без зміни її логіки.

Усі первинні перетворювачі обрано з цифровим вихідним сигналом. Порівняно з аналоговими давачами цифрові не потребують окремого аналого-цифрового перетворювача, нечутливі до завад у лінії зв'язку й передають уже відкаліброване значення безпосередньо на мікроконтролер, що спрощує систему та підвищує достовірність вимірювань.

Давач температури та вологості. Для вимірювання температури й відносної вологості повітря обрано цифровий давач Sensirion SHT35 з захисною мембраною (SHT35-DIS-F). Цей давач поєднує в одному корпусі

два чутливі елементи - температури й вологості - і передає дані за цифровим інтерфейсом I²C. Паспортна точність становить $\pm 0,1$ °C для температури та $\pm 1,5$ % для відносної вологості; діапазон вимірювання температури - від -40 до +125 °C, вологості - від 0 до 100 %. Живлення давача - від 2,4 до 5,5 В, що дає змогу під'єднати його безпосередньо до мікроконтролера без додаткового перетворювача рівня. Захисна мембрана забезпечує стійкість чутливого елемента до конденсату й пилу, що важливо для умов підвищеної вологості в теплиці. Зовнішній вигляд давача наведено на рисунку 2.1.



Рисунок 2.1 - Цифровий давач температури та вологості Sensirion SHT35-DIS-F

У системі використано два таких давачі. Перший встановлено всередині теплиці - він вимірює одночасно температуру (давач T_{in} контуру СК₁) і відносну вологість (давач контуру СК₃). Другий розміщено зовні й використовується лише як давач зовнішньої температури (T_{out}) для формування попередження про неможливість пасивного охолодження, як описано в пункті 2.2. Канал вологості зовнішнього давача в роботі не задіяно. Застосування однакового давача в обох точках уніфікує склад обладнання та спрощує його обслуговування.

Давач освітленості. Для вимірювання рівня фотосинтетично активної радіації обрано цифровий квантовий давач Apogee SQ-521-SS. На відміну від звичайних давачів освітленості, що працюють в одиницях люкс і розраховані на чутливість людського ока, квантовий давач вимірює щільність потоку фотонів ($\mu\text{mol}/\text{m}^2/\text{s}$) - тобто саме ту величину, яка є фізично коректною мірою світла для рослин (пункт 2.3).

Похибка калібрування давача становить $\pm 5\%$, повторюваність вимірювань - менше 1% , а діапазон вимірювання доходить до $4000\ \mu\text{mol}/\text{m}^2/\text{s}$, що з запасом перекриває весь робочий діапазон уставок освітленості системи.

Давач має цифровий вихід за протоколом SDI-12, призначеним саме для під'єднання вимірювальних давачів до контролерів, із захищеним самоочисним корпусом з анодованого алюмінію, придатним для роботи в умовах теплиці.

Зовнішній вигляд давача наведено на рисунку 2.2.



Рисунок 2.2 - Квантовий давач PAR Apogee SQ-521-SS

Мікроконтролер. Опитування давачів, порівнювання вимірних значень з заданими, визначення, які виконавчі механізми вмикати чи вимикати і саме керування виконавчими механізмами виконує мікроконтролер.

Для цього обрано модуль ESP32 (WROOM-32), що має апаратний інтерфейс I²C для під'єднання давачів SHT35, щонайменше чотири цифрові виходи - по одному на кожен виконавчий механізм та інтерфейс USB для зв'язку з комп'ютером, на якому працює програмна частина системи. Обчислювальних ресурсів модуля з надлишком достатньо для реалізованого двопозиційного алгоритму керування. Зовнішній вигляд модуля наведено на рисунку 2.3.

Виконавчі механізми та їх комутація. Система містить чотири виконавчі механізми: нагрівач і вентиляційну кватирку (контур температури СК₁), фітолампку (контур освітленості СК₂) та зволожувач повітря (контур вологості СК₃).



Рисунок 2.3 - Мікроконтролер ESP32 WROOM-32

Усі вони працюють у режимі «увімкнено / вимкнено», що відповідає двопозиційному закону регулювання. Нагрівач, фітолампа та зволожувач - це звичайні мережеві прилади, а кватирку відкриває й закриває лінійний електропривод.

Мікроконтролер не може вмикати мережеве навантаження напряму, тому між ним і виконавчими механізмами встановлено чотириканальний релейний модуль на 5 В з опторозв'язкою. Кожен канал модуля керує одним механізмом, а опторозв'язка електрично відокремлює силову частину від мікроконтролера й захищає його. Зовнішній вигляд модуля наведено на рисунку 2.4.

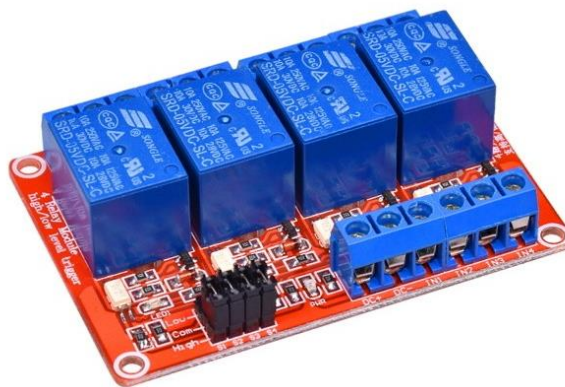


Рисунок 2.4 - Чотириканальний релейний модуль

Обрані засоби визначають характеристики, покладені в основу програмної моделі.

Похибки давачів задають рівень шуму вимірювання, врахований при виборі гістерезисів (пункт 2.6).

Узагальнений склад технічних засобів системи наведено в таблиці 2.3.

Таблиця 2.3 - Склад засобів вимірювання та виконавчих механізмів

Призначення	Обраний засіб	Інтерфейс / тип сигналу	Контур
Температура й вологість усередині	Sensirion SHT35-DIS-F	цифровий, I ² C	СК ₁ , СК ₃
Температура зовнішня	Sensirion SHT35-DIS-F	цифровий, I ² C	індикація
Освітленість (PAR)	Apogee SQ-521-SS	цифровий, SDI-12	СК ₂
Керування системою	ESP32 WROOM-32	мікроконтролер	усі
Комутація виконавчих механізмів	Релейний модуль 4-канальний, 5 В	реле з опторозв'язкою	усі

Керування виконавчими механізмами через релейний модуль відбувається в режимі «увімкнено / вимкнено», що відповідає обраному двопозиційному закону регулювання та визначає спосіб формування керуючих сигналів у моделі. Таким чином, параметри реальних технічних засобів безпосередньо відображені у програмній реалізації системи.

2.6. Нормативні діапазони та параметри регулювання

Уставки, розглянуті в попередніх пунктах, обрано в межах нормативних діапазонів параметрів мікроклімату, рекомендованих для тепличних культур. Узагальнені оптимальні діапазони основних параметрів наведено в таблиці 2.1.

Таблиця 2.1 - Оптимальні діапазони параметрів мікроклімату теплиці [Суворін]

Параметр	Одиниця	Оптимум	Спосіб регулювання
Температура (день)	°C	19-26.5	нагрівач / вентиляція
Температура (ніч)	°C	16.5-18	нагрівач
Відносна вологість	%	58-87	зволожувач / вентиляція
PAR (освітленість)	μmol/m ² /s	200-1600 (день)	додаткове освітлення (досвітка)
CO ₂	ppm	400-700	джерело CO ₂ / вентиляція

Наведені у таблиці значення є узагальненими для більшості овочевих і ягідних культур, що вирощуються в теплицях [Суворін]; конкретні установки можуть коригуватися залежно від виду культури та стадії її розвитку.

У роботі [Суворін] наведено диференціацію параметрів за стадіями розвитку рослини - від проростання до плодоношення. У межах даного проєкту реалізовано усереднений режим керування мікрокліматом без поділу за стадіями, що є достатнім для демонстрації роботи системи; врахування стадій розвитку є можливим напрямом її подальшого розвитку.

Прийняті за замовчуванням значення (22 °C удень, 17 °C уночі, 70 % вологості, 1000 $\mu\text{mol}/\text{m}^2/\text{s}$ освітленості) лежать у межах відповідних діапазонів. Параметр концентрації вуглекислого газу (CO_2), хоч і є нормованим, у розробленій системі не регулюється; його наведено в таблиці для повноти картини.

Зведені характеристики всіх трьох контурів - цільові значення, діапазони задання оператором, гістерезиси та виконавчі механізми - наведено в таблиці 2.2.

Наведені в таблиці 2.2 параметри є вихідними даними для проєктування системи керування. Вони однозначно визначають поведінку кожного контуру: за відомими уставкою та гістерезисом обчислюються межі зони нечутливості, і доки параметр перебуває в цих межах, стан виконавчого механізму не змінюється, а вмикання чи вимикання відбувається лише при виході параметра за межі зони.

Таблиця 2.2 - Параметри регулювання системи

Параметр	Цільове значення (за замовч.)	Діапазон задання оператором	Гістерезис Δ	Виконавчий механізм
Температура (день)	22°C	16...28°C	2°C	Нагрівач / Кватирка
Температура (ніч)	17°C	16...28°C	2°C	Нагрівач / Кватирка
Освітленість (PAR)	1000 $\mu\text{mol}/\text{m}^2/\text{s}$	100...1800	100 $\mu\text{mol}/\text{m}^2/\text{s}$	Фітолампа
Відносна вологість	70%	55...90%	5%	Зволожувач

Ці значення зберігаються в системі централізовано й використовуються всіма контурами під час роботи, що, по-перше, виключає неузгодженість параметрів між різними частинами системи, а по-друге, дає змогу оператору змінювати налаштування в одному місці, забезпечуючи адаптацію системи до конкретної культури та фази її розвитку.

Працездатність системи залежить не лише від значень параметрів, а й від періоду опитування датчиків. Період опитування датчиків визначає, як часто система перераховує стан параметрів і за потреби перемикає виконавчі механізми. Оскільки температура й вологість є інерційними й помітно змінюються лише за десятки секунд, надто часте опитування недоцільне - за короткий інтервал параметр не встигає змінитися настільки, щоб знадобилося втручання. Водночас період не має бути зовеликим, аби система вчасно реагувала на відхилення. З огляду на це, період опитування прийнято рівним 30-и секундам.

2.7. Принцип двопозиційного керування

Розглянемо загальний принцип, за яким система приймає рішення про вмикання та вимикання виконавчих механізмів, - двопозиційне регулювання із зоною нечутливості (гістерезисом), спільне для всіх трьох контурів.

У системі застосовано двопозиційний (релейний) закон регулювання, за якого кожен виконавчий механізм перебуває лише в одному з двох станів - увімкненому або вимкненому, без проміжних положень. Такий закон обрано з огляду на інерційність об'єктів керування та дискретний характер самих виконавчих механізмів: нагрівач, кватирка, фітолампа та зволожувач за своєю природою або працюють, або ні, тож плавне керування для них не передбачене. Детальне обґрунтування вибору саме цього закону, порівняно з неперервними законами регулювання, наведено в розділі 3.

Якби механізм вмикався й вимикався точно на рівні уставки, поблизу неї виникало б часте безперервне перемикавання, спричинене найменшими коливаннями вимірюваної величини. Для уникнення цього навколо уставки

формують зону нечутливості завширшки в гістерезис Δ , у межах якої стан механізму не змінюється. Нижня межа зони визначається як $Z_{зад} - \Delta/2$, верхня - як $Z_{зад} + \Delta/2$, де $Z_{зад}$ - задана оператором уставка. Механізм, що підвищує значення параметра, вмикається при досягненні нижньої межі й вимикається при досягненні верхньої; для механізму, що значення параметра знижує, - порядок зворотний. Доки параметр перебуває всередині зони, виконавчий механізм зберігає попередній стан, завдяки чому перемикання відбувається лише при суттєвому відхиленні параметра, а не при кожному незначному його коливанні.

Окремо слід враховувати, що реальні давачі мають скінченну точність, тому виміряне значення завжди містить шумову складову - випадкове відхилення від справжнього значення параметра. Якщо ширина зони нечутливості близька до похибки давача, шум вимірювання сам по собі спричиняє хибні перемикання механізму навіть тоді, коли параметр фактично не змінюється. Щоб цього не сталося, гістерезис кожного контуру свідомо обирають помітно більшим за похибку відповідного давача. Похибки прийнято відповідно до характеристик обраних у пункті 2.5 засобів вимірювання, з певним запасом у бік погіршення: $\pm 0,3$ °C для температури, $\pm 2,0$ % для вологості (давач SHT35) та ± 40 $\mu\text{mol}/\text{m}^2/\text{s}$ для освітленості (давач SQ-521). Гістерезиси контурів - 2 °C, 5 % та 100 $\mu\text{mol}/\text{m}^2/\text{s}$ відповідно - у кілька разів перевищують ці похибки, що й забезпечує стійкість регулювання до шуму вимірювання.

Висновки до розділу 2

У розділі визначено параметри мікроклімату теплиці, що підлягають автоматизованому регулюванню, обґрунтовано вимоги до них та обрано технічні засоби для їх вимірювання й регулювання.

Установлено, що регулюванню підлягають три параметри мікроклімату: -температура, освітленість і відносна вологість, які найсильніше впливають на розвиток рослин і піддаються прямому керуванню порівняно простими

виконавчими механізмами. Для кожного параметра визначено робочий діапазон, цільові значення та гістерезис, і кожному параметру відповідає окремий контур керування.

Показано, що температурний і світловий режими залежать від фази доби, тому для них передбачено окремі денні й нічні уставки, тоді як режим вологості від фази доби не залежить. Обґрунтовано вибір фотосинтетично активної радіації ($\mu\text{mol}/\text{m}^2/\text{s}$) як фізично коректної міри освітленості для рослин, а також реалізацію темного періоду через нульову нічну уставку освітленості.

Розглянуто взаємозв'язок контурів температури та вологості: відкривання вентиляційної кватирки одночасно знижує обидва параметри, через що передбачено координацію, за якої при відкритій кватирці зволожувач блокується.

Для вимірювання параметрів обрано цифрові давачі - Sensirion SHT35 для температури та вологості й Apogee SQ-521-SS для освітленості, - а керування виконавчими механізмами покладено на мікроконтролер ESP32, що комутує навантаження через релейний модуль з опторозв'язкою. Характеристики обраних засобів покладено в основу параметрів програмної моделі, завдяки чому модель відповідає реально існуючому обладнанню.

Уставки, прийняті в системі, узгоджено з нормативними діапазонами параметрів мікроклімату, рекомендованими для тепличних культур. Визначено вимоги до точності вимірювання та періоду опитування датчиків і показано, що гістерезис кожного контуру обирається більшим за похибку відповідного давача, що забезпечує стійкість регулювання до шуму вимірювання.

Сформовано зведену таблицю параметрів регулювання, яка разом із розглянутим принципом двопозиційного керування слугує вихідними даними для подальшого проектування системи.

РОЗДІЛ 3. ПРОЄКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ

3.1. Структурна схема системи

Структурна схема відображає склад системи керування та зв'язки між її елементами на рівні функціональних блоків, без деталізації технічної реалізації. Структурну схему розробленої автоматизованої системи керування мікрокліматом теплиці наведено на рис. 3.1.

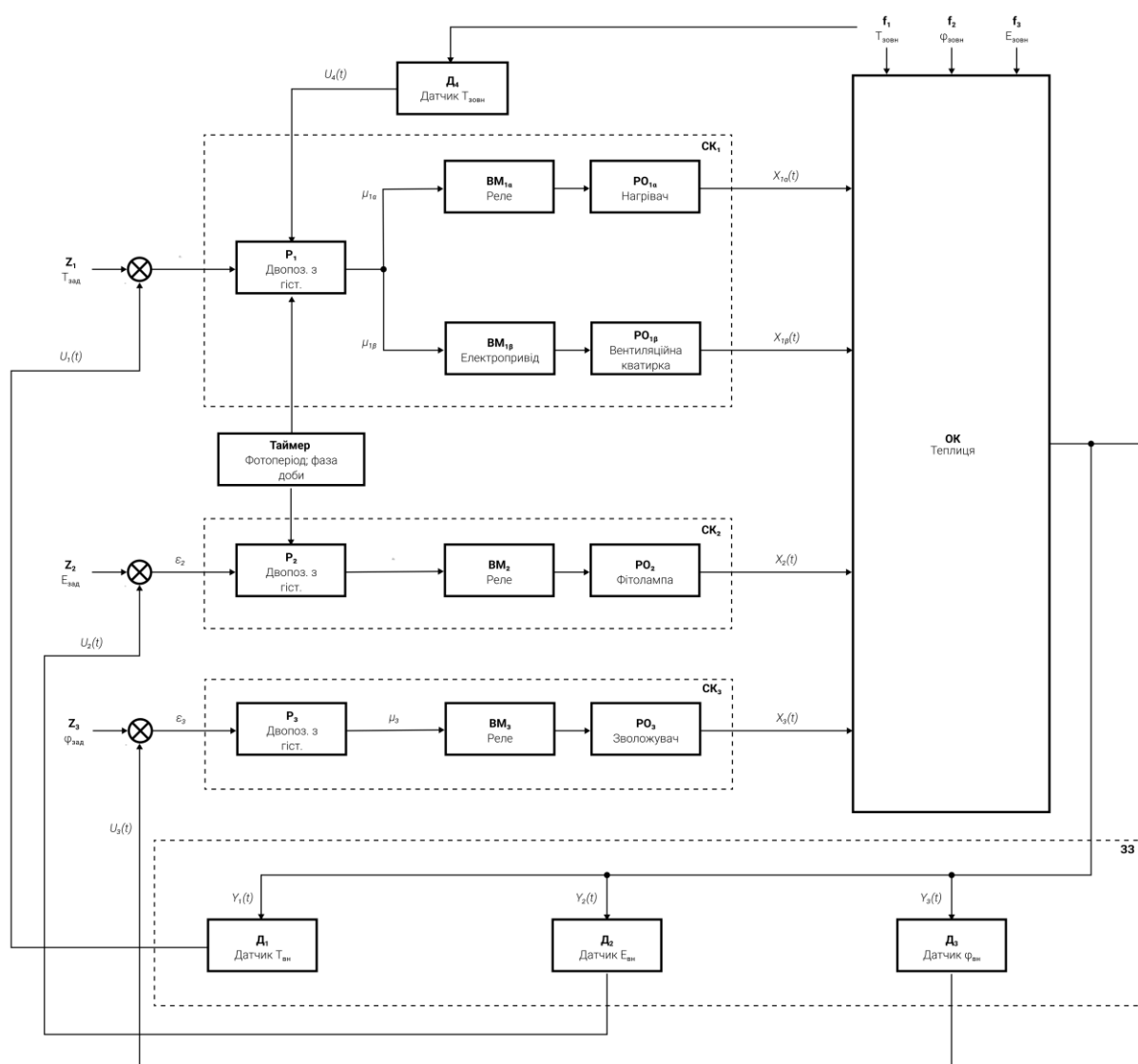


Рисунок 3.1 - Структурна схема автоматизованої системи керування мікрокліматом теплиці

Система побудована за принципом замкнутого регулювання з трьома незалежними контурами, кожен з яких відповідає за один параметр мікроклімату. Об'єктом керування (ОК) виступає теплиця, стан якої характеризується трьома регульованими величинами - температурою, освітленістю та відносною вологістю повітря. На об'єкт діють зовнішні збурення f_1, f_2, f_3 (зовнішня температура, вологість та освітленість), які відхиляють параметри від заданих значень.

Кожен контур керування реалізує однакову структуру замкнутого регулювання. На вхід контуру подається задане значення параметра (уставка) $Z - T_{зад}, E_{зад}$ або $\varphi_{зад}$ відповідно.

Виміряне значення параметра, отримане давачем, повертається до елемента порівняння, де віднімається від уставки. Різниця між ними утворює сигнал розузгодження ε . На основі цього сигналу регулятор Р приймає рішення про вмикання чи вимикання виконавчих механізмів.

Усі три регулятори (P_1, P_2, P_3) реалізують двопозиційний закон керування з гістерезисом. Кожен регулятор формує керувальний сигнал μ , який надходить на виконавчий механізм (ВМ), а той приводить у дію регульовальний орган (РО) - пристрій, що безпосередньо впливає на об'єкт.

Контури мають різну складність, що відображає особливості відповідних параметрів.

Контур температури ($СК_1$) містить два канали з протилежним впливом на об'єкт: канал нагрівання (реле $ВМ_{1\alpha} \rightarrow$ нагрівач $РО_{1\alpha}$), що підвищує температуру, та канал вентиляції (електропривід $ВМ_{1\beta} \rightarrow$ вентиляційна квартира $РО_{1\beta}$), що її знижує.

Контур освітленості ($СК_2$) і контур вологості ($СК_3$) мають по одному каналу: фітолампа (через реле $ВМ_2$) та зволожувач (через реле $ВМ_3$) відповідно.

Між контурами температури й освітленості на схемі розташовано таймер. Він задає фотоперіод - межі світлової та темної фаз доби. Поточна фаза доби, визначена за цими межами, одночасно впливає на два контури:

задає, яку уставку температури - денну чи нічну - використовує контур СК₁, та визначає, чи працює контур освітленості СК₂.

Таким чином, фаза доби є спільним чинником для контурів СК₁ і СК₂. Контур вологості СК₃ від неї не залежить.

Окремо від контурів регулювання на схемі показано давач зовнішньої температури Д₄. Він не входить до жодного контуру й не бере участі в регулюванні, а слугує для спостереження за зовнішньою температурою: за його показаннями система визначає, чи здатна вентиляція забезпечити охолодження, і формує попередження оператору в тому разі, якщо зовнішня температура не нижча за внутрішню.

Зворотний зв'язок у системі реалізовано трьома внутрішніми давачами, об'єднаними в блок (позначений на схемі як ЗЗ): давач температури Д₁, давач освітленості Д₂ та давач вологості Д₃.

Кожен з них вимірює відповідний параметр усередині теплиці (сигнали Y_1 , Y_2 , Y_3) і передає виміряне значення сигналами U_1 , U_2 , U_3 до елемента порівняння свого контуру, замикаючи контур регулювання.

3.2. Принцип двопозиційного регулювання з гістерезисом

Як зазначено в попередньому пункті, усі три контури системи реалізують двопозиційний закон керування з гістерезисом. Вибір саме цього закону регулювання зумовлений характером об'єкта та виконавчих механізмів.

Двопозиційне (релейне) регулювання передбачає, що виконавчий механізм може перебувати лише у двох станах - увімкненому або вимкненому, без проміжних положень. Це повністю відповідає природі виконавчих органів розробленої системи: нагрівач, фітолампа та зволожувач або працюють, або ні, а вентиляційна кватирка в моделі розглядається як відкрита чи закрита. Для таких механізмів плавне (пропорційне) регулювання технічно надлишкове, а двопозиційне є природним і найпростішим у реалізації. Доцільність двопозиційного підходу для керування мікрокліматом

теплиць підтверджується і практикою: він є найпоширенішим рішенням для подібних задач [\[Kokieva\]](#).

Якщо застосувати чисте двопозиційне регулювання з одним порогом, що дорівнює уставці, виникає проблема частих перемикань. Поблизу уставки навіть незначні коливання вимірюного значення - спричинені шумом давача або повільним дрейфом параметра - змушували б виконавчий механізм безперервно вмикатися й вимикатися. Таке «дрижання» спричиняє прискорений знос комутаційної апаратури та виконавчих органів і є неприпустимим.

Для усунення цього недоліку застосовують гістерезис - зону нечутливості навколо уставки, у межах якої стан виконавчого механізму не змінюється. Замість одного порогу вводять два: нижній і верхній, рознесені відносно уставки на половину ширини зони. Перемикання відбувається лише при перетині цих порогів, а доки виміряне значення перебуває між ними, механізм зберігає попередній стан. Завдяки цьому одне коливання параметра вже не здатне викликати повторне перемикання, і частота спрацювань різко зменшується.

Формально для механізму, що підвищує параметр (нагрівач, фітолампа, зволожувач), закон регулювання має такий вигляд: механізм вмикається, коли виміряне значення x опускається нижче нижнього порогу ($Z_{зад} - \Delta/2$), і вимикається, коли значення перевищує верхній поріг ($Z_{зад} + \Delta/2$), де $Z_{зад}$ - уставка, Δ - гістерезис (повна ширина зони нечутливості). У проміжку між порогами стан механізму не змінюється. Для механізму, що знижує параметр (вентиляційна кватирка), закон дзеркальний: він вмикається при перевищенні верхнього порогу й вимикається при опусканні нижче нижнього.

Ширину зони нечутливості Δ обирають як компроміс. Завузька зона погано захищає від частих перемикань, оскільки шум вимірювання легко її перекриває; заширока - знижує точність підтримання параметра, бо допускає більше відхилення від уставки до моменту спрацювання. Прийняті в системі значення гістерезису (2 °C для температури, 5 % для вологості, 100 $\mu\text{mol}/\text{m}^2/\text{s}$

для освітленості) обрано з огляду на це так, щоб забезпечити захист від хибних перемикачів і водночас зберегти прийнятну точність регулювання, як обґрунтовано в пункті 2.5.

Двопозиційний закон з гістерезисом застосовано однаково до всіх трьох контурів. Відмінності між контурами полягають не в законі регулювання, а в кількості та напрямі дії виконавчих механізмів: контур температури використовує два регулятори протилежної дії (на підвищення й на зниження), тоді як контури освітленості та вологості - по одному регулятору на підвищення. Конкретні алгоритми роботи кожного контуру розглянуто в розділі 4.

3.3. Функціональна схема автоматизації (ФСА)

Функціональна схема автоматизації деталізує систему на рівні конкретних засобів контролю та керування, показуючи, якими приладами вимірюється кожен параметр, де вони розташовані та як пов'язані з регулювальними й виконавчими пристроями. Схему розроблено відповідно до ДСТУ Б А.2.4-3:2009 і наведено на рис. 3.3.

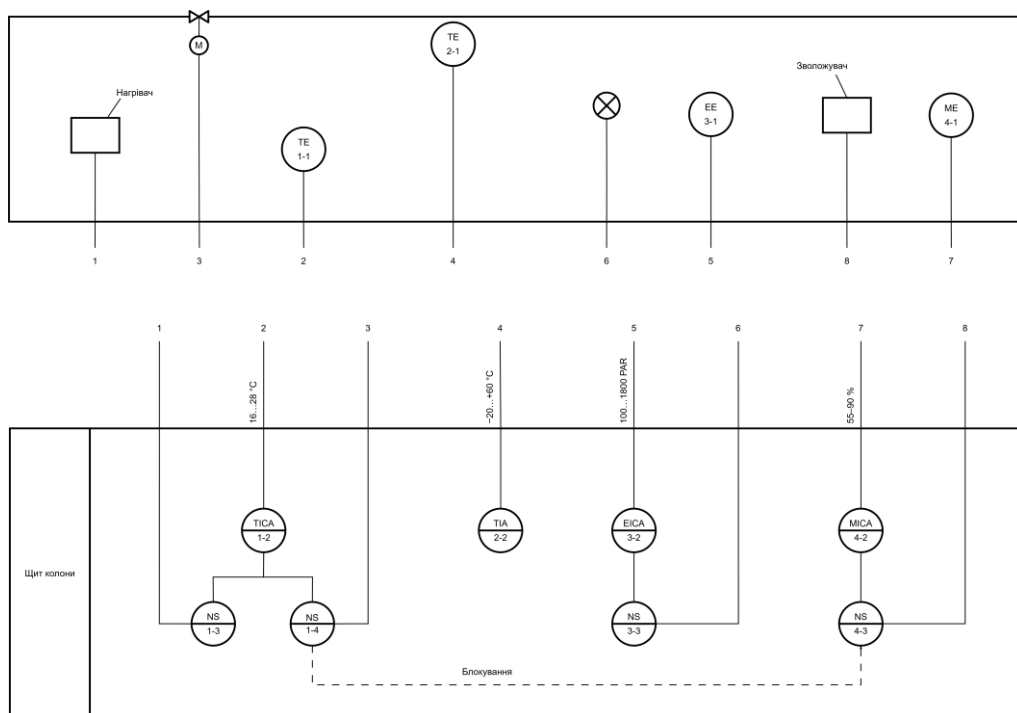


Рисунок 3.3 - Функціональна схема автоматизації системи керування мікрокліматом теплиці

На схемі виділено два рівні. Верхній прямокутник відповідає технологічному об'єкту – теплиці, на якому розташовані первинні перетворювачі (давачі) та регулювальні органи. Нижній прямокутник - щит керування, де зосереджено засоби обробки сигналів, регулювання та комутації.

Система містить чотири вимірювальні канали, пронумеровані як окремі контури. Перший контур (позиції 1-х) відповідає за температуру повітря теплиці, другий (2-х) - за зовнішню температуру, третій (3-х) - за освітленість, четвертий (4-х) - за вологість. У кожному контурі прилади з номером «-1» є первинними перетворювачами на об'єкті, з номером «-2» - вторинними приладами на щиті, з номерами «-3» та «-4» — виконавчими механізмами (реле). Первинні перетворювачі відповідають реальним приладам, обраним у пункті 2.5: температуру й вологість вимірює цифровий давач Sensirion SHT35-DIS-F, освітленість - квантовий давач Apogee SQ-521-SS.

Контур температури теплиці є найрозгалуженішим. Первинний перетворювач температури ТЕ 1-1 вимірює температуру повітря теплиці й передає сигнал на щит, де прилад ТІСА 1-2 виконує показання, двопозиційне регулювання й сигналізацію (літери Т - температура, І - показання, С - регулювання, А - сигналізація). Регулятор ТІСА 1-2 керує двома виконавчими механізмами через проміжні реле: NS 1-3 вмикає нагрівач, а NS 1-4 - електропривід вентиляційної квартирки. Нагрівач (регулювальний орган у каналі 1) та квартирка з електроприводом (канал 3) безпосередньо впливають на температуру об'єкта.

Контур зовнішньої температури спрощений порівняно з рештою. Первинний перетворювач ТЕ 2-1, розташований зовні теплиці, вимірює зовнішню температуру в діапазоні $-20...+60$ °С, а вторинний прилад ТІА 2-2 виконує лише показання та сигналізацію (літери Т, І, А), без функції регулювання (літера С відсутня). Це відповідає ролі зовнішнього давача: він не керує жодним механізмом, а лише інформує систему про зовнішні умови

для формування попередження, коли вентиляція не здатна забезпечити охолодження. Конструктивно зовнішній перетворювач TE 2-1 - це окремий давач SHT35-DIS-F, у якому задіяно лише канал температури.

Контур освітленості складається з первинного перетворювача PAR (позначеного EE 3-1, де E - освітленість), вторинного приладу EICA 3-2 (показання, регулювання, сигналізація) та реле NS 3-3, що вмикає фітолампку. Контур вологості побудовано аналогічно: первинний перетворювач ME 4-1 (M — вологість), вторинний прилад MICA 4-2 та реле NS 4-3, що керує зволожувачем. Первинні перетворювачі TE 1-1 і ME 4-1 фізично реалізовано одним давачем SHT35-DIS-F, що поєднує в одному корпусі чутливі елементи температури й вологості.

Окремо на схемі показано зв'язок блокування - штрихову лінію між реле квартирки (NS 1-4) та реле зволожувача (NS 4-3). Вона відображає координацію контурів температури й вологості: за умови, що квартира відкрита, реле зволожувача блокується, унеможливаючи одночасну роботу вентиляції та зволожувача. Цей зв'язок реалізує правило координації, докладно розглянуте в пункті 3.6.

Особливістю розробленої ФСА є те, що функції всіх вторинних приладів - показання, двопозиційного регулювання та сигналізації за всіма чотирма контурами - реалізовано програмно в єдиному мікроконтролері ESP32 WROOM-32. У специфікації до схеми (Додаток А) позиції 1-2, 2-2, 3-2 та 4-2 відповідають не окремим фізичним приладам, а функціям, які виконує мікроконтролер. Це відображає природу розробленої системи як програмної моделі: апаратні засоби автоматизації замінені їх програмною імітацією, що зберігає передбачену функціональну структуру системи.

3.4. Схема інформаційних потоків (DFD)

Якщо функціональна схема показує фізичну структуру засобів автоматизації, то схема інформаційних потоків (Data Flow Diagram, DFD) відображає систему з іншого боку - як рух і перетворення даних,

абстрагуючись від конкретного обладнання. Така схема дає змогу побачити, які дані виникають у системі, де вони зберігаються, якими процесами обробляються та між якими учасниками передаються. Схему інформаційних потоків розробленої системи виконано в нотації Гейна-Сарсона й наведено на рис. 3.4.

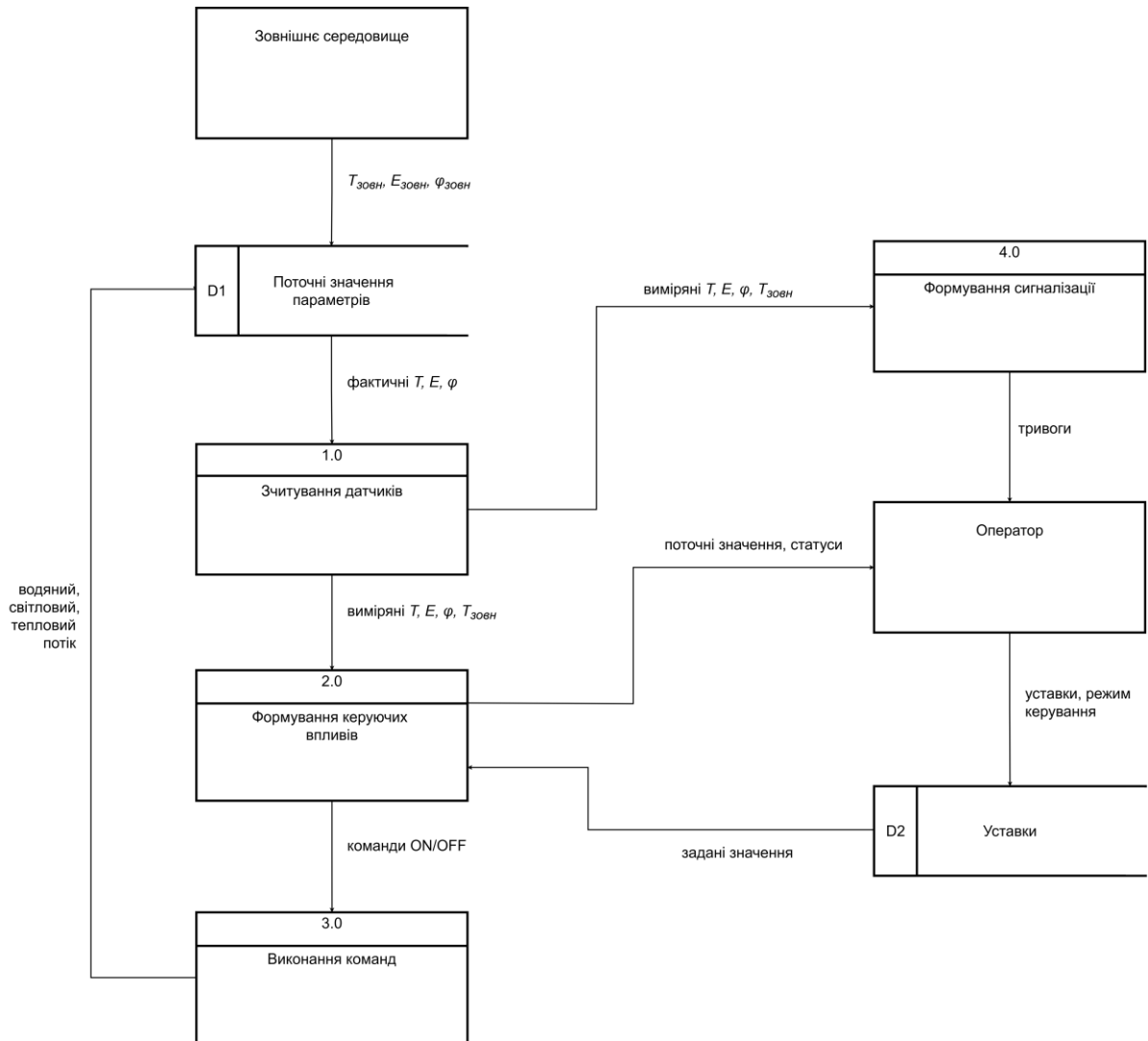


Рисунок 3.4 - Схема інформаційних потоків системи керування мікрокліматом теплиці

У системі визначено дві зовнішні сутності.

Перша - зовнішнє середовище, що є джерелом неконтрольованих впливів: зовнішніх температури, освітленості та вологості ($T_{зовн}$, $E_{зовн}$, $\varphi_{зовн}$), які надходять до системи й відображаються на стані параметрів теплиці.

Друга - оператор, який взаємодіє із системою у двох напрямках: задає уставки й режим керування, а також отримує від системи поточні значення параметрів, статуси та сигнали тривоги.

Обробку даних реалізовано чотирма процесами. Процес 1.0 «Зчитування датчиків» отримує фактичні значення параметрів і перетворює їх на виміряні величини (T , E , φ та зовнішню температуру), які далі передаються іншим процесам. Процес 2.0 «Формування керуючих впливів» є ядром системи: він порівнює виміряні значення з уставками, отриманими зі сховища, і формує рішення про стан виконавчих механізмів. Процес 3.0 «Виконання команд» перетворює ці рішення на команди ON/OFF, які впливають на об'єкт через водяний, світловий і тепловий потоки. Процес 4.0 «Формування сигналізації» аналізує виміряні значення та формує сигнали тривоги для оператора.

Система використовує два сховища даних.

Сховище D1 «Поточні значення параметрів» зберігає актуальні виміряні значення. До нього надходять дані від зовнішнього середовища та результати зчитування датчиків, а самі значення зі сховища використовують процеси регулювання (2.0) та сигналізації (4.0).

Сховище D2 «Уставки» зберігає задані оператором цільові значення й режим керування; його наповнює оператор, а використовує процес формування керуючих впливів, отримуючи звідти задані значення для порівняння з виміряними.

Рух даних утворює замкнутий цикл. Зовнішнє середовище та фактичний стан теплиці формують поточні значення параметрів (D1), які зчитуються процесом 1.0 і передаються процесу 2.0.

Той звіряє їх із уставками зі сховища D2 й формує керуючі рішення, які процес 3.0 перетворює на команди, що впливають на об'єкт.

Зміна стану об'єкта знову відображається в поточних значеннях, замикаючи цикл регулювання. Паралельно процес 4.0 відстежує виміряні значення й сповіщає оператора про відхилення, а оператор через сховище уставок впливає на поведінку системи.

Розподіл доступу до сховищ відображає принцип розмежування ролей у системі. Процес формування керуючих впливів лише зчитує уставки зі сховища D2, але не змінює їх - змінювати уставки може тільки оператор. І навпаки, поточні значення параметрів (D1) формуються вимірюванням, а оператор їх лише отримує для контролю, але не редагує.

Таке розмежування - оператор задає цілі, система їх виконує - закладено в архітектуру й забезпечує передбачуваність поведінки системи.

3.5. Об'єктно-орієнтована модель системи (UML)

Оскільки система є програмною моделлю, доцільно описати її засобами уніфікованої мови моделювання UML, що дає змогу подати систему з погляду програмної архітектури. Для цього використано два типи діаграм: діаграму використання (Use Case), яка описує систему з боку її користувачів та виконуваних функцій, і діаграму компонентів (Component), яка показує внутрішню структуру системи та зв'язки між її програмними модулями.

Діаграма використання. Діаграма використання, наведена на рис. 3.5, відображає функції системи з погляду тих, хто з нею взаємодіє. На діаграмі визначено двох акторів - оператора й таймер.

Оператор є основним користувачем системи й ініціює більшість функцій: задає уставки, переглядає поточний стан, здійснює ручне керування виконавчими механізмами, перемикає режим керування (автоматичний/ручний) та задає розклад фотоперіоду. Другим актором є таймер - він не є людиною, а представляє часовий механізм, що циклічно запускає регулювання й відстежує межі фаз доби.

Між окремими функціями встановлено стандартні відношення UML. Варіант використання «Задати розклад фотоперіоду» пов'язаний відношенням «include» з варіантом «Підтримати фотоперіод»: задання розкладу обов'язково включає підтримання фотоперіоду як свою невід'ємну складову.

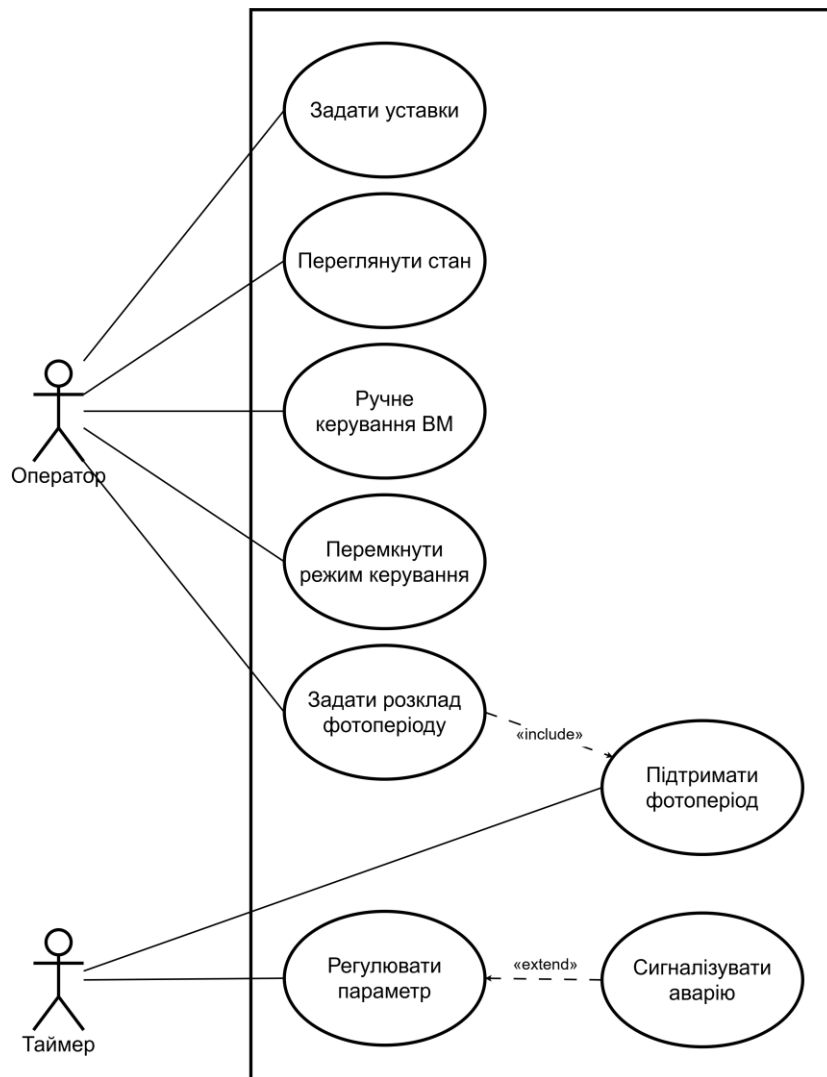


Рисунок 3.5 - Діаграма використання системи керування мікрокліматом теплиці

Варіант «Сигналізувати аварію» пов'язаний відношенням «extend» з варіантом «Регулювати параметр»: сигналізація є розширенням процесу регулювання, що активується лише за певної умови - виникнення аварійної ситуації, - а за нормальної роботи не виконується. Вибір саме «extend» підкреслює, що сигналізація не є обов'язковим кроком регулювання, а спрацьовує як виняток.

На діаграмі функцію «Регулювати параметр» ініціює актор Таймер, а не оператор. Це відображає характер регулювання: воно виконується автоматично й періодично, із заданим інтервалом, а не за командою користувача.

Оператор задає лише цілі та режим роботи, тоді як саме регулювання система виконує самостійно й циклічно.

Діаграма компонентів. Діаграма компонентів, наведена на рис. 3.6, відображає внутрішню будову програмної моделі - її поділ на модулі та потоки даних між ними. Кожен компонент відповідає окремому модулю програми з чітко визначеною функцією.

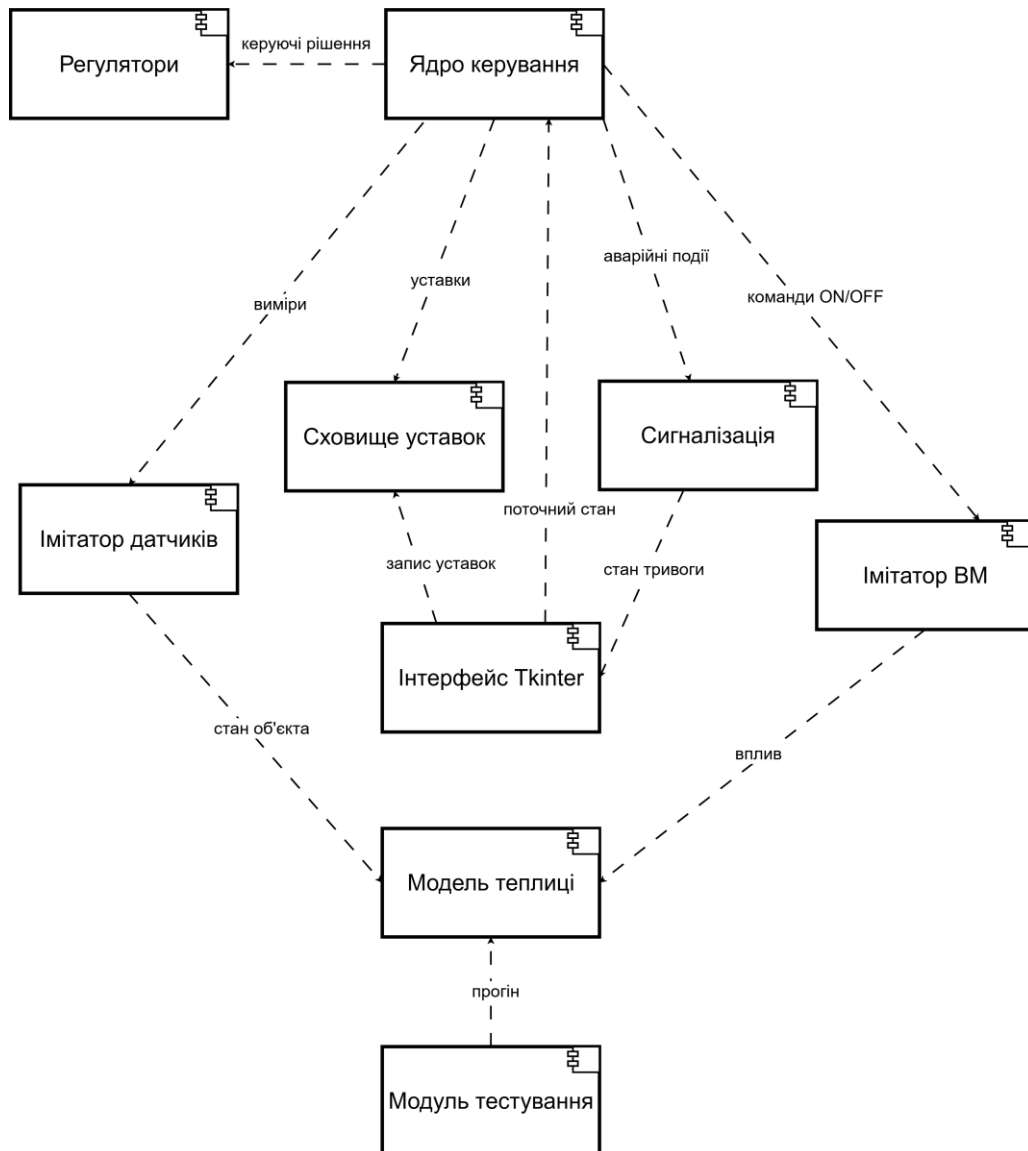


Рисунок 3.6 - Діаграма компонентів системи керування мікрокліматом теплиці

Центральним компонентом є ядро керування, що координує роботу системи. Воно отримує виміри від імітатора датчиків, звіряє їх з уставками зі

сховища уставок, передає регуляторам дані для прийняття рішень і отримує від них керуючі рішення, після чого надсилає команди ON/OFF імітаторові виконавчих механізмів.

Регулятори виділено в окремий компонент: регулятор є чистою функцією, яка приймає рішення на основі переданих їй значень і не має власного стану чи доступу до інших модулів.

Таке відокремлення робить логіку регулювання незалежною від решти системи.

Сховище уставок зберігає задані значення й обмінюється даними з інтерфейсом (запис уставок оператором) та ядром керування (зчитування для регулювання). Компонент сигналізації отримує від ядра аварійні події й передає стан тривоги інтерфейсу. Інтерфейс Tkinter забезпечує взаємодію з оператором: відображає поточний стан і стан тривоги, приймає введення уставок.

Імітатори датчиків і виконавчих механізмів моделюють апаратну частину. Імітатор датчиків зчитує стан об'єкта з моделі теплиці й передає виміри ядру; імітатор виконавчих механізмів отримує команди й чинить вплив на модель теплиці. Модель теплиці є програмним відображенням об'єкта керування - вона розраховує зміну параметрів під впливом виконавчих механізмів і зовнішніх умов.

Окремий компонент забезпечує автоматичний прогін моделі - багаторатне повторення такту регулювання без участі оператора - для отримання перехідних характеристик системи. Він взаємодіє з моделлю теплиці, а отримані результати використано для побудови графіків зміни параметрів (рис. 4.6).

Поділ системи на компоненти відображає принцип розмежування відповідальності: кожен модуль виконує одну чітко окреслену функцію, а зв'язки між ними реалізовано через визначені потоки даних. Зокрема, регулятори відокремлено від моделі об'єкта, а сховище уставок виступає

єдиною точкою доступу до заданих значень, що спрощує розуміння й супровід системи.

3.6. Координація контурів керування

Три контури регулювання працюють незалежно, доки їхні виконавчі механізми не впливають на той самий параметр. Виняток становлять контури температури й вологості: вентиляційна квартира, відкриваючись для охолодження, одночасно знижує вологість, оскільки вологе повітря теплиці заміщується сухішим зовнішнім. Через це за одночасного перегріву й нестачі вологості виникає конфлікт - зволожувач намагається підняти вологість, тоді як відкрита квартира її знижує, і зволожувач працює марно.

Для усунення конфлікту введено правило координації: за відкритою квартирою зволожувач блокується незалежно від показань датчика вологості й відновлює роботу, щойно квартира зачиняється.

Пріоритет надано контуру температури, оскільки перегрів небезпечніший для рослин, ніж тимчасове зниження вологості. Координація є односторонньою - квартира впливає на зволожувач, але не навпаки: охолодження виконується без обмежень.

У системі це правило діє як окремий етап, що застосовується після формування рішень усіма контурами, тому має пріоритет над рішенням контуру вологості.

Висновки до розділу 3

У розділі виконано проектування автоматизованої системи керування мікрокліматом теплиці на структурному, функціональному та програмному рівнях.

Розроблено структурну схему системи, що подає її як сукупність трьох незалежних замкнутих контурів регулювання - температури, освітленості та вологості - з єдиним об'єктом керування. Визначено склад кожного контуру: датчик, регулятор, виконавчий механізм і регульований орган, а також роль

таймера, як спільного чинника для контурів температури й освітленості, та зовнішнього давача температури для формування попереджень.

Обґрунтовано вибір двопозиційного закону керування з гістерезисом для всіх контурів. Показано, що цей закон відповідає релейному характеру виконавчих механізмів, а гістерезис усуває часті перемикання поблизу уставки, спричинені шумом вимірювання.

Розроблено функціональну схему автоматизації за ДСТУ Б А.2.4-3:2009, на якій конкретизовано засоби контролю та керування за чотирма вимірювальними каналами, та відображено особливості системи - програмну реалізацію функцій вторинних приладів у єдиному мікроконтролері.

Побудовано схему інформаційних потоків у нотації Гейна-Сарсона, що відображає рух і перетворення даних між процесами та сховищами, а також об'єктно-орієнтовану модель системи у вигляді діаграм використання та компонентів UML, яка описує функції системи та її поділ на програмні модулі.

Сформульовано правило координації контурів температури й вологості, згідно з яким за відкритої вентиляційної квартирки зволожувач блокується. Обґрунтовано пріоритет контуру температури та односторонній характер координації.

Спроектована структура є основою для розробки алгоритмічного забезпечення системи, розглянутого в наступному розділі.

РОЗДІЛ 4. АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

4.1. Головний алгоритм функціонування системи

Алгоритмічне забезпечення системи реалізує двопозиційне регулювання з гістерезисом, описане в розділі 3, у вигляді послідовності дій, що виконуються циклічно.

Головний алгоритм визначає загальний порядок роботи системи - від запуску до завершення - і координує виклик підпрограм керування окремими контурами.

Блок-схему головного алгоритму, побудовану відповідно до ГОСТ 19.701, наведено на рис. 4.1.

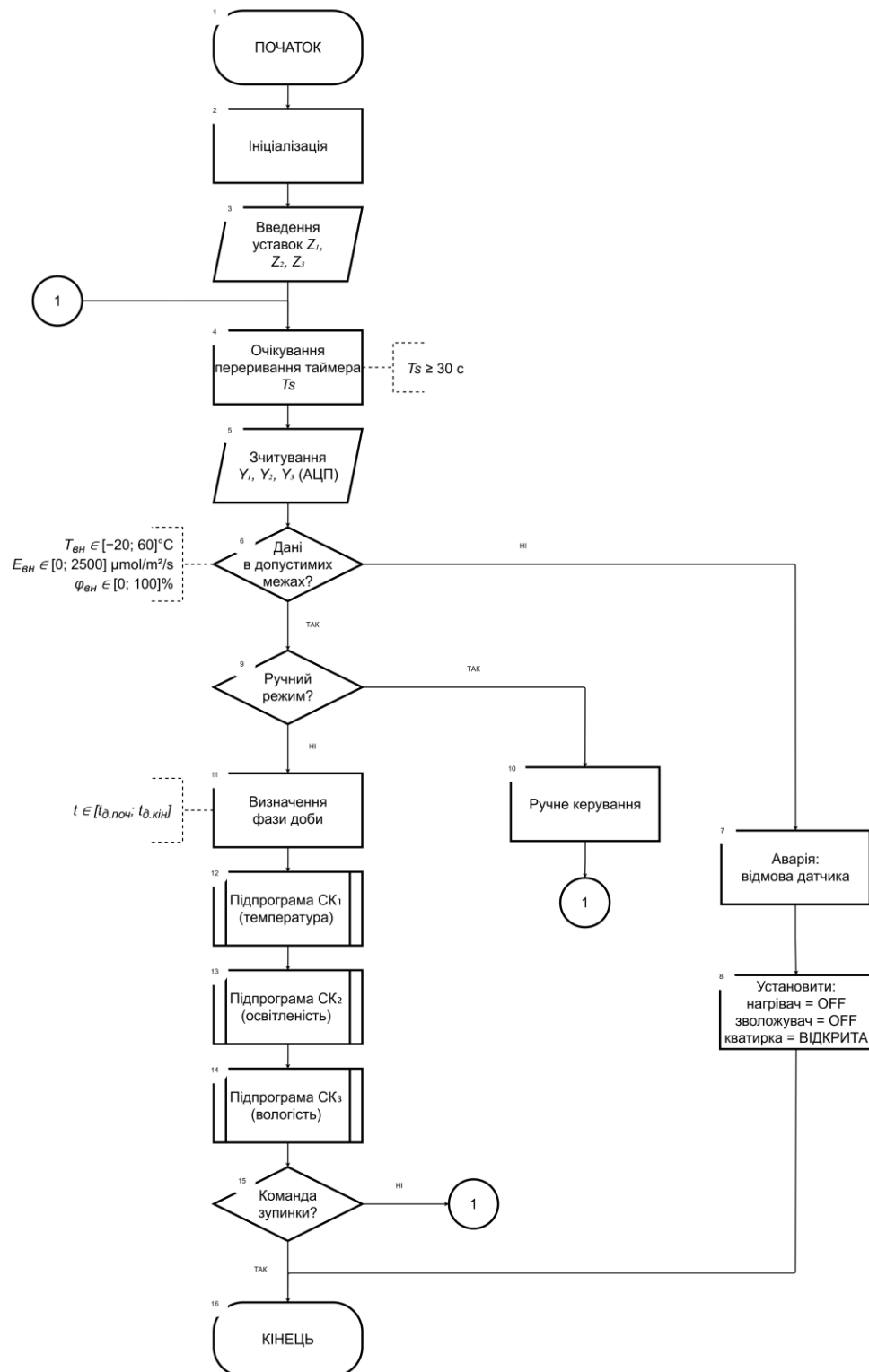


Рисунок 4.1 - Блок-схема головного алгоритму функціонування системи

Робота системи поділяється на дві фази: одноразову ініціалізацію при запуску та основний цикл, що повторюється протягом усієї роботи.

Ініціалізація виконується один раз і полягає в підготовці програмних компонентів та завантаженні уставок регулювання - цільових значень температури, освітленості й вологості - зі значень за замовчуванням або збережених налаштувань оператора. Після цього система переходить до основного циклу.

Основний цикл виконується періодично, із заданим інтервалом між проходами. На початку кожного проходу система очікує сигналу таймера, який задає період опитування, після чого зчитує поточні значення параметрів із давачів: температуру всередині теплиці, відносну вологість, освітленість та зовнішню температуру.

Наступним кроком є перевірка справності давачів. Отримані значення звіряються з фізично можливими межами вимірювання; якщо будь-яке значення виходить за ці межі, це свідчить про несправність давача.

У такому разі система формує сигнал аварії, переводить виконавчі механізми в безпечний стан і припиняє регулювання до усунення несправності. Ця перевірка передує регулюванню, оскільки керування за хибними показаннями недопустиме.

Якщо давачі справні, система визначає режим роботи. У ручному режимі автоматичне регулювання не виконується - виконавчими механізмами безпосередньо керує оператор через інтерфейс, а алгоритм одразу переходить до оновлення інтерфейсу. В автоматичному режимі система визначає поточну фазу доби (день або ніч) за таймером - від фази залежить вибір цільових значень (денних або нічних).

Після цього система послідовно викликає три підпрограми керування контурами: температури (СК₁), освітленості (СК₂) та вологості (СК₃). Кожна підпрограма обчислює межі зони гістерезису для свого параметра й формує команди відповідним виконавчим механізмам; детальну логіку підпрограм розглянуто в пунктах 4.2-4.4.

Після відпрацювання підпрограм система оновлює інтерфейс користувача, відображаючи поточні значення параметрів, стани виконавчих механізмів, активний режим керування та наявні сигнали аварій.

На завершення проходу перевіряється команда зупинки від оператора: за її наявності робота програми завершується, інакше система повертається до початку циклу й очікує наступного сигналу таймера.

Узагальнені параметри регулювання для всіх контурів наведено в таблиці 4.1. Вона показує, як з уставки та гістерезису кожного контуру утворюються конкретні пороги вмикання й вимикання виконавчих механізмів.

Таблиця 4.1 - Пороги спрацювання виконавчих механізмів за контурами

Параметр	Фаза	Уставка	Δ	Поріг ВКЛ	Поріг ВИМК	Механізм
Температура	день	22 °C	2 °C	нагрівач $T < 21$; кватирка $T > 23$	нагрівач $T > 23$; кватирка $T < 21$	Нагрівач / Кватирка
Температура	ніч	17 °C	2 °C	нагрівач $T < 16$; кватирка $T > 18$	нагрівач $T > 18$; кватирка $T < 16$	Нагрівач / Кватирка
Освітленість	день	1000	100	лампа $E < 950$	лампа $E > 1050$	Фітолампа
Освітленість	ніч	0	100	- (лампа вимкнена)	-	Фітолампа
Вологість	будь-яка	70 %	5 %	зволожувач $\varphi < 67.5$	зволожувач $\varphi > 72.5$	Зволожувач

Наведені в таблиці пороги є похідними від уставки й гістерезису: межі зони нечутливості обчислюються як уставка $\pm \Delta/2$. Контур температури має два механізми протилежної дії.

Контур освітленості вночі має нульову уставку, через що фітолампа лишається вимкненою незалежно від рівня PAR, забезпечуючи темновий період. Контур вологості від фази доби не залежить, тому має єдиний набір порогів.

Конкретні алгоритми, що реалізують ці пороги, розглянуто в наступних пунктах.

4.2. Алгоритм підпрограми СК₁ (керування температурою)

Підпрограма СК₁ реалізує керування температурою повітря теплиці. Це найскладніший із трьох контурів, оскільки він використовує два виконавчі механізми протилежної дії - нагрівач для підвищення температури та вентиляційну кватирку для її зниження. Блок-схему підпрограми, побудовану відповідно до ГОСТ 19.701, наведено на рис. 4.2.

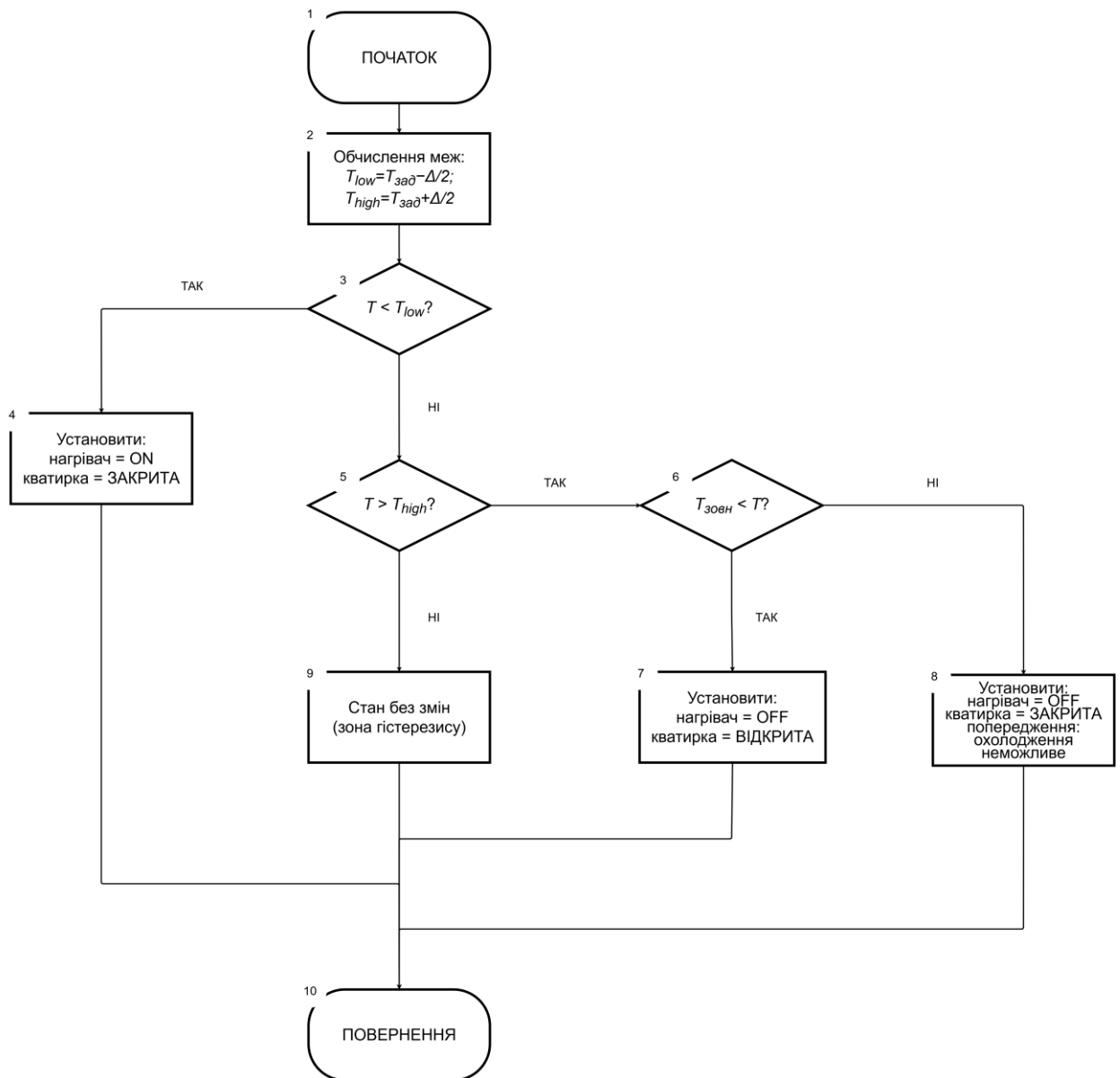


Рисунок 4.2 - Блок-схема підпрограми СК₁ «Керування температурою»

Підпрограма починається з обчислення меж зони гістерезису (блок 2). За поточною уставкою температури $T_{\text{зад}}$ - денною або нічною, залежно від фази доби - обчислюються нижня й верхня межі зони: $T_{\text{low}} = T_{\text{зад}} - \Delta T/2$ та $T_{\text{high}} =$

$T_{зад} + \Delta T/2$, де ΔT - гістерезис температурного контуру. Ці межі визначають пороги спрацювання механізмів. Для денної уставки $22\text{ }^{\circ}\text{C}$ при $\Delta T = 2\text{ }^{\circ}\text{C}$ межі становлять 21 і $23\text{ }^{\circ}\text{C}$.

Далі перевіряється, чи температура нижча за нижню межу (блок 3). Якщо $T < T_{low}$, температура впала нижче допустимої - потрібен обігрів: вмикається нагрівач, а квартира зачиняється (блок 4). Зачинення квартири разом із вмиканням нагрівача гарантує, що теплиця не втрачає тепло через вентиляцію під час обігріву.

Якщо температура не нижча за нижню межу, перевіряється, чи вона перевищує верхню (блок 5). Якщо T не перевищує T_{high} , температура перебуває в межах зони гістерезису - стан механізмів зберігається без змін (блок 9). Це зона нечутливості, у якій система не втручається.

Якщо ж температура перевищує верхню межу ($T > T_{high}$), потрібне охолодження, і тут перевіряється зовнішня температура (блок 6). Охолодження в системі досягається провітрюванням - відкриванням квартири, - а воно ефективне лише за умови, що зовнішнє повітря холодніше за внутрішнє. Тому перевіряється умова $T_{зовн} < T$, де T - поточна температура всередині теплиці.

Якщо зовнішня температура нижча за внутрішню, провітрювання здатне охолодити теплицю: нагрівач вимикається, квартира відкривається (блок 7). Якщо ж зовнішня температура не нижча за внутрішню ($T_{зовн} \geq T$), провітрювання охолодження не дасть. У цьому разі нагрівач лишається вимкненим, квартира зачиняється, а система формує попередження оператору про неможливість охолодження наявними засобами (блок 8). Квартирку зачинено навмисно: за вищої зовнішньої температури відкрита квартира лише впускала б тепліше повітря, погіршуючи ситуацію.

Незалежно від того, якою гілкою пройшло виконання, підпрограма завершується поверненням до головного алгоритму (блок 10).

4.3. Алгоритм підпрограми СК₂ (керування освітленістю)

Підпрограма СК₂ реалізує керування освітленістю за допомогою фітолампи - єдиного виконавчого механізму прямої дії, що підвищує рівень фотосинтетично активної радіації. Особливістю контуру є врахування фотоперіоду: фітолампа працює лише у світлій фазі доби, а вночі має бути вимкнена незалежно від рівня освітленості, щоб забезпечити рослинам темновий період.

Блок-схему підпрограми, побудовану відповідно до ГОСТ 19.701, наведено на рис. 4.3.

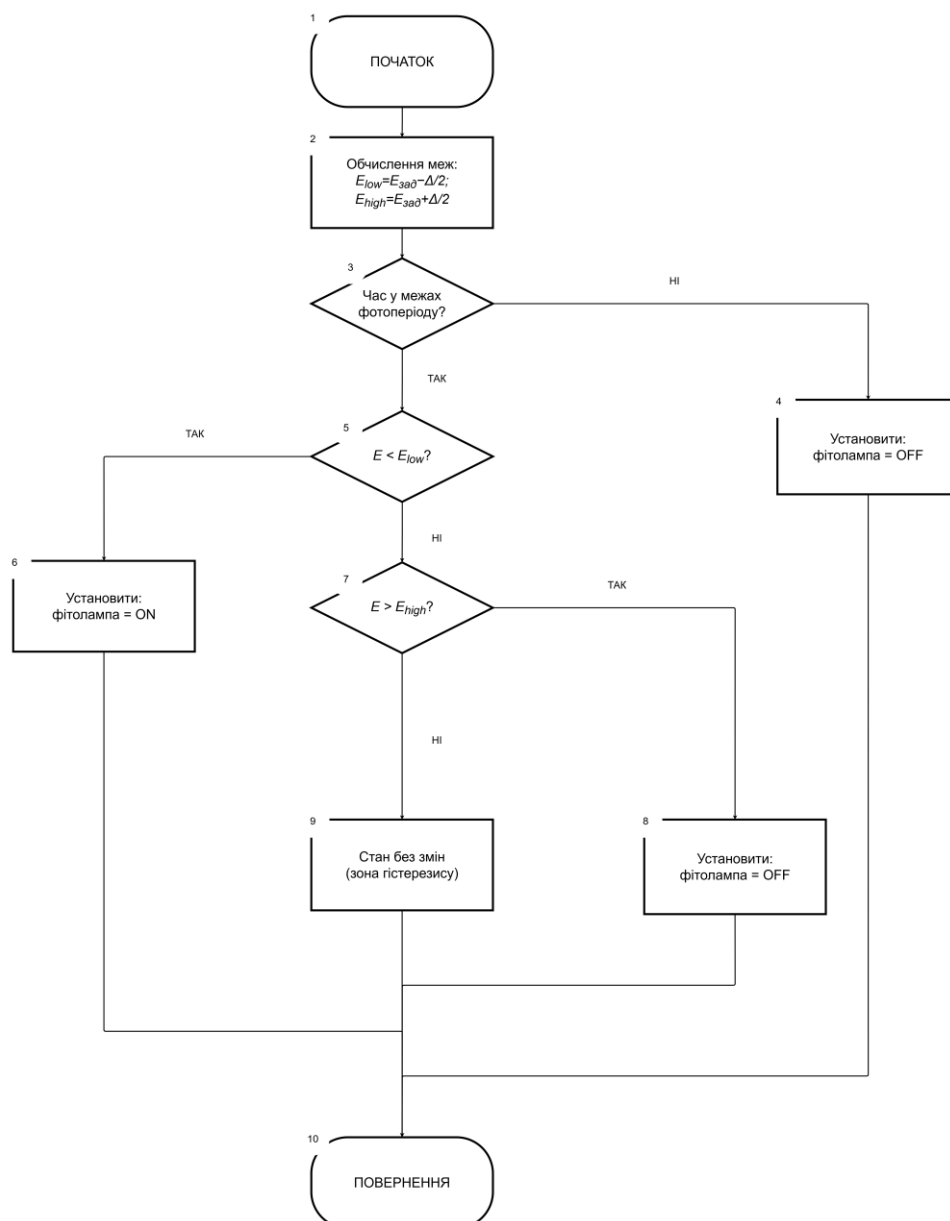


Рисунок 4.3 - Блок-схема підпрограми СК₂ «Керування освітленістю»

Підпрограма починається з обчислення меж зони гістерезису (блок 2): за уставкою освітленості $E_{зад}$ обчислюються нижня й верхня межі $E_{low} = E_{зад} - \Delta E/2$ та $E_{high} = E_{зад} + \Delta E/2$, де ΔE - гістерезис контуру освітленості. Для денної уставки $1000 \mu\text{mol}/\text{m}^2/\text{s}$ при $\Delta E = 100 \mu\text{mol}/\text{m}^2/\text{s}$ межі становлять 950 і $1050 \mu\text{mol}/\text{m}^2/\text{s}$.

Першою й визначальною є перевірка фотоперіоду (блок 3): чи перебуває поточний час у межах світлової фази доби. Якщо час поза межами фотоперіоду (нічна фаза), фітолампа безумовно вимикається (блок 4) незалежно від рівня освітленості, після чого підпрограма завершується. Завдяки цьому система не вмикає додаткове освітлення вночі, забезпечуючи рослинам темновий період.

Перевірка фотоперіоду має пріоритет над регулюванням за рівнем PAR - лише після неї, у світлій фазі, виконується двопозиційне керування.

У межах фотоперіоду перевіряється, чи освітленість нижча за нижню межу (блок 5). Якщо $E < E_{low}$, рівень PAR недостатній - фітолампа вмикається (блок 6). Якщо ні, перевіряється, чи освітленість перевищує верхню межу (блок 7). Якщо $E > E_{high}$, природного чи раніше доданого світла достатньо - фітолампа вимикається (блок 8). Якщо ж освітленість перебуває між межами, стан лампи зберігається без змін (блок 9) - це зона нечутливості.

Незалежно від гілки виконання підпрограма завершується поверненням до головного алгоритму (блок 10).

У програмній реалізації системи відсікання фітолампи в нічній фазі забезпечується не окремою перевіркою часу, а нульовою нічною уставкою освітленості ($E_{зад} = 0$). За нульової уставки нижня межа зони стає від'ємною, а оскільки рівень PAR не може бути меншим за нуль, умова вмикання лампи стає недосяжною - лампа лишається вимкненою протягом усієї нічної фази. Обидва підходи - явна перевірка фотоперіоду та нульова нічна уставка - дають однаковий результат: гарантовано вимкнену лампу вночі.

4.4. Алгоритм підпрограми СКз (керування вологістю)

Підпрограма СКз реалізує керування відносною вологістю за допомогою зволожувача - єдиного виконавчого механізму прямої дії, що підвищує вологість. Особливістю контуру є його координація з контуром температури: за відкритої вентиляційної квартирки зволожувач блокується, оскільки відкрита квартира виносить вологу назовні швидше, ніж зволожувач устигає її додавати.

Блок-схему підпрограми, побудовану відповідно до ГОСТ 19.701, наведено на рис. 4.4.

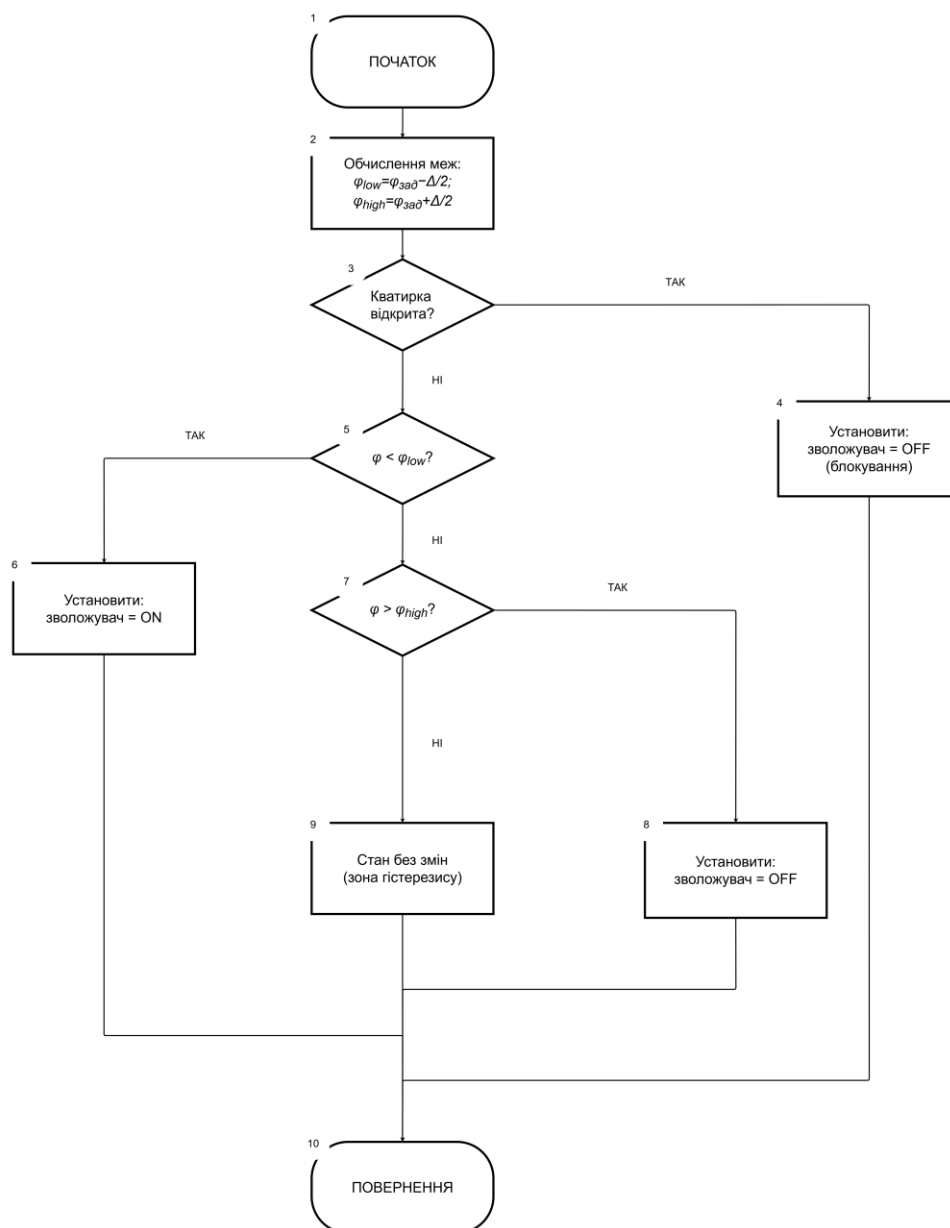


Рисунок 4.4 - Блок-схема підпрограми СКз «Керування вологістю»

Підпрограма починається з обчислення меж зони гістерезису (блок 2): за уставкою вологості $\varphi_{зад}$ обчислюються нижня й верхня межі $\varphi_{low} = \varphi_{зад} - \Delta\varphi/2$ та $\varphi_{high} = \varphi_{зад} + \Delta\varphi/2$, де $\Delta\varphi$ - гістерезис контуру вологості. Для уставки 70 % при $\Delta\varphi = 5$ % межі становлять 67,5 і 72,5 %.

Першою є перевірка стану вентиляційної квартирки (блок 3). Якщо квартирка відкрита, зволожувач безумовно вимикається (блок 4) незалежно від рівня вологості, після чого підпрограма завершується.

Це є реалізація правила координації: доки квартирка відкрита для охолодження, зволоження марне, бо волога виходить назовні разом із провітрюваним повітрям. Перевірка стану квартирки має пріоритет над регулюванням за рівнем вологості - лише за зачиненої квартирки виконується двопозиційне керування.

Якщо квартирка зачинена, перевіряється, чи вологість нижча за нижню межу (блок 5). Якщо $\varphi < \varphi_{low}$, повітря надто сухе - зволожувач вмикається (блок 6).

Якщо ні, перевіряється, чи вологість перевищує верхню межу (блок 7). Якщо $\varphi > \varphi_{high}$, вологість достатня - зволожувач вимикається (блок 8). Якщо ж вологість перебуває між межами, стан зволожувача зберігається без змін (блок 9) - це зона нечутливості.

Незалежно від гілки виконання підпрограма завершується поверненням до головного алгоритму (блок 10).

У програмній реалізації системи перевірка стану квартирки винесена в окремий етап координації, що виконується після відпрацювання всіх трьох контурів, але до передачі команд механізмам.

Завдяки цьому координація коригує рішення регулятора вологості перед його застосуванням: якщо квартирка відкрита, команда на ввімкнення зволожувача замінюється на вимкнення, тож фактично зволожувач не вмикається. Логічний результат збігається з наведеною блок-схемою.

4.5. Алгоритм обробки аварій і безпечного стану

Окрім регулювання, система виконує контроль власної справності та реагує на нештатні ситуації. Існують два рівні таких подій - попередження та аварія. Вони відрізняються причиною, наслідками й пріоритетом.

Попередження сигналізує про ситуацію, у якій обладнання справне, але система не здатна досягти мети наявними засобами. Єдиний такий випадок у системі - неможливість охолодження, коли температура перевищує норму, а зовнішня температура не нижча за внутрішню (підрозділ 4.2). Провітрювання за таких умов неефективне, тому система зачиняє квартиру й повідомляє оператора, проте продовжує нормальну роботу: попередження не зупиняє регулювання.

Аварія сигналізує про несправність обладнання, за якої подальше регулювання недопустиме. У системі аварія формується при відмові датчика, що виявляється автономно: кожне зчитане значення звіряється з фізично можливими межами вимірювання, і вихід за них означає несправність датчика. Оскільки регулювання за хибними показаннями могло б призвести до неправильних дій механізмів, при аварії система переводить виконавчі механізми в безпечний стан і припиняє автоматичне регулювання.

Безпечний стан визначено так: нагрівач вимкнено, зволожувач вимкнено, квартира відкрита. Логіка цього вибору - пасивна безпека: вимкнені нагрівач і зволожувач не можуть довести параметри до небезпечних значень, а відкрита квартира забезпечує природний повітрообмін із зовнішнім середовищем.

У такому стані теплиця залишається без активного керування, але захищеною від перегріву чи перезволоження, доки несправність не буде усунена.

Між двома рівнями встановлено пріоритет: аварія важливіша за попередження. Якщо система вже перебуває в стані аварії, попередження не знижує рівень тривоги, оскільки несправність обладнання є серйознішою проблемою, ніж тимчасова неможливість охолодження.

Перевірка справності давачів виконується на початку кожного циклу, перед регулюванням, тому аварійна ситуація розпізнається раніше, ніж формуються будь-які керувальні рішення.

4.6. Програмна реалізація системи

Розроблені алгоритми втілено у програмній моделі системи контролю мікроклімату. Модель імітує роботу теплиці та системи керування програмно, без фізичного обладнання, що дає змогу перевірити логіку алгоритмів у різних умовах. Модель написано мовою Python, а графічний інтерфейс оператора побудовано на стандартній бібліотеці Tkinter. Вибір Python зумовлений простотою реалізації алгоритмічної логіки та наявністю готових засобів побудови інтерфейсу й графіків, що дає змогу зосередитися на логіці керування, а не на технічних деталях реалізації.

Програму побудовано за модульним принципом: кожна частина системи відповідає окремому модулю, а взаємодія між ними повторює структуру, наведену на структурній схемі. Модель об'єкта керування імітує поведінку теплиці й реакцію параметрів на роботу виконавчих механізмів; імітатори давачів додають до справжніх значень шум вимірювання й зрідка видають збій, що дає змогу перевірити реакцію системи на несправність. Окремі модулі реалізують двопозиційні регулятори, ядро керування (головний цикл), сховище уставок та підсистему сигналізації. Центральним є модуль ядра, який на кожному такті послідовно виконує кроки головного алгоритму - зчитування давачів, перевірку справності, визначення фази доби, виклик трьох підпрограм керування контурами та подання команд виконавчим механізмам, тобто реалізує в коді блок-схему, наведену на рис. 4.1.

Підпрограми керування контурами реалізовано як окремі функції двопозиційного регулювання з гістерезисом, що відповідають блок-схемам рис. 4.2-4.4. Самі регулятори побудовано як прості функції, що за вимірним значенням, уставкою та гістерезисом повертають рішення про вмикання чи

вимикання механізму, не залежачи від інших частин системи. Стартові значення уставок і гістерезисів задано в окремому конфігураційному модулі, що слугує єдиним джерелом початкових параметрів для всіх контурів і відповідає зведеній таблиці параметрів регулювання. Уставки, які задає оператор під час роботи, зберігаються окремо - у сховищі уставок, куди записуються лише після перевірки на знаходження в допустимому діапазоні.

У двох випадках програмна реалізація відрізняється від блок-схем за побудовою, зберігаючи при цьому ту саму логіку роботи. По-перше, відсікання фітолампи в нічній фазі реалізовано не окремою перевіркою часу всередині підпрограми СК₂, а через нульову нічну уставку освітленості: за нульової уставки умова вмикання лампи стає недосяжною, тож лампа лишається вимкненою незалежно від рівня PAR. По-друге, правило координації контурів - блокування зволожувача за відкритої квартирки - винесено в окремий етап узгодження в ядрі керування, а не реалізовано всередині підпрограми СК₃, як показано на блок-схемі. В обох випадках блок-схема відображає логіку наочно, а програмна реалізація досягає того самого результату ефективнішим способом, уникаючи дублювання перевірок.

Взаємодію оператора із системою забезпечує графічний інтерфейс, який відображає поточні значення параметрів і стани виконавчих механізмів, дає змогу задавати уставки й межі фази доби в допустимих діапазонах, перемикає режим роботи та показує повідомлення про попередження й аварії. Загальний вигляд вікна інтерфейсу наведено на рис. 4.5.

Для спостереження за роботою системи модель веде запис значень параметрів на кожному такті й дає змогу побудувати графіки їх зміни в часі. На рис. 4.6 наведено такі графіки зміни параметрів трьох контурів у часі. На них видно, як кожен параметр від початкового значення виходить на задану уставку й надалі утримується в межах коридору гістерезису навколо неї.

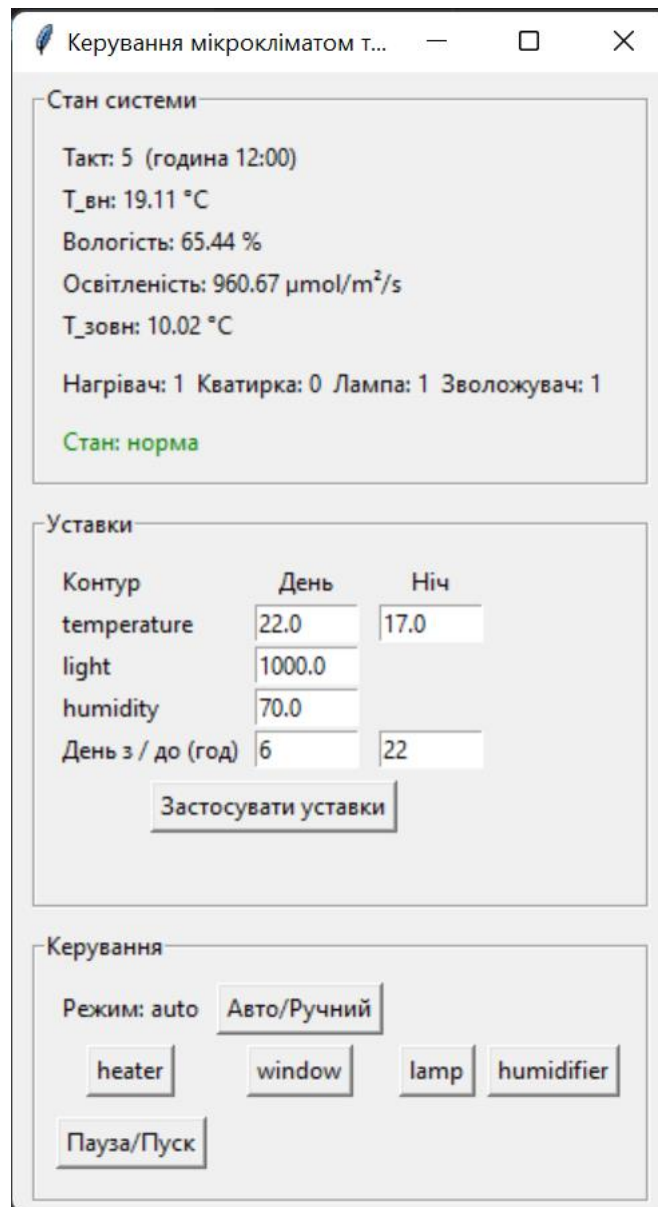


Рисунок 4.5 - Головне вікно інтерфейсу оператора

Графіки також наочно демонструють відмінність у динаміці контурів, відзначену в розділі 2: температура й вологість, як інерційні параметри, виходять на уставку поступово й коливаються в межах коридору, тоді як освітленість досягає заданого рівня практично миттєво після вмикання фітолампи. На графіку вологості помітно, що вона час від часу опускається нижче коридору гістерезису: це відбувається тоді, коли контур температури відкриває кватирку для провітрювання, а зволожувач при цьому заблоковано правилом координації, через що вологість тимчасово знижується.

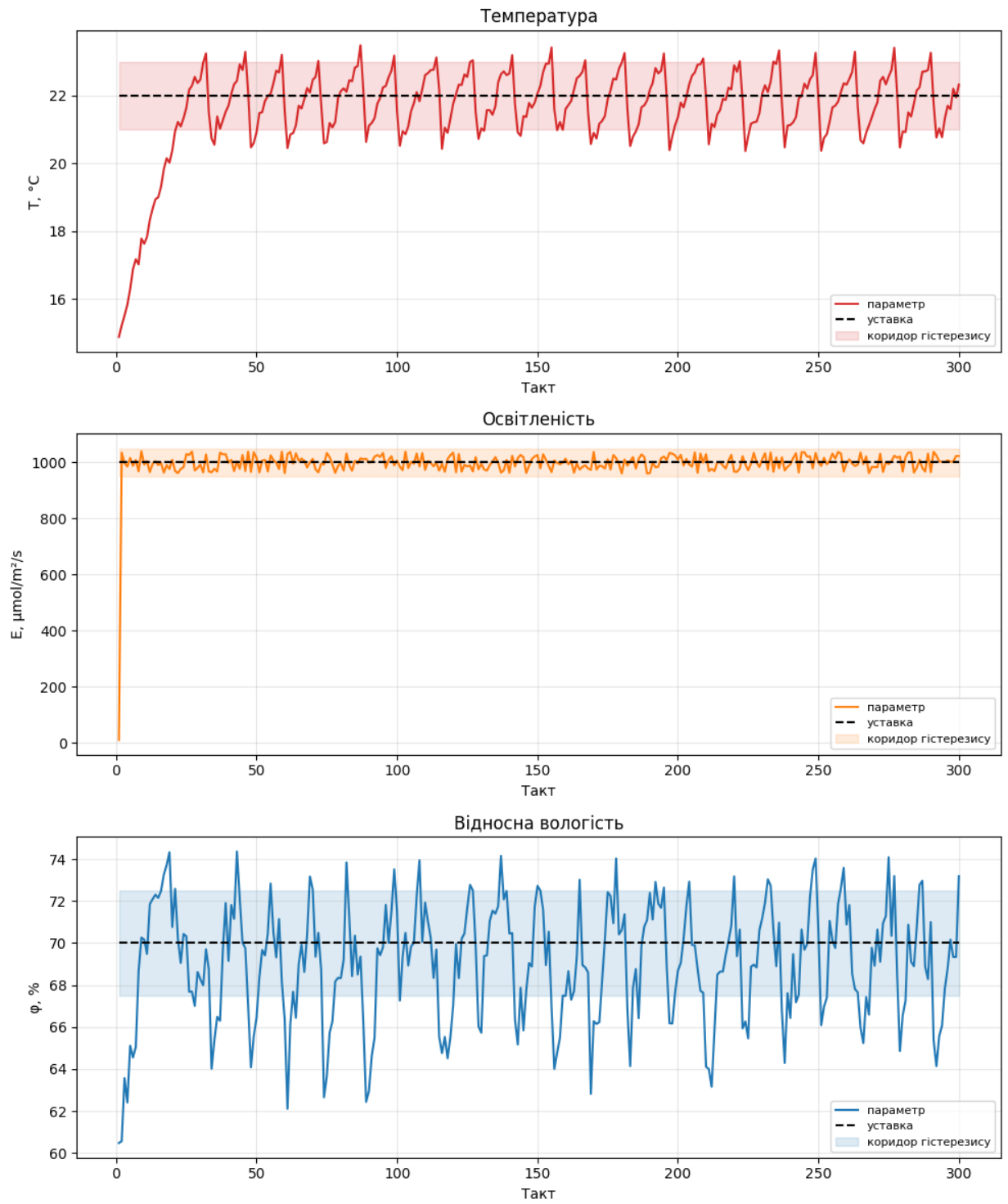


Рисунок 4.6 - Зміна параметрів мікроклімату в часі: утримання в коридорі гістерезису

Висновки до розділу 4

У розділі розроблено алгоритмічне забезпечення системи керування мікрокліматом теплиці, що реалізує двопозиційне регулювання з гістерезисом, обґрунтоване в розділі 3.

Розроблено головний алгоритм функціонування системи у вигляді циклічного процесу з одноразовою ініціалізацією. Визначено послідовність дій кожного циклу: зчитування показань давачів, перевірка їх справності, вибір режиму роботи й фази доби та виклик підпрограм керування контурами. Перевірку справності давачів розміщено перед регулюванням, що унеможливорює керування за хибними показаннями.

Розроблено алгоритми трьох підпрограм керування контурами у вигляді блок-схем за ДСТУ 7363:2013. Для контуру температури (СК₁) реалізовано керування двома механізмами протилежної дії з урахуванням зовнішньої температури при охолодженні. Для контуру освітленості (СК₂) реалізовано регулювання з перевіркою фотоперіоду, що гарантує темновий період. Для контуру вологості (СК₃) реалізовано регулювання з координацією, що блокує зволожувач за відкритої квартирки.

Показано, що в усіх контурах пороги спрацювання виконавчих механізмів є похідними від уставки й гістерезису та обчислюються як уставка $\pm \Delta/2$, а зведені значення порогів наведено в таблиці параметрів регулювання.

Розроблено механізм обробки нештатних ситуацій із розмежуванням на попередження та аварію. Визначено, що попередження формується за неможливості охолодження й не зупиняє роботу, тоді як аварія при відмові давача переводить систему в безпечний стан із припиненням регулювання. Обґрунтовано склад безпечного стану за принципом пасивної безпеки та пріоритет аварії над попередженням. Розроблені алгоритми є основою для програмної реалізації системи, розглянутої в наступному розділі.

Алгоритмічне забезпечення реалізовано у вигляді програмної моделі мовою Python із графічним інтерфейсом оператора. Підтверджено відповідність програмної реалізації розробленим алгоритмам, а на графіках

зміни параметрів показано, що система виводить кожен параметр на задану уставку й утримує його в межах коридору гістерезису.

ВИСНОВКИ

У роботі на тему «Автоматизована система контролю мікроклімату» досягнуто поставленої мети - створено автоматизовану систему керування мікрокліматом теплиці у вигляді програмної моделі, яка підтримує температуру, освітленість і вологість повітря в заданих межах за допомогою двопозиційного регулювання та узгоджує роботу виконавчих механізмів. Усі поставлені завдання виконано.

У результаті аналізу наукової літератури та наявних рішень встановлено, що теплиця є складним багатопараметричним об'єктом керування, основними параметрами якого є температура, відносна вологість та освітленість. Показано, що ці параметри взаємопов'язані й потребують узгодженого керування, а освітленість для рослин слід оцінювати через фотосинтетично активну радіацію, а не через освітленість у люксах. Порівняння законів регулювання та наявних систем автоматизації показало, що для теплиці доцільним є двопозиційне (релейне) регулювання, а для невеликих господарств раціональною є побудова системи на основі доступних цифрових давачів і мікроконтролера. Це визначило загальний підхід до подальшого проектування.

Обґрунтовано параметри регулювання та вимоги до системи. Для кожного з трьох контурів визначено робочий діапазон, цільові значення (уставки) та гістерезис, причому для температури й освітленості передбачено окремі денні й нічні уставки відповідно до добового ритму рослин, а темновий період реалізовано через нульову нічну уставку освітленості. Показано, що гістерезис кожного контуру обирається більшим за похибку відповідного давача, що забезпечує стійкість регулювання до шуму вимірювання. Обґрунтовано також взаємозв'язок контурів температури та вологості й потребу в їх координації.

Для прив'язки моделі до реальних умов обрано конкретні технічні засоби: цифровий давач температури та вологості Sensirion SHT35, квантовий давач фотосинтетично активної радіації Apogee SQ-521, мікроконтролер ESP32 та релейний модуль з опторозв'язкою. Характеристики цих засобів покладено в основу параметрів програмної моделі, завдяки чому модель відповідає реально існуючому обладнанню й може бути за потреби перенесена на фізичні засоби без зміни логіки керування.

Розроблено структуру системи та комплект проектної документації: структурну схему, функціональну схему автоматизації, схему інформаційних потоків та UML-діаграми системи. Окремо опрацьовано правила координації контурів, що усувають суперечливу роботу виконавчих механізмів - а саме блокування зволожувача за відкритої вентиляційної квартирки.

Розроблено алгоритмічне забезпечення системи у вигляді головного алгоритму функціонування та трьох підпрограм керування контурами температури, освітленості й вологості, поданих блок-схемами. Передбачено також алгоритм обробки нештатних ситуацій із розмежуванням на попередження (обладнання справне, але мети досягти неможливо) та аварію (несправність давача), із переведенням системи в безпечний стан.

Розроблені алгоритми втілено у програмній моделі мовою Python із графічним інтерфейсом оператора на основі бібліотеки Tkinter. Підтверджено відповідність програмної реалізації розробленим алгоритмам, а на графіках зміни параметрів показано, що система виводить кожен параметр на задану уставку й утримує його в межах коридору гістерезису, причому динаміка контурів відповідає очікуваній: температура й вологість змінюються поступово, а освітленість - практично миттєво.

Практичне значення роботи полягає в тому, що розроблена система та закладені в ній алгоритми керування можуть бути використані як основа для побудови реальних систем автоматизації теплиць на доступних мікроконтролерах і цифрових давачах, а також як навчальний засіб для вивчення принципів двопозиційного регулювання.

Напрямами подальшого розвитку системи можуть бути врахування стадій розвитку рослин із диференціацією уставок, додавання керування концентрацією вуглекислого газу та поливом, а також перенесення розробленого програмного забезпечення на фізичний мікроконтролер для апробації в реальній теплиці.

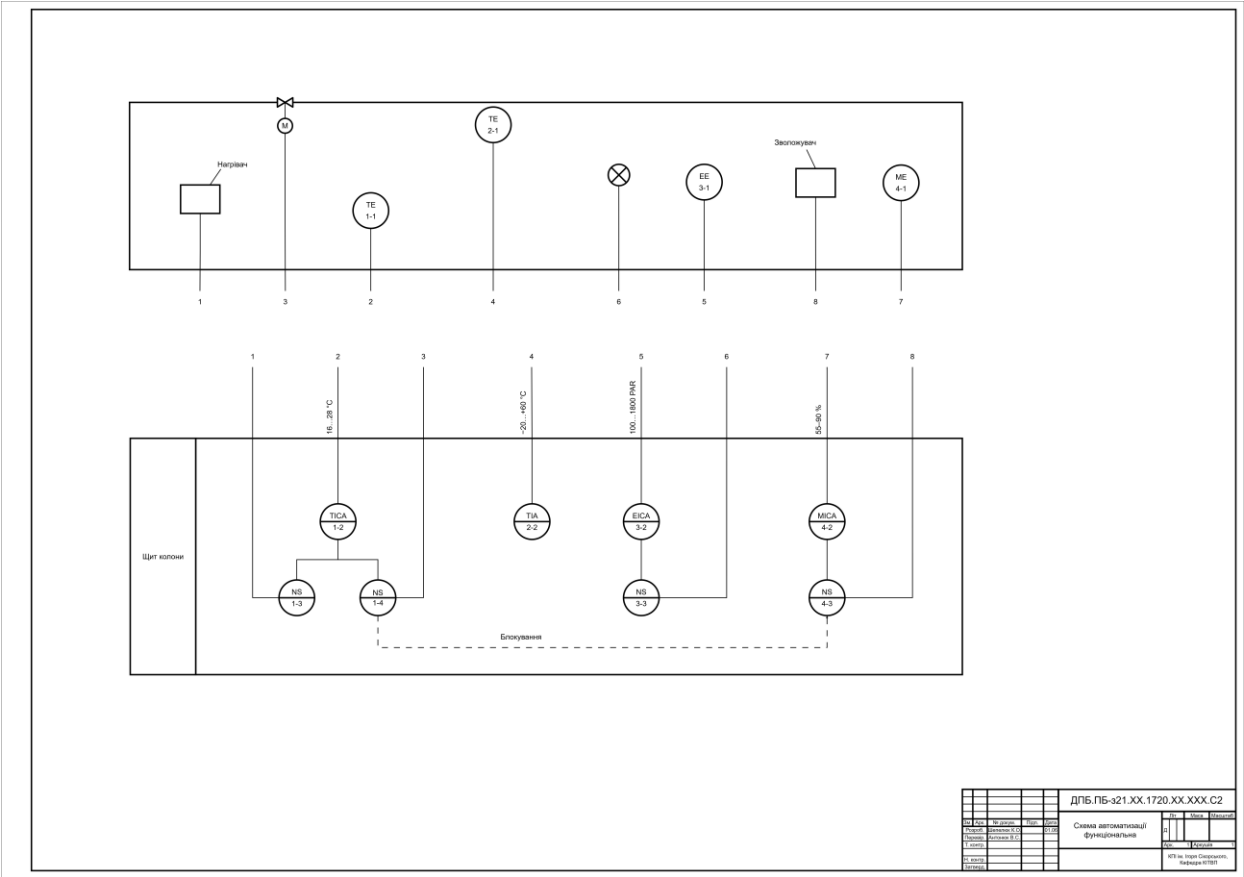
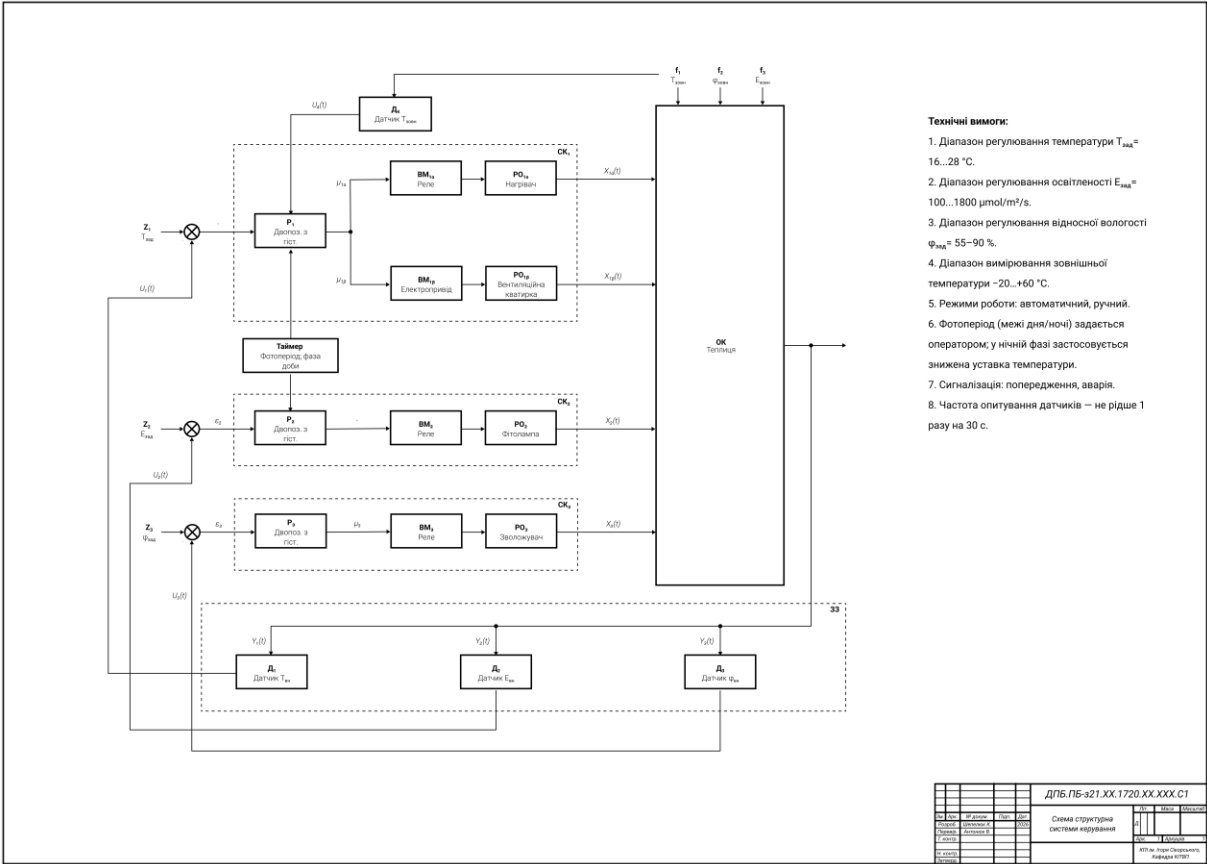
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

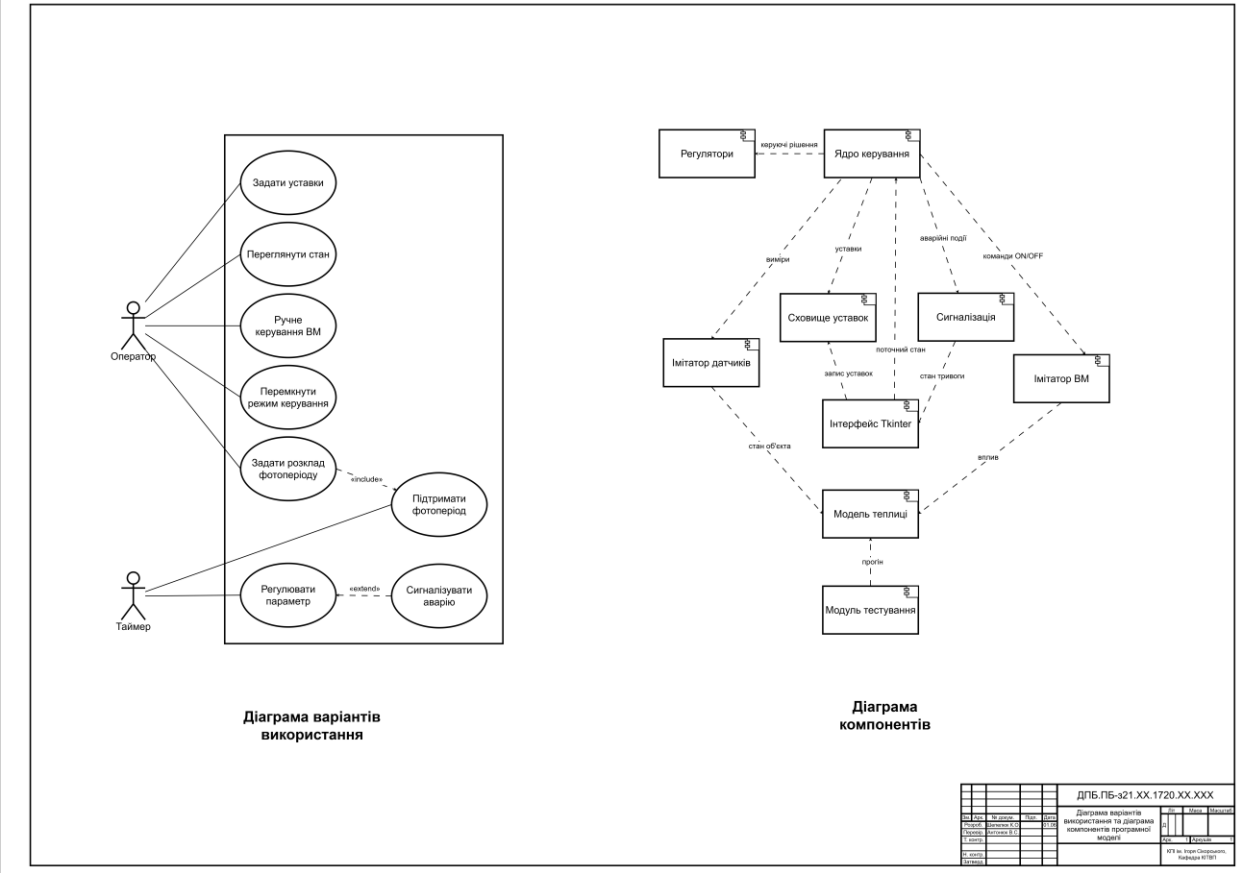
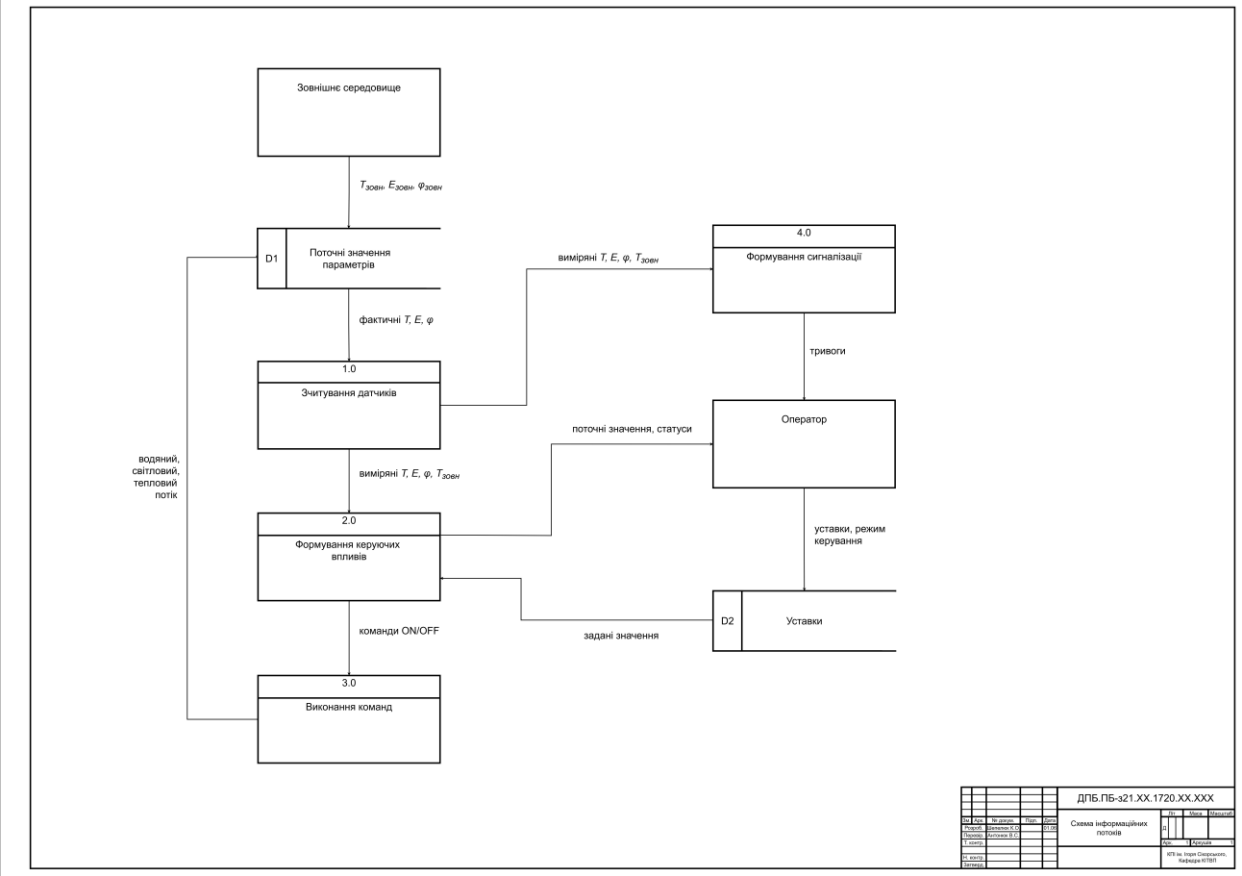
1. Kokieva, G. E., Trofimova, V. S., & Fedorov, I. R. (2020, December). Greenhouse microclimate control. In Iop conference series: Materials science and engineering (Vol. 1001, No. 1, p. 012136). IOP Publishing.
2. СУВОРІН, О., КАБАЧІЙ, В., КУЛИК, Я., & ОВЧИННИКОВ, К. (2025). Аналіз параметрів мікроклімату для управління теплицею. Herald of Khmelnytskyi National University. Technical sciences, 353(3.2), 315-324.
3. Антонюк В.С. Методологія наукових досліджень: [Текст] : навч. посіб./ В.С. Антонюк, Л.Г. Полонський, В.І. Аверченков, Ю.А. Малахов. - К.: НТУУ «КПІ», 2015. - 276 с. - Бібліогр. : с. 254-262. <https://oоер.kpi.ua/downloads/disc/anton.pdf>
4. Методологія наукових досліджень у галузі: практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / уклад.: Н.І. Бурау, В.С. Антонюк, Д.О. Півторак. - Електронні текстові дані (1 файл: 0,4 Мбайт). - КПІ ім. Ігоря Сікорського, 2021. - 58 с. https://ela.kpi.ua/bitstream/123456789/45871/1/Metod_doslid.pdf
5. Мережаний Ю.Г. Вимірювання аерозольних часток у повітрі в умовах складального приміщення прецизійного приладобудівного виробництва // Процеси механічної обробки в машинобудуванні: Зб. наук. пр. Житомир : 2011. - Вип. 10. - С. 42 - 53.
6. Антонюк В.С., Мережаний Ю.Г., Пономаренко А.І. Автоматизований моніторинг та керування мікрокліматом виробничого приміщення складання прецизійних приладів // Резание и инструмент в технологических системах. - Межд. науч.-техн. сборник. - Харьков: НТУ "ХПИ", 2011, Вып.80. - С. 3 - 14.
7. McCree, K. J. (1972). Test of current definitions of photosynthetically active radiation against leaf photosynthesis data. Agricultural meteorology, 10, 443-453.

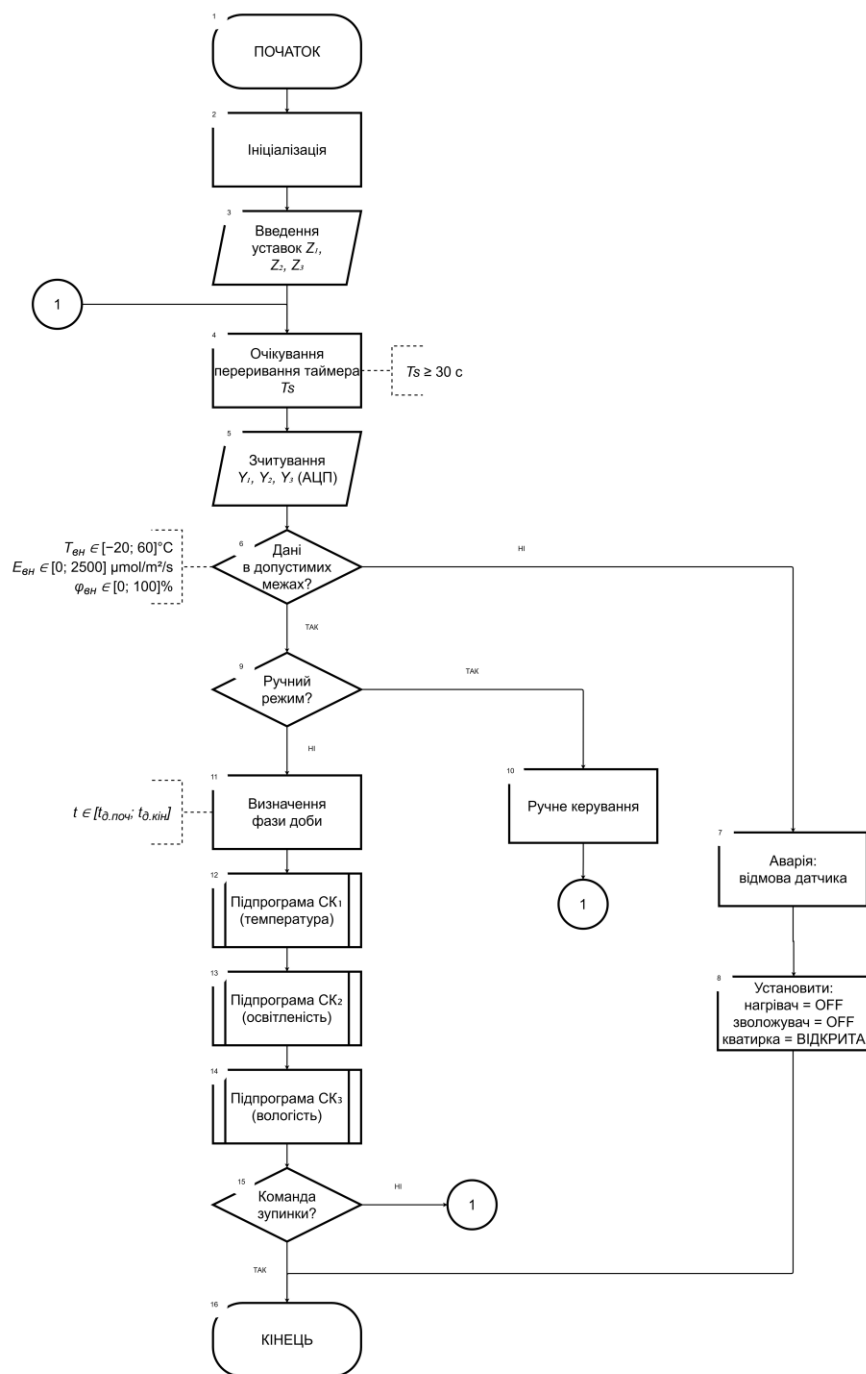
8. Попович, М. Г., & Ковальчук, О. В. (2007). Теорія автоматичного регулювання: Підручник. К.: Либідь, 1007, 544.
9. Ardiansah, I., Bafdal, N., Suryadi, E., & Bono, A. (2020). Greenhouse monitoring and automation using Arduino: a review on precision farming and internet of things (IoT). *Int. J. Adv. Sci. Eng. Inf. Technol*, 10(2), 703-709.
10. Губарев, О., & Синицина, Є. (2024). Мехатронна система керування вологістю мікроклімату теплиці. *Вісник машинобудування та транспорту*, 20(2), 38-47.
11. Priva Connex - climate and process computer [Електронний ресурс] : офіційний сайт компанії Priva. - Режим доступу: <https://www.priva.com/horticulture/solutions/climate-and-process-computers/priva-connex> (дата звернення: 08.06.2026).
12. Argus TITAN greenhouse control system [Електронний ресурс] : офіційний сайт компанії Argus Control Systems. - Режим доступу: <https://arguscontrols.com/products-and-solutions/control-systems/titan-system> (дата звернення: 08.06.2026).

ДОДАТКИ

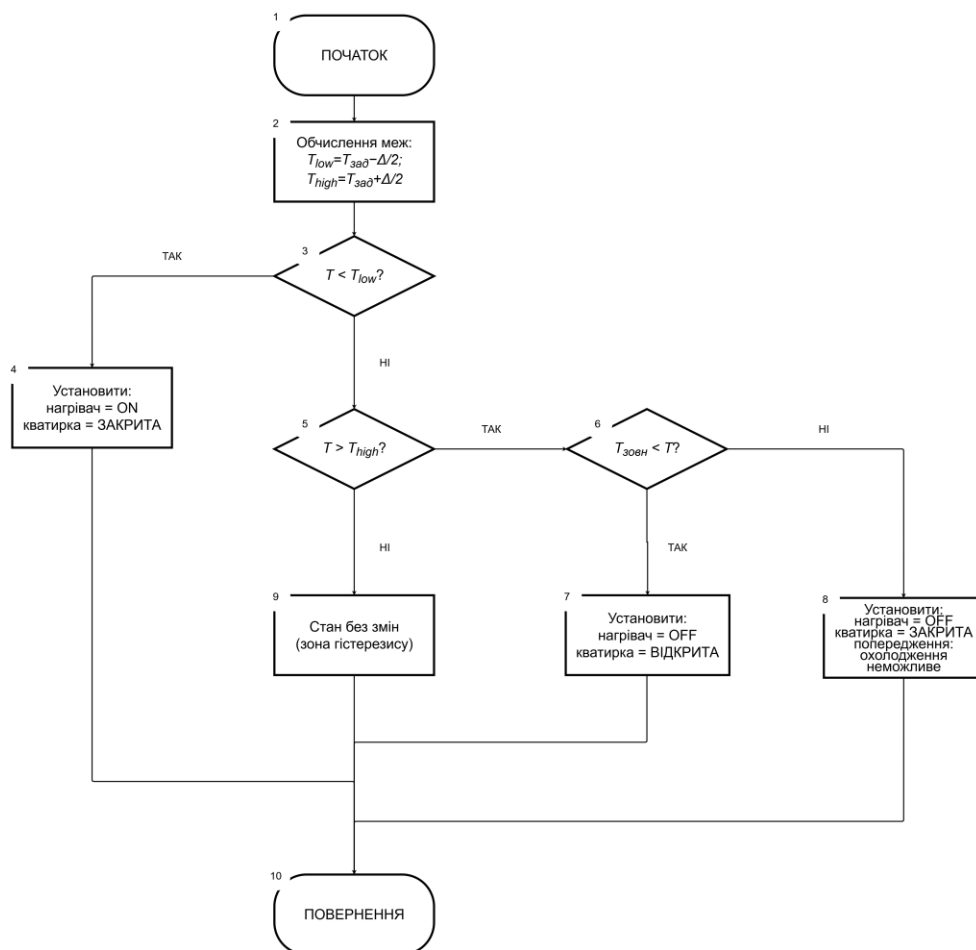
ДОДАТОК А. (Підписати кожен рисунок окремо)



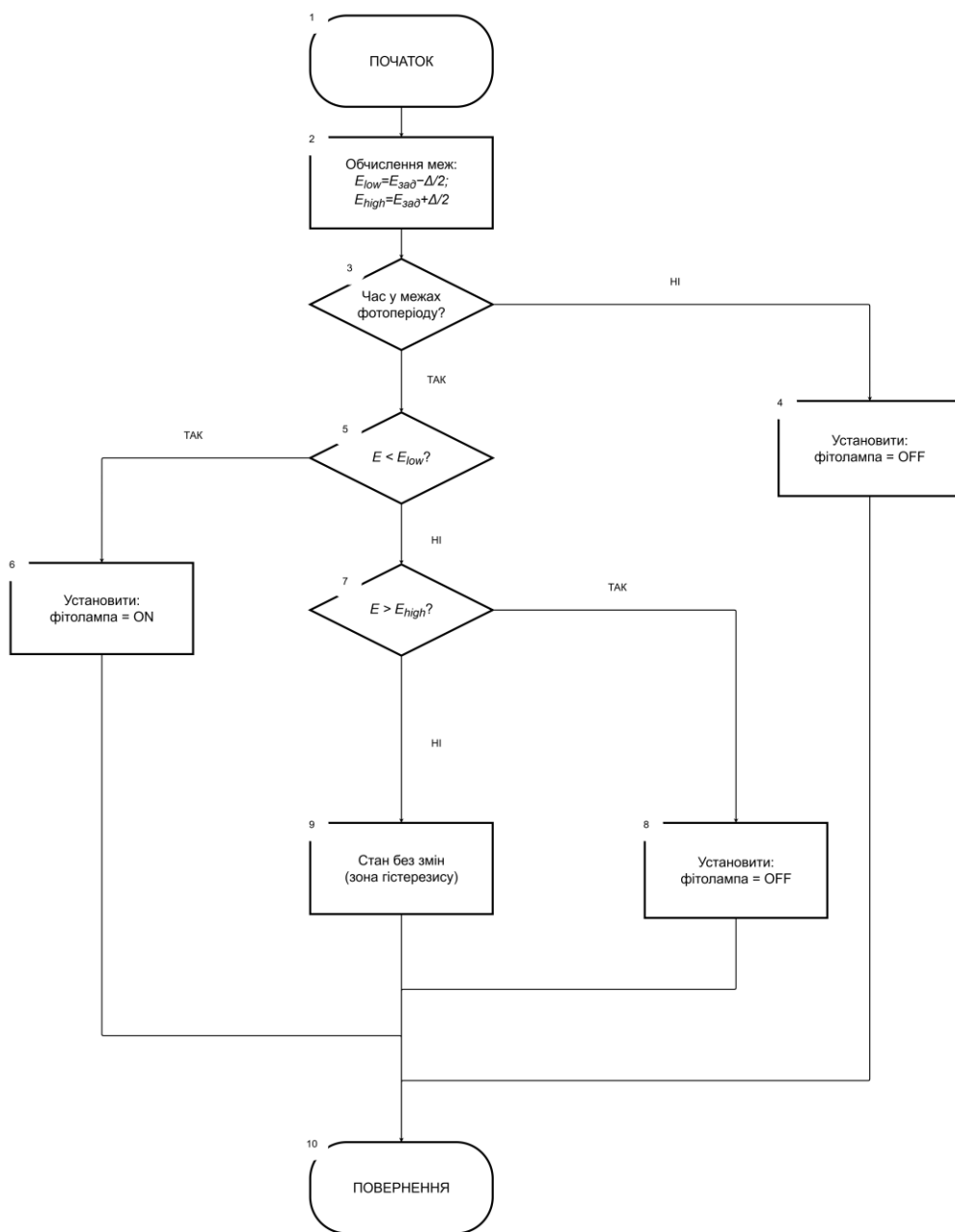




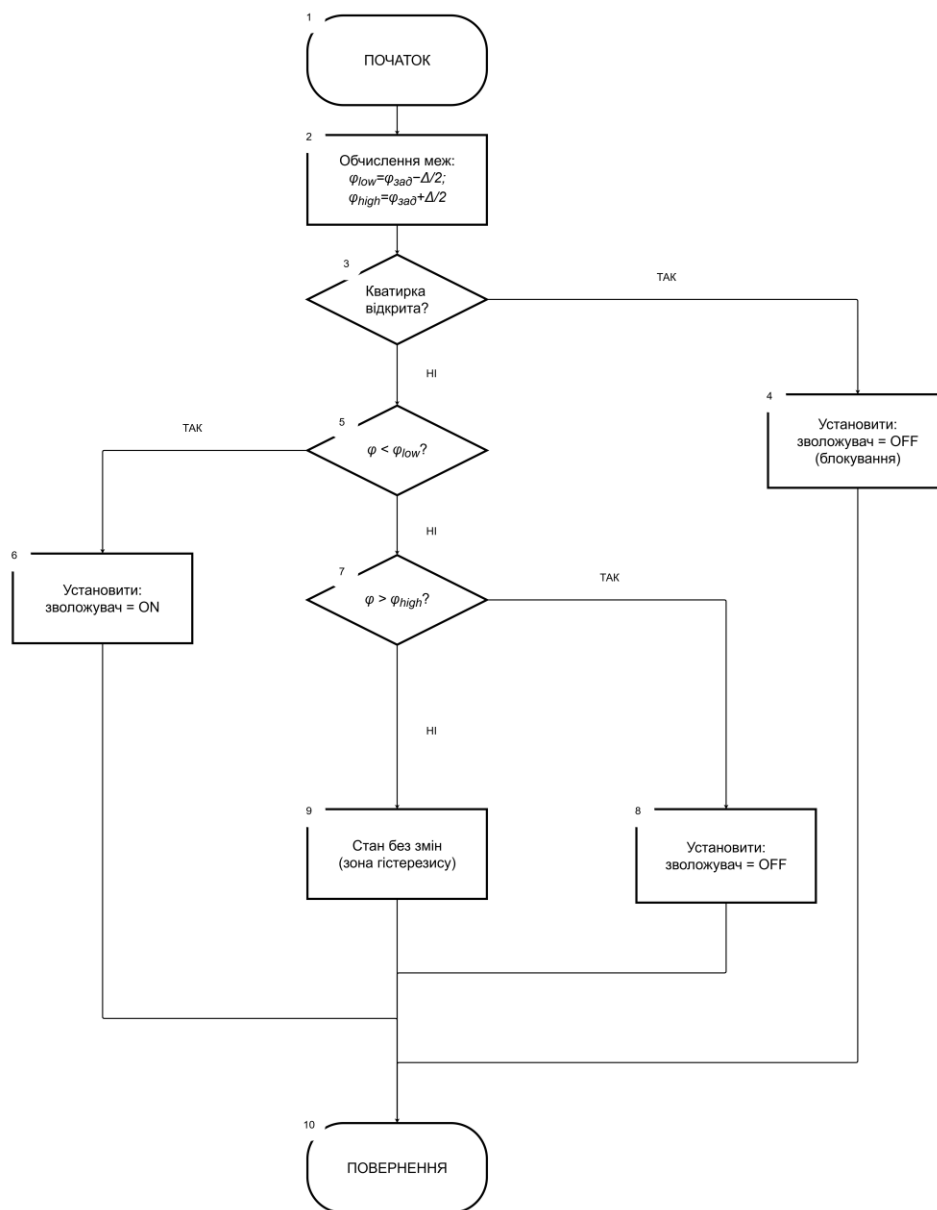
ДПБ.ПБ-321.XX.1720.XX.XXX									
№	Рік	№ докум.	Підп.	Дата	Схема алгоритму керування Головна програма.				
Розроб.	Детопенко В.О.			01.06					
Перевір.	Антоненко В.О.				Арх. 1 Апрошів				
Т.контр.									
Т.контр.					КПІ ім. Івора Сікорського, Кафедра КІТБТ				
Затверд.									



ДПБ.ПБ-321.XX.1720.XX.XXX									
№	Рік	№ докум.	Підп.	Дата	Схема алгоритму				
Розроб.	Депован.	В.О.		01.08	підпрограми СК ₁ (керування				
Перевір.	Виконан.	В.О.			температурою)				
Т.контр.					Арх.	1	Арх.	1	
Н.контр.					КПІ ім. Іллія Сікорського,				
Затверд.					Київська НТБТ				



ДПБ.ПБ-321.XX.1720.XX.XXX									
№	Рік	№ докум.	Підп.	Дата	Схема алгоритму				
Розроб.	Детальні	В.О.		01.08	підпрограми СК ₂ (керування				
Перевір.	Виконані	В.О.			освітленістю)				
Т.контр.					Арх.	1	Арх.	1	
П.контр.					КПІ ім. Іллія Сікорського,				
Затверд.					Київська НТБП				



ДПБ.ПБ-321.XX.1720.XX.XXX									
№	Рік	№ докум.	Підп.	Дата	Схема алгоритму				
Розроб.	Детальні	В.О.		01.08	підпрограми СКЗ (керування				
Перевір.	Автоматич.	В.О.			вологістю)				
Т.контр.					Арх.	1	Формат	1	
Н.контр.					КПІ ім. Іллії Сікорського,				
Затверд.					Київська 117Б1				

Виконавчі механізми										
1-3	—	—	—	Щит керування	Реле проміжне (пускове) керування нагрівачем; канал релейного модуля з оптороз'язкою; вихід дискретний (ON/OFF)	Модуль 4-кан., 5 В	—	—	*	—
1-4	—	—	—	Щит керування	Реле проміжне (пускове) керування електроприводом квартирки; канал релейного модуля; вихід дискретний (ON/OFF)	Модуль 4-кан., 5 В	—	—	*	—
3-3	—	—	—	Щит керування	Реле проміжне (пускове) керування фітолампкою; канал релейного модуля; вихід дискретний (ON/OFF)	Модуль 4-кан., 5 В	—	—	*	—
4-3	—	—	—	Щит керування	Реле проміжне (пускове) керування зволожувачем; канал релейного модуля; вихід дискретний (ON/OFF)	Модуль 4-кан., 5 В	—	—	*	—
—	—	—	—	У теплиці	Електропривід вентиляційної квартирки, лінійний; двопозиційний (відкрито/закрито)	—	—	—	1	—
Засоби обчислювальної техніки										
—	—	—	—	Щит керування	Мікроконтролер — опитування давачів, двопозиційне регулювання, сигналізація (контурі 1-2, 2-2, 3-2, 4-2); інтерфейс I ² C, дискретні виходи, USB; період опитування 30 с	ESP32 WROOM-32	—	Espressif	1	—
—	—	—	—	Щит керування	Модуль релейний чотириканальний з оптороз'язкою; комутація мережевого навантаження виконавчих механізмів	Модуль 4-кан., 5 В	—	—	1	—
Регулювальні органи										
—	—	—	—	У теплиці	Нагрівач електричний; двопозиційне керування (ON/OFF)	—	—	—	1	—
—	—	—	—	У теплиці	Квартирка вентиляційна із заслінкою; двопозиційна (відкрито/закрито)	—	—	—	1	—
—	—	—	—	У теплиці	Фітолампа (джерело світла); двопозиційне керування (ON/OFF)	—	—	—	1	—
—	—	—	—	У теплиці	Зволожувач повітря; двопозиційне керування (ON/OFF)	—	—	—	1	—
Щити та пульти										
—	—	—	—	Операторна	Щит керування (розміщення мікроконтролера, релейного модуля, елементів сигналізації)	—	—	—	1	—

					ДПБ.ПБ-з21.XX.1720.XX.XXX		
Зм.	Арк.	№ докум.	Підп.	Дат.	Специфікація		
Розроб.	Шепелюк К.		2026				
Перевір.	Антонюк В.						
Н. контр.							
Затверд.							
					Літ.	Арк.	Аркушів
					Д	2	3
					КПІ ім. Ігоря Сікорського, Кафедра КІТВП		

ДОДАТОК В.

Лістинг програмного коду

main.py

"""

Точка входу програми. Запускає інтерфейс оператора, який створює всю систему й веде неблокуючий цикл.

Запуск усієї програми: `python main.py`

Побудова графіків тестування - окремо: `python graphs.py`

"""

```
import tkinter as tk
```

```
from ui import GreenhouseUI
```

```
def main():
```

```
    """Створити вікно й запустити інтерфейс."""
```

```
    root = tk.Tk()
```

```
    app = GreenhouseUI(root)
```

```
    root.mainloop()
```

```
if __name__ == "__main__":
```

```
    main()
```

config.py

"""

Конфігурація - єдине місце для всіх параметрів керування.

Уставки, гістерезиси, межі фази доби, фізичні межі датчиків, період опитування, склад датчиків і виконавчих механізмів. Інші модулі лише читають звідси. Відповідає таблиці параметрів регулювання в записці.

"""

```
# період опитування датчиків (такт системи), секунди
```

```
# техвимога: не рідше 1 разу на 30 с
```

```
TS = 30
```

```
# уставки й гістерезиси по контурах
```

```
# target_day / target_night - цільові значення
```

```
# hysteresis - гістерезис  $\Delta$  двопозиційного регулятора
```

```
# op_range - діапазон, у якому оператор може задавати уставку
```

```
SETPPOINTS = {
```

```

    "temperature": {
        "target_day": 22.0,          # °C
        "target_night": 17.0,       # °C, знижена
        "hysteresis": 2.0,          # °C, Δ
        "op_range": (16.0, 28.0),
    },
    "light": {
        "target_day": 1000.0,        # μmol/m²/s
        "target_night": 0.0,         # вночі лампа вимкнена (фотоперіод)
        "hysteresis": 100.0,         # μmol/m²/s
        "op_range": (100.0, 1800.0),
    },
    "humidity": {
        "target_day": 70.0,          # %
        "target_night": 70.0,        # не залежить від фази доби
        "hysteresis": 5.0,           # %
        "op_range": (55.0, 90.0),
    },
}

```

```

# межі фази доби (день/ніч), години; оператор може змінювати
# день = [DAY_START; DAY_END), решта доби - ніч
DAY_START = 6
DAY_END = 22

```

```

# фізичні межі датчиків (блок 6 головного алгоритму)
# значення поза цими межами = несправність датчика -> аварія
# імітатор при відмові видає -999, що гарантовано поза межами
PHYSICAL_LIMITS = {
    "t_in": (-20.0, 60.0),          # °C
    "humidity": (0.0, 100.0),        # %
    "light": (0.0, 2500.0),          # μmol/m²/s
    "t_out": (-20.0, 60.0),          # °C
}

```

```

# склад датчиків: unit - одиниця виміру, contour - до якого контуру належить
# t_out лише для індикації (контур немає, у регулюванні не бере участі)
SENSORS = {
    "t_in": {"unit": "°C",          "contour": "temperature"},
    "humidity": {"unit": "%",        "contour": "humidity"},
    "light": {"unit": "μmol/m²/s",   "contour": "light"},
    "t_out": {"unit": "°C",          "contour": None},
}

```

```

# склад виконавчих механізмів
# contour - до якого контуру належить; type - тип виконавчого органу; action -
напрямок регулювання
ACTUATORS = {

```

```

    "heater":      {"contour": "temperature", "type": "relay", "action": "raise"},
    "window":      {"contour": "temperature", "type": "drive", "action": "lower"},
    "lamp":         {"contour": "light",      "type": "relay", "action": "raise"},
    "humidifier":  {"contour": "humidity",    "type": "relay", "action": "raise"},
}

```

```

def get_target(contour, phase):
    """
    Повернути цільову уставку для контуру залежно від фази доби.
    phase: "day" або "night".
    """
    if phase == "night":
        return SETPOINTS[contour]["target_night"]
    else:
        return SETPOINTS[contour]["target_day"]

```

```

def get_hysteresis(contour):
    """Повернути гістерезис Δ для контуру."""
    return SETPOINTS[contour]["hysteresis"]

```

greenhouse.py

```

"""
Модель теплиці - об'єкт керування.

Описує динаміку температури, освітленості й вологості з інерцією.
Датчики читають її стан, виконавчі механізми на неї впливають.
Сама логіки регулювання не містить.
"""

```

```

class GreenhouseModel:
    def __init__(self, t_out=10.0, humidity_out=50.0):
        # Поточний стан теплиці
        self.t_in = 18.0
        self.light = 0.0          # освітленість (PAR), μmol/m²/s
        self.humidity = 60.0

        # Зовнішні збурення
        self.t_out = t_out
        self.humidity_out = humidity_out

        # Стан виконавчих механізмів: 0 = вимкнено, 1 = увімкнено
        # кватирка: 0 = закрита, 1 = відкрита
        self.heater = 0
        self.window = 0
        self.lamp = 0

```

```

self.humidifier = 0

# Коефіцієнти швидкості зміни стану
self.k_exchange = 0.02      # пасивний теплообмін із зовнішнім
середовищем
self.k_heat = 0.45          # наскільки сильно гріє нагрівач
self.k_vent_t = 0.08        # додаткове охолодження при відкритій
кватирці
self.lamp_level = 1000      # освітленість при увімкненій лампі,
μmol/m²/s
self.k_dry = 0.01           # швидкість природного підсихання
self.k_humid = 0.05         # наскільки піднімає вологість зволожувач
self.k_vent_humid = 0.10    # швидкість вирівнювання вологості через
відкриту кватирку

def set_actuators(self, heater, window, lamp, humidifier):
    """Задати стан виконавчих механізмів (0 - вимкнено, 1 - увімкнено)."""
    self.heater = heater
    self.window = window
    self.lamp = lamp
    self.humidifier = humidifier

def step(self):
    """Оновити стан теплиці після кожного такту (раз на Ts)."""
    self._update_temperature()
    self._update_light()
    self._update_humidity()

# Методи оновлення значень параметрів

def _update_temperature(self):
    # Природне охолодження: температура тягнеться до зовнішньої
    # Різниця t_out - t_in задає напрям: якщо зовні холодніше - t_in падає,
    якщо тепліше - росте.
    change = self.k_exchange * (self.t_out - self.t_in)

    # Нагрівач підвищує температуру.
    if self.heater == 1:
        change += self.k_heat

    # Відкрита кватирка прискорює наближення до зовнішньої температури
    if self.window == 1:
        change += self.k_vent_t * (self.t_out - self.t_in)

    self.t_in += change

def _update_light(self):
    # Є лампа - є світло, нема - темно.
    if self.lamp == 1:
        self.light = self.lamp_level

```

```

    else:
        self.light = 0.0

def _update_humidity(self):
    # Природне підсихання: вологість сама помалу падає.
    change = -self.k_dry * self.humidity

    # Зволожувач піднімає вологість (тягне до 100%).
    if self.humidifier == 1:
        change += self.k_humid * (100 - self.humidity)

    if self.window == 1:
        change += self.k_vent_humid * (self.humidity_out - self.humidity)

    self.humidity += change

    # Вологість не може вийти за межі 0...100 %.
    self.humidity = max(0.0, min(100.0, self.humidity))

def get_state(self):
    """Повернути поточний стан теплиці у вигляді словника."""
    return {
        "t_in": round(self.t_in, 2),
        "light": round(self.light, 2),
        "humidity": round(self.humidity, 2),
        "t_out": self.t_out,
        "humidity_out": self.humidity_out,
    }

```

sensors.py

```
"""
```

Імітатор датчиків.

Читає справжній стан моделі й псує його, імітуючи реальний прилад: додає шум (завжди), іноді видає збій - фізично неможливе значення, яке ядро ловить як несправність (блок 6 головного алгоритму).

```
"""
```

```
import random
```

```

class SensorSimulator:
    def __init__(self, failure_prob=5e-06):
        # розмах шуму кожного датчика
        self.noise = {
            "t_in": 0.3,    # °C
            "humidity": 2.0, # %
            "light": 40.0,   # μmol/m²/s

```

```

        "t_out": 0.3, # °C
    }

    self.failure_prob = failure_prob

    # відмова - значення фізично неможливе, блок 6 впіймає його як вихід за
межі
    self.failure_value = {
        "t_in": -999.0,
        "humidity": -999.0,
        "light": -999.0,
        "t_out": -999.0,
    }

def read(self, model):
    """Зчитати всі датчики."""
    true_state = model.get_state()

    measured = {}
    for name in ("t_in", "humidity", "light", "t_out"):
        measured[name] = self._read_one(name, true_state[name])
    return measured

def _read_one(self, name, true_value):
    """Зчитати один датчик: спершу перевірка на відмову, потім шум."""
    if random.random() < self.failure_prob:
        return self.failure_value[name]

    spread = self.noise[name]
    error = random.uniform(-spread, spread)
    result = true_value + error

    # освітленість і вологість не бувають від'ємними - обмежуємо знизу нулем
    if name in ("light", "humidity"):
        result = max(0.0, result)

    return round(result, 2)

# Перевірка:
if __name__ == "__main__":
    from greenhouse import GreenhouseModel

    gh = GreenhouseModel(t_out=10.0, humidity_out=50.0)
    gh.set_actuators(heater=1, window=0, lamp=1, humidifier=1)

    # підвищена ймовірність відмови, щоб збій було видно за кілька тактів
    sensors = SensorSimulator(failure_prob=0.05)

    print("Такт | T_вн (вимір.) | Освітленість (вимір.) | Вологість (вимір.)")
    print("-" * 65)

```

```

    for tact in range(1, 21):
        gh.step()
        m = sensors.read(gh)
        print(f"{tact:4} | {m['t_in']:13} | {m['light']:21} | {m['humidity']:18}")

```

regulator.py

```

"""

```

Двопозиційні регулятори з гістерезисом.

Чисті функції: за значенням, уставкою й гістерезисом повертають рішення ON/OFF/без змін. Не знають про модель, датчики чи інші контури. regulate_raise - для механізмів, що підвищують параметр; regulate_lower - для тих, що знижують.

```

"""

```

```

def regulate_raise(value, target, hyst):

```

```

    """

```

Регулятор на підвищення (нагрівач, фітолампа, зволожувач).

value - поточне виміряне значення

target - уставка

hyst - гістерезис Δ

Повертає: True (увімкнути), False (вимкнути), None (лишити як було).

```

    """

```

```

    low = target - hyst/2

```

```

    high = target + hyst/2

```

```

    if value < low:

```

```

        return True

```

```

    elif value > high:

```

```

        return False

```

```

    else:

```

```

        return None          # у зоні гістерезису -> не чіпаємо

```

```

def regulate_lower(value, target, hyst):

```

```

    """Регулятор на зниження (кватирка). Дзеркальний до regulate_raise"""

```

```

    low = target - hyst/2

```

```

    high = target + hyst/2

```

```

    if value > high:

```

```

        return True

```

```

    elif value < low:

```

```

        return False

```

```

    else:

```

```

        return None

# Перевірка:
if __name__ == "__main__":
    target = 22.0
    hyst = 2.0

    print(f"Уставка {target}, гістерезис {hyst} (межі {target-hyst} і {target+hyst})")
    print("\nНАГРІВ (нагрівач):")
    for v in (18, 20, 21, 22, 23, 24, 26):
        r = regulate_raise(v, target, hyst)
        decision = {True: "увімкнути", False: "вимкнути", None: "лишити як було"}[r]
        print(f"    значення {v:>3} -> {decision}")

    print("\nОХОЛОДЖЕННЯ (кватирка):")
    for v in (18, 20, 21, 22, 23, 24, 26):
        r = regulate_lower(v, target, hyst)
        decision = {True: "відкрити", False: "закрити", None: "лишити як було"}[r]
        print(f"    значення {v:>3} -> {decision}")

```

controller.py

```

"""

```

Ядро керування - головний цикл системи (головний алгоритм, блоки 1-16).

Зводить усі частини до купи: читає датчики, перевіряє на наявність аварій, визначає

фазу доби, викликає регулятори, координує контури, командує механізмами.

Тільки читає уставки зі сховища; єдине керує виконавчими механізмами.

```

"""

```

```

import config
import regulator
import greenhouse
import sensors
import actuator
import storage
import alarm

class Controller:
    def __init__(self):
        # Створюємо всі частини системи.
        self.model = greenhouse.GreenhouseModel()
        self.sensors = sensors.SensorSimulator()

```

```

self.actuators = actuator.ActuatorSimulator()
self.storage = storage.SetpointStorage()
self.alarm = alarm.AlarmSystem()

# Режим керування: "auto" (автоматичний) або "manual" (ручний).
self.mode = "auto"

# Модельний час у секундах від початку доби (0..86399).
# Стартуємо опівдні: 12 год × 3600 = 43200 с.
# За такт час росте на TS (config), тож доба = 86400/TS тактів.
self.model_time = 12 * 3600

# Лічильник тактів (для журналу).
self.tact = 0

# Чи працює система (False -> зупинка, напр. після аварії).
self.running = True

# Останні виміряні значення (для інтерфейсу).
self.last_measured = {}

def step(self):
    """
    Один такт системи. Відповідає одному проходу головного алгоритму.
    Викликається інтерфейсом раз на Ts.
    """
    if self.running == False:
        return # система зупинена - нічого не відбувається

    self.tact += 1

    # 1. Модель робить крок.
    self.model.step()

    # 2. Читаємо датчики.
    measured = self.sensors.read(self.model)
    self.last_measured = measured

    # 3. Перевірка на аварію (несправність датчика).
    if self.alarm.check_failure(measured, tact=self.tact):
        self._handle_alarm() # безпечний стан + зупинка
        return

    # 4. Ручний режим?
    if self.mode == "manual":
        # У ручному режимі регулятори не працюють - команди задає оператор.
        self.actuators.push_to_model(self.model)
        return

    # 5. Автоматичний режим: визначаємо фазу доби.

```

```

phase = self._determine_phase()

# 6. Регулюємо кожен контур.
self._regulate(measured, phase)

# 7. Координація контурів (блокування зволожувача при відкритій
кватирці).
self._coordinate()

# 8. Передаємо підсумкові команди механізмам у модель.
self.actuators.push_to_model(self.model)

def _current_hour(self):
    """Поточна година доби (0..23), обчислена з модельного часу."""
    return self.model_time // 3600

def _determine_phase(self):
    """Блок 11: день чи ніч за поточною годиною й межами зі сховища."""
    start, end = self.storage.get_day_bounds()
    hour = self._current_hour()
    if start <= hour < end:
        return "day"
    else:
        return "night"

def _regulate(self, measured, phase):
    """
    Блоки 12-14: для кожного механізму викликати потрібний регулятор.
    Напрямок (raise/lower) береться з config (поле action).
    """
    for name, info in config.ACTUATORS.items():
        contour = info["contour"]
        action = info["action"]

        value = self._value_for_contour(measured, contour)

        target = self.storage.get_target(contour, phase)
        hyst = self.storage.get_hysteresis(contour)

        if action == "raise":
            decision = regulator.regulate_raise(value, target, hyst)
        else:
            decision = regulator.regulate_lower(value, target, hyst)

        self.actuators.apply(name, decision)

    # Окрема перевірка: попередження "охолодження неможливе".
    self._check_cooling_warning(measured)

def _value_for_contour(self, measured, contour):

```

```

        """Знайти виміряне значення датчика, що належить цьому контуру."""
        for sensor_name, info in config.SENSORS.items():
            if info["contour"] == contour:
                return measured[sensor_name]
        return None

def _check_cooling_warning(self, measured):
    """
    Попередження: перегрів, який неможливо усунути кватиркою,
    бо зовні не холодніше (T_зовн >= T_вн).
    """
    t_in = measured["t_in"]
    t_out = measured["t_out"]
    target = self.storage.get_target("temperature", self._determine_phase())
    hyst = self.storage.get_hysteresis("temperature")

    too_hot = t_in > target + hyst
    cant_cool = t_out >= t_in
    if too_hot and cant_cool:
        self.actuators.apply("window", False) # охолодження неможливе -
        закрити кватирку
        self.alarm.raise_warning(
            "temperature",
            "охолодження неможливе (T_зовн ≥ T_вн)",
            tact=self.tact,
        )

def _coordinate(self):
    """Координація контурів: при відкритті кватирці зволожувач блокується"""
    state = self.actuators.get_state()
    if state["window"] == 1:
        self.actuators.apply("humidifier", False) # примусово вимкнути

def _handle_alarm(self):
    """Блоки 7-8: аварія -> безпечний стан + зупинка."""
    self.actuators.set_safe_state()
    self.actuators.push_to_model(self.model)
    self.running = False # зупиняємо систему (блок 16: кінець)

def get_status(self):
    """Зібрати поточний стан для відображення в інтерфейсі."""
    return {
        "tact": self.tact,
        "measured": self.last_measured,
        "actuators": self.actuators.get_state(),
        "alarm": self.alarm.get_status(),
        "mode": self.mode,
        "hour": self._current_hour(),
        "running": self.running,
    }

```

actuator.py

"""

Імітатор виконавчих механізмів - посередник між ядром і моделлю.

Зберігає поточний стан кожного механізму, застосовує рішення регулятора (ON/OFF/без змін) і передає підсумкові стани моделі. Має окремий безпечний стан для аварійної зупинки.

"""

import config

class ActuatorSimulator:

def __init__(self):

 # стан кожного механізму (0/1); перелік береться з config,
 # тож новий VM у config автоматично з'явиться й тут

 self.state = {}

for name **in** config.ACTUATORS:

 self.state[name] = 0

def apply(self, name, decision):

 """

 Застосувати рішення регулятора до одного механізму.

 name - назва механізму ("heater", "window", ...)

 decision - True / False / None

 """

if decision **is** True:

 self.state[name] = 1

elif decision **is** False:

 self.state[name] = 0

 # None - нічого не робимо, стан лишається як був.

def set_safe_state(self):

 """

 Безпечний стан при аварії (відмова датчика):

 нагрівач OFF, зволожувач OFF, квартира ВІДКРИТА.

 Викликається ядром в аварійній гілці, оминаючи регулятори.

 """

 self.state["heater"] = 0

 self.state["humidifier"] = 0

 self.state["window"] = 1 # квартира ВІДКРИТА - пасивно безпечно при

 аварії

 self.state["lamp"] = 0

def push_to_model(self, model):

 """Передати поточні стани механізмів у модель теплиці."""

 model.set_actors(

```

        heater=self.state["heater"],
        window=self.state["window"],
        lamp=self.state["lamp"],
        humidifier=self.state["humidifier"],
    )

    def get_state(self):
        """Повернути поточні стани механізмів."""
        return dict(self.state)

```

storage.py

```
"""
```

Сховище уставок.

Зберігає поточні уставки контурів і межі фази доби.

Ядро лише читає (get_*), інтерфейс лише пише (set_*) - так завжди

зрозуміло, що уставку міг змінити тільки оператор. Запис із валідацією діапазону.

```
"""
```

```
import config
```

```
class SetpointStorage:
```

```

    def __init__(self):
        # копіюємо стартові уставки з config у власне сховище:
        # config лишається незмінним (стартові значення),
        # оператор далі міняє цю копію (поточні значення)
        self.setpoints = {}
        for contour, params in config.SETPOINTS.items():
            self.setpoints[contour] = {
                "target_day": params["target_day"],
                "target_night": params["target_night"],
                "hysteresis": params["hysteresis"],
            }

        # межі фази доби (поточні; стартові з config).
        self.day_start = config.DAY_START
        self.day_end = config.DAY_END

```

```
# --- ЧИТАННЯ (для ядра) ---
```

```

def get_target(self, contour, phase):
    """
    Цільова уставка контуру для заданої фази доби.
    phase: "day" або "night".
    """
    if phase == "night":
        return self.setpoints[contour]["target_night"]

```

```

        else:
            return self.setpoints[contour]["target_day"]

def get_hysteresis(self, contour):
    """Гістерезис Δ контуру."""
    return self.setpoints[contour]["hysteresis"]

def get_day_bounds(self):
    """Межі фази доби: (початок дня, кінець дня) у годинах."""
    return (self.day_start, self.day_end)

def set_target(self, contour, phase, value):
    """
    Задати уставку контуру для фази доби.
    Перевіряє, що значення в допустимому діапазоні (config op_range).
    Повертає True, якщо прийнято; False, якщо поза діапазоном.
    """
    if value < config.SETPOINTS[contour]["op_range"][0] or value >
config.SETPOINTS[contour]["op_range"][1]:
        return False # поза діапазоном - не приймаємо

    if phase == "night":
        self.setpoints[contour]["target_night"] = value
    else:
        self.setpoints[contour]["target_day"] = value
    return True

def set_day_bounds(self, start, end):
    """
    Задати межі фази доби (години 0..23).
    Повертає True, якщо прийнято; False, якщо некоректні.
    """
    if not (0 <= start <= 23) or not (0 <= end <= 23):
        return False
    if start == end:
        return False # рівні межі - день нульової довжини, не приймаємо
    self.day_start = start
    self.day_end = end
    return True

```

alarm.py

"""

Сигналізація - класифікація і реєстрація ненормальних ситуацій.

Аварію (несправність датчика, вихід за фізичні межі) виявляє сама.
 Попередження (проблема процесу при справному датчику) реєструє за
 повідомленням ядра. Пасивна: рішень про керування не приймає, веде журнал.
 """

```

import config

class AlarmSystem:
    def __init__(self):
        # рівень тривоги: None (норма), "warning" (попередження), "alarm"
        (аварія)
        self.level = None
        self.message = ""

        # журнал подій (такт, рівень, повідомлення)
        self.log = []

    def check_failure(self, measured, tact=None):
        """
        Перевірити виміряні значення на несправність датчика (аварія).
        measured - словник виміряних значень від датчиків.
        Повертає True, якщо виявлено аварію (вихід за фізичні межі).

        Цю перевірку сигналізація робить сама - вона універсальна.
        """
        for name, value in measured.items():
            low, high = config.PHYSICAL_LIMITS[name]
            if value < low or value > high:
                msg = f"Несправність датчика '{name}': значення {value} поза
                межами [{low}; {high}]"
                self._set(level="alarm", message=msg, tact=tact)
                return True
        return False

    def raise_warning(self, contour, message, tact=None):
        """
        Зареєструвати попередження, про яке повідомило ядро.
        contour - який контур ("temperature", ...).
        message - опис ситуації (формує ядро, напр. "охолодження неможливе").
        """
        full_msg = f"Попередження [{contour}]: {message}"
        if self.level != "alarm": # аварія важливіша - попередження не знижує
        рівень
            self._set(level="warning", message=full_msg, tact=tact)

    def clear(self):
        """Скинути поточну тривогу (ситуація нормалізувалась)."""
        self.level = None
        self.message = ""

    def get_status(self):
        """Поточний стан тривоги: (рівень, повідомлення)."""
        return (self.level, self.message)

```

```

def _set(self, level, message, tact):
    self.level = level
    self.message = message
    self.log.append({
        "tact": tact,
        "level": level,
        "message": message,
    })

# Перевірка:
if __name__ == "__main__":
    alarm = AlarmSystem()

    # нормальні дані - аварії немає
    normal = {"t_in": 22.0, "humidity": 70.0, "light": 1000.0, "t_out": 10.0}
    is_alarm = alarm.check_failure(normal, tact=1)
    print(f"Нормальні дані -> аварія: {is_alarm}, стан: {alarm.get_status()}")

    # попередження від ядра
    alarm.raise_warning("temperature", "охолодження неможливе ( $T_{\text{зовн}} \geq T_{\text{вн}}$ )",
    tact=5)
    print(f"Після попередження -> стан: {alarm.get_status()}")

    # відмова датчика (-999) - аварія
    failed = {"t_in": -999.0, "humidity": 70.0, "light": 1000.0, "t_out": 10.0}
    is_alarm = alarm.check_failure(failed, tact=8)
    print(f"Відмова датчика -> аварія: {is_alarm}, стан: {alarm.get_status()}")

    # журнал подій
    print("\nЖурнал подій:")
    for entry in alarm.log:
        print(f" такт {entry['tact']}: [{entry['level']}] {entry['message']}")

```

ui.py

"""

Інтерфейс оператора (Tkinter).

Показує стан системи (індикатори, тривога) і дозволяє керувати:
 задавати уставки, перемикає режим, керувати механізмами вручну.
 Бере дані в ядра й шле йому команди; ядро про інтерфейс не знає.
 Оновлення - неблокуючим циклом через root.after.
 """

```

import tkinter as tk
import controller

```

```

import config

# демонстраційний такт: 1 с замість реальних Ts=30 с (config.TS), щоб процес було
видно
TICK_MS = 1000

class GreenhouseUI:
    def __init__(self, root):
        self.root = root
        self.root.title("Керування мікрокліматом теплиці")

        # ядро - уся система
        self.ctrl = controller.Controller()

        self.auto_running = True # чи йдуть такти самі (автопрогін)

        self._build_indicators()
        self._build_setpoints()
        self._build_controls()

        self._tick() # запуск неблокуючого циклу оновлення

    def _build_indicators(self):
        """Числові індикатори: виміри, механізми, тривога."""
        frame = tk.LabelFrame(self.root, text="Стан системи", padx=10, pady=10)
        frame.pack(fill="x", padx=10, pady=5)

        self.lbl_tact = tk.Label(frame, text="Такт: -")
        self.lbl_tact.grid(row=0, column=0, sticky="w")

        self.lbl_t_in = tk.Label(frame, text="T_вн: -")
        self.lbl_t_in.grid(row=1, column=0, sticky="w")

        self.lbl_humidity = tk.Label(frame, text="Вологість: -")
        self.lbl_humidity.grid(row=2, column=0, sticky="w")

        self.lbl_light = tk.Label(frame, text="Освітленість: -")
        self.lbl_light.grid(row=3, column=0, sticky="w")

        self.lbl_t_out = tk.Label(frame, text="T_зовн: -")
        self.lbl_t_out.grid(row=4, column=0, sticky="w")

        self.lbl_actuators = tk.Label(frame, text="Механізми: -")
        self.lbl_actuators.grid(row=5, column=0, sticky="w", pady=(8, 0))

        # тривога - колір змінюється залежно від рівня
        self.lbl_alarm = tk.Label(frame, text="Стан: норма", fg="green")
        self.lbl_alarm.grid(row=6, column=0, sticky="w", pady=(8, 0))

```

```

def _build_setpoints(self):
    """Поля задання уставок і меж фази доби. День/ніч лише для температури;
    вологість однакова вдень і вночі; нічна освітленість фіксована (0)."""
    frame = tk.LabelFrame(self.root, text="Уставки", padx=10, pady=10)
    frame.pack(fill="x", padx=10, pady=5)

    tk.Label(frame, text="Контип").grid(row=0, column=0, sticky="w")
    tk.Label(frame, text="День").grid(row=0, column=1)
    tk.Label(frame, text="Ніч").grid(row=0, column=2)

    self.entries = {}
    row = 1
    for contour in ("temperature", "light", "humidity"):
        tk.Label(frame, text=contour).grid(row=row, column=0, sticky="w")

        entry_day = tk.Entry(frame, width=8)
        entry_day.insert(0, str(self.ctrl.storage.get_target(contour,
"day"))))
        entry_day.grid(row=row, column=1, padx=5)
        self.entries[contour] = {"day": entry_day}

        # нічне поле - лише для температури
        if contour == "temperature":
            entry_night = tk.Entry(frame, width=8)
            entry_night.insert(0, str(self.ctrl.storage.get_target(contour,
"night"))))
            entry_night.grid(row=row, column=2, padx=5)
            self.entries[contour]["night"] = entry_night

        row += 1

    # межі фази доби (фотоперіод): початок і кінець дня в годинах
    tk.Label(frame, text="День з / до (год)").grid(row=row, column=0,
sticky="w")

    self.entry_day_start = tk.Entry(frame, width=8)
    self.entry_day_start.insert(0,
str(self.ctrl.storage.get_day_bounds()[0]))
    self.entry_day_start.grid(row=row, column=1, padx=5)

    self.entry_day_end = tk.Entry(frame, width=8)
    self.entry_day_end.insert(0, str(self.ctrl.storage.get_day_bounds()[1]))
    self.entry_day_end.grid(row=row, column=2, padx=5)

    row += 1

    tk.Button(frame, text="Застосувати установки",
               command=self._apply_setpoints).grid(row=row, column=0,
columnspan=3, pady=5)

    self.lbl_setpoint_msg = tk.Label(frame, text="", fg="blue")

```

```

self.lbl_setpoint_msg.grid(row=row+1, column=0, columnspan=3)

def _build_controls(self):
    """Кнопки режиму й ручного керування."""
    frame = tk.LabelFrame(self.root, text="Керування", padx=10, pady=10)
    frame.pack(fill="x", padx=10, pady=5)

    self.lbl_mode = tk.Label(frame, text="Режим: auto")
    self.lbl_mode.grid(row=0, column=0, sticky="w")
    tk.Button(frame, text="Авто/Ручний",
               command=self._toggle_mode).grid(row=0, column=1, padx=5)

    # кнопки ручного керування механізмами (діють у ручному режимі)
    col = 0
    self.manual_buttons = {}
    for name in config.ACTUATORS:
        btn = tk.Button(frame, text=name,
                        command=lambda n=name: self._toggle_actuator(n))
        btn.grid(row=1, column=col, padx=3, pady=5)
        self.manual_buttons[name] = btn
        col += 1

    tk.Button(frame, text="Пауза/Пуск",
               command=self._toggle_pause).grid(row=2, column=0, pady=5)

def _apply_setpoints(self):
    """Застосувати введені уставки й межі фази доби."""
    messages = []
    for contour, fields in self.entries.items():
        for phase, entry in fields.items(): # тільки наявні поля
            text = entry.get()
            try:
                value = float(text)
            except ValueError:
                messages.append(f"{contour}/{phase}: не число")
                continue
            ok = self.ctrl.storage.set_target(contour, phase, value)
            if not ok:
                messages.append(f"{contour}/{phase}: поза діапазоном")

    # вологість: нічна уставка завжди дорівнює денній
    if contour == "humidity" and "day" in fields:
        day_val = self.ctrl.storage.get_target("humidity", "day")
        self.ctrl.storage.set_target("humidity", "night", day_val)

    # межі фази доби - цілі години
    try:
        start = int(self.entry_day_start.get())
        end = int(self.entry_day_end.get())
        if not self.ctrl.storage.set_day_bounds(start, end):

```

```

        messages.append("межі дня: некоректні (0-23, початок ≠ кінець)")
except ValueError:
    messages.append("межі дня: не цілі числа")

if messages:
    self.lbl_setpoint_msg.config(text="; ".join(messages), fg="red")
else:
    self.lbl_setpoint_msg.config(text="Уставки прийнято", fg="green")

def _toggle_mode(self):
    """Перемкнути режим керування auto <-> manual."""
    if self.ctrl.mode == "auto":
        self.ctrl.mode = "manual"
    else:
        self.ctrl.mode = "auto"
    self.lbl_mode.config(text=f"Режим: {self.ctrl.mode}")

def _toggle_actuator(self, name):
    """Ручне перемикання механізму (діє в ручному режимі)."""
    current = self.ctrl.actuators.get_state()[name]
    if current == 1:
        self.ctrl.actuators.apply(name, False)
    else:
        self.ctrl.actuators.apply(name, True)

def _toggle_pause(self):
    """Пауза або продовження автопрогону тактів."""
    self.auto_running = not self.auto_running

def _tick(self):
    """
    Один такт: крок системи + оновлення екрана.
    Не блокує вікно - Tkinter сам викличе цей метод знову через TICK_MS.
    """
    if self.auto_running:
        # прокручуємо модельний час на TS секунд за такт;
        self.ctrl.model_time = (self.ctrl.model_time + config.TS) % 86400
        self.ctrl.step()

    self._update_display()
    self.root.after(TICK_MS, self._tick)

def _update_display(self):
    """Оновити всі індикатори поточним станом ядра."""
    s = self.ctrl.get_status()
    m = s["measured"]

    self.lbl_tact.config(text=f"Такт: {s['tact']} (година {s['hour']}:00)")

    if m:

```

```

self.lbl_t_in.config(text=f"T_вн: {m['t_in']} °C")
self.lbl_humidity.config(text=f"Вологість: {m['humidity']} %")
self.lbl_light.config(text=f"Освітленість: {m['light']} μmol/m²/s")
self.lbl_t_out.config(text=f"T_зовн: {m['t_out']} °C")

a = s["actuators"]
self.lbl_actuators.config(
    text=f"Нагрівач: {a['heater']} Кватирка: {a['window']} "
    f"Лампа: {a['lamp']} Зволожувач: {a['humidifier']}"
)

level, message = s["alarm"]
if level == "alarm":
    self.lbl_alarm.config(text=f"АВАРІЯ: {message}", fg="red")
elif level == "warning":
    self.lbl_alarm.config(text=f"Попередження: {message}", fg="orange")
else:
    self.lbl_alarm.config(text="Стан: норма", fg="green")

```

graphs.py

```
"""
```

Модуль тестування - побудова графіків перехідних процесів для аркуша «Результати тестування».

Окремий офлайн-інструмент: проганняє систему задану кількість тактів, збирає історію й малює три графіки (температура, освітленість, вологість) з лініями уставок і коридорами гістерезису.

```
"""
```

```

import matplotlib
matplotlib.use("Agg") # режим без вікна - зберігаємо у файли
import matplotlib.pyplot as plt
import os

import config
import controller

def run_simulation(n_ticks=300, start_t_in=15.0, time_step_seconds=None):
    """
    Прогнати систему n_ticks тактів і зібрати історію.
    Повертає словник з історією по тактах.
    """
    if time_step_seconds is None:
        time_step_seconds = config.TS

    ctrl = controller.Controller()
    ctrl.model.t_in = start_t_in

```

```

history = {
    "tact": [],
    "t_in": [], "humidity": [], "light": [],
    "target_temp": [], "target_humid": [], "target_light": [],
}

for _ in range(n_ticks):
    ctrl.model_time = (ctrl.model_time + time_step_seconds) % 86400
    ctrl.step()

    if ctrl.running == False:
        break # система зупинена (аварія)

    s = ctrl.get_status()
    m = s["measured"]
    phase = ctrl._determine_phase()

    history["tact"].append(s["tact"])
    history["t_in"].append(m["t_in"])
    history["humidity"].append(m["humidity"])
    history["light"].append(m["light"])
    # уставки поточної фази - для лінії цілі на графіку
    history["target_temp"].append(ctrl.storage.get_target("temperature",
phase))
    history["target_humid"].append(ctrl.storage.get_target("humidity",
phase))
    history["target_light"].append(ctrl.storage.get_target("light", phase))

return history

def _plot_single(ax, tacts, values, targets, hyst, title, ylabel, color):
    """Намалювати один графік: параметр, уставка, коридор гістерезису."""
    ax.plot(tacts, values, color=color, label="параметр")
    ax.plot(tacts, targets, color="black", linestyle="--", label="уставка")

    # коридор гістерезису: смуга уставка ± Δ
    upper = [t + hyst/2 for t in targets]
    lower = [t - hyst/2 for t in targets]
    ax.fill_between(tacts, lower, upper, color=color, alpha=0.15,
label="коридор гістерезису")

    ax.set_title(title)
    ax.set_xlabel("Такт")
    ax.set_ylabel(ylabel)
    ax.legend(fontsize=8)
    ax.grid(True, alpha=0.3)

def build_graphs(history, filename="test_results.png"):

```

```

"""Побудувати три графіки (температура, освітленість, вологість) в один
файл."""
fig, axes = plt.subplots(3, 1, figsize=(10, 12))

_plot_single(
    axes[0], history["tact"], history["t_in"], history["target_temp"],
    config.get_hysteresis("temperature"),
    "Температура", "T, °C", "tab:red",
)

_plot_single(
    axes[1], history["tact"], history["light"], history["target_light"],
    config.get_hysteresis("light"),
    "Освітленість", "E, μmol/m²/s", "tab:orange",
)

_plot_single(
    axes[2], history["tact"], history["humidity"], history["target_humid"],
    config.get_hysteresis("humidity"),
    "Відносна вологість", "φ, %", "tab:blue",
)

fig.tight_layout()
fig.savefig(filename, dpi=100)
plt.close(fig)
return filename

if __name__ == "__main__":
    os.makedirs("data", exist_ok=True) # папка має існувати до збереження

    print("Проганяю модель...")
    history = run_simulation(n_ticks=300, start_t_in=15.0)
    print(f"Зібрано {len(history['tact'])} тактів.")

    out = build_graphs(history, "data/test_results.png")
    print(f"Графіки збережено: {out}")

```