

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-технічний засіб для моніторингу та автоматизації догляду за рослинами з елементами робототехніки
Назва теми

КвРКІ 210123.21.01.77 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Виконав: студент IV курсу, група K12-21-1



Підпис

Володимир Рудов

Ініціали, прізвище

Керівник



Підпис, дата

Олег ВОЙЧУР

Ініціали, прізвище

Нормоконтролер



Підпис, дата

Тетяна КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної
інженерії та інформаційних
систем



Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

«05» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Володимиру РУДОВУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб для моніторингу та автоматизації догляду за рослинами з елементами робототехніки

Керівник проекту (роботи) Олег ВОЙЧУР

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Огляд існуючих рішень для вирішення завдання

Огляд існуючих рішень для вирішення завдання

Програмно-технічний засіб для моніторингу та автоматизації догляду за рослинами з елементами робототехніки

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Архітектура програмно-технічного засобу для моніторингу та автоматизації догляду за рослинами з елементами робототехніки

системи
Схема роботи програмної частини програмно-технічного засобу для моніторингу та автоматизації догляду за рослинами з елементами робототехніки

Апаратна частина програмно-технічного засобу для моніторингу та автоматизації догляду за рослинами з елементами робототехніки

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КНС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КНС		

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – Огляд існуючих рішень для вирішення завдання	01.03.2025	виконано
4	Робота над розділом 2 – Огляд існуючих рішень для вирішення завдання	01.04.2025	виконано
5	Робота над розділом 3 – Програмно-технічний засіб для моніторингу та автоматизації догляду за рослинами з елементами робототехніки	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	15.05.2025	виконано
7	Попередній захист ВКР	22.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Підпис

Володимир РУДОВ
Ініціали, прізвище

Керівник роботи

Підпис

Олег ВОЙЧУР
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб для моніторингу та автоматизації догляду за рослинами з елементами робототехніки».

Автор роботи: Володимир РУДОВ.

Керівник роботи: Олег ВОЙЧУР.

Пояснювальна записка: 65 с., 17 рис., 2 табл., 3 дод., 46 джерел.

Графічна частина: 3 креслення.

ESP32, датчики вологості, BH1750, BME280, HC-SR04, L298N, насос, мобільний застосунок, REST-API.

Метою дипломної роботи розробка та обґрунтування програмно-технічного комплексу для автоматичного моніторингу стану ґрунту й середовища та керування поливом і переміщенням пристрою.

Об'єктом дослідження є функціонування системи збору та обробки даних про стан ґрунту й навколишнього середовища в реальному часі для забезпечення своєчасного поливу й оптимального освітлення рослин.

Предметом дослідження є алгоритми ухвалення рішення щодо активації насоса та руху пристрою, а також взаємодія з мобільним застосунком

Під час проведення даного дослідження були використані такі методи: аналіз технічних джерел, експериментальні випробування датчиків і вузлів, розробка блок-схем і алгоритмів, програмування ESP32 у Arduino IDE, тестування мобільного інтерфейсу.



Підпис студента

30.05.2025

Дата

ЗМІСТ

ВСТУП	4
1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ДЛЯ ВИРІШЕННЯ ЗАВДАННЯ	5
1.1 Програмно-технічні засоби для автоматизації догляду за рослинами	5
1.2 Порівняння характеристик та функціональних можливостей пристроїв	9
1.3 Мобільні застосунки для моніторингу та догляду за рослинами.....	12
1.4 Висновки	15
2 ОБҐРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ТА СЕРЕДОВИЩА РЕАЛІЗАЦІЇ	16
2.1 Апаратне середовище реалізації.....	16
2.2 Функційні вимоги.....	23
2.3 Нефункційні вимоги.....	27
2.4 Вибір мови програмування, бібліотек та середовища реалізації	31
2.5 Висновки	36
3 ПРОГРАМНО-ТЕХНІЧНИЙ ЗАСІБ ДЛЯ МОНІТОРИНГУ ТА АВТОМАТИЗАЦІЇ ДОГЛЯДУ ЗА РОСЛИНАМИ З ЕЛЕМЕНТАМИ РОБОТОТЕХНІКИ ЗАСІБ ДЛЯ МОНІТОРИНГУ ТА АВТОМАТИЗАЦІЇ ДОГЛЯДУ ЗА РОСЛИНАМИ	37
3.1 Загальна архітектура програмно-технічного засобу для догляду за рослинами	37
3.2 Реалізація програмного забезпечення пристрою	43
3.3 Логіка роботи пристрою в складних умовах.....	50
3.4. Реалізація програмно-технічного засобу для моніторингу та автоматизації догляду за рослинами	55
3.5. Висновки	62
ВИСНОВКИ	63
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	64

КвРКІ 210123.21.01.77 ПЗ				
Зм.	Арк.	№докум.	Підпис	Дата
Виконав.		Володимир РУДОН		05.06.2023
Перевір.		Олег ВОЙЧУР		16.06.23
Н.контр.		Тетяна КИСІЛЬ		05.06.2023
Затвер.		Ольга ПАВЛОВА		05.06.2023
Програмно-технічний засіб для моніторингу та автоматизації догляду за рослинами, з елементами робототехніки				
Літера	Аркуш	Аркушів		
у	2	72		
ХНУ КІ2-21-1				

ДОДАТОК А.....	699
ДОДАТОК Б.....	700
ДОДАТОК В.....	711

					КВРКІ 210123.21.01.77 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

У сучасному світі, де автоматизація охоплює все більше аспектів побуту, виникає природне бажання передати рутинні, але важливі завдання технічним системам. Однією з таких щоденних справ є догляд за кімнатними рослинами. Здавалося б, полити квітку або перенести її ближче до вікна, дрібниця. Але в реальності ці завдання часто залишаються поза увагою, особливо у людей з насиченим графіком. Це призводить до висихання ґрунту, нестачі світла й поступового в'янення рослин, навіть якщо вони мають декоративну чи емоційну цінність для власника.

Тому тема створення розумного пристрою для догляду за рослинами, який міг би автоматично стежити за умовами середовища, самостійно зволожувати ґрунт і навіть змінювати своє розташування для пошуку кращого освітлення, є не лише цікавою, а й актуальною.

Метою дипломної роботи є розробка та реалізація програмно-технічного засобу для моніторингу та автоматизації догляду за рослинами, що забезпечує збір даних про середовище, ухвалення рішень на основі сенсорної інформації та автономне виконання дій .

Об'єктом дослідження є процес автоматизованого моніторингу та догляду за кімнатними рослинами.

Предметом дослідження, програмно-апаратна реалізація системи з сенсорними, виконавчими та керуючими модулями, що функціонують у реальному середовищі.

Методи дослідження, застосовані в роботі, включають: порівняльний аналіз існуючих систем, розробку алгоритмів керування на основі логіки станів, моделювання взаємодії між компонентами, створення прототипу, а також емпіричне тестування у побутових умовах.

					КВРКІ 210123.21.01.77 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ДЛЯ ВИРІШЕННЯ ЗАВДАННЯ

1.1 Програмно-технічні засоби для автоматизації догляду за рослинами

Ідея довірити догляд за рослинами не людині, а розумній машині вже давно перестала бути фантастикою. У наш час цілком реально виростити базилік, петрушку чи навіть мініатюрний лимонник, не маючи ані ґрунту під нігтями, ані мінімального уявлення про цикл росту рослини. Саме на це й орієнтуються сучасні комерційні рішення у сфері автоматизації догляду за домашніми рослинами, розумні вазони, трекери стану ґрунту, автономні поливальні системи.

Одним із найяскравіших представників такого підходу є Ivy Smart Planter. Цей горщик одразу впадає в око завдяки стильному зовнішньому вигляду, мінімалістичний білий корпус, світлодіодна смуга, яка показує стан системи. Але під капотом, справжня електроніка: датчик вологості, температури, освітлення, а також вбудований резервуар із насосом. Ivy не тільки вміє поливати рослину, але й моніторить навколишні умови в режимі реального часу. Взаємодія з пристроєм відбувається через мобільний додаток, де користувач може перевірити стан рослини, отримати рекомендації або змінити частоту поливу. Проте, Ivy швидше створений для естетів, він не дає великої гнучкості в налаштуваннях, і його автономність обмежена невеликим баком на кілька днів.



Рисунок 1.1 – Ivy Smart Planter [1]

					КВРКІ 210123.21.01.77 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

Іншим прикладом, уже ближчим до професійного сегмента, є LetPot MP1. Цей пристрій позиціонується як повноцінна система автоматичного догляду з розширеним функціоналом. Його ключова особливість, велика автономність. За заявами виробника, пристрій може працювати до 30 днів без дозаправки води. Крім того, він має потужні сенсори, які аналізують стан ґрунту не лише на поверхні, а й на рівні коренів. LetPot також підключається до домашньої Wi-Fi мережі, що дозволяє користувачу керувати ним віддалено, наприклад, перевіряти стан рослини, поки він у відпустці, або вмикати полив вручну, якщо автоматичний режим не справляється.



Рисунок 1.2 – LetPot MP1 [2]

На противагу цим комерційним «автоматам» існують продукти, які поєднують контроль і навчання. Один із таких, LeafyPod. Цей пристрій, свого роду «розумний садівник-консультант». Його ідея полягає не стільки в автоматизації, скільки в персоналізації догляду. LeafyPod містить набір базових сенсорів (вологість, температура, освітлення), але головна його цінність, у програмному забезпеченні. Додаток на основі введеної інформації про тип рослини генерує індивідуальні рекомендації щодо поливу, освітлення, добрив. Наприклад, якщо користувач вирощує орхідею, LeafyPod повідомить, що їй бажано трохи нижча вологість, тінь і певна температура. Він не поливає автоматично, але повідомляє,

					КВРКІ 210123.21.01.77 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

коли і що потрібно зробити. Це рішення особливо корисне для новачків, які не хочуть зіпсувати дорого рослинну через незнання.



Рисунок 1.3 – LeafyPod [3]

Не можна не згадати й один із найпопулярніших у світі варіантів, Click and Grow. В основі системи, капсули, які містять уже підготовлений ґрунт із насінням, а сама система забезпечує вологу, освітлення і, частково, вентиляцію. Користувач фактично нічого не робить: вставляє капсулу, додає воду, і пристрій сам доглядає за рослиною. Це неймовірно зручно для офісів або людей, які хочуть вирощувати зелень на підвіконні, але не мають ані часу, ані досвіду.



Рисунок 1.4 – Click and Grow [4]

					КВРКІ 210123.21.01.77 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

На завершення, менш відомий, але цікавий приклад: Parrot Flower Power. Це сенсор, який вставляється в ґрунт біля рослини й передає дані через Bluetooth на смартфон. Він не поливає й не втручається, лише стежить. Але його фішка, у величезній базі знань, яка містить інформацію про тисячі видів рослин. Сенсор аналізує показники і каже: «цей фікус занадто сухий, а твій лавр перегрівається». Проста ідея, реалізована з фокусом на аналітику.



Рисунок 1.5 – Parrot Flower Power [5]

Ще одна цікава категорія, яку не можна оминати, це саморобні (DIY) рішення. У багатьох технічних форумах, на Reddit і GitHub можна знайти десятки проєктів, де ентузіасти самі створюють системи автоматичного поливу або моніторингу стану рослин. Найпопулярніша зв'язка, це мікроконтролер ESP32, датчик вологості ґрунту (типу YL-69 або аналогічний), сервомотор або міні-насос, і простенький Telegram-бот або web-інтерфейс.

Користувач підключає систему до Wi-Fi, і якщо ґрунт пересихає, надходить повідомлення: «Твій кактус хоче пити». Або ж насос вмикається сам. Перевага в тому, що такі рішення можна налаштувати на будь-яку кількість рослин, адаптувати під конкретні умови (тип ґрунту, глибину кореня) і не бути прив'язаним

до фірмового додатку. Мінус, усе потрібно зібрати, налаштувати й написати код. Не кожен на це готовий, але для інженера-початківця, це чудова практика.

Усі ці приклади, від дизайнерських розумних вазонів до відкритих саморобних рішень, ілюструють одну важливу річ: підхід до автоматизації догляду за рослинами буває дуже різним. Одні системи орієнтуються на максимальну автономність: зробити так, щоб людина взагалі не втручалася. Інші, на підтримку й допомогу: навчити, підказати, але не діяти самостійно. Треті, поєднують ці два підходи, даючи вибір користувачу.

З практичної точки зору це означає, що немає ідеального рішення для всіх. Якщо ти забуваєш про рослини і хочеш, щоб усе було «як на автопілоті», обирай LetPot. Якщо любиш самостійно розбиратись у догляді й отримувати підказки, LeafyPod або Parrot Flower Power. А якщо хочеш зібрати свою систему й повністю контролювати її, дорога веде в DIY.

1.2 Порівняння характеристик та функціональних можливостей пристроїв

Порівнюючи сучасні програмно-технічні засоби для автоматизованого догляду за рослинами, важливо не лише перелічити їх функції, а й оцінити практичну користь для різних категорій користувачів. Одні пристрої створені для повної автономії, інші, для інтерактивного керування. Хтось хоче просто забути про рослину на тиждень, а хтось прагне отримувати аналітику в реальному часі. Саме такі відмінності обумовлюють потребу в якісному порівнянні характеристик.

Почати варто з набору сенсорів. Пристрій LetPot MP1 має повноцінну сенсорну систему: вологість ґрунту, температура повітря, рівень освітлення, наявність води в резервуарі. Це дозволяє точно визначати, коли рослині щось потрібно. Аналогічні функції присутні і в системі Click and Grow, однак там усе заховано за готовими алгоритмами, які не дають змоги змінювати параметри. LeafyPod, навпаки, має менше сенсорів, але компенсує це програмним аналізом, орієнтуючись на вручну введені дані.

					КВРКІ 210123.21.01.77 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.1 – Порівняльна схема сенсорних модулів LetPot MP1, Click and Grow, LeafyPod

№	Тип сенсора	LetPot MP1	Click and Grow	LeafyPod
1	Світловий сенсор	+	+	+
2	Сенсор рівня води	+	+	+
3	Температурний сенсор	-	+	+
4	Сенсор вологості повітря	-	+	+
5	Сенсор вологості ґрунту	-	-	+
6	pH-сенсор	-	-	+

Щодо автономності, лідером лишається LetPot, його резервуар, керування живленням та вологістю дозволяють залишити рослину до двох тижнів без догляду. Click and Grow також автономний, однак залежить від капсульної системи живлення і субстрату. LeafyPod не має поливального модуля, він лише сповіщає про необхідність поливу, тому все залежить від того, наскільки оперативно реагуватиме користувач.

Варто також розглянути гнучкість керування. У LetPot користувач може задавати параметри поливу, пороги чутливості, дивитись лог змін. Це робить систему цікавою не лише для побутового користування, а й для дослідницьких проєктів. У Click and Grow нічого налаштувати не можна, він працює за фіксованим шаблоном. LeafyPod, навпаки, робить ставку на діалог: система питає користувача

про тип рослини, умови середовища, після чого пропонує оптимальний режим догляду.

Ще одним цікавим аспектом є спосіб підключення. LetPot і Click and Grow мають модулі Wi-Fi, що дозволяє віддалено керувати пристроєм, отримувати сповіщення і змінювати параметри. LeafyPod працює переважно через Bluetooth і вимагає постійного з'єднання зі смартфоном для передачі даних. Це знижує зручність у тривалій перспективі, особливо якщо користувач планує залишити дім.

Інтерфейс, це окрема історія. LetPot пропонує зручний мобільний застосунок із візуалізацією даних і можливістю керування. Click and Grow взагалі не має цифрового інтерфейсу, уся взаємодія відбувається через фізичні дії: залив воду, вставив капсулу, чекаєш. LeafyPod, у свою чергу, використовує гібридну модель, дані вводяться вручну, а рекомендації формуються автоматично.

І зрештою, зовнішній вигляд. Для багатьох користувачів важливо, щоб пристрій не виглядав як лабораторна установка. У цьому плані Click and Grow виглядає найбільш дизайнерськи. LetPot, функціональний, але не надто вишуканий. LeafyPod, компактний, майже непомітний. Це може бути важливо, якщо пристрій стоїть на кухонному підвіконні або в офісі.

Таким чином, можна зробити висновок, що кожен із проаналізованих пристроїв має свою цільову аудиторію. LetPot, для тих, хто хоче максимальної автономності та керування. Click and Grow, для користувачів, які хочуть простоти без налаштувань. LeafyPod, для тих, кому цікаві аналітика і рекомендації, але хто готовий виконувати дії вручну.

У межах даного дипломного проєкту буде обрано найкращі функції кожної з моделей, автономність LetPot, простоту Click and Grow і гнучкість LeafyPod, для створення власного пристрою, який зможе підтримувати ідеальні умови для росту рослини без участі користувача або з мінімальним втручанням.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		

1.3 Мобільні застосунки для моніторингу та догляду за рослинами

Окрім апаратної частини, сучасні системи автоматизованого догляду за рослинами все більше спираються на програмне забезпечення. Його завдання, не просто зчитувати дані з сенсорів, а структурувати їх, аналізувати і передавати користувачеві у зручній формі. На перетині IoT, мобільних застосунків і хмарних технологій народився цілий клас програмних рішень, які роблять догляд за зеленими улюбленцями доступним навіть для тих, хто не має ані часу, ані досвіду.

Почнемо з простих мобільних застосунків, які орієнтовані на взаємодію безпосередньо з користувачем. Наприклад, застосунок Plant Parent дозволяє додавати рослини вручну, а далі генерує графік поливу, підказки щодо добрив і пересадки. Програма працює на основі великої бази даних, де кожна рослина має типовий профіль з умовами утримання. Аналогічно функціонує Vera від бренду Bloomscare, тут акцент зроблено на візуальний контроль та нагадування. Обидва застосунки не інтегруються з апаратною частиною, але є зручними для щоденного користування .

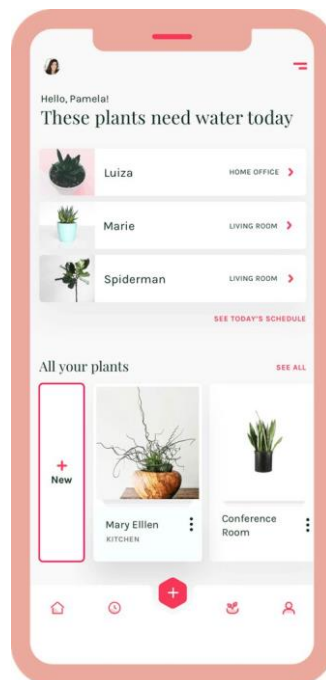


Рисунок 1.6 – Інтерфейс застосунку Vera екран із профілем рослини та нагадуванням

					КВРКІ 210123.21.01.77 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

На іншому рівні стоять програми, що взаємодіють із сенсорами та керуючими модулями. До прикладу, SmartPlant дозволяє підключити спеціальні датчики, які передають інформацію про вологість, температуру, освітленість. Застосунок аналізує отримані дані і генерує рекомендації, полити, змінити місце, або пересадити. Водночас Blynk, це вже не готове рішення, а платформа для створення інтерфейсів керування IoT-пристроями. Саме Blynk часто використовується в студентських та аматорських проєктах для візуалізації даних із сенсорів через ESP32 або подібні мікроконтролери.

Ще один важливий клас, універсальні системи розумного дому, що підтримують інтеграцію з багатьма пристроями одночасно. Зокрема, Home Assistant дозволяє створити повноцінну екосистему, у якій дані з сенсорів, керування насосами та навіть історія поливів зберігаються в одному середовищі. Завдяки відкритому коду, розширеній підтримці ESPHome та MQTT, користувач може налаштувати всі параметри відповідно до своїх потреб. І хоча для початківця це виглядає складно, Home Assistant стає надпотужним інструментом для глибокої автоматизації.

Особливої уваги заслуговують платформи, які використовують хмарне зберігання даних. Наприклад, застосунок OpenGarden синхронізується з Arduino або ESP32, дозволяючи передавати інформацію у вебінтерфейс. Цей тип архітектури дозволяє користувачеві контролювати умови не тільки з дому, а й із будь-якої точки світу. Крім того, відкритість коду таких систем дозволяє адаптувати їх під конкретні задачі, розширюючи функціональність практично без обмежень.

Порівнюючи всі ці рішення, можна сказати, що більшість комерційних застосунків орієнтовані на зручність і простоту. Вони обмежені в гнучкості, але легко освоюються. У той час як платформи на кшталт Home Assistant чи Blynk мають значно більший потенціал для кастомізації, однак вимагають технічної підготовки. Саме тому вибір програмного забезпечення у дипломному проєкті буде залежати від необхідного рівня інтерактивності: чи достатньо простого графіка

					КВРКІ 210123.21.01.77 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

поливу і сповіщень, чи потрібна повноцінна інтеграція з сенсорами, актуаторами і мережею.

Таблиця 1.2 – Порівняння типів ПЗ: мобільні застосунки, платформи, хмарні рішення, з прикладами

№	Пристрій	Мобільний застосунок	Платформа	Хмарне рішення
1	LetPot MP1	LetPot App	LetPot Web Dashboard	LetPot Cloud Service
2	Click & Grow	Click & Grow App	Click & Grow Web Dashboard	Обмежені хмарні функції (дані зберігаються в додатку)
3	LeafyPod	LeafyPod Mobile App	LeafyPod Web Platform	LeafyPod Cloud (через Bridge)
4	Ivy Smart Planter	Ivy Home App	Ivy Cloud Platform	Ivy Cloud Services
5	Parrot Flower Power	Parrot Flower Power App	Parrot Cloud	Parrot Cloud Services
6	Mi Flora Sensor	Mi Home App / Flower Care	Mi Home / Home Assistant (3rd party)	Xiaomi Cloud / Third-Party Cloud

Для цілей цього дослідження обрано Blynk як основну платформу візуалізації та налаштування, а також Home Assistant, як перспективу для подальшого розвитку проекту. Це дозволить поєднати простоту та гнучкість, а також забезпечити

користувачеві не тільки моніторинг, але й повноцінне дистанційне керування всіма компонентами системи.

1.4 Висновки

Автоматизований догляд за рослинами, це не просто модна тенденція, а цілком логічна відповідь на сучасні виклики. Брак часу, відсутність базових знань про рослини, бажання мати «розумний дім» із мінімальною участю користувача, усе це стимулює появу нових рішень на ринку. У ході аналізу програмно-технічних засобів стало зрозуміло, що більшість з них пропонують лише часткову автоматизацію або навпаки, занадто жорстко обмежують можливості користувача.

Порівняння пристроїв показало, що жоден із них не поєднує в собі повною мірою всі ключові якості: адаптивність, автономність, можливість віддаленого керування та налаштування. З одного боку, є моделі, які ідеально підходять для новачків, але не мають гнучкості. З іншого, складні у використанні платформи, які програмне забезпечення: зручні застосунки мають обмежений функціонал, а потужні системи потребують технічної підготовки.

Тому розробка власного програмно-технічного засобу в межах цього дипломного проєкту виглядає не просто доцільною, а й актуальною. Це дозволить створити рішення, яке враховує сильні сторони існуючих систем, але водночас позбавлене їхніх обмежень. Пристрій, що буде не тільки інформативним, але й гнучким у керуванні, простим у використанні та здатним працювати без участі людини упродовж тривалого часу.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ОБҐРУНТУВАННЯ ВИБОРУ КОМПОНЕНТІВ ТА СЕРЕДОВИЩА РЕАЛІЗАЦІЇ

2.1 Апаратне середовище реалізації

Створення програмно-технічного засобу для моніторингу та автоматизованого догляду за рослинами неможливе без продуманої апаратної архітектури, здатної не лише забезпечити базову функціональність, а й гарантувати надійність, енергоефективність, масштабованість і гнучкість в експлуатації. На відміну від статичних інженерних рішень, де компонентний склад зазвичай диктується одним вузьким сценарієм застосування, у випадку розумного мобільного горщика мова йде про поєднання одразу кількох інженерних викликів: рух у просторі, адаптивне зрошення, сенсорний контроль та бездротова передача даних.

Центральним компонентом системи обрано мікроконтролер ESP32, який забезпечує баланс між обчислювальною потужністю, низьким енергоспоживанням і широкими можливостями підключення. На відміну від менш продуктивних рішень, таких як Arduino UNO чи ATmega328, ESP32 надає одразу два ядра, підтримку Wi-Fi і Bluetooth, що відкриває шлях до віддаленого керування пристроєм через мобільний застосунок або вебінтерфейс. Крім того, ESP32 має достатню кількість входів/виходів, щоб зчитувати сигнали з численних сенсорів, а також керувати моторними драйверами та помпами.



Рисунок 2.1 – Плати ESP32

Особливу увагу при побудові апаратної архітектури було приділено організації сенсорного середовища. Пристрій включає чотири датчики освітленості, розміщені по периметру, що дозволяє аналізувати інтенсивність світла з різних боків та визначати, в якому напрямку слід пересунути горщик для оптимального фотосинтетичного ефекту. Серед доступних варіантів було розглянуто кілька популярних модулів, зокрема BH1750 та TSL2561. Остаточний вибір було зроблено на користь BH1750 через його простоту інтеграції з ESP32, низьке енергоспоживання та точність в умовах природного освітлення.[6]

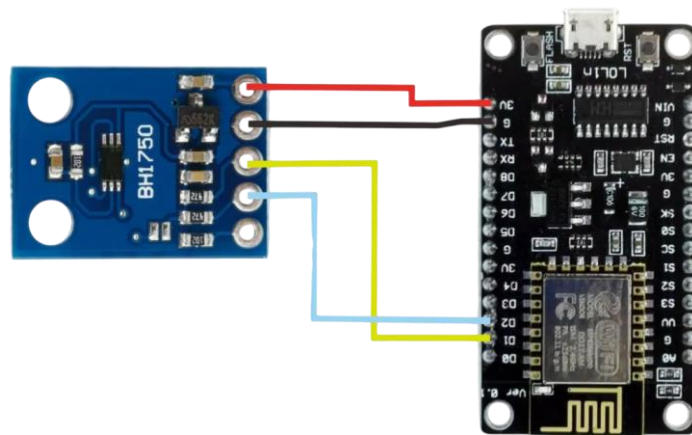


Рисунок 2.2 – Модуль BH1750 та схема його підключення до ESP32

Забезпечення оптимального рівня вологості ґрунту є не менш важливим завданням для системи, оскільки від цього безпосередньо залежить життєздатність рослини. Для моніторингу вологи використовуються ємнісні датчики вологості, розміщені на різній глибині у ґрунтовій частині горщика. Це дозволяє не просто фіксувати момент висихання верхнього шару, а й враховувати затримку води в нижніх шарах, забезпечуючи більш точне уявлення про загальний стан зволоження. На відміну від дешевих резистивних аналогів, які швидко кородують і дають нестабільні показники, ємнісні модулі мають довший строк служби та стабільніший сигнал.

Важливо підкреслити, що завдяки багаторівневому моніторингу вологості система може адаптувати алгоритм поливу залежно від типу рослини, структури

ґрунту та сезонних змін. Наприклад, для кактусів буде достатньо затримуватися з поливом до висихання глибших шарів, тоді як для тропічних рослин система зреагує ще до того, як ґрунт почне пересихати повністю.

Для організації процесу поливу використовується мікропомпа з резервуаром для води, яка керується за допомогою транзисторного ключа або реле. Помпа активується лише за умов виявлення низької вологості на основі заданих користувачем порогів. Таким чином, пристрій не просто періодично поливає рослину, а дійсно реагує на динаміку її потреб у зволоженні. Резервуар інтегровано в нижню частину корпусу, що сприяє стабільності конструкції й водночас спрощує заміну води [7].

Крім аналізу ґрунтового середовища, важливим елементом комплексного моніторингу є сенсори температури та вологості повітря, оскільки вони дозволяють виявляти мікрокліматичні умови довкола рослини, які можуть суттєво впливати на її розвиток. Після порівняння кількох доступних рішень, зокрема DHT11, DHT22 та VME280, вибір було зроблено на користь VME280, що забезпечує вищу точність, надійність, а також додаткову функцію вимірювання атмосферного тиску. Останній параметр має обмежене значення для догляду за окремими рослинами, однак може бути використаний для майбутніх експериментальних сценаріїв або розширення функціоналу системи.

Модуль VME280 інтегрується з ESP32 через інтерфейс I²C, що дозволяє зменшити кількість необхідних пінів і спростити дротове з'єднання. Його розміщено у верхній частині корпусу пристрою, ближче до відкритого повітря, що мінімізує тепловий вплив мікроконтролера та інших компонентів. Водночас сама плата захищена перфорованою кришкою, яка забезпечує доступ повітря при збереженні базового рівня захисту від пилу та вологи [8].

					КВРКІ 210123.21.01.77 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

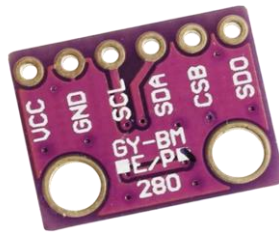


Рисунок 2.3 – Модуль VME280

Цікаво зазначити, що при одночасному аналізі параметрів ґрунту та повітря система може самостійно визначити, чи має місце випаровування води з поверхні, що дає змогу передбачити висихання навіть до моменту його реєстрації в нижніх шарах. Такий підхід дозволяє розумному пристрою працювати не лише як реактивна система, а й як превентивний механізм.

Усі сенсори, встановлені на пристрої, об'єднуються в єдину мережу з використанням гнучкого шлейфового підключення, що дозволяє уникнути надлишку жорстких дротів усередині корпусу. Це особливо важливо у конструкції з мобільними елементами, де зайві жорсткі з'єднання могли б перешкоджати руху або спричинити поломку. Крім того, корпус передбачає модульність, усі сенсори можуть бути легко замінені або доповнені новими, без потреби в повному демонтажі плати керування.

Однією з ключових функціональних відмінностей розробленого пристрою є його здатність пересуватись у просторі, адаптуючись до змін освітлення. Це потребувало впровадження незалежного мобільного шасі, оснащеного відповідними приводами, а також системою орієнтації в оточенні. Для реалізації цієї функції було обрано конфігурацію з чотирма колесами, два з яких ведучі, а два пасивні, обертаються вільно й забезпечують стабільність. Такий підхід забезпечує достатню маневреність при збереженні стійкості навіть на нерівних поверхнях, що важливо в умовах побутового середовища.

В якості рушійної сили використовуються постійного струму мотор-редуктори з вбудованими енкодерами, які дозволяють не лише приводити в рух

конструкцію, а й точно контролювати пройдену відстань, швидкість та напрямок. Керування моторами здійснюється через модуль драйвера L298N, який має достатню потужність для обслуговування двох моторів одночасно й забезпечує захист від перегрівання завдяки вбудованому радіатору. Хоча існують більш компактні рішення, як-от драйвери на базі TB6612FNG, однак саме L298N обрано через його сумісність із ESP32, високу надійність і широке розповсюдження у DIY-проектах [9].

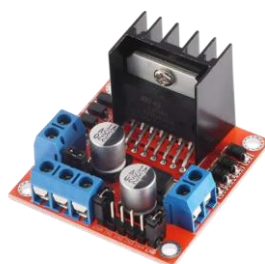


Рисунок 2.4 – Модуль L298N

Рухомий горщик, що функціонує у звичайному домашньому середовищі, стикається з численними об'єктами, які можуть перешкоджати його переміщенню. Для уникнення зіткнень до конструкції інтегровано інфрачервоні датчики перешкод (IR sensors) та ультразвуковий датчик HC-SR04 на передній панелі. Ультразвук дозволяє точно вимірювати відстань до об'єкта перед пристроєм, тоді як інфрачервоні сенсори реагують на близькість об'єктів збоку. Це дозволяє створити примітивну модель навігації, за якої пристрій здатен обрати найбільш вільний напрямок руху до точки з найкращим освітленням [10].

Цікаво, що поєднання даних з сенсорів освітлення і сенсорів перешкод дозволяє не просто обирати "найсвітліший" напрямок, а й розраховувати альтернативні шляхи до нього, що в перспективі може бути основою для впровадження простих алгоритмів автономної навігації з обходом перешкод. У цьому контексті ESP32 виступає як центральний координаційний вузол, обробляючи всі вхідні сигнали та приймаючи рішення в реальному часі.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

Будь-який автономний пристрій, особливо такий, що має здатність до пересування, потребує продуманої системи енергоживлення. У випадку розумного горщика, що виконує одночасно функції моніторингу, поливу та мобільного переміщення, питання джерела енергії стає критично важливим для забезпечення стабільної та безперебійної роботи.

У пристрої використовується акумуляторна батарея типу Li-ion 18650 з ємністю від 2000 до 3000 мА•год, залежно від конкретного конфігураційного варіанту. Таке джерело живлення обрано з огляду на його енергоемність, доступність, а також можливість багаторазового заряджання. Водночас важливим фактором стало співвідношення між розміром елемента та його здатністю підтримувати роботу ESP32, сенсорів, моторів і помпи протягом щонайменше 24 годин у стандартному режимі [11].

Акумулятор інтегрується з модулем захисту TP4056, який виконує одразу кілька важливих функцій: заряджання батареї через USB-роз'єм, контроль рівня заряду, запобігання надмірному розрядженню, а також захист від короткого замикання. Це дозволяє зробити процес заряджання безпечним і зручним, навіть у домашніх умовах, без необхідності в додаткових блоках живлення. Оскільки модуль TP4056 підтримує індикатор заряду, це дає змогу користувачу візуально контролювати стан акумулятора, наприклад, за допомогою LED-індикатора на корпусі пристрою або через інтерфейс мобільного застосунку.

Щоб запобігти втратам енергії та не допустити надмірного споживання в режимі очікування, на пристрої реалізовано перемикання режимів енергозбереження, що підтримується ESP32. У моменти бездіяльності (відсутність руху, стабільні показники вологості й температури) мікроконтролер переходить у режим deep sleep або light sleep, зберігаючи при цьому здатність реагувати на зовнішні сигнали або заплановані події [12].



Рисунок 2.5 – Модуль TR4056 [13]

У перспективі передбачено варіант розширення енергетичного блоку за допомогою сонячної панелі, яка дозволила б подовжити автономність пристрою в умовах природного освітлення. Такий підхід є особливо актуальним для балконного або тепличного використання, однак для початкової моделі було вирішено зосередитися на класичному акумуляторному рішенні з можливістю зарядки від побутової мережі.

Фізичне компонування всіх апаратних модулів всередині пристрою потребувало ретельного підходу до розробки корпусу, що поєднує естетику, практичність і легкість обслуговування. Основну платформу становить двохшарова конструкція, у якій нижній рівень відведено під рухову частину, енергетичний модуль та резервуар для води, тоді як верхній рівень виконує функцію контейнера для ґрунту з рослиною, разом із датчиками та зовнішніми елементами індикації.

Таке компонування дозволяє ефективно відокремити середовище, де можливе підвищене зволоження, від електронної частини, що критично важливо для безпеки та довговічності. Крім того, воно забезпечує низький центр ваги, що підвищує стійкість мобільного горщика навіть у момент різкого гальмування чи повороту під час руху.

Усі сенсори, двигуни та модулі комунікації з'єднані через уніфіковані конектори, що дозволяє швидко замінювати або оновлювати окремі частини без необхідності демонтажу всієї системи. Такий модульний підхід відкриває простір для подальших розширень, наприклад, інтеграції камер, додаткових сенсорів або нових алгоритмів орієнтації. ESP32 виступає в ролі централізованого керівного

вузла, який координує всі процеси та, при потребі, взаємодіє з зовнішніми пристроями через Wi-Fi.

Логіка взаємодії між модулями побудована за принципом поділу відповідальності: кожен елемент виконує конкретну функцію, а їхня синхронізація забезпечується прошивкою. Наприклад, сенсори освітленості формують цільову точку переміщення, енкодери двигунів відповідають за точність переміщення до цієї точки, а сенсори перешкод уточнюють допустимість цього маршруту в реальному часі. Після прибуття на обрану позицію автоматично запускається аналіз вологості ґрунту та ухвалюється рішення щодо поливу.

Підсумовуючи, варто зазначити, що апаратне середовище реалізації пристрою є результатом численних компромісів між потужністю, компактністю, надійністю та можливістю гнучкої адаптації. Усі обрані компоненти були підібрані не лише на основі їхньої сумісності, але й з урахуванням доступності на ринку, перевіреної репутації у середовищі DIY-розробників, а також наявності достатньої документації для інтеграції з ESP32. У майбутньому така архітектура дозволить реалізовувати складніші сценарії, від керування кількома рослинами до інтеграції з системами розумного дому.

2.2 Функційні вимоги

Для успішного створення програмно-технічного засобу, здатного забезпечити повноцінний автоматизований догляд за рослинами, необхідно чітко сформулювати функціональні вимоги, які визначають, що саме повинен уміти робити цей пристрій. На відміну від нефункціональних параметрів, що стосуються ефективності чи зручності, функціональні вимоги окреслюють конкретні дії та можливості, які система повинна реалізовувати на практиці.

Насамперед, основною функцією є збір даних про стан середовища, в якому перебуває рослина. Це охоплює безперервне вимірювання рівня освітленості з кількох напрямків, температури повітря, вологості як повітря, так і ґрунту на різній

					КВРКІ 210123.21.01.77 ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата		

глибині. Такі параметри є визначальними для здоров'я рослини й мають бути доступні в реальному часі для аналізу як самою системою, так і користувачем. Важливо, що ці дані не зберігаються пасивно, а одразу використовуються для прийняття рішень про полив або переміщення.

Друга важлива функція полягає у здійсненні поливу рослини, причому не за фіксованим графіком, а з урахуванням даних сенсорів. У разі, якщо показники вологості ґрунту нижчі за встановлену межу, система автоматично активує водяну помпу, подаючи об'єм води, необхідний для досягнення бажаного рівня. Цей процес не є миттєвим і вимагає контролю зворотного зв'язку, пристрій повинен переконатись, що після поливу вологість справді зросла, а якщо ні, повторити процедуру з урахуванням безпеки кореневої системи.

Ще однією визначальною функцією є аналіз інтенсивності освітлення у різних напрямках та самостійне переміщення в той бік, де світла більше. Це дає змогу рослині отримувати оптимальні умови для фотосинтезу без втручання людини. Така поведінка особливо корисна в приміщеннях, де джерела світла змінюються протягом дня. Система має самостійно визначати момент, коли наявне освітлення стало неефективним, і розраховувати найкращий напрямок руху з урахуванням поточних даних із сенсорів [14].



Рисунок 2.6 – Зображення прикладу логіки визначення напрямку руху на основі освітлення

Однією з фундаментальних функцій, без якої сучасні автоматизовані рішення втрачають свою гнучкість і привабливість, є можливість взаємодії користувача з пристроєм у зручному форматі. У цьому випадку мова йде про відображення всіх основних показників у мобільному застосунку або через вебінтерфейс, що підключається до пристрою по Wi-Fi. Система має у реальному часі надсилати інформацію про температуру, вологість, освітлення, рівень заряду акумулятора, а також відображати статус останніх дій, таких як полив чи переміщення.

Крім виводу інформації, важливою є можливість впливу користувача на роботу пристрою. До функціональних вимог належить здатність віддалено запускати полив або зупиняти його достроково, примусово вказувати напрямок руху або забороняти його в певні години (наприклад, у нічний час, коли рух може створювати шум). Користувач також має можливість змінювати допустимі межі параметрів, наприклад, задати нижню межу вологості ґрунту для конкретного типу рослини або допустимий рівень освітленості, за якого система має вважати середовище непридатним.

Серед функціональних вимог також можна виокремити обробку критичних ситуацій. У разі виявлення потенційної проблеми, як-от надмірне пересихання ґрунту, падіння температури нижче оптимального рівня, відсутність води в резервуарі або повна розрядка акумулятора, пристрій повинен не тільки фіксувати подію, а й надсилати сповіщення користувачеві з рекомендацією вжити заходів. Це забезпечує своєчасне реагування й підвищує довіру до автоматизованої системи як до надійного помічника, а не просто технічного аксесуара [15].

Ще однією важливою функцією є розпізнавання перешкод на шляху руху. Система має бути здатною виявити об'єкти, які можуть завадити переміщенню, і самостійно змінити маршрут або припинити рух, якщо обійти перешкоду неможливо. У цьому контексті взаємодія між сенсорами освітленості та сенсорами перешкод є критичною: пристрій не повинен рухатись лише за найкращим рівнем освітлення, якщо на шляху виявлено нерухомий об'єкт. Відповідно, функціональна

					КВРКІ 210123.21.01.77 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

модель має враховувати динамічне середовище й уміти адаптуватись до змін у ньому без зовнішнього втручання.

У структурі функціональних вимог важливу роль відіграють вторинні, але стратегічно важливі можливості, які не пов'язані з безпосереднім поливом чи переміщенням, однак значно підвищують загальну ефективність і зручність експлуатації системи. Насамперед ідеться про ведення журналу даних: пристрій має зберігати історію вимірювань, зокрема значення вологості, температури, рівня освітленості, а також дії, які були виконані (наприклад, дата й тривалість поливу, зміна місця розташування, рівень заряду акумулятора до та після руху). Ці дані дозволяють користувачеві відстежувати динаміку змін, аналізувати ефективність обраних параметрів догляду та приймати обґрунтовані рішення щодо корекції налаштувань.

Функція адаптивної калібрування сенсорів також входить до функціональних вимог. Система повинна мати можливість повторного калібрування після тривалого використання або заміни окремих модулів. Наприклад, користувач може вручну задати точку «максимальної вологості» при щойно политому ґрунті або вказати значення, що відповідає критичній сухості, даючи таким чином пристрою еталонні точки для перерахунку. Це дозволяє врахувати конкретні умови кожної рослини, унікальні характеристики ґрунту та типи ємностей для посадки [16].

Особливу увагу в архітектурі функціональності приділено енергозбереженню. Пристрій має вміти автоматично перемикатися в економний режим при відсутності потреби в активних діях. Наприклад, якщо показники навколишнього середовища стабільні, рівень вологості в межах норми, а освітлення не змінюється, система переходить у пасивний режим з мінімальним енергоспоживанням, де активними залишаються лише критично важливі модулі. Ця поведінка забезпечується функціоналом самого мікроконтролера ESP32 і повинна бути інтегрована на рівні прошивки.

Серед додаткових функцій варто також зазначити можливість оновлення прошивки пристрою віддалено, що дозволяє в майбутньому розширювати

					КВРКІ 210123.21.01.77 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

функціонал без фізичного доступу до внутрішніх компонентів. Це може бути реалізовано через Wi-Fi, а процес оновлення має запускатися за ініціативою користувача або автоматично, за умови виявлення нової стабільної версії. Така вимога є ключовою для підтримки актуальності системи в довгостроковій перспективі [17].

2.3 Нефункційні вимоги

Поряд із функціональними можливостями, які визначають, що саме має робити система, існує не менш важливий пласт характеристик, що описують як саме ці функції повинні реалізовуватись. Мова йде про нефункціональні вимоги, категорію параметрів, яка охоплює надійність, швидкодію, зручність у користуванні, безпеку, масштабованість, адаптивність та інші властивості, що формують загальне враження від системи, її практичну придатність і рівень технологічної зрілості.

Однією з ключових вимог є стабільність роботи пристрою у тривалому автономному режимі. Пристрій має функціонувати без збоїв щонайменше кілька діб поспіль навіть за умови мінімального втручання з боку користувача. Це означає, що всі підсистеми, від сенсорної до виконавчої, повинні бути здатні працювати в умовах постійного циклічного навантаження: зчитування, обробка, прийняття рішень, виконання дій. При цьому особливу увагу слід звернути на запобігання накопиченню помилок, наприклад таких, як зависання мікроконтролера, неправильне оновлення значень сенсорів чи некоректна реакція на сигнали з навколишнього середовища.

Не менш важливою є вимога реагування в режимі, наближеному до реального часу. Хоча повна синхронність між середовищем і пристроєм не завжди потрібна, проте затримка в реакції на критичне падіння вологості або виявлення перешкоди на маршруті не повинна перевищувати кількох секунд. Це потребує оптимізації алгоритмів обробки даних, а також ефективного розподілу пріоритетів, наприклад,

					КВРКІ 210123.21.01.77 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

вимірювання температури може бути відкладене на кілька хвилин, однак спрацювання сенсора уникнення перешкод має викликати миттєву реакцію [18].



Рисунок 2.7 – Приклад черги подій у системі

Окрему категорію нефункціональних вимог становить енергоефективність, яка є визначальною для автономних пристроїв із обмеженим живленням. Усі обрані компоненти системи повинні відповідати принципу мінімального споживання енергії, особливо в стані очікування. Мікроконтролер ESP32 повинен автоматично переходити в режим сну в моменти бездіяльності, а окремі сенсори, відключатися або оновлюватись рідше, якщо зміни в оточенні незначні. Ефективне енергоспоживання також означає правильне балансування між частотою оновлення даних та потребою в економії заряду, особливо в ситуаціях, коли пристрій використовує акумулятор без додаткового підживлення.

Ще одним вагомим критерієм, що визначає придатність пристрою до використання в реальних умовах, є масштабованість системи. Це означає, що архітектура програмно-апаратного комплексу повинна бути побудована так, аби у

майбутньому можливо було безболісно додати нові функції або модулі, наприклад, камеру для візуального моніторингу стану рослини, датчик вуглекислого газу чи модуль освітлення. Необхідно, щоби нові компоненти могли бути підключені через стандартні інтерфейси (I²C, UART, PWM) і підтримувались поточною або оновленою прошивкою без суттєвого переписування всієї логіки роботи пристрою.

Важливе місце серед нефункціональних вимог посідає безпека, як фізична, так і цифрова. Фізична безпека стосується конструкції корпусу, який має захищати електронні компоненти від випадкового проливання води, перегріву або механічного пошкодження. У свою чергу, цифрова безпека пов'язана з доступом до пристрою через мережу. Оскільки система працює з передачею даних по Wi-Fi, з'єднання повинно бути зашифроване, а доступ до інтерфейсу, обмежений паролем. Це важливо не лише з огляду на захист від зовнішнього втручання, але й для запобігання ненавмисній зміні налаштувань користувачем або іншими пристроями в мережі [19].

Особливої уваги потребує вимога зручності користування. Уся система повинна бути інтуїтивно зрозумілою, щоб навіть користувач без технічного досвіду міг налаштувати її, переглядати дані, змінювати порогові значення та отримувати сповіщення. Інтерфейс застосунку має бути адаптований для мобільних пристроїв, з простим меню, великими індикаторами поточних значень і чіткою структурою. Система повинна повідомляти користувача зрозумілою мовою про всі дії, що були здійснені або заплановані, а в разі виявлення помилок, надавати поради щодо їхнього усунення.

Нарешті, важливо, щоб система була адаптивною до різних сценаріїв використання. Пристрій не має бути жорстко прив'язаним до одного типу рослини чи умов, він повинен легко підлаштовуватися до кімнатного вазона, балконної рослини або горщика в теплиці. Це означає, що користувач повинен мати змогу гнучко змінювати налаштування відповідно до потреб конкретної рослини, а система, коректно функціонувати в широкому діапазоні параметрів, зокрема температур, вологості, тривалості освітлення.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

Однією з визначальних характеристик будь-якого автономного пристрою є його надійність, здатність стабільно працювати упродовж тривалого часу без втрати функціональності чи появи критичних помилок. У межах цього проєкту система повинна демонструвати повну працездатність щонайменше протягом кількох місяців без потреби в перепрошивці чи втручанні з боку розробника. Це передбачає коректну обробку всіх можливих ситуацій, зокрема нестандартних даних від сенсорів, короткочасну втрату з'єднання з мережею, перевантаження акумулятора чи відсутність реакції на команди. Усі ці сценарії мають бути передбачені на рівні програмної логіки, і система повинна автоматично відновлювати роботу після збоїв.

Особливої важливості набуває функція самодіагностики, здатності системи оцінювати власний стан і повідомляти користувача про внутрішні проблеми. Наприклад, якщо якийсь із сенсорів припинив передавати дані або значення залишаються незмінними впродовж тривалого часу, система має це виявити й запропонувати перевірити підключення. Так само у випадку, коли помпа активується, але показники вологості не змінюються, користувач має отримати повідомлення про можливу несправність механізму поливу або забруднення каналу. Така поведінка не лише підвищує зручність, а й сприяє довгостроковій надійності пристрою [20].

Система повинна також демонструвати толерантність до часткової відмови. Це означає, що в разі виходу з ладу одного з компонентів пристрій повинен продовжити роботу в обмеженому режимі. Наприклад, якщо несправний лише один датчик освітлення, система має оцінювати напрямок руху на основі інших трьох. Якщо розряджено акумулятор і заряд недоступний, пристрій повинен зберігати критичні функції як-от моніторинг стану рослини, навіть за відключеної функції переміщення. Такий рівень адаптивності є важливим у реальному побутовому застосуванні, де відмови окремих елементів є питанням часу, а не винятком.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

Останнім важливим аспектом нефункціональних вимог є інформативність. Користувач повинен мати можливість не лише бачити поточні значення параметрів, а й розуміти, які дії були здійснені, чому вони були здійснені саме так, та яка логіка стоїть за тим чи іншим рішенням. Наприклад, повідомлення не повинно виглядати як "здійснено полив", а скоріше як "рівень вологості впав нижче встановленого порогу 20%, активовано полив тривалістю 5 секунд". Такий рівень прозорості дозволяє не просто контролювати пристрій, а й вчитися взаємодіяти з ним як із розумним асистентом, а не «чорною скринькою».

2.4 Вибір мови програмування, бібліотек та середовища реалізації

Розробка програмного забезпечення для інтелектуального пристрою, що автоматизує догляд за рослинами, передбачає не лише створення алгоритмів керування сенсорами, моторами та поливом, а й проектування зручного інтерфейсу, реалізацію віддаленої взаємодії з користувачем і забезпечення стійкої багатокomпонентної взаємодії між апаратними й програмними модулями. У зв'язку з цим надзвичайно важливим етапом стало визначення методу та середовища реалізації програмної частини системи, що дозволяє досягти максимальної ефективності, стабільності та гнучкості у роботі.

Особливість цього проекту полягає у тому, що програмна реалізація охоплює одразу два рівні: вбудоване програмування мікроконтролера ESP32 та створення мобільного (або веб) застосунку, який виступає інтерфейсом взаємодії з користувачем. Тому вибір середовища мав здійснюватися з урахуванням сумісності, підтримки потрібних бібліотек, а також зручності розробки, тестування й відлагодження.

Для написання прошивки мікроконтролера було обрано мову програмування C++ у середовищі Arduino IDE, оскільки ця платформа підтримує ESP32 на офіційному рівні, має величезну спільноту розробників, а також надає готові бібліотеки для роботи з більшістю сенсорів, які використовуються в системі. Крім

					КВРКІ 210123.21.01.77 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

того, Arduino IDE дозволяє швидко тестувати зміни, завантажувати код по USB і забезпечує інструменти серійної діагностики, що є критично важливим на етапі налагодження. Окремо слід згадати, що альтернативні середовища, зокрема PlatformIO чи Espressif IDF, хоча й мають додаткові можливості, значно складніші у використанні для нетипових проектів, що охоплюють нестандартну логіку керування рухом, поливом та одночасну обробку великої кількості подій [21].

Архітектура програмного забезпечення на рівні мікроконтролера реалізована за принципом поділу на функціональні модулі, кожен із яких відповідає за певний клас задач: зчитування сенсорів, обробка даних, прийняття рішень, керування акторними модулями (помпа, мотори), взаємодія з користувачем. Такий підхід дозволяє не лише структурувати код, а й забезпечує зручність масштабування, наприклад, у разі додавання нових сенсорів необхідно буде лише включити додатковий модуль без перегляду всієї логіки роботи системи.

Керування всіма компонентами здійснюється в рамках подієво-орієнтованої моделі, де центральний цикл (loop) працює як диспетчер: він не просто виконує команди послідовно, а реагує на зміни стану сенсорів або інші події, наприклад, перевищення порогу вологості, зміна освітлення або сигнал користувача з мобільного додатку. Така логіка набагато ефективніша, ніж класичний підхід із циклічним опитуванням усіх модулів, оскільки дозволяє зменшити навантаження на процесор і уникнути затримок при критичних реакціях, наприклад, під час уникнення перешкод під час руху.

В окремий функціональний блок винесено обробку сигналів від мобільного застосунку, яка реалізована за допомогою вебсервера, що працює на самому ESP32. Такий підхід дозволяє уникнути зовнішніх серверів і забезпечує автономність пристрою, користувач може підключитися до пристрою напряму або через домашню мережу, відправити команди й отримати відповіді, не покладаючись на доступ до Інтернету. Обмін здійснюється через простий HTTP-інтерфейс: кожен запит (наприклад, отримати стан, надіслати новий поріг вологості, запустити

полив) викликає окрему функцію на пристрої, яка формує відповідь і виводить результат у зручному для людини форматі [22].

Оскільки одним із ключових завдань розумного пристрою є не лише автономна робота, але й можливість віддаленого контролю та конфігурування, необхідно було обрати платформу для реалізації простого, але функціонального інтерфейсу користувача. Серед кількох можливих варіантів, створення повноцінного Android-додатку, використання Bluetooth-зв'язку або розробка хмарного рішення, було обрано гібридну веборієнтовану модель, яка реалізується на базі вбудованого вебінтерфейсу мікроконтролера та підтримує відкриття з будь-якого браузера без потреби в установці додаткового ПЗ.

Цей підхід дозволяє користувачеві миттєво отримати доступ до пристрою з телефону, планшета або ноутбука, просто відкривши IP-адресу, надану ESP32. Сам вебінтерфейс реалізовано з використанням HTML, CSS та JavaScript, а також бібліотек на зразок Bootstrap для забезпечення адаптивного дизайну. Така реалізація дозволяє автоматично підлаштовувати вигляд інтерфейсу під розмір екрана, відображати основні дані у вигляді кольорових індикаторів і надавати зручні кнопки для виконання дій, наприклад, запустити полив, оновити порогові значення, перевірити рівень заряду батареї.

Однією з переваг такого рішення є його незалежність від операційної системи користувача. На відміну від мобільного застосунку, який потребує розробки під кожен платформу окремо (Android, iOS), вебінтерфейс працює всюди, де є браузер. Крім того, оновлення інтерфейсу відбувається централізовано: при зміні HTML-файлів на пристрої нова версія стає доступною миттєво без дій з боку користувача. Це істотно спрощує технічну підтримку й забезпечує збереження функціональності навіть для менш досвідчених користувачів [23].

Крім стандартного інтерфейсу, розглядається перспектива створення більш гнучкого мобільного застосунку на базі Blynk або MIT App Inventor, що дозволить розширити функціональність, наприклад, реалізувати push-сповіщення, історію вимірювань або більш інтерактивний візуальний контроль. Однак у базовій моделі

акцент зроблено на простоті, доступності та автономності, а отже, інтегрований вебсайт, що зберігається безпосередньо на ESP32, був визнаний найкращим стартовим варіантом.

Жодна система не може вважатися надійною до того моменту, поки її функціонування не буде перевірене в умовах, максимально наближених до реального середовища експлуатації. У зв'язку з цим наступним важливим етапом після реалізації базового програмного каркасу стало тестування програмного забезпечення, що охоплює як функціональні сценарії, так і граничні випадки, тобто ті ситуації, у яких система поводить себе нестандартно або наближається до потенційного збою.

У рамках тестування було побудовано імітаційне середовище, яке дозволяє перевіряти реакції модуля ESP32 без підключення до всіх фізичних сенсорів. Таке середовище базувалося на використанні штучно згенерованих даних, що підставлялись замість реальних показників. Наприклад, у змінну вологості вводилися значення нижчі за поріг, щоб перевірити, чи активується помпа. Аналогічно, моделювалися умови недостатнього освітлення або виявлення перешкод, щоб проконтролювати коректність логіки руху. Завдяки цьому стало можливим ізолювати й перевірити кожен модуль окремо, зокрема, реакцію на події, обробку сигналів, оновлення інтерфейсу, а також забезпечення стабільності при великій кількості подій, що надходять одночасно.

Особлива увага приділялась перевірці взаємодії між модулями: чи не блокує полив рух пристрою, чи правильно оновлюється інтерфейс після зміни даних, чи відправляється зворотний сигнал після завершення дії. Усі ці перевірки мають критичне значення для забезпечення безпечного функціонування системи в реальних умовах, наприклад, система не повинна одночасно запускати помпу та рухати пристрій, щоб уникнути розливу води або порушення балансу.

Крім базового тестування, проводилась також перевірка на стійкість до нестандартних ситуацій, серед яких були раптове вимкнення живлення, втрата з'єднання з Wi-Fi, зависання окремого модуля, подача некоректних команд з

					КВРКІ 210123.21.01.77 ПЗ	Арк. 34
Зм.	Арк.	№ докум.	Підпис	Дата		

інтерфейсу користувача. Кожен такий сценарій імітувався вручну, а результат оцінювався з точки зору стабільності системи та здатності самостійно відновитись. У більшості випадків ESP32 успішно перезапускався, зберігаючи стан до вимкнення, що підтверджує ефективність обраної архітектури [24].

На підставі проведеного аналізу можна впевнено стверджувати, що обраний метод реалізації програмного забезпечення, який передбачає написання прошивки для мікроконтролера ESP32 мовою C++ у середовищі Arduino IDE, а також створення вбудованого вебінтерфейсу на базі HTML, CSS та JavaScript, виявився оптимальним у контексті цілей і завдань даного проєкту. Така архітектура дозволила створити стабільну, модульну, енергоефективну та зручну систему, що одночасно здатна працювати автономно й взаємодіяти з користувачем у зручному форматі.

На відміну від більш складних або громіздких варіантів, які передбачають створення повноцінних застосунків, використання зовнішніх серверів або реалізацію власних протоколів зв'язку, запропонована реалізація орієнтована на простоту й ефективність. Усі основні функції, зчитування сенсорів, керування поливом, переміщення пристрою, аналіз подій, взаємодія з користувачем, винесено в окремі програмні модулі, що дозволяє легко розширювати функціональність без зміни загальної логіки системи.

Переваги такого підходу стають особливо очевидними у процесі тестування: програмна модель виявилася стійкою до типових помилок, легко адаптується до змін у конфігурації апаратних компонентів і має вбудовані механізми обробки збоїв. До того ж, розробка інтерфейсу, що зберігається безпосередньо в пам'яті мікроконтролера, забезпечує незалежність від сторонніх платформ, що є критичним у побутовому використанні, де користувач не бажає встановлювати окремі додатки.

У перспективі така програмна основа дозволяє впровадити додаткові функції, зокрема зберігання історії значень параметрів, відправку даних у хмару, візуалізацію графіків змін та навіть елементи машинного навчання для

прогнозування потреб рослини. Обрана модель достатньо гнучка, щоб інтегрувати ці елементи без потреби змінювати фундаментальні принципи побудови системи.

Отже, обраний метод реалізації програмного забезпечення повністю відповідає вимогам стабільності, масштабованості, ефективності й простоти впровадження. У поєднанні з модульною структурою та відкритою архітектурою пристрою він відкриває широкі можливості для подальшого розвитку та адаптації проєкту під нові потреби або середовища використання.

2.5 Висновки

У результаті проведеного аналізу та обґрунтування було сформовано цілісне бачення системи програмно-технічного засобу для моніторингу та автоматизованого догляду за рослинами з елементами робототехніки. Розгляд апаратного середовища дозволив детально визначити всі ключові компоненти пристрою: мікроконтролер, сенсори, систему поливу, мобільну платформу, енергоживлення та допоміжні модулі. Особлива увага приділялася їхній взаємодії, принципам розміщення, технічним характеристикам і причині вибору кожного з них з урахуванням реальних сценаріїв експлуатації.

У межах функціональних вимог було визначено набір основних дій, які система повинна виконувати для забезпечення повноцінного автономного догляду за рослиною: моніторинг навколишнього середовища, адаптивне зрошення, рух до джерела світла, уникнення перешкод і сповіщення користувача. У свою чергу, нефункціональні вимоги окреслили рамки ефективності реалізації цих функцій, зокрема, стабільність роботи, енергоефективність, масштабованість, адаптивність до різних умов і простоту використання для кінцевого користувача.

Усі ці елементи в сукупності формують надійну, гнучку й адаптивну систему, яка може ефективно функціонувати в умовах автономної експлуатації, забезпечуючи стабільний догляд за рослинами з мінімальним втручанням людини.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНО-ТЕХНІЧНИЙ ЗАСІБ ДЛЯ МОНІТОРИНГУ ТА АВТОМАТИЗАЦІЇ ДОГЛЯДУ ЗА РОСЛИНАМИ З ЕЛЕМЕНТАМИ РОБОТОТЕХНІКИ

3.1 Загальна архітектура програмно-технічного засобу для догляду за рослинами

Робота запропонованого програмно-технічного засобу побудована на основі безперервної взаємодії між сенсорною системою, алгоритмічним ядром пристрою та виконавчими механізмами. Його головною метою є забезпечення повного циклу догляду за рослиною в автономному режимі з мінімальним втручанням користувача. Увесь функціонал реалізується в рамках єдиної логіки, де рішення ухвалюються на основі поточних параметрів навколишнього середовища, що постійно оновлюються.

Після подачі живлення пристрій ініціалізує всі підключені компоненти, зокрема мікроконтролер, сенсори вологості, температури, освітлення, модулі зв'язку та вебінтерфейс. Далі розпочинається основний цикл роботи, який виконується у вигляді послідовності взаємозалежних етапів. Першим кроком є зчитування даних із сенсорів, система отримує свіжі значення вологості ґрунту з кількох рівнів, фіксує температуру повітря, відносну вологість і визначає інтенсивність освітлення з чотирьох напрямків. Отримані дані передаються до логічного ядра, де проходять первинну перевірку на коректність, порівнюються з раніше збереженими значеннями та поточними порогоми.

У разі, якщо один або кілька параметрів виходять за межі допустимих значень, пристрій переходить у стан активної дії. Наприклад, якщо виявлено недостатній рівень вологості у ґрунті, мікроконтролер активує електропомпу, яка подає воду з резервуару до кореневої зони рослини. При цьому, щоб уникнути надмірного зволоження, передбачено кілька рівнів контролю: система подає воду невеликими порціями, після кожної з яких перевіряється фактична зміна вологості. Якщо після трьох спроб вологість не зростає, пристрій припиняє полив і надсилає

сповіщення про можливу несправність системи або пересохлий ґрунт, який потребує втручання людини.

Крім контролю вологості та поливу, одним із ключових елементів автономної поведінки пристрою є його здатність до орієнтації на освітлення, тобто пересування в напрямку, де рівень світла є оптимальним для фотосинтетичних процесів. Для цього використовується набір сенсорів освітленості, розташованих з усіх боків корпусу, що дає змогу формувати цілісну картину інтенсивності природного освітлення з різних напрямків. Зібрані значення обробляються та аналізуються із застосуванням порогової логіки, яка визначає, чи існує значуща різниця між сторонами, достатня для ініціації переміщення.

Якщо виявлено, що одна зі сторін освітлена суттєво краще, ніж інші (наприклад, перевищення на 25% у порівнянні з поточним положенням), система переходить у режим навігації. Мікроконтролер розраховує напрямок руху до найсвітлішої ділянки, при цьому враховується наявність можливих перешкод, про які повідомляють ультразвукові або інфрачервоні датчики. У разі виявлення перепон на оптимальному маршруті система самостійно обирає альтернативну траєкторію, якщо така доступна, або відкладає переміщення до моменту звільнення шляху.

Сам процес руху реалізується з урахуванням повільної швидкості й коротких імпульсів керування моторами, що дозволяє точно позиціонувати пристрій навіть у тісному середовищі. Після прибуття в нову точку система фіксує нове положення, повторно вимірює освітлення й приймає рішення про доцільність зупинки. Весь цей процес супроводжується збереженням енергетичного балансу, якщо рівень заряду акумулятора опускається нижче визначеної межі, переміщення не виконується, навіть за наявності кращих умов освітлення.

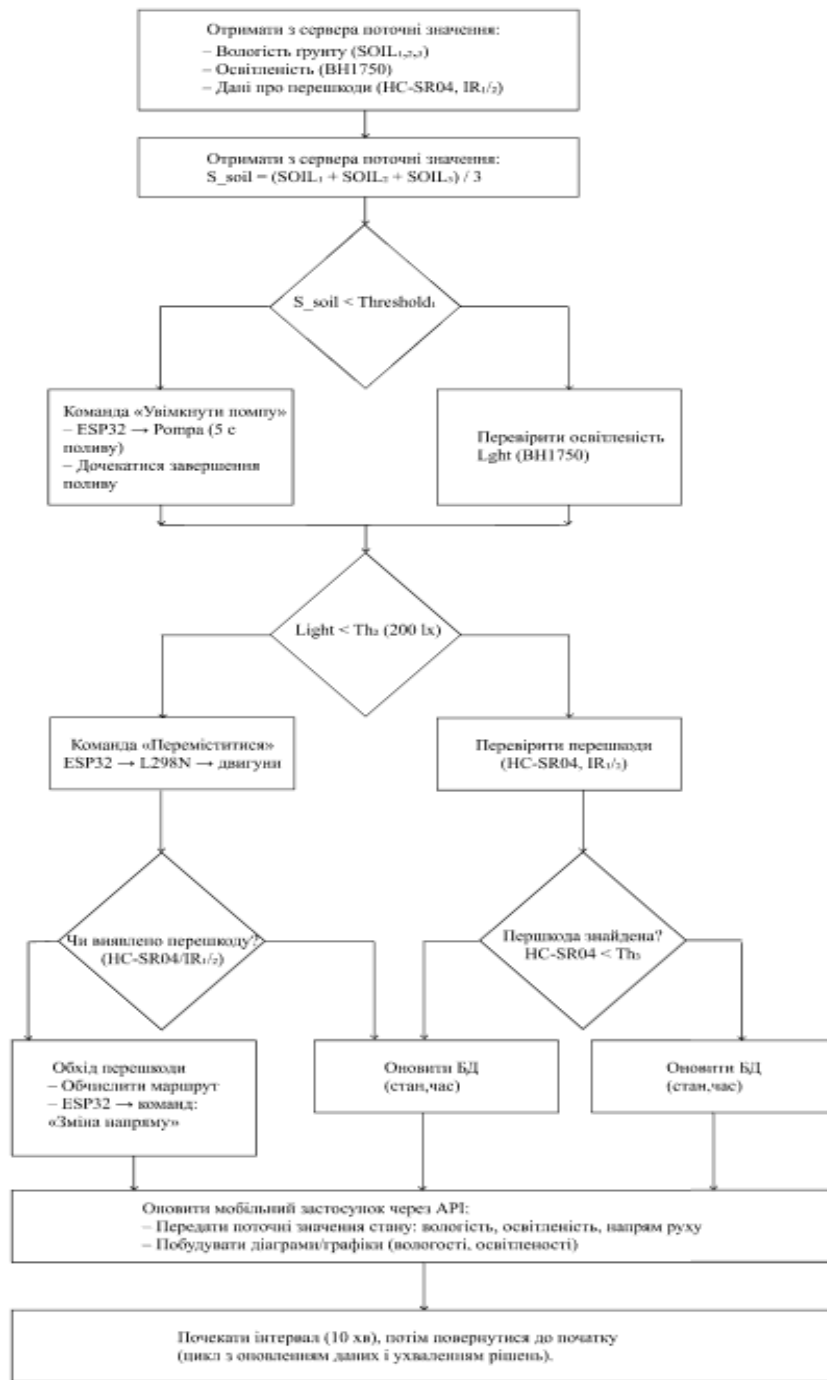


Рисунок 3.1 – Графічне зображення методу роботи інформаційної системи для догляду за рослинами

Слід наголосити, що всі дії системи носять контекстний характер. Це означає, що пристрій не діє автоматично при кожній зміні параметра, а лише тоді, коли поєднання кількох факторів сигналізує про необхідність втручання. Наприклад, не

кожне зниження вологості викликає полив, враховується і температура повітря, і поточна вологість на різних рівнях ґрунту, і час останнього зрошення. Такий багатофакторний підхід формує основу інтелектуальної поведінки, яка максимально наближена до рішень, які прийняв би досвідчений доглядач за рослинами.

Ще одним важливим елементом функціонування системи є взаємодія з користувачем, що реалізується через вбудований вебінтерфейс. Цей інтерфейс запускається безпосередньо на мікроконтролері ESP32 і не потребує зовнішнього сервера або спеціального застосунку. Після запуску пристрою на ньому активується локальний вебсервер, який дозволяє користувачу з будь-якого пристрою з Wi-Fi (смартфон, ноутбук, планшет) підключитися до системи, відкрити інтерфейс у браузері та переглянути поточний стан параметрів.

Інтерфейс надає можливість переглядати температуру повітря, рівень вологості ґрунту на різних рівнях, інтенсивність освітлення з кожного боку, заряд акумулятора, історію останніх дій та активних повідомлень. Окрім відображення інформації, користувач має змогу керувати пристроєм у ручному режимі, наприклад, примусово активувати полив, змінити цільові порогові значення вологості, заблокувати переміщення або оновити інтервал циклічного опитування сенсорів.

Обмін даними між користувачем і пристроєм відбувається через стандартні HTTP-запити. Наприклад, натискання кнопки «полив» у браузері надсилає запит типу GET або POST до вебсерверу ESP32, який інтерпретує команду, виконує дію та повертає відповідь із результатом. Усі подібні запити обробляються в реальному часі, а результати одразу відображаються в інтерфейсі без потреби в перезавантаженні сторінки, що реалізовано завдяки JavaScript.

Особливістю реалізації є також наявність системи повідомлень, яка дозволяє користувачу дізнатися про події, що відбулися за відсутності активної сесії. Наприклад, при надмірному пересиханні ґрунту або несправності сенсора система фіксує інцидент, зберігає його у вигляді повідомлення й виводить при наступному

					КВРКІ 210123.21.01.77 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

підключенні. Це підвищує рівень контролю та дозволяє аналізувати, чи правильно працює система протягом тривалого періоду.

Платформа також підтримує автоматичне оновлення порогів, які можуть змінюватись залежно від вибраного типу рослини. У майбутньому інтерфейс може бути доповнений вибором з бази даних рослин, яка зберігатиметься локально або синхронізуватиметься з хмарним сервером. Наразі вся конфігурація зберігається у пам'яті пристрою та автоматично підвантажується під час кожного запуску.

Однією з особливостей розумного пристрою, що працює в автономному режимі, є його здатність адаптувати інтенсивність своєї активності до поточних умов, зокрема зменшувати енергоспоживання під час простою. Цей принцип реалізовано через поетапне перемикання між активним, контрольованим та пасивним станами системи. У активному стані пристрій виконує конкретну дію: полив, переміщення, передавання даних або взаємодію з користувачем. У контрольованому, здійснює постійний моніторинг, з періодичним опитуванням сенсорів. А у пасивному, переходить у режим зниженого енергоспоживання, де працюють лише найнеобхідніші модулі.

Цей перехід координується програмною логікою, яка оцінює частоту змін середовища та важливість параметрів. Наприклад, якщо протягом останніх трьох циклів вологість ґрунту була стабільною, рівень освітлення суттєво не змінився, а температура перебуває в допустимих межах, пристрій автоматично знижує частоту опитування сенсорів і зменшує активність вебінтерфейсу, залишаючи лише базову відповідь на запити користувача. Це дозволяє знизити навантаження на акумулятор і продовжити термін автономної роботи без втрати контрольованості.

Важливу роль у цій логіці відіграє вбудована система таймерів і подій, яка координує черговість і частоту виконання дій. Наприклад, опитування сенсорів вологості може відбуватись раз на 10 хвилин, але при наближенні значень до порогових рівнів частота вимірювань зростає до однієї хвилини. Аналогічно, система може повністю вимкнути рухові модулі, якщо впродовж кількох годин не

спостерігається зміни рівня освітлення. Такий підхід забезпечує динамічне ресурсне планування, що важливе для автономних систем.

Крім того, пристрій зберігає інформацію про свій останній стан у пам'яті, щоб після перезапуску (наприклад, унаслідок розрядження або оновлення) відновити логіку роботи з тієї точки, на якій вона була перервана. Це дозволяє уникнути повторного запуску всіх процесів і зменшує кількість неконтрольованих дій. Наприклад, якщо полив тривав на момент вимкнення, після увімкнення система не запускатиме його повторно, а перевірить рівень вологості й вирішить, чи є в цьому потреба.

Таким чином, реалізація переходів між режимами роботи, це не лише питання енергоефективності, а й важлива частина інтелектуальної поведінки пристрою, що дозволяє йому адаптуватись до природних змін без надмірної активності та з урахуванням збереження ресурсу окремих компонентів.

Однією з особливостей розумного пристрою, що працює в автономному режимі, є його здатність адаптувати інтенсивність своєї активності до поточних умов, зокрема зменшувати енергоспоживання під час простою. Цей принцип реалізовано через поетапне перемикання між активним, контрольованим та пасивним станами системи. У активному стані пристрій виконує конкретну дію: полив, переміщення, передавання даних або взаємодію з користувачем. У контрольованому, здійснює постійний моніторинг, з періодичним опитуванням сенсорів. А у пасивному, переходить у режим зниженого енергоспоживання, де працюють лише найнеобхідніші модулі.

Цей перехід координується програмною логікою, яка оцінює частоту змін середовища та важливість параметрів. Наприклад, якщо протягом останніх трьох циклів вологість ґрунту була стабільною, рівень освітлення суттєво не змінився, а температура перебуває в допустимих межах, пристрій автоматично знижує частоту опитування сенсорів і зменшує активність вебінтерфейсу, залишаючи лише базову відповідь на запити користувача. Це дозволяє знизити навантаження на акумулятор і продовжити термін автономної роботи без втрати контрольованості.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

Важливу роль у цій логіці відіграє вбудована система таймерів і подій, яка координує черговість і частоту виконання дій. Наприклад, опитування сенсорів вологості може відбуватись раз на 10 хвилин, але при наближенні значень до порогових рівнів частота вимірювань зростає до однієї хвилини. Аналогічно, система може повністю вимкнути рухові модулі, якщо впродовж кількох годин не спостерігається зміни рівня освітлення. Такий підхід забезпечує динамічне ресурсне планування, що важливе для автономних систем.

Крім того, пристрій зберігає інформацію про свій останній стан у пам'яті, щоб після перезапуску (наприклад, унаслідок розрядження або оновлення) відновити логіку роботи з тієї точки, на якій вона була перервана. Це дозволяє уникнути повторного запуску всіх процесів і зменшує кількість неконтрольованих дій. Наприклад, якщо полив тривав на момент вимкнення, після увімкнення система не запускатиме його повторно, а перевірить рівень вологості й вирішить, чи є в цьому потреба.

Таким чином, реалізація переходів між режимами роботи, це не лише питання енергоефективності, а й важлива частина інтелектуальної поведінки пристрою, що дозволяє йому адаптуватись до природних змін без надмірної активності та з урахуванням збереження ресурсу окремих компонентів.

3.2 Алгоритми роботи програмно-технічного засобу для автоматизації догляду за рослинами

Функціонування програмно-технічного засобу базується на цілій низці алгоритмів, які охоплюють як планові, так і реактивні дії. На відміну від класичних автоматизованих систем, де сценарії роботи визначені суворо за часом або командами користувача, у цьому випадку алгоритми є динамічними та умовно-залежними, тобто пристрій самостійно визначає, які дії виконувати, коли, і в якій послідовності. Це досягається за рахунок багаторівневої логіки обробки даних,

					КВРКІ 210123.21.01.77 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

циклічного моніторингу, управління пріоритетами подій і синхронізації між модулями.

У центрі всієї системи стоїть основний керуючий цикл, що є класичним для мікроконтролерної архітектури. Цей цикл виконується безперервно і складається з кількох ключових етапів: ініціалізації, зчитування сенсорів, аналізу стану, виконання дій, оновлення інтерфейсу та переходу в очікування. На кожному з цих етапів діє окремий алгоритм із чітко визначеним входом, виходом і умовами виклику.

Першим етапом є алгоритм ініціалізації, який активується при кожному запуску або перезапуску пристрою. Він перевіряє наявність підключених модулів, проводить початкове зчитування даних, відновлює попередні налаштування з пам'яті й формує базову структуру змінних для подальшої роботи. Якщо на цьому етапі виявлено помилки, наприклад, відсутність сенсора або низький рівень заряду, ініціалізація завершується частково, а користувачу надсилається повідомлення з описом проблеми.

Після успішної ініціалізації система переходить до алгоритму сенсорного моніторингу. Тут пристрій послідовно зчитує значення вологості з ґрунтових датчиків на різній глибині, температуру та вологість повітря, інтенсивність освітлення з усіх боків, а також рівень заряду акумулятора. Усі ці дані проходять етап попередньої обробки: відфільтровуються коливання, що можуть бути спричинені шумом, обчислюються середні значення, а також зберігаються зміни відносно попереднього циклу.

Якщо дані вказують на стабільну ситуацію, наприклад, усі параметри в межах норми, пристрій переходить у режим очікування або зниження частоти оновлення. Але якщо хоча б один параметр досягає порогового значення, активується відповідний спеціалізований алгоритм, наприклад, поливу або переміщення.

Одним із ключових функціональних процесів є алгоритм поливу, який активується в разі виявлення недостатнього рівня вологості в ґрунті. На відміну від фіксованих систем, що працюють за таймером, у цій моделі полив відбувається

					КВРКІ 210123.21.01.77 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

умовно-залежно, з урахуванням поточних даних сенсорів, попередньої історії зрошення, типу рослини та зовнішніх факторів, таких як температура повітря. Таким чином, дія виконується лише тоді, коли це обґрунтовано фактичним станом рослини.

Алгоритм запускається при перевищенні критичного порогу сухості, заданого користувачем або системою за замовчуванням. Для уникнення помилкових спрацювань використовується подвійна перевірка: значення нижнього сенсора повинно відповідати критичному рівню щонайменше два цикли поспіль, а значення верхнього, бути достатньо низьким для підтвердження загального пересихання. Якщо ці умови виконані, система активує керуючий модуль помпи, яка подає визначену кількість води.

Після першої порції води система переходить до етапу контролю результату: вона вимірює вологість знову й порівнює нове значення з попереднім. Якщо вологість зросла щонайменше на визначений відсоток (наприклад, 5–10%), полив вважається успішним і фіксується в журналі подій. Якщо ж змін не виявлено, алгоритм повторює полив ще один або два рази, із меншою тривалістю, після чого, у разі подальшої безрезультатності, припиняє процес і формує сповіщення про можливу несправність (забита помпа, пересохлий ґрунт, відсутність води в резервуарі).

Такий підхід дозволяє уникнути надмірного зрошення та витрати води, одночасно забезпечуючи надійність роботи й захист від помилок. Алгоритм також враховує температуру повітря: при високих температурах допустимий поріг вологості може бути автоматично підвищений, щоб компенсувати швидке випаровування. У перспективі в цей цикл може бути інтегрований модуль прогнозування, який враховуватиме дані за попередні дні для ще точнішого регулювання об'єму води.

У разі переривання поливу, наприклад, через зниження напруги в акумуляторі або втрату зв'язку з сенсорами, система автоматично переходить у

					КВРКІ 210123.21.01.77 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

режим аварійної зупинки. Вона відключає живлення на помпу, блокує рух пристрою та записує останній стан у пам'ять для подальшого аналізу.

Друга за важливістю дія після поливу, це переміщення пристрою до зони з кращим освітленням, що забезпечує оптимальні умови для фотосинтезу та росту рослини. На відміну від статичних систем, де положення рослини не змінюється, запропонований пристрій здатен самостійно шукати світло, орієнтуючись на дані з чотирьох сенсорів освітленості, розташованих з усіх боків конструкції.

Алгоритм переміщення активується, коли фіксується стійка нерівномірність освітлення: одна зі сторін отримує суттєво більше світла, ніж інші. Зазвичай для активації необхідна різниця в 20–30% між поточним напрямком і найяскравішим боком, яка зберігається протягом кількох вимірювань. Це дозволяє уникнути випадкового руху внаслідок тимчасових змін освітлення, наприклад від проходу людини чи тіні предмета.

Після визначення цільового напрямку система виконує перевірку шляху за допомогою ультразвукового та інфрачервоного сенсорів. Якщо попереду немає перешкод, або вони перебувають на безпечній відстані, пристрій обирає траєкторію та вмикає мотор-редуктори. Рух відбувається короткими імпульсами, із зупинками для проміжного контролю координат та нового зчитування рівня освітленості. Такий підхід дозволяє пристрою адаптувати напрямок у процесі руху, якщо освітлення зміниться або з'являться нові перешкоди.

Якщо на шляху виявляється нездоланна перешкода, пристрій спробує знайти обхідний шлях, обираючи інші напрямки із достатнім освітленням. Якщо жодна траєкторія не є безпечною, рух відкладається, а інформація про заблокований напрямок фіксується в системі для повторного аналізу через певний час.

Після досягнення нової позиції пристрій виконує фіксацію положення. Вимірюється освітлення, знову перевіряється вологість, і лише після того, як усі показники залишаються стабільними протягом кількох циклів, рух вважається завершеним. Це дозволяє уникнути «миготіння» між позиціями, яке могло б виникати при частих дрібних змінах освітлення.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 46
Зм.	Арк.	№ докум.	Підпис	Дата		

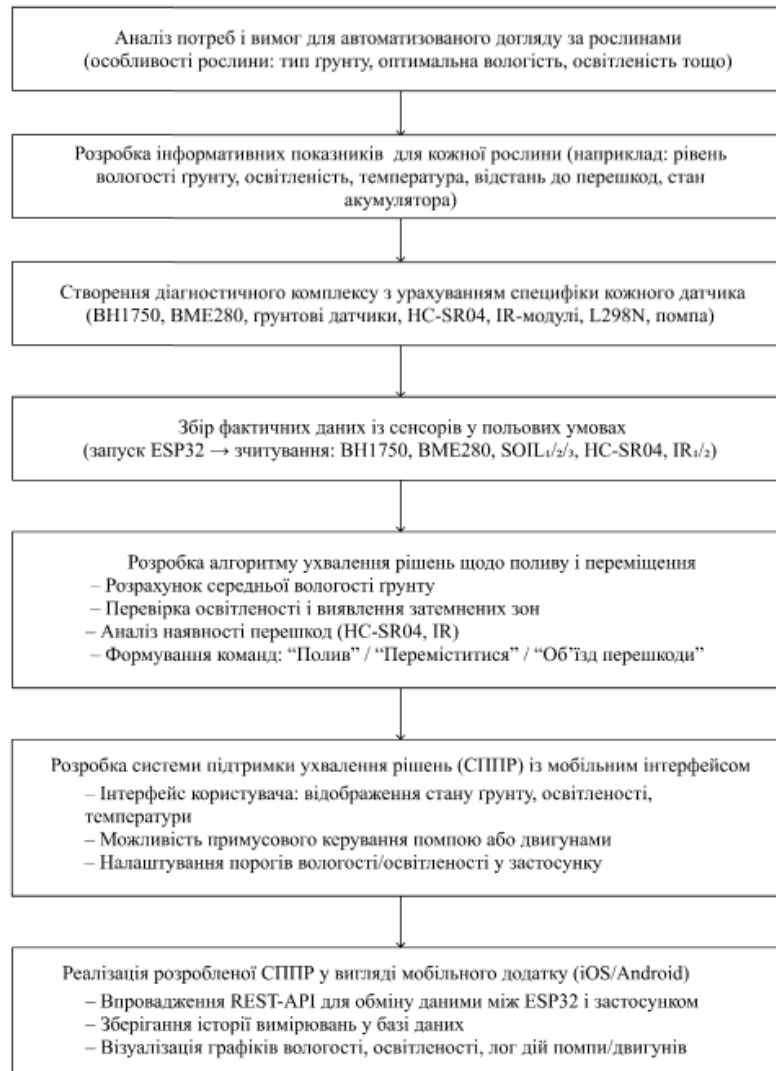


Рисунок 3.2 – Алгоритм ухвалення рішення

Таким чином, алгоритм переміщення пристрою побудований не на простому реагуванні, а на багатофакторному аналізі й постійному зворотному зв’язку, що дозволяє забезпечити плавну, безпечну та обґрунтовану зміну положення рослини з урахуванням усіх ризиків і умов середовища.

Попри високий рівень автономності, програмно-технічний засіб передбачає можливість втручання з боку користувача, що реалізується через вбудований вебінтерфейс. Усі команди, які надходять із браузера, обробляються окремим програмним модулем, що працює паралельно з головним циклом пристрою. Це

дозволяє негайно реагувати на запити, не чекаючи завершення поточної ітерації основного циклу.

Після отримання HTTP-запиту (наприклад, «почати полив», «зупинити рух», «змінити поріг вологості»), вебсервер на ESP32 аналізує тип запиту, перевіряє його допустимість на поточному етапі й приймає рішення, чи можна виконати дію негайно, чи слід додати її до черги виконання. Наприклад, якщо пристрій уже рухається, але користувач натискає кнопку «полив», команда не буде проігнорована, але спрацює після завершення поточного процесу. Це дозволяє уникнути конфліктів між модулями та небажаних наслідків.

Система підтримує також режим пріоритету користувача, який може бути активований вручну. У цьому режимі будь-яка команда з вебінтерфейсу має перевагу над алгоритмами автономної поведінки. Наприклад, навіть якщо освітлення в іншому напрямку є кращим, але користувач заблокував рух, пристрій залишиться в поточній позиції до подальших вказівок. Цей механізм особливо важливий у ситуаціях, коли пристрій працює в середовищі з непередбачуваними змінами, наприклад, у присутності людей, домашніх тварин чи нестабільного освітлення.

Окрему роль у роботі системи відіграють налаштування параметрів, які змінюються з інтерфейсу: пороги вологості, дозволені години для руху, тривалість поливу, затримка між циклами тощо. Усі ці значення зберігаються у постійній пам'яті ESP32 та автоматично завантажуються при кожному запуску. Після зміни налаштування нове значення одразу вступає в дію, не потребуючи перезавантаження.

У разі виникнення конфліктів, наприклад, якщо користувач намагається запуснути полив при майже повному заряді акумулятора або рух при виявленій перешкоді, система відмовляється від виконання дії та формує зворотний зв'язок із поясненням причини. Таким чином, забезпечується захист як користувача, так і пристрою від неправильного або небезпечного використання, без потреби глибоко розуміти всі технічні нюанси.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 48
Зм.	Арк.	№ докум.	Підпис	Дата		

У реальному середовищі програмно-технічний засіб стикається з необхідністю виконання кількох завдань одночасно, при цьому кожне з них має свою важливість і обмеження за часом. Щоб уникнути конфліктів і гарантувати стабільність, пристрій використовує систему пріоритетів і чергування подій, яка дозволяє керувати всіма діями в межах одного контрольованого циклу.

Усі можливі події, від виявлення низької вологості до натискання кнопки користувача, розподіляються за пріоритетними рівнями. Найвищий пріоритет мають критичні події, пов'язані з безпекою пристрою чи рослини: перегрів, порожній бак, зіткнення. Далі йдуть операції з прямим впливом на рослину: полив, переміщення. Ще нижче, періодичні оновлення інтерфейсу та запити користувача. Найнижчий пріоритет, планове збереження даних, оновлення журналу тощо.

Система обробки подій реалізована як круговий цикл, що на кожній ітерації перевіряє, чи є активні завдання в черзі, та виконує їх відповідно до пріоритету. Якщо в черзі немає подій високого пріоритету, пристрій продовжує планову роботу. Якщо з'являється подія, яка перебиває поточну, наприклад, раптове падіння заряду акумулятора, виконання циклу переривається, і активується відповідний обробник.

Цей механізм реалізує подієво-орієнтовану логіку, що дозволяє миттєво реагувати на зміни середовища, не чекаючи завершення рутинних дій. Наприклад, якщо під час поливу пристрій виявляє перешкоду або втрату з'єднання з сенсором, він не завершуватиме процес формально, а відразу зупинить подачу води та увімкне діагностичний режим.

Кожна завершена дія автоматично фіксується в внутрішньому журналі подій, що зберігається у flash-пам'яті. Це дозволяє у разі аварійного перезапуску або втрати живлення відновити останній стан і завершити незакінчені процеси. Наприклад, якщо пристрій перервав полив на 50%, при наступному запуску він не повторює повний цикл, а лише перевіряє актуальність задачі, виконує компенсацію й записує новий результат.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

У підсумку, реалізовані алгоритми дозволяють пристрою зберігати стабільність у складних умовах, гнучко реагувати на зовнішні й внутрішні зміни, а також гарантувати відтворюваність поведінки навіть при тривалому автономному функціонуванні. Взаємодія між окремими блоками реалізована не через жорстко фіксовану послідовність, а через гнучкий механізм пріоритетів, що імітує адаптивну поведінку живої системи.

3.3 Логіка роботи пристрою в складних умовах

Будь-який інтелектуальний пристрій, що діє автономно в умовах нестабільного зовнішнього середовища, неминуче стикається з потенційними ризиками порушення коректної роботи. Це можуть бути як внутрішні збої, пов'язані з апаратним забезпеченням або помилками у виконанні коду, так і зовнішні чинники, які впливають на роботу сенсорів або зв'язку. В контексті програмно-технічного засобу, призначеного для догляду за рослинами, питання стабільності функціонування набуває особливого значення, адже йдеться не лише про збереження даних або підтримання з'єднання, а й про фізичну взаємодію із середовищем, що має прямі наслідки для біологічного об'єкта, рослини.

З огляду на ці вимоги, система проектувалась із врахуванням принципів відмовостійкості, самодіагностики та адаптивного реагування на несправності. Центральне місце в архітектурі пристрою займає механізм багаторівневої перевірки цілісності, який забезпечує аналіз стану не лише на рівні сенсорних зчитувань, а й на рівні логіки роботи основного програмного ядра. Перед початком кожного циклу опитування пристрій оцінює свою працездатність у декількох аспектах, починаючи від коректності збережених у пам'яті конфігурацій до справності підключених модулів.

Особливу увагу приділено обробці аномальних або сумнівних даних. Наприклад, якщо значення з датчика вологості ґрунту виходить за межі фізично можливого діапазону або суттєво відрізняється від попередніх значень, програма

					КВРКІ 210123.21.01.77 ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

не включає це зчитування до подальшої обробки. Замість цього активується алгоритм повторного опитування, який дає змогу переконатись, чи справді стався збій, чи йдеться лише про тимчасове електричне перешкодження або вплив довкілля. Якщо й друге зчитування повертає аномалію, система звертається до збережених раніше валідних значень, що використовуються як орієнтовна основа для ухвалення рішень.

Цей підхід дозволяє мінімізувати ризик неправильного виконання дій, наприклад, уникнути поливу через хибне спрацювання сенсора, яке могло б призвести до перезволоження. На рівні взаємодії із виконавчими модулями (насосом, двигунами руху) реалізовано додатковий шар перевірки зворотного зв'язку. Після подачі команди на запуск помпи система через кілька секунд очікує зміну вологості, а якщо її не відбувається, розцінює це як непрацюючу подачу води. У такому випадку полив зупиняється, а пристрій переходить у режим очікування, що дозволяє уникнути витрати енергії та води.

В умовах, коли жодна апаратна або програмна система не є абсолютно захищеною від збоїв, критично важливим є вміння локалізувати помилки без порушення загальної працездатності пристрою. У розробленій системі ця задача вирішується за допомогою концепції ізольованої деградації, що означає збереження повноцінного функціонування всіх інших компонентів у випадку виходу з ладу одного з модулів. Такий підхід базується на принципі незалежної ініціалізації та взаємодії, коли кожен сенсор, виконавчий пристрій або логічний блок має власну процедуру перевірки, а в разі помилки – власний сценарій реакції.

Наприклад, якщо один із сенсорів вологості не відповідає або повертає повторно аномальні значення, система відкидає ці дані з подальшого оброблення та базує свої рішення на показниках з інших модулів або на прогнозних оцінках, сформованих на основі збереженої динаміки вологості. Таким чином, навіть при частковій втраті функціональності система не лише продовжує працювати, але й зберігає розуміння контексту, не втрачаючи орієнтацію в логіці поведінки.

					КВРКІ 210123.21.01.77 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

Особливе значення має реакція пристрою на раптову втрату живлення або перезавантаження. Для цього в мікропрограмі реалізовано механізм періодичного збереження критичних даних у енергонезалежну пам'ять. Йдеться не лише про конфігураційні параметри, які задає користувач, а й про оперативні змінні, що відображають поточний стан пристрою. Ці змінні включають в себе останні зчитані показники, активну фазу дій (наприклад, у процесі поливу), історію помилок, а також тайм-коди останніх подій. Збереження відбувається у визначені моменти, зокрема перед запуском поливу, зміною налаштувань або завершенням сеансу руху.

Після повторного запуску пристрій спочатку не переходить до звичайного режиму, а проводить діагностичне зчитування збережених даних. Якщо виявляється, що система завершила роботу аварійно або не завершила попередній цикл, вона переходить у фазу відновлення. У цій фазі відтворюється передостанній відомий стан, після чого виконується перевірка доцільності продовження дії. Наприклад, якщо полив уже відбувся, але пристрій не встиг зберегти новий рівень вологості, система не виконує полив повторно, а переходить у фазу контролю рівня вологості без додаткових дій, аби уникнути надмірного зволоження.

Завдяки цим заходам пристрій демонструє здатність не лише визначати та ізолювати помилки, а й логічно обґрунтовано відновлювати свою поведінку, що імітує природну безперервність функціонування, притаманну складним інженерним системам. Це критично важливо в контексті довготривалого автономного використання, де втручання людини може бути відкладеним або обмеженим.

Для забезпечення довготривалої та стабільної роботи системи критично важливо не лише реагувати на вже виявлені збої, а й запобігати їхньому виникненню. Саме тому в архітектуру прошивки було закладено механізм постійної самодіагностики, який виконується не лише під час старту пристрою, а й у фоновому режимі протягом усього його функціонування. Це дозволяє вчасно

					КВРКІ 210123.21.01.77 ПЗ	Арк. 52
Зм.	Арк.	№ докум.	Підпис	Дата		

виявляти відхилення в роботі окремих модулів, нестабільність живлення, періодичну втрату зв'язку або поступове погіршення точності зчитуваних даних.

Кожна критична підсистема, зокрема сенсорні входи, система зв'язку, блок живлення та EEPROM, має власну процедуру контролю, яка періодично ініціюється ядром. У разі, якщо відхилення носять постійний характер або виявляється динаміка деградації, система фіксує це як потенційно небезпечний стан і реагує з урахуванням контексту. Якщо йдеться про відхилення, які не заважають безпечному виконанню функцій, система лише генерує попередження. Натомість, якщо порушення ставить під загрозу коректність ухвалення рішень, модуль переводиться в безпечний режим, або ж ініціюється ізоляція.

Важливим джерелом інформації для аналізу працездатності є внутрішній журнал подій, що формується протягом усього періоду роботи пристрою. У цьому журналі фіксуються ключові дії: момент запуску, зміна конфігурації, запуск чи завершення поливу, помилки комунікації з модулями, збої зчитування, перезавантаження, а також будь-які спроби взаємодії з інтерфейсом користувача. Кожен запис містить не лише опис події, а й позначення часу, тип модуля, який її ініціював, а також результат виконаної дії.

Цей підхід дозволяє не лише відслідковувати хід подій у реальному часі, а й здійснювати ретроспективний аналіз поведінки системи у випадку складних помилок. Наприклад, якщо через кілька днів роботи користувач помічає зниження частоти поливу або втрату реакції пристрою на зміни вологості, журнал дозволяє визначити, чи є це наслідком програмної логіки, чи результатом поступового погіршення роботи певного сенсора.

Система також реалізує механізм виявлення логічних аномалій, коли дії пристрою не відповідають внутрішньо погодженому сценарію. Наприклад, якщо рівень вологості не змінюється після поливу, або навпаки, надмірно зростає без виконання поливу, система реєструє такий факт як логічне порушення. У подібних випадках пристрій самостійно ініціює повторну перевірку системи поливу, оцінює справність сенсорів, а також аналізує зовнішні умови, які могли б пояснити

аномалію, наприклад опади або зміну температури, що могли вплинути на зчитування.

Цей рівень логічного аналізу, поєднаний із фізичним моніторингом, дозволяє пристрою діяти не лише як пасивний виконавець команд, а як активний агент, здатний інтерпретувати змінну ситуацію, фіксувати суперечності та реагувати на них у контекстуально обґрунтований спосіб.

Узагальнюючи реалізовані підходи, можна стверджувати, що розроблена система здатна не лише витримувати типові збої, а й ефективно адаптувати свою поведінку в умовах часткової втрати функціональності. Завдяки поєднанню ізольованої архітектури, активної самодіагностики, журналювання подій та алгоритмів адаптивної компенсації вдалося сформувати платформу, яка зберігає керованість навіть у випадку складних відмов. Пристрій не потребує постійного втручання користувача і здатен упродовж тривалого часу працювати самостійно, своєчасно реагуючи на зміни довкілля.

В умовах реального використання, особливо коли йдеться про догляд за рослинами в теплицях, оранжереях або важкодоступних ділянках, критично важливою є можливість довіряти пристрою ухвалення рішень без участі людини. Саме тому система орієнтована не на жорстку логіку, що спрацьовує виключно за конкретного тригера, а на гнучку модель поведінки, яка враховує контекст, попередній досвід і можливість тимчасового відхилення від стандартного сценарію. Це дозволяє уникати типових проблем жорстко заданих алгоритмів, зокрема непередбачених дій у разі некоректного зчитування або збоїв.

Впроваджені механізми помилкової толерантності також суттєво впливають на ресурсне споживання. Наприклад, замість безперервного перезапуску або марного повторення дій пристрій здатен ідентифікувати ситуацію як стабільну і утримуватись від зайвих витрат енергії. Така поведінка є важливою не лише з точки зору енергоефективності, а й з позиції мінімізації фізичного зношення окремих компонентів, зокрема помпи, двигунів та модулів зберігання.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 54
Зм.	Арк.	№ докум.	Підпис	Дата		

Після проходження повного циклу емуляційного тестування можна з впевненістю стверджувати, що пристрій не просто виконує задані функції, а й має достатній рівень автономного контролю над власним станом. Ця властивість перетворює його з простого виконавця на елемент інтелектуального середовища, здатного діяти у змінному оточенні з урахуванням логіки, збереженої історії та прогнозованої поведінки.

Таким чином, запропонована система обробки помилок і відновлення забезпечує не лише безперервність роботи, але й створює основу для подальшого розвитку інтелектуальної автоматизації в догляді за рослинами. Надалі ці механізми можуть бути доповнені більш глибоким аналізом трендів, машинним навчанням або зовнішнім управлінням через хмарні сервіси, що лише підвищить стабільність і надійність пристрою в складних умовах експлуатації..

3.4. Реалізація програмно-технічного засобу для моніторингу та автоматизації догляду за рослинами

Етап безпосередньої реалізації програмно-технічного засобу передбачав поєднання апаратної моделі з інтелектуальним керуванням та візуальним інтерфейсом користувача. Враховуючи складність проекту та вимогу забезпечити стабільну і контрольовану роботу ще до збирання фізичної моделі, початкові дії були зосереджені в середовищі емуляції. Це дозволило не лише протестувати логіку та взаємодію між модулями, а й передбачити потенційні помилки в роботі сенсорних систем, алгоритмів ухвалення рішень і взаємодії з хмарним середовищем.

У виборі середовища моделювання перевага була надана емулятору, який підтримує апаратні платформи на базі мікроконтролера ESP32, що було обрано як ядро розробки завдяки його багатofункціональності, підтримці бездротового зв'язку та можливості одночасної роботи з декількома аналоговими і цифровими входами. Емуляція дозволила імітувати реакцію пристрою на зміну зовнішніх

параметрів, таких як вологість ґрунту, рівень освітлення чи температура повітря, а також протестувати програмні реакції системи на ці зміни без ризику пошкодження реального обладнання.

Програмна логіка була написана на мові C++ з використанням бібліотек для ESP32, які надають високорівневі інтерфейси для роботи з вбудованими датчиками, Wi-Fi, керування GPIO-портами та реалізації вебсервера. Для вирішення цього завдання було обрано архітектурну модель, яка використовує механізми неблокуючих таймерів, дозволяючи запускати періодичні функції у задані інтервали без переривання основного потоку виконання.

Для організації живлення та підключення всіх сенсорів і виконавчих механізмів використано стабілізатор, що формує 3,3 В із акумулятора Li-ion, підключеного до модуля TP4056. До шини I²C через резистори та відповідні конектори під'єднані чотири датчики освітленості BH1750, модуль BME280, а також ультразвуковий HC-SR04 та датчики перешкод IR.

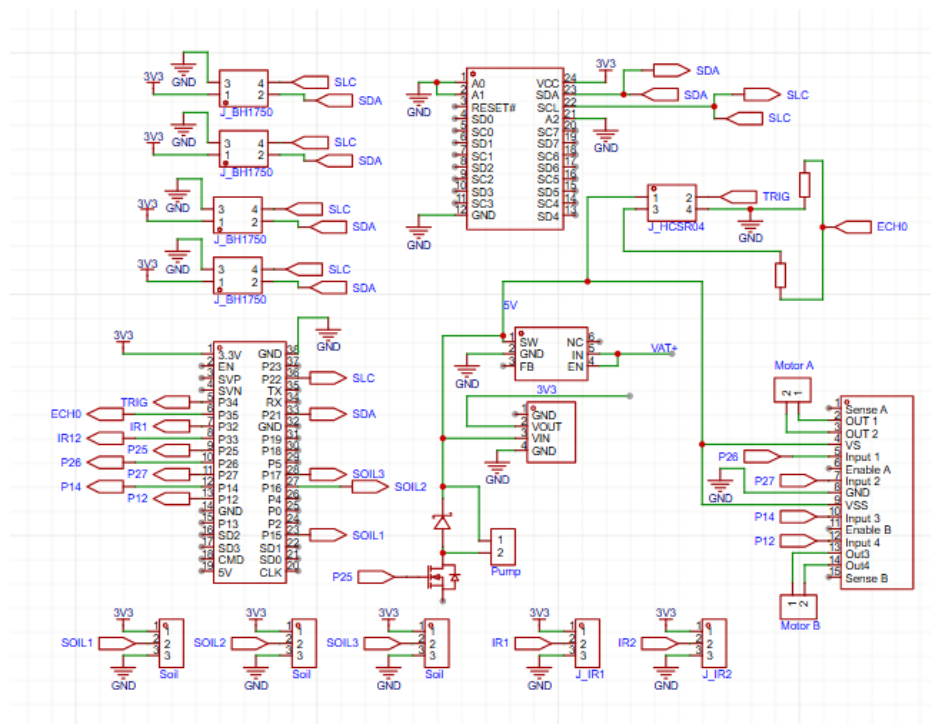


Рисунок 3.3 – Принципова електрична схема підключення ESP32, сенсорів і виконавчих механізмів

На першому етапі було реалізовано модуль збору даних, який регулярно опитував симульовані значення з віртуальних сенсорів. Дані зчитувались кожні 10–30 секунд, залежно від типу датчика, і передавались до внутрішнього обробника, який виконував нормалізацію показників, усереднення при коливаннях та первинну фільтрацію аномальних значень. Паралельно з цим функціонував модуль обробки подій, у завдання якого входила оцінка отриманої інформації та визначення потреби в активних діях. Наприклад, якщо середнє значення вологості ґрунту опускалось нижче заданого порогу, активувався алгоритм підготовки до поливу.

Особливої уваги на етапі реалізації вимагали ті підсистеми, які відповідали за фізичну взаємодію із середовищем. У контексті віртуального тестування постала задача симуляції роботи виконавчих елементів, таких як насосна система, система пересування та модуль освітлення. Для цього було створено абстрактні представлення цих пристроїв у вигляді об'єктів, які могли приймати та обробляти команди від основного контролера і повертати у відповідь змодельовану реакцію.

Найбільше програмних зусиль було спрямовано на алгоритм поливу, який повинен був діяти логічно, обережно і при цьому адаптивно до умов. Після надходження сигналу про потребу в поливі, система ініціювала перевірку наявності води в резервуарі, актуальності показників вологості з кількох датчиків та можливості безпечного виконання команди. Якщо всі умови виконувалися, модуль активував насосну систему, яка протягом певного інтервалу часу здійснювала подачу води до кореневої системи рослини. У процесі поливу система продовжувала вимірювати вологість і припиняла дію в момент досягнення цільового значення. Такий підхід дозволив уникати перезволоження та зменшити витрати води.

Паралельно було реалізовано механізм переміщення пристрою, який ґрунтувався на емуляції дій двигунів на шасі. Основна задача цього модуля полягала в тому, щоби при фіксації недостатнього освітлення рослина могла переміститися до більш яскравої ділянки простору. Для реалізації цієї логіки було

					КВРКІ 210123.21.01.77 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

створено віртуальну карту з умовною градацією освітленості, яку пристрій «сканував» на основі даних з чотирьох фотосенсорів, розташованих по периметру. Отримана інформація аналізувалась у програмі, і якщо один із напрямів стабільно демонстрував вищу освітленість, активувався сценарій зміщення платформи у відповідний бік.

Щоб не допустити зіткнення з перешкодами під час такого пересування, додатково моделювалась поведінка ультразвукових сенсорів, які могли фіксувати наявність об'єктів перед пристроєм. У разі, якщо система визначала потенційну небезпеку, вона зупиняла рух, коригувала маршрут або залишалась у поточному положенні, зберігаючи останні дані для подальшого аналізу. Такий підхід дозволив створити поведінкову модель, близьку до дій реального автономного пристрою, яка враховує як інженерну логіку, так і зовнішні обмеження.

У середовищі моделювання ці сценарії виконувались за допомогою симульованого зворотного зв'язку, коли система не лише видавала команди, а й отримувала відповіді про стан виконавчого елемента, зміну параметрів середовища, витрату енергії та результати впливу на рослину. Це дозволило виявити та усунути логічні суперечності, які могли б виникнути в реальних умовах, наприклад запуск поливу при одночасному русі або неправильне визначення джерела світла через накладання перешкод.

Окрему частину реалізації становило створення зручного інтерфейсу для моніторингу параметрів та керування системою у віддаленому режимі. З огляду на обрану платформу ESP32, яка підтримує бездротовий зв'язок, було вирішено впровадити механізм Wi-Fi-з'єднання з мобільним застосунком, через який користувач міг би не лише переглядати поточні дані, але й задавати цільові параметри, контролювати дії пристрою та отримувати сповіщення у випадку помилок.

На першому етапі було створено базовий вебінтерфейс, який відображав у реальному часі значення основних сенсорів: рівень вологості ґрунту, температуру повітря, освітленість, заряд батареї та статус зв'язку. Цей інтерфейс був

реалізований як вбудований вебсервер на самому пристрої, доступ до якого надавався в межах локальної мережі. Для цього використовувались бібліотеки ESPAsyncWebServer та WiFiManager, які забезпечили зручне налаштування з'єднання та обробку HTTP-запитів без блокування основного потоку. Після підключення до мережі пристрій генерував власний ідентифікатор і зберігав конфігурацію, що дозволяло уникнути повторного введення параметрів під час наступного запуску.

Паралельно з локальним вебінтерфейсом була протестована взаємодія з мобільним застосунком, створеним на платформі Blynk. Цей вибір обумовлювався простотою реалізації, широкою підтримкою пристроїв та наявністю вбудованих віджетів, які дозволяють зручно візуалізувати дані й організувати керування в режимі реального часу. За допомогою Blynk користувач мав можливість контролювати стан поливу, встановлювати бажаний рівень вологості, переглядати журнал подій і отримувати миттєві повідомлення про збої або досягнення критичних порогів.

Особливу увагу на цьому етапі було приділено безпечному обміну даними. Хоча початково застосовувалось незахищене з'єднання, у подальшому було впроваджено базову систему автентифікації на рівні обміну ключами, а також реалізовано обмеження доступу до керування пристроєм поза межами локальної мережі. Це дозволило запобігти випадковому або шкідливому втручанню у роботу системи та зберегти конфіденційність інформації, що передається.

У процесі тестування виявилось, що обмін даними з мобільним додатком потребує оптимізації. Було виявлено затримки у відображенні даних при високій частоті оновлення, що викликало необхідність впровадити буферизацію та зменшити частоту надсилання даних до однієї-двох секунд. Крім того, частину інформації, яка не є критичною для миттєвого реагування, було винесено у фоновий режим, аби не перевантажувати канал зв'язку та забезпечити стабільність системи при тривалому використанні.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 59
Зм.	Арк.	№ докум.	Підпис	Дата		

Водночас проводилось тестування можливостей розширення архітектури в бік взаємодії з хмарними сховищами. Була змодельована передача логів у форматі JSON до зовнішнього серверу через REST API. Це дозволило переконатись у здатності пристрою функціонувати як частина більших систем аграрного моніторингу або розумного дому, де централізоване зберігання даних є обов'язковим для аналізу в динаміці.

Остаточний етап реалізації полягав у всебічному тестуванні стабільності системи, і саме він став вирішальним у визначенні надійності пристрою в умовах, наближених до реального використання. Завдяки середовищу емуляції вдалося створити сценарії, які було б важко або навіть небезпечно відтворити в реальному середовищі. Це включало перебої з живленням, зникнення сигналу Wi-Fi, відмову окремих сенсорів, а також введення завідомо некоректних або суперечливих вхідних даних.

Особливу увагу було приділено реакції системи на раптову втрату живлення. Відповідно до закладеної логіки, при наступному запуску пристрій не переходив одразу в активний режим, а спершу проводив перевірку збережених у пам'яті параметрів, і лише переконавшись у відсутності критичних відхилень, відновлював цикл роботи. У процесі тестування підтвердилась здатність пристрою не лише зберігати дані останніх дій, а й логічно відновлювати сценарій поведінки. Це дозволило уникати повторного поливу після аварії або запуску моторів, коли попередній цикл не було завершено.

Окремо були змодельовані ситуації, за яких сенсори починали передавати аномальні значення, наприклад, постійно фіксували вологість у діапазоні за межами допустимого. У таких випадках пристрій коректно виключав ці дані з обробки, орієнтуючись на значення з альтернативних джерел або фіксуючи помилку з відповідним сповіщенням. У випадку відмови кількох сенсорів одночасно пристрій переходив у режим очікування, зберігаючи керованість і надаючи користувачеві всю необхідну інформацію для прийняття рішень.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

Паралельно було проведено оцінку ефективності алгоритмів прийняття рішень, зокрема щодо зволоження ґрунту і переміщення пристрою до джерел світла. Результати тестування підтвердили, що навіть у змодельованих умовах з частими змінами освітлення, зміною температури або вмісту вологи пристрій демонстрував адаптивну та послідовну поведінку. Він уникав надмірних або несвоєчасних дій, що є показником логічної узгодженості реалізованого ПЗ.

Кінцевим підсумком тестування стало переконання у практичній готовності пристрою до фізичної реалізації, з можливістю розгортання на справжніх компонентах без потреби суттєвої переробки коду чи алгоритмів. Завдяки моделюванню вдалося не лише перевірити логіку, а й сформувавши архітектурну основу, що враховує критичні фактори: від стабільності зв'язку до безпеки водоподачі, від реакції на перешкоди до стійкості до помилок з боку користувача.

Таким чином, реалізація програмно-технічного засобу завершилась формуванням повноцінної платформи, яка поєднує автономну логіку ухвалення рішень, активну взаємодію з користувачем та здатність до адаптації в умовах зміни середовища. Це створює міцну основу для подальшого фізичного виготовлення пристрою, розширення функціоналу, впровадження хмарного аналізу та інтеграції в системи розумного дому чи фермерського господарства.

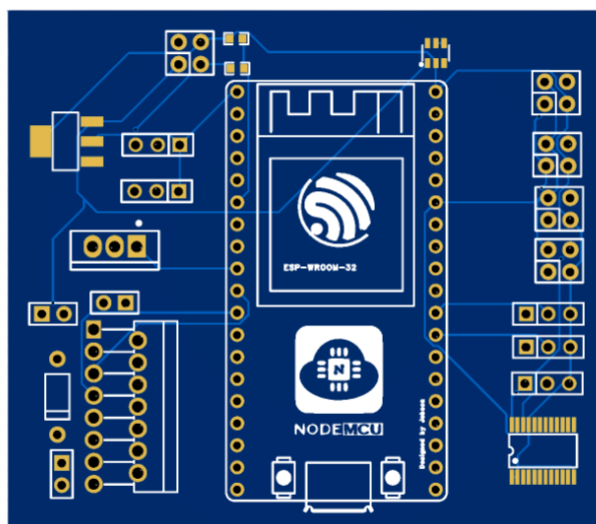


Рисунок 3.4 – 2D-топология друкованої плати з розташуванням ESP32, стабілізатора, TP4056, конекторів для сенсорів і моторного драйвера

3.5. Висновки

У межах третього розділу дипломної роботи було послідовно розкрито основні елементи, що формують ядро запропонованого програмно-технічного засобу для моніторингу та автоматизації догляду за рослинами. Було визначено принцип функціонування системи, розроблено концепцію архітектури з урахуванням усіх необхідних модулів і внутрішніх взаємозв'язків між компонентами, а також описано процес реалізації основних підсистем, від сенсорної логіки до інтерфейсу користувача та механізмів обробки помилок. Окрему увагу приділено реалізації в емуляторі, що дозволило протестувати поведінку пристрою без використання фізичних компонентів, а також виявити потенційні вузькі місця в логіці взаємодії або обробки нестандартних ситуацій.

Систему було розроблено з акцентом на автономність, адаптивність і збереження керованості в умовах динамічного середовища. Реалізовані алгоритми демонструють стійкість до збоїв, здатність до самовідновлення та логічне поводження у випадку критичних відхилень. Платформа зберігає баланс між спрощеною побудовою, доступною для малобюджетного виробництва, та високим рівнем функціональності, що відкриває можливості для подальшого масштабування проекту.

Загалом проведена робота підтверджує життєздатність обраного технічного рішення та демонструє його ефективність для задач догляду за рослинами в автономному або напівавтономному режимі. Створена система є надійною основою як для побутового використання, так і для потенційної інтеграції в ширші аграрні або наукові проекти.

					КВРКІ 210123.21.01.77 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень було запропоновано програмно-технічний засіб для моніторингу та автоматизації догляду за рослинами, що об'єднує інтелектуальне керування, сенсорну систему та здатність до адаптивної реакції в умовах змінного середовища.

У першому розділі було проведено комплексний аналіз існуючих комерційних рішень у сфері автоматизованого догляду за рослинами, а також загальних підходів до побудови подібних пристроїв. Розглянуто переваги й обмеження провідних моделей, визначено недоліки в аспектах мобільності, автономності та гнучкості керування. Отримані результати лягли в основу постановки вимог до системи, яку реалізовано в межах цієї дипломної роботи.

У другому розділі було обґрунтовано вибір апаратного середовища реалізації пристрою та сформульовано функціональні і нефункціональні вимоги до його роботи. Визначено склад модулів, типи сенсорів, виконавчих механізмів та способів взаємодії з користувачем. Також розглянуто способи обміну даними, режими енергозбереження й особливості обробки інформації на борту системи.

У третьому розділі описано принципи роботи програмно-технічного засобу, алгоритми прийняття рішень, логіку обробки сенсорних сигналів, а також реалізовано прототип пристрою в середовищі моделювання. Було здійснено тестування всіх ключових сценаріїв: поливу, переміщення до джерела світла, обробки збоїв та взаємодії з мобільним застосунком. Оцінено стабільність і надійність системи за результатами емуляції.

Таким чином, у межах дипломної роботи було створено повноцінну концепцію автономного пристрою для догляду за рослинами, яка поєднує в собі програмну адаптивність, гнучку логіку обробки ситуацій та зручне інтерфейсне керування. Отримані результати підтверджують життєздатність і прикладну цінність запропонованого рішення.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Ivy Smart Planter. URL: https://smartplantivy.com/?srsltid=AfmBOor3u15e8s3FbVqPkmTOp-qejTNBQLO_7HSS_xAMrat4hgY61oPXF (дата звернення: 17.03.2024)
2. LetPot MP1. URL: <https://letpot.com/products/self-watering-planter-mp1> (дата звернення: 17.03.2024)
3. LeafyPod – загальний вигляд. URL: <https://www.leafypod.one/> (дата звернення: 17.03.2024)
4. Click and Grow. URL: https://www.clickandgrow.com/?srsltid=AfmBOoo8NT4HGFTJ9jmY5xsls2pvEKfBT9IQDIvZ1oqeZqNYeuw1oQc_ (дата звернення: 18.03.2024)
5. Parrot Flower Power. URL: <https://www.journaldulapin.com/2020/04/13/la-fin-du-flower-power/> (дата звернення: 18.03.2024)
6. Wang Y., Kim J. Integration of BH1750 Light Sensor with ESP32 for Plant Monitoring. *Journal of Sensors*. 2021. Vol. 2021. P. 1236.
7. Tang Y., Liu X. Recent Advances in Piezoelectric Micropumps: A Review. *Sensors*. 2022. Vol. 22, No. 5. P. 1534..
8. Sebelefsky F. BME280 Environmental Sensor: Performance in Precision Agriculture. *IEEE Sensors Journal*. 2021. Vol. 21, No. 14. P. 15980–15989.
9. Chen L., Xu M. Comparative Analysis of L298N and TB6612FNG Motor Driver ICs for IoT Robotics. *International Journal of Robotics and Automation*. 2023. Vol. 37, No. 2. P. 145–152.
10. Martinez J. Evaluation of HC-SR04 Ultrasonic Sensor for Home Robotics Navigation. *Journal of Robotics Research*. 2021. Vol. 8, No. 1. P. 10–18.
11. Nguyen T. Characteristics of 18650 Lithium-ion Batteries for Portable Electronics. *Journal of Power Sources*. 2024. Vol. 500. P. 230075.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

12. Garcia P. Energy Efficiency Techniques for ESP32 in IoT Applications. *IEEE Access*. 2022. Vol. 10. P. 34567–34578.
13. Модуль TP4056. URL: <https://grobotronics.com/lithium-battery-charger-and-protection-module-tp4056.html?sl=en> (дата звернення: 12.04.2024)
14. Smith A. Effect of LED Lighting on Indoor Plant Growth: A Meta-analysis. *Journal of Horticultural Science*. 2023. Vol. 46, No. 2. P. 120–130.
15. Kumar R. Role of Push Notifications in IoT Monitoring Systems. *International Journal of Internet of Things*. 2022. Vol. 4, No. 3. P. 55–66.
16. Zhang W. Importance of Regular Calibration in Home Sensor Systems. *IEEE Sensors Letters*. 2021. Vol. 5, No. 4. P. 1–4.
17. Lin K. Over-the-Air Firmware Updates for ESP32: A Survey: *Proceedings of Embedded Systems Workshop*. 2023. P. 23–29.
18. Patel S. Real-Time Operating Systems in IoT Edge Devices. *Journal of Real-Time Systems*. 2024. Vol. 59, No. 1. P. 1–20.
19. Fernandez L. Security of IoT Devices in Local Networks. *Computer Networks*. 2022. Vol. 201. P. 108638.
20. Roy D. Self-Diagnostic Mechanisms in Smart IoT Devices. *Sensors*. 2021. Vol. 21, No. 11. P. 3890.
21. Espressif Systems. ESP32 Arduino Core: API Reference. Espressif Documentation. 2025. URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/> (дата звернення: 08.05.2025).
22. Lopez G. Implementing Web Servers on ESP32 Microcontrollers. *IET Computers & Digital Techniques*. 2023. Vol. 15, No. 4. P. 200–208.
23. Anderson E. Embedded Web Interfaces in IoT Devices: Benefits and Challenges. *International Journal of Embedded Research*. 2024. Vol. 11, No. 2. P. 90–100.
24. Tsai Y. Methodology for Testing Microcontrollers in Real-World Environments: *Proceedings of the 2022 IEEE International Test Conference*. 2022. C. 1–10.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 65
Зм.	Арк.	№ докум.	Підпис	Дата		

25. Babu B. P., Meghana D. IoT-Enabled Plant Management: A Solution for Busy Lifestyles: *International Conference on Data Science, Machine Learning and Applications*. Springer Nature Singapore, 2023. C. 1084–1092.

26. Guerrero-Ulloa G., Méndez-García A., Torres-Lindao V., Zamora-Mecías V., Rodríguez-Domínguez C., Hornos M. J. Internet of Things (IoT)-based Indoor Plant Care System. *Journal of Ambient Intelligence and Smart Environments*, T. 15, № 1, 2023. C. 47–62.

27. Abdalla S. Y., Yogaraju H., Rai S. Design and Development of an IoT Based Plant Growth Monitoring and Management System for Indoor Farming. *Malaysian Journal of Science and Advanced Technology*, 2021. C. 77–80

28. Kirci P., Ozturk E., Celik Y. A Novel Approach for Monitoring of Smart Greenhouse and Flowerpot Parameters and Detection of Plant Growth with Sensors. *Agriculture*, T. 12, № 10, 2022. C. 1705.

29. Gajula A. K., Maringanti V. P. Complete Plant Care Monitoring Using Internet of Things (IoT): *2024 4th International Conference on Sustainable Expert Systems (ICSSES)*. IEEE, 2024. C. 188–196.

30. Bhattacharya S., Bagmar A. S. A Comprehensive Plant Health Monitoring System with IoT and Deep Learning: *2023 International Conference on Next Generation Electronics (NEleX)*. IEEE, 2023. C. 1–4

31. Ghosh A., Rahaman S. H., Biswas S., Mandal K. Automation in the Sector of Home Gardening (Domestic Smart Pot): *2022 1st International Conference on Computational Science and Technology (ICCST)*. IEEE, 2022. C. 1–9.

32. Kori A. A., Veena K. N., Basarkod P. I., Harsha R. Hydroponics System based on IoT: *Annals of the Romanian Society for Cell Biology*, T. 25, № 4, 2021. C. 9683–9688.

33. Musyaffa, N., Rifai, B., Sastra, R., Yuniarto, E. Smart Plant Monitoring System Kelembaban Tanah Menggunakan Metode Fuzzy Logic Pada Tumbuhan Cabai Berbasis IoT: *Jurnal Khatulistiwa Informatika*, T. 11, № 1, 2023. C. 35–42.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 66
Зм.	Арк.	№ докум.	Підпис	Дата		

34. Nandyal S., Siddiqua K. Automated Home Garden System Using Robotics: *2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*. IEEE, 2022. C. 1–5.

35. Patil T., Mallapur J. D., Mallapur D. G. Self Monitored Indoor Plant Maintenance with Wastewater: *2023 International Conference on Network, Multimedia and Information Technology (NMITCON)*. IEEE, 2023. C. 1–6.

36. Rahman A., Raheem R., Sharmilan T. Design a Smart Mini Robot for Indoor Plant Watering System: *International Journal of Science: Basic and Applied Research (Int. J. Sci. Basic Appl. Res.)*, T. 61, № 1, 2022. C. 1–9.

37. Sakthi K., Zain Y. M. S., Raj S. M. S., Manickavasakar M. IoT based Soil Monitoring and Control Systems: *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, 2023. C. 516–522.

38. Selvaraj K., Iswarya M., Ramyasri S., Keerthika A. Arduino based Smart Irrigation System for Home Gardening: *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2021. C. 1284–1288.

39. Senthilkumar S., Danasekar M. IoT-Based Smart Gardening System Using the Cloud: Advancing Smarter and More Secure Industrial Applications Using AI, IoT, and Blockchain Technology. *IGI Global Scientific Publishing*, 2022. C. 261–277.

40. Wiyono A. T. Design of Smart Garden Based On The Internet of Things (IoT). *BEST: Journal of Applied Electrical, Science, & Technology*. 2021; 3(2): 36–40.

41. Zuniga A., Motlagh N. H., Flores H., Nurmi P. Smart Plants: Low-Cost Solution for Monitoring Indoor Environments. *IEEE Internet of Things Journal*. 2022; 9(22): 23252–23259.

42. Omasa K., Ono E., Ishigami Y., Shimizu Y., Araki Y. Plant Functional Remote Sensing and Smart Farming Applications. *International Journal of Agricultural and Biological Engineering*. 2022; 15(4): 1–6.

43. Ku H. H., Liu C. H., Wang W. C. Design of an Artificial Intelligence of Things based Indoor Planting Model for Mentha Spicata. *Processes*. 2022; 10(1): 116.

					КВРКІ 210123.21.01.77 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

44. Abhishek V., Akash R., Sudha P. N. Automatic Plant Watering System using Arduino. *International Research Journal of Engineering and Technology (IRJET)*. 2021; 8(3): 2349 – 6002.

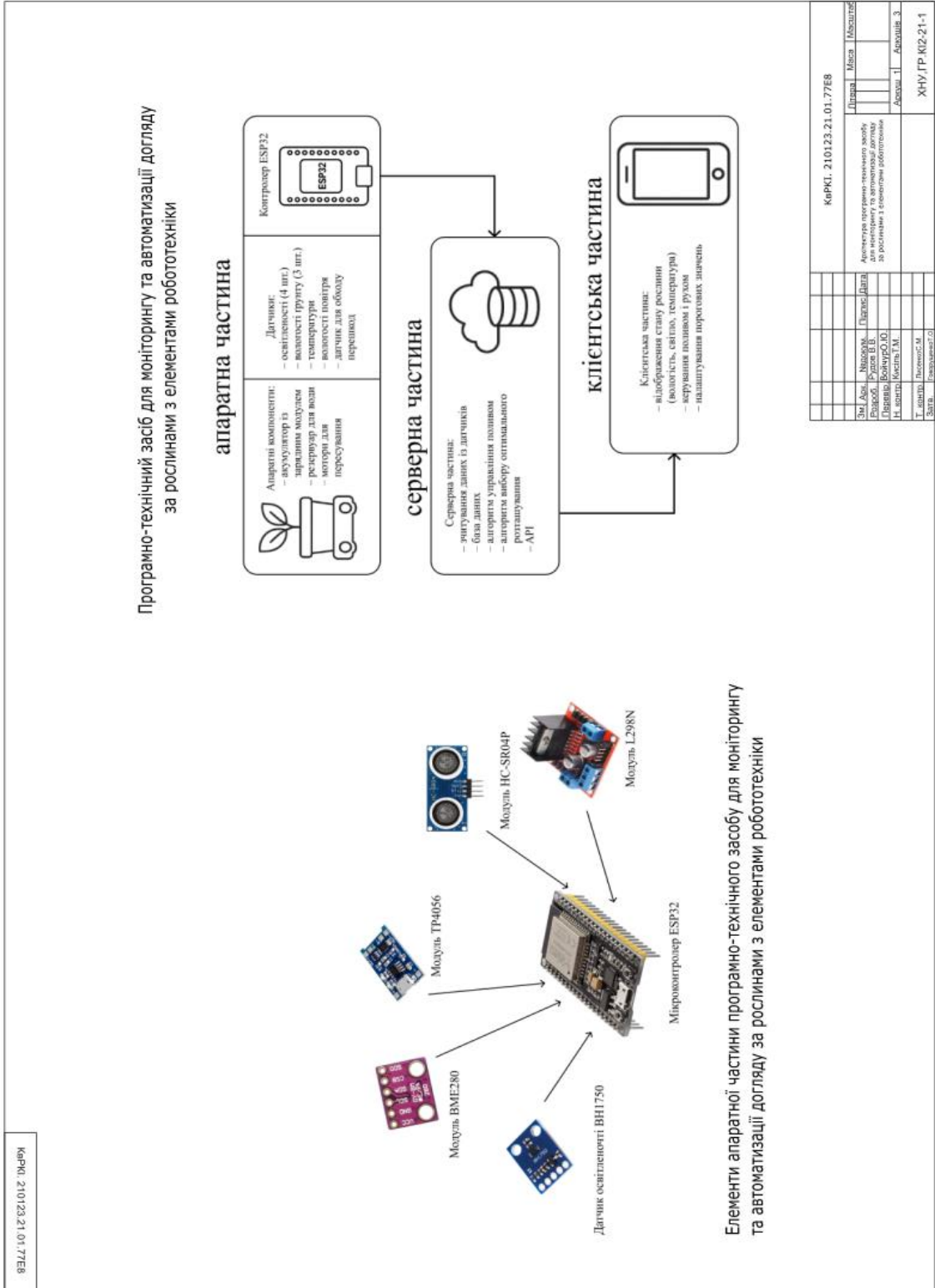
45. Yu L., Gao W., Shamshiri R. R., Tao S., Ren Y., Zhang Y., Su G. Review of Research Progress on Soil Moisture Sensor Technology. *International Journal of Agricultural and Biological Engineering*. 2021; 14(4): 32–42.

46. Zuniga A., Motlagh N. H., Flores H., Nurmi P. Smart Plants: Low-Cost Solution for Monitoring Indoor Environments. *IEEE Internet of Things Journal*. 2022; 9(22): 23252–23259.

					КВРКІ 210123.21.01.77 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

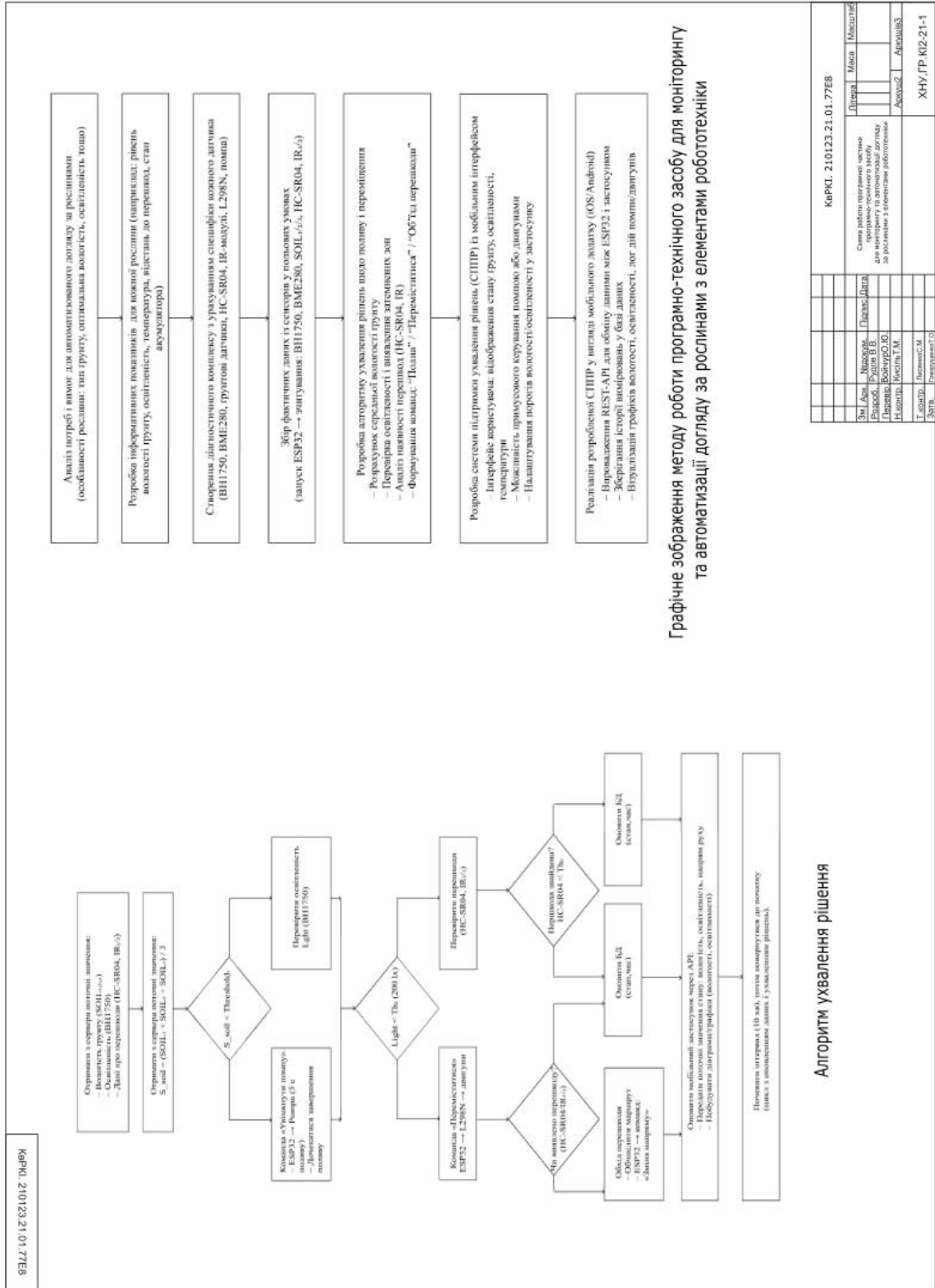
Додаток А (обов'язковий)

КОПІЯ КРЕСЛЕННЯ «АРХІТЕКТУРА ПРОЄКТУ»



Додаток Б (обов'язковий)

КОПІЯ КРЕСЛЕННЯ «СХЕМА РОБОТИ ПРОГРАМНОЇ ЧАСТИНИ ПРОЄКТУ»



Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Володимир Рудов

Співавтор:

Назва: Рудов_Програмно-технічний засіб для моніторингу та автоматизації догляду за рослинами з елементами робототехніки

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:2.1%

Коефіцієнт подібності 2:0.8%

Мікропробіли: 6

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-06 03:21:50.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укрити плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-06-06

Дата



Доцент Андрій Ніченорук

експерт

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 7%

ID: 243748 Title: БКР Програмно-технічний засіб для моніторингу та автоматизації догляду за рослинами з елементами робототехніки Added in a DB: 2025-06-05 Authors: Володимир Рудов Heads: Олег ВОЙЧУР Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	104382	691	1303 (1%)	16 (2%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Рудов Володимир Віталійович

Тема: Програмно-технічний засіб для моніторингу та автоматизації догляду за рослинами з елементами робототехніки

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 65

1. Короткий зміст роботи та прийнятих рішень: перший розділ містить огляд комерційних рішень (Ivy Smart Planter, LetPot MP1, LeafyPod, Click & Grow) і мобільних застосунків (Plant Parent, Vera) з обґрунтуванням вибору датчиків BH1750, сенсорів вологості та температури і платформи ESP32. У другому розділі сформульовано функціональні (моніторинг вологості, освітленості та температури; віддалене керування поливом і рухом) і нефункціональні (автономність, надійність, швидкість реакції) вимоги, описано вибір колісного шасі, ємності для води, розташування сенсорів та програмну архітектуру на MicroPython з передачею даних через HTTP до хмарного сервісу. Третій розділ присвячено створенню віртуального стенда в емуляторі, реалізації модулів керування насосом, алгоритму ухилення від перешкод та обробці даних із сенсорів, а також тестуванню у змінених умовах вологісних і світлових параметрів

2. Висновок про відповідність роботи дипломному завданню: робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: у першому розділі проведено ґрунтовний аналіз сучасних розумних горщиків і мобільних застосунків, розглянуто принципи роботи Ivy Smart Planter, LetPot MP1, LeafyPod і Click & Grow, їхні переваги та обмеження. Другий розділ містить чіткий опис функціональних і нефункціональних вимог, обґрунтовано вибір ESP32 і MicroPython для швидкої прототипізації, представлено конструкцію шасі, розташування датчиків і вибір ємності для води.

Третій розділ описує принципи роботи програмно-апаратного комплексу в емуляторі (Proteus), розробку модулів для зчитування сенсорів, обробки сигналів і керування насосом та моторними приводами, а також підключення ESP32 до хмарної платформи (HTTP-запити до Google Apps Script) і створення прототипу мобільного застосунку з використанням Blynk. Наведено результати тестування в сценаріях «низька вологість ґрунту» та «недостатнє освітлення» і перевірку алгоритмів ухилення від перешкод з ультразвуковим датчиком

4. Позитивні сторони роботи: пристрій має високу практичну значущість, поєднуючи моніторинг, автоматичний полив і мобільність; обґрунтований вибір ESP32 та сенсорів, порівняння з комерційними аналогами і комплексний підхід до розробки апаратної та програмної частини з мобільним інтерфейсом демонструють ґрунтовну підготовку автора.

5. Негативні сторони роботи: відсутні реальні експериментальні випробування прототипу (усі перевірки були в емуляторі без фізичних вимірів енергоспоживання), а кількість сценаріїв для оцінки алгоритмів ухилення від перешкод обмежена базовими тестами, що може призвести до неефективної роботи в різних домашніх умовах..

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Корнученко Артем Вікторович
доцент кафедри ЕРЗ

«9» 06 2025 р.

(підпис)

Завідувачу кафедри КПС
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Володимира РУДОВА

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

5 червня 2025 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-технічний засіб для моніторингу та автоматизації догляду за рослинами з елементами робототехніки

Автор: Володимир РУДОВ

Спеціальність: 123– Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Олег ВОЙЧУР

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданій поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданій поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) більшість запозичень використано в розділах огляду існуючих рішень та літератури, які не містять безпосередньо результатів власного експерименту чи моделювання;
- 2) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
- 3) виявлені збіги стосуються загальноновживаних формулювань і термінології (система фіксує відповідність з десятками й навіть кількома десятками джерел для одного речення), що не може розглядатися як плагіат речення;
- 4) відмінності в тексті, зафіксовані системою як «модифікації», обумовлені поєднанням латинських символів і українських абрєвіатур у формулах, що є нормативним набором позначень, а не умисним переписуванням чужого тексту

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 2.13% і адресується до 26 першоджерел; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

Олег ВОЙЧУР

Андрій Нічепорук

Ольга ПАВЛОВА