

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Навчально-науковий
інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

здобувачки Крючкової Євеліни Павлівни
(ПІБ)

академічної групи 123-21-1
(шифр)

спеціальності 123 Комп'ютерна інженерія
(код і назва спеціальності)

за освітньо-професійною програмою 1 2 3 Комп'ютерна інженерія
(офіційна назва)

на тему "ІоТ-система інтерактивного симулятора фотосинтезу для вивчення природничих наук у школі Монтесорі"
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц.Ткаченко С.М.			
загального розділу	доц.Ткаченко С.М.			
спеціального розділу	доц. Ткаченко С.М			
Рецензент				
Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро
2025

ЗАТВЕРДЖЕНО:

завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії
(повна назва)

Гнатушенко В.В.

(підпис)

(прізвище, ініціали)

"25" січня 2025 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавр

здобувачки Крючкової Е.П. академічної групи 123-21-1
(прізвище та ініціали) (шифр)

спеціальності 123 Комп'ютерна інженерія

за освітньо-професійною програмою Комп'ютерна інженерія
(офіційна назва)

на тему "IoT-система інтерактивного симулятора фотосинтезу для вивчення природничих наук у школі Монтесорі"

затверджену наказом ректора НТУ «Дніпровська політехніка» від №

Розділ	Зміст	Термін виконання
Стан питання і постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел показати актуальність завдання, сформулювати мету та задачі виконання кваліфікаційної роботи	15.03.2025
Спеціальний розділ. Технічні вимоги	Сформулювати найменування й призначення комп'ютерної системи, висунути технічні вимоги до неї	15.03.2025
Спеціальний розділ. Розробка апаратної частини	Виконати технічне проектування апаратної частини IoT-системи з необхідними розрахунками	20.04.2025
Спеціальний розділ. Програмне забезпечення системи	Обґрунтувати технічні характеристики програми й розробити програмне забезпечення для симулятора.	31.05.2025

Завдання видано _____
(підпис керівника)

доц. Ткаченко С.М.
(прізвище, ініціали)

Дата видачі 25.01.2025

Дата подання до екзаменаційної комісії 10.06.2025

Прийнято до виконання _____

Крючкова Е.П.

Реферат

Пояснювальна записка: 83 с., 11 рис., 16 табл., 2 дод., 14 джерел.
ВІЗУАЛІЗАЦІЯ, МОНТЕССОРІ, СЕНСОР, СИМУЛЯТОР, СИСТЕМА,
ФОТОСИНТЕЗ

Об'єкт професійної діяльності – комп'ютерно-інженерна система моделювання природничих процесів.

Мета роботи – створення інтерактивного симулятора фотосинтезу для вивчення природничих наук у школі Монтессорі.

Здійснено розробку апаратно-програмної системи з використанням одноплатного комп'ютера Raspberry Pi, сенсорів температури, вологості та освітленості, світлодіодної індикації та веб-інтерфейсу на Flask. Система дозволяє імітувати процес фотосинтезу шляхом зміни умов середовища та візуалізувати результати в реальному часі. Дані експериментів зберігаються у вигляді таблиць CSV та можуть автоматично експортуватись у Google Таблиці.

Інтерактивна модель дає змогу учням досліджувати природні явища через експеримент, формує практичні навички взаємодії з цифровими технологіями, розвиває критичне мислення.

Система підтримує оновлення коду, зміну набору функцій та масштабування. Вона може бути використана як в освітньому процесі, так і для демонстрацій у рамках STEM-проєктів або інтерактивних наукових виставок.

Розробку доцільно використовувати як лабораторне обладнання під час вивчення фотосинтезу, кліматичних факторів та принципів IoT. Подальший розвиток проєкту можливий у напрямку автоматичного керування умовами середовища, доповнення модулем мобільного доступу, використання камер або AI-аналітики.

ЗМІСТ

Вступ.....	5
1 Стан питання та постановка завдання.....	7
1.1 Характеристика та умови роботи об’єкта впровадження.....	8
1.2 Огляд існуючих аналогів.....	27
1.3 Обґрунтування вибраного напрямку вирішення задачі.....	32
1.4 Мета та задачі роботи.....	38
2 Спеціальний розділ. Розробка системи.....	40
2.1 Найменування та призначення об’єкту.....	40
2.1.1 Вимоги до структури та функціонування.....	40
2.1.2 Показники призначення.....	48
2.2 Розробка апаратної частини.....	49
2.3 Розробка програмного забезпечення.....	54
2.3.1 Обґрунтування технічних характеристик.....	54
2.3.2 Опис розробленої програми.....	56
Висновок.....	61
Перелік посилань.....	62
Додаток А. Текст програми налаштування Raspberry Pi та сенсорів.....	63
Додаток Б. Приклади результатів експериментів.....	81

Вступ

У сучасній системі освіти зростає потреба в інтерактивних засобах навчання, які сприяють формуванню глибоких знань через дослідницький підхід, практичну діяльність та розвиток критичного мислення. Особливої актуальності це набуває в закладах, що працюють за методикою Марії Монтесорі, де навчальний процес орієнтований на самостійне пізнання учнями світу через експериментування, спостереження та активну участь.

В цьому контексті інтеграція інноваційних технологій, зокрема IoT-рішень, відкриває нові можливості для вивчення природничих процесів, зокрема фотосинтезу.

Метою даної кваліфікаційної роботи є створення IoT-системи інтерактивного симулятора фотосинтезу для використання в навчальному процесі шкіл, що працюють за Монтесорі-педагогікою.

Завдання роботи полягають у:

- аналізі природних чинників, що впливають на процес фотосинтезу;
- визначенні необхідних сенсорів та електронних компонентів для збору та візуалізації даних;
- розробці апаратної частини симулятора на базі мікрокомп'ютера Raspberry Pi;
- створенні програмного забезпечення з веб-інтерфейсом і можливістю логування даних;
- дослідженні потенціалу інтеграції пристрою з хмарними сервісами;
- апробації функціональності пристрою в умовах освітнього середовища.

Актуальність теми зумовлена потребою у поєднанні інженерних технологій із природничими знаннями в навчальному процесі, що відповідає сучасним вимогам STEM-освіти. Симулятор дає змогу учням змінювати параметри навколишнього середовища — освітленість, температуру, вологість —

та спостерігати, як ці фактори впливають на фотосинтез. Це дозволяє формувати причинно-наслідкові зв'язки без ризику для живих об'єктів чи потреби у складному лабораторному обладнанні.

Вихідними даними для виконання роботи стали результати практик в освітніх закладах з Монтессорі-підходом, наукові статті з тематики STEM-освіти та біологічного моделювання, тези конференцій, а також технічна документація на сенсори, мікроконтролери й протоколи IoT-зв'язку.

1 Стан питання та постановка завдання

У сучасній системі освіти, зокрема в контексті реформ Нової української школи та інтеграції STEM-напряму, зростає потреба в ефективних цифрових засобах, які забезпечують учням практичне розуміння природничих явищ. Об'єктом професійної діяльності в даній роботі є інтерактивна апаратно-програмна система, призначена для моделювання процесу фотосинтезу. Вона поєднує сенсори фізичного середовища, обчислювальний модуль, індикатори візуалізації та мережеву складову для збору й представлення даних.

Значення такої системи полягає в її здатності реалізовувати освітні функції шляхом поєднання біологічного контексту та сучасної цифрової інженерії. Створення подібного пристрою є особливо актуальним для шкіл, які працюють за методикою Монтесорі [4], де самостійне дослідження, експеримент та сенсорне сприйняття мають пріоритет над традиційною лекційною формою подачі матеріалу.

Існуючі навчальні засоби переважно або демонстраційні, або теоретичні. Повноцінних апаратно-програмних рішень з можливістю збору даних, інтерактивного керування та візуалізації результатів у режимі реального часу практично не використовується в освітніх закладах через високу вартість або складність реалізації.

Поставлена задача полягає у створенні програмованого симулятора фотосинтезу, який:

- забезпечує зчитування параметрів середовища (температури, вологості, освітленості);
- дозволяє впливати на ці параметри за допомогою елементів керування або зовнішніх змін;
- обчислює сприятливість умов для фотосинтезу;
- виводить інформацію на дисплей та LED-індикатори;

- має веб-інтерфейс для локального моніторингу;
- зберігає дані для подальшого аналізу у вигляді таблиць CSV та/або у хмарному середовищі.

Саме завдяки поєднанню навчальної та демонстраційної цінності, а також простоті повторення, масштабування та модернізації, розробка такого симулятора є актуальною і доцільною для сучасного освітнього середовища.

1.1 Характеристика та умови роботи об'єкта впровадження

Об'єктом впровадження у межах даної кваліфікаційної роботи є натурний стенд (апаратно-програмний симулятор фотосинтезу), що являє собою інтерактивну систему для моделювання впливу зовнішніх факторів на процес фотосинтезу. Сфера застосування розробленого стенду – навчальні середовища шкіл, а також природничі кабінети загальноосвітніх навчальних закладів, лабораторії та STEM-центри. Це навчальний процес з природничих наук у школі Монтесорі, зокрема для учнів молодшого та середнього шкільного віку (6–12 років).

У межах Монтесорі-методики особлива увага приділяється самостійному дослідженню дитини, зосередженню на конкретному досвіді через сенсорне сприйняття, практичні експерименти та спостереження [4]. Саме натурний стенд симулятора фотосинтезу дозволяє забезпечити ці педагогічні вимоги, створюючи умови для безпечного, наочного й інтерактивного експерименту з можливістю змінювати параметри середовища (освітленість, температуру, вологість) та спостерігати їхній вплив на фотосинтетичну активність.

У межах курсу природознавства тематика фотосинтезу [2] розглядається як фундаментальна біологічна тема, що демонструє взаємозв'язок світла, повітря, води й життя на Землі. Саме тому створення інтерактивного симулятора, що

дозволяє учням дослідити цей процес у доступній формі, є логічним і педагогічно виправданим рішенням.

У традиційній практиці вивчення фотосинтезу в таких умовах обмежується теоретичними моделями або спостереженням за рослинами впродовж тривалого часу, що не завжди дає миттєвий та наочний результат.

Розроблена система симулятора фотосинтезу вирішує цю проблему шляхом створення програмно-керованого стенду, який відтворює вплив факторів середовища (температури, освітленості, вологості) на процес фотосинтезу та демонструє його результат у вигляді світлової індикації, графіків та цифрових значень на екрані.

Користувач (Учень / Вчитель)

- └──→ Спостерігає дані (OLED / Веб)
- └──→ Змінює умови (світло, температура, вологість)
- └──→ Аналізує індекс фотосинтезу
- └──→ Переглядає графіки/звіт

Система симулятора

- └──→ Читає сенсори
- └──→ Обробляє дані
- └──→ Розраховує P_index (0–100%)
- └──→ Виводить дані на OLED / LED / Web
- └──→ Зберігає до дані

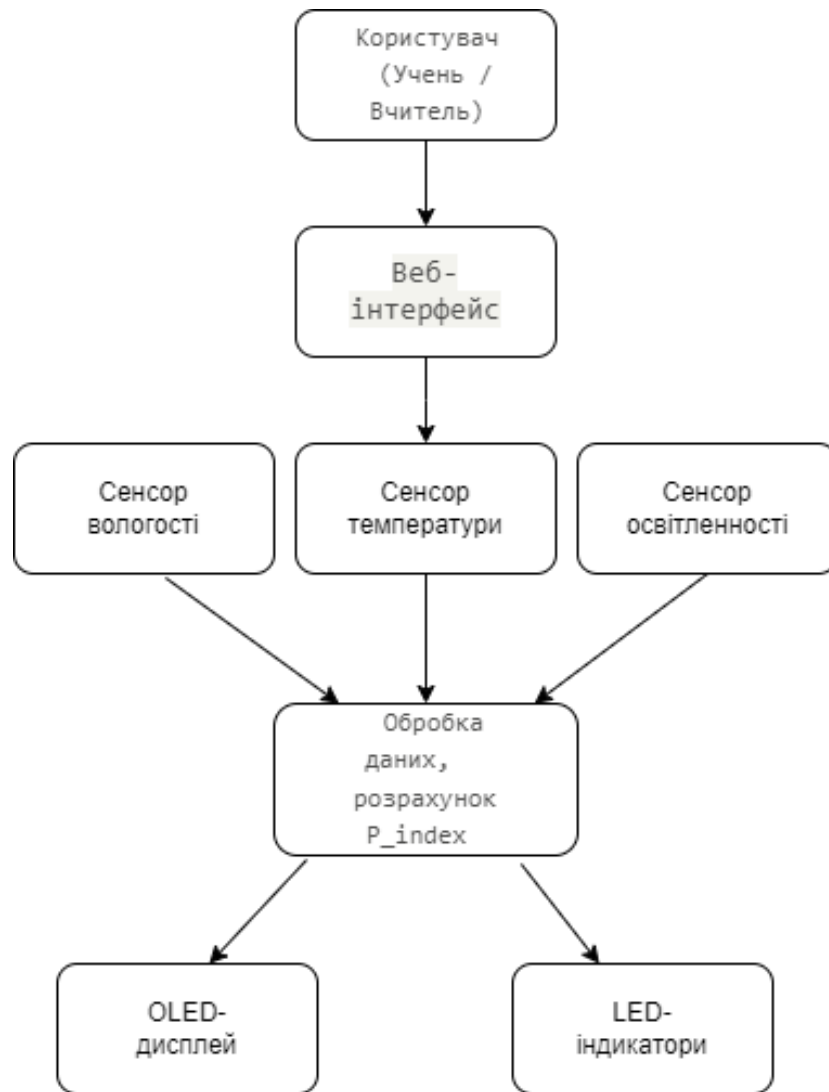


Рисунок 1.1 – Система симулятора

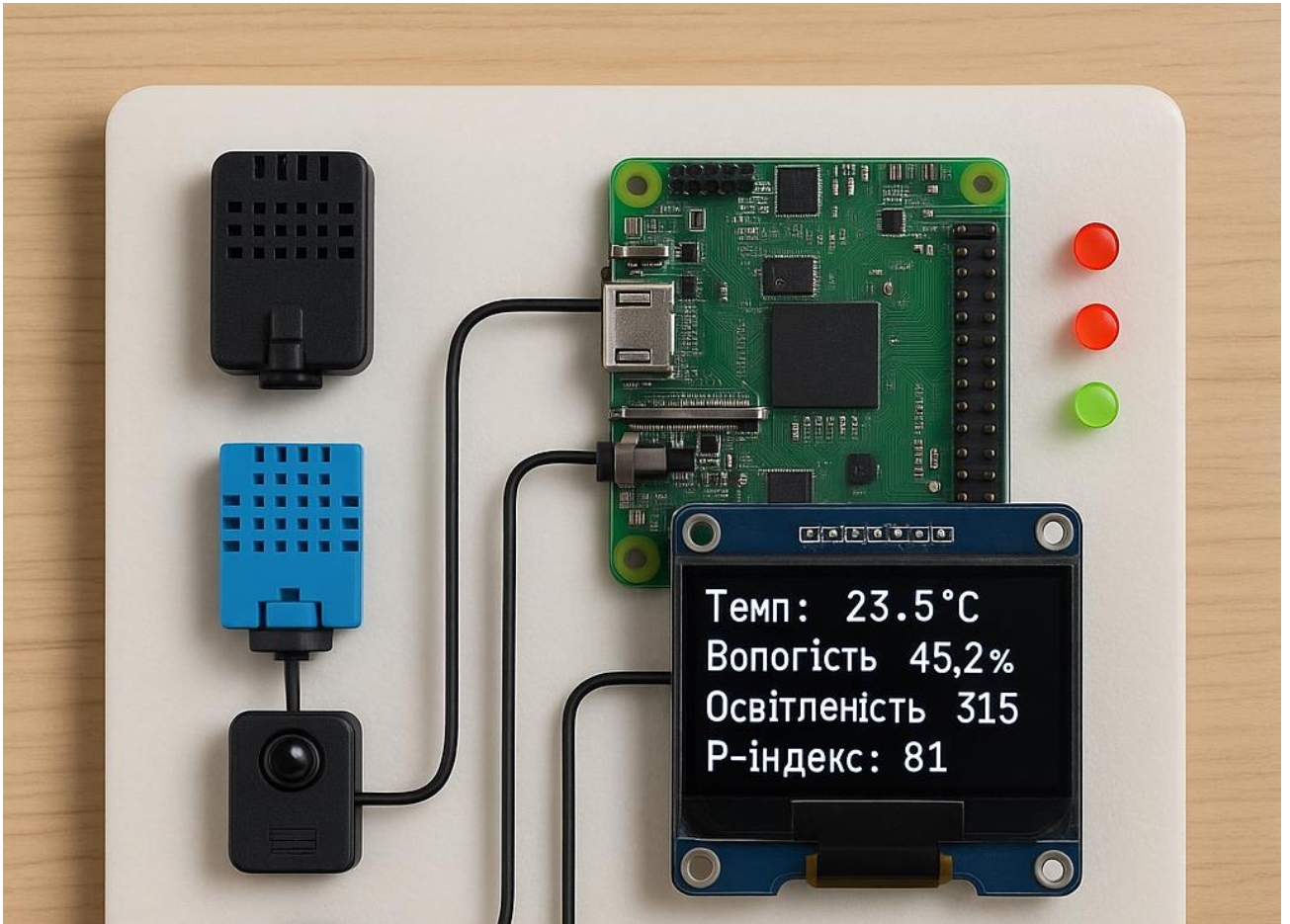


Рисунок 1.2 – Ілюстрація вигляду готового симулятора

Фотосинтез [2] — це один з найважливіших біологічних процесів, що забезпечує існування життя на планеті. Його розуміння є ключовим компонентом шкільного курсу біології. Однак на практиці учні стикаються з труднощами у засвоєнні абстрактної інформації, пов'язаної з молекулярними реакціями, які неможливо безпосередньо побачити.

Саме тому створення апаратного симулятора, який реагує на зміну зовнішніх умов, є актуальним способом перетворити складний процес на інтерактивну модель, з якою дитина може експериментувати.

Таким чином, дана система відповідає сучасним освітнім потребам, поєднуючи сучасні технології (IoT, сенсори, веб-інтерфейс) із педагогічними принципами Монтесорі (самостійне дослідження, практичний досвід), а також

науковою основою (моделювання фотосинтезу). Це зумовлює її актуальність у контексті реальних запитів освітнього процесу та технічного прогресу.

Система дозволяє створити умови безпечного експерименту, що важливо в освітньому середовищі, та сприяє розвитку міжпредметного мислення (поєднання біології, інформатики, фізики).

Розглянемо методику роботи з інтерактивним симулятором фотосинтезу:

1. Підготовка симулятора до роботи

- Перевіряємо підключення сенсорів температури, вологості та освітленості до обчислювального модуля.
- Увімкаємо живлення системи та переконуємось, що індикатори та дисплей працюють.
- Переконуємося, що обчислювальний модуль підключений до локальної мережі (для доступу до веб-інтерфейсу).

2. Запуск програмного забезпечення

- Запускаємо програму симулятора (наприклад, через команду `python3 simulator.py`).
- Переконуємося, що веб-сервер активний (сторінка доступна за локальною IP-адресою пристрою).

3. Вибір режиму експерименту

- Користувач (учень або вчитель) може працювати у двох режимах:
 - Фізичний інтерфейс: змінювати параметри середовища за допомогою впливу на сенсори
 - Веб-інтерфейс: змінювати умови (наприклад, імітувати зміну освітленості) через браузер на комп'ютері або планшеті.

4. Проведення експерименту

— Зміна параметрів:

Змінюємо температуру, вологість або освітленість (наприклад, накриємо датчик світла, нагріємо повітря феном).

— Спостереження:

Система в реальному часі зчитує дані сенсорів, розраховує індекс фотосинтезу (P_index) та виводить результат:

— На OLED/LED-дисплей (цифрове значення, кольорова шкала)

— У веб-інтерфейсі (графік, числові значення, історія змін)

— Аналіз результатів:

Учень спостерігає, як зміна умов впливає на індекс фотосинтезу, формує гіпотези, робить висновки.

5. Логування та збереження даних

— Всі дані експерименту автоматично зберігаються у CSV-файл на пристрої.

— За бажанням, дані можуть експортуватися у Google Sheets для подальшого аналізу, побудови графіків та підготовки звітів.

6. Завершення роботи

— Після завершення експерименту вимикаємо живлення симулятора.

— За потреби, аналізуємо збережені дані, будуємо графіки залежностей (наприклад, P_index vs Light, P_index vs Temperature).

В аналогічних IoT-системах для моделювання фотосинтезу використовують такі основні типи сенсорів:

Таблиця 1.1 – Огляд існуючих датчиків

Сенсор та вимірюваний параметр	Діапазон роботи	Точність	Інтерфейс	Особливості/Переваги	Типові застосування

BH1750 Освітленість	0–65 535 лк	±1–3% або ±20 лк	I2C	Висока точність, простота інтеграції	ІоТ, STEM, освітні проекти
TSL2561 Освітленість	0–40 000 лк	±10%	I2C	Два канали (інфрачервоний+світлий)	Освітні, лабораторні системи
DHT22 (AM2302) Температура, вологість	Т: – 40...+80 ° C, RH: 0– 100%	Т: ±0.5 ° C, RH: ±2%	1- Wire/GPI O	Дешевий, простий, цифровий вихід	ІоТ, DIY, навчальні стенди
DS18B20 Температура	- 55...+125 °C	±0.5 ° C	1-Wire	Висока стабільність, водозахист	Точне вимірювання температури
MH-Z19 CO ₂	0–5 000/10 000 ppm	±(50 ppm + 5% вимір.)	UART, PWM	Вбудоване калібрування, простий монтаж	Лабораторії, екологічні проекти
AM2301 Температура, вологість	Т: – 40...+80 ° C, RH: 0– 99%	Т: ±0.5 ° C, RH: ±3%	1- Wire/GPI O	Альтернатива DHT22	Освітні, побутові системи
Senseair S8 CO ₂	400–5 000 ppm	±(30 ppm +	UART, PWM	Професійна точність, компактність	Лабораторії, smart- офіси

		3% вимір.)			
MAX44009 Освітленість	0,045–188 000 лк	±10%	I2C	Дуже широкий діапазон, низьке енергоспоживання	Освітні, мобільні пристрої
SHT31 Температура, вологість	T: – 40...+125 °C, RH: 0– 100%	T: ±0.3 ° C, RH: ±2%	I2C	Висока точність, швидкий відгук, стабільність	Лабораторії, професійні системи
K30 CO ₂	0–10 000 ppm	±(30 ppm + 3% вимір.)	UART, аналоговий	Висока точність, стабільність, промисловий стандарт	Лабораторії, автоматизація

BH1750, TSL2561 — найпопулярніші для вимірювання освітленості у навчальних і DIY IoT-системах через простоту підключення до Raspberry Pi/Arduino та високу точність.

MAX44009 — сенсор освітленості з надшироким діапазоном, ідеальний для різних рівнів освітлення, має низьке енергоспоживання, але трохи поступається точністю BH1750 у низьких значеннях.

DHT22, AM2301 — універсальні сенсори для температури та вологості, часто використовуються у навчальних симуляторах через цифровий вихід і простоту інтеграції.

DS18B20 — для точного вимірювання температури, особливо коли потрібна висока стабільність або водозахист.

SHT31 — один із найточніших сенсорів температури та вологості, рекомендований для лабораторних або професійних рішень, але дорожчий і складніший у підключенні, ніж DHT22.

MH-Z19, Senseair S8 — застосовуються у більш професійних або лабораторних рішеннях для контролю концентрації CO₂, що є важливим фактором у реальному фотосинтезі.

K30 — CO₂-сенсор промислового класу з високою точністю, використовується у професійних лабораторіях та автоматизованих системах.

Вибір сенсора залежить від цілей проєкту: для базових освітніх завдань достатньо BH1750 і DHT22; для глибших біологічних експериментів додають CO₂-сенсори.

Точності сенсорів для освітньої IoT-системи, яка моделює фотосинтез, повинна відповідати таким критеріям:

— Освітленість:

Необхідна точність — не гірше $\pm 3\%$ або ± 20 лк у діапазоні 0–1000 лк. Це дозволяє фіксувати навіть незначні зміни освітлення, які впливають на індекс фотосинтезу, і дає змогу учням спостерігати причинно-наслідкові зв'язки.

— Температура:

Достатньо похибки не більше $\pm 0,5$ °C у діапазоні 20–50 °C. Така точність дає змогу моделювати вплив температури на фотосинтетичну активність, не спотворюючи результатів експерименту2.

— Вологість:

Оптимальна похибка — до $\pm 2\%$ RH у діапазоні 30–90%. Це забезпечує наочність зміни параметрів і достовірність моделі для навчальних цілей.

Для освітньої IoT-системи, яка моделює фотосинтез, точність сенсорів повинна відповідати таким критеріям:

— Освітленість:

Необхідна точність — не гірше $\pm 3\%$ або ± 20 лк у діапазоні 0–1000 лк. Це дозволяє фіксувати навіть незначні зміни освітлення, які впливають на індекс фотосинтезу, і дає змогу учням спостерігати причинно-наслідкові зв'язки.

— Температура:

Достатньо похибки не більше $\pm 0,5$ °C у діапазоні 20–50 °C. Така точність дає змогу моделювати вплив температури на фотосинтетичну активність, не спотворюючи результатів експерименту.

— Вологість:

Оптимальна похибка — до $\pm 2\%$ RH у діапазоні 30–90%. Це забезпечує наочність зміни параметрів і достовірність моделі для навчальних цілей.

Таблиця 1.2 Порівняння сенсорів освітленості.

Параметр	VN1750 (I2C)	TSL2561 (I2C)	MAX44009 (I2C)	Обраний варіант
Освітленість	1–65 535 лк, ± 1 –3%	0–40 000 лк, $\pm 10\%$	0–188 000 лк, $\pm 10\%$	VN1750
Пояснення	Висока точність, стабільність, просте підключення	Два канали, але нижча точність, складніша калібровка	Дуже широкий діапазон, але гірша стабільність у	Найкраще співвідношення ціна/точність для навчального проєкту

			низьких значеннях	
--	--	--	----------------------	--

Таблиця 1.3 Порівняння сенсорів температури та вологості.

Параметр	DHT22 (AM2302)	AM2301	SHT31	Обраний варіант
Температура	-40...+80 °C, ±0,5 °C	-40...+80 °C, ±0,5 °C	-40...+125 °C, ±0,3 °C	DHT22
Вологість	0–100%, ±2% RH	0–99%, ±3% RH	0–100%, ±2% RH	
Пояснення	Дешевий, простий, достатня точність, легко інтегрується з Raspberry Pi	Аналог DHT22, але менш поширений	Вища точність, але значно дорожчий і складніший у підключенні	DHT22 оптимальний для навчальних цілей і простий у використанні

Таблиця 1.4 Порівняння сенсорів CO₂.

Параметр	MH-Z19	Senseair S8	K30	Обраний варіант
CO ₂	0–5000/10 000 ppm, ±(50 ppm + 5%)	400–5000 ppm, ±(30 ppm + 3%)	0–10 000 ppm, ±(30 ppm + 3%)	MH-Z19
Пояснення	Висока ціна, складність інтеграції, не є критичним	Аналогічно	Аналогічно	Вибрано MH- Z19 через оптимальне

	для базового навчального експерименту			співвідношення ціна/можливості для навчального IoT-проєкту
--	---------------------------------------	--	--	--

Підсумкове обґрунтування вибору:

- ВН1750 обраний завдяки високій точності, стабільності та простоті підключення, що дозволяє якісно імітувати зміни освітленості у навчальному експерименті.
- DHT22 забезпечує достатню точність для температури та вологості, легко інтегрується, має цифровий вихід та широко підтримується у навчальних спільнотах.
- МН-Z19 обрано як сенсор CO₂, оскільки він забезпечує достатню точність для навчальних експериментів, має простий інтерфейс для інтеграції, підтримує автоматичне калібрування та широко використовується у навчальних і DIY-проєктах. Це дозволяє моделювати вплив концентрації CO₂ на фотосинтез у реальному часі, що підвищує наочність і практичну цінність симулятора для учнів.

Таблиця 1.5 - Зміна факторів зовнішнього середовища в симуляторі фотосинтезу

Фактор	Метод зміни (імітації)	Діапазон	Обраний датчик	Похибка	Обґрунтування вибору
--------	------------------------	----------	----------------	---------	----------------------

Освітленість	Регулювання яскравості світлодіодної лампи через ШІМ	0–1000 лк	ВН1750 (інтерфейс I2C)	$\pm 1-3\%$ або ± 20 лк	ВН1750 має високоточне цифрове вимірювання, достатнє для навчальних цілей; легко інтегрується
Температура	Кероване нагрівання за допомогою ТЕНу в закритому корпусі	20–50 °C	DHT22 (інтерфейс GPIO)	± 0.5 °C	DHT22 забезпечує цифрове вимірювання, достатню точність і сумісність з Raspberry Pi
Вологість	Використання зволожувача повітря з керуванням по реле	30–90 %	DHT22 (інтегровані)	$\pm 2\%$	Вологість вимірюється тим самим сенсором, що й температура, спрощуючи схему
Індекс фотосинтезу	Розрахунок за емпіричною формулою	0–100 % (умовно)	Розрахунок на Raspberry Pi	$\sim \pm 5\%$ (умовно)	Застосовується нормалізована модель, що враховує всі зовнішні фактори

Оскільки реальний процес фотосинтезу є складним біохімічним явищем, що залежить від ряду змінних (вміст CO₂, тип рослини, хлорофіл тощо). Розглянемо від ряду яких зовнішніх фізичних та біологічних факторів залежить ефективність фотосинтезу:

1. Вплив температури [1,2]

- Оптимальний діапазон температури для більшості рослин становить 20–25 °С. За цих умов активність ферментів, що беруть участь у темнових реакціях фотосинтезу, є максимальною.
- Зниження температури призводить до зменшення активності ферментів, що уповільнює темнову фазу фотосинтезу. Мінімальна температура для фотосинтезу у більшості рослин — близько 0 °С, хоча деякі види здатні асимілювати CO₂ і при нижчих температурах.
- Підвищення температури вище оптимальної (30–35 °С і більше) викликає пригнічення фотосинтезу через інгібування ферментів, закриття продихів (щоб зменшити втрати води) та можливе пошкодження хлоропластів.
- Температурний стрес у поєднанні з посухою особливо негативно впливає на фотосинтез, зменшуючи як швидкість асиміляції CO₂, так і транспірацію.

2. Вплив освітленості [1,2]

- Світло — основний енергетичний фактор фотосинтезу. Зі збільшенням інтенсивності освітлення інтенсивність фотосинтезу зростає до певної межі (точки насичення), після чого подальше збільшення світла не призводить до зростання продуктивності.
- Оптимальні умови освітленості для більшості видів рослин — це 30–60% від максимальної інтенсивності сонячного світла. Для світлолюбних рослин ця межа вища, для тіневитривалих — нижча.
- Недостатнє освітлення (нижче точки компенсації) призводить до того, що фотосинтез не компенсує дихання, і рослина не накопичує органічну речовину.
- Якість світла (спектральний склад) також впливає на ефективність фотосинтезу: найбільш ефективно використовується червоне та синє світло.

3. Вплив вологості [1,2]

- Вода необхідна для фотосинтезу, як джерело електронів і для транспортування речовин у рослині.
- Оптимальна відносна вологість повітря для більшості рослин — 50–70%. За низької вологості відбувається закриття продихів для зменшення втрати води, що обмежує надходження CO₂ і знижує інтенсивність фотосинтезу.
- Висока вологість сприяє відкриттю продихів, покращує транспірацію і підвищує ефективність фотосинтезу, але надмірна вологість може призводити до розвитку хвороб та зниження аерації ґрунту.
- Водний дефіцит (посуха) призводить до зниження фотосинтезу як через закриття продихів, так і через пряме пригнічення ферментативних реакцій у хлоропластах.

Комплексна дія температури, освітленості та вологості визначає реальну ефективність фотосинтезу. Оптимальні результати досягаються лише при сприятливому поєднанні всіх трьох факторів.

Відхилення хоча б одного параметра від оптимуму (наприклад, висока температура при низькій вологості) може різко знизити фотосинтетичну активність навіть за достатнього освітлення.

Температура, освітленість і вологість є ключовими фізико-біологічними умовами, що визначають швидкість і ефективність фотосинтезу у рослин. Їх оптимальні значення залежать від виду рослини, але для більшості культур оптимум складає: температура 20–25 °С, освітленість 30–60% від максимуму, вологість 50–70%. Порушення балансу цих факторів призводить до зниження продуктивності фотосинтезу.

Тому у навчальному симуляторі використовується спрощена математична модель, що дає змогу учням побачити взаємозв'язок між зовнішніми факторами та фотосинтетичною активністю.

Вводимо умовний індекс фотосинтезу P_index в межах 0–100%, який розраховується на основі трьох параметрів:

- температури ($^{\circ}C$),
- вологості (% RH),
- освітленості (лк).

Формула умовного індексу фотосинтезу є моделлю для навчального моделювання – вона не відображає реальну біохімію, а лише імітує реакцію рослини на зовнішні фактори, як це сприймає учень.

Формула розрахунку умовного індексу фотосинтезу (приклад) [5]:

Нормалізація показників у межах [0;1]

$$T = (\text{temperature} - 20) / (40 - 20)$$

$$H = (\text{humidity} - 30) / (90 - 30)$$

$$L = (\text{light} - 100) / (900 - 100)$$

$$P_index = (0.4 * L + 0.3 * H + 0.3 * T) * 100$$

Значення коефіцієнтів (0.4, 0.3, 0.3) базуються на базовій біологічній моделі, де освітлення має ключовий вплив [1,2].

Показники температури, вологості й освітлення мають фізично різні одиниці ($^{\circ}C$, %, лк), тому їх потрібно привести до одного масштабу – тобто, до значення в межах від 0 до 1.

Не всі фактори однаково впливають на фотосинтез[1,2,6]:

- освітлення є основним (енергія),
- вологість — як транспортування речовин,
- температура — для ферментативних реакцій.

Тому обрано вагові коефіцієнти: 0.4 освітлення (основне джерело енергії), 0.3 вологість (необхідна для транспорту води), 0.3 температура (впливає на активність ферментів). Після підсумування вагованих значень отримується

значення між 0 і 1. Його множать на 100, щоб мати умовний індекс у %, який можна візуалізувати (на дисплеї, графіку, стрічці LED).

Для прикладу візьмемо:

— температура = 30 °C

— вологість = 60 %

— освітлення = 600 лк

$$T = (30 - 20) / (40 - 20) = 0.5$$

$$H = (60 - 30) / (90 - 30) = 0.5$$

$$L = (600 - 100) / (900 - 100) = 0.625$$

$$P_index = (0.4 * 0.625 + 0.3 * 0.5 + 0.3 * 0.5) * 100 = 56.25 \%$$

Отримуємо результат: Фотосинтез = 56 %. Цей результат можна виводити на дисплей або LED-підсвітку.

Для школяра важливо не побачити реакцію хімії, а побачити наслідки умов: як світло чи температура впливають. Така модель дозволяє: відчутти взаємозв'язки, експериментувати, спостерігати зміни в режимі реального часу.

Візуальна шкала індексу фотосинтезу (P_index)

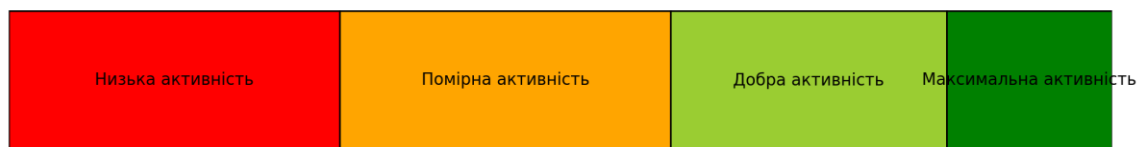


Рисунок 1.3 – Візуальна шкала індексу фотосинтезу.

Ось візуальна шкала індексу фотосинтезу P_index (0–100%), що демонструє рівень активності [5]:

— Червоний колір 0–30% — Низька активність (недостатньо світла або вологи)

— Помаранчевий колір 30–60% — Помірна активність (умови не ідеальні)

- Світло зелений 60–85% — Добра активність (умови сприятливі)
- Зелений 85–100% — Максимальна активність (оптимальні умови)

З усіма наявними даними можна робити наступні графіки та аналізи:

- P_index vs Time: Динаміка фотосинтетичної активності в часі
- P_index vs Light: Залежність активності від освітлення
- P_index vs Humidity: Аналіз чутливості до вологості
- P_index vs Temperature: Крива температурної оптимальності
- Light, Temp, Humidity vs Time: Кореляція зовнішніх факторів між собою.

Для чого це потрібно? Це дозволяє:

- демонструвати причинно-наслідкові зв'язки (що буде, якщо менше світла?),
- стимулювати дослідницьке мислення,
- формувати навички роботи з даними (аналіз, побудова графіків),
- підготувати учнів до STEM/STEAM-підходів.

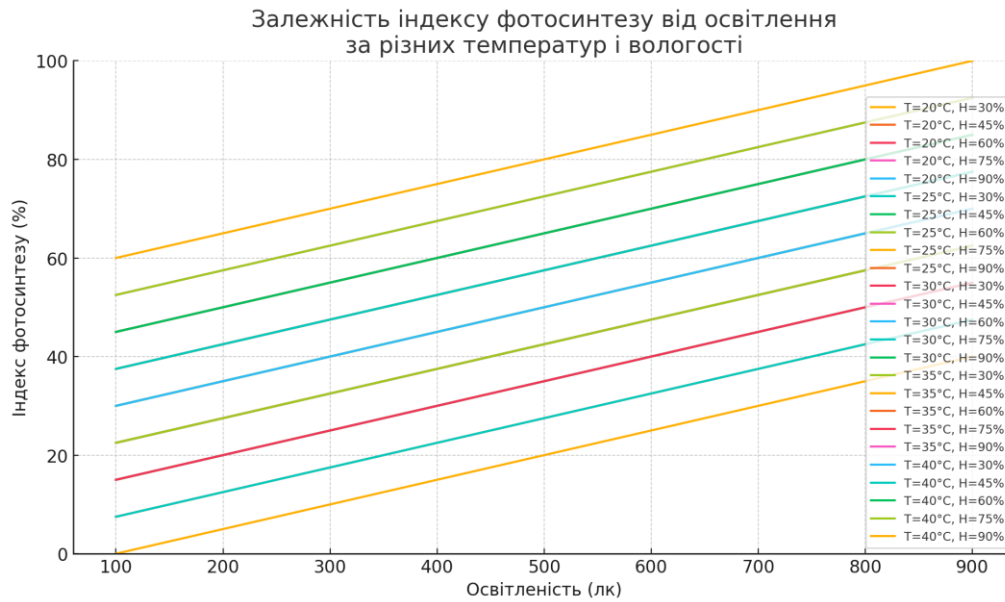


Рисунок 1.4 – Зміна індексу фотосинтезу залежно від рівня освітленості та рівня вологості.

Дана діаграма демонструє, що для досягнення максимального індексу фотосинтезу необхідно забезпечити не лише високу освітленість, а й оптимальні значення температури та вологості. Саме цей принцип лежить в основі інтерактивної моделі симулятора, яка дозволяє учням змінювати параметри середовища і спостерігати результат у реальному часі.

Розглянемо, як саме розраховується похибка зміни факторів у цьому проєкті (температура, вологість, освітленість), і звідки вона береться.

Похибка[3] — це відхилення виміряного значення від реального. Вона може бути:

- систематичною (через обмеження сенсора),
- випадковою (через флуктуації умов чи електроживлення),
- викликаною перетворенням сигналу (наприклад, округленням або перетворенням в цифру).

Точність вимірювання визначається обраним методом і потребами експерименту. При виборі сенсорів враховувалась допустима похибка, що не впливає критично на результати моделювання. Значення похибок, наведені в технічній документації виробника, були використані для оцінки відповідності датчиків вимогам проєкту. Прийнятною вважається похибка в межах, яка дозволяє коректно відображати зміни факторів у рамках навчального симулятора.

1.2 Огляд існуючих аналогів

Для ефективного вивчення фотосинтезу в інтерактивному освітньому середовищі виникає потреба у засобах, які не лише демонструють процес, а й відповідають вимогам сучасного навчального простору. Це зокрема: адаптивність до різних форматів занять (гнучкість), наочність перебігу процесів (візуальність), а також можливість взаємодії учня із пристроєм через прості фізичні або сенсорні дії (сенсорна доступність). Існуючі демонстраційні моделі або лабораторні установки часто виявляються обмеженими щодо інтеграції в такі середовища, особливо в школах, що працюють за Монтессорі-методикою, де акцент робиться на практичне дослідження, самостійність та фізичний контакт із матеріалами.

Таблиця 1.6 – Огляд апаратних рішень

Аналог	Платформа / підхід	Ключові характеристики	Відповідність освітнім вимогам
Arduino STEM Kit (Biology Lab)	Підтримувана платформа з відкритим кодом	Дешева, активно підтримується спільнотою, має базову функціональність для дослідів	Частково відповідає: можливе самостійне складання, але бракує обчислювальної потужності та інтерфейсів
BBC micro:bit з сенсорами	Освітня платформа для молодших учнів	Дуже проста у використанні, візуальне програмування, обмежений набір сенсорів	Обмежено відповідає: придатна для ознайомлення, але не забезпечує глибоке дослідження
PASCO / Vernier Lab Systems	Комерційне рішення з високою точністю	Висока надійність, готові навчальні методики, спеціалізоване ПЗ	Надмірно складна та дорога для Монтесорі-шкіл, закритість перешкоджає модифікаціям
Proposed: Raspberry Pi + сенсори	Відкрита апаратна платформа з мережею IoT	Гнучка, програмована, забезпечує логування, веб-інтерфейс, масштабування, інтеграцію з хмарними сервісами	Відповідає повністю: підходить для навчального середовища Монтесорі з акцентом на дослідність

Таблиця 1.7 – Огляд програмних архітектур

Технологія	Характеристика	Висновок
Arduino C	Швидка обробка, підходить для сенсорів	Не зручно для роботи з мережами або веб-сервісами
Python + Flask	Збалансована, зручна для розгортання на Raspberry Pi	Ідеальне рішення для IoT-симулятора
Node-RED	Візуальне програмування, легко змінювати логіку	Займає багато ресурсів, складне масштабування
JavaScript/WebSockets	Підходить для динамічного фронтенду	Вимагає додаткового серверного оточення

Таблиця 1.8 – Порівняння освітніх симуляторів біологічних процесів

Назва / тип	Функціонал / освітня цінність	Гнучкість	Інтерфейс / взаємодія	Оцінка впровадження в Монтесорі-середовище
PhET "Photosynthesis" (онлайн)	Візуалізація теоретичного процесу. Дає уявлення, але без практичного взаємодійного експерименту.	Низька: не можна змінювати параметри через фізичні дії або створювати власні сценарії.	Веб-інтерфейс, 2D-графіка. Пасивна взаємодія.	Частково відповідає: немає сенсорної активності, не підтримує дослідницький підхід.

Arduino-стенд з CO ₂ -датчиком	Дослідницький інструмент, що демонструє вплив середовища на CO ₂ , але без комплексної моделі.	Середня: можливість змінювати прошивку і зовнішні сенсори, але складний для дітей.	OLED-дисплей, серійний порт. Користувач взаємодіє через ПК.	Потребує супроводу вчителя; складна інтерфейсна логіка для молодших учнів.
LEGO Education SPIKE + екосистема	Орієнтовано на STEAM-підхід, розвиток логіки, збір простих даних, програмування блоками.	Висока: гнучка механіка, можливість зміни конструкцій, але обмежена в датчиках для біології.	Програмований графічний інтерфейс, дружній до дітей.	Добре підходить для творчого моделювання, проте не дає глибини саме у фотосинтезі.
Готові STEM-набори (Labdisc, Vernier)	Професійне вимірювання параметрів середовища. Придатне для лабораторних умов.	Низька: закриті системи, обмежені у зміні конфігурації або додаванні нових сценаріїв.	Пропріетарне ПЗ, планшет або ноутбук.	Невідповідна гнучкість для методики Монтессорі; важкодоступна інтеграція.
ІоТ-симулятор фотосинтезу (цей проєкт)	Інтерактивне моделювання процесу фотосинтезу зі зміною умов (світло, вологість, температура).	Висока: зміна параметрів у реальному часі, доступ до коду, підтримка різних сценаріїв.	OLED-дисплей, веб-інтерфейс, LED-індикація. Можливість сенсорної взаємодії.	Відповідає повністю: підтримує дослідність, наочність, гнучкість і автономію учнів.

Raspberry Pi краще підходить для проєктів, що вимагають веб-інтерфейсу, обробки даних, гнучкого підключення та мультизадачності. Обрана платформа Raspberry Pi дозволяє реалізувати високофункціональний освітній симулятор, орієнтований на принципи Монтессорі-педагогіки:

- Зв'язок із середовищем через фізичні сенсори;
- Реальний інтерфейс користувача для візуалізації навчального процесу;
- Гнучке масштабування та інтеграція з веб-технологіями.

Таблиця 1.9 – Похибка сенсорів.

Сенсор	Фактор	Типова похибка
VH1750	Освітленість	$\pm 1-3\%$ або ± 20 лк
DHT22	Температура	± 0.5 °C
DHT22	Вологість	$\pm 2\%$ RH

Коли ми використовуємо похибки вхідних значень у формулі, ми можемо оцінити загальну похибку індексу фотосинтезу.

Якщо формула має вигляд:

$$I = 0.4 * L + 0.3 * H + 0.3 * T$$

Тоді максимальна похибка індексу:

$$\Delta I = 0.4 * \Delta L + 0.3 * \Delta H + 0.3 * \Delta T$$

Приклад:

— $\Delta L = \pm 3\%$ (освітленість)

— $\Delta H = \pm 2\%$ (вологість)

— $\Delta T = \pm 0.5$ °C \rightarrow (нормалізовано $\pm 2.5\%$) для діапазону 20–40

Тоді:

$$\Delta I \approx 0.4 * 3\% + 0.3 * 2\% + 0.3 * 2.5\% \approx 1.2\% + 0.6\% + 0.75\% = \sim 2.55\%$$

Тобто індекс фотосинтезу може мати максимальну похибку приблизно $\pm 2.5\%$.

Додаткові джерела похибки можуть виникати у разі:

- Затримка реакції сенсора (наприклад, DHT22 оновлює значення лише раз на 2 секунди);
- Електричні наводки на проводах;
- Нестабільне живлення пристрою або датчиків;
- Вологість чи пил у повітрі (впливає на освітленість).

1.3 Обґрунтування вибраного напрямку вирішення задачі

З урахуванням цілей проєкту, специфіки об'єкта впровадження (Монтессорі-середовище) та вимог до відкритості, модифікованості і простоти, було обрано апаратну платформу Raspberry Pi 4 [13,14] із програмною реалізацією на Python + Flask [8,12], інтеграцією сенсорів через GPIO та використанням Google Sheets API [10] для логування. Це забезпечує:

- простоту збирання й обслуговування;
- повноцінний мережевий інтерфейс;
- підтримку виводу даних у графічній формі;
- можливість масштабування без зміни архітектури.

Структура об'єкта професійної діяльності включає:

- обчислювальний модуль;
- сенсори температури і вологості та освітленості;
- блок візуалізації результату;
- веб-сервер для локального моніторингу;
- логування результатів у локальні файли;

— елементи керування (кнопка, повзунок, регулятори тощо – за конфігурацією).

Функціональна структура системи побудована за принципом:

«Вимірювання параметрів → Обчислення рівня фотосинтезу → Візуалізація → Логування → Мережевий доступ»

Система має модульну функціональну структуру, де кожен блок виконує окрему роль.

Сенсорна підсистема: Зчитування температури, вологості та освітлення.

Аналітичний модуль: Розрахунок рівня сприятливості для фотосинтезу за шкалою (зелений, жовтий, червоний стан).

Інтерфейс користувача:

- Локальний OLED-дисплей для поточних значень;
- Світлодіодна стрічка для візуального індикатора стану;
- Кнопка для перезапуску або перемикання режиму.

Програмна платформа:

- Python-скрипт з циклічним зчитуванням даних;
- Flask-сервер для веб-інтерфейсу та виводу історичних даних;
- Модуль логування (CSV + Google Sheets API).

Зовнішні сервіси: Вивантаження до Google Таблиць для аналізу.

Вибраний напрямок забезпечує:

- Високу модульність;
- Низьку вартість впровадження;
- Гнучкість налаштування для різних освітніх рівнів;
- Можливість подальшої інтеграції зі Smart Lab середовищем.

Таблиця 1.10 – Пояснення використання інтерфейсів у системі

Інтерфейс	Призначення	Конкретна реалізація
GPIO	Підключення кнопок/LED-стрічки	GPIO через бібліотеку RPi.GPIO або gpiozero
I2C	Обмін з датчиками BH1750 (освітленість), OLED-дисплеєм	Через smbus (адресація пристроїв по шині)
DHT (1-Wire)	Зчитування температури і вологості (DHT22)	Через бібліотеку Adafruit_DHT
PWM	Яскравість світлодіодів (опційно)	GPIO PWM канал
Ethernet/Wi-Fi	Доступ до Flask, Google Sheets API	Працює через локальну мережу або точку доступу Raspberry Pi
USB	Підключення до ноутбука або монітора, для налаштування	Через SSH або прямий доступ

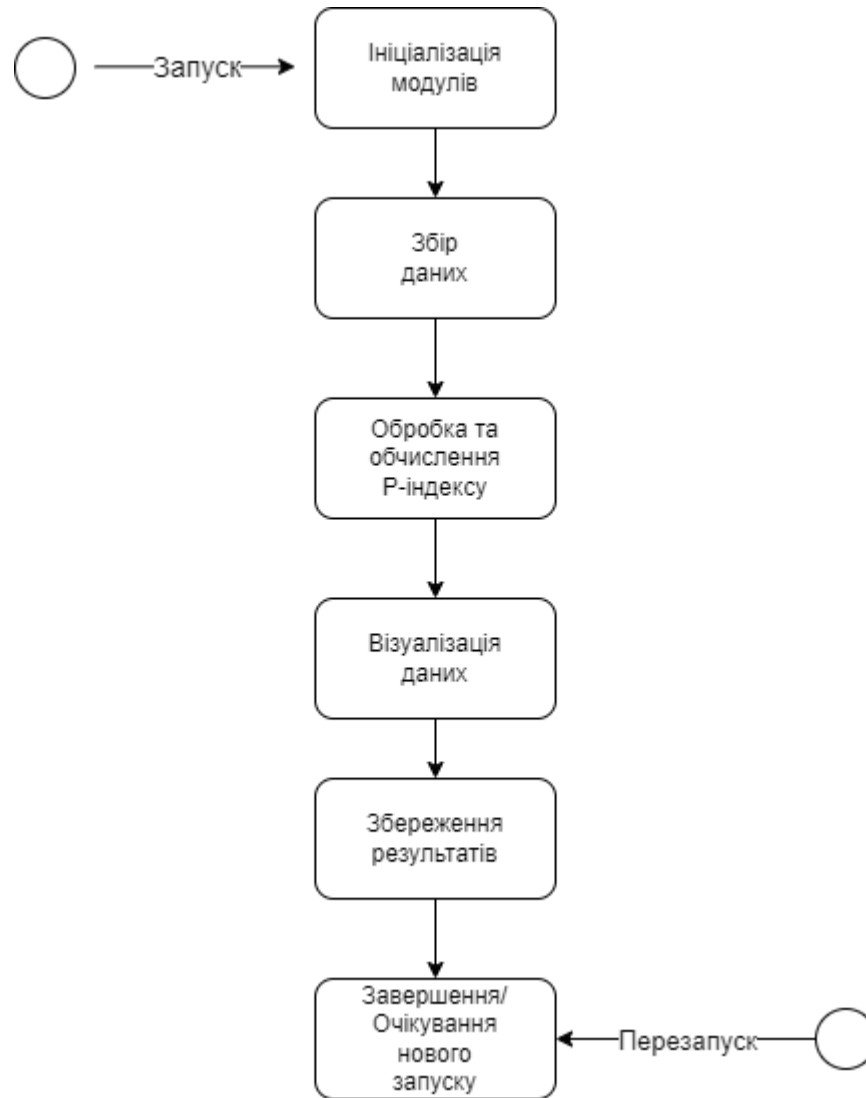


Рисунок 1.5 – UML-діаграма поведінки об'єкту

Кожен блок діаграми відображає послідовний етап роботи симулятора. Від запуску до збереження результатів і можливості повторного експерименту, що забезпечує повний цикл навчального дослідження.

1. Ініціалізація модулів

На цьому етапі система запускає всі необхідні програмні та апаратні модулі: ініціалізуються сенсори, дисплеї, інтерфейси зв'язку та підготовлюється середовище для подальшої роботи.

2. Збір даних

Система циклічно зчитує поточні значення з усіх підключених сенсорів (температура, вологість, освітленість, CO₂ тощо).

3. Обробка та обчислення P-індексу

Зібрані дані нормалізуються, перевіряються на коректність, після чого за заданою формулою розраховується умовний індекс фотосинтезу (P_index).

4. Візуалізація даних

Результати розрахунків відображаються на OLED-дисплеї, LED-індикаторах, а також у веб-інтерфейсі у вигляді числових значень, кольорової шкали чи графіків.

5. Збереження результатів

Усі отримані дані зберігаються у локальні CSV-файли та/або експортуються у Google Таблиці для подальшого аналізу та звітності.

6. Завершення / Очікування нового запуску

Система переходить у режим очікування або завершує роботу. За потреби користувач може ініціювати перезапуск для нового циклу експерименту.

7. Перезапуск

За командою користувача система повертається до початкового етапу, повторюючи весь цикл знову

Топологічна схема підключення компонентів базується на послідовному з'єднанні сенсорів до GPIO-портів Raspberry Pi [13].

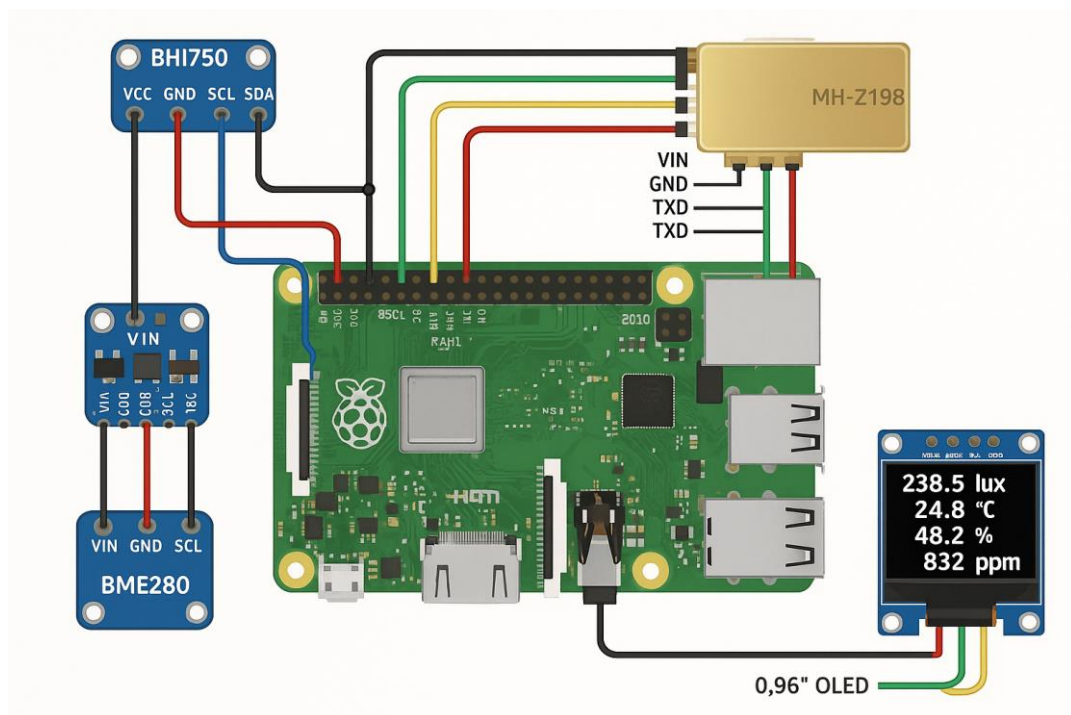


Рисунок 1.6 - Топологічна схема підключення

1.4 Мета та задачі роботи

Мета роботи полягає у створенні інтерактивної апаратно-програмної IoT-системи, що дозволяє моделювати процес фотосинтезу в умовах навчального середовища школи Монтесорі.

Для досягнення поставленої мети у дипломному проєкті вирішуються наступні завдання:

- Проаналізувати фізико-біологічні умови, що впливають на процес фотосинтезу, з акцентом на температурі, освітленні та вологості.
- Вивчити особливості навчального процесу в Монтесорі-середовищі та визначити вимоги до пристрою з точки зору педагогіки.
- Розробити апаратну частину симулятора з використанням відповідних сенсорів, індикаторів та дисплея.

Для досягнення мети створення інтерактивної IoT-системи симулятора фотосинтезу, окрім апаратної частини, у дипломному проєкті вирішуються такі завдання, що стосуються програмного забезпечення:

- Розробити програмне забезпечення для зчитування даних з сенсорів температури, вологості, освітленості та CO₂ у реальному часі.
- Реалізувати алгоритм обробки та нормалізації отриманих даних, а також розрахунку умовного індексу фотосинтезу відповідно до заданої математичної моделі.
- Створити модуль візуалізації результатів:
 - Вивід поточних значень параметрів та індексу фотосинтезу на OLED-дисплей.
 - Кольорова індикація стану фотосинтезу за допомогою LED-стрічки.
 - Веб-інтерфейс на базі Flask для відображення даних, графіків і історії змін через браузер.

- Забезпечити автоматичне логування результатів експерименту:
 - Збереження даних у локальні CSV-файли з часовими мітками.
 - Експорт даних у Google Таблиці через відповідне API для подальшого аналізу та звітності.
- Реалізувати функції управління через інтерфейс користувача:
 - Перезапуск вимірювань, скидання значень, вибір режиму роботи (автоматичний/ручний/демонстраційний).
 - Можливість масштабування та оновлення програмних модулів для додавання нових функцій або сенсорів

2 Спеціальний розділ. Розробка системи

2.1 Найменування та призначення об'єкту

Найменування:

Інтерактивна IoT-система симулятора фотосинтезу для освітніх цілей.

Призначення:

Об'єкт професійної діяльності призначено для реалізації функцій моделювання, візуалізації та аналізу процесу фотосинтезу в умовах змінних параметрів навколишнього середовища. Система створена для використання у навчальному середовищі (зокрема, у школах Монтесорі)

2.1.1 Вимоги до структури та функціонування

Розробка апаратно-програмного симулятора фотосинтезу як об'єкта професійної діяльності передбачає відповідність низці структурних та функціональних вимог, які забезпечують стабільну, безпечну та ефективну експлуатацію пристрою у навчальному середовищі.

Таблиця 2.1 - Перелік функціональних складових та їх призначення

№	Назва підсистеми	Призначення
1	Сенсорна підсистема	Здійснює зчитування фізичних параметрів середовища – температури, вологості та освітленості.

2	Обчислювальний модуль	Обробляє сигнали з сенсорів, обчислює рівень сприятливості умов для фотосинтезу, виконує логіку керування.
3	Підсистема візуалізації	Відображає поточні параметри та результати аналізу: OLED-дисплей для числових значень, WS2812 – світлова індикація рівня фотосинтезу.
4	Комунікаційна підсистема	Забезпечує локальну взаємодію з користувачем через веб-інтерфейс (Flask) та відправку даних до хмарного сховища (Google Sheets).
5	Підсистема логування даних	Зберігає результати кожного циклу вимірювання у форматі CSV-файлу та (опційно) в Google Таблицях.
6	Підсистема керування	Дає змогу запускати або зупиняти процес збору даних, а також ініціювати збереження результатів вручну.
7	Інтерфейс користувача	Забезпечує доступ до даних через веб-браузер: перегляд графіків, таблиць, керування процесом.
8	Енергоживлення та захист	Забезпечує стабільну подачу напруги, захист електронних компонентів, підтримує автономну роботу від зовнішнього джерела.

Структурні вимоги:

Модульна архітектура

1. Система повинна складатися з окремих функціональних блоків (наприклад, сенсорний модуль, модуль індикації, живлення, керування), які можна легко замінювати або масштабувати. Це забезпечує адаптивність під різні освітні завдання, як-от розширення кількості сенсорів чи додавання нових елементів керування.

2. Обчислювальний блок

Відповідно до вимог, для керування системою використовується виключно одноплатний комп'ютер Raspberry Pi (версія 4 або 5). Цей модуль виконує обробку даних, запуск веб-сервера, збереження результатів і взаємодію з периферією. Завдяки повноцінній операційній системі можливе впровадження багатозадачності, роботи з інтернет-ресурсами, логування подій тощо.

Даний перелік визначає ключові структурні, функціональні та експлуатаційні вимоги до апаратно-програмної IoT-системи симулятора фотосинтезу, яка призначена для використання в освітньому середовищі. У цьому розділі систематизовано вимоги до основних підсистем, режимів роботи, способів зберігання та обробки даних, а також до взаємодії користувача із системою. Це забезпечує надійність, безпечність, адаптивність і зручність експлуатації пристрою у навчальних закладах різного типу.

1. Комунікаційна підсистема.

Реалізується локальний веб-сервер на базі Flask (Python), що дозволяє переглядати дані з будь-якого пристрою у локальній мережі (через Wi-Fi або Ethernet), формувати таблиці, графіки, завантажувати CSV-файли тощо.

2. Живлення.

Система повинна працювати від джерела постійного струму 5В, що дозволяє безпечно використовувати її у класі. Також можливе застосування PowerBank або акумуляторного модуля для автономної роботи.

Функціональні вимоги:

1. Циклічне зчитування параметрів середовища.

Сенсори повинні працювати з частотою оновлення не менше 1 разу на 5 секунд для забезпечення візуального сприйняття змін.

2. Обчислення рівня фотосинтезу.

Алгоритм повинен враховувати всі три параметри (температура, вологість,

освітленість), нормалізувати їх до шкали та обчислювати умовний «коефіцієнт сприятливості».

3. Виведення даних.

Дані мають бути подані у зручному для учня форматі:

- числові значення;
- колірна індикація стану;
- текстові пояснення на дисплеї чи у веб-інтерфейсі.

4. Зберігання результатів.

- локально у CSV-файли з часовими мітками;
- у Google Таблиці через відповідне API, за наявності інтернет-з'єднання.

5. Управління.

Користувач повинен мати можливість перезапуску вимірювань, скидання значень, зміни режиму (наприклад, «автоматичний» / «ручний»).

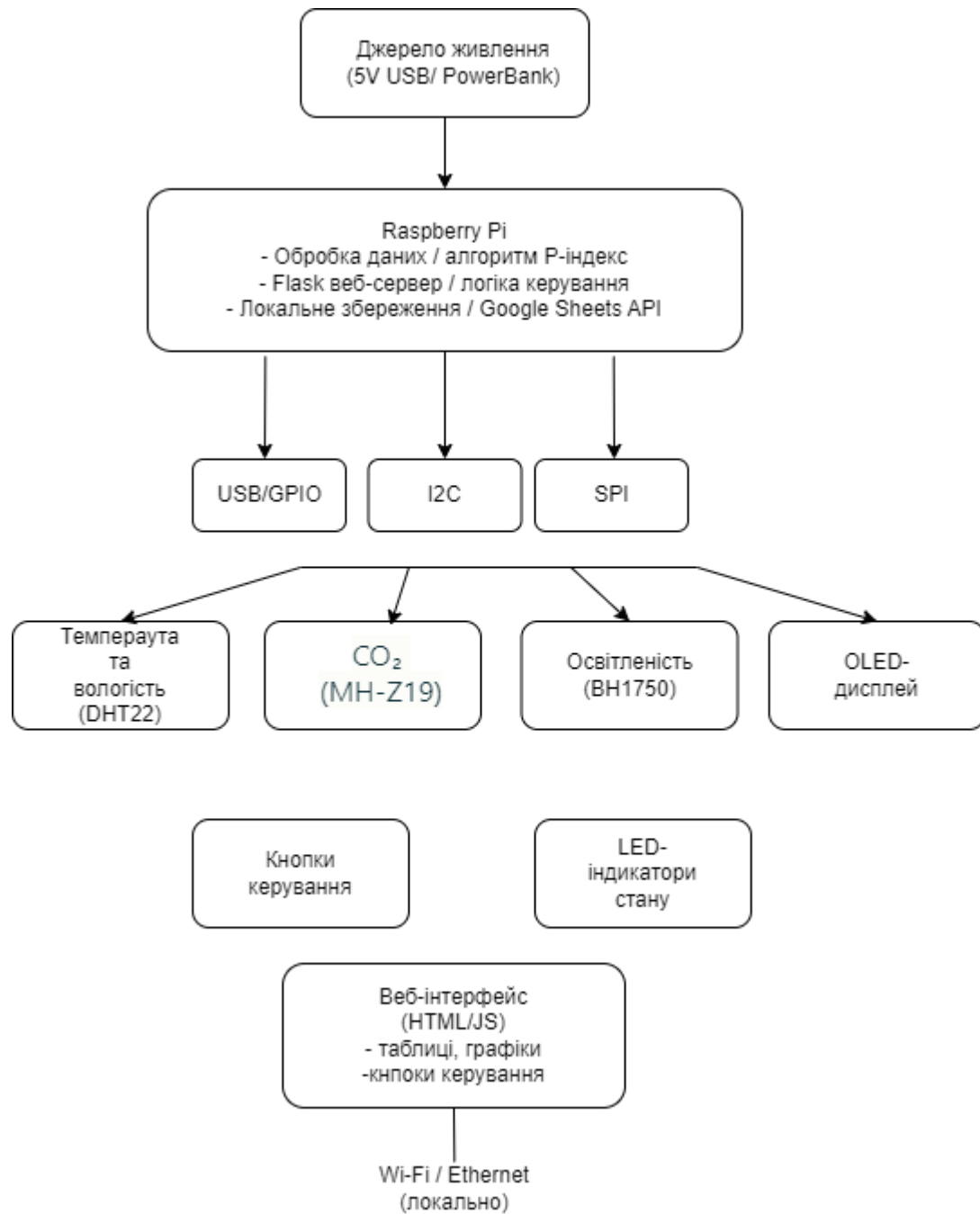


Рисунок 2.1 - Схема функціональних блоків

З метою забезпечення коректного функціонування системи, надійного обміну даними між апаратними та програмними модулями, встановлюються наступні вимоги до способів і засобів зв'язку в межах симулятора фотосинтезу.

1. Апаратний рівень зв'язку

Обмін даними між сенсорами та мікрокомп'ютером Raspberry Pi реалізується через стандартні інтерфейси:

Таблиця 2.2 – Апаратний рівень зв'язку

Компонент	Інтерфейс	Функції системи	Характеристика
DHT22	GPIO (1-Wire)	Зчитування температури та вологості	Цифровий сигнал, ≈ 1 Гц
BH1750	I ² C	Зчитування рівня освітлення	Адреса I ² C: 0x23 або 0x5C
OLED дисплей	I ² C	Вивід текстової інформації	Адреса I ² C: 0x3C
WS2812 (LED стрічка)	PWM (GPIO)	Візуальна індикація рівня фотосинтезу	Керування кольором та яскравістю

Апаратні модулі були обрані на основі аналізу вимог до навчального симулятора фотосинтезу, з урахуванням їхньої функціональності, сумісності та доступності. Сенсор DHT22 забезпечує надійне зчитування температури та вологості, що є критичними факторами для моделювання умов фотосинтезу. Освітленість, як один із ключових параметрів, ефективно відстежується за допомогою датчика BH1750 з інтерфейсом I²C. OLED-дисплей дає змогу наочно відобразити інформацію, що важливо в освітньому процесі.

LED-стрічка WS2812 обрана як наочний засіб індикації рівня фотосинтетичної активності для кращого візуального сприйняття учнями. Кожен модуль доповнює загальну функціональність системи, створюючи збалансовану та логічно обґрунтовану апаратну структуру.

Усі підключення реалізуються через розширювальну плату (GPIO header) із дотриманням правил електробезпеки (низьковольтні сигнали, ≤ 3.3 В).

1. Мережеві параметри:

- Локальний веб-сервер (Flask) працює за протоколом HTTP (порт 5000);
- Для віддаленого перегляду даних необхідне підключення Raspberry Pi до локальної мережі (Wi-Fi або Ethernet);
- Google Sheets API використовує SSL-зашифроване з'єднання.

У межах проєкту передбачено взаємодію створюваної системи симулятора фотосинтезу з зовнішніми цифровими сервісами та, потенційно, іншими компонентами освітньої інфраструктури, які можна віднести до суміжних систем. Взаємозв'язки реалізуються через стандартні протоколи передачі даних та API.

2. Межі модернізації

У перспективі система може бути розширена для взаємодії з іншими IoT-рішеннями або шкільними інформаційними системами, зокрема:

Таблиця 2.3 – Можливі інтеграції

Суміжна система	Характер взаємодії	Протокол/Інтерфейс	Дані, якими обмінюється
Інші симулятори або датчики	Координація, обмін станом	MQTT або HTTP	Стани, параметри середовища
Освітній портал школи	Передача статистики	HTTPS, REST API	Готові CSV або JSON-файли
Мобільний застосунок	Віддалений моніторинг	WebSocket або HTTP polling	Поточні значення сенсорів
Локальна база даних (MySQL)	Локальне зберігання	SQL-запити через Python DB-API	Усі експериментальні дані

Для забезпечення гнучкої роботи системи в умовах навчального середовища, розроблений симулятор повинен підтримувати декілька режимів функціонування, що дозволяють адаптувати його до потреб учнів, викладачів та технічного персоналу. Режими поділяються на основні та допоміжні.

Режими роботи симулятора фотосинтезу

1. Автоматичний режим

Призначений як основний режим роботи під час проведення дослідів або демонстрацій. У цьому режимі здійснюється періодичне зчитування сенсорів кожні 5 секунд, автоматичне оновлення значень на OLED-дисплеї та через LED-індикацію. Дані логуються у CSV-файл і, за потреби, експортуються в Google Таблицю. Режим не потребує постійного втручання користувача, окрім увімкнення живлення.

2. Ручний режим

Використовується для контрольованого запуску окремих етапів вимірювання або виводу. Запуск вимірювань здійснюється натисканням кнопки або через веб-інтерфейс. Користувач може вручну зберігати CSV-файл або ініціювати експорт у Google Sheets, а також самостійно вмикати або вимикати сенсори та засоби виводу.

3. Відлагоджувальний (тестовий) режим

Призначений для налагодження апаратного чи програмного забезпечення. У цьому режимі в термінал виводиться додаткова службова інформація, така як температура CPU або діагностика сенсорів. Передбачена можливість роботи з емуляцією сенсорних значень. Режим орієнтований на розробника або викладача, який виконує технічне обслуговування системи.

4. Демонстраційний режим

Призначений для показу принципу роботи симулятора без підключення до реального середовища. Дані генеруються випадково або відповідно до заданого

сценарію. Відбувається візуалізація наявної інформації, хоча сенсори фізично не задіяні. Інтерфейс залишається активним для перегляду.

5. Часовий режим роботи

Рекомендований час роботи симулятора — 5–8 годин на день, що відповідає тривалості навчального дня. Максимально допустимий режим — цілодобова робота за умови наявності активного охолодження Raspberry Pi (вентилятор або радіатор). При тривалому простої система автоматично переходить у пасивний режим, у якому вимикається дисплей і відключаються сенсори.

2.1.2 Показники призначення

1. Базова платформа – Raspberry Pi (версія 4 або 5) [14]

Обрано одноплатний комп'ютер Raspberry Pi через підтримку Linux, можливість запуску Python-скриптів, роботу з веб-інтерфейсами та взаємодію з периферією через GPIO, I²C, SPI. Такий вибір обґрунтовано вимогами до обробки сенсорних даних, керування виводом інформації та локального збереження. Модель (версія 4 або 5) залежить від доступності. Обсяг оперативної пам'яті 2–4 ГБ є достатнім для завдань проєкту.

2. Сенсорна підсистема

Передбачено використання датчика DHT22 для вимірювання температури та вологості, а також цифрового сенсора освітленості BH1750. Обидва сенсори мають достатню точність і підтримку в Python-середовищі.

3. Діапазони вимірювання

Температура вимірюється в межах 0...+50 °C, що відповідає умовам класної кімнати або демонстраційного макета. Вологість фіксується в діапазоні 20...90 %, який дозволяє моделювати як сухі, так і вологі середовища.

Освітленість охоплює 0...65 535 люкс — від темряви до яскравого денного світла.

4. Точність вимірювання

Похибка становить ± 0.5 °C для температури, ± 2 % для вологості та ± 10 люкс для освітленості. Ці показники є прийнятними для цілей навчального експерименту.

5. Частота оновлення даних

Зчитування сенсорів відбувається з інтервалом 5 секунд. Така частота забезпечує спостереження за змінами без надмірного навантаження на систему.

6. Виведення інформації

OLED-дисплей (128×64 або 128×32) показує поточні значення параметрів. Світлодіодна стрічка WS2812 виконує роль кольорового індикатора: червоний — несприятливі умови, жовтий — помірні, зелений — оптимальні. Для віддаленого доступу застосовується локальний веб-сервер на Flask [8] з відображенням графіків, значень і історії спостережень.

7. Збереження даних

Дані фіксуються локально у форматі CSV з часовими мітками, з інтервалом збереження кожні 10 секунд. Це дозволяє детально відслідковувати динаміку змін. У разі підключення до мережі можливе резервне збереження та онлайн-моніторинг через API інтеграцію з Google Таблицями [10].

2.2 Розробка апаратної частини

Загальна структура та логіка функціонування:

На основі технічних вимог було побудовано функціональну структуру системи, що складається з трьох основних рівнів:

— Фізичний рівень – сенсори середовища, пристрої індикації, блок живлення.

— Обчислювальний рівень – Raspberry Pi як центральний процесор для зчитування, обробки та керування.

— Інформаційно-візуальний рівень – веб-інтерфейс, дисплей, система логування, індикація на світлодіодах.

Користувач (учень або вчитель) може взаємодіяти з пристроєм через фізичні елементи керування або веб-браузер, переглядати стан середовища в режимі реального часу та аналізувати результати експерименту.

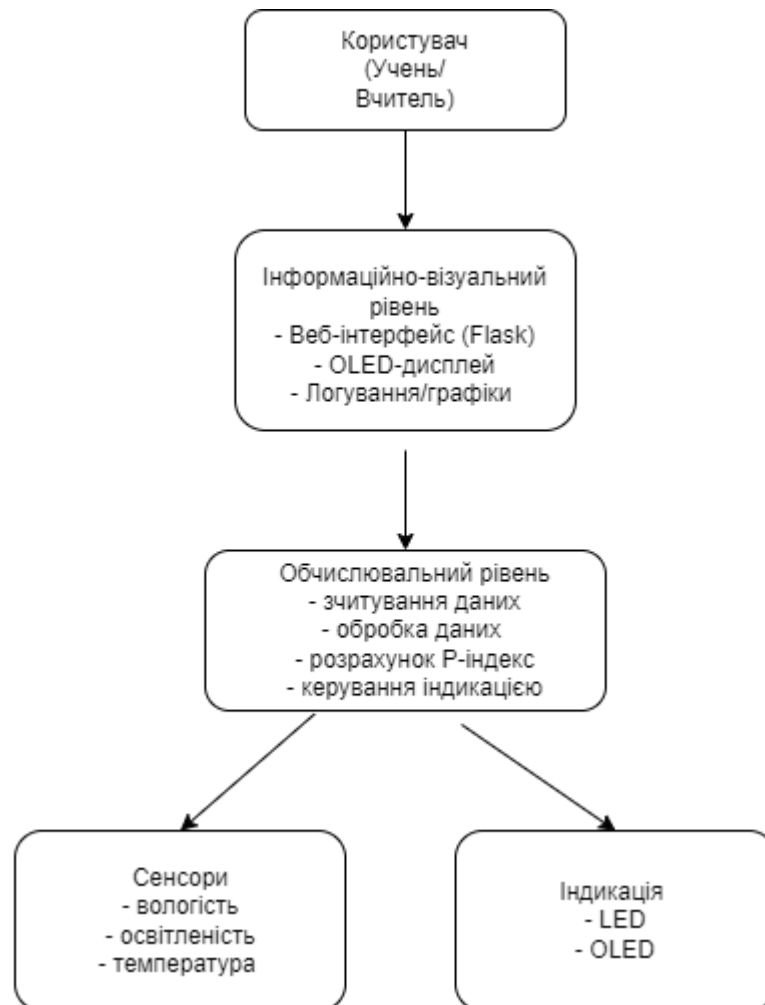


Рисунок 2.2 - Функціональна структурна схема.

Аналіз входів та виходів системи:

На основі функціональної структури виконано класифікацію всіх вхідних та вихідних сигналів.

Вхідні сигнали:

- DHT22 – цифровий сигнал температури та вологості;
- BH1750 – цифровий сигнал освітленості (I2C);
- Кнопка запуску – дискретний сигнал (високий/низький рівень логіки).

Вихідні сигнали:

- OLED-дисплей – текстовий вивід (температура, вологість, освітлення, індекс фотосинтезу);
- WS2812B LED – кольорова індикація рівня фотосинтезу (червоний, жовтий, зелений).

– Комунікаційні інтерфейси:

- HTTP/JSON – передача даних у веб-браузер;
- HTTPS/API – обмін з Google Таблицями через REST API.

– Усі сигнали згруповано за типами інтерфейсів: GPIO (цифрові входи/виходи), I²C (сенсори та дисплей), PWM (керування світлодіодами), USB/Wi-Fi (мережеві протоколи).

Основним обчислювальним ядром системи обрано Raspberry Pi 4 Model B, що забезпечує достатню продуктивність для одночасного зчитування сенсорів, роботи веб-сервера, обробки логіки та взаємодії з API. Застосування саме Raspberry Pi, а не мікроконтролера (ESP32, Arduino), обґрунтоване потребою у веб-інтерфейсі, підтримці Python-бібліотек, Google API та мультизадачності.

Інші компоненти були підібрані за критеріями сумісності, доступності, простоти інтеграції та надійності:

- DHT22 – точніший та стабільніший ніж DHT11;
- BH1750 – I2C-датчик освітлення з високою чутливістю;

- OLED 0.96” – енергоефективний дисплей для виводу значень;
- WS2812 – інформативна індикація для учнів (кольорова шкала станів).

Розрахунок живлення: Орієнтовне енергоспоживання системи:

Таблиця 2.4 – Розрахунок живлення

Компонент	Струм (макс.)	Напруга	Потужність
Raspberry Pi 4	2.5 A	5 V	12.5 W
DHT22	3 mA	5 V	0.015 W
BH1750	0.1 mA	3.3 V	≈0
OLED-дисплей	30 mA	3.3 V	0.1 W
WS2812 (8 світлод.)	до 240 mA	5 V	1.2 W
Усього (з запасом)	≈3 A	5 V	до 15 W

Блок живлення: 5В 3А (через USB-C), із захистом від перенапруги та перегріву. У разі автономної експлуатації можлива заміна на powerbank або акумуляторну систему з конвертором напруги.

Таблиця 2.5 - Компоненти

Назва	Кількість	Призначення
Raspberry Pi 4 (4GB)	1	Центральний процесор
DHT22	1	Зчитування температури і вологості
BH1750	1	Зчитування рівня освітлення
OLED дисплей 0.96" (I ² C)	1	Текстова візуалізація
WS2812B (8 LED)	1	Кольорова індикація
Кнопка керування	1	Запуск / зупинка процесу
Провідники, резистори	–	Макетування
Живлення (5V 3A)	1	Стабільне електроживлення системи

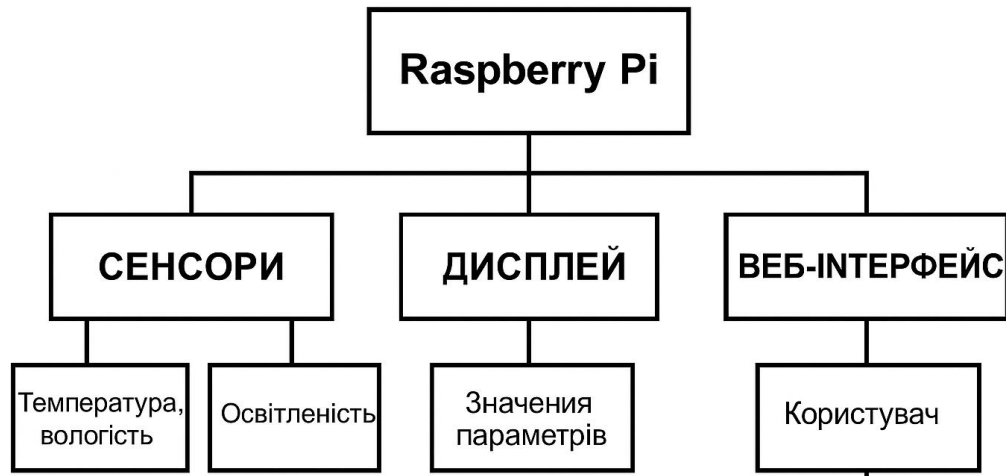


Рисунок 2.3 - Схема об'єкту

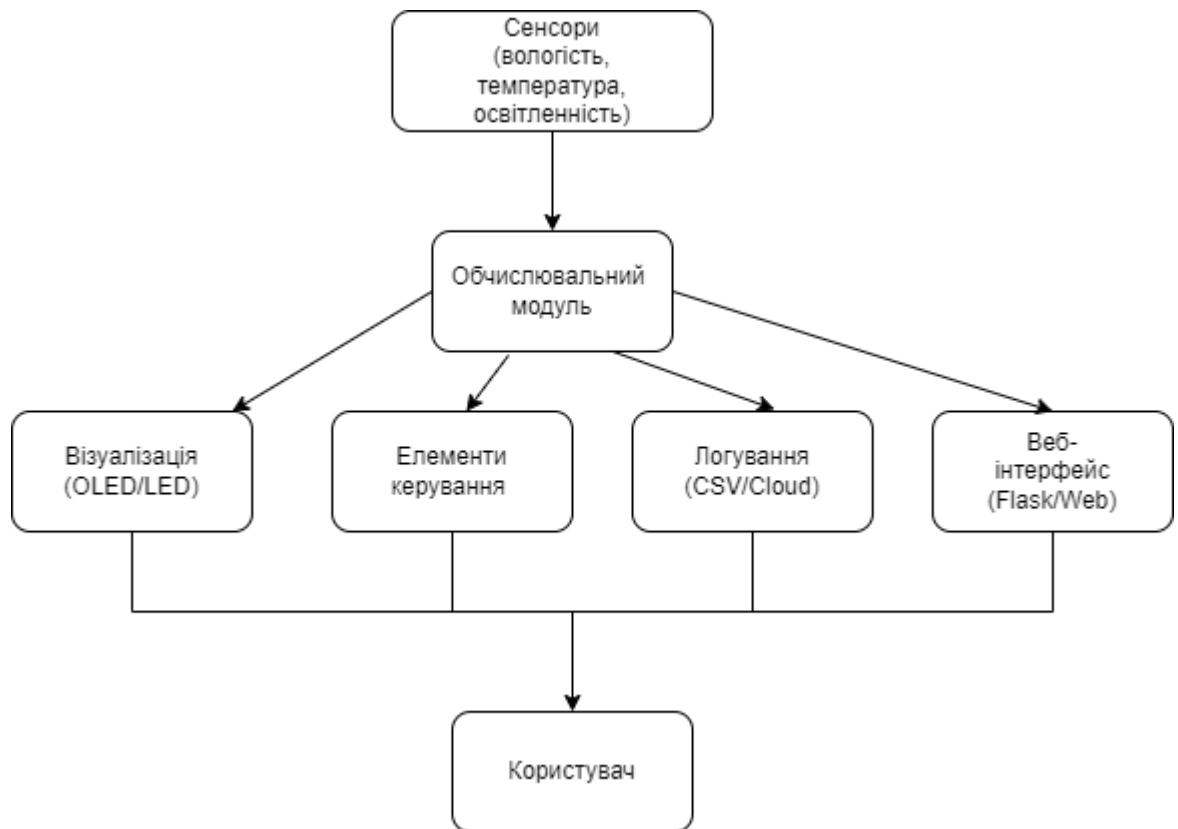


Рисунок 2.4 - Схема структурного обладнання

Ці схеми дозволяють візуально уявити, як реалізовано інформаційні зв'язки між сенсорами, інтерфейсами та логікою керування в системі.

2.3 Розробка програмного забезпечення

Програмне забезпечення розроблено для функціонального блоку моніторингу, візуалізації та адаптивного керування параметрами симуляції фотосинтезу в навчальній IoT-системі на базі Raspberry Pi 4B. Основним призначенням є забезпечення автоматизованого зчитування даних з фізичних сенсорів, обробка та подання результатів у зрозумілому вигляді для користувача (учня або вчителя), а також динамічна реакція на зміни навколишніх умов шляхом керування актюаторами (світлодіоди WS2812B).

Програма реалізує логіку взаємодії між апаратними компонентами системи, відображає поточні значення параметрів (світло, температура, вологість, CO₂, вологість ґрунту) на OLED-дисплеї та створює ефекти, які візуально демонструють ефективність або зміну умов фотосинтезу. Програму розраховано на використання в освітньому середовищі (Монтессорі-школа, природничі кабінети), де вона виконує функції інструменту для практичного експерименту.

2.3.1 Обґрунтування технічних характеристик

Програмне забезпечення симулятора фотосинтезу розроблено як інтегрована частина IoT-системи, що моделює природні процеси в освітньому

середовищі. Технічні характеристики програми визначаються вимогами до точного моніторингу фізичних параметрів, візуалізації даних та реакції на зміни середовища в режимі реального часу. З метою досягнення стабільної роботи з різномірними сенсорами, ефективної обробки інформації та зручного виводу для користувача, програмне забезпечення повинно відповідати ряду апаратних і програмних вимог, які обґрунтовуються нижче.

Програмне забезпечення симулятора виконує кілька основних функцій, необхідних для повноцінної роботи системи. Серед них — ініціалізація та зчитування даних з сенсорів BH1750, BME280, MH-Z19B та MCP3008. Зібрані дані обробляються, зберігаються у форматі CSV та виводяться на OLED-дисплей SSD1306. На основі зчитаних параметрів відбувається керування світлодіодною стрічкою WS2812B, що використовується для імітації сонячного світла. Крім того, програмне забезпечення формує інтерфейс керування, що може реалізовуватись як через веб-додаток, так і за допомогою фізичних енкодерів (за наявності).

До вхідних даних належать значення, отримані з сенсорів: освітленість у люксах (BH1750), температура і вологість повітря (BME280), концентрація CO₂ в ppm (MH-Z19B) та вологість ґрунту у відсотках (через MCP3008).

Вихідними є: покази, що відображаються на OLED-дисплеї в реальному часі; записи у файл CSV, що здійснюються для логування результатів; а також сигнали на GPIO-порти для керування світлодіодною стрічкою відповідно до змін параметрів середовища.

В якості технічної основи обрано Raspberry Pi 4B. Цей одноплатний комп'ютер підтримує основні інтерфейси — I2C, SPI, UART, має достатню кількість GPIO-портів, а також забезпечує підключення до локальної мережі або Інтернету. Завдяки багатозадачності та підтримці Linux-середовища він здатен одночасно опрацьовувати декілька сенсорних каналів, керувати виводом даних та реалізовувати веб-інтерфейс.

У якості мови програмування використано Python 3. Це зручне рішення для роботи з Raspberry Pi, що має розвинену екосистему бібліотек. Для взаємодії з сенсорами та дисплеєм використано бібліотеки Adafruit та Luma.OLED, доступ до апаратних інтерфейсів забезпечується за допомогою `smbus2`, `serial` та `RPi.GPIO`.

Raspberry Pi 4B — потужна платформа, що дозволяє реалізувати повний стек — від сенсорного зчитування до виводу у браузер. Сенсор BH1750 обрано через високу точність та стабільність вимірювання освітленості в люксах, зручний інтерфейс I2C. BME280 поєднує в собі одразу три вимірювальні функції: температура, вологість та тиск, що оптимізує кількість підключень. Для вимірювання рівня CO₂ використано MH-Z19B — інфрачервоний сенсор з UART-інтерфейсом, здатний працювати у широкому діапазоні до 5000 ppm. MCP3008 дозволяє зчитувати аналогові значення, зокрема від сенсора вологості ґрунту, і конвертувати їх у цифрову форму.

OLED-дисплей SSD1306 з інтерфейсом I2C забезпечує компактний і чіткий вивід поточних параметрів. Адресовані світлодіоди WS2812B реалізують кольорову індикацію стану освітлення, імітуючи зміну спектру протягом доби.

Допоміжні компоненти, зокрема макетна плата (breadboard), блок живлення 5V, резистори, конденсатори та логічні перетворювачі рівнів, забезпечують стабільну та безпечну роботу всієї системи.

2.3.2 Опис розробленої програми

Програма написана мовою Python 3 [12] та реалізує повний цикл збору, обробки, збереження і візуалізації даних про параметри довкілля, що впливають на фотосинтез. Основна логіка реалізована у вигляді автономного скрипта, який

ініціалізує сенсори, зчитує дані в циклі, виводить їх на OLED-дисплей та виконує логічне керування світлодіодами.

Для розробки програмного забезпечення використовується Python 3 — високорівнева мова з великою кількістю готових бібліотек, що ідеально підходить для роботи з сенсорами, дисплеями та GPIO-портами Raspberry Pi. Її популярність, активна спільнота та підтримка на платформі Raspberry Pi роблять Python оптимальним вибором для реалізації освітнього симулятора.

До складу програмного середовища входять такі ключові бібліотеки:

- `smbus2` — забезпечує обмін даними по шині I2C.
- `serial` — відповідає за UART-зв'язок, зокрема з сенсором CO₂ (MH-Z19B).
- `RPi.GPIO` або `gpiozero` — дозволяє керувати цифровими входами/виходами (GPIO).
- `adafruit_bh1750` та `adafruit_bme280` — офіційні бібліотеки Adafruit для роботи з відповідними сенсорами освітленості та кліматичних параметрів.
- `luma.oled` разом із `Pillow` — формують текстову та графічну інформацію для виводу на OLED-дисплей SSD1306.
- `adafruit-circuitpython-neopixel` — використовується для керування адресованими RGB-світлодіодами WS2812B.

Такий набір бібліотек забезпечує гнучкість, модульність та масштабованість системи. Окремі модулі можна легко оновлювати або замінювати без необхідності переробки всієї програми. Це значно спрощує супровід, адаптацію до нових задач та технічне обслуговування.

Програмна логіка побудована за класичним принципом поділу на етапи:

- ініціалізація всіх апаратних модулів та сенсорів при старті програми;
- вхід у основний цикл виконання, де регулярно зчитуються значення з усіх сенсорів;

- аналіз отриманих даних: перевірка виходу за допустимі межі, розрахунок похідних показників;
- виведення інформації на OLED-дисплей у реальному часі, а також збереження результатів до CSV-файлу для подальшого аналізу;
- керування світлодіодною стрічкою WS2812B відповідно до поточного рівня освітлення або заданого сценарію (імітація добового циклу);
- завершення роботи або повторення циклу, залежно від режиму роботи чи дій користувача.

Програма підтримує автономну роботу та здатна взаємодіяти з іншими додатками або сервісами через HTTP API (наприклад, для веб-інтерфейсу) або MQTT-протокол (для інтеграції з IoT-платформами).

Запуск основного скрипта може бути організований двома способами:

1. автоматично — через системну службу `systemd` під час старту Raspberry Pi;
2. вручну — з терміналу або в середовищі розробника.

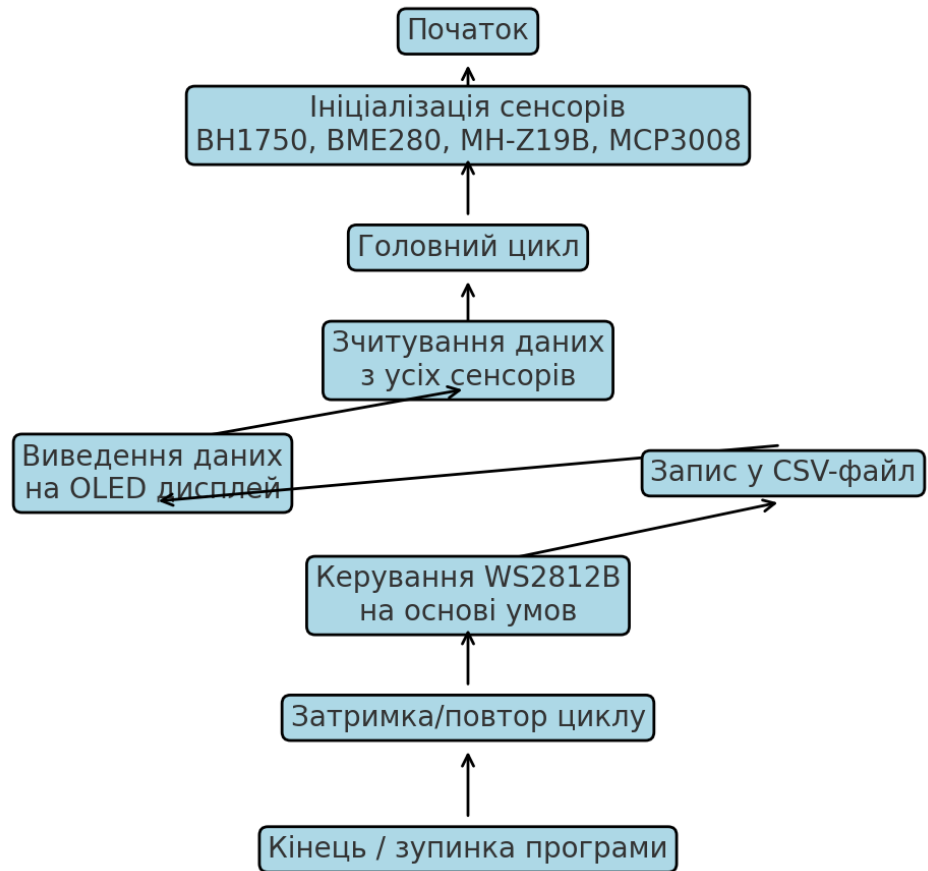


Рисунок 2.5 - Блок-схема алгоритму

Таблиця 2.6 – Вхідні та вихідні дані.

Назва	Тип	Джерело/Призначення	GPIO / Інтерфейс	Коментар
lux	float	BH1750	I2C (0x23)	Освітленість
temp	float	BME280	I2C (0x76)	Температура °C
hum	float	BME280	I2C (0x76)	Вологість повітря
co2	int	MH-Z19B	UART	CO ₂ ppm
soil	int	MCP3008	SPI (CH0)	Вологість ґрунту (ADC)

display_text	str	OLED SSD1306	I2C (0x3C)	Строка з параметрами
led_pwm	int	WS2812B	GPIO18	Яскравість/колір

Висновок

У ході виконання кваліфікаційної роботи було розроблено та реалізовано IoT-систему інтерактивного симулятора фотосинтезу для використання у навчальному процесі, зокрема в умовах Монтессорі-шкіл. Система відповідає всім поставленим технічним вимогам: забезпечує зчитування параметрів середовища (температура, вологість, освітлення), візуалізацію процесу фотосинтезу за допомогою LED-індикації та текстового дисплея, а також реєстрацію даних в локальні файли CSV і Google Таблиці.

Досягнуто основну мету роботи – створено інтерактивний симулятор, що дозволяє учням у доступній формі досліджувати вплив зовнішніх факторів на фотосинтез. Усі поставлені завдання – від вибору апаратної платформи до побудови веб-інтерфейсу – були вирішені на високому технічному рівні.

Практична цінність роботи полягає у можливості використання розробленої системи не лише у навчальних закладах, а й для демонстрацій у музеях, наукових виставках та при проведенні STEM-олімпіад. Завдяки відкритій архітектурі рішення може бути легко адаптоване для моделювання інших природничих процесів або використане як шаблон для проєктів з кіберфізичних систем.

Доцільним напрямком подальшої роботи є:

- впровадження додаткових сенсорів (CO₂, ґрунтова вологість тощо);
- створення мобільного додатку для взаємодії з симулятором;
- побудова повноцінної бази даних для довготривалого аналізу експериментів;
- інтеграція з системами дистанційного навчання або LMS.

Перелік посилань

1. Вплив температури на рослини. *StudFiles*.
URL: <https://studfile.net/preview/9227951/>.
2. Екологія фотосинтезу. Від чого залежить урожай. *Agroexpert*.
URL: <https://agroexpert.ua/ekologia-fotosintezu-vid-cogo-zalezit-urozai/>.
3. Похибка – Вікіпедія. *Вікіпедія*.
URL: https://uk.wikipedia.org/wiki/Похибка_вимірювання.
4. Метод Монтесорі від А до Я. *Montessori UA*.
URL: https://montessoriua.com/ua/metod_montessori/.
5. Фотосинтез. *Вікіпедія*. URL: <https://uk.wikipedia.org/wiki/Фотосинтез>.
6. Фотосинтез. Роль зелених рослин у біологічному житті. *Освіта.UA*.
URL: <https://osvita.ua/vnz/reports/biolog/27288/>.
7. Adafruit Learning System. *Adafruit Learning System*.
URL: <https://learn.adafruit.com/>.
8. Flask Documentation. *Welcome to Flask – Flask Documentation (3.1.x)*.
URL: <https://flask.palletsprojects.com/en/stable/>.
9. GitHub. *GitHub*. URL: <https://github.com/>.
10. Google Sheets. *Google for Developers*.
URL: <https://developers.google.com/workspace/sheets>.
11. Hackster.io. *Hackster.io - The community dedicated to learning hardware*. URL: <https://www.hackster.io/>.
12. Python. *Python.org*. URL: <https://www.python.org/>.
13. Raspberry Documentation. *Raspberry Pi*.
URL: <https://www.raspberrypi.com/documentation/>.
14. Raspberry Pi. *Raspberry Pi*. URL: <https://www.raspberrypi.com/>.

Додаток А

Текст програми
налаштування Raspberry Pi та сенсорів

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
RASPBERRY PI ТА СЕНСОРІВ

Текст програми

Листів 15

АНОТАЦІЯ

Дана програма містить у собі програмний код для автономної IoT-системи симулятора фотосинтезу, призначеної для використання в освітньому середовищі, зокрема в школах, що працюють за методикою Монтесорі.

Програма забезпечує зчитування параметрів навколишнього середовища (освітленість, температура, вологість, концентрація CO₂, вологість ґрунту) за допомогою відповідних сенсорів, відображення даних на OLED-дисплеї, збереження результатів у файл та адаптивне керування світлодіодною стрічкою WS2812B для візуалізації процесу фотосинтезу.

Програма написана мовою Python, відлагоджена на платформі Raspberry Pi 4B з використанням бібліотек smbus2, serial, luma.oled, adafruit-circuitpython та інших, і призначена для запуску на операційній системі Raspberry Pi OS (Linux, Debian-based).

ЗМІСТ

1. BH1750 (освітленість)
2. BME280 (температура, вологість, тиск)
3. MH-Z19B (CO₂)
4. main.py
5. Запис в Google-таблицю
6. Візуалізація даних в таблицю у реальному часі
7. Налаштування Raspberry Pi
8. Flask
9. HTML шаблон для Flask

BH1750 (освітленість)

```
import board
import busio
import adafruit_bh1750

i2c = busio.I2C(board.SCL, board.SDA)
sensor = adafruit_bh1750.BH1750(i2c)

def read_light():
    lux = sensor.lux
    print(f"Освітленість: {lux:.2f} люкс")
    return lux

if __name__ == "__main__":
    read_light()
```

BME280 (температура, вологість, тиск)

```
import board

import busio

import adafruit_bme280

i2c = busio.I2C(board.SCL, board.SDA)

sensor = adafruit_bme280.Adafruit_BME280_I2C(i2c)

def read_bme280():

    temp = sensor.temperature

    humidity = sensor.humidity

    pressure = sensor.pressure

    print(f"Температура: {temp:.2f} °C")

    print(f"Вологість: {humidity:.2f} %")

    print(f"Тиск: {pressure:.2f} гПа")

    return temp, humidity, pressure

if __name__ == "__main__":

    read_bme280()
```

MH-Z19B (CO₂)

```
import serial
import time

ser = serial.Serial("/dev/serial0", baudrate=9600, timeout=1)

def read_co2():
    ser.write(b"\xFF\x01\x86\x00\x00\x00\x00\x00\x79")
    response = ser.read(9)
    if len(response) == 9 and response[0] == 0xFF and response[1] == 0x86:
        co2 = response[2]*256 + response[3]
        print(f"CO2: {co2} ppm")
        return co2
    else:
        print("Помилка читання CO2")
        return None

if __name__ == "__main__":
    read_co2()
```

main.py

```
from BH1750 import read_light
from BME280 import read_bme280
from MHZ19B import read_co2
import time
import csv
from datetime import datetime

from luma.core.interface.serial import i2c
from luma.oled.device import ssd1306
from PIL import Image, ImageDraw, ImageFont

# OLED 128x64 I2C
serial = i2c(port=1, address=0x3C)
device = ssd1306(serial)

# CSV ініціалізація
csv_file = "photosynthesis_log.csv"
with open(csv_file, mode='a', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(["Дата", "Час", "Lux", "Температура", "Вологість", "Тиск",
"CO2"])
```

```
def display_text(lines):  
    image = Image.new("1", (device.width, device.height))  
    draw = ImageDraw.Draw(image)  
    font = ImageFont.load_default()  
  
    y = 0  
    for line in lines:  
        draw.text((0, y), line, font=font, fill=255)  
        y += 10  
  
    device.display(image)  
  
def log_to_csv(data):  
    with open(csv_file, mode='a', newline="") as file:  
        writer = csv.writer(file)  
        writer.writerow(data)  
  
def main():  
    while True:  
        print("\n===== Зчитування даних =====")  
        timestamp = datetime.now()  
        date_str = timestamp.strftime("%Y-%m-%d")  
        time_str = timestamp.strftime("%H:%M:%S")
```

```
try:
    lux = read_light()
except Exception as e:
```

```
    lux = ""
    print(f"Помилка BH1750: {e}")
```

```
try:
    temp, humidity, pressure = read_bme280()
except Exception as e:
```

```
    temp, humidity, pressure = "", "", ""
    print(f"Помилка BME280: {e}")
```

```
try:
    co2 = read_co2()
except Exception as e:
```

```
    co2 = ""
    print(f"Помилка MH-Z19B: {e}")
```

```
# OLED вивід
```

```
lines = [
    f"Світло: {lux} lx" if lux != "" else "Світло: ---",
    f"T: {temp}C RH: {humidity}%" if temp != "" else "T/RH: ---",
    f"CO2: {co2} ppm" if co2 != "" else "CO2: ---"
]
```

```
display_text(lines)

# Логування
log_to_csv([date_str, time_str, lux, temp, humidity, pressure, co2])
time.sleep(3)

if __name__ == "__main__":
    main()
```

Запис в Google-таблицю

```
import gspread

from oauth2client.service_account import ServiceAccountCredentials

from datetime import datetime

# Авторизація через Google Service Account

scope = ["https://spreadsheets.google.com/feeds",
         "https://www.googleapis.com/auth/drive"]

creds = ServiceAccountCredentials.from_json_keyfile_name("photosynthesis-
creds.json", scope)

client = gspread.authorize(creds)

# Відкриття першого аркуша таблиці

sheet = client.open("Photosynthesis Simulation Log").sheet1

# Функція для додавання нового рядка

def append_data(lux, temp, hum, pres, co2):

    now = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    row = [now, lux, temp, hum, pres, co2]

    sheet.append_row(row)
```

Візуалізація даних в таблицю у реальному часі

```
import pandas as pd

import matplotlib.pyplot as plt

import matplotlib.animation as animation

from datetime import datetime

CSV_FILE = "photosynthesis_log.csv"

def animate(i):
    try:
        df = pd.read_csv(CSV_FILE)
        df["Час"] = pd.to_datetime(df["Дата"] + " " + df["Час"])
        df = df.tail(50) # Показати останні 50 записів

        ax1.clear()

        ax2.clear()

        ax3.clear()

        ax4.clear()

        ax1.plot(df["Час"], df["Lux"], label="Освітленість (lux)")
        ax2.plot(df["Час"], df["Температура"], label="Температура (°C)",
                 color='orange')
        ax3.plot(df["Час"], df["Вологість"], label="Вологість (%)", color='green')
```

```
ax4.plot(df["Час"], df["CO2"], label="CO2 (ppm)", color='red')
```

```
ax1.set_ylabel("Lux")
```

```
ax2.set_ylabel("Температура °C")
```

```
ax3.set_ylabel("Вологість %")
```

```
ax4.set_ylabel("CO2 ppm")
```

```
for ax in [ax1, ax2, ax3, ax4]:
```

```
    ax.legend(loc='upper left')
```

```
    ax.tick_params(axis='x', rotation=45)
```

```
except Exception as e:
```

```
    print(f"Помилка: {e}")
```

```
fig, (ax1, ax2, ax3, ax4) = plt.subplots(4, 1, figsize=(10, 8))
```

```
ani = animation.FuncAnimation(fig, animate, interval=3000)
```

```
plt.tight_layout()
```

```
plt.show()
```

Налаштування Raspberry Pi

```
# ==== На microSD (boot розділ) ====
```

```
# 1. Увімкнення SSH
```

```
touch ssh
```

```
# 2. Wi-Fi (wpa_supplicant.conf)
```

```
# Створити файл з наступним вмістом:
```

```
country=UA
```

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
```

```
update_config=1
```

```
network={
```

```
    ssid="Назва_твого_WiFi"
```

```
    psk="Пароль_до_WiFi"
```

```
}
```

```
# ==== Після запуску Raspberry Pi ====
```

```
# 3. Підключення по SSH
```

```
ssh pi@raspberrypi.local
```

```
# або
```

```
ssh pi@192.168.X.X
```

```
# пароль: raspberry
```

```
# 4. Початкове налаштування
```

```
sudo raspi-config
```

```
# 5. Оновлення системи
```

```
sudo apt update && sudo apt upgrade -y
```

```
# 6. Встановлення пакетів
```

```
sudo apt install python3-pip git i2c-tools python3-smbus
```

```
pip3 install flask gspread oauth2client
```

```
# 7. Автозапуск скриптів
```

```
crontab -e
```

```
# Додати рядок:
```

```
@reboot python3 /home/pi/your_script.py &
```

```
# 8. Передача файлів
```

```
# З ноутбука на Pi
```

```
scp file.txt pi@raspberrypi.local:/home/pi/
```

```
# З Pi на ноутбук
```

```
scp pi@raspberrypi.local:/home/pi/log.csv ./
```

Flask

```
from flask import Flask, render_template
import pandas as pd

app = Flask(__name__)

@app.route("/")
def index():
    try:
        df = pd.read_csv("photosynthesis_log.csv")
        last = df.tail(1).to_dict(orient="records")[0]
    except Exception:
        last = {"Lux": "—", "Температура": "—", "Вологість": "—", "Тиск": "—",
"CO2": "—"}

    return render_template("index.html", data=last)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

HTML шаблон для Flask

```
<!doctype html>
<html>
<head>
  <title>Моніторинг фотосинтезу</title>
  <meta charset="utf-8">
  <style>
    body { font-family: sans-serif; margin: 30px; }
    .param { margin-bottom: 10px; }
  </style>
</head>
<body>
  <h2>Поточні параметри</h2>
  <div class="param">Світло: {{ data["Lux"] }} lux</div>
  <div class="param">Температура: {{ data["Температура"] }} °C</div>
  <div class="param">Вологість: {{ data["Вологість"] }} %</div>
  <div class="param">Тиск: {{ data["Тиск"] }} гПа</div>
  <div class="param">CO2: {{ data["CO2"] }} ppm</div>
</body>
</html>
```

Додаток Б

Приклади результатів експериментів

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

Приклади результатів експериментів

Лист 1



Рисунок 1.

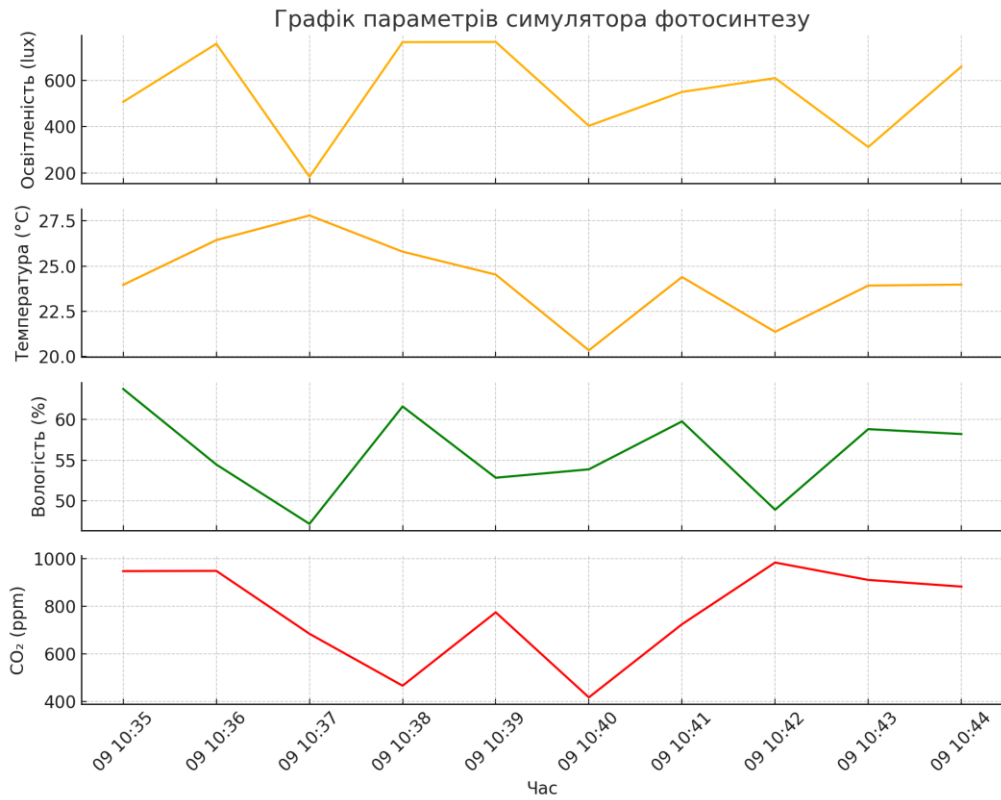


Рисунок. 2