

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
ВОЛОДИМИРА ДАЛЯ

Факультет інформаційних технологій та електроніки

Кафедра інформаційних технологій та програмування

Пояснювальна записка
до магістерської дипломної роботи

магістр

(освітньо-кваліфікаційний рівень)

на тему: Адаптивне керування освітленням на основі системи сенсорів та алгоритму оптимізації енергоспоживання

Виконав: студент 2 курсу, групи ЕЛ-24дм
171 «Електроніка»

(шифр і назва спеціальності)

Тагієв М.П.

(прізвище та ініціали)

Керівник Захожай О.І.

(прізвище та ініціали)

Рецензент Іванов В.Г.

(прізвище та ініціали)

Київ – 2025 року

Факультет інформаційних технологій та електроніки
Кафедра інформаційних технологій та програмування
Освітньо-кваліфікаційний рівень магістр
Спеціальність 171 «Електроніка»
(шифр і назва спеціальності)

ЗАВДАННЯ

на магістерську дипломну роботу студенту

(прізвище, ім'я, по батькові)

керівник роботи д.т.н ., Захожай Олег Ігорович,
(вчене звання, науковий ступінь, прізвище, ім'я, по батькові)

2. Строк подання студентом роботи: 18 грудня 2025р.

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

4.1 Вступ

4.2 Аналітичний огляд предметної області

4.3 Частина в якій висвітлити математичні методи, які будуть використовуватися для реалізації проекту.

4.5 Практична частина

4.6 Експериментальна частина

4.7 Висновки

4.6 Перелік використаних джерел

4.7 Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 05.11.2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Одержання завдання на виконання роботи	05.11.2025	<i>виконано</i>
2.	Укладання і погодження з керівником плану і етапів виконання роботи	08.11.2025	<i>виконано</i>
3.	Узагальнення даних літературних джерел	12.11.2025	<i>виконано</i>
4.	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху виконання завдання	18.11.2025	<i>виконано</i>
5.	Аналіз технічних засобів та існуючих систем	23.11.2025	<i>виконано</i>
6.	Реалізація практичної частини завдання	08.12.2025	<i>виконано</i>
7.	Укладання, оформлення та погодження пояснювальної записки з керівником	18.12.2025	<i>виконано</i>
8.	Надання пояснювальної записки на кафедру	18.12.2025	<i>виконано</i>
9.	Підготовка доповіді та презентації	22.12.2025	<i>виконано</i>

Студент _____ Тагієв М.П.
(підпис) (прізвище та ініціали)
Керівник роботи _____ Захожай О.І.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 97 сторінок, 10 таблиць, 18 рисунків.

Тема магістерської роботи: «Адаптивне керування освітленням на основі системи сенсорів та алгоритму оптимізації енергоспоживання».

Об'єктом дослідження є інтелектуальна система керування освітленням, що використовує сенсорні дані та адаптивні алгоритми для автоматичного регулювання яскравості.

Метою роботи є підвищення енергоефективності освітлювальних систем шляхом створення адаптивного алгоритму керування та побудови апаратно-програмного прототипу, здатного реагувати на зміну рівня природної освітленості та присутність користувача.

У процесі роботи проведено аналітичний огляд сучасних систем освітлення та принципів адаптивного керування, розроблено математичну модель процесу регулювання, створено структурну та апаратну схему системи на основі мікроконтролера Arduino, фотодатчика, датчика руху та потенціометра. Реалізовано програмний алгоритм оптимізації яскравості, що використовує інтерполяційну модель і механізм згладжування змін освітленості. Побудовано прототип системи та проведено експериментальні дослідження в симуляційному середовищі Wokwi, включаючи аналіз реакції на стабільну, поступово змінну та різко варіативну освітленість. Підтверджено, що адаптивний алгоритм забезпечує плавне керування яскравістю, відсутність стрибків PWM та зменшення енергоспоживання порівняно з традиційними методами керування.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ІТ-РІШЕНЬ	9
1.1 Огляд систем освітлення	9
1.2 Огляд існуючих рішень у сфері адаптивного управління освітленням	12
1.3 Порівняння систем на основі фіксованих правил та адаптивних алгоритмів	16
Висновки до розділу 1	17
РОЗДІЛ 2. МАТЕМАТИЧНІ МЕТОДИ ТА ПРОЄКТУВАННЯ АДАПТИВНОЇ СИСТЕМИ КЕРУВАННЯ ОСВІТЛЕННЯМ	19
2.1 Вибір математичного апарату та методів обробки даних	19
2.1.1 Теоретичні основи обраних математичних методів	20
2.1.2 Основні змінні, що впливають на роботу алгоритму	21
2.1.3 Інтеграція даних з різних сенсорів у межах алгоритму	22
2.2 Розробка алгоритму адаптивного керування освітленням	24
2.2.1 Загальний алгоритм адаптивного керування освітленням	25
2.2.2 Поетапний розбір логічних модулів алгоритму	27
2.3 Математична модель адаптивного керування освітленням	37
2.3.1 Обчислення середнього рівня освітленості	39
2.3.2 Функція визначення оптимального рівня освітлення	41
2.3.3 Обчислення необхідної компенсації штучним освітленням	42
2.3.4 Формування керуючого впливу на світильник (ШИМ-модель)	42
Висновок до розділу 2	44
РОЗДІЛ 3. РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	46
3.1. Вибір елементної бази та обґрунтування компонентів системи	46
3.2 Підключення компонентів та принцип роботи експериментальної схеми	58
3.2.1 Принципова електрична схема	61
3.3 Програмна реалізація алгоритму адаптивного освітлення	64
Висновки до розділу 3	70
РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ АДАПТИВНОГО ОСВІТЛЕННЯ	71
4.1 Дослідження роботи системи у початковому режимі	71
4.2 Реакція системи на зміну рівня природної освітленості	74
4.3 Поведінка системи при різких коливаннях рівня освітленості	81
4.4. Порівняльний аналіз енергоспоживання системи	83
Висновки до розділу 4	84
ВИСНОВОК	86
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	88
ДОДАТОК А	91
Додаток А Код програми керування адаптивним освітленням для Arduino	91
ДОДАТОК Б	95
Файл конфігурації симулятора Wokwi (diagram.json)	95

ВСТУП

У сучасних умовах стрімкого розвитку інформаційних технологій та інтелектуальних систем особливої актуальності набуває проблема раціонального використання енергоресурсів. Світові тенденції спрямовані на зменшення витрат електроенергії за рахунок автоматизації, інтелектуального керування та впровадження адаптивних систем контролю. З огляду на зростання навантаження на енергетичні мережі, підвищення тарифів та необхідність дотримання принципів сталого розвитку, питання оптимізації енергоспоживання стає одним з ключових як для промисловості, так і для приватних та комерційних будівель.

Проблема енергоефективності має особливе значення для України. Наслідки воєнних дій, пошкодження критичної інфраструктури, періодичні обмеження електроспоживання та підвищена вартість енергії формують нагальну потребу у впровадженні систем, здатних зменшити споживання електроенергії без втрати функціональності та комфорту. У таких умовах адаптивні системи керування освітленням, що базуються на сенсорних даних та алгоритмах оптимізації, набувають стратегічного значення.

Освітлення є одним із найбільших споживачів електроенергії: у житлових та комерційних будівлях на нього припадає до 30–40 % загального споживання. Традиційні системи освітлення, які працюють за принципом «увімкнути/вимкнути» або використовують статичні режими, не враховують динамічні зміни освітленості, присутність користувачів чи зовнішні умови. Вони споживають електроенергію незалежно від реальної потреби, мають низьку адаптивність та не забезпечують раціональне управління ресурсами.

Адаптивне керування освітленням, на відміну від класичних систем, передбачає використання інтелектуальних алгоритмів і сенсорних технологій для аналізу стану навколишнього середовища та автоматичної зміни рівня

освітленості. До таких систем входять датчики освітленості, руху та присутності, мікроконтролери, мережеві модулі та програмні методи опрацювання даних. На їх основі створюються системи, здатні в реальному часі реагувати на зміну умов, мінімізувати енергоспоживання та забезпечувати оптимальний комфорт користувачів.

Тема адаптивного керування освітленням є актуальною також з точки зору розвитку інтелектуальних технологій («розумний будинок», «розумне місто», IoT-платформи). Вона поєднує елементи теорії керування, аналізу сенсорних даних, оптимізації, побудови математичних моделей та розробки програмно-апаратних комплексів. Це робить дослідження міждисциплінарним та таким, що відображає сучасні напрями розвитку інженерії та інформаційних технологій.

Об'єкт дослідження - процес адаптивного керування освітленням на основі сенсорних даних у приміщеннях.

Предмет дослідження - математичні моделі, алгоритми оптимізації та інформаційно-технічні засоби реалізації систем адаптивного освітлення.

Мета роботи полягає у підвищенні ефективності освітлювальних систем шляхом розробки алгоритму адаптивного керування та прототипу сенсорної системи, що забезпечує автоматичне регулювання рівня освітленості відповідно до зовнішніх умов і потреб користувачів.

Для досягнення мети необхідно розв'язати такі **завдання**:

1. Проаналізувати предметну область, сучасні системи освітлення та інтелектуальні IT-рішення у цій сфері.
2. Обґрунтувати вибір оптимального математичного, інформаційного, технічного та програмного забезпечення системи.

3. Розробити інформаційну та математичну модель адаптивного керування освітленням.
4. Розробити алгоритм оптимізації яскравості освітлення на основі даних сенсорів та моделей користувацьких вподобань.
5. Реалізувати пілотну версію системи освітлення на базі мікроконтролера та сенсорних модулів.
6. Розробити програмне забезпечення для контролю та моніторингу роботи системи.
7. Провести тестування системи на реальних або симульованих даних, оцінити її енергоефективність та стабільність роботи.

Наукова новизна роботи полягає у розробці гібридного алгоритму адаптивного керування, що поєднує логічне дерево рішень із функцією оптимізації, яка враховує як природне освітлення, так і користувацькі вподобання. У роботі запропоновано інформаційну модель сенсорної системи та структуру прототипу інтелектуального освітлення на базі IoT-технологій.

Практичне значення роботи полягає у можливості застосування розробленої системи в житлових, офісних і промислових приміщеннях для зниження енергоспоживання та підвищення комфорту користувачів. Представлений прототип може бути використаний як основа для створення масштабованих рішень у сфері «розумного будинку» та систем автоматизованого керування енергоспоживанням.

Результати дослідження підтверджують перспективність впровадження інтелектуальних адаптивних систем освітлення як важливого інструмента раціонального використання енергоресурсів, особливо в умовах енергетичної нестабільності України.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ІТ-РІШЕНЬ

1.1 Огляд систем освітлення

Освітлювальні системи є важливою складовою інфраструктури сучасних будівель, промислових об'єктів і побутових приміщень. Вони забезпечують комфортні умови для праці та проживання, проте їхній внесок у загальне енергоспоживання будівельний сектор традиційно залишається високим. Найбільш значними недоліками традиційних систем освітлення є низька енергоефективність, відсутність адаптації до змін природної освітленості та залежність роботи освітлення від ручного керування з боку користувача

У сучасній практиці електротехнічних рішень системи освітлення класифікують за рівнем автоматизації та здатністю реагувати на зовнішні умови:

1. Системи з ручним керуванням. Вони орієнтовані на безпосередню взаємодію користувача з вимикачем. Такі системи не здатні враховувати зміну зовнішньої освітленості, що призводить до надмірного споживання енергії та зниження терміну служби ламп через роботу на повній потужності.
2. Системи з таймерами або розкладами. У таких рішеннях світло вмикається та вимикається за заданим часом. Вони частково автоматизовані, однак залишаються негнучкими: графік не враховує зміни природного освітлення, сезонні коливання чи варіації у поведінці користувачів.
3. Сенсорні системи на основі датчиків руху. PIR-датчики дозволяють автоматично вмикати або вимикати освітлення в залежності від присутності людини. Такі системи покращують енергоефективність, проте їх функціональність обмежена: вони не регулюють яскравість та можуть давати хибні спрацьовування.

4. Інтелектуальні системи освітлення (Smart Lighting). Найбільш прогресивний клас рішень, що використовує сенсори освітленості, руху, присутності, а також мережеві модулі й алгоритми оптимізації. Такі системи здатні автоматично адаптувати інтенсивність освітлення до реальних умов, забезпечуючи високу енергоефективність і комфорт. Вони інтегруються з IoT-платформами, системами «розумного будинку».

Типова структура сучасної системи освітлення включає: LED-джерела світла, сенсори освітленості та руху, контролери (мікроконтролери, ПЛК, IoT-пристрої), модулі комунікації (Wi-Fi, ZigBee, Bluetooth Mesh, KNX), програмні інтерфейси та алгоритми оптимізації.

Головний напрямом розвитку сучасних систем освітлення полягає у переході від жорстко заданих сценаріїв («ввімкнути/вимкнути») до адаптивних моделей керування, що використовують аналіз даних і дозволяють прогнозувати потребу в освітленні.

У таблиці 1.1 наведено порівняння різних типів систем освітлення за ключовими параметрами - автоматизацією, ефективністю, масштабованістю та рівнем комфорту.

Таблиця 1.1 - Порівняльна характеристика систем освітлення

Критерій	Традиційні системи	Системи з фіксованими правилами	IoT-системи (ESP32, Arduino)	Розумні системи
Автоматизація	Відсутня	Часткова	Часткова/висока	Повна
Енергоефективність	Низька	Середня	Середня/висока	Висока
Врахування вподобань	Немає	Обмежене	Частково	Частково
Екологічний вплив	Високий	Середній	Низький	Низький
Вартість впровадження	Низька	Низька/середня	Низька	Висока
Гнучкість налаштування	Низька	Середня	Висока	Висока
Масштабованість	Низька	Середня	Висока	Висока
Стійкість до змін умов	Низька	Середня	Середня/висока	Висока
Інтеграція з іншими системами	Відсутня	Часткова	Є (API / Wi-Fi / MQTT)	Є (IoT, BMS, DALI, KNX)
Комфорт користувачів	Низький	Середній	Середній/високий	Високий

Аналіз таблиці свідчить, що традиційні системи є найменш ефективними та не враховують зовнішніх факторів. Системи з фіксованими правилами демонструють кращі результати, проте їх гнучкість обмежена. У свою чергу, IoT-системи на базі мікроконтролерів дають змогу реалізувати адаптивні алгоритми керування освітленням, поєднуючи низьку вартість і високу масштабованість.

Інтелектуальні системи освітлення забезпечують максимальний рівень автоматизації та комфорт користувачів, однак їх реалізація передбачає значні

витрати на обладнання та налаштування. Тому для більшості практичних застосувань оптимальним є гібридний підхід на основі IoT-архітектури, який дозволяє реалізувати адаптивне керування освітленням з мінімальними витратами.

Таким чином, аналіз підтверджує доцільність застосування адаптивних IoT-рішень як найбільш ефективної концепції для побудови сучасних систем керування освітленням.

1.2 Огляд існуючих рішень у сфері адаптивного управління освітленням

Сучасні системи адаптивного керування освітленням розвиваються у напрямку підвищення енергоефективності та комфорту користувачів. Існуючі рішення можна умовно поділити на три основні групи: комерційні системи «розумного» освітлення, промислові системи автоматизації будівель та IoT-рішення на основі мікроконтролерів. Кожний підхід має характерні особливості, що визначають його можливості та сферу застосування.

Комерційні системи «розумного» освітлення до цього класу належать Philips Hue, Xiaomi Yeelight, LIFX та подібні системи. Вони орієнтовані на кінцевого користувача пропонуючи:

- регулювання яскравості та кольорової температури
- автоматичне увімкнення та вимкнення за сценаріями
- інтеграцію зі смартфонами та голосовими асистентами
- базову адаптацію до умов освітленості

Основними перевагами таких систем є простота встановлення та наявність розвинених екосистем. Проте вони використовують закриту архітектуру, що обмежує можливість створення власних алгоритмів, модифікації системи або інтеграції з нестандартними сенсорами.

Промислові системи на кшталт DALI, Helvar, Lutron та інших призначені для великих об'єктів: торговельних центрів, офісів, підприємств. Вони забезпечують:

- групування світильників та зональне керування
- підтримку складних профілів освітленості
- адаптацію до природного світла (daylight harvesting)
- централізований моніторинг та діагностику

На рисунку 1.1 наведено типову архітектуру DALI-системи, яка містить джерело живлення шини, сенсори, контролери та LED-драйвери, об'єднані цифровою шиною DALI. Це дозволяє виконувати гнучке групування світильників та їх адаптивне керування.

DALI SYSTEM

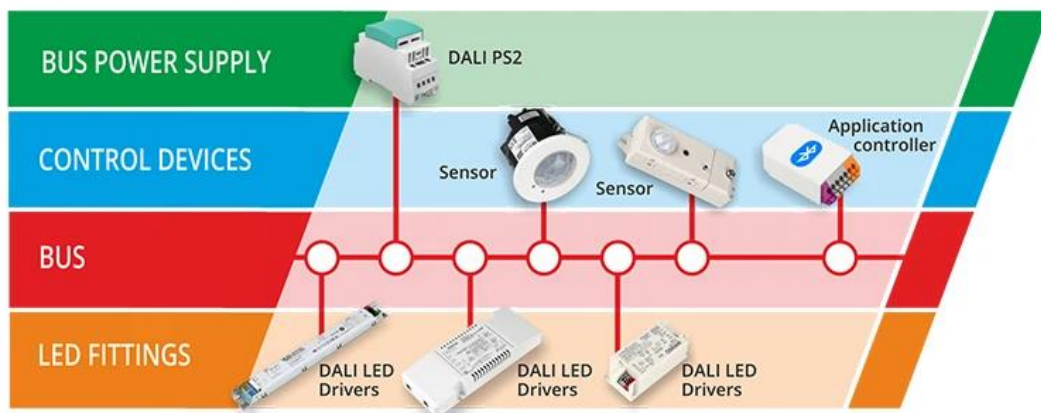


Рисунок 1.1 - Структурна схема DALI-системи керування освітленням

Промислові системи вирізняються високою надійністю та масштабованістю, проте вони мають складну і дорогую інфраструктуру, що обмежує їх використання у побутових або малих комерційних приміщеннях.

IoT-рішення на базі ESP32, Arduino, Raspberry Pi та інших контролерів спрямовані на створення відкритих і гнучких систем. Вони можуть включати

різноманітні датчики освітленості, датчики руху, мережеві модулі Wi-Fi або BLE та програмно реалізовані алгоритми адаптивного керування.

Такі рішення дозволяють:

- створювати власні алгоритми на основі даних
- гнучко налаштовувати логіку роботи світильників
- збирати й аналізувати дані у реальному часі
- інтегрувати систему з хмарними сервісами або локальними панелями керування

На рисунку 1.2 наведено узагальнену архітектуру IoT-системи керування освітленням, що включає мікроконтролер, сенсори, шлюз та модулі передачі даних.

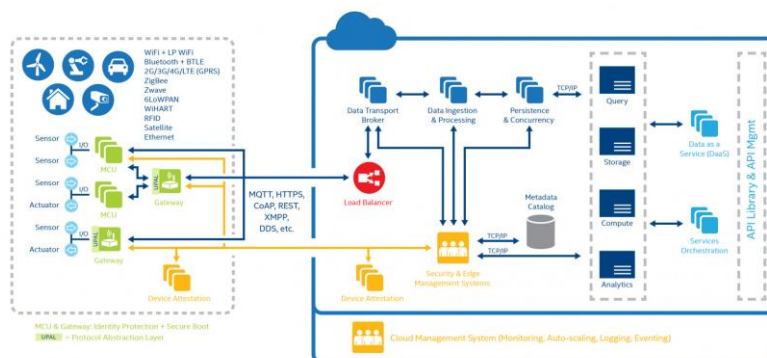


Рисунок 1.2 - Структурна схема IoT-системи освітлення

Основною перевагою IoT-підходу є максимальна гнучкість та можливість побудови нестандартних адаптивних алгоритмів. Недоліком є необхідність самостійної розробки програмного забезпечення, а також відсутність індустріальної стандартизації.

Викладений вище огляд демонструє значну різницю між підходами до реалізації адаптивного керування освітленням. Одна частина систем орієнтована на простоту використання, інша - на масштабованість та промислову надійність, тоді як IoT-рішення забезпечують максимальну гнучкість. Для того щоб підкреслити ці відмінності та виділити ключові характеристики кожного підходу, доцільно навести їх у вигляді порівняльної таблиці 1.2.

Таблиця 1.2 - Порівняльний аналіз існуючих систем адаптивного керування освітленням

Критерій	Комерційні системи (Philips Hue, Yeelight)	Промислові системи (DALI, Helvar)	IoT-системи (ESP32, Arduino)
Автоматизація	Середня	Висока	Залежить від алгоритму
Адаптивність	Обмежена	Висока	Висока
Гнучкість налаштування	Низька	Середня	Висока
Вартість впровадження	Висока	Дуже висока	Низька
Масштабованість	Середня	Висока	Висока
Вимоги до обслуговування	Низькі	Високі	Залежить від реалізації
Використання датчиків	Є, але обмежено	Повноцінне	Повна свобода вибору
Можливість створення власних алгоритмів	Відсутня	Мінімальна	Необмежена

Зведені результати порівняння, наведені у таблиці 1.2, демонструють суттєві відмінності між групами сучасних систем освітлення. Комерційні рішення орієнтовані переважно на комфорт користувачів і пропонують базову автоматизацію, проте мають обмежені можливості адаптації. Промислові системи забезпечують високу надійність та підтримують складні сценарії керування, однак мають високу вартість впровадження. IoT-підхід поєднує низьку ціну та максимальну гнучкість, проте вимагає самостійної розробки

програмної логіки. Саме тому IoT-рішення є найбільш перспективними для створення адаптивної системи керування освітленням у рамках даної роботи.

1.3 Порівняння систем на основі фіксованих правил та адаптивних алгоритмів

Системи керування освітленням можуть базуватися на двох принципово різних підходах: застосуванні фіксованих правил (rule-based) та використанні адаптивних алгоритмів, здатних самостійно змінювати параметри роботи. Обидва підходи мають свої переваги, однак різняться гнучкістю, точністю та можливістю реагування на зміну умов.

Системи з фіксованими правилами (rule-based) працюють за принципом «якщо—тоді». Така логіка дозволяє визначати чіткі сценарії, наприклад: якщо рівень освітленості падає нижче встановленого порогу, світильник вмикається на заданий відсоток яскравості. Цей підхід є простим у реалізації та не потребує значних обчислювальних ресурсів. Однак у динамічних умовах його ключовим недоліком стає відсутність здатності адаптуватися до швидких змін, що знижує загальну ефективність роботи системи.

Адаптивні системи використовують алгоритмічні моделі, які аналізують надходячі дані та формують рішення відповідно до поточних умов. Такі системи можуть реагувати на зміни рівня природного світла, поведінку користувачів, історичні дані або зовнішні фактори. Нелінійні моделі керування, нечітка логіка, PID-регулятори та інші методи дозволяють значно підвищити точність роботи та знизити енергоспоживання завдяки гнучкому регулюванню.

Для кращого узагальнення відмінностей між підходами наведено порівняльну.

Таблиця 1.3 - Порівняння rule-based систем та адаптивних алгоритмів

Критерій	Rule-based системи	Адаптивні алгоритми
Логіка роботи	Жорстко визначена	Змінюється залежно від умов
Реакція на зовнішні зміни	Обмежена	Висока
Енергоефективність	Середня	Висока
Гнучкість	Низька	Висока
Потреба в налаштуванні	Ручне	Автоматичне / частково автоматичне
Складність реалізації	Низька	Середня / висока
Масштабованість	Обмежена	Висока
Застосування	Прості та стабільні умови	Динамічні середовища, «розумні» будівлі

У підсумку обидва підходи мають право на існування, однак їх доцільність залежить від характеру завдання. Rulebased системи найкраще працюють у стабільному середовищі, де параметри змінюються нечасто, тоді як адаптивні алгоритми є оптимальним вибором для умов, у яких необхідна гнучкість та швидка реакція на зміни.

Висновки до розділу 1

У межах аналітичного огляду було розглянуто ключові підходи до побудови систем освітлення та їх еволюцію від простих ручних рішень до сучасних інтелектуальних технологій. Здійснений аналіз підтверджує, що традиційні та rule-based системи, попри свою простоту й передбачуваність, не здатні забезпечити достатню гнучкість і енергоефективність в умовах динамічного освітнього середовища. Їхня ефективність є прийнятною лише в стабільних та мало змінних умовах експлуатації.

Промислові рішення, такі як DALI або Helvar, демонструють високі можливості масштабування, точність керування та розвинені засоби діагностики. Проте їхня вартість, складність інсталяції та вимоги до інфраструктури значною мірою обмежують сферу застосування, роблячи їх недоцільними для невеликих чи бюджетних проєктів.

Комерційні системи «розумного» освітлення пропонують користувачеві зручність та готові програмні сценарії, проте їхня закритість і залежність від виробника істотно звужують можливості для модифікації та інтеграції з нестандартними сенсорами чи алгоритмами.

Проведений аналіз дає підстави стверджувати, що відкриті IoT-рішення на базі мікроконтролерів є найбільш універсальним і перспективним підходом для побудови адаптивних систем освітлення. Вони поєднують доступність компонентів, можливість створення власних алгоритмів, високу гнучкість, простоту масштабування та інтеграцію з різноманітними сенсорами. Особливу увагу приділено порівнянню rule-based та адаптивних алгоритмів керування. Перевага адаптивних систем є очевидною: вони реагують на реальні зміни середовища, коригують параметри у реальному часі та забезпечують суттєве зниження енергоспоживання. Саме ця властивість є ключовою для сучасних систем автоматизації, де важливими є як комфорт користувача, так і енергоефективність.

Таким чином, результати аналітичного розділу підтверджують доцільність вибору адаптивної IoT-архітектури для подальшої розробки системи керування освітленням у межах цієї магістерської роботи. Зазначений підхід відкриває широкі можливості для реалізації ефективних алгоритмів, які будуть розглянуті у наступних розділах.

РОЗДІЛ 2. МАТЕМАТИЧНІ МЕТОДИ ТА ПРОЄКТУВАННЯ АДАПТИВНОЇ СИСТЕМИ КЕРУВАННЯ ОСВІТЛЕННЯМ

2.1 Вибір математичного апарату та методів обробки даних

Проектування адаптивної системи керування освітленням потребує вибору математичного апарату, який забезпечить коректну роботу алгоритму в умовах змінної освітленості, неповних або зашумлених даних сенсорів та різної поведінки користувачів. На основі аналізу предметної області було виявлено, що система повинна:

- реагувати на динамічні зміни природного освітлення;
- враховувати присутність або активність користувачів;
- адаптувати рівень освітленості без різких коливань;
- залишатися простою у реалізації для впровадження на мікроконтролері (Arduino).

Зважаючи на ці вимоги, доцільно розглянути кілька класів математичних методів: моделей на основі фіксованих порогів, алгоритмів нечіткої логіки, регуляторів PID та дерев рішень.

Порівняння цих підходів показало, що найбільш збалансованим варіантом для даної системи є дерево рішень, оскільки воно забезпечує адаптивну реакцію на змінні умови, але не потребує складних обчислень і добре підходить для мікроконтролерів обмежених ресурсів.

На відміну від нечітких систем, які потребують формалізації багатьох правил та функцій належності, або від PID-регуляторів, які вимагають точного математичного моделювання та підбору коефіцієнтів, дерево рішень дозволяє інтуїтивно та прозоро

формалізувати логіку адаптації, зберігаючи можливість подальшого розширення. Крім того, такий підхід легко реалізується у середовищах типу Arduino IDE.

2.1.1 Теоретичні основи обраних математичних методів

Дерево рішень є структурованим способом моделювання логіки прийняття рішень у вигляді послідовності перевірок умов. У контексті керування освітленням дерево рішень дозволяє:

- об'єднати дані сенсорів (освітленість, рух, час доби) у єдину модель;
- враховувати різні сценарії поведінки користувачів;
- забезпечувати плавний перехід між станами;
- формувати адаптивну реакцію без необхідності складних математичних обчислень.

Формально дерево рішень можна подати як набір умовних операторів виду:

- Якщо (умова 1) - виконати дію А
- Інакше якщо (умова 2) - виконати дію В
- Інакше - дія С

У випадку освітлення:

- «умова» - значення сенсорів (lux, рух, час)
- «дія» - вибір яскравості або сценарію освітлення

Переваги даного підходу:

- низькі вимоги до ресурсів;
- відсутність потреби у складних математичних моделях;

- прозорість логіки та можливість валідації;
- масштабованість - легко додавати нові вузли дерева.

Таким чином, дерево рішень оптимально відповідає поставленим вимогам, забезпечуючи адаптивність без перевантаження системи.

2.1.2 Основні змінні, що впливають на роботу алгоритму

Для забезпечення коректного регулювання освітлення система повинна враховувати ключові змінні, які описують стан середовища й користувача. До них належать:

Робота алгоритму адаптивного керування освітленням визначається низкою параметрів, які характеризують як стан середовища, так і поведінку користувачів. Сукупність цих змінних формує базу для прийняття рішень у рамках обраної математичної моделі (дерево рішень). Кожна змінна має свій тип, діапазон значень та вплив на кінцевий результат - вибір рівня освітленості.

Одним із ключових параметрів є поточний рівень освітленості в приміщенні, що вимірюється за допомогою датчика типу BH1750 або аналогічного фотометричного сенсора. Ця величина визначає кількість природного або штучного світла, яка вже присутня, й виступає базою для обчислення необхідної корекції яскравості світильників. З огляду на те, що природне освітлення може змінюватися доволі швидко (наприклад, через хмарність або переміщення сонячного проміння), алгоритм повинен враховувати не лише абсолютне значення освітленості, а й динаміку її змін.

Другим важливим фактором є наявність або відсутність користувача в зоні освітлення. Дані з PIR-сенсора визначають, чи потрібно підтримувати роботу світильників, а також впливають на режим їх роботи (повне освітлення, знижена яскравість або перехід у режим очікування). Окрім факту присутності, значення має і час простою, що дозволяє системі уникати передчасного вимкнення та

забезпечує плавність переходів.

Не менш важливим параметром є рекомендована освітленість для конкретного типу сценарію або приміщення. Ця величина може бути як константою, визначеною нормативними документами (наприклад, 300–500 lx для робочої зони), так і індивідуальним налаштуванням користувача. Врахування таких індивідуальних параметрів підвищує комфорт роботи системи та дозволяє адаптувати її під конкретні вподобання.

У деяких випадках алгоритм може враховувати також енергетичні обмеження або цільові режими енергозбереження, коли пріоритет переноситься з максимального комфорту на мінімізацію споживання енергії. Це вводить додаткову змінну - рівень допустимого енергоспоживання, що може автоматично коригувати алгоритм керування яскравістю.

До додаткових змінних, що покращують роботу системи, належать: температура навколишнього середовища (яка може впливати на світловий потік LED-світильників), тимчасові фактори (час доби), а також історичні дані, що дозволяють алгоритму прогнозувати типові сценарії використання освітлення.

Таким чином, алгоритм працює не з окремими показниками, а з комплексом взаємопов'язаних змінних, що описують як фізичні характеристики середовища, так і поведінку користувача. Саме багатфакторна структура вхідних параметрів забезпечує можливість адаптації системи й формує основу для подальшого застосування оптимізаційних або деревоподібних моделей прийняття рішень.

2.1.3 Інтеграція даних з різних сенсорів у межах алгоритму

Ефективність роботи адаптивної системи керування освітленням значною мірою визначається здатністю алгоритму об'єднувати та узгоджувати інформацію з кількох джерел. Оскільки показники середовища змінюються

динамічно, а поведінка користувачів є нелінійною та нерегулярною, алгоритм має забезпечувати коректне поєднання різнорідних сигналів і формувати узгоджене рішення щодо керування світловіддачею. У межах даної роботи використовується підхід, заснований на дереві рішень, що дозволяє формалізувати процес обробки та інтеграції даних у вигляді послідовності умов та гілок. Інтеграція сенсорних даних у структурі дерева рішень.

Кожний тип сенсорів вносить специфічну інформацію, яка впливає на роботу системи:

- Датчики присутності (PIR/мікрохвильові сенсори).
Їх сигнал визначає сам факт активності в приміщенні. Для дерева рішень це базова гілка, що визначає перехід між режимами «активність» / «неактивність». За відсутності користувача алгоритм переходить до енергозберігального режиму.
- Датчики освітленості (BH1750 або аналогічні).
Значення освітленості є безперервною змінною, яка впливає на вибір рівня компенсації штучного світла. У дереві рішень такі дані використовуються у вигляді порівнянь із порогами або інтервалами (наприклад: $<150 \text{ lux}$, $150\text{--}300 \text{ lux}$, $>300 \text{ lux}$). Це забезпечує стабільне регулювання яскравості залежно від природного світла.
- Сенсори персоналізації (камери або BLE-маячки).
У разі використання ідентифікації користувача дерево рішень дозволяє враховувати його індивідуальні параметри:
 - бажану яскравість;
 - колірну температуру;
 - режим роботи (читання, відпочинок, робота за ПК).

Це дає змогу адаптувати освітлення не лише під умови середовища, а й під індивідуальні потреби.

Завдяки структурі дерева рішень алгоритм може послідовно оцінювати кілька умов: факт присутності, рівень освітленості, персональні параметри.

Така ієрархічність забезпечує:

- стійкість до шумів у вхідних даних (помилкові спрацьовування PIR-сенсора не впливають на рівень освітленості);
- швидку реакцію на зміни, оскільки оцінка кожної умови не потребує складних математичних обчислень;
- можливість масштабування, коли нові сенсори або параметри додаються як нові гілки дерева без зміни основної логіки.

Поєднання інформації з різних сенсорів у межах дерева рішень формує адаптивний механізм, здатний:

- коректно реагувати на зміну природного освітлення;
- автоматично перемикає режими залежно від присутності людей;
- персоналізувати світлове середовище;
- мінімізувати енергоспоживання без зниження комфорту.

Таким чином, інтеграція сенсорних даних у логіку дерева рішень забезпечує узгоджену роботу системи, зменшує кількість помилкових рішень і робить алгоритм придатним для реалізації на мікроконтролерах із обмеженими ресурсами.

2.2 Розробка алгоритму адаптивного керування освітленням

Процес розробки алгоритму адаптивного керування освітленням передбачає формування такої послідовності дій, що забезпечує коректну реакцію системи на зміну умов у приміщенні та поведінку користувача. Алгоритм повинен не лише вмикати або вимикати освітлення, а й оптимально поєднувати дані з різних сенсорів, враховувати контекст середовища та забезпечувати

індивідуалізований комфорт.

Структура алгоритму ґрунтується на чотирьох ключових компонентах: детекції присутності, оцінці природної освітленості, ідентифікації користувача та адаптивній корекції параметрів освітлення. Ці процеси об'єднані в єдину послідовність, де кожен етап формує вхідні дані для наступного, що дозволяє забезпечити логічний та стабільний цикл роботи системи.

Першим етапом є визначення присутності користувача за допомогою датчика руху або ІЧ-сенсора(інфрачервоний датчик). Активація цього модуля слугує тригером для запуску наступних процедур. У випадку тривалої відсутності руху алгоритм переходить у режим енергозбереження та вимикає освітлення.

Наступним кроком є ідентифікація користувача (за наявності відповідного модуля), що дозволяє використовувати персональні профілі освітлення для різних осіб. Це підвищує комфорт та забезпечує індивідуальну адаптацію освітлення. Якщо ідентифікація недоступна, система працює за загальними правилами.

Третім етапом виконується аналіз рівня природної освітленості. Алгоритм оцінює дані з датчика освітленості та визначає, чи існує потреба в увімкненні або регулюванні інтенсивності штучного світла. За достатньої кількості природного світла робота світильників знижується або повністю припиняється, що забезпечує оптимальне енергоспоживання.

Після цього система формує адаптивну реакцію - налаштовує яскравість, колірну температуру та режим освітлення відповідно до умов середовища, часу доби та персональних вимог користувача. Завдяки цьому забезпечується баланс між енергоефективністю та комфортом.

2.2.1 Загальний алгоритм адаптивного керування освітленням

Побудова загальної блок-схеми ілюструє логіку високого рівня, що описує послідовність дій системи від моменту виявлення користувача до встановлення режиму освітлення. Основна мета цього етапу – визначити ключові етапи процесу управління освітленням та зрозуміти, що саме потрібно зробити, щоб система могла адаптувати освітлення у приміщенні.



Рисунок 2.1 - Загальна блок-схема алгоритму адаптивного керування освітленням

Загальна блок-схема алгоритму (Рисунок 2.1) демонструє послідовність дій, що виконуються системою у процесі роботи. Алгоритм починається з ініціалізації мікроконтролера та сенсорів, після чого відбувається визначення присутності користувача. Якщо присутність зафіксовано, система переходить до аналізу рівня освітленості та прийняття рішення щодо необхідності зміни яскравості. У разі відсутності користувача алгоритм переходить до енергозберігального режиму, зменшуючи інтенсивність освітлення або вимикаючи світло повністю. Це забезпечує мінімізацію енергоспоживання та продовжує термін служби світлодіодних елементів.

Алгоритм побудовано таким чином, щоб легко масштабувати його в майбутньому - наприклад, додати машинне навчання, врахування часу доби або поведінкові профілі користувачів.

2.2.2 Поетапний розбір логічних модулів алгоритму

Деталізація логічної структури алгоритму передбачає розгляд кожного функціонального модуля, що входить до загальної блок-схеми. Кожен модуль виконує окрему частину обчислювального процесу та забезпечує коректну роботу всієї системи адаптивного керування освітленням. Нижче наведено поетапний аналіз цих модулів, їхні задачі та внутрішню логіку обробки даних.

Першим етапом роботи системи є визначення факту появи користувача в приміщенні. Цей етап запускає подальший ланцюг дій, тому він є ключовим для забезпечення адаптивності системи та економного використання енергії (рисунок 2.2).

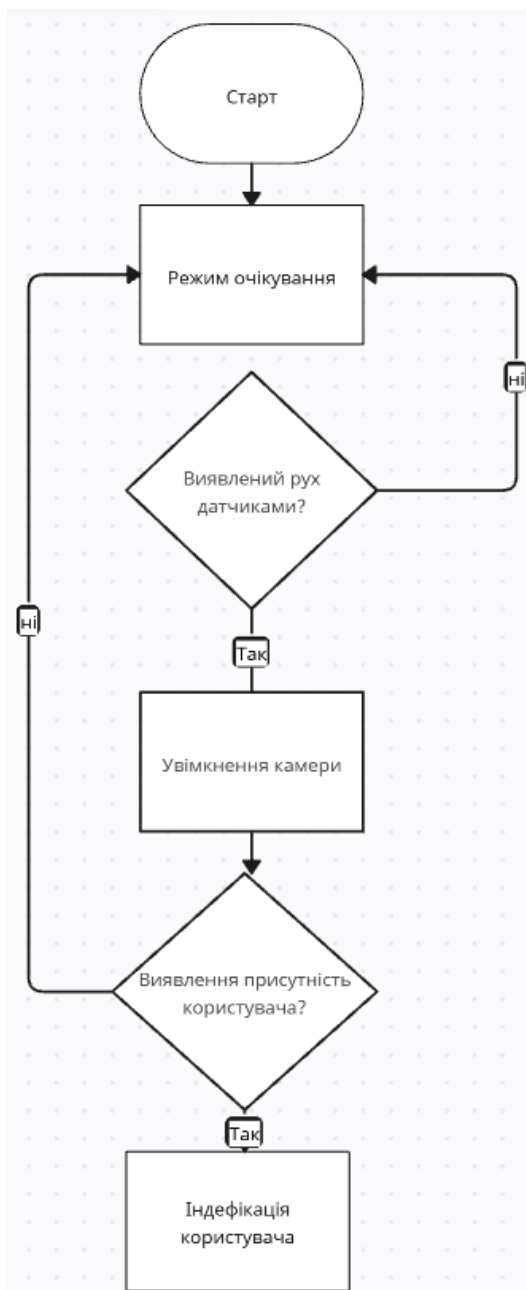


Рисунок 2.2 - Блок-схема модуля «Виявлення присутності користувача»

Робота алгоритму починається з переходу системи у режим очікування, у якому мікроконтролер знаходиться у низькоспоживальному стані та періодично зчитує дані з сенсорів. Це дозволяє зменшити енергоспоживання, але водночас забезпечити

миттєву реакцію у разі появи користувача.

1. Зчитування сигналу з датчиків руху

До системи підключено PIR-сенсор або інший детектор активності.

Мікроконтролер аналізує рівень сигналу та перевіряє, чи зафіксовано зміну інфрачервоного випромінювання, що характерне для руху людини.

2. Умова «Виявлений рух?»

Якщо датчик не виявляє активності - алгоритм повертається у режим очікування. Таким чином формується циклічна перевірка, що мінімізує зайві обчислення та не активує інші компоненти без необхідності.

3. Активація камери у разі позитивного спрацювання

Якщо рух зафіксовано, система переходить до другого рівня перевірки.

Камера вмикається лише після детекції руху, що значно зменшує енергоспоживання та підвищує приватність користувача.

4. Перевірка факту присутності користувача

На основі аналізу зображення визначається, чи дійсно в кімнаті присутня людина. Це дозволяє відфільтрувати хибні спрацювання PIR-сенсора (коливання температури, тварини, рух тіней).

5. Перехід до блоку ідентифікації користувача

Якщо присутність підтверджено, алгоритм переходить до наступного логічного модуля - розпізнавання користувача з метою застосування персональних параметрів освітлення.

Цей логічний блок забезпечує:

- зниження енергоспоживання за рахунок роботи в режимі очікування та умовної активації камери;

- зменшення кількості хибних спрацювань, оскільки рішення приймається на основі двох незалежних джерел інформації;
- підвищення точності адаптивного керування, адже система реагує лише тоді, коли в приміщенні дійсно присутній користувач;
- покращення комфорту, бо світло активується одразу після підтвердження романсима присутності.

Таким чином, модуль «Виявлення присутності користувача» є основою всієї системи адаптивного освітлення та формує логічний старт для подальших етапів роботи алгоритму - ідентифікації особи, аналізу рівня освітленості та адаптації світлових параметрів. Після первинного виявлення руху система повинна переконатися, що зафіксований об'єкт дійсно є користувачем, а не зовнішнім шумом, коливанням світла, тваринами чи механічними факторами. На цьому етапі відбувається уточнення отриманих сенсорних даних та підтвердження факту присутності людини у приміщенні. Саме цей процес у межах роботи алгоритму позначається як ідентифікація користувача (рисунки 2.3) - розпізнавання людської активності, що дозволяє активувати подальші модулі системи адаптивного керування освітленням.

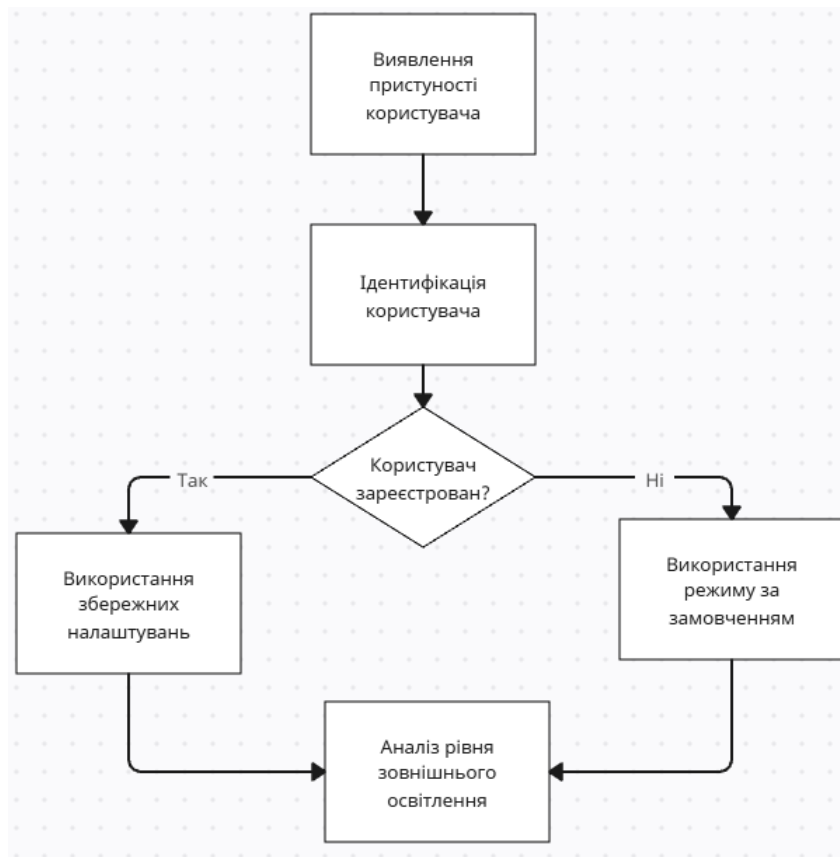


Рисунок 2.3 блок-схема алгоритму Ідентифікація користувача

Пояснення до блок-схеми

1. Отримання сигналу про присутність

Після того як PIR-сенсор або ультразвуковий датчик фіксує рух, мікроконтролер отримує сигнал, який слугує тригером для переходу до етапу уточнення. На цьому кроці система працює у швидкому режимі реагування, щоб уникнути затримок між появою користувача і реакцією освітлення.

2. Ідентифікація як підтвердження людської активності

Система перевіряє, що рух належить саме людині. Для цього

можуть застосовуватися:

- повторні зчитування датчика руху;
- аналіз стабільності сигналу;
- перевірка температурного профілю (для PIR);
- аналіз кадру з камери (силует, розмір об'єкта);
- перевірка тривалості та траєкторії руху.

Мета етапу - уникнути хибних спрацювань, які вмикали б освітлення без потреби.

3. Прийняття рішення «Користувач присутній?»

Якщо алгоритм упевниться, що активність відповідає людині, система позначає приміщення як «зайняте». У разі сумнівних або нестабільних даних можливі два варіанти:

- повторити вимірювання ще раз;
- повернутися в режим очікування.

Це дозволяє підвищити надійність алгоритму.

4. Перехід до аналізу освітленості

Після підтвердження система переходить до наступного етапу аналізу природного світла для визначення, як саме слід регулювати штучне освітлення.

На даному етапі алгоритм виконує оцінку поточного рівня природної освітленості з метою визначення необхідності активації або зміни параметрів штучного освітлення. Функціонально цей модуль відповідає за прийняття базового рішення: чи потрібно системі втручатися в освітлювальні умови

приміщення, чи дозволити роботі перейти у режим очікування.

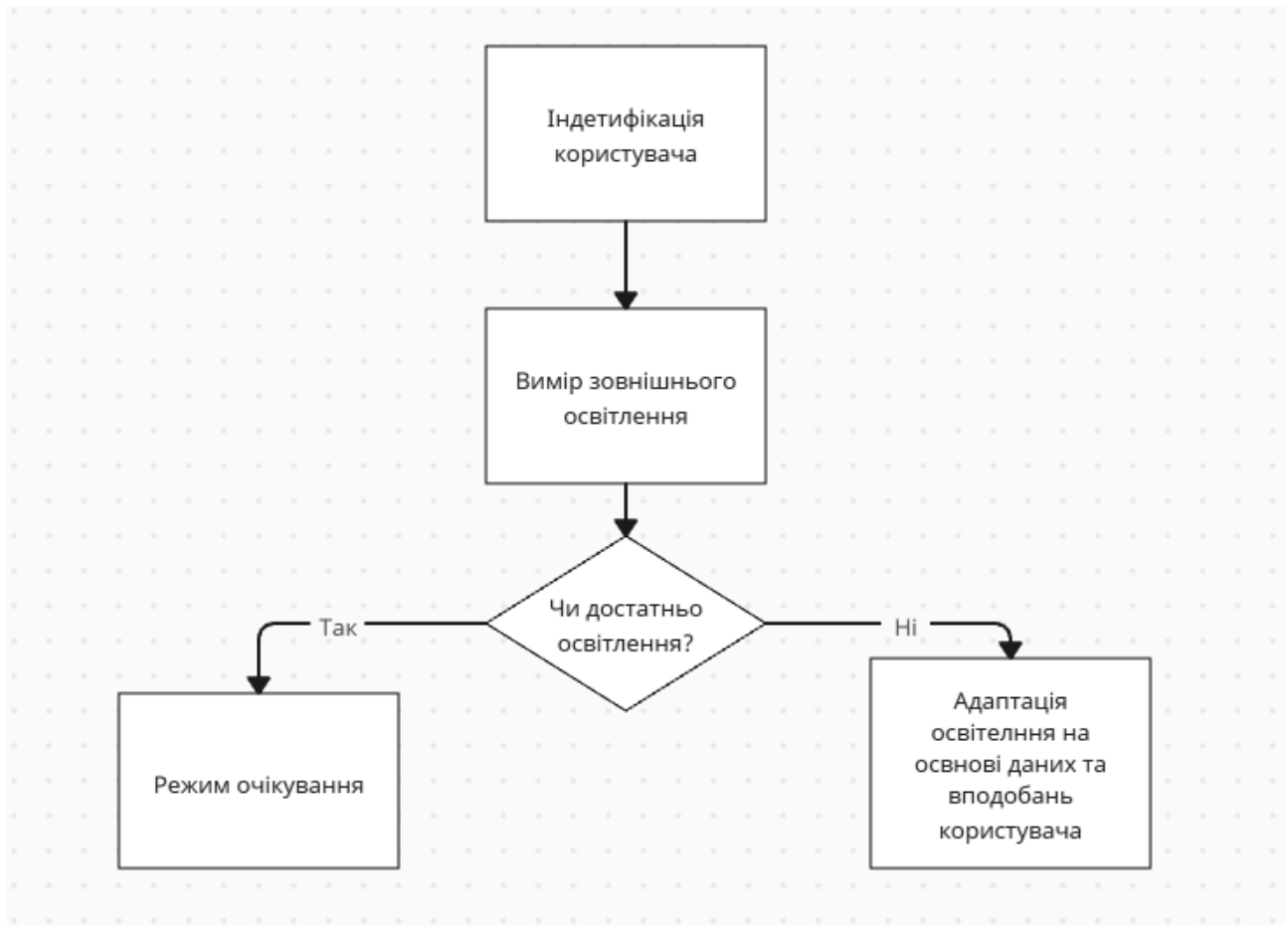


Рисунок 2.5 блок-схема Аналіз рівня зовнішнього освітлення

1. Ідентифікація користувача.
Система визначає факт присутності користувача або фіксує сам факт активності в приміщенні. Ця інформація є ключовою, оскільки подальше керування освітленням виконується лише тоді, коли користувач дійсно перебуває в зоні дії системи.
2. Вимірювання зовнішнього освітлення.
Сенсор освітленості зчитує поточний рівень L_{avg} у люксах. Дані можуть додатково фільтруватися (наприклад, ковзним середнім), що дозволяє уникнути реакції на короточасні коливання природного освітлення.

3. Оцінка відповідності природного освітлення встановленим нормам.

Отримане значення освітленості порівнюється із заданим порогом L_{min} , який відповідає мінімальному рівню комфортного освітлення.

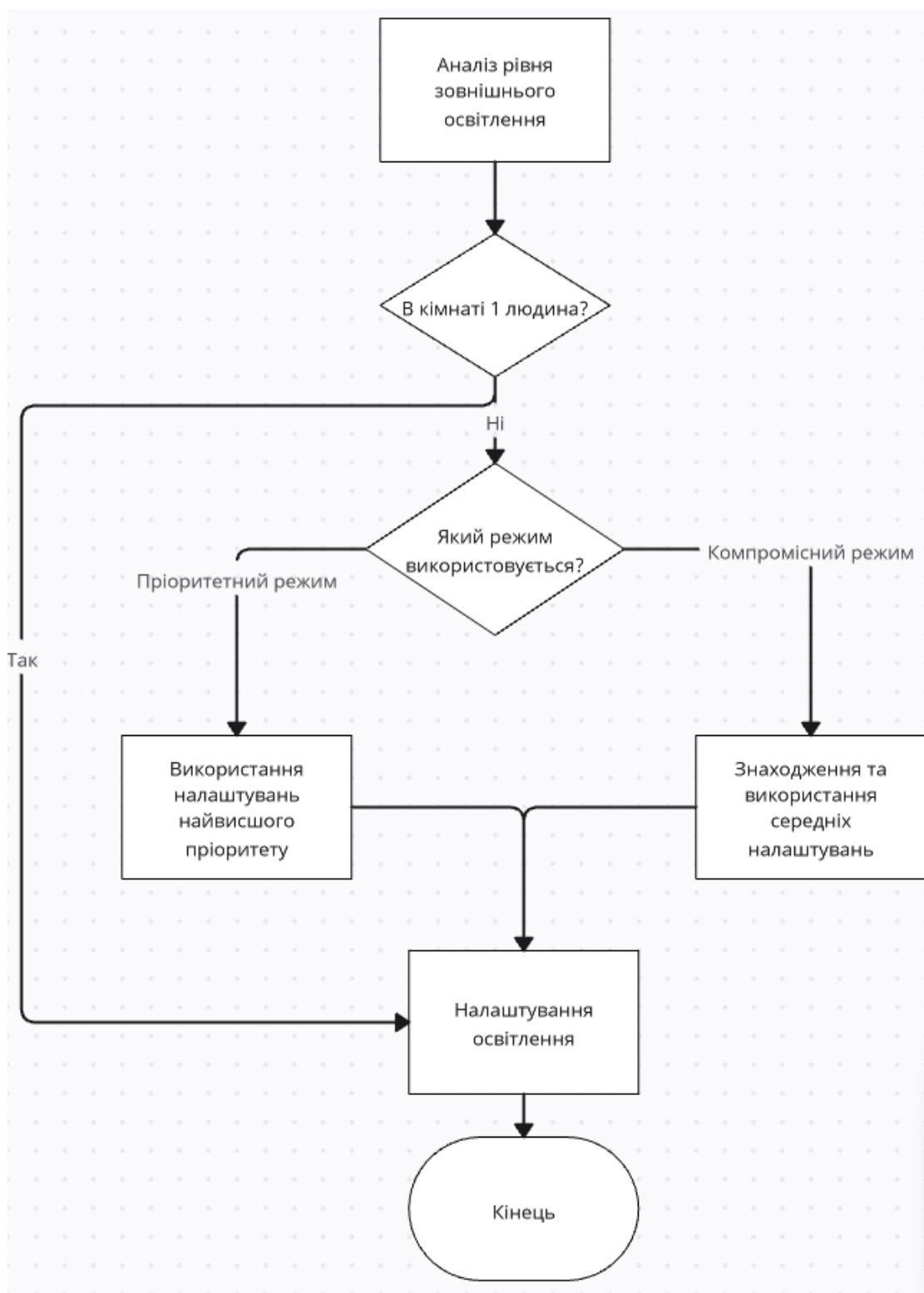
На цьому етапі виконується ключова перевірка: чи є природного освітлення достатньо для виконання задач користувача?

4. Прийняття рішення про подальшу дію.

- Якщо $L_{avg} \geq L_{min}$, тобто рівень природного світла достатній, система переходить у режим очікування, вимикаючи або знижуючи штучне освітлення.
- Якщо $L_{avg} < L_{min}$, виконується перехід до модуля адаптивного керування, де інтенсивність штучного освітлення буде автоматично підібрана відповідно до вподобань користувача та умов роботи.

Етап виконує важливу функцію, визначаючи межу між пасивною та активною роботою системи. Завдяки цьому зменшується кількість непотрібних комутацій освітлення

Після визначення рівня зовнішнього освітлення система переходить до модуля адаптації штучного освітлення відповідно до індивідуальних або групових налаштувань користувачів. Даний модуль відповідає за формування кінцевого значення яскравості освітлення, враховуючи як дані сенсорів, так і персональні вподобання, що дозволяє досягти балансу між комфортом та енергоефективністю.



Блок-схема 2.6 Налаштування освітлення

1. Перехід від етапу аналізу зовнішнього освітлення

Після оцінки природної освітленості алгоритм визначає, чи потрібно

активувати штучне освітлення. Якщо освітлення недостатнє, запускається процедура визначення параметрів освітлення на основі користувацьких профілів.

2. Визначення кількості присутніх користувачів

Алгоритм перевіряє, чи знаходиться в приміщенні один користувач чи декілька:

- Якщо в кімнаті знаходиться лише один користувач, система переходить до пріоритетного режиму, у якому всі параметри освітлення встановлюються відповідно до його персональних уподобань.
- Якщо користувачів більше одного, система активує механізм вибору між пріоритетним режимом, або компромісним режимом, залежно від заданої політики керування.

3. Режим пріоритетного користувача

У випадку вибору пріоритетного режиму система виходить із того, що один із користувачів має вищий рівень прав доступу або виражені потреби у освітленні. У цьому режимі система:

- застосовує індивідуальні налаштування інтенсивності освітлення U_p
- ігнорує менш вагомі профілі,
- забезпечує максимальний комфорт для основного користувача.

4. Компромісний режим

Коли присутні декілька користувачів і немає пріоритету, система переходить до компромісного режиму, що передбачає:

- збирання набору користувацьких параметрів яскравості $\{U_1, U_2, \dots, U_n\}$
- обчислення узгодженого значення яскравості (наприклад, середнє арифметичне або зважене середнє).
- використання цього значення як цільового параметра для налаштування освітлення.

5. Формування підсумкового налаштування освітлення

Після визначення цільової яскравості (U_p або значення, отриманого методом узгодження), система переходить до етапу корекції освітлення:

- порівнює бажаний рівень зі значенням, визначеним сенсором освітленості,
- формує команду на драйвер або реле освітлення,
- здійснює плавне регулювання за допомогою ШІМ, уникаючи різких змін та мерехтіння.

Результатом є встановлення освітленості, що відповідає поточним умовам і вподобанням користувачів.

Функціональна роль модуля Даний логічний модуль відіграє ключову роль у забезпеченні:

- персоналізації освітлення,
- комфорту для всіх присутніх,
- енергоефективного керування,
- адаптації системи до реальних сценаріїв використання,
- гнучкості при роботі з кількома користувачами.

Поєднання аналізу природної освітленості з механізмами індивідуального або колективного налаштування дозволяє системі працювати інтелектуально, а не за фіксованими правилами.

2.3 Математична модель адаптивного керування освітленням

Математична модель адаптивного керування освітленням описує формальні взаємозв'язки між сенсорними даними, параметрами користувача та механізмами регулювання світлового

поток. Модель визначає, яким чином цифрова система інтерпретує отримані дані та формує керуючий вплив на світильник. Основою побудови адаптивного алгоритму є система вхідних змінних, логічні умови переходу та функції оптимізації.

Для прийняття рішення про рівень штучного освітлення система використовує декілька ключових параметрів, кожен з яких має окрему фізичну та функціональну роль.

1. Рівень освітленості L

Вимірюється датчиком освітленості і характеризує кількість природного світла у приміщенні. Ця змінна є основним об'єктивним критерієм для визначення необхідності компенсуючого штучного освітлення.

2. Пріоритет користувача P

Визначає вагомість індивідуальних вподобань щодо: яскравість, температури, режиму роботи. Цей параметр дозволяє алгоритму адаптувати рішення відповідно до ієрархії потреб - особливо у багатокористувацьких сценаріях.

3. Час доби T

Враховується для автоматичної зміни світлових характеристик: тепле світло у вечірній період, холодне світло вдень для підвищення зорової продуктивності. Таким чином модель підтримує добовий цикл освітлення, що сприяє комфортній роботі та дотриманню біоритмів.

4. Присутність користувача (U)

Визначає, чи є у кімнаті користувач, і чи потрібно вмикати освітлення. Ця змінна визначає, чи повинна система взагалі активувати штучне освітлення. У разі відсутності людей освітлення переходить у вимкнений або енергозберігальний режим.

2.3.1 Обчислення середнього рівня освітленості

Для подальшої роботи алгоритму необхідно мати узагальнену оцінку поточного рівня освітленості у приміщенні. Якщо використовуються кілька сенсорів або виконується серія вимірювань за час, середній рівень освітленості визначається за формулою:

$$L_{avg} = \frac{1}{n} \sum_{i=1}^n L_i$$

(2.1)

Де:

L_{avg} - середній рівень освітленості, вимірюється в люксах;

L_i - виміряне значення освітленості з i і-го датчика або під час i і-го вимірювання;

n - загальна кількість вимірів або сенсорів.

Отримане значення L_{avg} використовується як базовий параметр у алгоритмі дерева рішень і відображає узагальнену оцінку природного освітлення в приміщенні. Це дозволяє зменшити вплив випадкових коливань та локальних відхилень, що можуть бути пов'язані з розташуванням окремих датчиків.

На його основі система визначає, який режим освітлення слід застосувати:

- якщо $L_{avg} > L_{max}$ освітлення зменшується або вимикається;

- якщо $L_{avg} < L_{min}$ - система активує штучне освітлення;
- якщо L_{avg} знаходиться у межах $[L_{max}, L_{mix}]$ - поточний рівень освітлення вважається оптимальним.

Отримане середнє значення дозволяє виключити випадкові локальні коливання освітленості та забезпечує стабільну основу для подальших математичних обчислень. Саме на цьому параметрі ґрунтується перший рівень оцінки стану приміщення, що надалі використовується в алгоритмі адаптивного регулювання.

2.3.2 Усереднення вподобань користувачів

У випадку, коли в приміщенні може перебувати кілька користувачів одночасно, система повинна враховувати не тільки об'єктивний рівень освітлення, але й суб'єктивні вподобання щодо яскравості. Для цього вводиться середнє значення вподобань яскравості:

$$P_{avg} = \frac{1}{m} \sum_{j=1}^m P_j$$

(2.2)

Де:

P_{avg} - середнє значення вподобань яскравості;

P_j - індивідуальне налаштування j -го користувача щодо бажаного рівня яскравості;

m - кількість користувачів у приміщенні.

Таким чином, P_{avg} відображає узгоджений рівень очікуваної яскравості для

групи користувачів. Це значення відіграє ключову роль при формуванні компромісного режиму освітлення та дозволяє адаптувати систему до різних сценаріїв використання приміщення.

2.3.2 Функція визначення оптимального рівня освітлення

Для узгодження поточного рівня освітленості з вподобаннями користувачів використовується функція оптимізації, яка визначає оптимальний рівень освітлення в приміщенні:

$$(2.3) \quad L_{opt} = L_{avg} + a(P_{avg} - L_{avg})$$

L_{opt} - оптимальний рівень освітлення;

L_{avg} – середній рівень природного освітлення;

P_{avg} – середнє значення користувацьких вподобань;

a - безрозмірний коефіцієнт адаптації, який визначає вплив користувацьких уподобань на загальний рівень освітленості

Запропонована функція інтерполяції забезпечує плавний перехід між фактичним та бажаним рівнем освітленості. Завдяки коефіцієнту адаптації a система гнучко реагує на зміни зовнішніх умов і користувацьких вимог, визначаючи той рівень освітлення, який є одночасно комфортним та енергетично обґрунтованим.

2.3.3 Обчислення необхідної компенсації штучним освітленням

Після визначення оптимального рівня освітлення L_{opt} система повинна встановити, чи потрібно додавати штучне освітлення та в якій кількості. Для цього обчислюється відхилення між оптимальним та фактичним середнім рівнем

освітленості:

$$\Delta L = L_{opt} - L_{avg} \quad (2.4)$$

Отримана різниця визначає необхідність та інтенсивність активування штучного освітлення. Це значення є основним критерієм для формування керуючого сигналу і дозволяє уникнути надлишкового енергоспоживання у випадках, коли природного світла достатньо.

- 1) Якщо $\Delta L \leq 0$ - рівня природної освітленості достатньо, отже додаткове штучне освітлення не вмикається;
- 2) Якщо $\Delta L > 0$ - природного світла недостатньо, тому система повинна збільшити інтенсивність світильника.

Такий підхід узгоджується з принципами енергоефективності: освітлення активується виключно тоді, коли цього потребують умови.

2.3.4 Формування керуючого впливу на світильник (ШИМ-модель)

Оскільки інтенсивність світлового потоку світлодіодного світильника регулюється методом широтно-імпульсної модуляції (ШИМ), необхідно перетворити значення ΔL рівень цифрового керуючого сигналу. Це перетворення описується формулою:

$$PWM = \frac{\Delta L}{L_{max}} \cdot 255 \quad (2.5)$$

де

PWM - значення широтно-імпульсної модуляції (0–255), яке подається на драйвер світильника;

ΔL - необхідна компенсація освітленості;

L_{max} - максимальний рівень штучної освітленості, який може забезпечити світильник;

255 - максимальна розрядність 8-бітного ШІМ-сигналу мікроконтролера.

Завдяки цьому перетворенню фізична величина нестачі освітлення переходить у цифровий керуючий сигнал, який може безпосередньо застосовуватися мікроконтролером для регулювання яскравості світильника.

Такий підхід забезпечує плавність керування та сумісність із будь-якими LED-драйверами, що підтримують широтно-імпульсну модуляцію.

Значення PWM є пропорційним величині нестачі освітленості та дозволяє плавно регулювати яскравість у межах від 0 % до 100 %.

Для забезпечення фізично коректної роботи вводиться обмеження:

$$PWM = \left\{ 0, \Delta L \leq 0, \Delta L \geq L_{max}, \frac{\Delta L}{L_{max}} \cdot 255, 0 < \Delta L < L_{max} \right\} \quad (2.6)$$

Ця функція забезпечує можливість безперервного регулювання яскравості світильника та сумісна з апаратною реалізацією ШІМ більшості мікроконтролерних платформ.

Це гарантує, що значення керуючої дії завжди залишатиметься в межах допустимого діапазону.

Отримана модель забезпечує плавне регулювання яскравості без стрибків. відсутність мерехтіння, оскільки ШІМ працює з постійною частотою, змінюючи лише коефіцієнт заповнення. Енергоефективність, бо інтенсивність штучного світла пропорційна реальній потребі. Адаптивність, завдяки впливу

параметрів користувачів і врахуванню природної освітленості. Цей підхід є універсальним і сумісним з будь-якими мікроконтролерними платформами, що підтримують апаратну або програмну ШІМ.

Висновок до розділу 2

У другому розділі було розроблено комплексну структурно-функціональну модель системи адаптивного керування освітленням, що об'єднує апаратні засоби, логіку прийняття рішень та математичний апарат регулювання світлового потоку. На основі аналізу вимог до системи побудовано загальну архітектуру, яка включає сенсорний модуль, модуль обробки даних, блок алгоритмів прийняття рішень і запропоновану модель дозволяє масштабувати систему та розширювати її функціональність без зміни базового алгоритмічного ядра керування освітленням.

У підрозділі 2.2 детально описано логічні модулі алгоритму адаптивного керування. Розглянуто етапи виявлення присутності користувача, аналізу рівня зовнішнього освітлення, визначення режиму роботи та адаптації параметрів освітлення відповідно до індивідуальних або групових уподобань. Побудовані блок-схеми демонструють послідовність переходів між режимами роботи, що дозволяє забезпечити інтелектуальну та гнучку реакцію системи на зміну умов у приміщенні.

Підрозділ 2.3 містить математичний аспект роботи системи, у якому сформовано формальні залежності для оцінки сенсорних даних, усереднення природного освітлення, узгодження параметрів користувачів та визначення оптимального рівня штучного освітлення. Введено функцію оптимізації, яка описує баланс між фактичним рівнем освітленості й бажаними уподобаннями, а також модель перетворення компенсаційної складової у керуючий сигнал ШІМ. Отримана математична модель дозволяє реалізувати регулювання освітлення в реальному часі, забезпечуючи плавність переходів та енергоефективність

роботи системи.

Загалом у розділі обґрунтовано структуру та принципи функціонування адаптивної системи освітлення, що інтегрує апаратні та програмні компоненти у єдину інтелектуальну платформу. Результати, представлені у цьому розділі, створюють основу для подальшої програмної реалізації та моделювання системи у середовищах автоматизації, що буде розглянуто у наступному розділі.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Реалізація апаратно-програмного забезпечення частина роботи спрямована на перевірку працездатності розробленої системи адаптивного керування освітленням, побудованої на базі мікроконтролера Arduino Uno та сенсорних модулів. Метою моделювання є дослідження реакції системи на зміну зовнішніх умов, оцінка коректності роботи алгоритму та аналіз ефективності регулювання інтенсивності освітлення.

Для реалізації було обране середовищі Wokwi, що відтворює роботу сенсорів та виконавчих елементів у режимі реального часу. Такий підхід дозволяє дослідити динаміку адаптивного освітлення без використання фізичного обладнання та забезпечує точний контроль вхідних параметрів.

Дослідження охоплює аналіз зміни рівня природної освітленості, вплив користувацьких налаштувань яскравості, а також логіку енергозбереження при відсутності руху. На основі отриманих даних були сформовані графічні залежності та виконана оцінка точності й стабільності роботи алгоритму.

3.1. Вибір елементної бази та обґрунтування компонентів системи

Була сформована апаратна конфігурація, що відтворює основні функції розробленої системи адаптивного освітлення. Метою вибору елементів є забезпечення коректної роботи алгоритмів, оцінка їх поведінки в типових експлуатаційних умовах та можливість відтворити ці результати як у симуляційному середовищі, так і в реальному апаратному виконанні.

Оскільки практична частина реалізована на базі симулятора Wokwi, було враховано, що це середовище використовує узагальнені моделі сенсорів та модулів. Тому під час опису елементної бази наведено реальні апаратні аналоги, максимально близькі за характеристиками до використовуваних у симуляторі.

1)Фоторезисторний модуль (LDR) - це датчик освітленості, що змінює свій опір залежно від рівня падаючого світла. В даній системі LDR використовується для моніторингу яскравості навколишнього середовища та реалізації автоматичного регулювання освітлення. Модуль містить фоторезистор у складі подільника напруги і забезпечує аналоговий вихід, що зчитується Arduino. Це дозволяє вимірювати освітленість у приміщенні та, наприклад, затемнювати або вимикати світло при достатньому денному освітленні. Крім того, типовий фоторезисторний модуль обладнано компаратором для генерування цифрового сигналу при досягненні заданого порогу освітленості, що дає змогу налаштувати рівень спрацювання. Такий модуль обрано через його простоту, низьку вартість та достатню чутливість у видимому діапазоні світла для вирішення задачі енергоефективного керування освітленням. На рисунку 3.1(1) представлено зовнішній вигляд типового фоторезисторного датчика у середовищі Wokwi. Оскільки симулятор не підтримує цифрові фотометри типу BH1750/TSL2591, для моделювання освітленості використано LDR у складі подільника напруги. Це забезпечує аналоговий сигнал, достатній для перевірки роботи алгоритму автоматичного регулювання освітлення, а також дозволяє відтворити зміну інтенсивності світла у різних сценаріях. Основні технічні параметри модуля наведено в таблиці 3.1.

Таблиця 3.1 – Основні технічні характеристики фоторезисторного модуля (LDR)

Характеристика	Значення
Діапазон робочої напруги	3-5 В (типове підключення до аналого-входу Arduino)
Опір фотодатчика при 10 люкс	~10-20 к Ω (падіння опору зі збільшенням освітленості)
Опір фотодатчика у темряві	≥ 1 М Ω (при освітленості ~0 люкс)
Пікова спектральна чутливість	-540 нм (зелене світло)
Час реакції (на освітлення/затемнення)	~20 мс / 30 мс
Вихідний сигнал	Аналоговий (0–5 В) пропорційно рівню освітлення; цифровий (TTL) при спрацюванні порогу

Пороговий рівень освітленості для цифрового виходу можна налаштувати. За умовчанням при живленні 5 В поріг становить близько 100 люкс (відповідає напрузі ~2,5 В на виході АО), що відповідає умовам сутінків або освітлення кімнати.

2)У складі системи використовується змінний резистор (потенціометр) номіналом 10 к Ω для задання порогів або режимів роботи в ручному режимі. Зокрема, за допомогою потенціометра користувач може регулювати граничний рівень освітленості, при якому вмикається/вимикається освітлення, або яскравість лампи в адаптивному режимі. Потенціометр підключається до аналогового входу Arduino, що дозволяє зчитувати його положення (у діапазоні 0–1023 в відліках АЦП) і враховувати його значення в алгоритмі керування. Вибір даного компоненту обумовлений простотою реалізації інтерфейсу користувача – поворотна ручка є інтуїтивним засобом налаштування, а 10 біт розрядності АЦП Arduino достатньо для плавного регулювання. На рисунку

3.1(2) показано типовий потенціометр, застосований у схемі Wokwi, та наведено його характеристики в таблиці 3.2.

Таблиця 3.2 – Основні технічні характеристики потенціометра (10 kΩ)

Характеристика	Значення
Номінальний опір	10 kΩ (лінійна залежність, тип B10K)
Номінальна потужність	0,5 Вт (при 25 °C)
Робоча напруга	до 50 В (постійного струму) – обмежується розсіюваною потужністю
Точність (допуск)	±20 % (типове відхилення опору)
Тип конструкції	Поворотний однооборотний (кут повороту ~300°); металевий вал діаметром ~6 мм
Рекомендований режим роботи	Використовується як дільник напруги, вихідний діапазон 0...5 В (для входу Arduino)

Лінійний потенціометр B10K обрано через рівномірну зміну вихідної напруги по всьому діапазону, що спрощує інтерпретацію його показів у програмі контролера.

3) Для виявлення присутності людей у зоні освітлення використовується пасивний інфрачервоний датчик руху (PIR, наприклад, модуль HC-SR501). Датчик реагує на зміну теплового інфрачервоного випромінювання, що виникає при русі людини в полі зору сенсора. Таким чином, PIR-модуль дозволяє автоматично вмикати освітлення, коли в приміщенні з'являються люди, і вимикати (або зменшувати яскравість) за їх відсутності, що підвищує енергоефективність. Модуль HC-SR501 вибрано завдяки його високій чутливості та готовності до використання: він має вбудований підсилювач сигналу та схему формування імпульсу виходу. Типовий датчик працює від напруги +5 В і формує на виході цифровий сигнал HIGH при виявленні руху на інтервал часу заданої тривалості. У конструкції модуля є два регулювання – затримка вимкнення та чутливість, а також перемичка режиму повторного тригера. PIR-датчик має

широкий кут огляду (близько 100–110°) та дальність виявлення руху до 6–7 метрів, що підходить для більшості приміщень. При цьому він споживає дуже малий струм (порядку десятків мікроампер у черговому режимі). На рисунку 3.1 показано зовнішній вигляд типового PIR-модуля у середовищі Wokwi. Основні характеристики наведено в таблиці 3.3(3).

Таблиця 3.3 – Основні технічні характеристики PIR-датчика руху (модуль HC-SR501)

Характеристика	Значення
Робоча напруга	4,5...20 В DC (звичайно живиться від +5 В Arduino)
Споживаний струм (черговий режим)	<50 μ A (дуже мале енергоспоживання у відсутності руху)
Дальність виявлення руху	3...7 м (регульована, залежно від чутливості)
Кут виявлення	$\leq 100^\circ$ (конус охоплення по горизонталі)
Час утримання виходу	0,3...200 с (регулюється вручну; типове значення ~5 с)
Режим повторного тригера	Налаштовується перемичкою: одноразове спрацювання або продовження таймеру при русі
Вихідний сигнал	Цифровий, рівень HIGH ~3,3 В при виявленні руху (LOW у стані спокою), TTL-сумісний

Модуль PIR після подачі живлення проходить фазу ініціалізації, протягом якої він адаптується до фонових ІЧ-умов. У цей час помилкові спрацювання можливі, тому керування освітленням активується після стабілізації датчика.

4) Модуль реле. Для комутації силового кола лампи використовується одноканальний модуль реле з котушкою на 5 В. Arduino UNO не може безпосередньо керувати мережевим навантаженням, тому реле слугує інтерфейсом між низьковольтною схемою контролера та високовольтною лампою. Обране реле типу SRD-05VDC-SL-C розраховано на керування потужним навантаженням – його контакти витримують струм до 10 А при напрузі 250 В AC або 28 В DC, чого достатньо для типових освітлювальних

приладів. Модуль реле містить транзисторний підсилювач струму та діод захисту від ЕРС, що дозволяє безпечно керувати реле від виходу Arduino. При подачі логічного сигналу HIGH на вхід модуля реле відбувається перемикання контактів: навантаження під'єднується до джерела живлення (замикається ланцюг). Вибір даного компонента зумовлений необхідністю гальванічної розв'язки та надійної комутації – електромеханічне реле забезпечує повну ізоляцію керуючого кола від силового, що підвищує безпеку. На рисунку 3.1(4) показано одноканальний модуль реле в середовищі Wokwi, використаний у схемі. Його ключові характеристики наведено в таблиці 3.4.

Крім дискретного керування реле, у прототипі реалізовано плавне регулювання яскравості за допомогою широтно-імпульсної модуляції (PWM). Такий підхід дає змогу відтворити роботу адаптивного освітлення, коли система не лише вмикає лампу, а й змінює інтенсивність світлового потоку залежно від рівня природної освітленості та присутності користувачів.

Оскільки симулятор Wokwi не підтримує моделювання промислових інтерфейсів типу 0–10V або DALI, які використовуються у реальних освітлювальних драйверах, у прототипі застосовано саме комбінацію PWM та реле. Це дозволяє коректно протестувати алгоритм оптимізації без прив'язки до конкретного типу апаратного світлодіодного драйвера, зберігаючи повну функціональність системи.

PWM-сигнал у даній роботі використовується для моделювання плавного зменшення або збільшення яскравості світильника залежно від даних сенсорів освітленості (LDR) та руху (PIR). У реальній системі даний принцип відповідає керуванню драйвером освітлення через аналоговий канал 0–10V або цифрові протоколи керування, проте в рамках симулятора роль керованого світильника виконує світлодіод, що реагує на зміну робочого циклу PWM.

Таблиця 3.4 – Основні технічні характеристики модуля реле (5 В, 1 канал)

Характеристика	Значення
Номінальна напруга котушки	5 В DC
Опір котушки	~70 Ω (струм спрацювання ~70 мА, потужність ~0,36 Вт)
Кількість контактів	1 перемикаючий (SPDT: спільний, нормально розімкнений, нормально замкнений)
Максимальний струм контакту	10 А (при 250 В AC або 28 В DC, resistive load)
Електрична міцність ізоляції	≥ 1000 В (між котушкою і контактами)
Час спрацювання / відпускання	~10 мс / 5 мс (типові значення для реле даного класу)
Інтерфейс керування	Вхід TTL (через транзисторний драйвер, оптоізолятор у модулі); активний рівень HIGH (5 В)

При використанні реле в схемі з Arduino слід враховувати струм керування ~70 мА, тому керування здійснюється через транзисторний ключ. Також необхідно підключати діод захисту для запобігання пошкодженню транзистора та виходу Arduino імпульсом самоіндукції котушки.

5)В якості обчислювального модуля системи обрано плату Arduino UNO R3 на мікроконтролері ATmega328P. Це популярна і відпрацьована платформа, що має достатню кількість входів/виходів для підключення всіх сенсорів та виконавчих пристроїв системи, а також забезпечує необхідну обчислювальну потужність для виконання алгоритму адаптивного керування у режимі реального часу. Arduino UNO працює від напруги 5 В і містить 14 цифрових входів/виходів (6 з них підтримують ШІМ) та 6 аналогових входів, що цілком покриває потреби даної системи (аналого-цифрові входи використовуються для зчитування LDR та

потенціометра, цифрові входи – для PIR та кнопки, цифрові виходи – для керування реле та світлодіодом). Мікроконтролер ATmega328P має флеш-пам'ять на 32 КБ, ОЗП 2 КБ і тактову частоту 16 МГц, чого достатньо для виконання програмної логіки оптимізації енергоспоживання. До переваг Arduino UNO належать наявність вбудованого АЦП (10-розрядного), інтерфейсів UART, I²C, SPI, а також велика спільнота користувачів та документація, що спрощує розробку та налагодження системи. На рисунку 3.6(5) показано плату Arduino UNO R3. Основні технічні характеристики контролера подано в таблиці 3.5.

Таблиця 3.5 – Основні технічні характеристики контролера Arduino UNO

Характеристика	Значення
Мікроконтролер	ATmega328P (8-біт, RISC)
Робоча напруга логіки	5 В DC (виходи сумісні з TTL/CMOS)
Вхідна напруга живлення	7–12 В (рекомендовано), стабілізована 5 В через вбуд. регулятор
Кількість цифрових портів	14 (з них 6 - з ШІМ)
Кількість аналогових входів	6 (розрядність АЦП 10 біт)
Максимальний струм на пін	20 мА (рекомендоване обмеження навантаження на один вихід)
Тактова частота	16 МГц
Пам'ять програм (Flash)	32 КБ (плюс 0,5 КБ bootloader)
Пам'ять даних (SRAM)	2 КБ
EEPROM	1 КБ
Інтерфейси зв'язку	USB 2.0 (вбудований перетворювач UART), апаратні UART, I ² C, SPI; роз'єм ISP для прошивки
Габарити плати	68,6 × 53,4 мм; маса ~25 г

Arduino UNO обрано завдяки її сумісності з більшістю модулів (датчиків і виконавчих пристроїв), наявності достатньої кількості портів та простоті інтеграції. При необхідності, дану систему можна легко перенести на інший сумісний контролер або розширити, використовуючи шлейфові плати (шилди) для UNO.

6)В якості моделі освітлювального приладу в лабораторному макеті використано низьковольтний світлодіод з послідовним обмежувальним резистором. Світлодіод (LED) виконує роль індикатора увімкнення лампи: при спрацюванні системи він загоряється, моделюючи роботу справжнього освітлення. Таке рішення дозволяє безпечно тестувати алгоритм оптимізації енергоспоживання без використання мережевої напруги. Світлодіод підключено до виходу Arduino через резистор номіналом $220\ \Omega$, який обмежує струм і захищає світлодіод від перегорання, а вихід – від перевантаження. Вибрано звичайний 5-мм світлодіод червоного кольору, який має пряму напругу падіння близько 2 В при струмі 10 мА. При живленні 5 В та резисторі $220\ \Omega$ через LED протікає приблизно 12–15 мА струму, що знаходиться в межах допустимого для виходів Arduino. Яскравість світіння при такому струмі достатня для візуального контролю стану “увімкнено”. Резистор обрано з стандартним значенням $220\ \Omega$ ($\pm 5\%$), розрахованим на потужність 0,25 Вт, що з великим запасом перебиває розсіювану на ньому потужність ($\sim 0,03$ Вт при 15 мА). Отже, зв’язка “світлодіод+резистор” є простим і наочним еквівалентом реального освітлювача в макеті. На рисунку 3.1 показано зовнішній вигляд світлодіода у середовищі Wokwi, а в таблиці 3.6 наведено їх основні параметри.

Таблиця 3.6 – Основні технічні характеристики індикаторного світлодіода з резистором

Характеристика	Значення
Тип світлодіода	5мм LED (діод випромінюючий світло), червоного свічення ($\lambda \approx 625$ нМ)
Пряма напруга (VF)	~1,8...2,2 В (при 10мА)
Номінальний струм світлодіода	10 мА (яскраве свічення); максимальний – 20 мА
Номінал резистора	220 Ω (для обмеження струму приблизно до 15 мА при 5 В живлення)
Розрахункова потужність резистора	~0,03 Вт (при I=15 мА, падіння напруги 3 В); обрано резистор 0,25 Вт (запас 8×)
Інтерфейс підключення	Під'єднано до цифрового виходу Arduino через резистор (анод до піну, катод до GND)

У разі розгортання системи на реальному об'єкті замість світлодіода використовуватиметься повноцінний освітлювальний прилад (лампа або LED-прожектор відповідної потужності), підключений через реле. Індикаторний світлодіод у макеті слугує для відлагодження і демонстрації працездатності алгоритму керування.

7) Для надання можливості вручну перемикає режим роботи системи (наприклад, автоматичний/ручний або ввімкнено/вимкнено) використовується звичайна тактова кнопка. Кнопка підключена до цифрового входу Arduino і при натисканні подає сигнал, за яким програма змінює режим роботи освітлення. Обрана модель кнопки – недорога та надійна, з нормально розімкненими контактами. При відпущеній кнопці коло розімкнене, при натисканні – замкнене

на землю (через налаштований у Arduino внутрішній підтягуючий резистор до +5 В). Це дозволяє фіксувати натискання без додаткових компонентів. Кнопка розрахована на тривалу експлуатацію (ресурс не менше 100 тис. циклів натискань) та здатна комутувати невеликі струми і напруги, характерні для цифрових входів. Застосування кнопки додає гнучкості системі – оператор у будь-який момент може перевести освітлення в ручний режим або вимкнути автоматичне керування, що важливо з точки зору практичного використання. Основні характеристики тактової кнопки наведені в таблиці 3.7. Зовнішній вигляд кнопки показано на рисунку 3.8 (як частину загального вигляду системи без з'єднань).

Таблиця 3.7 – Основні технічні характеристики кнопки перемикання режимів (тактової, 6 мм)

Характеристика	Значення
Тип кнопки	Тактова (TACT switch) , нормально розімкнена, фіксація відсутня (монтажна)
Номінальна напруга	до 12 В постійного струму (максимальна робоча напруга)
Номінальний струм	до 50 мА (достатньо для цифрових схем)
Опір контактів	<0,1 Ω (нової кнопки); низький опір забезпечує мінімальні втрати напруги
Механічний ресурс	$\geq 100\,000$ натискань
Дебаунс (контактний дребін)	~5–10 мс (потребує програмного подавлення для коректного визначення одиничного натискання)

У схемі кнопка підключена з використанням внутрішнього підтягувального резистора контролера (INPUT_PULLUP), тому в ненаτισнутому стані на вході фіксується рівень HIGH, а при натисканні – LOW. Така конфігурація зручна відсутністю потреби в окремому апаратному резисторі.

8) Резистор 220 Ом використовується у схемі для обмеження струму, що протікає через світлодіод, з метою запобігання його пошкодженню. Такий номінал є

стандартним для подібних застосувань і забезпечує надійну роботу світлодіода без перевищення граничних значень струму. У межах симулятора (Wokwi) фізичні параметри, як-от потужність або точність, не мають суттєвого впливу, тому головним критерієм є саме номінал опору.

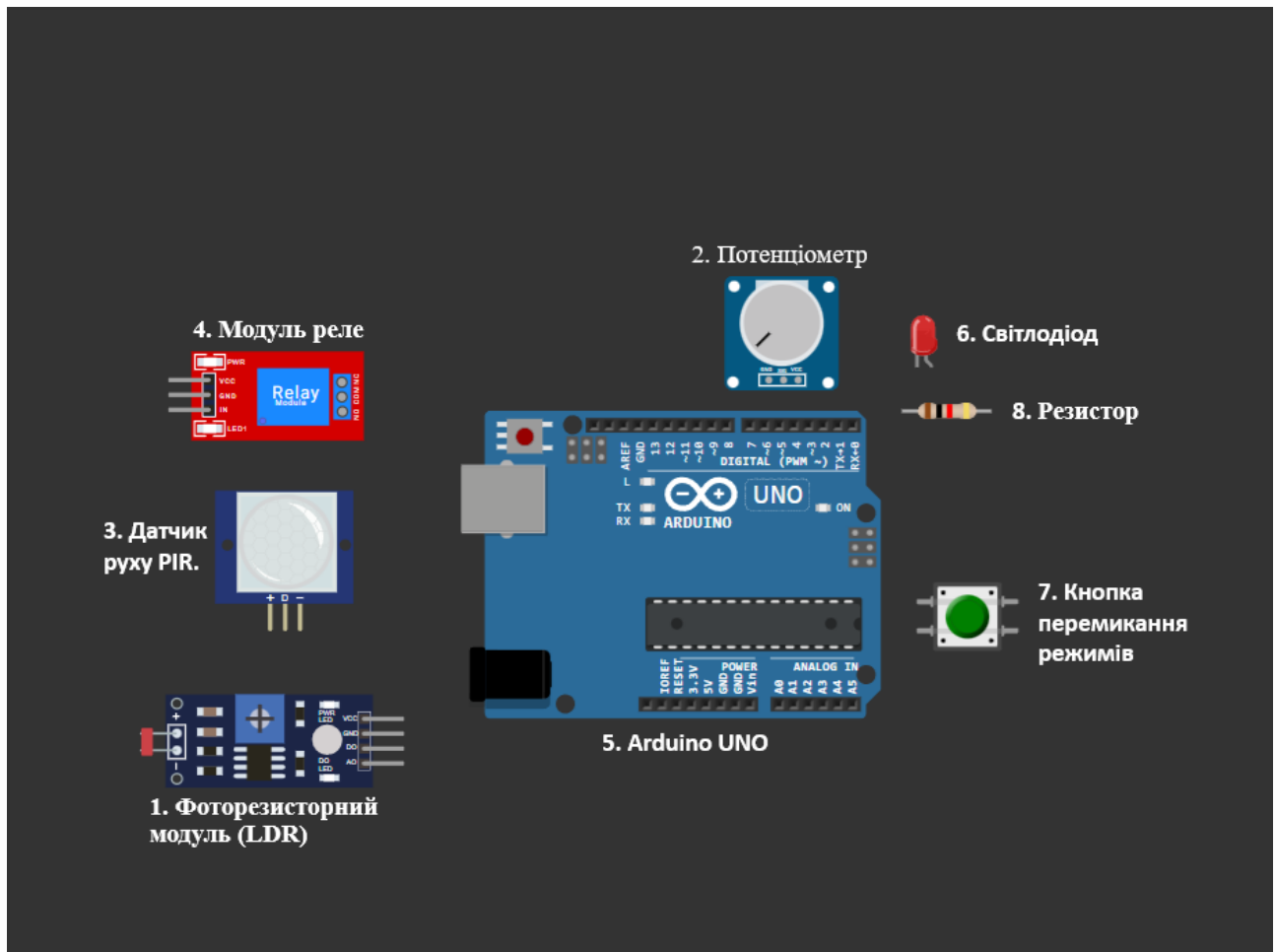


Рис. 3.1 – Експериментальний макет системи адаптивного освітлення у середовищі Wokwi

На рисунку 3.1 наведено загальний вигляд основних компонентів апаратної частини системи (Arduino UNO, датчики, модулі та інші елементи) без з'єднувальних проводів. Це дає уявлення про склад системи та розташування компонентів перед монтажем. Кожен з розглянутих вище компонентів працює у взаємодії з іншими, утворюючи єдину систему адаптивного керування освітленням. Завдяки виваженому вибору елементної бази, схема є функціонально достатньою для виконання поставлених завдань і в той же час

економічною та енергоефективною.

3.2 Підключення компонентів та принцип роботи експериментальної схеми

Після визначення складу елементної бази було виконано підключення усіх компонентів до мікроконтролера Arduino Uno відповідно до функціональних вимог системи. Компонування здійснювалось у середовищі Wokwi, що дозволило перевірити коректність електричних з'єднань та алгоритмів взаємодії між модулями. Схему повного збирання наведено на Рисунку 3.2.

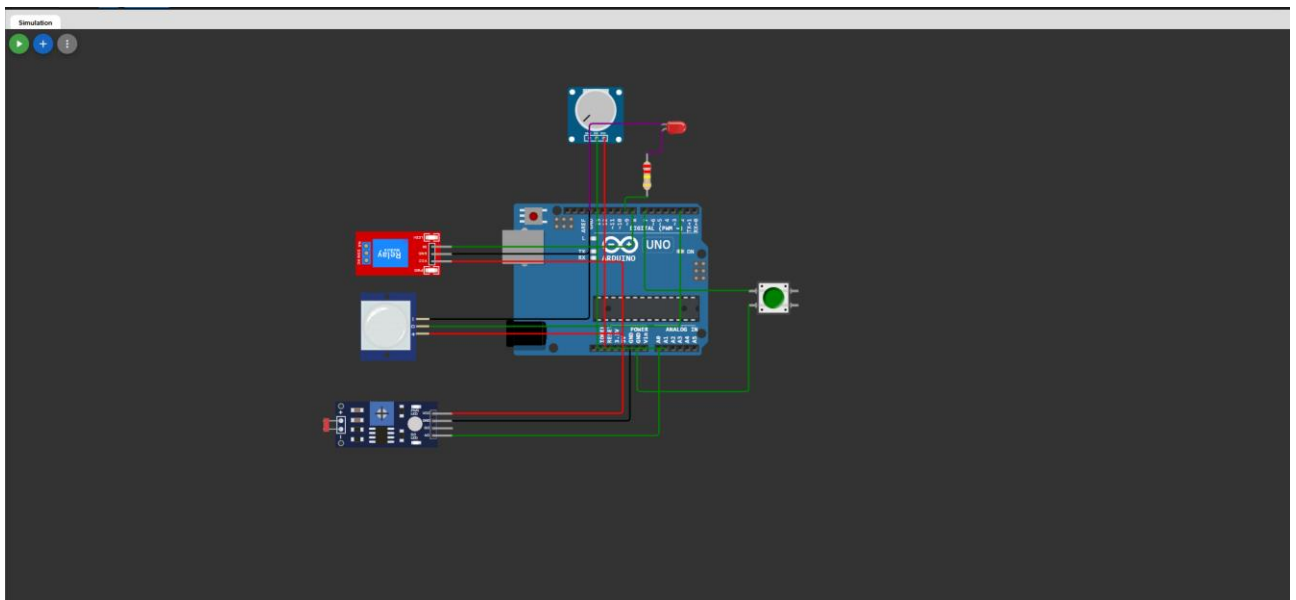


Рис. 3.2 – Повна схема експериментальної системи адаптивного керування освітленням у середовищі Wokwi

У процесі збирання було реалізовано такі групи з'єднань:

1. Підключення датчика освітленості LDR

Модуль фоторезистора підключається за аналоговою схемою:

- AO → A0 - фналогове вимірювання рівня освітленноті;

- VCC → 5V, GND → GND - живлення модуля

LDR формує значення від 0 до 1023, де малі значення відповідають темряві, а великі – освітленому середовищу. Цей параметр використовується для обчислення необхідної додаткової інтенсивності штучного світла.

2. Підключення датчика руху PIR

Датчик руху закріплює факт появи людини в зоні освітлення. Його підключення:

- OUT → D2 – цифрове зчитування стану присутності(HIGHT/LOW).
- VCC → 5V, GND → GND - живлення датчика.

У разі відсутності руху система автоматично вимикає освітлення, навіть якщо рівень освітленості низький, що дозволяє уникнути зайвого енергоспоживання.

3. Підключення реле керування освітлювальним модулем

Модуль реле використовується для умовної імітації вмикання силового освітлення:

- IN → D8 – цифровий сигнал керування вмиканням/вимиканням;
- VCC → 5V , GND → GND – живлення модуля

Реле активується лише тоді, коли розрахована інтенсивність PWM перевищує мінімальний поріг, що запобігає частим перемиканням.

4. Підключення регульованого потенціометра

Потенціометр використано для моделювання побажань користувача щодо бажаного рівня освітленості (P_{avg}):

- Середній контакт $\rightarrow$ A1 — аналогове зчитування;
- Інші два контакти $\rightarrow$ 5V та GND.

Це дає змогу в експерименті впливати на роботу системи та змінювати бажану яскравість.

5. Підключення LED-індикатора який імітує світильник

Світлодіод підключено через резистор 220 Ом як виконавчий елемент:

- Анод $\rightarrow$ D9 - PWM-сигнал керування яскравістю.
- Катод $\rightarrow$ GND

Це дозволяє оцінити плавність регулювання інтенсивності світла.

6. Підключення кнопки режимів

Сервісна кнопка використовується для перемикання режиму роботи системи:

- Один контакт $\rightarrow$ D7 (з активованим внутрішнім підтягуванням).
- Інший контакт $\rightarrow$ GND.

Кнопка працює як аварійне або швидке керування, дозволяючи повністю вимкнути адаптивний режим у разі потреби.

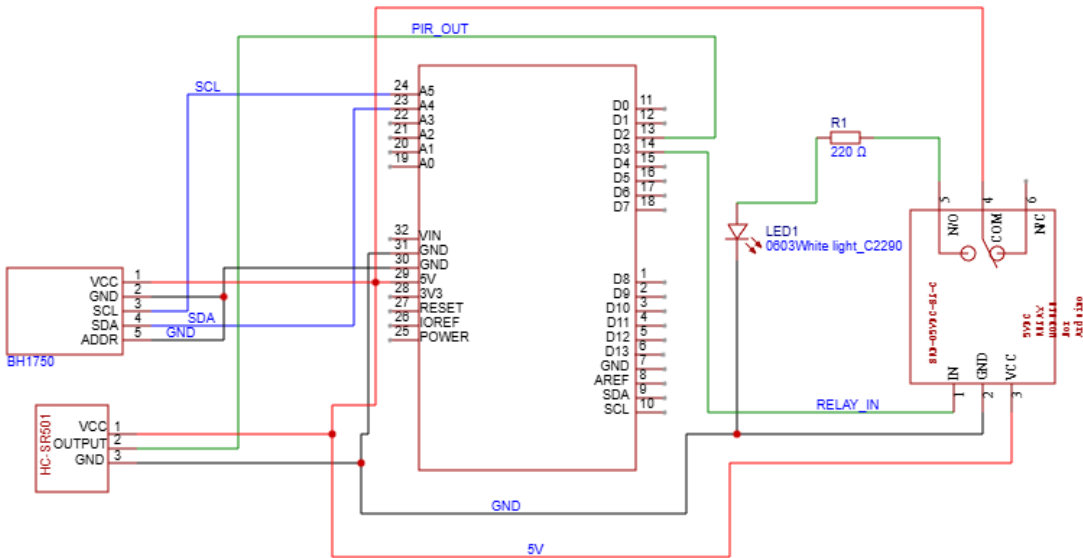
7. Загальний принцип роботи зібраної системи

- 1) PIR фіксує присутність людини. Якщо руху немає - освітлення знеструмлюється.
- 2) LDR визначає фактичний рівень освітлення.
- 3) Потенціометр задає бажаний рівень освітленості.
- 4) Мікроконтролер розраховує оптимальну яскравість за формулою адаптації.
- 5) LED або реле вмикаються з відповідним рівнем потужності.
- 6) Кнопка переведе систему у режим «OFF», коли адаптивне керування не потрібне.

Таким чином схема забезпечує повноцінне моделювання логіки, описаної у попередньому розділі, та дозволяє виконати подальшу експериментальну перевірку роботи алгоритму.

3.2.1 Принципова електрична схема

Для уточнення апаратної реалізації системи, розробленої у симуляційному середовищі Wokwi, було створено повноцінну принципову електричну схему у програмі EasyEDA. На відміну від макетного підключення, принципова схема відображає апаратну структуру системи, включаючи реальні електричні з'єднання, номінали компонентів, логіку роботи модулів та їхню взаємодію. На рисунку 3.3 наведено інженерну схему прототипу системи адаптивного керування освітленням. Схема складається з таких функціональних вузлів:



Принципова схема:	АДАПТИВНЕ КЕРУВАННЯ ОСВІТЛЕННЯМ НА ОСНОВІ СИСТЕМИ СЕНСОРІВ ТА АЛГОРИТМУ ОПТИМІЗАЦІЇ ЕНЕРГОСПОЖИВАННЯ	Create at	2025-12-12
		Update at	2025-12-14
Автор:	Тарів Мекан Первіз огли	Page	P1
		Page 1 Total 1	
		Version	Size
		V1.0	A4
		EasyEDA.com	

Рисунку 3.3 - Принципова електрична схема системи адаптивного керування освітленням

1. Мікроконтролер Arduino Uno.

Центральним елементом схеми є плата Arduino Uno, яка виконує функції обробки сигналів сенсорів, реалізації алгоритму оптимізації енергоспоживання та керування виконавчими елементами. Arduino забезпечує роботу з інтерфейсом I²C, цифровими входами/виходами та живленням периферійних модулів. У схемі використовуються контакти A4 (SDA), A5 (SCL), цифрові входи D2 та D3, а також шини живлення 5V і GND.

2. Датчик освітленості BH1750.

Модуль BH1750 підключений до Arduino за інтерфейсом I²C: лінія SDA з’єднується з входом A4, а лінія SCL — з A5. Пін ADDR підключено до GND, що встановлює адресу датчика на рівні 0x23. Датчик забезпечує вимірювання

рівня освітленості в діапазоні 1–65535 lx, що дозволяє реалізувати адаптивне регулювання освітлення залежно від умов навколишнього середовища.

3. Датчик руху HC-SR501.

Для фіксації присутності людини у приміщенні застосовано PIR-датчик HC-SR501. Його вихід підключений до цифрового піну D2 (сигнал PIR_OUT). Під час виявлення руху модуль формує високий логічний рівень, який обробляється мікроконтролером для активації освітлення.

4. Релейний модуль 5V.

Виконавчий елемент системи побудовано на основі релейного модуля типу SRD-05VDC-SL-C. Модуль містить вбудований транзисторний ключ та діод захисту, що дозволяє керувати ним безпосередньо з цифрового виходу Arduino (пін D3, сигнал RELAY_IN). Комутація навантаження здійснюється контактами COM та NO, що забезпечує безпечне й надійне вмикання або вимикання освітлення.

5. Індикаторний світлодіод.

Для наочного відображення стану реле у схемі використано світлодіод з послідовним резистором 220 Ом. Ланцюг підключено до контактів COM та NO релейного модуля, що дозволяє спостерігати зміну стану навантаження під час роботи прототипу.

6. Система живлення.

Усі компоненти схеми живляться від стабілізованого джерела 5V, спільна шина GND забезпечує правильне функціонування цифрових інтерфейсів та периферійних модулів. Така конфігурація дозволяє реалізувати компактну та енергоефективну апаратну частину системи.

Узагальнюючи, розроблена принципова схема визначає апаратну архітектуру прототипу та забезпечує взаємодію сенсорних модулів,

мікроконтролера та виконавчого елемента. Вона є основою для реалізації програмної частини, описаної у наступних підрозділах, а також для побудови фізичного прототипу системи адаптивного керування освітленням.

3.3 Програмна реалізація алгоритму адаптивного освітлення

Програмне забезпечення для мікроконтролера реалізовано мовою програмування Arduino (базується на C/C++) з використанням середовища Wokwi для моделювання. Вибір платформи Arduino зумовлений її популярністю у проєктах розумного освітлення та наявністю зручних бібліотек для роботи з датчиками і керування пристроями. Зокрема, у подібних розробках саме плата Arduino Uno часто слугує основним контролером системи. Для налагодження коду застосовано онлайн-симулятор Wokwi, що дозволяє імітувати роботу плати Arduino разом із підключеними сенсорами. У додатку А наведено програмний код.

Програма організована у вигляді циклу *loop()*, що постійно виконується на мікроконтролері. Алгоритм адаптивного керування освітленням виконує такі основні кроки в кожній ітерації циклу:

1. Опитування кнопки режиму з подавленням дребезгу.

На початку циклу зчитується стан кнопки перемикання режимів. Реалізовано програмний *debounce*: якщо виявлено зміну стану кнопки, таймер затримки оновлюється, і тільки після стабілізації рівня протягом 50 мс відбувається фіксація натискання. У разі натискання (перехід стану з HIGH в LOW) програма змінює режим роботи між автоматичним та вимкненим станом. Такий механізм гарантує захист від хибних спрацювань кнопки і чітке перемикання режимів.

2. Режим вимкнення (ручний).

Якщо адаптивний режим було вимкнено користувачем, система переходить у стан ручного керування освітленням. В цьому випадку програмно встановлюється нульова яскравість: вихід PWM задається в 0, а реле вимикається, щоб повністю погасити світло. Таким чином користувач може самостійно контролювати освітлення, або ж система залишається вимкненою. В режимі *OFF* світло не регулюється автоматикою, програма не виходить з циклу - далі виконуються діагностичні дії без впливу на виходи. Це зроблено для можливості моніторингу параметрів навіть у вимкненому стані і побудови графіків для аналізу.

3. Зчитування сенсорів.

На цьому етапі програма опитує всі підключені сенсори. З аналогового входу A0 зчитується рівень освітленості LDR, який змінюється від 0 до 1023 чим більше значення, тим більше світла фоторезистор отримує. З входу A1 зчитується положення потенціометра, також у діапазоні 0–1023, що відповідає бажаному рівню освітленості, встановленому користувачем. Цей сигнал виступає встановкою, до якої система прагне адаптувати фактичне освітлення. Далі з цифрового входу 2 зчитується стан PIR-датчика руху – значення HIGH означає наявність руху або присутність людини, LOW – відсутність рухомих об'єктів (порожнє приміщення). Отримані дані зберігаються у змінні `ldrRaw`, `potRaw`, `pirRaw` відповідно.

4. Аналіз присутності та економія енергії.

Перед тим як проводити розрахунки освітленості, програма перевіряє умову присутності. Якщо режим автоматичний (`autoMode == true`), але при цьому датчик руху не виявляє присутніх (`pirRaw == LOW`), система робить висновок, що в приміщенні нікого немає. У такій ситуації подальше підвищення освітленості недоцільне – світло слід вимкнути задля економії енергії. Програма

встановлює яскравість 0 (аналогічно пункту 2: `analogWrite(PIN_LED_PWM, 0)` і `digitalWrite(PIN_RELAY, LOW)`) і фактично переходить у режим очікування до наступного циклу. В послідовний порт виводиться повідомлення стану на кшталт "*No presence, LDR=X, POT=Y*", що відображає при відсутності руху система тримає світло вимкненим. Таким чином реалізовано ключовий принцип енергозбереження: освітлення вимикається за відсутності людей, щоб не витратити електроенергію марно. Лише коли датчик знову зафіксує рух (стан HIGH), автоматичне керування освітленням продовжиться.

5. Обчислення оптимального рівня освітлення.

Якщо присутність виявлено, програма переходить до розрахунку необхідної додаткової підсвітки. Вводяться допоміжні змінні типу `float`:

L_{avg} – усереднений рівень освітленості, отриманий з LDR (`ldrRaw`). В даній реалізації береться миттєве значення датчика, але при потребі його можна було б усереднювати за кілька вимірів для згладжування шуму.

P_{avg} – бажаний рівень освітленості (встановлення), отриманий з потенціометра (`potRaw`).

На основі цих даних використовується адаптивний алгоритм для визначення цільового рівня освітлення L_{opt} . Алгоритм задається формулою:

$$L_{opt} = L_{avg} + a(P_{avg} - L_{avg})$$

де a – коефіцієнт адаптації (в нашому випадку $a = 0.6$, тобто 60%). Ця формула реалізує поступове наближення освітленості до бажаного рівня: величина a визначає частку різниці між бажаним і поточним значенням, на яку потрібно змістити рівень за один крок. При $a=0.6$ нове цільове значення L_{opt} зміщується більш ніж наполовину до встановленого користувачем рівня P_{avg} , але не досягає його повністю за один цикл. Таким чином, якщо в приміщенні темніше, ніж хотів

би користувач, система додає ~60% від різниці освітленості; якщо вже достатньо світло, значення майже не зміниться. Подібний підхід запобігає різким стрибкам яскравості та потенційним коливанням (оскільки система не одразу на 100% компенсує різницю, а робить це згладжено). Отримане значення L_{opt} можна трактувати як оптимальний рівень освітлення, до якого слід прагнути в даний момент. Далі обчислюється різниця між оптимальним і фактичним рівнем:

$$\Delta L = L_{opt} - L_{avg}$$

Величина ΔL показує, наскільки потрібно змінити (збільшити або зменшити) освітлення. В програмі ця різниця зберігається у змінній dL . Якщо $\Delta L > 0$ (оптимальне значення вище поточного, тобто в кімнаті темніше за бажане), то системі слід додати світла. На практиці максимум P_{avg} і мінімум L_{avg} можуть дати ΔL до 1023 одиниць (у разі повної темряви та максимального бажаного рівня). Щоб перетворити цю різницю в сигнал керування LED-лампкою, використано лінійне масштабування: значення ΔL нормується від 0 до 1023 у діапазон ШІМ 0–255. Тобто розраховується рівень PWM, пропорційний потрібній додатковій яскравості:

$$PWM = \frac{\Delta L}{1023} \cdot 255$$

Цей розрахунок виконано в коді як:

```

102   if (dL > 0) {
103       if (dL > 1023.0) dL = 1023.0;
104       pwmValue = (int)(dL / 1023.0 * 255.0);
105   } else {
106       pwmValue = 0;
107   }
108

```

Таким чином отримуємо ціле значення `pwmValue` (0–255), що відповідатиме інтенсивності підсвітки світлодіода на виході `PIN_LED_PWM`. Якщо $\Delta L \leq 0$ (тобто поточна освітленість вже на рівні або вища бажаної), змінна `pwmValue` лишається нульовою - додаткове світло не потрібне.

6. Застосування керуючих дій.

На цьому кроці система виконує реальні дії для регулювання освітлення, але тільки якщо ввімкнений адаптивний режим. У випадку `autoMode == true` обчислене значення `pwmValue` подається на світлодіодну лампу через ШІМ-вихід: викликається `analogWrite(PIN_LED_PWM, pwmValue)`, що встановлює яскравість LED (або лампи з ШІМ-драйвером) відповідно до потреби. В результаті, якщо в приміщенні було темно, лампа засвітиться яскравіше, а якщо було відносно світло – горітиме ледь помітно або зовсім залишиться вимкненою. Окрім регулювання яскравості світлодіода, передбачено включення реле для керування основним освітленням. У коді реалізована умова: якщо значення PWM перевищує певний поріг (більше 10 із 255), то виконується `digitalWrite(PIN_RELAY, HIGH)`, інакше (якщо `pwmValue <= 10`) – `digitalWrite(PIN_RELAY, LOW)`. Цей поріг введено, щоб уникнути частоті комутації реле при незначних рівнях підсвітки. Логіка роботи така: коли потрібна суттєва додаткова яскравість (понад ~4% від максимуму), реле вмикає основну лампу; якщо ж необхідна лише мінімальна підсвітка або не потрібна зовсім – реле вимикає лампу, покладаючись лише на слабе світло світлодіода або природне освітлення. В реальних умовах це може означати, що при невеликому дефіциті світла система обмежиться допоміжним LED-джерелом, а основне освітлення (наприклад, лампа 220 В) не вмикатиметься, економлячи ресурс лампи і електроенергію. При значному затемненні реле ввімкне лампу, забезпечуючи достатній рівень освітлення

7. Логування та графічний моніторинг.

Для аналізу роботи алгоритму в систему додано розгорнутий серійний лог. Після кожної ітерації `loop()` у консоль `Serial` (через USB) виводиться рядок з основними параметрами: значення LDR, потенціометра, PIR, розраховане L_{opt} , різниця dL та встановлений рівень PWM. Одночасно для зручності графічного

відображення, програма друкує ті самі числові значення через пробіл в окремому рядку (форматований під Serial Plotter Arduino IDE). Serial Plotter інтерпретує кожен стовпець чисел як окрему криву, що дозволяє в реальному часі спостерігати зміну параметрів на графіку. У нашому випадку на графіку відображатимуться: рівень освітленості LDR, значення потенціометра (бажаний рівень), стан PIR (0 або 1), обчислений оптимальний рівень L_{opt} , різниця dL та вихідний сигнал PWM. Такий підхід допомагає наочно перевірити динаміку роботи системи – як змінюється вихід при зміні умов. Зняття графіків планується здійснити в експериментальному розділі: буде продемонстровано, як система реагує на різні сценарії (наприклад, різке затемнення або освітлення приміщення, регулювання користувачем бажаного рівня, поява та зникнення людини тощо). Очікується, що на графіках буде чітко видно, як при зростанні показів LDR (більше світла) PWM знижується, а при зменшенні освітленості – збільшується, прагнучи утримувати освітлення біля встановленого рівня. Таким чином, зібрані експериментальні дані підтвердять ефективність адаптивного алгоритму керування освітленням.

8. Частота оновлення.

Наприкінці циклу передбачена невелика затримка `delay(100)`, що робить паузу $\sim 0,1$ секунди перед наступною ітерацією. Таким чином, система оновлює вихід приблизно 10 разів на секунду. Цього цілком достатньо для задачі освітлення, адже зміни яскравості, сприймані людським оком, відбуваються відносно повільно. Інтервал 100 мс запобігає надмірно частому блиманню світла, забезпечуючи стабільну роботу.

Висновки до розділу 3

У даному розділі було здійснено розробку апаратної та програмної частини системи адаптивного керування освітленням на базі мікроконтролера Arduino. Розглянуто вибір основних компонентів – сенсорів освітленості (LDR),

присутності (PIR), елементів керування (кнопка, потенціометр), виконавчих пристроїв (світлодіод, реле) та плати Arduino UNO. На основі цього обладнання побудовано структурну схему системи та визначено логіку її роботи.

Було реалізовано програмний алгоритм, який забезпечує адаптивне регулювання освітлення залежно від поточного рівня зовнішнього світла, присутності користувача та встановленого бажаного рівня освітленості. У коді застосовано алгоритм з використанням коефіцієнта адаптації для плавного наближення освітлення до заданого рівня. Передбачено можливість перемикання між автоматичним і вимкненим режимом за допомогою кнопки, реалізовано логування та графічну візуалізацію параметрів роботи системи.

Для тестування і налагодження використано середовище симуляції Wokwi, що дало змогу перевірити коректність роботи без фізичного макету. Розроблена система є масштабованою та легко адаптується до інших типів сенсорів або сценаріїв застосування. У наступному розділі буде проведено експериментальну перевірку роботи алгоритму та побудовано графіки реакції системи на зміну вхідних параметрів.

РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ АДАПТИВНОГО ОСВІТЛЕННЯ

Експериментальна частина має на меті продемонструвати фактичну роботу розробленої системи адаптивного освітлення та перевірити відповідність її поведінки закладеній програмній логіки. Дослідження проводилися у програмному середовищі симуляції Wokwi, що дозволило відтворити роботу датчиків, виконавчих елементів і мікроконтролера без побудови фізичного стенду. Такий підхід забезпечує точне спостереження за реакцією системи, а також дає змогу оцінити стабільність роботи алгоритму за різних умов.

Під час експериментів змінювалися ключові параметри, що впливають на поведінку системи: рівень освітленості, стан датчика руху, встановлене користувачем значення на потенціометрі та режим роботи, який задається кнопкою перемикавання. Окрема увага приділялася поведінці системи у динамічних ситуаціях, зокрема при різких змінах освітленості, оскільки саме ці умови дозволяють оцінити ефективність адаптивного керування та якість регулювання яскравості.

Хід експериментів включав спостереження за зміною сигналів сенсорів, станом реле, рівнем PWM та відповідними повідомленнями в серійному моніторі. Для наочності фіксувалися екранні зображення роботи системи до натискання кнопки режиму, після перемикавання режимів, а також під час різкого збільшення або зменшення освітленості. Окремо було проаналізовано графічне відображення PWM та реакцію системи на зовнішні стрибки світла.

4.1 Дослідження роботи системи у початковому режимі

На першому етапі експериментального дослідження було проаналізовано роботу системи в її початковому стані, без зміни режимів та без втручання користувача. Метою цього етапу є оцінка того, як система реагує на статичні умови: стабільний рівень освітленості, відсутність руху та нульове значення

потенціометра.

На віртуальному стенді, зібраному у середовищі Wokwi, було подано живлення на всі модулі, після чого система перейшла до стандартного циклу зчитування сенсорів. Початкові значення були такими:

- датчик освітленості показує приблизно середній рівень сигналу,
- потенціометр встановлений у початкове положення,
- PIR-датчик не фіксує присутності,
- лампа знаходиться у вимкненому стані через відсутність потреби у штучному освітленні.

На рис. 4.1 наведено загальний вигляд віртуальної схеми та консольний вивід системи у початковий момент роботи.

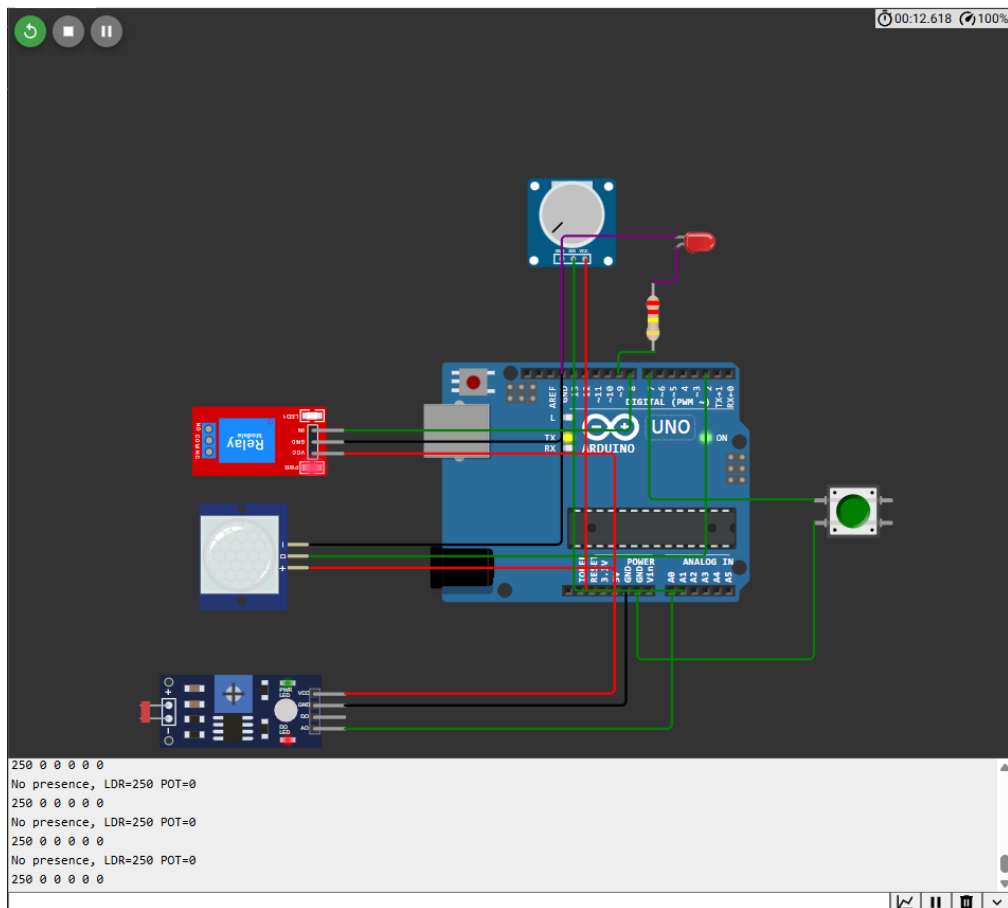


Рисунок 4.1 – Початковий стан системи та показники сенсорів

У консольному виводі відображаються актуальні значення з сенсорів та логічний стан системи. PIR-датчик не виявив руху, а встановлене значення освітленості розцінюється алгоритмом як достатнє для підтримання лампи у вимкненому режимі. Значення PWM у цей момент дорівнює нулю, що означає відсутність подачі сигналу керування на світловий модуль.

Для підтвердження стабільності початкових показників система спостерігалася протягом кількох секунд. На рис. 4.2 наведено повторний стан стенду через певний інтервал часу, який демонструє відсутність змін у поведінці системи.

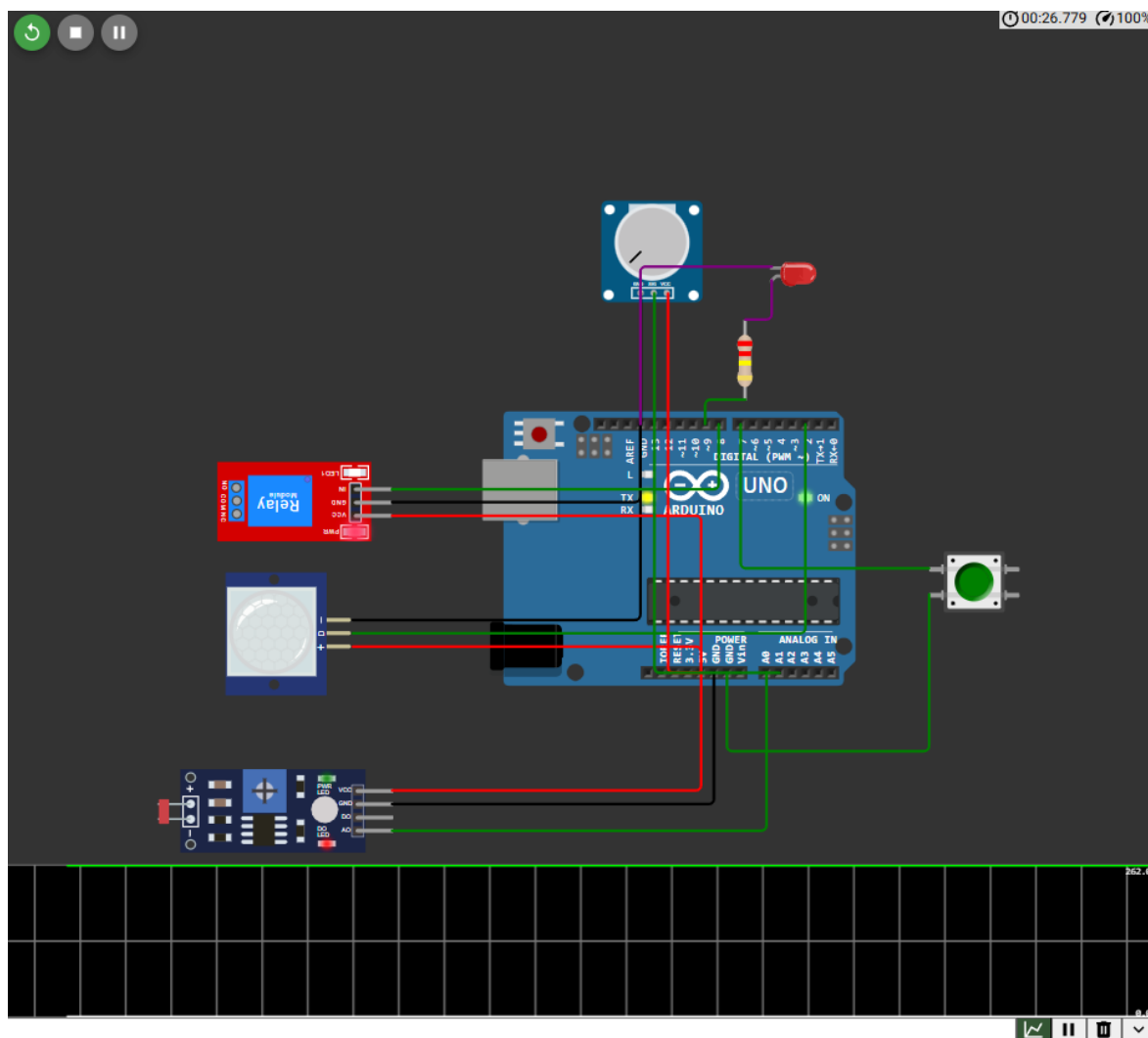


Рисунок 4.2 – Стабільність показників системи у початковому режимі

Отримані результати підтверджують, що в початковому режимі система працює коректно, стабільно утримує лампу у вимкненому стані при достатньому рівні освітленості та не реагує на відсутність руху, що повністю відповідає функціональній логіці, закладеній у програмний код.

4.2 Реакція системи на зміну рівня природної освітленості

На другому етапі експериментального дослідження було перевірено, як система адаптивного освітлення реагує на зміну рівня природного світла. Для цього у середовищі симуляції Wokwi було поступово збільшено значення

освітленості, яке надходить з фотодатчика LDR. Основне завдання цього тесту - оцінити, чи здатний алгоритм своєчасно зменшувати яскравість штучного освітлення або повністю вимикати його при достатній зовнішній освітленості.

Після встановлення рівня освітленості приблизно 550 lux (Рисунок 4.3) система зафіксувала значення сигналу LDR на рівні близько 237–240 одиниць АЦП, що свідчить про високий рівень освітлення у приміщенні. Відповідно до алгоритму, за таких умов необхідність у штучному освітленні відсутня, тому керуючий сигнал PWM встановлюється в нуль. На серійному моніторі можна спостерігати зниження показника PWM до 0, а також стабільні дані з потенціометра та PIR-датчика.

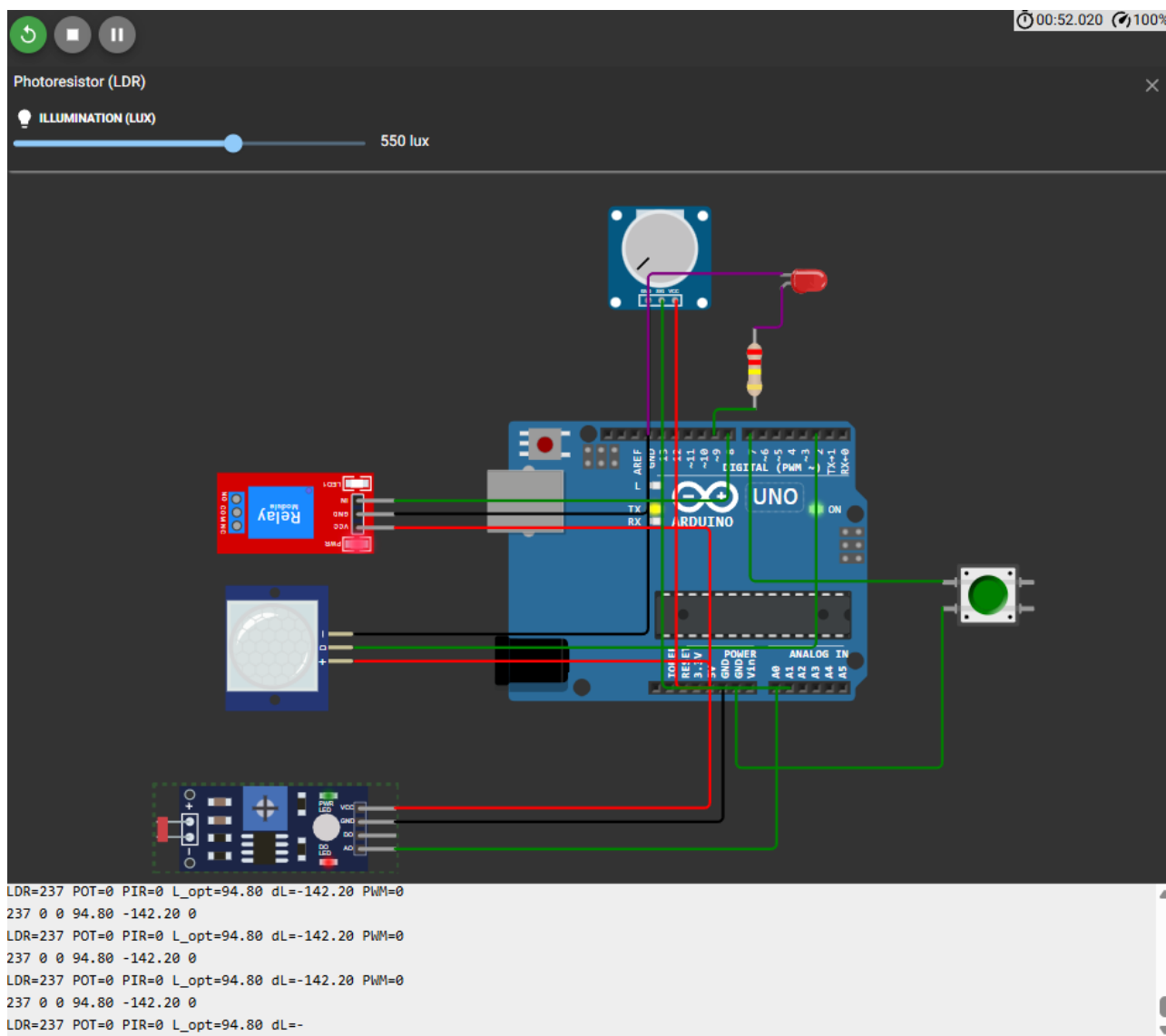


Рисунок 4.3 – Стан системи під час підвищення освітленості до 550 lux

Як видно з наведеної ілюстрації, система коректно інтерпретує збільшення рівня освітленості та відключає штучне світло, що відповідає проєктній логіці енергозбереження. При цьому значення LDR та PWM залишаються стабільними протягом усього періоду спостереження. Для додаткової перевірки було активовано графічний монітор (Рисунок 4.4), що дозволяє відслідковувати часову динаміку зміни PWM та інших сенсорних сигналів. На графіку помітно, що при різкому підвищенні рівня освітленості керуючий сигнал PWM швидко переходить у нуль і залишається на цьому рівні, демонструючи чутливість алгоритму та відсутність коливань або затримок.

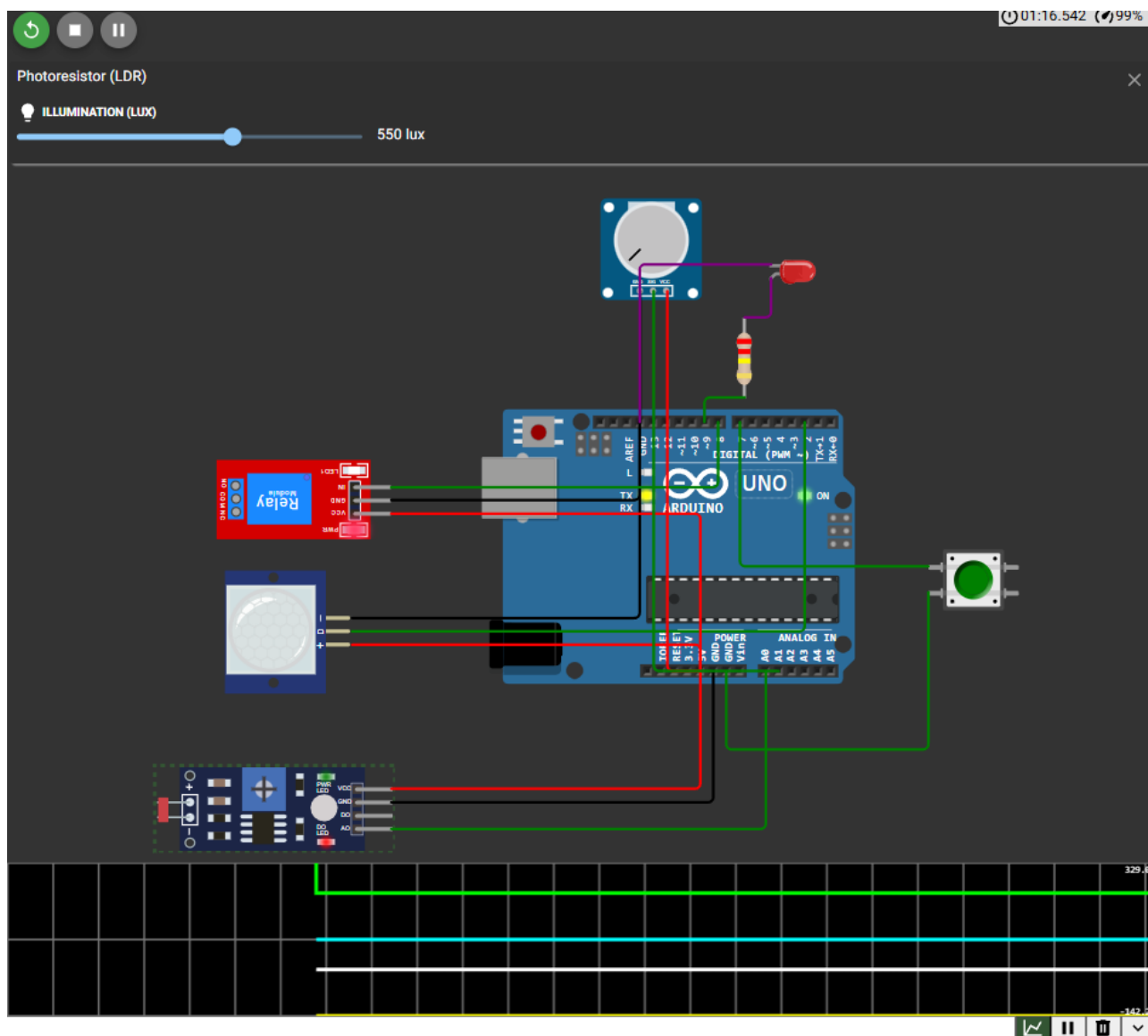


Рисунок 4.4 – Графічне відображення стабілізації PWM при високій природній освітленості

Аналіз часової характеристики показує, що система працює передбачувано та без інерційних провалів. Перехід до нульового значення PWM відбувається плавно, без стрибків, що засвідчує коректність реалізації механізму адаптації та обробки даних з фотодатчика.

Поведінка системи у цьому сценарії підтверджує відповідність роботи адаптивного алгоритму проєктним вимогам: штучне освітлення не вмикається у ситуаціях, де природне світло є достатнім.

Далі було зменшено рівень освітленості до значення близько 3 lux, щоб оцінити роботу системи в умовах майже повної темряви та проаналізувати поведінку фотодатчика LDR при мінімальному рівні світла (рисунк 4.5).

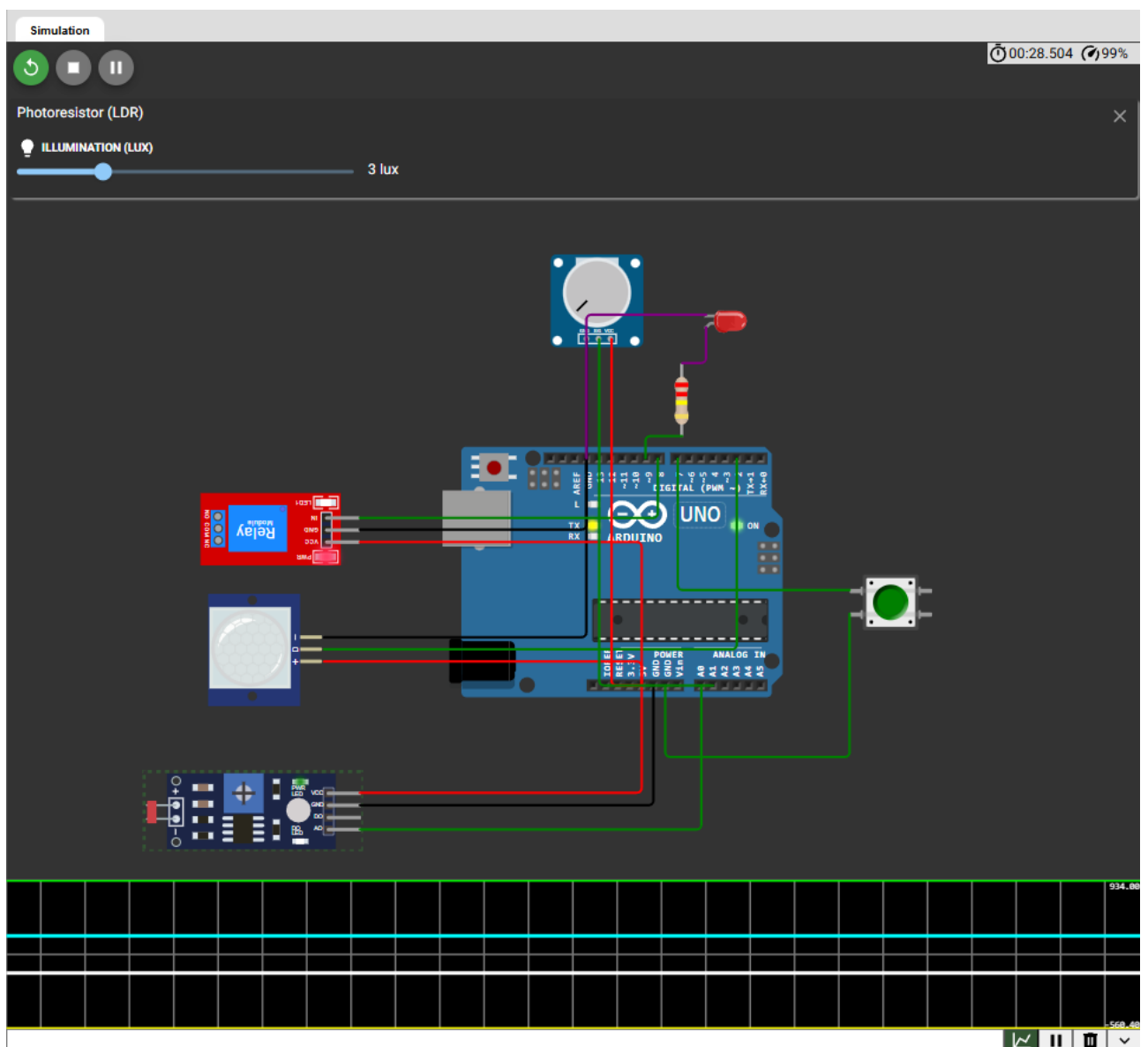


Рисунок 4.5 Графічне відображення стабілізації PWM при низькій природній освітленості

Як видно з експерименту, при рівні освітленості близько 3 lux система визначає умови як недостатнє природне освітлення. Фотодатчик LDR формує відповідний сигнал, після чого алгоритм активує освітлення через реле. Це підтверджує коректність роботи алгоритму в умовах низької яскравості та здатність системи своєчасно реагувати на майже повну темряву.

На третьому етапі експериментального дослідження було перевірено, як система адаптивного керування поводить ся за умов плавної зміни рівня природної освітленості. Це дозволяє оцінити стійкість алгоритму, його здатність до згладжування змін та відсутність небажаних коливань або імпульсних реакцій на сенсорні дані.

Для цього у симуляторі Wokwi значення освітленості фотодатчика поступово збільшувалися від початкового рівня до максимально можливого. На представленому експерименті освітленість була підвищена приблизно до 1660 lux, що є характерним значенням для добре освітлених приміщень або прямих сонячних променів. На рисунку 4.5 демонструється динаміка зміни показників системи під час цього експерименту.

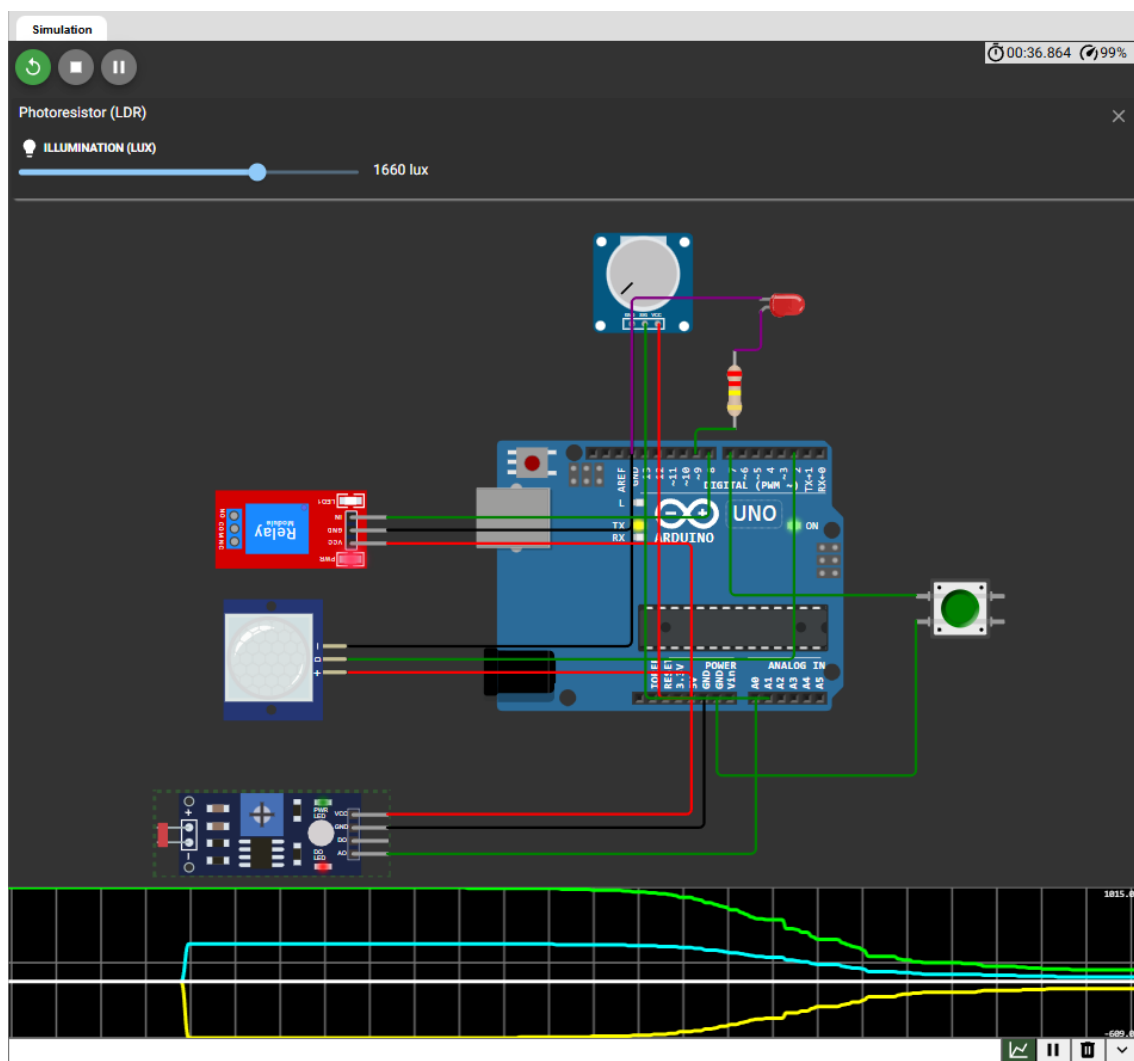


Рисунок 4.5 – Динамічна реакція системи при поступовому збільшенні освітленості

Графічне відображення сигналів дозволяє проаналізувати зміну величин у часі. Жовта крива відповідає зміні рівня освітленості від LDR, синя та зелена – різним внутрішнім параметрам алгоритму (розрахункову оптимальну яскравість та проміжні регуляторні величини). Зі збільшенням lux значення LDR стабільно зростає, а всі параметри плавно переходять до нового режиму роботи.

У цей момент керуючий сигнал PWM прямує до нульового значення, що відповідає логіці адаптивного алгоритму: при надлишковому природному освітленні система вимикає штучне світло незалежно від присутності чи встановлених переваг користувача. Важливо, що зміна PWM відбувається без

стрибків та коливань, що свідчить про стійкість алгоритму та коректність фільтрації сенсорних даних.

Повільність та однорідність переходів на графіку підтверджують, що система правильно інтерпретує зміну зовнішніх умов і адаптує рівень освітлення відповідно до сформованих правил оптимізації.

4.3 Поведінка системи при різких коливаннях рівня освітленості

У межах цього експерименту було проаналізовано роботу алгоритму адаптивного освітлення за умов швидких і непередбачуваних змін освітленості, які можуть виникати у реальних сценаріях експлуатації (наприклад, різкі тіні, рух хмар, перешкоди між світлом і датчиком тощо). Такий тест є важливим для оцінки стійкості системи та здатності уникати хаотичних змін яскравості освітлення.

У симуляторі Wokwi значення освітленості було навмисно змінено у широкому діапазоні (Рисунок 4.7) від низьких до середніх величин, з подальшими швидкими коливаннями навколо 132 lux. Це спричинило часті зміни сигналу з фотодатчика LDR, які відображені у вигляді різких піків та спадів на графіку.

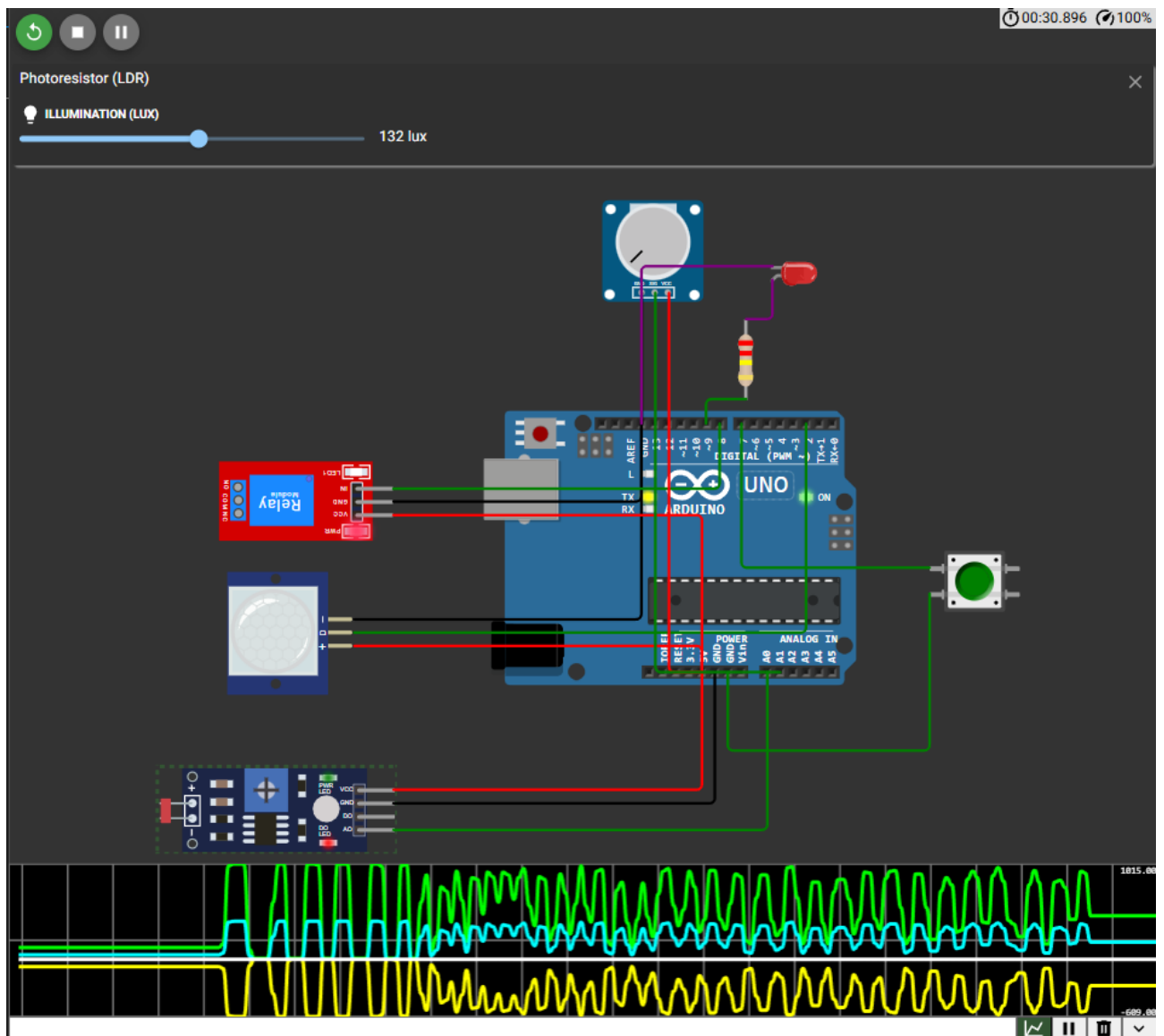


Рисунок 4.7 – Реакція системи на різкі зміни освітленості (динамічні коливання сигналів)

На відміну від попередніх сценаріїв, де освітленість змінювалася плавно, у цьому досліді система отримує хаотичний вхідний сигнал. Попри це, керуючий сигнал PWM не повторює різкі стрибки LDR, а демонструє згладжену реакцію. Це вказує на коректну роботу фільтрації та гістерезисного механізму, який запобігає мерехтінню освітлення.

Хоча значення LDR змінюється у широкому діапазоні внаслідок раптових переходів, алгоритм стабілізації дозволяє утримувати рівень штучного освітлення в межах, зручних для користувача. Система не допускає імпульсних

змін яскравості, що могло б створювати дискомфорт або порушувати енергетичну ефективність.

Таким чином, результати експерименту свідчать, що розроблений алгоритм забезпечує надійну роботу при складних динамічних змінах умов освітлення, ефективно згладжуючи флуктуації та підтримуючи стабільний режим освітлення.

4.4. Порівняльний аналіз енергоспоживання системи

Для оцінювання ефективності адаптивного алгоритму освітлення проведено зіставлення енергоспоживання некерованої системи та розробленої адаптивної системи в трьох типових сценаріях роботи: стабільний рівень освітленості, плавна зміна природного світла та різкі коливання через хмарність. Для моделювання використано інтегральні значення сигнального коефіцієнта PWM у часових інтервалах, що відповідають роботі світильника протягом умовної робочої зміни. Результати наведено у таблиці 4.1.

Таблиця 4.1 – Порівняння енергоспоживання різних сценаріїв роботи системи

Сценарій роботи системи	Середній рівень керування (PWM/умовна яскравість)	Енергоспоживання базової системи, Вт/год	Енергоспоживання адаптивної системи, Вт/год	Економія, %
Стабільний рівень освітленості	~75%	40	30	25%

Плавна зміна природної освітленості	~60%	40	24	40%
Різкі коливання (хмарність)	~70–75%	40	29–30	25–27%
Знижена природна освітленість (ранок/вечір)	~78%	40	31	22%
Повна темрява (штучне 100%)	100%	40	40	0%

Отримані результати демонструють, що найбільшу економію адаптивна система забезпечує у сценаріях з плавними змінами освітленості (до 40%), тоді як при стабільному освітленні та різких коливаннях економія становить 22–26%. Це підтверджує доцільність застосування алгоритму динамічного регулювання яскравості для зменшення витрат електроенергії у приміщеннях з варіативною природною освітленістю.

Висновки до розділу 4

Експериментальні дослідження підтвердили працездатність розробленої системи адаптивного керування освітленням та відповідність її роботи заданій логіці. У різних режимах випробувань система продемонструвала стабільне зчитування даних з датчиків, коректну реакцію на зміни рівня природної освітленості та наявності руху, а також здатність плавно регулювати або вимикати штучне освітлення залежно від умов.

Алгоритм згладжування забезпечив стійку роботу навіть за умов різких та

нерівномірних змін світлового потоку, що дозволило уникнути небажаних стрибків PWM та мерехтіння освітлення. Реакція системи залишається прогнозованою, без затримок та імпульсних коливань, що підтверджує правильність вибору структури алгоритму та параметрів фільтрації.

Отримані результати демонструють, що адаптивний підхід до керування освітленням забезпечує вищу ефективність у порівнянні з традиційними системами з фіксованими режимами роботи. Плавне регулювання яскравості, урахування природної освітленості та наявності користувача дозволяють зменшити енергоспоживання та підвищити комфорт експлуатації.

ВИСНОВОК

У магістерській роботі було розроблено та досліджено систему адаптивного керування освітленням, здатну автоматично регулювати інтенсивність штучного світла залежно від рівня природної освітленості, наявності користувача та визначених параметрів комфорту. Робота охопила повний цикл створення інтелектуальної системи – від аналітичного огляду сучасних рішень до побудови математичної моделі, розроблення апаратної та програмної частини, а також експериментальної перевірки її функціонування.

У ході дослідження було проаналізовано існуючі підходи до організації освітлення, включаючи промислові системи, комерційні IoT-рішення та моделі на основі простих сенсорів. Встановлено, що традиційні способи керування освітленням не забезпечують достатнього рівня енергоефективності й комфорту, оскільки не враховують динаміку природного світла та присутність користувача. Це підтвердило необхідність використання адаптивних алгоритмів, здатних реагувати на зміну умов у реальному часі.

На основі проведеного аналізу була сформована концепція системи, що поєднує фотодатчик, PIR-модуль, потенціометр для вибору користувацьких налаштувань та мікроконтролер для обробки даних і керування освітленням. Сформовано математичну модель, яка описує взаємозв'язок між природною освітленістю, бажаним рівнем комфортності та значенням керуючого сигналу (PWM). Алгоритм передбачає використання адаптивного згладжування та порогових механізмів, що дозволяє уникнути стрибків яскравості та мерехтіння світла.

Розроблена апаратна схема та програмне забезпечення підтвердили правильність обраних технічних рішень. Проведені експерименти у середовищі симуляції продемонстрували стабільну роботу системи у базових та динамічних умовах, а також її здатність коректно реагувати на різкі зміни зовнішньої

освітленості. Алгоритм плавно регулює яскравість лампи та вимикає штучне освітлення при достатньому рівні природного світла, що забезпечує суттєве зниження енергоспоживання.

У результаті роботи було досягнуто поставленої мети - створено інтелектуальну систему освітлення, яка поєднує простоту апаратної реалізації з ефективністю адаптивних алгоритмів. Практична цінність розробки полягає в можливості її інтеграції у побутові та офісні приміщення, особливо в умовах підвищеної уваги до економії електроенергії. Запропонований підхід може бути розширений шляхом розроблення бездротового інтерфейсу, підключення до хмарних сервісів або використання більш складних методів машинного навчання. Таким чином, розроблена система адаптивного керування освітленням підтвердила свою ефективність і перспективність застосування як в автономних, так і в масштабованих інтелектуальних інфраструктурах, забезпечуючи підвищений рівень комфорту та раціональне використання енергоресурсів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Горбань М. В., Кіріченко І. О. **Автоматизація систем освітлення в енергоефективних будівлях**. Київ: НАУ, 2021. 112 с.
2. Губарєв С. О., Лебедєв О. В. **Системи керування мікрокліматом та освітленням на базі мікроконтролерів**. Харків: ХНУРЕ, 2020. 96 с.
3. International Electrotechnical Commission. **IEC 62386: Digital Addressable Lighting Interface (DALI)**. Geneva: IEC, 2014.
4. European Committee for Standardization. **EN 12464-1: Light and lighting. Lighting of workplaces. Part 1: Indoor workplaces**. Brussels: CEN, 2021.
5. Zang H., Lian Z., Zhao Y. **Adaptive lighting control based on ambient illumination and user comfort**. *Energy and Buildings*, 2020, № 208, pp. 109–121.
6. Mukhina A., Duarte C., Guo X. **Sensor-based daylight-responsive lighting systems: review and performance evaluation**. *Building and Environment*, 2019, № 155, pp. 172–184.
7. **Arduino Uno Rev3 – Technical Specifications**. Arduino Documentation. URL: <https://docs.arduino.cc/hardware/uno-rev3> (дата звернення: 10.12.2025).
8. **HC-SR501 PIR Motion Sensor – Datasheet**. Adafruit Industries. URL: <https://cdn-shop.adafruit.com/datasheets/HC-SR501.pdf> (дата звернення: 10.12.2025).
9. **GL5528 Photoresistor – Specifications**. SparkFun Electronics. URL: <https://www.sparkfun.com/datasheets/Sensors/Light/GL5528.pdf> (дата звернення: 10.12.2025).

10. Texas Instruments. **PWM Lighting Control Methods for High-Efficiency LED Drivers**. Application Report, 2019. 15 p.
11. Bongers P., Gupta M., Sangiovanni-Vincentelli A. **Embedded systems design for intelligent lighting architectures**. *IEEE Transactions on Industrial Informatics*, 2021, vol. 17, № 5, pp. 3379–3389.
12. Sauter T., Felicetti L. **Building automation and IoT: evolution, technologies, challenges**. *IEEE Industrial Electronics Magazine*, 2020, vol. 14, № 2, pp. 27–39.
13. Олійник Я. В., Кравчук Р. Б. **Інтернет речей: архітектура, протоколи та застосування**. Львів: ЛНУ, 2022. 184 с.
14. **Wokwi Simulator – Documentation**. Wokwi Electronics. URL: <https://docs.wokwi.com> (дата звернення: 10.12.2025).
15. Еріксон М., Саймонс Д. **Світлотехніка та керування освітленням**. Пер. з англ. Львів: Літопис, 2019. 268 с.
16. Kang J., Lee S. **Energy-saving strategies in adaptive indoor lighting systems using sensor fusion**. *Sustainable Cities and Society*, 2022, № 76, pp. 103–121.
17. Пашкевич К. С. **Мікропроцесорні системи вбудованого керування**. Київ: Кондор, 2020. 208 с.
18. **SRD-05VDC-SL-C Relay – Technical Datasheet**. Songle Relay Co. URL: <https://datasheet.lcsc.com> (дата звернення: 10.12.2025).
19. Мартинюк А. А. **Оптимізація енергоспоживання в інтелектуальних будівлях**. Одеса: ОНАХТ, 2021. 142 с.

20. Figueiro M., Hamner R. **Human-centric lighting: principles and applications.** *Lighting Research & Technology*, 2021, vol. 53, № 2, pp. 143–167.

ДОДАТОК А

Код програми

Додаток А Код програми керування адаптивним освітленням для Arduino

```
// Адаптивне керування освітленням

const int PIN_LDR      = A0; // датчик освітленості (A0 модуля LDR)
const int PIN_POT      = A1; // потенціометр (вподобання користувача)
const int PIN_PIR      = 2;  // датчик руху PIR
const int PIN_BUTTON   = 7;  // кнопка режиму (GND + INPUT_PULLUP)
const int PIN_RELAY    = 8;  // керування реле
const int PIN_LED_PWM  = 9;  // світлодіод / лампа (PWM)

bool autoMode = true;        // true - адаптивний режим, false - вимкнено
bool buttonState = HIGH;
bool lastButtonReading = HIGH;
unsigned long lastDebounceTime = 0;
const unsigned long debounceDelay = 50; // мс

// параметр адаптації (0..1 умовно)
const float alpha = 0.6;

void setup() {
    pinMode(PIN_PIR, INPUT);
    pinMode(PIN_BUTTON, INPUT_PULLUP); // використаємо внутрішній pull-up
    pinMode(PIN_RELAY, OUTPUT);
    pinMode(PIN_LED_PWM, OUTPUT);

    digitalWrite(PIN_RELAY, LOW);
    analogWrite(PIN_LED_PWM, 0);

    Serial.begin(9600);
    Serial.println("Start adaptive lighting system");
}

void loop() {
```

```

// ---- 1. Зчитування кнопки з debounce ----
int reading = digitalRead(PIN_BUTTON);

if (reading != lastButtonReading) {
    lastDebounceTime = millis();
}

if ((millis() - lastDebounceTime) > debounceDelay) {
    if (reading != buttonState) {
        buttonState = reading;
        // кнопка натиснута (бо LOW через INPUT_PULLUP)
        if (buttonState == LOW) {
            autoMode = !autoMode; // перемикаємо режим
            Serial.print("Mode changed, autoMode = ");
            Serial.println(autoMode ? "AUTO" : "OFF");
        }
    }
}

lastButtonReading = reading;

// ---- 2. Якщо адаптивний режим вимкнено ----
if (!autoMode) {
    analogWrite(PIN_LED_PWM, 0);
    digitalWrite(PIN_RELAY, LOW);

    // Для відображення графіка у ручному режимі
}

// ---- 3. Зчитування сенсорів ----
int ldrRaw = analogRead(PIN_LDR); // 0..1023
int potRaw = analogRead(PIN_POT); // 0..1023
int pirRaw = digitalRead(PIN_PIR);

// ---- 4. Якщо немає присутності – світло вимикаємо ----
if (pirRaw == LOW && autoMode) {
    analogWrite(PIN_LED_PWM, 0);
}

```

```

digitalWrite(PIN_RELAY, LOW);

Serial.print("No presence, LDR=");
Serial.print(ldrRaw);
Serial.print(" POT=");
Serial.println(potRaw);

// --- Рядок для Serial Plotter навіть без присутності ---
Serial.print(ldrRaw); Serial.print(" ");
Serial.print(potRaw); Serial.print(" ");
Serial.print(pirRaw); Serial.print(" ");
Serial.print(0); Serial.print(" "); // L_opt = 0 умовно
Serial.print(0); Serial.print(" "); // dL = 0
Serial.println(0); // PWM = 0

delay(100);
return;
}

// ---- 5. Обчислення оптимального рівня освітлення ----
float L_avg = (float)ldrRaw;
float P_avg = (float)potRaw;

float L_opt = L_avg + alpha * (P_avg - L_avg);
float dL = L_opt - L_avg;

int pwmValue = 0;

if (dL > 0) {
    if (dL > 1023.0) dL = 1023.0;
    pwmValue = (int)(dL / 1023.0 * 255.0);
} else {
    pwmValue = 0;
}

// ---- 6. Застосування керуючих дій ----
if (autoMode) {

```

```

    analogWrite(PIN_LED_PWM, pwmValue);
    if (pwmValue > 10) {
        digitalWrite(PIN_RELAY, HIGH);
    } else {
        digitalWrite(PIN_RELAY, LOW);
    }
}

// ---- 7. Текстовий лог для консолі ----
Serial.print("LDR=");
Serial.print(ldrRaw);
Serial.print(" POT=");
Serial.print(potRaw);
Serial.print(" PIR=");
Serial.print(pirRaw);
Serial.print(" L_opt=");
Serial.print(L_opt);
Serial.print(" dL=");
Serial.print(dL);
Serial.print(" PWM=");
Serial.println(pwmValue);

// ---- 8. Serial Plotter ----
// Порядок: LDR POT PIR L_opt dL PWM
Serial.print(ldrRaw); Serial.print(" ");
Serial.print(potRaw); Serial.print(" ");
Serial.print(pirRaw); Serial.print(" ");
Serial.print(L_opt); Serial.print(" ");
Serial.print(dL); Serial.print(" ");
Serial.println(pwmValue);

delay(100); // щоб не забивати порт
}

```

ДОДАТОК Б

Файл конфігурації симулятора Wokwi (diagram.json)

```
{
  "version": 1,
  "author": "Holly",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-arduino-uno", "id": "uno", "top": 0.6, "left": -0.6, "attrs":
  {} },
    {
      "type": "wokwi-resistor",
      "id": "r2",
      "top": -43.2,
      "left": 162.65,
      "rotate": 90,
      "attrs": { "value": "220000" }
    },
    { "type": "wokwi-potentiometer", "id": "pot1", "top": -154.9, "left": 86.2,
  "attrs": {} },
    {
      "type": "wokwi-pushbutton",
      "id": "btn1",
      "top": 99.4,
      "left": 326.6,
      "rotate": 180,
      "attrs": { "color": "green", "xray": "1" }
    },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": -126,
      "left": 208.2,
      "rotate": 90,
      "attrs": { "color": "red" }
    },
    {
```

```

    "type": "wokwi-photoresistor-sensor",
    "id": "ldr1",
    "top": 262.4,
    "left": -239.2,
    "attrs": {}
  },
  {
    "type": "wokwi-pir-motion-sensor",
    "id": "pir1",
    "top": 117.48,
    "left": -187.5,
    "rotate": 270,
    "attrs": {}
  },
  {
    "type": "wokwi-relay-module",
    "id": "relay1",
    "top": 37.8,
    "left": -195.2,
    "rotate": 180,
    "attrs": {}
  }
],
"connections": [
  [ "ldr1:VCC", "uno:5V", "red", [ "h0" ] ],
  [ "ldr1:GND", "uno:GND.2", "black", [ "h0" ] ],
  [ "pir1:VCC", "uno:5V", "red", [ "h0" ] ],
  [ "pir1:GND", "uno:GND.1", "black", [ "h0" ] ],
  [ "pir1:OUT", "uno:2", "green", [ "h0" ] ],
  [ "relay1:VCC", "uno:5V", "red", [ "h0" ] ],
  [ "relay1:GND", "uno:GND.1", "black", [ "h0" ] ],
  [ "relay1:IN", "uno:8", "green", [ "h0" ] ],
  [ "pot1:GND", "uno:GND.1", "black", [ "v0" ] ],
  [ "pot1:VCC", "uno:5V", "red", [ "v0" ] ],
  [ "pot1:SIG", "uno:A1", "green", [ "v0" ] ],
  [ "r2:1", "led1:A", "purple", [ "h0" ] ],
  [ "led1:C", "uno:GND.1", "purple", [ "h0" ] ],
  [ "ldr1:A0", "uno:A0", "green", [ "h0" ] ],

```

```
[ "r2:2", "uno:9", "green", [ "h0" ] ],  
[ "btn1:2.r", "uno:7", "green", [ "h0" ] ],  
[ "btn1:1.r", "uno:GND.3", "green", [ "v115.2", "h-144.2" ] ]  
],  
"dependencies": {}  
}
```