ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Автоматизована система адаптивного освітлення в Smart Home із застосуванням мікрокомп'ютера Raspberry Pi»

на здобуття освітнього ступеня магістра
зі спеціальності 123 Комп'ютерна інженерія
освітньо-професійної програми «Комп'ютерні системи та мережі»

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Владислав ТКАЧУК
(підпис)

Виконав: здобувач вищої освіти групи КСДМ-61
Владислав Ткачук

Керівник: Віталій ВОЛОХІН
доцент кафедри, к.т.н., доцент

Рецензент: _____
науковий ступінь, вчене звання *Ім'я, ПРІЗВИЩЕ*

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерна інженерія

Ступінь вищої освіти Магістр

Спеціальність 123 Комп'ютерна інженерія

Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Комп'ютерної інженерії
Наталія ЛАЩЕВСЬКА
«_____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Ткачуку Владиславу Олександровичу

1. Тема кваліфікаційної роботи: «Автоматизована система адаптивного освітлення в Smart Home із застосуванням мікрокомп'ютера Raspberry Pi»

керівник кваліфікаційної роботи Віталій ВОЛОХІН, доцент кафедри, к.т.н., доцент, затверджені наказом Державного університету інформаційно-комунікаційних технологій від «___» _____ 2025 р. № _____.

2. Строк подання кваліфікаційної роботи «26» грудня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, підходи до написання скриптів для мікрокомп'ютерів, дослідження економічної доцільності проєкту, вимоги до рівня освітленості робочих об'єктів.

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

1. Дослідження відомих систем та засобів автоматизованого освітлення.
2. Аналіз технологій пропрієтарних рішень сфери.

3. Розробка та аналіз автоматизованої системи адаптивного освітлення на мікрокомп’ютері.

5. Перелік ілюстративного матеріалу: *презентація*

1. Компонентна база проєкту.
2. Схематика апаратної частини системи.
3. Програмна частина системи.
4. Керування системою освітлення.
5. Втілення проєкту.
6. Аналіз економічного компонента.
7. Модернізація адаптивності (1).
8. Модернізація адаптивності (2).

6. Дата видачі завдання «01» вересня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	10.09.2025	Виконано
2	Вивчення матеріалів для аналізу розвитку технологій автоматизованого освітлення	15.09.2025	Виконано
3	Дослідження доступних мікрокомп’ютерних рішень	01.10.2025	Виконано
4	Аналіз особливостей розробки та інтеграції систем з мікрокомп’ютерів у сферу Smart Home	10.10.2025	Виконано
5	Розробка та конструювання системи автоматизованого адаптивного освітлення	02.11.2025	Виконано
6	Економічний аналіз та розробка можливих модифікацій системи	14.11.2025	Виконано
7	Оформлення роботи: вступ, висновки, реферат	15.11.2025	Виконано
8	Розробка демонстраційних матеріалів	16.11.2025	Виконано
9	Попередній захист роботи	16.12.2025	Виконано

Здобувач вищої освіти _____ Владислав ТКАЧУК
(підпис)

Керівник
кваліфікаційної роботи _____ Віталій ВОЛОХІН
(підпис)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 89 стор., 5 табл., 57 рис., 50 джерел.

Мета роботи – мінімізація вартості модуля автоматизованого контролю світлом, за рахунок використання мікрокомп'ютера Raspberry Pi та його модулів.

Об'єкт дослідження – процес автоматизації контролю рівня освітлення.

Предмет дослідження – технології автоматизованого контролю рівня освітлення на базі модулів мікрокомп'ютерів.

Короткий зміст роботи: У роботі розглянуто розробку та впровадження систем та модулів автоматичного контролю рівня освітлення у приміщеннях на базі мікрокомп'ютера Raspberry Pi 4.

Проведено аналіз сучасних методів розробки автоматизованих систем адаптивного освітлення на базі мікрокомп'ютерів.

Розроблено та оптимізовано модуль адаптивного освітлення для Smart Home.

Проведено експерименти для перевірки та подальшого аналізу економічної складової проєкту в порівнянні з пропрієтарними рішеннями.

КЛЮЧОВІ СЛОВА: АДАПТИВНЕ ОСВІТЛЕННЯ, РОЗУМНИЙ БУДИНОК, RASPBERRY PI, ІНТЕРНЕТ РЕЧЕЙ (IOT), ЕНЕРГОЕФЕКТИВНІСТЬ, МІКРОКОМП'ЮТЕР, АВТОМАТИЗАЦІЯ, МОДУЛЬ.

ABSTRACT

Text part of the master's qualification work: 89 pages, 57 pictures, 5 tables, 50 sources.

The purpose of the work is to minimize the cost of an automated light control module by using the Raspberry Pi microcomputer and its modules.

Object of research – lighting level control automation process.

Subject of research – technology of automated lighting level control based on Smart Home modules.

Summary of the work: The work considers the development and implementation of systems and modules for automatic control of the level of lighting in rooms based on the Raspberry Pi 4 microcomputer.
An analysis of modern methods for developing automated adaptive lighting systems based on microcomputers was conducted.
An adaptive lighting module for Smart Home was developed and optimized.
Experiments were conducted to verify and further analyze the economic component of the project in comparison with proprietary solutions.

KEYWORDS: ADAPTIVE LIGHTING, SMART HOME, RASPBERRY PI, INTERNET OF THINGS (IOT), ENERGY EFFICIENCY, MICROCOMPUTER, AUTOMATION, MODULE.

ЗМІСТ

ВСТУП 9
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ АВТОМАТИЗОВАНИХ СИСТЕМ ОСВІТЛЕННЯ 11
1.1 Поняття та класифікація автоматизованих систем 11
1.2 Принципи та визначення систем освітлення 14
1.3 Сучасні технології автоматизації освітлення 19
1.4 Адаптивні системи освітлення 24
РОЗДІЛ 2 МІКРОКОМП'ЮТЕРИ ТА МОДУЛІ У СФЕРІ ТЕХНОЛОГІЙ РОЗУМНОГО ДОМУ (SMART HOME) 27
2.1 Класифікація та введення у мікрокомп'ютери Raspberry Pi 27
2.2 Класифікація та порівняння виробників мікрокомп'ютерів 47
2.3 Технології та модулі автоматизації освітлення в Smart Home 56
2.4 Визначення та аналіз використовуваної технології 65
РОЗДІЛ 3 РОЗРОБКА ТА АНАЛІЗ СИСТЕМИ АВТОМАТИЗАЦІЇ КЕРУВАННЯ ОСВІТЛЕННЯМ 68
3.1 Вибір та технічні характеристики компонентної бази 68
3.2 Побудова та налаштування апаратно-програмного комплексу 75
3.3 Розробка програмного забезпечення та алгоритмів керування 85
3.4 Техніко-економічне обґрунтування розробки 87
ВИСНОВКИ 95
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 98
Додаток А 104
Додаток Б 107
Додаток В. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ 111

ВСТУП

У сучасному світі стрімкий розвиток цифрових технологій, інтернету речей (IoT) та мікрокомп'ютерних систем суттєво впливає на побут людини, змінюючи традиційні підходи до організації житлового простору. Концепція «розумного дому» (Smart Home) стала одним із найкращих прикладів інтеграції інтелектуальних систем у повсякденне життя. Така концепція передбачає використання автоматизованих пристроїв та програмних рішень для підвищення рівня комфорту, безпеки, енергоефективності та зручності користувача. Одним із напрямів цієї концепції є автоматизація систем освітлення, адже саме освітлення безпосередньо впливає на якість життя, емоційний стан людини, продуктивність праці та загальний рівень енергоспоживання в будинку.

Традиційні системи освітлення зазвичай потребують ручного керування, що часто призводить до нераціонального використання електроенергії. Людина не завжди вимикає світло при виході з кімнати або не змінює рівень яскравості відповідно до умов навколишнього середовища. Це знижує енергоефективність та створює додаткові фінансові витрати. Тому актуальним завданням сучасної техніки є створення інтелектуальних систем освітлення, які б автоматично реагували на зміну умов освітленості, наявність людей у приміщенні, час доби та інші фактори.

Застосування мікрокомп'ютера Raspberry Pi відкриває широкі можливості для побудови таких систем. Цей пристрій поєднує в собі невисоку вартість, гнучкість у налаштуванні, підтримку великої кількості периферійних сенсорів та широкий вибір програмних інструментів. На його основі можна створити ефективну автоматизовану систему адаптивного освітлення, що не потребує постійного втручання користувача та забезпечує оптимальний баланс між комфортом і економією енергії.

Адаптивне освітлення є не лише технічним нововведенням, а й важливою складовою сучасної екосистеми Smart Home. Воно дозволяє системі самостійно аналізувати рівень природного світла, визначати присутність людини, враховувати час доби та регулювати інтенсивність штучного освітлення відповідно до цих параметрів. Такий підхід сприяє не лише зниженню енергоспоживання, а й підвищенню якості життя мешканців, створюючи комфортне середовище для відпочинку, роботи чи навчання.

Апробація результатів та публікації, які були опубліковані в ході проведення дослідження:

1. Волохін В. В., Ткачук В. О. АЛЬТЕРНАТИВА ІСНУЮЧИМ ПРОФІЛЬНИМ МОДУЛЯМ КЕРУВАННЯ В SMART HOME: АНАЛІЗ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ МІКРОКОМП'ЮТЕРІВ RASPBERRY PI. «Проблеми комп'ютерної інженерії» : VI науково-практична конференція, м. Київ, 3 грудня 2025 р. С. 1–3. URL: https://duikt.edu.ua/ua/news-1-641-14852-kafedra-kompyuternoi-inzhenerii-zaproshue-do-uchasti-u-vi-naukovo-praktichniy-konferencii-problemi-kompyuternoi-inzhenerii_kafedra-kompyuternoi-inzhenerii.
2. Торошанко Я. І., Торошанко О. С., Куфтеріна С. Р., Ткачук В. О. Стаття «МАСШТАБОВАНИЙ БАГАТОСТУПЕНЕВИЙ ШИФРАТОР». Журнал «ТЕЛЕКОМУНІКАЦІЙНІ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ» : номер 1, м. Київ, 2025 р. С. 201–209. URL: https://tit.dut.edu.ua/index.php/telecommunication/article/view/2600.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ АВТОМАТИЗОВАНИХ СИСТЕМ ОСВІТЛЕННЯ

1.1 Поняття та класифікація автоматизованих систем

Автоматизація є однією з ключових поточних тенденцій розвитку сучасної техніки та технологій, що спрямована на зменшення рівня участі людини у процесах керування, чим зменшує відсоток «людського фактору» та підвищує ефективність функціонування технічних систем. Автоматизована система розуміється як сукупність технічних, програмних та інформаційних засобів, призначених для виконання певних операцій керування без або з мінімальним втручанням людини. У широкому сенсі це інтегрована система, яка реалізує процеси збору даних про стан об’єкта управління, їх обробки, прийняття рішень за заздалегідь визначеними алгоритмами та впливом на об’єкт через виконавчі механізми з метою досягнення заданих показників роботи (ефективності, надійності, безпеки тощо). Ключовими ознаками автоматизованої системи є наявність сенсорної підсистеми для вимірювання параметрів, контролера або процесора для обробки даних та формування управляючих сигналів, а також виконавчих пристроїв, що реалізують ці сигнали на фізичному рівні[1].

Структурно типова автоматизована система містить такі функціональні блоки як датчики і пристрої збору інформації, підсистема комунікацій для передачі даних, обчислювальну або логічну одиницю (контролер, мікрокомп’ютер), що реалізує алгоритми керування, виконавчі пристрої (реле, приводи, силові ключі), користувацькі інтерфейси та системи моніторингу, сховище даних і, опціонально, модулі аналітики або штучного інтелекту. Така архітектура забезпечує циклічність

процесу «вимір – обробка – дія – контроль», що є ознакою систем з замкнутим циклом керування. Класифікація автоматизованих систем здійснюється за кількома критеріями які доповнюють одне одного, відображають як їхню технічну організацію, так і характер функціонування[2]. Основні класифікації автоматизованих систем в контексті технологій Smart Home:

– класифікація за ступенем автономності та роллю оператора:

a) ручні (автоматизовані окремі функції) – системи, де основне рішення приймає людина, а автоматизація лише полегшує виконання операцій (наприклад, пульти та дистанційне керування),

b) автоматичні (напівавтоматичні) – системи, що виконують стандартні операції самостійно, проте допускають участь оператора у виняткових випадках,

c) автономні інтелектуальні – системи, що самостійно приймають рішення на основі адаптації та/або методів машинного навчання.

– класифікація за характером керування:

a) відкритого циклу (без зворотного зв'язку) – керування здійснюється за заздалегідь закладеним законом (правилом) без контролю результату,

b) закритого циклу (зі зворотним зв'язком) – рішення коригуються на основі вимірюваних параметрів об'єкта, що забезпечує стійкість та точність регулювання.

– класифікація за архітектурою управління:

a) централізовані – всі обчислення і прийняття рішень виконуються на єдиному вузлі (наприклад, центральний контролер будинку),

b) розподілені – функції поділені між кількома вузлами, що виконують локальне керування (типово для мережевих рішень на базі ZigBee, KNX тощо),

c) ієрархічні (багаторівневі) – поєднують локальне та централізоване управління: нижчі рівні виконують миттєве регулювання, вищі – стратегії та політики.

– класифікація за функціональною спрямованістю:

a) регуляторні – контролюють безперервні параметри (наприклад, рівень освітленості, яскравість),

b) дискретні – керують включенням/вимкненням або вибором режимів.

– класифікація за галуззю застосування та умовами експлуатації:

a) промислові, комерційні, побутові – відрізняються вимогами до надійності, швидкодії, інтерфейсів та стандартів,

b) критичні системи – вимагають підвищеної стійкості та резервування (наприклад, аварійне освітлення).

– класифікація за способом інтеграції в інформаційну інфраструктуру:

a) локальні автономні рішення – працюють незалежно від зовнішніх мереж,

b) мережеві – інтегруються у локальні та хмарні сервіси, забезпечуючи віддалений доступ, аналітику та оновлення.

Підсумовуючи, поняття автоматизованої системи охоплює як технічні елементи, так і алгоритмічні підходи до керування. Класифікація за наведеними ознаками дозволяє систематизувати вимоги до проєктування, вибору апаратної та програмної платформи, а також визначити критерії оцінки ефективності, які будуть застосовні при розробці автоматизованої системи освітлення на базі мікрокомп'ютера Raspberry Pi у наступних розділах цієї роботи.

1.2 Принципи та визначення систем освітлення

У будівлях і приміщеннях різних типів та призначення зазвичай використовують кілька базових типів освітлення. Загальне (основне) освітлення забезпечує рівномірне наповнення приміщення світлом та підтримує базовий рівень освітленості, воно створює фон для більш конкретних зонових освітлювальних завдань. Місцеве (робоче) освітлення концентрує світловий потік на конкретній робочій зоні або на об’єкті виконання завдання (наприклад, настільна лампа над робочим столом), воно підвищує рівень освітленості в зоні роботи, що сприяє підвищенню точності виконання та зниженню напруги зору. Акцентне (декоративне) освітлення виділяє архітектурні деталі, мистецькі об’єкти або декоративні елементи інтер’єру, змінюючи тон і настрій простору. Аварійне освітлення у свою чергу забезпечує мінімально необхідну освітленість для безпечної евакуації або аварійного обслуговування у разі відмови основної мережі живлення. У розумних системах (Smart Home) до освітлення нерідко додають і охоронне (чергове) підсвічування, яке активується під час руху чи виклику охоронної системи для покращення безпеки приміщення. Усі ці види освітлення часто комбінуються для створення необхідного світлового середовища, а вимоги до їх параметрів викладені в нормативних документах[3], на базі яких була побудована таблиця щодо вимог до освітлення робочих приміщень (див. таблицю 1.1).

Таблиця 1.1

Вимоги до освітлення робочих приміщень для зорових робіт високої якості згідно ДБН В.2.5-28:2018

Характеристика зорової роботи	Найменший або еквівалентний розмір об'єкта розрізнення, мм	Розряд зорової роботи	Підрозряд зорової роботи	Контраст об'єкта з фоном	Характеристика фону	Штучне освітлення		
						Освітленість, лк		
						при системі комбінованого освітлення		при системі загального освітлення
						всього	у т.ч. від загального	
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Найвищої точності	Менше ніж 0,15	I	а	малий	темний	5000, 4500	500,500	-,-
			б	малий, середній	середній, темний	4000, 3500	400,400	1200,1000
			в	малий, середній, великий	світлий, середній, темний	2500, 2000	300,200	750, 600
			г	середній, великий, великий	світлий, світлий, середній	1500, 1250	200, 200	400, 300

Продовження таблиці 1.1

Характеристика зорової роботи	Найменший або еквівалентний розмір об'єкта розрізнення, мм	Розряд зорової роботи	Підрозряд зорової роботи	Контраст об'єкта з фоном	Характеристика фону	Штучне освітлення		
						Освітленість, лк		
						при системі комбінованого освітлення		при системі загального освітлення
						всього	у т.ч. від загального	
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Дуже високої точності	Від 0,15 до 0,3 включно	I I	а	малий	темний	4000, 3500	400, 400	-,-
			б	малий, середній	середній, темний	3000, 2500	300, 300	750, 600
			в	малий, середній, великий	світлий, середній, темний	2000, 1500	200, 200	500, 400
			г	середній, великий, великий	світлий, світлий, середній	1000, 750	200, 200	300, 200
Високої точності	Від 0,3 до 0,5 включно	I I I	а	малий	темний	2000, 1500	200, 200	500, 400
			б	малий, середній	середній, темний	1000, 750	200, 200	300, 200
			в	малий, середній, великий	світлий, середній, темний	750, 600	200, 200	300, 200
			г	середній, великий, великий	світлий, світлий, середній	400	200	200

Якість освітлення визначається сукупністю світлотехнічних величин таких як люмени та люкси. Світловий потік (люмени) і освітленість (люкси) визначають кількість світла та його інтенсивність на одиницю площі. Для робочих місць нормативно регламентуються мінімальні рівні освітленості: для житлових кімнат це

зазвичай 100-300 лк (внутрішнє освітлення), для офісів – 300-500 лк чи вище, згідно з EN 12464-1:2021 та ДБН[3],[4]. Яскравість (люмен/кандела на м²) характеризує сприйману інтенсивність світла від поверхні чи світильника і впливає на контрастність сцен і ймовірність осліплення. Важливими також є коефіцієнти відбиття поверхонь, наприклад, світлі стіни й стелі відбивають більше світла, покращуючи загальну освітленість. Напрямок світла визначає, які ділянки будуть висвітлені (пряме, рефлекторне, дифузне освітлення). Також досить важливим критерієм є спектральний склад випромінювання та колірна температура світла: "тепле" світло (~2700 – 3500 К) сприяє розслабленню, тоді як “холодне” (~5000 – 6500 К) підвищує зосередженість. Вплив спектру на біоритми досліджено в наукових дослідженнях[2]: вечірнє світло з колірною температурою ~3400 К стимулює вироблення мелатоніну, а “холодне денне” світло (~5600 К) його пригнічує. Важливі й показники якості світла, а саме індекс передачі кольору (Ra), який показує, наскільки природними виглядають кольори при штучному світлі, а також однорідність освітленості і відсутність яскравих виблисків чи пульсацій (мерехтіння), що впливають на візуальний комфорт. Згідно з гігієнічними нормами, системи освітлення мають забезпечувати рівномірність освітленості й спектральний склад, наближений до природного, а також виключати неприйнятні осліплювальні ефекти.

Зоровий аналізатор людини забезпечує 65-90 % усієї отримуваної інформації[2], тому світлове середовище суттєво впливає на самопочуття. Око адаптується до різних рівнів освітленості – від яскравого сонячного світла до темряви, шляхом зміни діаметра зіниці і регулювання чутливості сітківки. Процес адаптації ока людини потребує часу, раптове відмикання світла може спочатку засліпити людину, а вплив слабкого освітлення довго, викликає втому зору. Крім того, у сітківці є спеціальні циркадні фоторецептори, які реагують на спектр світла і впливають на вироблення гормону мелатоніну. Від цілеспрямованого використання цих знань виникло поняття людино-центричного освітлення (Human-Centric Lighting) – автоматизація кольорової температури світильників за часом доби і сценаріями, щоб імітувати природний ритм

освітленості. У природному циклі денне світло високої колірної температури допомагає підтримувати бадьорість, тоді як ввечері набагато ефективнішим для підготовки до сну є тепле світло. Ці біологічні ритми освітлення особливо враховуються в сучасних системах розумного дому та офісного планування, де освітлення адаптується автоматично.

У розумному будинку автономні світильники і датчики створюють середовище, яке адаптується до життєвого ритму мешканців. Завдяки LED-технологіям та інтернет-з'єднанню, системи дозволяють дистанційно регулювати яскравість і колір світла, планувати графіки ввімкнення/вимкнення та динамічно змінювати температуру світла в залежності від часу доби і активності користувачів. Наприклад, вранці система може забезпечити прохолодне денне світло для бадьорості, а ввечері – тепле та приглушене для розслаблення. У сучасних офісах об'єднуються природне і штучне освітлення: автоматичні жалюзі і фотореле оптимізують сонячне світло, а LED-панелі з димерами регулюють довільні рівні для робочих станцій. Такі рішення не лише підвищують зручність і продуктивність праці, а й сприяють енергоефективності – датчики руху й освітленості вимикають світло в порожніх приміщеннях або доповнюють природне світло. У цілому штучне освітлення виступає ключовим елементом ефективності та комфорту, у Smart Home воно інтегрується з охоронними системами, а в офісах – з ергономічними вимогами, як наприклад, згідно з EN 12464-1:2021, в офісах мають бути враховані і візуальні, і не візуальні потреби людини.

Сучасні системи передбачають гнучке керування світильниками – від простих вимикачів і димерів до складних мережевих протоколів (DALI, ZigBee, Wi-Fi). Базові методи управління включають ручне та автоматичне (за таймером, за датчиком руху чи освітленості) керування. У Smart Home широко застосовують сценарії, наприклад, «робочий», «релакс» або «економний», які в один дотик змінюють набір світильників і їхні параметри. Алгоритмічне керування використовує зворотний зв'язок із

датчиками освітленості, які підтримують заданий рівень люксів, та з датчиками присутності, які вмикають світло лише за наявності людей. У схемах з Raspberry Pi чи іншими контролерами також використовують ШІ (Штучний інтелект), наприклад, система може адаптувати освітлення на основі прогнозу погоди або календаря. Головні принципи таких рішень, це енергозбереження, зручність користування та відповідність нормам. У результаті правильно спроєктоване керування освітленням створює безпечне та комфортне середовище, одночасно оптимізуючи витрати енергії та підтримуючи здоров'я користувачів.

Проєктування освітлення ґрунтується на діючих стандартах. В Україні діє ДБН В.2.5-28:2018 «Природне і штучне освітлення», що містить вимоги до мінімальних освітленостей і організації світлового середовища в будівлях. Для офісних і інших робочих місць застосовують стандарт ДСТУ EN 12464-1:2016 (IDT ISO 8995-1:2002), який регламентує гігієнічні норми кількості та якості світла в приміщеннях. Норма EN 12464 передбачає, наприклад, мінімальну освітленість 500 лк на робочих столах із дисплеями з високими вимогами до кольоропередачі. Ці документи встановлюють також вимоги до одиничних показників такі як коефіцієнт відбивання поверхонь, однорідності розподілу світла, обмежень осліплення та ін.

1.3 Сучасні технології автоматизації освітлення

Автоматизація освітлення є ключовим напрямом розвитку інтелектуальних будівельних систем, її основна мета полягає у створенні енергоефективного, адаптивного та зручного світлового середовища, яке самостійно реагує на зміни умов освітленості, присутність людей або сценарії використання приміщення. Сучасні технології автоматизації поєднують датчики, контролери, бездротові комунікації та програмне керування, забезпечуючи динамічне регулювання світла відповідно до

потреб користувачів. На сьогоднішній час використовується широкий спектр технологій і протоколів керування освітленням, серед яких виділяються DALI, KNX, ZigBee, Z-Wave, Wi-Fi та Bluetooth.

DALI (*Digital Addressable Lighting Interface*) – це стандартизований цифровий протокол (IEC 62386), розроблений спеціально для керування освітленням у комерційних і громадських будівлях. Його основна перевага полягає у можливості адресного керування кожним світильником або групою без необхідності прокладання окремих ліній живлення для кожного пристрою. DALI забезпечує двосторонній обмін даними, контролер може не лише змінювати яскравість або колір, а й отримувати зворотну інформацію про стан лампи. Це робить систему зручною для великих офісів, торговельних центрів та промислових об’єктів, де важлива надійність і масштабованість.

KNX – універсальний стандарт для автоматизації будівель (EN 50090, ISO/IEC 14543-3), який охоплює не лише освітлення, а й опалення, вентиляцію, безпеку тощо. KNX може працювати по дротових або бездротових лініях і дозволяє об’єднувати пристрої різних виробників у єдину мережу. Система характеризується високою стабільністю та сертифікацією, тому широко використовується у професійних рішеннях для Smart Home.

ZigBee і *Z-Wave* – це бездротові протоколи, розроблені спеціально для побудови мереж IoT-пристроїв з низьким енергоспоживанням. ZigBee працює у діапазоні 2,4 ГГц і підтримує топологію «mesh», де кожен пристрій може передавати сигнал далі, розширюючи радіус дії системи. Це робить технологію оптимальною для домашніх систем освітлення, зокрема таких, як Philips Hue, IKEA Tradfri, Xiaomi Mi Smart Lighting тощо. Z-Wave функціонує у нижчому частотному діапазоні (868 МГц у Європі), що забезпечує кращу проникну здатність сигналу крізь стіни і стабільність з’єднання, проте має меншу швидкість передачі даних.

Wi-Fi і *Bluetooth Low Energy* (*BLE*) – це протоколи загального призначення, які сьогодні активно застосовуються і в тому числі для керування освітленням. Їхня головна перевага, це простота інтеграції з мобільними пристроями без додаткового шлюзу. Wi-Fi-світильники можуть безпосередньо підключатися до домашньої мережі і керуватися через мобільний застосунок або голосових асистентів. Bluetooth LE, у свою чергу, забезпечує енергозбереження і підходить для локального керування (наприклад, у лампах Philips Hue Bluetooth або Xiaomi Mi Smart Bulb). У великих будівлях часто комбінують кілька протоколів, наприклад, DALI для дротового керування групами світильників та ZigBee або KNX для централізованої координації системи через шлюз або сервер керування.

Інтернет речей (IoT) є технологічною основою сучасних систем Smart Lighting. У цій парадигмі кожен світильник або контролер виступає як “розумний вузол”, здатний обмінюватися даними з іншими пристроями та хмарними сервісами. IoT-світильники мають убудовані мікроконтролери та модулі зв’язку (наприклад, ESP32, ESP8266, Nordic nRF52), які дозволяють реалізувати функції по типу:

– автоматичне регулювання яскравості залежно від рівня природного освітлення (з використанням фоторезисторів або датчиків освітленості LDR);

– виявлення присутності людей за допомогою PIR-датчиків або ультразвукових сенсорів;

– дистанційне керування і моніторинг через інтернет або мобільний застосунок;

– аналіз споживання електроенергії та оптимізація режимів роботи світильників.

Інтеграція освітлення у IoT-екосистему дозволяє будувати адаптивні сценарії. Також можливе поєднання з іншими системами «розумного дому», такі як кліматконтроль, відео-спостереження, мультимедіа. Важливою складовою автоматизованих систем освітлення є мікроконтролери (Arduino, ESP32, STM32) і мікрокомп’ютери (Raspberry Pi, Orange Pi, BeagleBone Black тощо). Вони виконують роль центральних

вузлів, які збирають інформацію з датчиків і керують виконавчими пристроями, такі як світлодіодний драйвер, реле або димер світла.

Беручи до уваги найвідоміші комерційні системи Smart Lighting, можна навести приклади успішних рішень:

– *Philips Hue*:

екосистема бездротового освітлення, що працює за протоколом ZigBee Light Link або Bluetooth. Дає змогу керувати кольором, яскравістю і створювати сцени освітлення через мобільний застосунок або голосових асистентів, як зображується на рисунку 1.1.



Рисунок 1.1 Створення заднього освітлення за допомогою системи Philips Hue

– *Xiaomi Mi Smart Lighting*:

модульна система розумних ламп і світильників з Wi-Fi та BLE-підтримкою, інтегрована в екосистему Mi Home, що дозволяє будувати сценарії взаємодії з іншими

IoT-пристроями (датчиками руху, камерами, вимикачами), як зображено на рисунку 1.2.



Рисунок 1.2 Інтеграція димерного світла Xiaomi Mi Smart Lighting в простір

– *IKEA TRADFRI, Yeelight, LIFX*:

приклади бездротових систем з елементами динамічного керування та інтеграцією у платформи Google Home, Apple HomeKit, Amazon Alexa, як зображено на рисунку 1.3.



Рисунок 1.3 Адаптивна зміна рівня освітлення системи IKEA TRADFRI

1.4 Адаптивні системи освітлення

Інтелектуальні системи, що здатні автоматично змінювати параметри світлового середовища (яскравість, колірну температуру, напрямок світлового потоку) залежно від зовнішніх умов, часу доби, присутності людей або конкретних сценаріїв використання приміщення, називаються адаптивними системами освітлення. Головна ідея таких систем полягає у створенні динамічного світлового середовища, яке підлаштовується під фізіологічні, емоційні та робочі потреби користувача, водночас мінімізуючи споживання енергії. На відміну від традиційних систем, де освітлення працює у фіксованому режимі, адаптивне освітлення використовує зворотний зв’язок між сенсорами та контролером. Контролер отримує дані про поточний стан середовища, аналізує їх за певними алгоритмами і коригує роботу світильників у реальному часі. Такий підхід відповідає принципам Smart Lighting і Human-Centric Lighting (HCL), де важливою є не лише енергоефективність, а й відповідність освітлення біологічним ритмам людини.

Для реалізації адаптивного керування використовуються різні типи датчиків, кожен з яких відповідає за окремий аспект освітлення:

– Датчики освітленості (фоторезистори, фотодіоди, LDR, BH1750, TSL2561), визначають рівень природного чи штучного світла у приміщенні. Контролер аналізує отримані дані і автоматично регулює яскравість світильників, підтримуючи заданий рівень освітленості. Це дозволяє уникати надмірного освітлення у денний час і забезпечувати достатній рівень світла в темний період доби.

– Датчики руху (PIR, мікрохвильові, ультразвукові), реагують на переміщення об’єктів у зоні дії. Вони вмикають світло при вході людини у приміщення та вимикають після певного періоду бездіяльності. Це один з найефективніших способів зменшення енергоспоживання в коридорах, санвузлах, офісах та зовнішньому освітленні.

– Датчики присутності, більш чутливі, ніж звичайні датчики руху, здатні фіксувати навіть незначні рухи (наприклад, роботу за комп’ютером) і підтримують освітлення доти, доки в приміщенні перебуває людина. У комерційних будівлях їх застосовують для оптимізації робочих зон та зменшення навантаження на систему.

– Датчики часу та астрономічні таймери, керують освітленням залежно від пори доби або сезону. Наприклад, у Smart Home система може автоматично приглушувати освітлення у вечірній час чи вимикати зовнішні ліхтарі після світанку.

– Комбіновані сенсори (мультисенсорні системи), поєднують кілька функцій, такі як рух, освітленість та температура, забезпечуючи більш точну реакцію системи на зміну умов. Такі сенсори застосовуються у високотехнологічних офісах і «розумних» будівлях з інтегрованими системами автоматизації KNX, DALI чи ZigBee.

Якщо ж говорити про загальні переваги автоматизованих, адаптивних систем освітлення, варто зазначити наступні критерії:

1. *Енергоефективність*:

Згідно з дослідженням[5], впровадження адаптивного керування може скоротити споживання електроенергії до 17% тільки на адаптивних світильниках. Якщо ж порівнювати з традиційними системами освітлення, то заміна їх на повноцінні автоматизовані системи може зекономити від 30% до 50% електроенергії. Це досягається завдяки вимиканню світла у порожніх приміщеннях, приглушенню світильників та використанню природного освітлення.

2. *Комфорт*:

Автоматичне регулювання яскравості й колірної температури відповідно до часу доби зменшує зорове навантаження, підтримує працездатність і покращує якість відпочинку. Людино-центричне освітлення, засноване на адаптивних алгоритмах, позитивно впливає на самопочуття та циркадні ритми користувача. Також до

функціоналу додається можливість віддаленого доступу, що в свою чергу збільшує комфорт користування автоматизованими системами освітлення.

3. *Гнучкість і масштабованість*:

Адаптивні системи легко інтегруються в інші елементи «розумного дому», системи опалення, вентиляції, охорони. Завдяки відкритим протоколам (DALI, KNX, ZigBee) користувач може змінювати сценарії та додавати нові пристрої без складних налаштувань.

Отже, адаптивні системи освітлення є важливим етапом розвитку сучасних технологій керування світловими середовищами, що поєднують принципи енергоефективності, комфорту та інтелектуальної взаємодії людини з простором. Їхня ключова особливість полягає у здатності автоматично реагувати на зміни умов навколишнього середовища таких як рівень природного освітлення, присутність людей, час доби чи характер діяльності користувача. Застосування сенсорних технологій, мікроконтролерів, алгоритмів машинного аналізу даних та мережевої інтеграції забезпечує високий рівень гнучкості, надійності та точності керування. У комерційних і громадських будівлях такі системи дозволяють зменшити експлуатаційні витрати, оптимізувати роботу інженерних мереж і підвищити продуктивність персоналу.

Впровадження адаптивних систем освітлення є одним із ключових напрямів реалізації концепції Smart Home, де інтегроване керування всіма інженерними підсистемами забезпечує раціональне використання енергоресурсів і сталий розвиток. Таким чином, адаптивне освітлення можна вважати невід’ємною складовою майбутніх енергоефективних і екологічно збалансованих середовищ, що поєднують технічну досконалість і орієнтацію на потреби людини[6].

РОЗДІЛ 2 МІКРОКОМП'ЮТЕРИ ТА МОДУЛІ У СФЕРІ ТЕХНОЛОГІЙ РОЗУМНОГО ДОМУ (SMART HOME)

2.1 Класифікація та введення у мікрокомп'ютери Raspberry Pi

Мікрокомп'ютери Raspberry Pi становлять серію одноплатних комп'ютерів (Single Board Computers, SBC), розроблених фондом Raspberry Pi Foundation з метою популяризації комп'ютерної освіти, програмування та доступних апаратних рішень для побудови вбудованих систем. Від моменту появи першої моделі у 2012 році лінійка значно розширилась, що зумовило необхідність їх систематизації. Лінійка Raspberry Pi розвивалася поступово, удосконалюючи обчислювальні можливості, енергоефективність та інтеграцію периферії[7].

Перше покоління Raspberry Pi (2012):

Перше покоління мікрокомп'ютерів Raspberry Pi стало проривом у сфері доступних одноплатних комп'ютерів. Модель була створена з метою популяризації освіти у галузі програмування та апаратної інженерії.

Основними моделями першого покоління є:

- Raspberry Pi Model A – спрощена версія з одним USB-портом та без Ethernet,
- Raspberry Pi Model B – повноцінна плата з Ethernet та двома USB-портами, як зображено на рис. 2.1-2.2,
- Model A+ та B+ – оновлені плати з покращеним енергоспоживанням і збільшеною кількістю GPIO.

Ключові характеристики моделей:

- Процесор: ARM11 (700 МГц), однокористувацька архітектура ARMv6;
- Оперативна пам'ять: 256-512 МБ;
- Графіка: VideoCore IV;
- Накопичувач: SD-карта як основний носій;
- Порти: HDMI, USB 2.0, GPIO 26-40 пінів.



Рисунок 2.1-2.2 Мікрокомп’ютер Raspberry Pi 1 Model B+, вид зверху та знизу

Перше покоління Raspberry Pi, представлене у 2012 році, стало фундаментальною основою для всієї подальшої еволюції одноплатних комп’ютерів цієї серії. Це покоління започаткувало ідею створення надзвичайно доступного, компактного та функціонального мікрокомп’ютера, який міг би використовуватися в освітніх цілях та сприяти популяризації програмування серед молоді. Створення Raspberry Pi було мотивоване прагненням зменшити бар’єр входу у світи комп’ютерних наук та електроніки, адже до цього подібні платформи були або дорогими, або орієнтованими на вузькі інженерні сфери. Перше покоління продемонструвало, що повноцінний комп’ютер може мати надзвичайно низьку ціну, працювати під керуванням Linux та забезпечувати базові можливості для проєктування, автоматизації та навчання. Апаратною основою першого покоління Raspberry Pi стала система-на-чипі Broadcom BCM2835, яка містила однокристальний

процесор ARM11 із тактовою частотою 700 МГц. Архітектура ARMv6, на якій базувався цей процесор, була відносно простою і за сучасними мірками малопотужною, проте саме її застосування дозволило досягти низької собівартості пристрою. ARM11 не мав високої обчислювальної продуктивності, проте був достатнім для виконання базових завдань: роботи з простими графічними інтерфейсами, запуску легких настільних програм, виконання скриптів, опрацювання сигналів з периферії та роботи з інтернетом. У порівнянні з пізнішими поколіннями він мав обмежений набір інструкцій, менший розмір кешу та однопотоковість, що знижувало його швидкодію. Однак саме ARM11 дозволив створити перший одноплатний комп'ютер, доступний широкому колу користувачів. Одним із ключових компонентів плати був графічний процесор VideoCore IV, який забезпечував апаратне прискорення мультимедіа. Хоча GPU не можна назвати високопродуктивним, він підтримував апаратне декодування відео 1080p, роботу з OpenGL ES 2.0 та базові можливості 3D-графіки. Ця особливість стала важливою для використання Raspberry Pi як медіацентру або навчальної платформи з візуалізацією, адже у той час більшість дешевих пристроїв не мали продуктивного апаратного прискорення відео. Саме GPU VideoCore дав змогу значному колу ентузіастів розвивати проєкти мультимедійних програвачів, таких як Kodi та XBMC, які стали популярними завдяки Raspberry Pi. Перші моделі Raspberry Pi комплектувалися невеликим обсягом оперативної пам'яті розміром у 256 МБ у Model A та 256 або 512 МБ у Model B. Пам'ять типу SDRAM була інтегрована у той самий чип BCM2835, що зменшувало вартість, але обмежувало продуктивність. Незважаючи на це, мала оперативна пам'ять не стала критичною для задач, для яких створювалося перше покоління, а саме, навчання основам операційних систем та програмування. Обмеження RAM стимулювали оптимізацію програмного забезпечення та створення полегшених версій Linux-дистрибутивів, у тому числі Raspbian, розробленого спеціально для Raspberry Pi. Склад плати першого покоління відзначався мінімалізмом і простотою. Моделі мали від одного до двох портів USB 2.0, порт Ethernet (у версії Model B), вихід HDMI для підключення моніторів,

композитний відеовихід, аудіороз'єм, а також 26 або 40 контактів GPIO залежно від ревізії. GPIO-інтерфейс став однією з ключових особливостей пристрою, що забезпечила інтеграцію з датчиками, модулями, контролерами та іншими апаратними компонентами. Саме можливість поєднання програмування і фізичної електроніки зробила Raspberry Pi надзвичайно популярним у сферах DIY та STEM-освіти. Як основний носій даних використовувалася SD-карта, що дозволяло легко замінювати або відновлювати операційну систему. Застосування Raspberry Pi першого покоління було досить різноманітним, хоча у більшості випадків пов'язувалося з навчальними та експериментальними задачами. Він широко використовувався для ознайомлення з Linux, мовами програмування (особливо Python та Scratch), основами мережевої взаємодії, принципами роботи електронних схем і систем автоматизації. Значна кількість користувачів використовувала Raspberry Pi для створення простих медіацентрів, ретроконсольних ігрових систем та домашніх серверів невеликої потужності. У робототехніці перше покоління стало основою для навчальних роботів, сенсорних систем, контролерів двигунів і модулів збору даних. Крім того, Raspberry Pi добре зарекомендував себе у навчальних лабораторіях університетів, де він використовувався як доступна апаратна платформа для вивчення операційних систем, архітектури комп'ютерів та взаємодії з периферією.

Друге покоління Raspberry Pi (2015):

Raspberry Pi 2 став суттєвим кроком вперед, адже перехід до чотириядерної архітектури дозволив значно підвищити обчислювальні можливості.

Основними моделями другого покоління є:

– Raspberry Pi 2 Model B – покоління має лише Model B, Model A для другого покоління офіційно не випускалися;
– Raspberry Pi Zero.

Raspberry Pi 2 Model B (див. рис. 2.3-2.4)

Ключові характеристики моделі:

- Процесор: Чотириядерний ARM Cortex-A7 з частотою 900 МГц (у ревізіях 1.1 та 1.2);
- Оперативна пам'ять: 1 ГБ SDRAM;
- Графіка: VideoCore IV;
- Порти: Full HDMI, 4×USB 2.0, гігабітний Ethernet (через USB-шину).

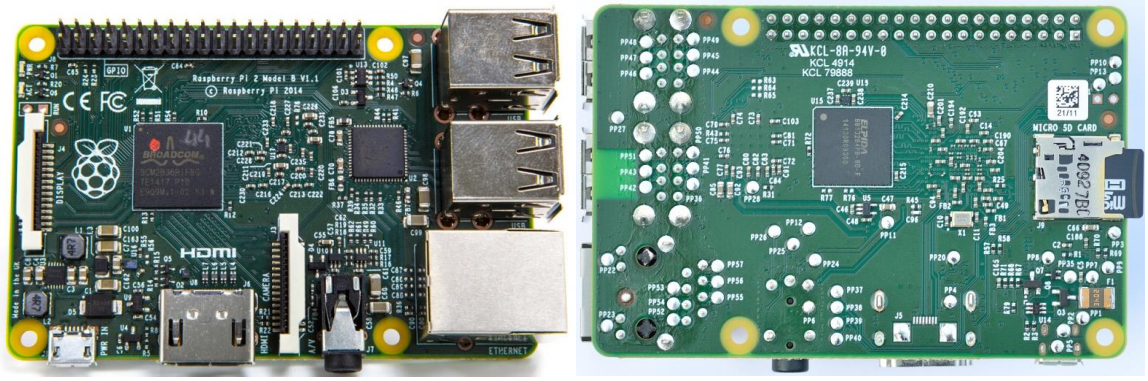


Рисунок 2.3-2.4 Мікрокомп’ютер Raspberry Pi 2 Model B, вид зверху та знизу

Друге покоління Raspberry Pi, представлене у лютому 2015 року, стало важливою еволюційною віхою в розвитку лінійки мікрокомп’ютерів Raspberry Pi. Воно не лише продемонструвало істотний приріст обчислювальної продуктивності порівняно з першим поколінням, а й сформувало основу для сприйняття Raspberry Pi як універсальної платформи не тільки для навчальних і хобі-проєктів, а й для більш комплексних технічних розробок. Ключовою відмінністю Raspberry Pi 2 Model B стало застосування значно потужнішої архітектури процесора ARM Cortex-A7 із чотирма обчислювальними ядрами, що суттєво підвищило можливості пристрою під час виконання багатопоточних додатків і дало змогу використовувати його у сферах, де перше покоління було недостатньо продуктивним. У технічному плані Raspberry Pi 2 отримав нову системну мікросхему Broadcom BCM2836, яка замінила попередній

BCM2835, що застосовувався у всіх моделях першого покоління. Перехід від одноядерного процесора ARM1176JZF-S з тактовою частотою 700 МГц до 4-ядерного ARM Cortex-A7 з тактовою частотою 900 МГц забезпечив відчутне зростання продуктивності. За оцінками розробників, приріст продуктивності становить збільшення приблизно у 4-6 разів залежно від типу задачі. Важливо, що при збільшенні кількості ядер було збережено енергоефективність пристрою, оскільки архітектура Cortex-A7 є оптимізованою для зниженого енергоспоживання та роботи з мінімальним тепловиділенням. Це дало змогу використовувати Raspberry Pi 2 у компактних та автономних системах без необхідності застосування активного охолодження. Графічна підсистема Raspberry Pi 2 залишилася незмінною у порівнянні з першим поколінням, оскільки продовжувала використовувати GPU VideoCore IV, однак збільшена загальна продуктивність системи позитивно позначилася на роботі мультимедійних застосунків і відтворенні відео високої чіткості. Мікрокомп’ютер, так само як і попередник, оснащувався 1 ГБ оперативної пам’яті стандартизованого типу LPDDR2, проте завдяки новій архітектурі ЦПУ та оптимізованим інструкціям перерозподілення пам’яті стало ефективнішим, особливо у застосунках із паралельними процесами. З точки зору апаратного інтерфейсу пристрій отримав ту саму кількість портів USB 2.0, роз’єм Ethernet, 40-контактний GPIO та відеовиходи HDMI і DSI/CSI, тому повністю зберігав сумісність з периферією, корпусами, HAT-модулями та аксесуарами від Raspberry Pi 1. Переваги Raspberry Pi 2 перед попереднім поколінням були досягнуті завдяки комплексному покращенню обчислювального ядра, оптимізації взаємодії апаратних модулів та збереженню високої сумісності з уже наявною екосистемою. Можливість роботи з більш вимогливими операційними системами, включно з Windows 10 IoT Core, стала важливим аспектом адаптації Raspberry Pi у професійне середовище розробників вбудованих систем. Зростання продуктивності в режимах обробки даних, виконання серверних завдань або моделювання дозволило використовувати Raspberry Pi 2 у якості міні-серверів, SSH-вузлів, вузлів систем відеоспостереження, налаштовуваних шлюзів IoT та платформ

для взаємодії з датчиками й виконавчими механізмами. У практичному використанні друге покоління Raspberry Pi зайняло важливе місце в освітніх, наукових та технічних середовищах. У навчальних закладах Raspberry Pi 2 застосовували для викладання основ інформатики, схемотехніки та робототехніки, оскільки приріст продуктивності дозволяв запускати більш складні інтерактивні середовища програмування, розгорнуті IDE, емулятори та навчальні платформи. У сфері IoT пристрій став основою для побудови інтелектуальних сенсорних мереж, домашньої автоматизації, систем енергомоніторингу, «розумних» контролерів і користувацьких пристроїв для збирання та обробки даних. Значний потенціал Raspberry Pi 2 проявився також у проєктах, пов’язаних із робототехнікою, де багатопоточна обробка дозволяла одночасно керувати рухом, обробляти сигнали від сенсорів і виконувати локальні алгоритми навігації.

Raspberry Pi Zero (див. рис. 2.5-2.6)

Ключові характеристики моделі:

- Процесор: 1 ГГц ARMv6 (оригінальна модель) або чотириядерний 64-bit 1 ГГц ARM Cortex-A53 (Zero 2 W);
- Оперативна пам'ять: 512 МБ LPDDR2;

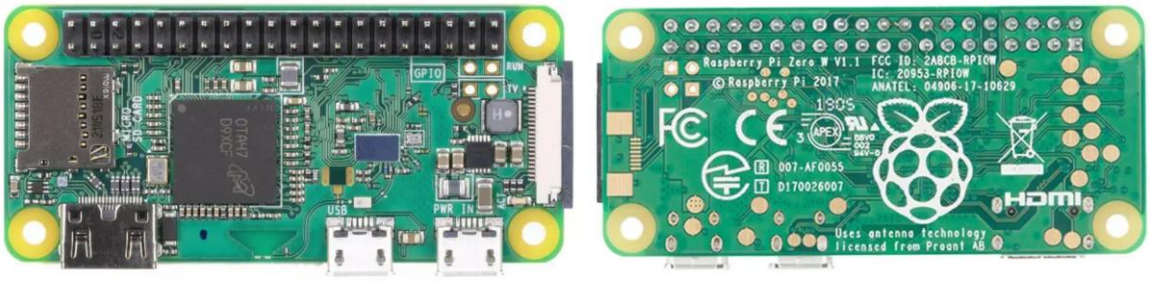


Рисунок 2.5-2.6 Мікрокомп’ютер Raspberry Pi Zero W, вид зверху та знизу

Raspberry Pi Zero є одним із найкомпактніших та найекономічніших представників сімейства мікрокомп’ютерів Raspberry Pi, офіційно представленим у

листопаді 2015 року. Його поява стала відповіддю на потребу ринку у надзвичайно мініатюрних, малопотужних, але повністю функціональних обчислювальних платах, які можна інтегрувати в обмежені простором пристрої або застосовувати у великосерійних проєктах з мінімальним бюджетом. Raspberry Pi Zero суттєво відрізняється від основної лінійки Raspberry Pi не лише розмірами, а й концепцією застосування: він орієнтований передусім на вбудовані системи, де пріоритетом є компактність і низьке енергоспоживання, а не максимальна продуктивність. Основою Raspberry Pi Zero стала мікросхема Broadcom BCM2835 – та сама, що використовувалася в першому поколінні Raspberry Pi Model B. Однак на відміну від початкових пристроїв, у Zero процесор ARM1176JZF-S отримав підвищену тактову частоту 1 ГГц, що забезпечувало приблизно 40% приросту продуктивності порівняно з оригінальним Raspberry Pi 1. Хоча процесор залишається одноядерним і базується на архітектурі ARMv6, що обмежує виконання важких розрахункових завдань, цього достатньо для легковагових ОС Linux, простих інтерфейсів, серверних служб і керування периферією. Обсяг оперативної пам’яті складає 512 МБ типу LPDDR2, що дозволяє забезпечувати стабільну роботу системних процесів та скромних користувацьких застосунків, хоча відкриває обмежені можливості для серйозної багатозадачності. Фізично Raspberry Pi Zero має надзвичайно компактну друковану плату розміром лише 65 × 30 мм, що приблизно вдвічі менше за стандартний Raspberry Pi Model A+. Такий форм-фактор досягається за рахунок мінімізації кількості портів і компонентів. Пристрій оснащений портом mini-HDMI для відеовиходу, micro-USB для живлення, другим micro-USB OTG для під’єднання периферії, роз’ємом microSD для завантаження операційної системи та нерозпаяним 40-контактним GPIO, який можна активувати вручну за потреби. Незважаючи на компактність, плата зберігає повну функціональну сумісність з більшістю бібліотек і програмного забезпечення Raspberry Pi, що дозволяє переносити проєкти з інших моделей без значних змін. У 2017 році була представлена вдосконалена версія Raspberry Pi Zero W, яка отримала інтегровані бездротові модулі Wi-Fi 802.11n та Bluetooth 4.1. Це значно розширило

область застосування Zero, перетворивши його на повноцінну платформу для IoT-рішень, віддалених сенсорних вузлів, автономних пристроїв та компактних точок збору і передавання даних. Пізніше з'явилася версія Zero 2 W, яка хоч і належить до окремого покоління, але стала прямим розвитком ідей, закладених у Zero. Ця версія базується на багатоядерному процесорі Cortex-A53, який збільшує продуктивність до п'яти раз, зберігаючи той самий форм-фактор. Компактність Raspberry Pi Zero визначає його ключові переваги над іншими моделями. Мінімальне енергоспоживання, яке зазвичай не перевищує 0,5–1,2 Вт, робить Zero ідеальним вибором для автономних систем на батарейному живленні або для невеликих пристроїв, де неприпустиме значне тепловиділення. Крім того, надзвичайно низька ціна Zero в близько 5 доларів США для базової моделі зробила її доступним для масового застосування в освітніх проєктах, експериментальних конструкціях і серійних виробах, де важливо мінімізувати вартість апаратної частини.

Третє покоління Raspberry Pi (2016):

Серія Raspberry Pi 3 стала ключовою у популяризації концепції «розумних» пристроїв, оскільки вперше інтегрувала бездротові модулі Wi-Fi та Bluetooth.

Основними моделями третього покоління є:

– Raspberry Pi 3 Model B;
– Raspberry Pi 3 Model B+ – підвищена частота CPU до 1.4 ГГц, швидший Ethernet і покращене Wi-Fi-покриття (див. рис. 2.7-2.8).

Ключові характеристики моделей:

- Процесор: ARM Cortex-A53 (64-bit), 1.2-1.4 ГГц, 4 ядра;
- Оперативна пам'ять: 1 ГБ SDRAM;
- Графіка: Broadcom BCM2837;
- Бездротова комунікація: Wi-Fi 2.4 GHz, Bluetooth 4.1;
- Порти: HDMI, 4×USB, Ethernet, GPIO 40 pin.

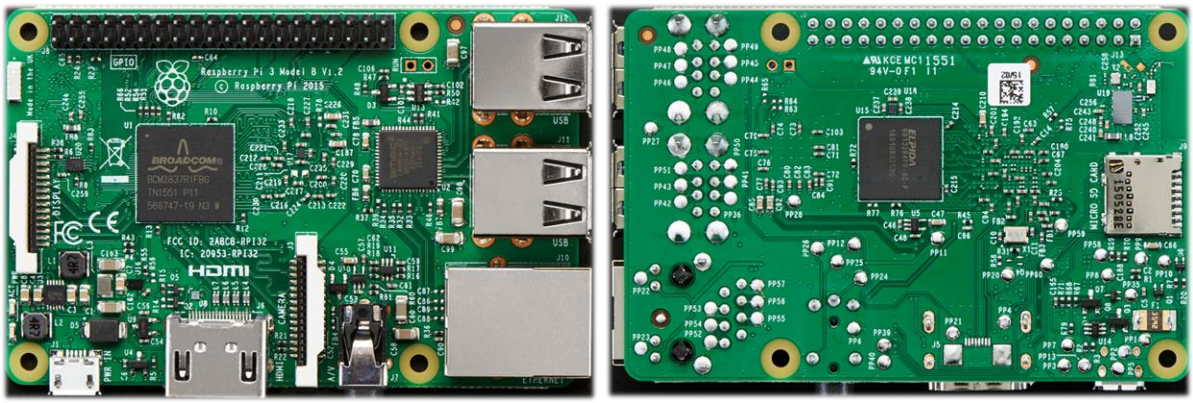


Рисунок 2.7-2.8 Мікрокомп’ютер Raspberry Pi 3 Model B, вид зверху та знизу

Третє покоління Raspberry Pi, представлене у 2016 році, стало черговим ключовим етапом розвитку лінійки одноплатних мікрокомп’ютерів і суттєво розширило можливості платформи як у функціональному, так і в технічному аспектах. Основною особливістю Raspberry Pi 3 стало інтегрування бездротових модулів Wi-Fi та Bluetooth безпосередньо на плату, що значно підвищило зручність використання пристрою та закріпило його позиції як універсального інструмента для створення рішень у сфері Інтернету речей, побудови мережевих пристроїв та розробки автономних систем. Це покоління орієнтоване не лише на освітні чи аматорські проєкти, але й на професійних розробників, яким потрібна стабільна, енергоефективна та водночас достатньо продуктивна обчислювальна платформа для реалізації прототипів і робочих систем. Центральною частиною Raspberry Pi 3 є нова система-на-чипі Broadcom BCM2837, яка прийшла на зміну BCM2836, що використовувався у другому поколінні. Вона містить 64-бітний чотирядерний процесор ARM Cortex-A53 із тактовою частотою 1,2 ГГц, що забезпечило приблизно 50–60% приросту продуктивності в порівнянні з Raspberry Pi 2. Хоча GPU VideoCore IV залишився тим самим, збільшення обчислювальної потужності ядра та оптимізації в архітектурі ARMv8 позитивно вплинули на швидкість обробки мультимедійних даних, загальну реактивність системи та ефективність роботи з системами віртуалізації, компіляції чи

обробки потоків даних. Однією з найбільш помітних технічних змін стала поява вбудованих бездротових комунікацій. Raspberry Pi 3 отримав модуль Wi-Fi 802.11 b/g/n та Bluetooth 4.1 (BLE), що усунуло потребу в зовнішніх USB-адаптерах і значно підвищило стабільність бездротового з'єднання. Це особливо важливо для систем, розгорнутих у великій кількості або в умовах обмеженого доступу, наприклад, у мережах IoT, сенсорних вузлах, мобільних роботах чи домашніх інтелектуальних системах. Вбудована бездротова частина дозволила створювати компактніші конструкції й забезпечила кращу інтеграцію плати в модульні системи автоматизації, моніторингу та керування. Плата Raspberry Pi 3 зберегла більшість апаратних інтерфейсів попередніх поколінь, включно з чотирма USB 2.0, HDMI, Ethernet, 40-контактним GPIO, роз'ємами CSI та DSI, що гарантувало повну сумісність із HAT-модулями, датчиками, камерами та аксесуарами, які вже сформували велику екосистему навколо Raspberry Pi. Оперативна пам'ять обсягом 1 ГБ стандарту LPDDR2 залишилася незмінною, проте завдяки вдосконаленій архітектурі процесора й підвищенню частотних характеристик стало можливим досягти кращого розподілу ресурсів і підвищеної продуктивності у багатозадачних сценаріях. Переваги третього покоління Raspberry Pi перед своїми попередниками були досягнуті за рахунок поєднання кількох факторів. По-перше, перехід на 64-бітну архітектуру та підвищення тактової частоти забезпечили суттєве прискорення виконання прикладних програм, серверних компонентів і обчислювальних алгоритмів. По-друге, інтеграція бездротових модулів створила передумови для більш ефективного використання плати у сфері мережевих рішень та IoT-застосунків. По-третє, попри значні покращення, було збережено сумісність з попередніми апаратними рішеннями, що дало змогу користувачам плавно перейти до нового покоління без потреби у повному оновленні периферії. Застосування Raspberry Pi 3 охоплює широке коло сфер. У проєктах Інтернету речей це покоління стало одним із найпоширеніших рішень для створення домашніх хабів, мережевих шлюзів, сенсорних вузлів та автономних контролерів. Універсальність і наявність бездротових інтерфейсів

роблять цю модель особливо зручною для лабораторних робіт, пов’язаних із передаванням даних, сенсорними мережами та побудовою експериментальних систем кіберфізичних об’єктів.

Четверте покоління Raspberry Pi (2019):

Поява Raspberry Pi 4 стала етапною подією, оскільки плата отримала значний приріст продуктивності і стала придатною для напівпрофесійних задач.

Основними моделями четвертого покоління є:

– Raspberry Pi 4 Model B;
– вбудована версія Compute Module 4 (CM4);
– інтегрований варіант у клавіатурі Raspberry Pi 400.

Raspberry Pi 4 Model B (див. рис. 2.9-2.10):

Ключові характеристики моделі:

- Процесор: ARM Cortex-A72 (64-bit), 1.5 ГГц, 4 ядра;
- Оперативна пам'ять: 1 (знято з виробництва)/2/4/8 ГБ LPDDR4 SDRAM;
- Графіка: VideoCore VI, присутня підтримка 4K;
- Порти: 2×Micro HDMI (до 4K), 2×USB 3.0 + 2×USB 2.0, Гігабітний Ethernet (не через USB), USB-C для живлення;
- Бездротова комунікація: Wi-Fi 2.4/5 GHz + Bluetooth 5.0.



Рисунок 2.9-2.10 Мікрокомп’ютер Raspberry Pi 4 Model B, вид зверху та знизу

Четверте покоління мікрокомп’ютерів Raspberry Pi, представлене моделлю Raspberry Pi 4, стало одним із найважливіших етапів розвитку всієї лінійки одноплатних комп’ютерів. Поява у 2019 році нового покоління ознаменувала перехід від використання Raspberry Pi виключно в освітніх або аматорських проєктах до широкого застосування в інженерних, напівпромислових та професійних сферах. Плата четвертого покоління зазнала суттєвих апаратних змін, отримала нову архітектуру процесора, перероблену підсистему пам’яті, швидкі порти введення-виведення та значно збільшену пропускну здатність периферії, що в сукупності зробило її якісно новим етапом розвитку сімейства Raspberry Pi. Основою Raspberry Pi 4 став чотириядерний процесор ARM Cortex-A72 із тактовою частотою 1.5 ГГц. Це сучасне 64-бітне ядро з архітектурою ARMv8-A, яке працює значно продуктивніше за Cortex-A53, встановлений у попередньому Raspberry Pi 3. Підвищення продуктивності досягається не лише більшою тактовою частотою, але й глибшою конвеєризацією, покращеним прогнозуванням переходів, збільшеними кешами L1 та L2, а також більшою кількістю виконуваних інструкцій за цикл. Завдяки цьому Raspberry Pi 4 став у декілька разів швидшим за свого попередника в задачах обробки даних, обчисленнях, роботі з веб-сервісами та мультимедіа. Окрім процесора суттєво оновлено й графічний прискорювач VideoCore VI, який отримав покращену підтримку виводу 4K-відео, декодування сучасних відеоформатів та підвищену пропускну здатність GPU-підсистеми, що дозволило вперше використовувати Raspberry Pi як повноцінний мультимедійний комп’ютер. Важливою зміною у четвертому поколінні стала переробка підсистеми пам’яті. Raspberry Pi 4 вперше отримав варіанти із 2, 4 та 8 ГБ оперативної пам’яті стандарта LPDDR4, яка забезпечує значно вищу пропускну здатність порівняно з LPDDR2, що застосовувалася у попередніх моделях. Саме завдяки переходу на LPDDR4 вдалося збільшити швидкість обміну даними між процесором та пам’яттю, що позитивно відобразилося на роботі операційних систем, запуску контейнерів, виконанні серверних задач та роботи з великими масивами інформації. Зростання обсягу оперативної пам’яті до 8 ГБ стало одним із ключових

факторів, що дозволили використовувати Raspberry Pi 4 для складніших застосувань, зокрема, віртуалізації легких контейнерів або запуску веб-сервісів одночасно з графічним інтерфейсом. Кардинальні зміни були внесені у систему портів та інтерфейсів. Raspberry Pi 4 отримав порти USB 3.0, які забезпечують суттєве збільшення швидкості роботи із зовнішніми накопичувачами та периферією. Це стало можливим завдяки застосуванню окремого контролера USB 3.0 та виділених шин передачі даних, що зменшило затримки й уникнуло «вузького місця», характерного для Raspberry Pi 3. Також модель вперше отримала повноцінний гігабітний Ethernet, який більше не обмежувався пропускною здатністю USB-шини. Це забезпечило стабільну швидкість у мережах з високим навантаженням і вдосконалило можливість використання плати як міні-сервера для локальних або навіть глобальних мережевих застосувань. Зміни торкнулися й відеовиходу, а саме, замість стандартного HDMI вперше застосовані два micro-HDMI порти з підтримкою двох незалежних дисплеїв та роздільністю до 4K. Таким чином, Raspberry Pi 4 перетворився на повноцінний комп’ютер, здатний працювати у мультимоніторних середовищах, що раніше було недоступно для одноплатних рішень даної серії. Живлення плати було переведене на сучасний стандарт USB Type-C, що забезпечило стабільніше енергозабезпечення та можливість використання потужніших блоків живлення, необхідних для роботи високопродуктивних компонентів. Переваги четвертого покоління над попередніми моделями виникли завдяки комплексному оновленню архітектури. Збільшення тактової частоти, перехід до Cortex-A72 та нового графічного ядра зумовили багаторазове зростання продуктивності. Впровадження LPDDR4 забезпечило суттєво вищу пропускну здатність пам'яті, а наявність гігабітного Ethernet і USB 3.0 дозволила підвищити показники вводу-виводу, які традиційно були слабким місцем Raspberry Pi. Покращені інтерфейси та підтримка 4K-дисплеїв розширили сферу застосування плати, зробивши її придатною як для мультимедійних систем, так і для офісних чи інженерних робочих станцій. Сфера застосування Raspberry Pi 4 значно ширша, ніж у попередніх поколінь. Завдяки збільшеній продуктивності його використовують як

домашні або корпоративні сервери: для реалізації VPN, DNS-фільтрів, медіасерверів, сервісів резервного копіювання, веб-сайтів або приватних хмарних сховищ. Завдяки підтримці 4K Raspberry Pi 4 широко застосовується у мультимедійних системах, зокрема як центр домашнього кінотеатру, рекламні дисплеї, інтерактивні кіоски або панелі інформації у громадських місцях. Значно покращені мережеві та обчислювальні можливості дозволяють використовувати його у сфері IoT та автоматизації для побудови розумних будинків, сенсорних мереж, контролерів та шлюзів між різними типами пристроїв. Наявність великих обсягів оперативної пам'яті та швидких портів робить Raspberry Pi 4 придатним для розробки та тестування програмного забезпечення, машинного навчання у легкому режимі, аналізу даних або управління робото-технічними комплексами. У галузі освіти Raspberry Pi 4 використовується як універсальна платформа для навчання програмуванню, адмініструванню, електроніці та робототехніці, оскільки дає можливість працювати як з сервісами низького рівня (GPIO), так і з повноцінними операційними системами.

Raspberry Pi Compute Module 4 (CM4) (див. рис. 2.11-2.12):

Ключові характеристики моделі:

- Процесор: Broadcom BCM2711 чотириядерний Cortex-A72 (ARM v8) 64-bit SoC 1.5GHz;
- Оперативна пам'ять: 1/2/4/8 GB LPDDR4-3200 SDRAM;

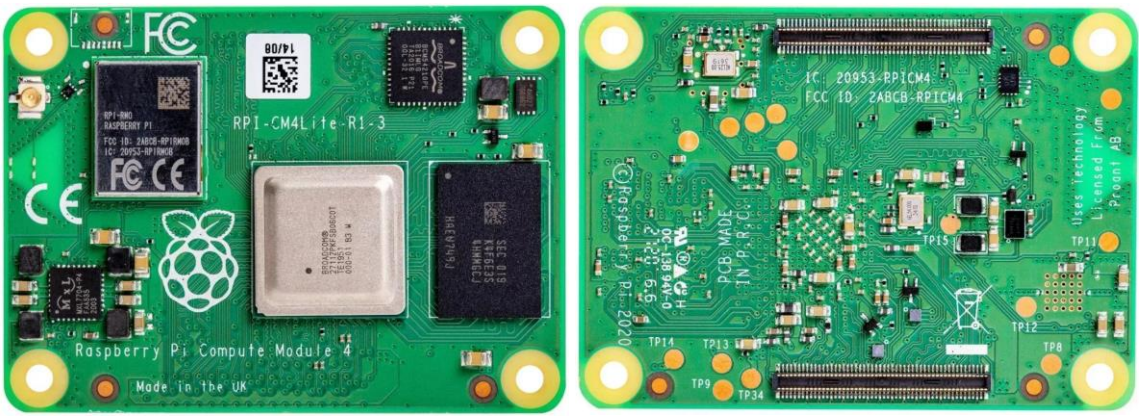


Рисунок 2.11-2.12 Raspberry Pi Compute Module 4 (CM4), вид зверху та знизу

Raspberry Pi Compute Module 4 (CM4), представлений у 2020 році, є спеціалізованим варіантом Raspberry Pi четвертого покоління, створеним для використання у вбудованих системах, промислових рішеннях та комерційних продуктах. На відміну від стандартних моделей Raspberry Pi, які позиціонуються як універсальні одноплатні комп'ютери, CM4 має форму компактного модуля, призначеного для інтеграції в кастомні плати-носії. Це робить його особливо привабливим для розробників, виробників електроніки та інженерів, яким потрібне потужне, гнучке та енергоефективне обчислювальне ядро для створення власних пристроїв. Основою Compute Module 4 є та ж система-на-чипі Broadcom BCM2711, яка використовується в Raspberry Pi 4 Model B. Вона включає чотириядерний 64-бітний процесор ARM Cortex-A72 з тактовою частотою 1,5 ГГц, що забезпечує високий рівень продуктивності та дозволяє модулю виконувати задачі, які раніше були доступні лише для повнорозмірних одноплатних комп'ютерів. Архітектура Cortex-A72 з покращеною паралелізацією, більшою пропускною здатністю кеш-пам'яті та оптимізованим виконанням інструкцій робить CM4 придатним для роботи з інтенсивними обчисленнями, обробкою мультимедійних потоків, мережевими завданнями та алгоритмами машинного навчання на локальному рівні. Compute Module 4 доступний у численних варіаціях, що дає змогу обирати конфігурацію відповідно до потреб конкретного застосування. Модуль може оснащуватися 1, 2, 4 або 8 ГБ оперативної пам'яті LPDDR4, а також постачається у версіях з eMMC-накопичувачем обсягом від 8 до 32 ГБ або без вбудованого флеш-накопичувача, якщо передбачається використання зовнішніх систем збереження даних. Наявність eMMC є критично важливою для промислових систем, де потрібна висока надійність та стійкість до зношування, яку не завжди можуть забезпечити карти microSD. Одним із ключових аспектів CM4 є його підтримка сучасних та високошвидкісних інтерфейсів. На відміну від стандартної плати Raspberry Pi 4, де інтерфейси виведені на роз'єми, у CM4 вони доступні через два багатоконтактні конектори високої щільності. Завдяки цьому під час проєктування плати-носія розробники можуть інтегрувати саме ті

порти, які потрібні у конкретному пристрої: USB 2.0 або USB 3.0 через зовнішні контролери, камери (дві шини MIPI CSI), дисплеї (дві шини MIPI DSI), PCIe 2.0 x1 для підключення швидких периферійних пристроїв або твердотільних накопичувачів. Вбудований інтерфейс PCIe є однією з найбільш цінних функцій модуля, оскільки відкриває шлях до створення високошвидкісних промислових пристроїв збору даних, серверів на NVMe-накопичувачах, маршрутизаторів та мережевих шлюзів. Compute Module 4 доступний у двох основних варіантах, із бездротовими інтерфейсами та без них. Моделі з підтримкою бездротового зв'язку оснащені модулем Wi-Fi 802.11ac та Bluetooth 5.0, що забезпечує високу швидкість та стабільність. Для застосувань із підвищеними вимогами до електромагнітної сумісності або в середовищах, де бездротові інтерфейси заборонені, передбачено версії без Wi-Fi та Bluetooth, що відповідає промисловим нормам і вимогам сертифікації. Підтримка подвійного виходу HDMI у CM4 (через інтерфейси на платі-носії) дозволяє створювати системи цифрових вивісок, мультимедійні інформаційні панелі, інтелектуальні кіоски та термінали. Завдяки потужнішому графічному ядру VideoCore VI модуль здатний відтворювати відео у форматі 4K і забезпечувати апаратне розкодування сучасних кодеків, що робить його корисним для систем відеоспостереження чи медіастендів.

Raspberry Pi 400 (див. рис. 2.13-2.14):

Ключові характеристики моделі:

- Процесор: Broadcom BCM2711, 4-ядерний Cortex-A72 64-bit SoC 1,8 ГГц;
- Оперативна пам'ять: 4 ГБ LPDDR4-3200.

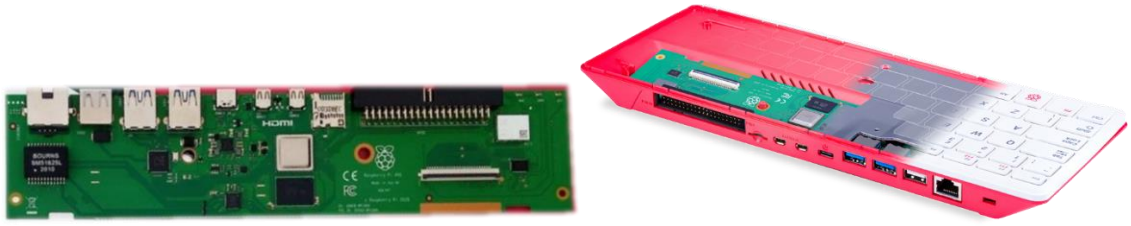


Рисунок 2.13-2.14 Raspberry Pi 400, вигляд плати в клавіатурі

Raspberry Pi 400 є унікальним представником сімейства Raspberry Pi, оскільки поєднує мікрокомп’ютер та повнорозмірну клавіатуру в одному пристрої. Представлений у 2020 році, він орієнтований на освітні, офісні та домашні застосування, де важливими є компактність, зручність користування та інтегрований форм-фактор. На відміну від традиційних моделей Raspberry Pi, які потребують окремого корпусу та підключення периферії, Raspberry Pi 400 є готовим повноцінним робочим місцем, яке необхідно лише під’єднати до монітора. Задня панель Raspberry Pi 400 містить усі необхідні інтерфейси: два порти micro-HDMI з підтримкою 4K, два USB 3.0 і один USB 2.0, роз’єм USB-C для живлення, гігабітний Ethernet та слот для microSD-карти. Також присутній 40-контактний GPIO, що зберігає сумісність із більшістю периферійних модулів і дає можливість застосовувати пристрій у навчальних та інженерних проєктах.

П’яте покоління Raspberry Pi (2023):

Raspberry Pi 5 – найпродуктивніше покоління, що робить одноплатний комп’ютер конкурентом деяких міні-ПК (див. рис. 2.15-2.16).

Ключові характеристики моделі:

- Процесор: ARM Cortex-A76, 2.4 ГГц, 4 ядра;
- Оперативна пам'ять: 4/8 ГБ LPDDR4X;
- Графіка: VideoCore VII (значно покращена підтримка графіки та мультимедіа);
- Порти: підтримка PCIe 2.0 (через FFC), 2×USB 3.0, гігабітний Ethernet, 2×4K HDMI;
- Інше: апаратний контролер енергоживлення (PMIC), покращене охолодження.

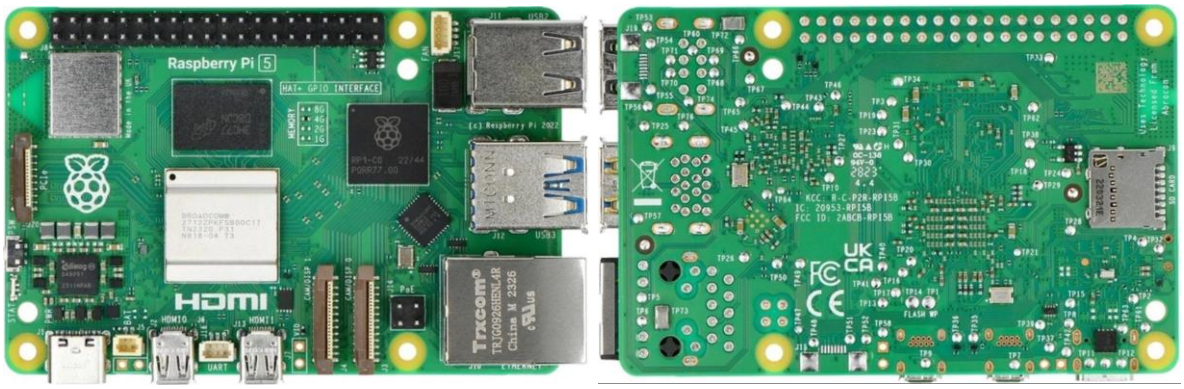


Рисунок 2.15-2.16 Raspberry Pi 5, вигляд моделі зверху та знизу

П’яте покоління Raspberry Pi, офіційно представлене у 2023 році, стало найбільш значущим технічним оновленням за всю історію платформи. На відміну від попередніх поколінь, де вдосконалення здебільшого виражалися в кращій продуктивності процесора та появі нових периферійних можливостей, Raspberry Pi 5 являє собою майже повністю переосмислену апаратну архітектуру. Воно вперше використовує власний мікроконтролер вводу-виводу RP1, який замінює традиційні периферійні рішення Broadcom і дозволяє суттєво збільшити швидкість роботи інтерфейсів, підвищивши загальну пропускну здатність системи. Raspberry Pi 5 був розроблений як високопродуктивний мікрокомп’ютер, здатний виконувати завдання рівня настільних систем, при цьому зберігаючи компактність, енергоефективність та низьку вартість, традиційно притаманні Raspberry Pi. Одним з ключових технічних нововведень п’ятого покоління стала нова система-на-чипі Broadcom BCM2712, побудована на базі чотириядерного 64-бітного процесора ARM Cortex-A76 з тактовою частотою 2,4 ГГц. Перехід на архітектуру Cortex-A76, яка використовується у сучасних смартфонах і компактних обчислювальних пристроях преміум-класу, забезпечив близько 2,5-3-кратний приріст продуктивності порівняно з Raspberry Pi 4. Завдяки збільшеній кількості інструкцій, покращеному конвеєру обробки команд і оптимізованому кешуванню Cortex-A76 дозволяє Raspberry Pi 5 виконувати складніші обчислення, працювати з важчими програмами та забезпечувати вищу стабільність

системи під навантаженням. Особливо значним є приріст продуктивності в задачах аналізу даних, компіляції, машинного навчання на локальному рівні, віртуалізації та мультимедійної обробки. Оперативна пам’ять у Raspberry Pi 5 представлена модулями LPDDR4X із частотою до 4267 MT/c (*Megatransfers per second*), що є помітним кроком уперед порівняно з LPDDR4 у попередньому поколінні. Це забезпечує значно швидший доступ до даних, підвищує пропускну здатність пам’яті та сприяє ефективнішій роботі багатозадачних застосунків. Варіації з різним обсягом пам’яті (4 або 8 ГБ) роблять Raspberry Pi 5 придатним як для базових навчальних цілей, так і для складних серверних або високонавантажених IoT-систем. Порівняно з попереднім поколінням, Raspberry Pi 5 отримав суттєві переваги, які базуються не лише на збільшенні обчислювальної потужності, але й на глибокому архітектурному переосмисленні. Перехід на новий клас процесорів, впровадження власного периферійного контролера, поява PCIe, швидшої пам’яті та удосконалена теплова система дозволили пристрою працювати стабільніше та ефективніше навіть під тривалим високим навантаженням. Графічна система зазнала певного оновлення, GPU VideoCore VII забезпечує кращу обробку графіки, покращену роботу мультимедійних застосунків, плавніший інтерфейс при роботі з настільними системами та повноцінну підтримку двох моніторів із роздільною здатністю 4K через два порти micro-HDMI. Це дає змогу використовувати Raspberry Pi 5 у ролі настільного ПК для офісних завдань, медіацентра, інформаційної панелі або станції керування у виробничих системах. Уперше модель Raspberry Pi була розрахована на роботу з активним охолодженням, що враховано при проєктуванні офіційного корпусу, який здатен значно знижувати температуру та підтримувати максимальну продуктивність без тротлінгу. Це покоління позначило перехід Raspberry Pi від пристрою для хобі або освіти до високопродуктивної платформи, що може конкурувати з традиційними міні-ПК та вбудованими обчислювальними модулями.

2.2 Класифікація та порівняння виробників мікрокомп'ютерів

Ринок мікрокомп'ютерів складається з кількох великих категорій гравців: фонди та спільноти, що проєктують «народні» одноплатні комп'ютери з орієнтацією на освіту й хобі; промислові та комерційні виробники модулів систем-на-чипі (SoM / Compute Module) для інтеграторів; компанії, що спеціалізуються на високопродуктивних платформах для AI/edge. Ключовими критеріями при порівнянні виробників є технічні характеристики (CPU, GPU, пам'ять), набір інтерфейсів і пропускна здатність вводу-виводу (USB, PCIe, Ethernet, MIPI), підтримка операційних систем і софту, рівень документації та спільноти, ціна та доступність, а також промислова надійність і варіанти сертифікації.

Основні виробники та їх позиції.

Raspberry Pi Foundation є еталоном у сегменті доступних одноплатних комп'ютерів із сильною спільнотою, широкою екосистемою HAT-модулів і регулярними апаратними оновленнями (лінійка включає Raspberry Pi 4, Pi 5 та Compute Module серії). Платформи Raspberry Pi орієнтовані на освіту, prototyping, IoT і легкі серверні/медіа застосунки, вони поєднують помірну ціну, обширну документацію та великий вибір ОС-зборок (Raspberry Pi OS, Debian/Ubuntu-based, спеціалізовані образи). Raspberry Pi підтримує як споживчі платформи (Model B, Zero, 400), так і промислові модулі (Compute Module) для інтеграції.

Nvidia позиціонує свою лінійку Jetson (див. рис. 2.16) як сімейство високопродуктивних платформ для edge-AI та комп'ютерного зору. Технічно Jetson-модулі (Nano, Xavier, Orin та їхні варіанти) поєднують багатоядерні ARM-CPU з потужними GPU/тензорними прискорювачами (Ampere/Maxwell/Turing-кластери з тензорними одиницями), апаратним блоком трасування/рендерингу в деяких серіях та широким набором інтерфейсів для камер і сенсорів. Важливий компонент екосистеми

NVIDIA, це програмний стек JetPack (включно з CUDA, cuDNN, TensorRT, VPI), що дає розробникам оптимізовані інструменти для розгортання нейромереж на пристрої. Платформи Jetson призначені для задач, в яких потрібні десятки або сотні TOPS: автономні роботи, промислова відеоаналітика, многокамерні системи спостереження, робототехніка і edge-сервери для інференсу. Головними перевагами лінійки є апаратне прискорення нейромереж і зріла софт-платформа, але вартість і енергоспоживання вищі за «народні» SBC.



Рисунок 2.17 Мікрокомп’ютер NVIDIA Jetson

ASUS увійшла в ринок SBC через серію Tinker Board (див. рис. 2.17-2.18), орієнтовану на користувачів, яким потрібні кращі мультимедійні можливості та комерційна апаратна надійність у порівнянні з типовими «хобі» платами. Технічно Tinker Board побудований на Rockchip/подібних SoC (залежно від моделі), часто має потужніший аудіо-кодек, апаратне відео-прискорення, gigabit Ethernet і варіанти з eMMC. ASUS позиціонує Tinker Board для застосувань у цифрових вивісках, мультимедіа-плеєрах, інтегрованих ПЗ і перших прототипах промислового обладнання, де важливі сумісність із промисловими інтерфейсами, стабільність і

підтримка від великого бренду. Документація і служба підтримки є перевагою для комерційних проєктів, хоча екосистема менша за Raspberry Pi.

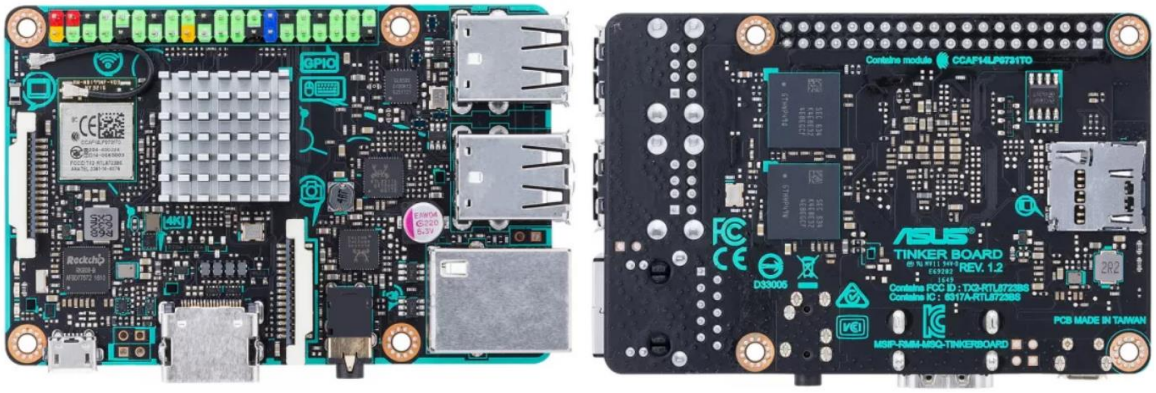


Рисунок 2.18-2.19 Мікрокомп’ютер ASUS Tinker Board, вид зверху та знизу

BeagleBoard (BeagleBone), проєкт із відкритим апаратним дизайном, який давно відомий у спільноті розробників і інженерів за свій акцент на низькорівневому контролі апаратури та наданні детальної технічної документації. Плати BeagleBone (див. рис. 2.19-2.20) часто мають багатий набір портів вводу-виводу, додаткові PRU (Programmable Real-time Units) у деяких SoC для задач реального часу, а також стабільну підтримку Linux і реального часу. Завдяки цьому BeagleBone добре підходить для систем автоматизації, промислових контролерів, апаратного навчання в університетах і проєктів, де критична точність таймінгів і розширений I/O. Спільнота та документація одні з найкращих у класі «open-hardware».

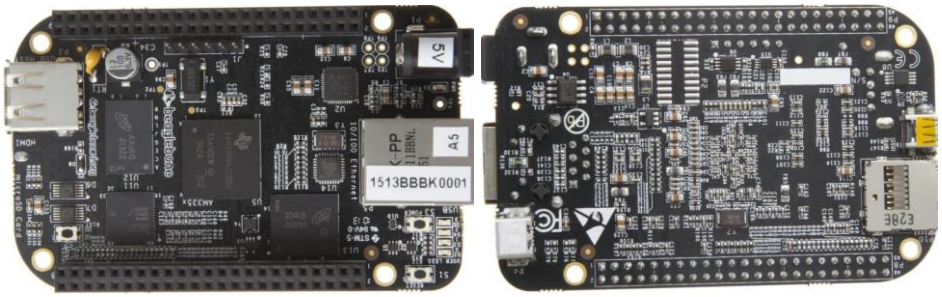


Рисунок 2.20-2.21 Мікрокомп’ютер BeagleBone Black, вид зверху та знизу

PINE64, компанія з орієнтацією на «відкриті» апаратні продукти та спільноту ентузіастів. Вони пропонують широкий спектр плат, від недорогих SBC до більш продуктивних рішень типу ROCKPro64 (на Rockchip RK3399) і навіть ноутбуків, телефонів власної розробки. Технічні особливості PINE64-плат: використання потужних мобільних SoC (RK3399, RK3588 тощо), наявність PCIe-слотів, підтримка eMMC, M.2 NVMe у деяких моделях. PINE64 приваблює тих, хто хоче «залізо з більше інтерфейсів» за нижчу ціну, і тих, хто цінує відкриту розробку та спільнотну підтримку кастомних firmware або дистрибутивів.

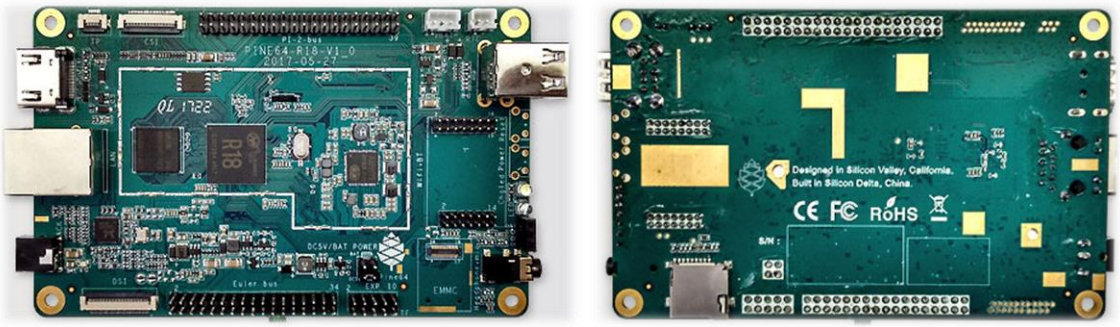


Рисунок 2.22-2.23 Мікрокомп’ютер PINE A64-LTS, вид зверху та знизу

Hardkernel (бренд ODROID) відомий серією продуктивних SBC (див. рис. 2.23), що пропонують високу апаратну якість і привабливе співвідношення «продуктивність-ціна». Технічно ODROID-плати використовують SoC від Amlogic, Rockchip та ін. і часто надають варіанти з eMMC, SATA/PCIe (у десктопоподібних моделях), а також чудову апаратну сумісність із Linux/Android. Позиціонування Hardkernel, це платформи для тих, хто хоче продуктивні медіа-центри, домашні сервери, ретро-ігри, а також компактні обчислювальні вузли. Сильна сторона серії, її технічна надійність, активні форуми та регулярні оновлення.

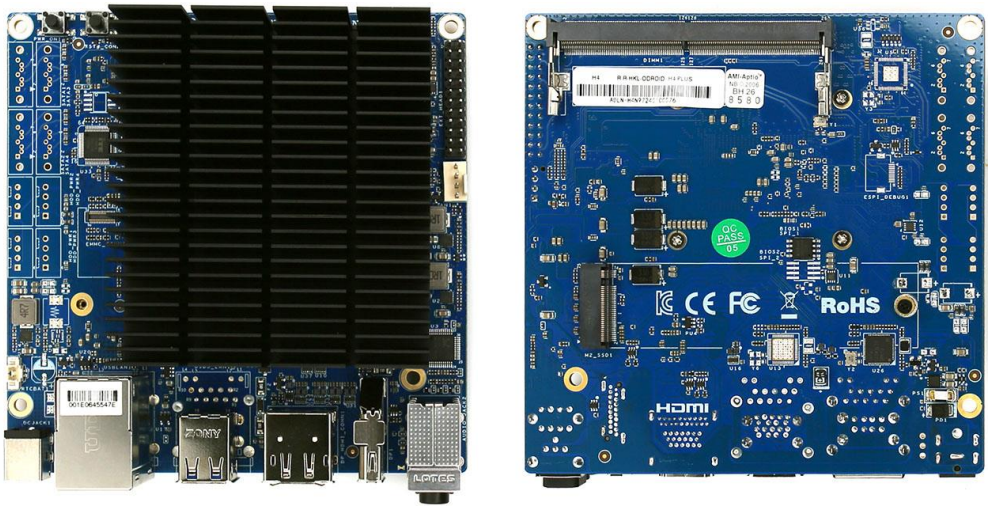


Рисунок 2.24 Мікрокомп’ютер ODROID-H4, вид зверху та знизу

Orange Pi, одна з найдоступніших за ціною серій SBC (див. рис. 2.23-2.24). Компанія Shenzhen Xunlong пропонує велику кількість різних моделей з різними SoC (Allwinner, Rockchip, Amlogic), іноді з дуже агресивним співвідношенням «ціна-функції»: моделі з підтримкою SATA, M.2 або Wi-Fi/Bluetooth за невелику ціну. Платформи Orange Pi забезпечують базову сумісність з Linux/Android, мають власну спільноту та офіційні/неофіційні образи ОС. Основний ризик при використанні, це варіабельність якості драйверів та менш систематизована офіційна документація порівняно з Raspberry Pi або Toradex, проте для бюджетних проєктів та прототипів Orange Pi є привабливим вибором.

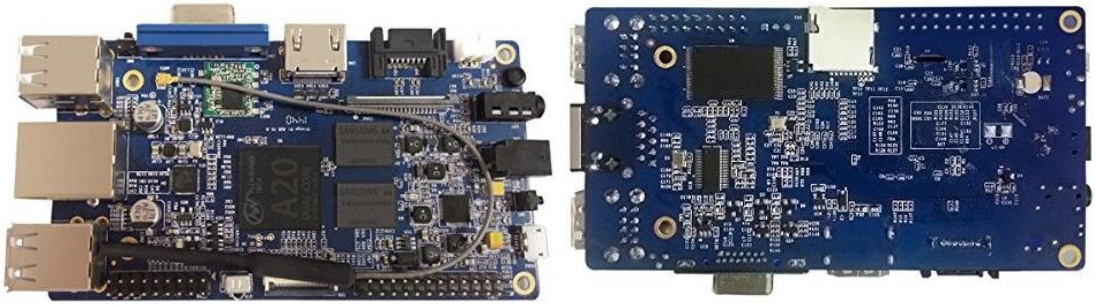


Рисунок 2.25-2.26 Мікрокомп’ютер Orange Pi, вид зверху та знизу

На ринку присутні ще багато менш відомих компаній виробників мікрокомп'ютерів, такі як «FriendlyELEC (NanoPi)», «Khadas», «SolidRun», «LattePanda», «Toradex», «AAEON», «Advantech».

FriendlyELEC (раніше FriendlyARM) виробляє серію NanoPi/NanoPC, які охоплюють широкий спектр рішень: від ультракомпактних плат до продуктивних моделей з підтримкою M.2 NVMe та апаратними NPU (у новіших моделях). Компанія робить ставку на варіативність конфігурацій, хорошу ціну та підтримку великої кількості дистрибутивів (Debian, Ubuntu, Armbian). NanoPi часто обирають для IoT-шлюзів, компактних серверів, інфраструктури домашньої автоматизації і проєктів з камерою або стрімінгом. FriendlyELEC активно розвиває власні образи та драйвери, та пропонує пристойну кількість готових прикладів і прошивок.

Khadas сфокусований на преміальних SBC типу VIM-серії (VIM1/VIM2/VIM3/VIM4), які зазвичай використовують SoC від Amlogic і надають доволі високу продуктивність, апаратну підтримку NPU для прискорення AI-задач, M.2/PCIe варіанти і добру мультимедійну здатність (апаратне декодування AV1/4K і т.д.). Khadas орієнтується на розробників, що потребують компактної платформи з гарною підтримкою AI-прискорення для застосувань в intelligent media players, медіасерверах, smart-edge пристроях та прототипах з комп'ютерним зором. Компанія також активно підтримує open-source програмне забезпечення й надає докладну технічну документацію.

SolidRun пропонує сімейство модулів та плат для комерційних та промислових застосувань: SOM/COM (System on Module), компактні SBC та edge-сервери. Традиційно SolidRun орієнтований на мережеві застосунки (маршрутизатори, шлюзи), індустріальні IoT-шлюзи та edge-сервери, пропонуючи стійкі до експлуатації апаратні рішення з довгим життєвим циклом, опціями кастомізації для OEM/ODM. Їхні продукти часто застосовуються там, де потрібен баланс між апаратною гнучкістю, стабільністю і професійною підтримкою.

LattePanda, виробник x86-орієнтованих SBC/модулів (Intel-based), які можуть запускати Windows і Linux «із коробки». Це робить їх привабливими для застосувань, де необхідна сумісність із Windows-пакетами, специфічним x86-ПЗ або для розробки IoT-рішень на знайомому середовищі. Технічні конфігурації охоплюють діапазон від енергоефективних N100/N150 платформ до високопродуктивних Sigma/Delta моделів з потужними Intel-CPU. LattePanda поєднує SBC та можливості Arduino-сумісного контролера у деяких моделях, що корисно для гібридних апаратно-програмних проєктів.

Toradex, виробник Compute Modules або ж Computer-on-Modules (Colibri, Apalis, Verdin), який орієнтований на промислові, медичні та автомобільні застосунки. Toradex надає модулі із широкою підтримкою промислових SoC (NXP i.MX, NVIDIA, Qualcomm у деяких партнерствах), довгим життєвим циклом продуктів, сертифікаціями та підтримкою промислових температур. Компанія робить акцент на якісній підтримці програмного забезпечення, наявності BSP (board support packages), довгостроковій підтримці і можливості кастомізації модулів для OEM-виробництва. Це робить Toradex гарним вибором для комерційних продуктів, де критична сертифікація та гарантії постачання.

AAEON, це великий тайванський виробник промислових комп'ютерів та SBC, що пропонує як готові одноплатні рішення (на Intel, AMD, ARM), так і модулі і готові системи (панельні ПК, fanless комп'ютери, індустріальні контролери). Платформи AAEON орієнтовані на промислову автоматизацію, Smart Factory, AIoT, інфраструктуру транспортних і медичних рішень. Технічна відмінність полягає в широкому модельному ряді, корпоративна підтримка, можливість отримати промислові сертифікації та опції кастомізації під вимоги великих системних інтеграторів.

Advantech, глобальний лідер у галузі промислових вбудованих рішень, від SBC і COM до повних платформ для автоматизації. Advantech пропонує продукти у

стандартизованих форм-факторах (Pico-ITX, 3.5", PC/104 тощо), має багаті опції сертифікації, програмну платформу WISE-PaaS для IoT-інтеграцій і сильну сервісну підтримку для промислових замовників. Advantech часто використовується в критичних системах (транспорт, енергетика, медицина), де потрібна довгострокова підтримка, сертифікація та інтеграційні послуги.

Узагальнення та структуризація типових параметрів та характеристик серій і моделей SBC наведених вище виробників для полегшеного сприйняття та аналізу наведено в таблиці 2.1

Таблиця 2.1

Узагальнена структура характеристик та параметрів мікрокомп'ютерів виробників на ринку

Виробник	**Типові SoC**	**Архітектура CPU**	**Типовий обсяг RAM**	**Ключові інтерфейси**	**Основне призначення**
Raspberry Pi	Broadcom BCM2711/BCM2712	ARM Cortex-A53/A72/A76	512 МБ - 8 ГБ	USB 2/3, HDMI, GPIO 40-pin, CSI/DSI, Ethernet	Освіта, прототипування, IoT, домашні сервери
NVIDIA (Jetson)	Nano/Xavier/Orin (Tegra)	ARM + CUDA GPU	2 - 64 ГБ	PCIe, CSI, Ethernet, GPIO, M.2	AI/ML edge, робототехніка, комп'ютерний зір
ASUS Tinker Board	Rockchip RK3288/RK3399	ARM Cortex-A17/A72/A53	2 - 4 ГБ	HDMI, USB, eMMC, Ethernet	Мультимедіа, signage, панелі
BeagleBoard / BeagleBone	TI Sitara AM335x/AM57x	ARM Cortex-A8/A15	512 МБ - 2 ГБ	GPIO, PRU, USB, Ethernet	Промислові контролери, RT-завдання
PINE64	RK3399/RK3588	ARM Cortex-A53/A72/A76	2 - 16 ГБ	PCIe, eMMC, USB-C, HDMI	DIY, NAS, медіа
Hardkernel (ODROID)	Amlogic/Rockchip	ARM Cortex-A55/A73/A76	1 - 8 ГБ	eMMC, HDMI, USB, Ethernet	Медіацентри, сервери, емулювання
Orange Pi	Allwinner/Rockchip/Amlogic	ARM Cortex-A7/A53/A55	1 - 8 ГБ	HDMI, USB, SATA, Wi-Fi	Бюджетні SBC, DIY

Продовження таблиці 2.1

Виробник	**Типові SoC**	**Архітектура CPU**	**Типовий обсяг RAM**	**Ключові інтерфейси**	**Основне призначення**
Banana Pi	Allwinner/MediaTek	ARM Cortex-A7/A53	1 - 4 ГБ	SATA, GPIO, Ethernet, CSI/DSI	NAS, роутери
FriendlyELEC (NanoPi)	Amlogic/Rockchip	ARM Cortex-A53/A55/A76	1 - 8 ГБ	M.2, eMMC, HDMI, Ethernet	IoT, edge-сервери
Khadas	Amlogic/Rockchip	ARM Cortex-A53/A73/A76	2 - 16 ГБ	M.2, HDMI, USB-C, GPIO	AI edge, мультимедіа
SolidRun	NXP i.MX / Marvell	ARM Cortex-A7/A53/A72	1 - 8 ГБ	PCIe, M.2, Ethernet, GPIO	Промислові SBC
LattePanda	Intel Atom / Intel N100/N200	x86	4 - 16 ГБ	HDMI, M.2, USB, GPIO	Windows-системи, інтеграція
Toradex	NXP i.MX / NVIDIA	ARM Cortex-A7/A53/A72	512 МБ - 8 ГБ	SOM/COM, PCIe, LVDS	Промислові вбудовані системи
AAEON	Intel/AMD/ARM	x86/ARM	2 - 32 ГБ	PCIe, SATA, COM, GPIO	Промислове IoT, медичні системи
Advantech	Intel/ARM	x86/ARM	2 - 32 ГБ	PCIe, RS-232/485, GbE	Автоматизація, SmartCity

2.3 Технології та модулі автоматизації освітлення в Smart Home

Автоматизація освітлення в системах «розумного дому» сьогодні представляє собою поєднання апаратних модулів, мережевих шлюзів, сенсорів і програмної логіки, що разом дозволяють перетворити просте вмикання ламп на багатофункціональний сервіс керування комфортом, енерговитратами та безпекою. У її основі лежать прилади різного призначення такі як інтелектуальні лампи та світильники з вбудованими цифровими драйверами, модулі-реле і димери, які встановлюються в розподільних коробках або на DIN-рейку для централізованого управління, датчики присутності та освітленості, а також керувальні панелі й програмні платформи для сценарного управління та інтеграції з голосовими помічниками й системами енергоменеджменту. Технологічно сучасні модулі автоматизації освітлення розрізняються за трьома великими ознаками: спосіб з’єднання з мережею (бездротовий або провідний), метод керування навантаженням (пряме керування лампою через реле чи димер, або управління драйвером через цифрові інтерфейси), і рівень інтеграції в загальну екосистему будинку (локальне автономне керування або хмарна чи гібридна інтелектуальна логіка).

На ринку присутні декілька груп постачальників, кожна з яких займає свою певну нішу. Виробники масових споживчих рішень, такі як Signify з продуктами під брендом Philips Hue, роблять ставку на зручність установки, естетичність пристроїв і широку екосистему аксесуарів. Їхні лампи, стрічки та модулі інтегровані за допомогою власних мостів і протоколів на основі Zigbee, при цьому Philips Hue активно розвиває можливості для синхронізації світла з мультимедіа та голосовими платформами. Компанії, що спеціалізуються на професійному або напів-професійному керуванні освітленням, наприклад, Lutron пропонують як споживчі серії для зручності встановлення в житлових інтер’єрах, так і масштабовані системи для багатозонального управління й інтеграції зі шторними приводами та HVAC. Їхні

контролери і бездротові комунікаційні рішення (наприклад Caseta або RadioRA) роблять наголос на надійності, високій якості димування та простоті інсталяції. Одночасно на ринку з’явилися й відносно дешеві та гнучкі модулі для ретрофіту, відомі завдяки брендам типу Shelly та Sonoff, які пропонують компактні Wi-Fi або Zigbee реле й вимикачі, що легко встановлюються за вимикачем або в розподільній коробці й дають змогу швидко впровадити в дім автоматизацію без капітального переобладнання електропроводки. Ці рішення вирізняються низькою ціною, простотою інтеграції в популярні платформи автоматизації й можливістю локальної роботи або управління через хмару. Паралельно існують виробники, які пропонують компактні, але промислово орієнтовані модулі та розв’язки для OEM-проєктів, наприклад, Fibaro відома своїми компактними вбудованими реле та модулями для Z-Wave мереж, що дозволяють перетворити звичайний двопровідний чи трипровідний вимикач у вузол розумного дому з довгою ланкою сумісності у Z-Wave екосистемі. Протоколи й технології передачі даних визначають архітектуру системи автоматизації та її можливості, саме тому у побутовому сегменті вже тривалий час домінують бездротові мережі Zigbee та Z-Wave, що створюють самовідновні mesh-мережі, оптимізовані для низької енергоспоживаної комунікації й великої кількості кінцевих модулів, такі як лампи, вимикачі, датчики. Zigbee останніми роками зберігає тенденцію розвитку, що підвищує його адаптивність для великих розгортань як у житловому, так і в комерційному середовищі. Паралельно на ринку закріпився «Thread», як сучасна IPv6-орієнтована mesh-технологія з низьким енергоспоживанням, що разом із протоколом Matter (новим промисловим стандартом для взаємодії пристроїв) стає основою для безшовної сумісності між виробниками. «Matter», особливо в останніх оновленнях, розширює підтримку типів пристроїв і функцій, що робить його привабливим для уніфікації керування освітленням у мульти-екосистемному середовищі. Для більш професійних і промислових інсталяцій існують провідні стандарти DALI (Digital Addressable Lighting Interface) і KNX, які проєктувалися спеціально для керування освітленням у будівлях, та пропонують

багаторівневий контроль, адресацію світильників, конфігуровані сценарії енергоменеджменту і високу надійність. «DALI», у своїй сучасній версії DALI-2, є де-факто стандартом для цифрового керування світильниками в комерційних проєктах, тоді як KNX широко застосовується для інтеграції освітлення з іншим інженерним обладнанням у «розумних» будівлях[8].

Технічні підходи до реалізації димування, сценаріїв і енергоменеджменту також суттєво відрізняються залежно від рівня системи. У споживчих магазинах і домашніх інсталяціях димування часто реалізується апаратно через фазове регулювання (leading або trailing edge) або цифрові драйвери для LED із сумісністю 0-10V чи «DALI», тоді як у високопрофільних комерційних рішеннях застосовують адресовані DALI-шини, які дозволяють не лише змінювати яскравість і колірну температуру окремих світильників, але й збирати телеметрію по енергоспоживанню та стану. Сучасні цифрові димери і LED-драйвери підтримують плавні криві затемнення, калібрування по температурі і режимам збереження енергії, а при поєднанні з датчиками присутності та освітленості можуть створювати сенсорні сцени, що автоматично підлаштовують світло під природні умови або активність людей у приміщенні. Інтеграція та архітектура системи включають кілька підходів. У схожих установках «все-в-одному» (smart bulbs + bridge) рішення зручні для швидкого розгортання, не вимагають змін у проводці і дозволяють миттєво отримати програмовані сценарії та сумісність із голосовими помічниками. У випадку ремонту чи комерційного монтажу важливішим можна позначити рішення на основі вбудованих модулів-реле й димерів, встановлених за вимикачем або у щиті, що дозволяє зберегти існуючі стиль і механіку вимикачів, але додати інтелект. У такому сценарії часто застосовують DIN-реле або модулі розміру в одну-дві DIN-шини, які надають статистику енергоспоживання, таймери і локальну логіку незалежно від хмарних сервісів. Для великих проєктів інтеграційні шлюзи між KNX чи DALI і IP-сегментом або між Z-Wave чи Zigbee і хмарними платформами забезпечують уніфіковану панель керування та аналітику. Шлюзи такого типу виробляють як глобальні системні виробники, так і спеціалізовані

вендори мережевих контролерів, бувають випадки у професійних інсталяціях, коли вони часто вбудовуються в розподільні шафи поряд з серверами керування будівлею.

Технічний опис основних протоколів і технологій, які використовуються виробниками модулів автоматизації освітлення в Smart Home[9]:

– *Zigbee*, це пакетний бездротовий протокол верхнього рівня, який працює поверх фізичного і канального рівнів IEEE 802.15.4. Архітектурно Zigbee реалізує mesh-топологію: кінцеві вузли (лампи, вимикачі, датчики) можуть виступати як маршрутизатори, а мережа автоматично відбудовується при зміні топології. Останні релізи Zigbee додають підтримку підсмуг Sub-GHz (800/900 МГц, «Suzi») та поліпшення в області безпеки й сертифікації, що покращує дальність і стійкість у складних RF-умовах. За рахунок mesh-архітектури Zigbee добре масштабується по числу вузлів у межах будинку та знижує енергоспоживання для батарейних сенсорів. Переваги Zigbee: низьке енергоспоживання, самоорганізуюча мережа, широка підтримка виробників, особливо у сфері розумних ламп і HUE-екосистем, і велика база сертифікації. Недоліки: частотний діапазон 2,4 ГГц у більш ранніх версіях створює вразливість до завад від Wi-Fi і Bluetooth, складність сумісності між пристроями різних вендорів може зберігатися через варіації профілів і реалізацій, для великих інсталяцій іноді потрібно планувати додаткові маршрутизатори/повторювачі. Zigbee часто використовується виробниками ламп і модулів, і зараз його еволюція поєднується зі спробами краще інтегруватися у загальний екосистемний стандарт Matter.

– *Z-Wave*, це пропрієтарна (але широко ліцензована) бездротова технологія, що працює у суб-гігагерцових діапазонах (залежно від регіону: ~868 МГц у Європі, ~908–916 МГц у США). На відміну від Zigbee, Z-Wave орієнтований виключно на Smart Home застосування, з принципово простішими стеком і жорстко визначеними профілями пристроїв. Z-Wave також підтримує mesh-топологію, його сильні сторони, це хороша стійкість до перешкод , передбачувана продуктивність мережі та відносно

прості правила сумісності завдяки жорсткому процесу сертифікації. Головні мінуси Z-Wave: менший ринок виробників порівняно з Zigbee, ліцензійні обмеження й обмеження максимальної кількості вузлів у мережі в деяких версіях стандарту, хоча сучасні ревізії знімають частину цих обмежень, вони все ж присутні. Z-Wave часто зустрічається у професійних комплектуваннях і рішеннях, де важлива надійність бездротового керування освітленням.

– *Thread*, це IP-орієнтований низькоенергетичний mesh-протокол, побудований на IEEE 802.15.4 (2.4 GHz) з використанням 6LoWPAN (IPv6) для адресації пристроїв. Головна відмінність Thread від Zigbee в тому, що Thread є нативно IP-сумісним, тобто, пристрої стають вузлами IPv6-мережі, що робить інтеграцію з локальними інфраструктурами і мережевими сервісами значно простішою без переходу через «протокольні» шлюзи. Thread розроблений з акцентом на безпеку, надійність маршрутизації та низьке енергоспоживання. Thread добре підходить для великомасштабних та складних mesh-мереж розумного дому, де важлива IP-сумісність і майбутня масштабованість. Він також є одним із рекомендованих низькоенергетичних радіопротоколів у екосистемі Matter, що робить Thread привабливим вибором для нових продуктів, орієнтованих на міжплатформову сумісність. Переваги: IPv6-сумісність, висока надійність і безпека. Недоліки: для користувача інсталяції може знадобитися border-router для з’єднання з Wi-Fi/Ethernet, рівень підтримки на ринку знаходиться в процесі зростання, в порівнянні з Zigbee та Z-Wave.

– *Matter*, це сучасний стандарт для інтероперабельності Smart Home, ініційований Connectivity Standards Alliance, що підтримується Apple, Google, Amazon та ін. Matter не є самостійним «радіопротоколом», він визначає прикладні та мережеві шари з використанням IP, а для радіо-транспорту використовує Thread для низькоенергетичних пристроїв або Wi-Fi чи Ethernet для більш «важких» пристроїв. Matter спрямований на безпечну, уніфіковану взаємодію пристроїв різних виробників, спрощення процесу налаштування і мінімізацію залежності від хмарних сервісів.

Переваги Matter: перспективна широкомасштабна сумісність, сучасні функції безпечного підключення, налаштування і місцевого керування. Недоліки: реалізація стандарту та повна підтримка нових типів пристроїв, наприклад, камер чи зарядних станцій від деяких виробників іноді відстає, не всі існуючі пристрої миттєво переходять на Matter, через що, іноді потрібні оновлення прошивки або нові шлюзи. Matter швидко набирає обертів як «уніфікатор» для брендового розмаїття в розумному домі.

– *Wi-Fi*, загальновживаний IP-протокол у 2.4/5/6 GHz для широкосмугових пристроїв. Застосування Wi-Fi у модулях освітлення має ряд основних переваг такі як пряма IP-ідентифікація пристрою, простота інтеграції з існуючою домашньою мережею та можливість обслуговувати пристрої без окремих шлюзів. Багато «доступних» рішень такі як Shelly, Sonoff, використовують Wi-Fi і пропонують HTTP або MQTT API для місцевого керування і повної інтеграції у платформи домашньої автоматизації. Переваги Wi-Fi: висока пропускна спроможність, відсутність потреби в окремому мосту та прямий доступ з інтернету. Недоліки: високе енергоспоживання, що погано для батарейних датчиків, вища ймовірність радіозавантаження (перешкоди від інших Wi-Fi-пристроїв). Виробники, що працюють у сегменті масового ринку і ретрофіту, охоче використовують Wi-Fi через простоту інтеграції та низьку бар’єрність для користувачів.

– *Bluetooth Low Energy*, добре підходить для пристроїв з низьким енергоспоживанням, такі як сенсори, пульти, а Bluetooth Mesh робить Bluetooth придатним для mesh-мережі освітлення. BLE виграє в простоті підключення до смартфонів і не вимагає додаткових мостів, що зручно для локальних сценаріїв налаштування. Bluetooth Mesh додає топологію, але має обмеження в масштабі й інколи більшу складність управління станом мережі порівняно з Thread чи Zigbee. BLE часто використовується для сценарних дистанцій і для специфічних продуктів, проте в масовому сегменті освітлення його позиція слабша за Zigbee, Thread чи Matter.

– *DALI*, це провідний стандарт, який був спеціально розроблений для цифрового керування освітленням у будівлях. У своїй сучасній формі DALI-2 підтримує адресацію до 64 приладів на підмережу, роботу з різними типами драйверів і датчиків, а також стандартизовані профілі пристроїв для забезпечення сумісності. DALI оптимізований під професійні інсталяції, через що він дає дуже точне димування, можливість опитування стану драйверів, моніторинг споживання і комплексні сценарії керування. Переваги: промислова надійність, діагностика, сумісність з професійними монтажними практиками. Недоліки: така провідна інфраструктура вимагає спеціального кабелю або електромонтажу і контролера та меншу зручність «ретрофіту» в побутовому середовищі. DALI широко застосовується у комерційних і офісних проєктах, де важливі енергоменеджмент і сертифікація.

– *KNX*, це модульований, відкритий стандарт домашньої і будівельної автоматизації, який орієнтований на інтеграцію широкого набору інженерних систем, по типу освітлення, HVAC, штори чи безпека. KNX відомий своєю надійністю, можливістю реалізувати складні взаємодії між системами та довгим життєвим циклом пристроїв. Переваги KNX: універсальність, промислова сертифікація, підтримка великої кількості норм і інтеграцій. Недоліки: висока вартість впровадження, потреба у кваліфікованій інсталяції і складність для швидкого «DIY» застосування. KNX зазвичай використовується в професійних інсталяціях великих будівель, готелів та інших об’єктів з високими вимогами інтеграції.

– *MQTT*, це легкий протокол підписки, часто використовуваний як «скелет» для обміну повідомленнями між локальними пристроями і системою автоматизації, такі як Home Assistant, Node-RED або ж шлюзи. Він не замінює радіо-протокол, але широко уживаний у парі з Wi-Fi, Ethernet або іншими IP-транспортами для передачі статусів, команд і телеметрії. Переваги MQTT: простота, низька накладна частка, ефективність у роботі з великою кількістю повідомлень. Недоліки: базова специфікація не встановлює жорстких механізмів безпеки по типу TLS, ACL, вони додаються патчами, MQTT сам по собі не вирішує питання мережевої топології чи

живлення пристроїв. Для ентузіастів і інтеграторів MQTT, це зручний спосіб об’єднати Wi-Fi-пристрої такі як Shelly чи Sonoff і прозоро інтегрувати їх у системи автоматизації.

– *Пропрієтарний протокол*. Деякі виробники, наприклад, такі як Lutron використовують власні, оптимізовані RF-технології, які розроблені для високої надійності, відсутності інтерференцій та дуже точної роботи димерів. Перевага таких рішень: повна оптимізація під конкретні функції, по типу «реагування», «стабільність», «сумісність обладнання одного виробника», а недоліки: замкнутість екосистеми й вищі витрати при інтеграції з іншими брендами, хоча у Lutron існують офіційні API та шлюзи для інтеграції з більш широкими системами. Пропрієтарні протоколи часто застосовують у преміум-сегменті і професійних інсталяціях, де клієнт готовий платити за гарантовану роботу і підтримку.

Аналізуючи наведені протоколи та наведених розробників на ринку, можна підвести лінію для споживачів:

– Якщо потрібна максимальна сумісність між брендами і простота підключення до основних екосистем по типу Apple, Google чи Amazon, то Matter у поєднанні з Thread для низькоенергетичних пристроїв, або Wi-Fi для «важчих» пристроїв – найбільш перспективний вибір. Matter дає уніфікований прикладний рівень, сучасні можливості онбордінгу та місцевий контроль. Проте потрібно враховувати, що перехід на Matter може вимагати оновлення прошивок або заміни шлюзів для старого обладнання.

– Якщо важлива енергоефективність батарейних сенсорів і надійність mesh-мережі, то вибір між Thread і Zigbee буде залежати від бажаної IP-сумісності. Thread забезпечує нативну IPv6-інтеграцію, що корисно для майбутньої масштабованості і прямої мережевої взаємодії. Тоді як Zigbee має широку базу вже доступних пристроїв і багато вендорів. Сьогодні Zigbee може бути практичним вибором для швидкого розгортання, але Thread виглядає краще у далекій перспективі.

– Якщо пріоритет на стабільність радіо-зв’язку в середовищі з великою кількістю Wi-Fi-пристроїв, то варто розглядати підсмуги Sub-GHz (Z-Wave або нові Sub-GHz реалізації Zigbee 4.0). Працюючи поза 2.4 GHz, ці протоколи краще уникатимуть взаємних завад. Z-Wave добре зарекомендував себе в цьому плані, при чому нові версії Zigbee також додають Sub-GHz опції.

– Для професійних комерційних інсталяцій у офіси чи готелі, де потрібна точна адресація, телеметрія енергоспоживання, діагностика і довготривала підтримка, DALI-2 або KNX залишаються стандартом. Вони забезпечують точне димування, масштабовану адресовану логіку і хороші можливості інтеграції з BMS (Battery Management System). Для таких проєктів вартість монтажу і потреба у кваліфікованому налаштуванні прийнятні, а надійність і сертифікація ключові.

– Якщо на меті швидкий та дешевий ретрофіт у побуті, тобто наприклад, просте додавання «розумних» реле під вимикач, то Wi-Fi-базовані рішення від Shelly чи Sonoff із підтримкою локальних API та MQTT дають найкраще співвідношення ціна-гнучкість. Однак при великій кількості таких пристроїв потрібно стежити за навантаженням Wi-Fi-мережі й безпекою.

– Якщо потрібна надвисока надійність і плавність димування в преміум-рішенні, Lutron з їх Clear Connect і професійними димерними драйверами часто виправдовують інвестиції, тому що це професійний клас з перевіреними RF-алгоритмами для установки та налаштування складних домашніх проєктів. Єдині мінуси, це закритість екосистеми й вартість інтеграції.

– Щодо інтеграції в систему керування, то в цьому випадку «MQTT» лишається найзручнішим інструментом для інтеграторів і ентузіастів, він дозволяє узгоджено керувати пристроями різних типів і створювати сценарії в Home Assistant чи Node-RED. Але MQTT, це не мережевий транспорт, він працює поверх IP-мереж (часто Wi-Fi), тому його використання слід поєднувати з підходящими RF-технологіями.

2.4 Визначення та аналіз використовуваної технології

Використання мікрокомп’ютерів як основи для власних модулів контролю або автоматизації системи адаптивного освітлення відкриває значно ширші можливості, ніж застосування готових вузькоспеціалізованих «Smart Home» пристроїв. Основна причина криється у гнучкості, масштабованості, можливості модифікації та адаптації системи під конкретні вимоги, що особливо важливо в інженерних, освітніх, дослідницьких чи нестандартних проєктах. Типові готові смарт-рішення по типу Philips Hue, Aqara, Sonoff або інші, створені насамперед для масового користувача, а тому спроєктовані з акцентом на простоту встановлення та типові сценарії. Їх можна інтегрувати у загальну екосистему, але вони практично не дозволяють змінювати внутрішню логіку роботи, алгоритми обробки даних або структуру управління, що робить їх малопридатними для глибокої кастомізації. Водночас мікрокомп’ютери відкривають можливість створення власних алгоритмів адаптивного освітлення, побудови локальних обчислювальних моделей, застосування сенсорів різного типу, кастомізації протоколів зв’язку та створення повністю автономної інтелектуальної системи. Мікрокомп’ютери володіють відкритим програмним середовищем, здебільшого працюють на Linux-подібних платформах та дозволяють інсталювати будь-яке програмне забезпечення, від легких скриптів на Python до повноцінних серверних застосунків, аналітичних систем або ML-моделей (Machine Learning)[10]. У той час як готові пристрої переважно підтримують лише обмежений набір протоколів, мікрокомп’ютер може працювати з MQTT, Zigbee, Z-Wave, Modbus, Wi-Fi, Bluetooth Low Energy та будь-якими іншими технологіями, які можна інтегрувати через зовнішні модулі. Тобто відкриваються можливості створення універсального багатопротокольного шлюзу освітлення, який не лише керує лампами або димерами, а й за потреби аналізує навколишнє середовище, обробляє статистику, оптимізує сценарії та забезпечує адаптивність системи відповідно до рівня природного освітлення, часу доби, присутності людей або енергоспоживання. При виборі

виробника мікрокомп’ютерів оптимальним рішенням є платформа Raspberry Pi, яка стала стандартом у сфері DIY-проєктів, освітніх програм, лабораторних робіт та прототипування. У порівнянні з альтернативами (Orange Pi, Banana Pi, Odroid, ASUS Tinker Board, Jetson Nano та ін.), Raspberry Pi вирізняється стабільністю роботи, надійністю у довготривалих проєктах та виключно широкою підтримкою з боку спільноти при доступній ціні моделей. Це означає, що переважна більшість технічних питань вже були колись розв’язані, а отже, доступ до знань, документації, бібліотек і прикладів значно полегшує процес розробки. Raspberry Pi підтримується великою кількістю офіційних і сторонніх модулів, має стабільні дистрибутиви операційних систем і найкращу екосистему та інфраструктуру для роботи з периферією (GPIO, I2C, SPI, UART). На відміну від багатьох інших виробників, Raspberry Pi Foundation гарантує тривалу доступність моделей, що важливо в наукових або комерційних проєктах, де обладнання має бути уніфікованим і взаємозамінним протягом років.

Серед усіх моделей Raspberry Pi саме Raspberry Pi 4 вважається найбільш збалансованою платформою для систем адаптивного освітлення. Вона забезпечує суттєво вищу продуктивність у порівнянні з Raspberry Pi 3, зокрема завдяки використанню чотириядерного процесора ARM Cortex-A72, який дає помітний приріст у швидкості обробки даних, роботі мережевих служб, аналітичних модулів та систем реального часу. На відміну від поколінь Pi 1 і Pi 2, четверта версія підтримує 1-8 ГБ оперативної пам'яті, що дозволяє запускати складніші сервіси, наприклад, бази даних, системи автоматизації на кшталт Home Assistant або OpenHAB, сервер обробки сенсорів чи веб-інтерфейси керування освітленням. Raspberry Pi 4 отримала повноцінний гігабітний Ethernet, який більше не прив’язаний до USB-шини, а також підтримку USB 3.0, що важливо для підключення високошвидкісних модулів[11]. Окрім цього, саме ця модель забезпечує стабільну роботу подвійного дисплея через порти micro-HDMI, що може бути корисно для локальної візуалізації або налаштування системи. У порівнянні з Raspberry Pi 3 нове покоління отримало Wi-Fi 5 і Bluetooth 5.0, що підвищує стабільність зв’язку та дозволяє надійніше інтегрувати

систему з бездротовими сенсорами або контролерами. Однією з ключових переваг Raspberry Pi 4 перед більш старими моделями є можливість запускати декілька паралельних служб автоматизації без відчутної втрати продуктивності. Це має особливо велике значення для адаптивного освітлення, де одночасно можуть працювати алгоритми аналізу яскравості, обробка сигналів від датчиків руху, мережеві служби MQTT і веб-панелі. Попри відносну доступність наступних поколінь Raspberry Pi, таких як Raspberry Pi 5, їхнє використання для систем освітлення у межах таких проєктів не завжди є доцільним. Raspberry Pi 5, хоча й забезпечує значно вищу продуктивність і нову архітектуру, потребує складнішого охолодження, є дорожчою, а її продуктивний потенціал у завданнях освітлювальній автоматизації не буде повністю задіяним. Також Raspberry Pi 4 вже перевірена часом, має максимально зріле програмне забезпечення та драйвери, а більшість бібліотек і плагінів розроблялися саме з орієнтацією на її апаратну архітектуру. Таким чином, Raspberry Pi 4 пропонує найкращий баланс між ціною, продуктивністю, стабільністю та енергоефективністю, що робить її оптимальним вибором для створення власної системи адаптивного освітлення.

Тобто, використання мікрокомп’ютерів, а саме Raspberry Pi 4, забезпечує можливість повного контролю над алгоритмами, протоколами, сенсорами та структурою автоматизації. Розробник системи отримує не тільки готовий інструмент для управління освітленням, а й платформу, здатну еволюціонувати, масштабуватися й адаптуватися разом із зростанням вимог до системи, чого жоден закритий, вузькоспеціалізований комерційний пристрій дати не може.

РОЗДІЛ 3 РОЗРОБКА ТА АНАЛІЗ СИСТЕМИ АВТОМАТИЗАЦІЇ КЕРУВАННЯ ОСВІТЛЕННЯМ

3.1 Вибір та технічні характеристики компонентної бази

Ефективність та надійність будь-якої автоматизованої системи безпосередньо залежить від правильності підбору апаратних компонентів. Для реалізації системи адаптивного освітлення на базі одноплатного мікрокомп'ютера Raspberry Pi було обрано модульний підхід, який дозволяє гнучко масштабувати систему та забезпечує високу ремонтопридатність. Ринок компонентів для платформи Raspberry Pi пропонує широкий спектр рішень: від готових плат розширення (HAT – Hardware Attached on Top) до дискретних електронних компонентів для поверхневого монтажу. В рамках даного дипломного проєкту було прийнято рішення використовувати комбінований підхід із застосуванням промислових стандартів з'єднання та спеціалізованих модулів узгодження, що забезпечує стабільність роботи системи керування освітленням.

Опис та технічні характеристики обраних компонентів:

– *Керуючий модуль*: Raspberry Pi 4 Model B.

Ядром системи виступає одноплатний комп'ютер Raspberry Pi 4 Model B (див. рис. 3.1). Вибір саме цієї моделі зумовлений її високою обчислювальною потужністю, необхідною для роботи алгоритмів адаптивного освітлення у реальному часі, та наявністю розвинених інтерфейсів вводу-виводу. Також на модуль можна додати пасивне охолодження у вигляді радіаторів (див. рис. 3.2).



Рисунок 3.1 Керуючий модуль Raspberry Pi 4 Model B, без радіаторів

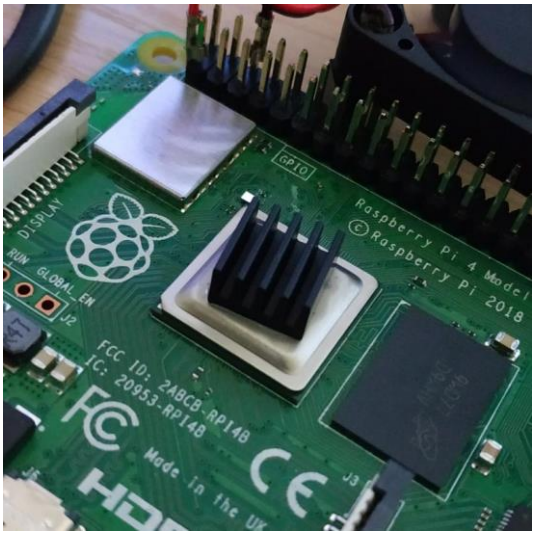


Рисунок 3.2 Пасивне охолодження (радіатор) на чіпах модуля Raspberry Pi

Основні характеристики модуля:

– *процесор*: Broadcom BCM2711, чотириядерний Cortex-A72 1.5GHz,

– *оперативна пам'ять*: LPDDR4-3200 SDRAM,

– *інтерфейси GPIO*: Стандартний 40-піновий роз'єм,

– *мережеві інтерфейси*: Дводіапазонний Wi-Fi (2.4 GHz та 5.0 GHz IEEE 802.11ac) та Gigabit Ethernet, що є критичним для віддаленого керування системою через веб-інтерфейс[12].

Для підключення периферії використовується гребінка GPIO. У проєкті задіяні піни живлення (5V, GND) та пін GPIO 18 (PCM_CLK/PWM0), який підтримує апаратну генерацію сигналу для протоколу WS2812B (див. рис. 3.3).

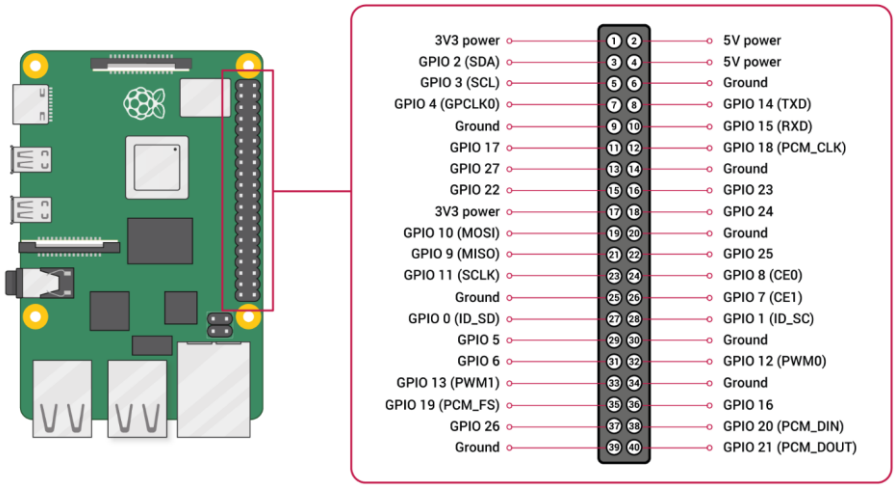


Рисунок 3.3 Схема розпінування GPIO роз’єму Raspberry Pi 4

– *Виконавчий модуль*: Адресна LED-стрічка WS2812B.

На відміну від класичних аналогових RGB-стрічок, де керування кольором відбувається для всієї стрічки одночасно, у проєкті використано цифрову адресну стрічку на базі чіпів WS2812B (див. рис. 3.4). Принцип роботи та структура, полягає в тому, що кожен світлодіод у стрічці є "розумним пікселем", що складається з RGB-діода (SMD 5050) та вбудованого контролера керування.



Рисунок 3.4 Адресна LED-стрічка WS2812B, IP30, 60

Стрічка має три контактні площадки: +5V (живлення), GND (земля) та DIN (вхід даних). Дані передаються послідовним протоколом NZR (Non-Return-to-Zero). Контролер першого світлодіода зчитує перші 24 біти даних (по 8 біт на R, G, B), запам'ятовує їх, а решту пакету даних ретранслює на наступний світлодіод через вихід DO (Data Out) (див рис. 3.5). Це дозволяє індивідуально керувати кожним сегментом освітлення, реалізуючи плавні градієнти.

Основні характеристики модуля:

– *напруга живлення*: 5 В постійного струму (DC),

– *споживання*: До 60 мА на один піксель при білому кольорі,

– *щільність*: 60 світлодіодів на метр.

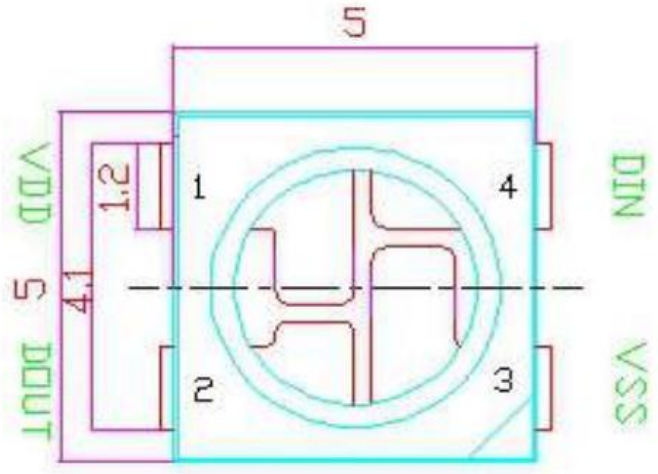


Рисунок 3.5 Внутрішня конфігурація світлодіода WS2812B

– *Система живлення*: Імпульсний блок живлення 5V 10A.

Для забезпечення стабільної роботи системи, яка характеризується високим піковим споживанням струму, обрано промисловий імпульсний блок живлення SMPS (Switching Mode Power Supply) номіналом 5В 10А (50 Вт). Вибір єдиного джерела напруги 5В дозволяє уникнути необхідності використання додаткових DC-DC перетворювачів, оскільки ця напруга є рідною як для Raspberry Pi, так і для стрічки

WS2812B. Особливості конструкції полягають в тому, що блок виконаний у перфорованому металевому корпусі для пасивного охолодження (див. рис. 3.6). На вхідній частині розташовані клеми для підключення мережі змінного струму 220В (L, N, PE), на вихідній – клеми постійного струму (+V, COM). Вбудований потенціометр (+V ADJ) дозволяє точно відкалібрувати вихідну напругу до рівня 5.1-5.2В для компенсації падіння напруги на з'єднувальних проводах.



Рисунок 3.6 Імпульсний блок живлення 5V 10A у перфорованому корпусі

– *Модуль узгодження рівнів*.

Критичним елементом надійності системи є перетворювач логічних рівнів: логіка GPIO Raspberry Pi працює з напругою 3.3В. Логічна одиниця для чіпа WS2812B (відповідно до даташиту) має складати мінімум 0.7×Vcc, що при живленні 5В дорівнює 3.5В.

Вирішенням проблеми є використання двонаправленого перетворювача рівнів (Logic Level Shifter) на базі польових транзисторів (див. рис. 3.7-3.8). Цей модуль підтягує сигнал 3.3В від мікрокомп'ютера до необхідних 5В для стрічки, запобігаючи помилкам передачі даних (мерехтінню, артефактам кольору).

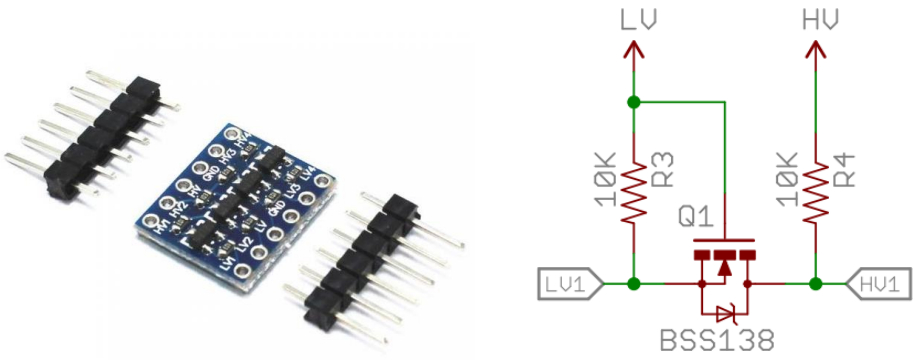


Рисунок 3.7-3.8 Перетворювач рівнів та схема його першої групи контактів

– *Комутаційні елементи та кабельні з'єднання.*

Для силових ліній живлення обрано *кабель калібру 18AWG* у високоякісній силіконовій ізоляції. Такий кабель здатний безпечно пропускати струм до 10-15А, що перекриває максимальне споживання системи з запасом. Силіконова ізоляція забезпечує високу гнучкість кабелю при монтажі та стійкість до температурних навантажень (до 200°C), що важливо при паянні силових роз'ємів[13]. Було використано кабелі 18AWG різної колірної схеми, за стандартом EIA: Червоний (+5V), Чорний (GND), Жовтий (Data/Signal) (див. рис. 3.9). Для підключення блоку живлення до мережі 220В, використано шнур живлення з литою євровилкою.



Рисунок 3.9 З'єднувальні дроти 18AWG згідно стандарту EIA

– *Програмне середовище та носій інформації.*

В якості носія операційної системи використано карту пам'яті MicroSD обсягом 32GB стандарту Class 10 U1 A1. Маркування A1 (Application Performance Class) (див. рис. 3.10) гарантує високу швидкість випадкового читання/запису (IOPS), що є критичним для роботи ОС.



Рисунок 3.10 Носій операційної системи модуля MircoSD 32 GB

На мікрокомп'ютер встановлено Raspberry Pi OS (раніше Raspbian) – офіційний дистрибутив на базі Debian Linux (див. рис. 3.11). Вибір цієї ОС забезпечує найкращу сумісність з бібліотеками керування GPIO (RPi.GPIO, rpi_ws281x) та стабільність роботи серверної частини (Python Flask) у режимі 24/7.

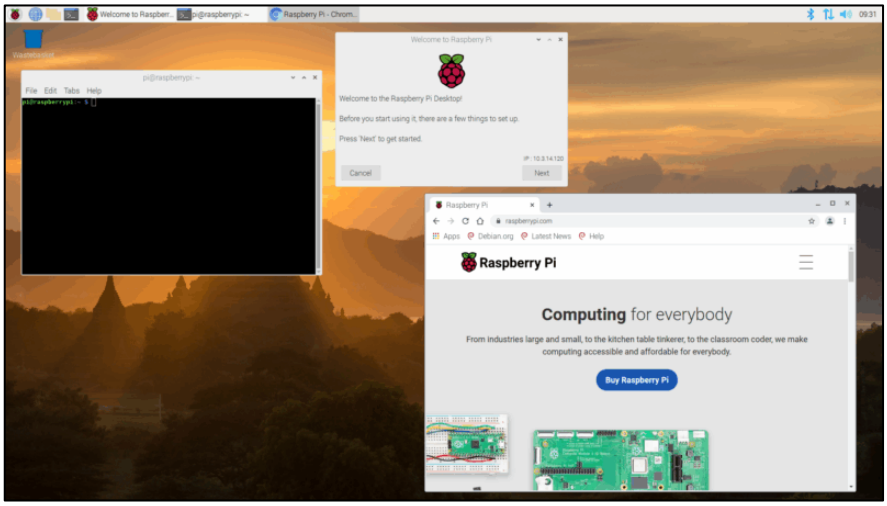


Рисунок 3.11 Графічна оболонка Raspberry Pi OS для керування мікрокомп'ютером

3.2 Побудова та налаштування апаратно-програмного комплексу

Процес інтеграції компонентів системи виконувався відповідно до розробленої структурно-принципової схеми (див. рис. 3.12). Головною вимогою до монтажу було забезпечення надійності електричних з'єднань, спільна шина заземлення для всіх низьковольтних компонентів та гальванічна розв'язка сигнальних ліній.

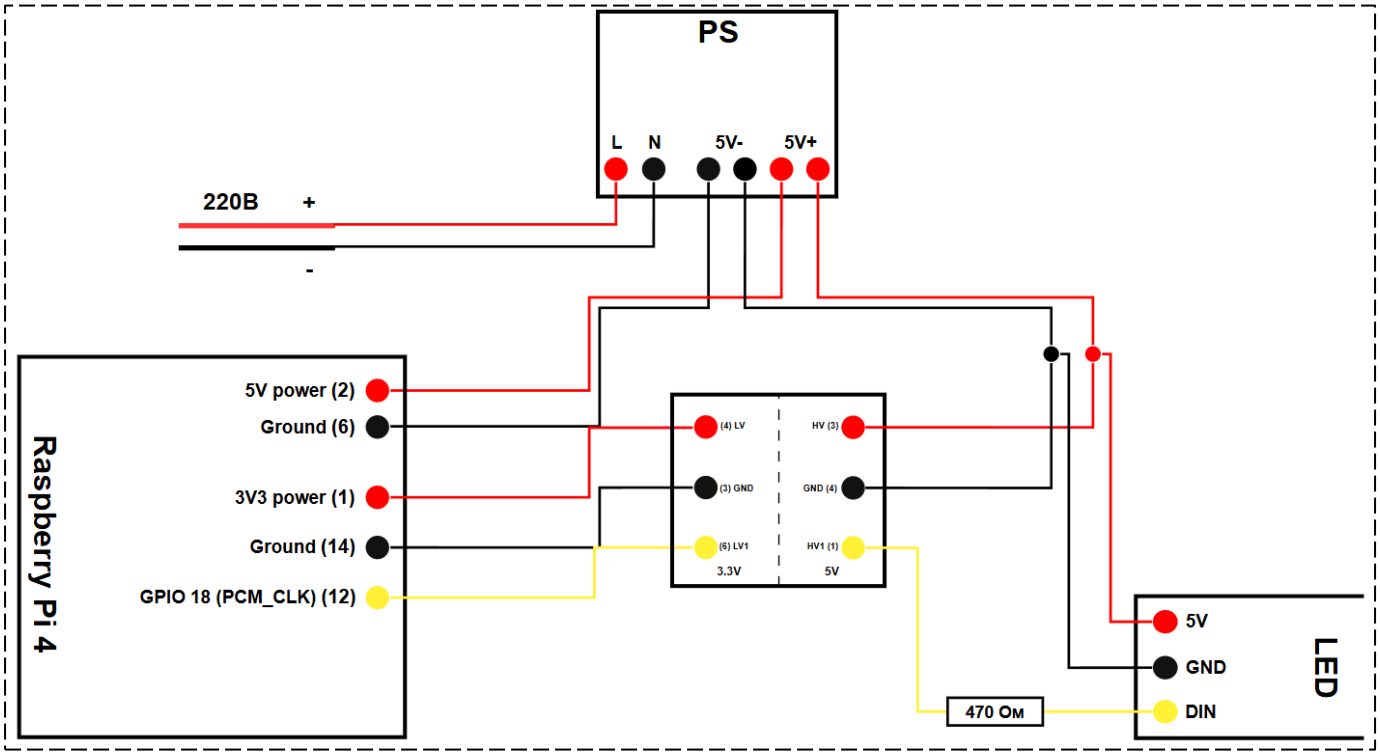


Рисунок 3.12 Структурно-принципова схема підключень системи керування освітленням на базі мікрокомп’ютера Raspberry Pi 4

На схемі (рис. 3.12) зображено підключення модулів проєкту, в першу чергу йде підключення основної напруги, яка запитує весь проєкт, через мережу АС 220В – звичайна європейська вилка з двожильним проводом достатнього поперечного перерізу, підключається у мережу АС 220В, інший край проводу зачищається від ізоляції, дістається два кабелі які в свою чергу також зачищаються від ізоляції, якщо

кабелі багатожильні, вони облужуються та підключаються на клеми імпульсного блоку живлення «L» (фаза) та «N» (нуль), також для більш правильного, стандартизованого, підключення контактів до клем блоку живлення потрібно використовувати кабельні наконечники або гільзи, які надягаються на оголені кінці проводу і затискаються. При використанні кабельних наконечників або гільз важливо також залишати правильну довжину оголеної жили кабелю, щоб після затискання, з під наконечника не виступали оголені жили, які у випадку контакту з елементами приладів або людиною можуть привести як до короткого замикання так і до летальних наслідків. Після підключення блоку живлення до мережі AC 220В, йде підключення живлення на плату мікрокомп'ютера Raspberry Pi 4 Model B, підключення відбувається за допомогою кабелів 18AWG: діаметр провідника – 1,19 мм, площа перетину – 0,75 мм^2, максимальний питомий опір – 0.0395 Ом/м[13], кольори – чорний (нуль), червоний (плюс), жовтий (дані). Від клем «5V+» та «5V-» блоку живлення йдуть червоний та чорний проводи відповідно, червоний провід «плюса» подається другий пін GPIO колодки (5V power), чорний провід «мінуса» подається на шостий пін GPIO колодки (Ground). Після подання живлення на плату мікрокомп'ютера, з других колодок блоку живлення «5V+» та «5V-» виходять ще пара проводів, а саме проводи плюса та мінуса, які на їх кінцях будуть з'єднуватися у розгалуження на два проводи кожне, з метою підключення першої пари «плюса» та «мінуса» до плати 4-ри канального перетворювача логічних рівнів 3.3V на 5V, червоний провід «плюса» підключається до другого контакту High Voltage частини плати, а саме контакт «HV», чорний провід «мінуса» підключається до контакту, тієї ж частини плати, «GND». Друга пара проводів, червоний та чорний підключаються до адресної LED-стрічки, на контакти «5V» та «GND», відповідно. На даному моменті підключення від блоку живлення завершені, наступними йдуть підключення з плати мікрокомп'ютера Raspberry Pi 4: з першого піна GPIO колодки, «3V3 power» йде підключення червоного проводу «плюса» на четвертий контакт «LV» Low Voltage частини плати; з чотирнадцятого піна GPIO колодки, «Ground» йде підключення чорного проводу

«мінуса» на третій контакт «GND» Low Voltage частини плати; з дванадцятого піна GPIO колодки, «GPIO 18 (PCM_CLK)» йде підключення жовтого проводу даних на шостий контакт «LV1» Low Voltage частини плати. Далі продовження підключення з плати 4-х канального перетворювача логічних рівнів: з першого контакту High Voltage частини плати «HV1» виходить жовтий провід даних, який далі підключається на резистор номіналом 470 Ом, для прибирання мерехтіння світлодіодів від «відлуння» в проводі та для захисту першого діоду LED-стрічки: якщо після вмикання блоку живлення він на мікросекунду дасть стрибок напруги, або якщо випадково спочатку подати сигнал даних, а потім живлення – перший чіп у стрічці може згоріти. Резистор обмежує струм і рятує піксель. З резистора жовтий провід даних виходить далі на контакт LED-стрічки «DIN», на який будуть передаватися керуючі ШІМ сигнали.

Подальше налаштування включає у себе завантаження на SD карту пам’яті операційну системи для мікрокомп’ютера Raspberry Pi 4, встановлення карти пам’яті-носія у мікрокомп’ютер, налаштування ОС та її запуск на мікрокомп’ютері, розробка скрипту для автоматизації керування освітленням та його перевірка.

Підключаємо SD карту пам’яті в кардрідер і підключаємо його до ПК або ноутбука, завантажуємо утиліту «Raspberry Pi Imager» з офіційного сайту виробника Raspberry Pi. Запускаємо утиліту Raspberry Pi Imager, після цього на головній сторінці обираємо модель використовуваної плати Raspberry Pi (див. рис. 3.13) та натискаємо кнопку «Далі». Після вибору плати, переходимо в меню вибору операційної системи (див. рис. 3.14), обираємо пункт Raspberry Pi OS (other), та обираємо операційну систему «Raspberry Pi OS Lite (64-bit)».

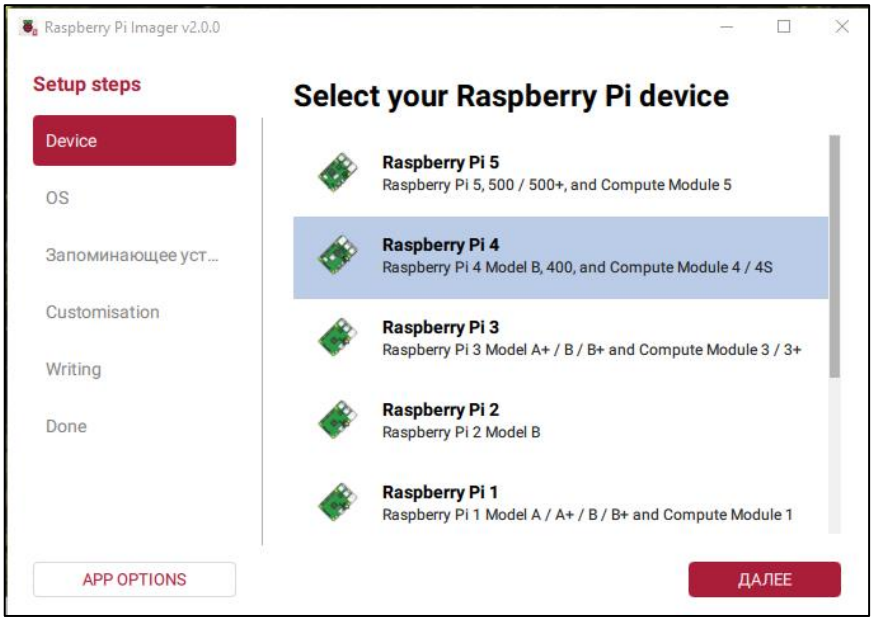


Рисунок 3.13 Меню вибору використовуваної плати Raspberry Pi

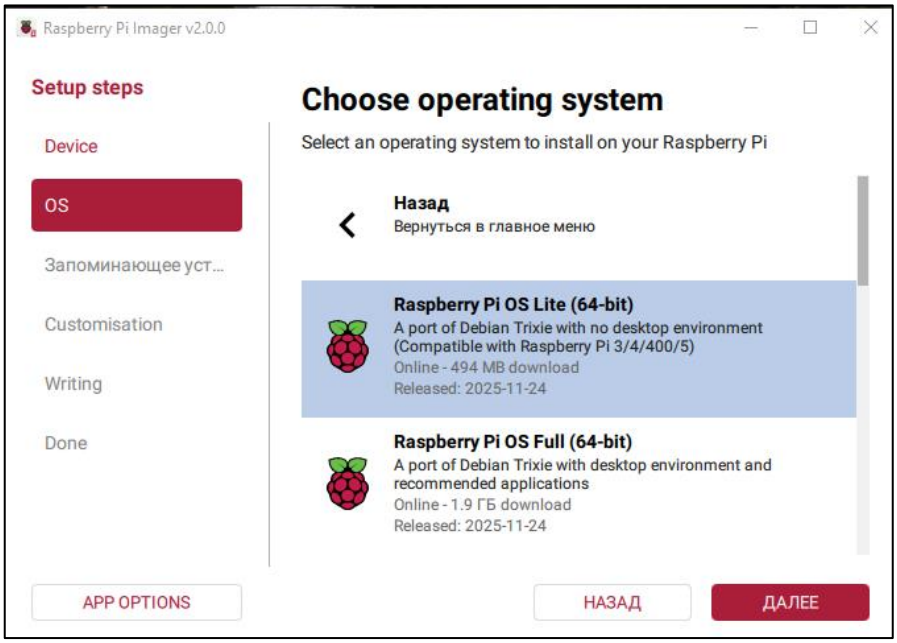


Рисунок 3.14 Меню вибору операційної системи до встановлення

Після, переходимо у наступне меню в якому обираємо підключену через кардрідер SD карту пам’яті (див. рис. 3.15).

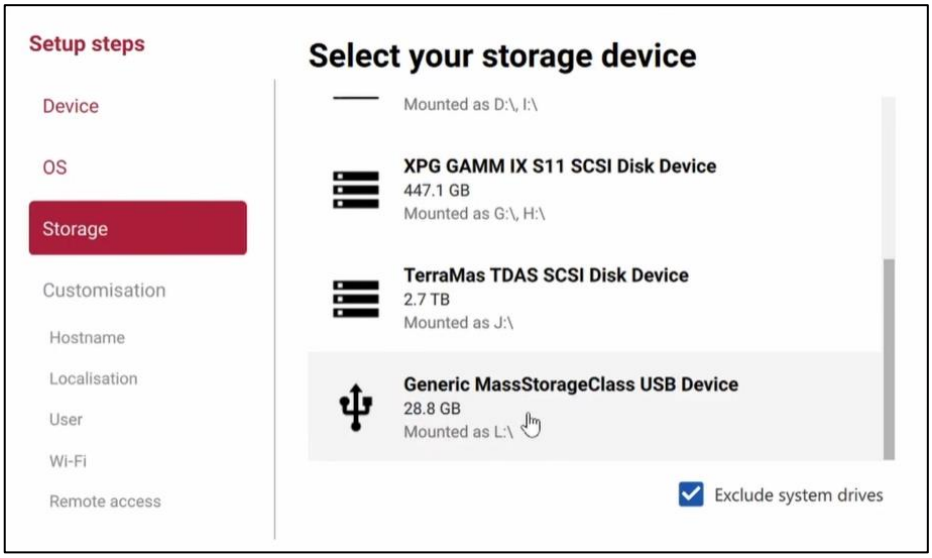


Рисунок 3.15 Меню вибору носія для встановлення ОС

В наступному меню налаштовуємо параметри ОС такі як «host-name», «username», «password» та налаштовуємо мережеве бездротове підключення: вводимо SSID та пароль мережі (див. рис. 3.16), також обов’язково вмикаємо SSH.

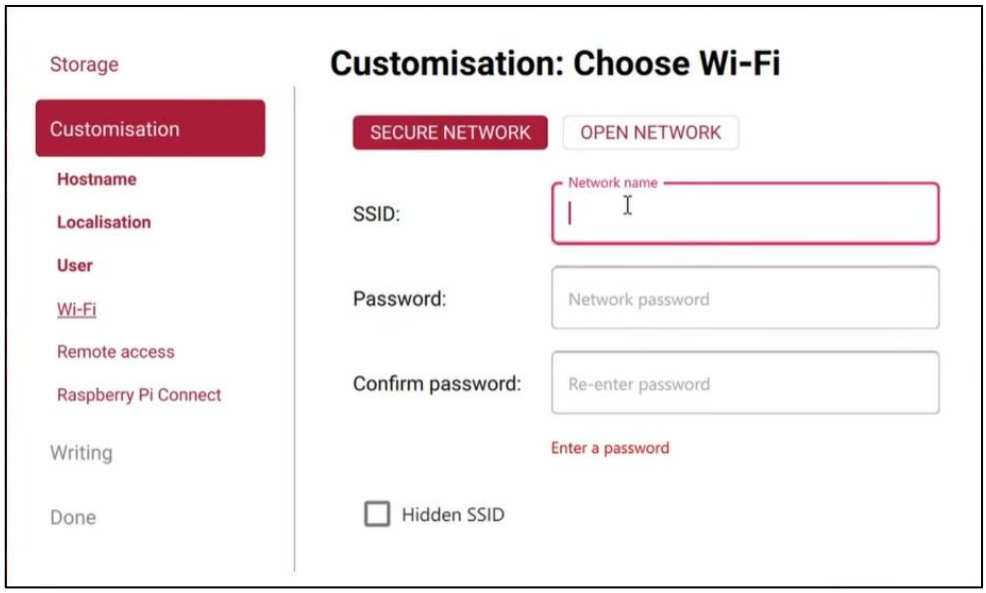


Рисунок 3.16 Меню налаштування бездротового підключення

Коли виконали всі налаштування, натискаємо кнопку «Write» (див. рис. 3.17), щоб записати ОС на носій.

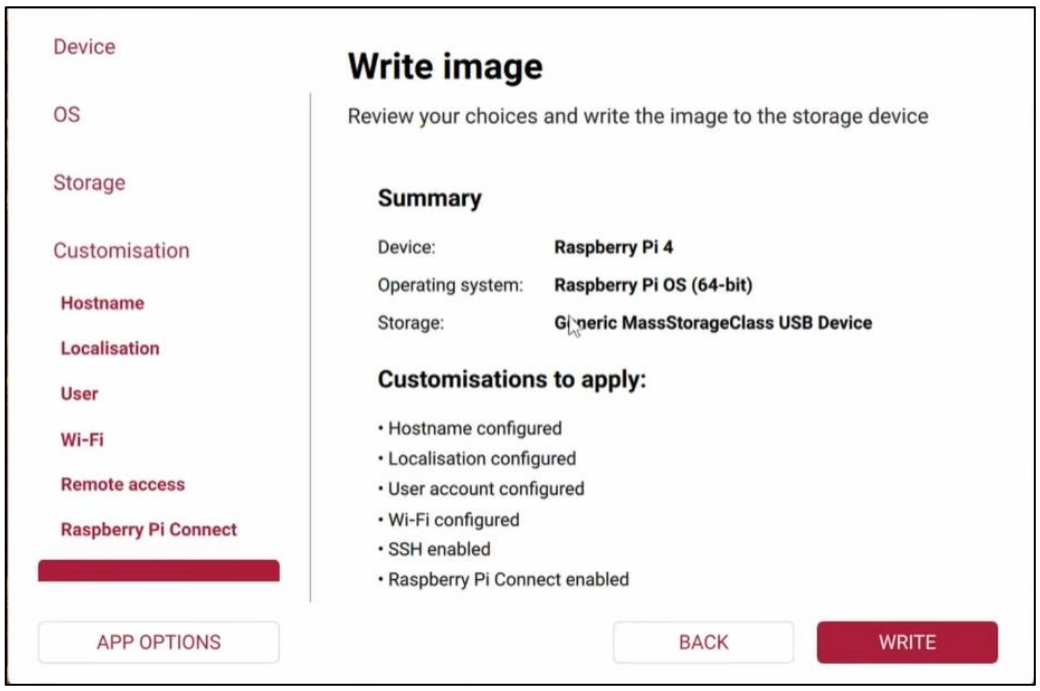


Рисунок 3.17 Меню підтвердження запису ОС на носій

Після закінчення завантаження ОС на носій, витягуємо його з кардрідера, вставляємо SD-карту в слот на Raspberry Pi знизу плати. Ретельно перевіряємо правильність підключення всіх проводів та вмикаємо блок живлення у розетку. Якщо все добре, то на платі мікрокомп’ютера засвітиться червоний світлодіод, який буде сигналізувати, що плата отримує живлення, та буде блимати зелений світлодіод, який показує, що відбувається зчитування даних з SD-карти. Після проходження деякого часу і певної кількості перезавантажень плати, підключаємося до Raspberry Pi по SSH: при введенні налаштувань на етапі запису ОС на носій, утиліта створила в спеціальному розділі «boot» два текстові файли: «*wpa_supplicant.conf*» та «*ssh*»[14], у першому файлі записана назва Wi-Fi та пароль, наявність другого файлу вказує операційній системі, щоб при першому запуску увімкнувся протокол SSH на 22-му порті, через що, на даному етапі Raspberry Pi вже у мережі та очікує підключення,

підключаємося за допомогою команди в командному рядку комп’ютера (див. рис. 3.18):

```
ssh pi@smart-light.local
```

Де «*pi*» – це логін, а «*smart-light*» – це hostname. Після цього вводимо пароль.

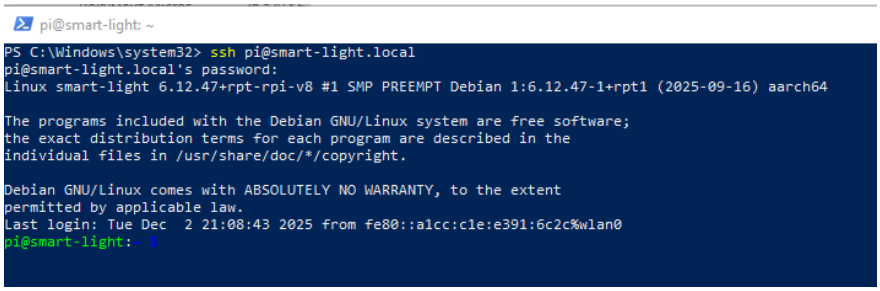


Рисунок 3.18 Підключення до модуля через протокол SSH

Наступний етап, це найважливіша частина, «мозок» системи. Тут реалізується логіка автоматизації, тобто код, та скрипти. Перш за все після підключення по безпечному протоколу SSH до Raspberry Pi, потрібно виконати встановлення ряду бібліотек, драйверів та веб-фреймворку.

В рядок Bash по черзі вписуємо команди:

```
sudo apt-get update
```

```
sudo apt-get install -y python3-pip
```

```
sudo pip3 install rpi_ws281x adafruit-circuitpython-neopixel flask
```

Після чого, оскільки було встановлено версію Lite (без графіки), потрібно буде використовувати вбудований текстовий редактор «nano» для створення файлу конфігурації зі скриптом, цей редактор дуже простий і працює прямо в терміналі. В командному рядку з підключенням до Raspberry Pi вписуємо команду яка запустить редактор «nano» та створить файл:

```
nano main.py
```

Прописуємо в коді логіку роботи, інтерфейс та скрипт за яким буде працювати автоматизація в системі освітлення. Після написання коду, для збереження файлу потрібно натиснути комбінацію клавіш «Ctrl + O», після чого система запитає підтвердження запису файлу під певним ім'ям, натискаємо «Enter», після чого натискаємо комбінацію клавіш «Ctrl + X» для виходу з редактору. Для перевірки правильності виконаних дій, виконуємо команду:

```
ls
```

Ця команда відобразить каталог (див. рис. 3.19), у цьому каталозі має бути створений раніше файл.



Рисунок 3.19 Перевірка правильності виконаних дій та наявності файлу скрипта

Наступний етап це запуск файлу зі скриптом та його перевірка, для чого запускаємо наступну команду:

```
sudo python3 main.py
```

Примітка №1. Повний код скрипту розташований у додатку А.

Після перевірки справності (див. рис. 3.20), за необхідності, можна додавати любий другий блок коду для світлодіодної стрічки, проте на момент тесту у скрипта стоїть задача адаптивного освітлення, яким можна керувати вручну.

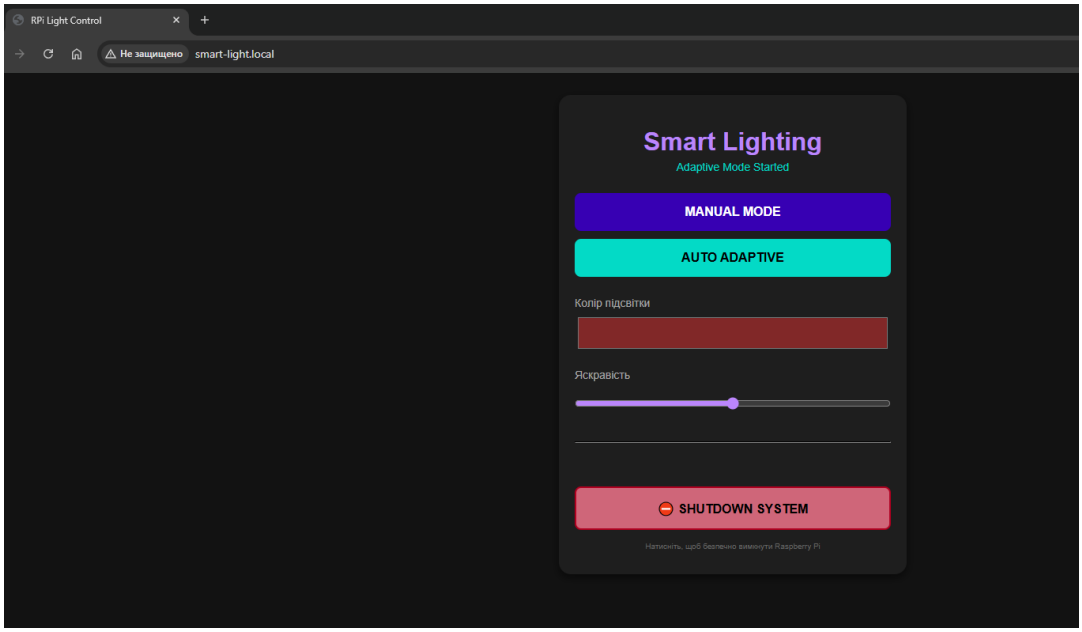


Рисунок 3.20 Веб-застосунок для керування світлодіодним освітленням

Для даного скрипту функціонал включає в себе вибір двох режимів: «Ручний режим», «Авто режим». При ввімкненні «Ручного режиму», система буде використовувати встановлені в ручну яскравість та колірну гаму стрічки які можна зручно змінювати. При ввімкненні «Авто режиму», система кожні 30 секунд виконує зчитування поточного часу доби, після чого звіряє з передвстановленими правилами у скрипті, наприклад, якщо час доби від 8:00 до 12:00, то світло встановлюється у режим №1, у якому світло тепле, з яскравістю у 80%; якщо час доби від 12:00 до 16:00, то світло встановлюється у режим №2, у якому світло більш «холодне», з яскравістю у 100%. У графі «Колір підсвітки», можна вручну вибрати кольорову гаму підсвітки LED-стрічки, у графі «Яскравість», можна вручну виставити рівень яскравості світлодіодної стрічки. Червона кнопка «SHUTDOWN SYSTEM» слугує для безпечного вимкнення модуля освітлення з метою його подальшого відключення від живлення, на що і вказує підказка під кнопкою, для захисту від випадкового натиснення на кнопку вимкнення системи, стоїть подвійне підтвердження із pop-up-сповіщенням з можливістю відмінити дію.

Після зашиття скрипта в модуль адаптивного керування освітленням та тестового запуску веб-застосунка було проведено повне підключення всіх компонентів модуля (див. рис. 3.21) і проведений заключний тест. Було випробувано правильність зміни яскравості та кольорової гами освітлення (див. рис. 3.22-23).

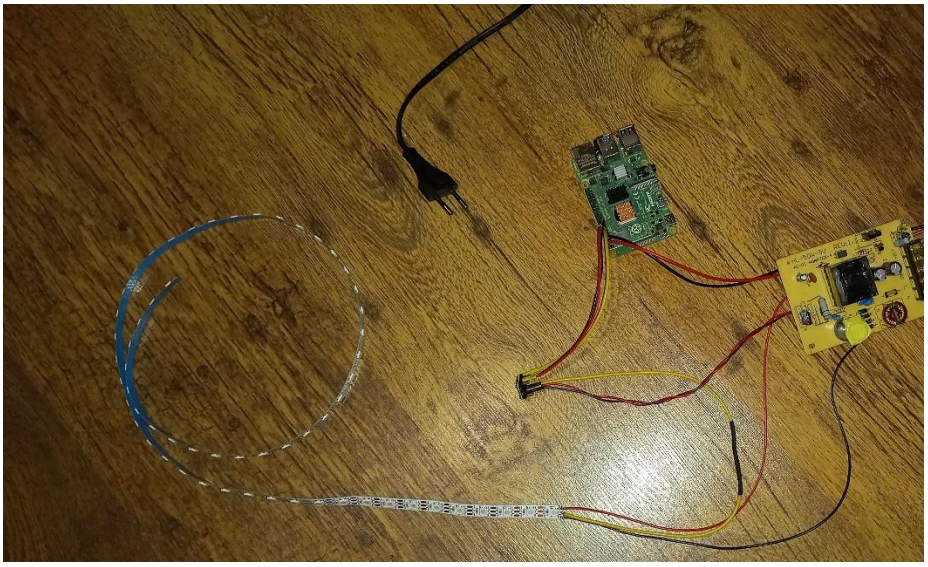


Рисунок 3.21 Фото повністю зібраного модуля адаптивного освітлення

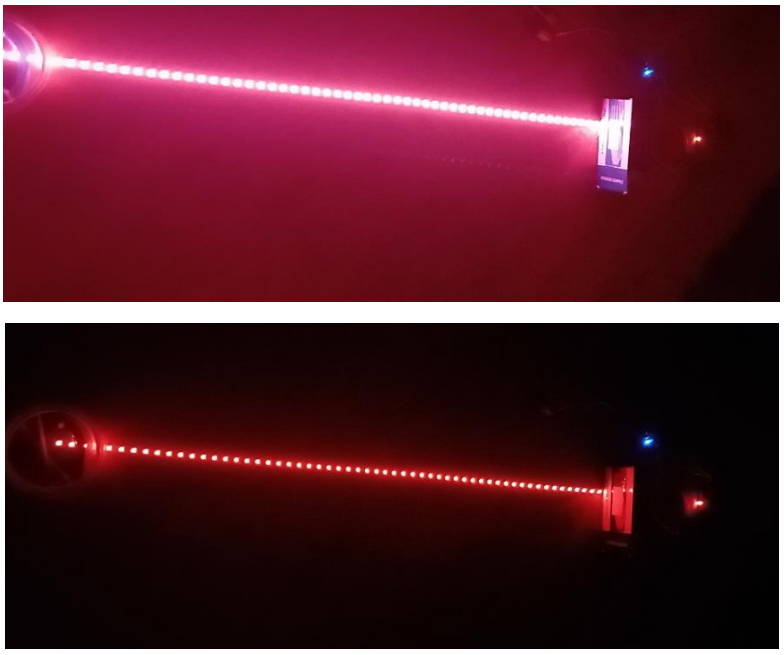


Рисунок 3.22-23 Візуалізація керування яскравістю та кольоровою гамою

3.3 Розробка програмного забезпечення та алгоритмів керування

Для реалізації логіки адаптивного освітлення було обрано клієнт-серверну архітектуру. Це дозволяє розділити процеси безпосереднього керування апаратною частиною (Low-level control) та взаємодію з користувачем (High-level interface). Програмний комплекс розроблено мовою програмування Python 3, яка є стандартом де-факто для систем IoT завдяки наявності потужних бібліотек для роботи з GPIO та веб-сервісами.

Архітектура програмного комплексу складається із системи з трьох основних модулів, що працюють паралельно в багатопотоковому режимі:

1. Модуль керування світлодіодами (LED Controller), використовує бібліотеку «rpi_ws281x» для генерації ШІМ-сигналу[15]. Цей модуль працює на найнижчому рівні, звертаючись до DMA (Direct Memory Access) контролера Raspberry Pi, що дозволяє формувати високочастотний сигнал без завантаження центрального процесора.

2. Модуль адаптивної логіки,, (Adaptive Core) є фоновим процесом (Daemon Thread), який запускається при старті системи. Його задача циклічно (кожні 30 секунд) опитувати системний час, порівнювати його з заданими часовими інтервалами (ранок, день, вечір, ніч) та розраховувати цільову колірну температуру. Алгоритм розрахунку кольору базується на моделі циркадних ритмів людини (див. рис. 3.24):

- 06:00 – 10:00 (Фаза пробудження): Плавний перехід від теплих тонів до нейтральних.
- 10:00 – 18:00 (Активна фаза): Холодне біле світло (RGB: 255, 255, 255) для стимуляції концентрації.
- 18:00 – 22:00 (Фаза розслаблення): Тепле світло (RGB: 255, 147, 41), що імітує захід сонця.

- 22:00 – 06:00 (Нічна фаза): Мінімальна яскравість, переважання червоного спектру (RGB: 50, 0, 0) для збереження вироблення мелатоніну.

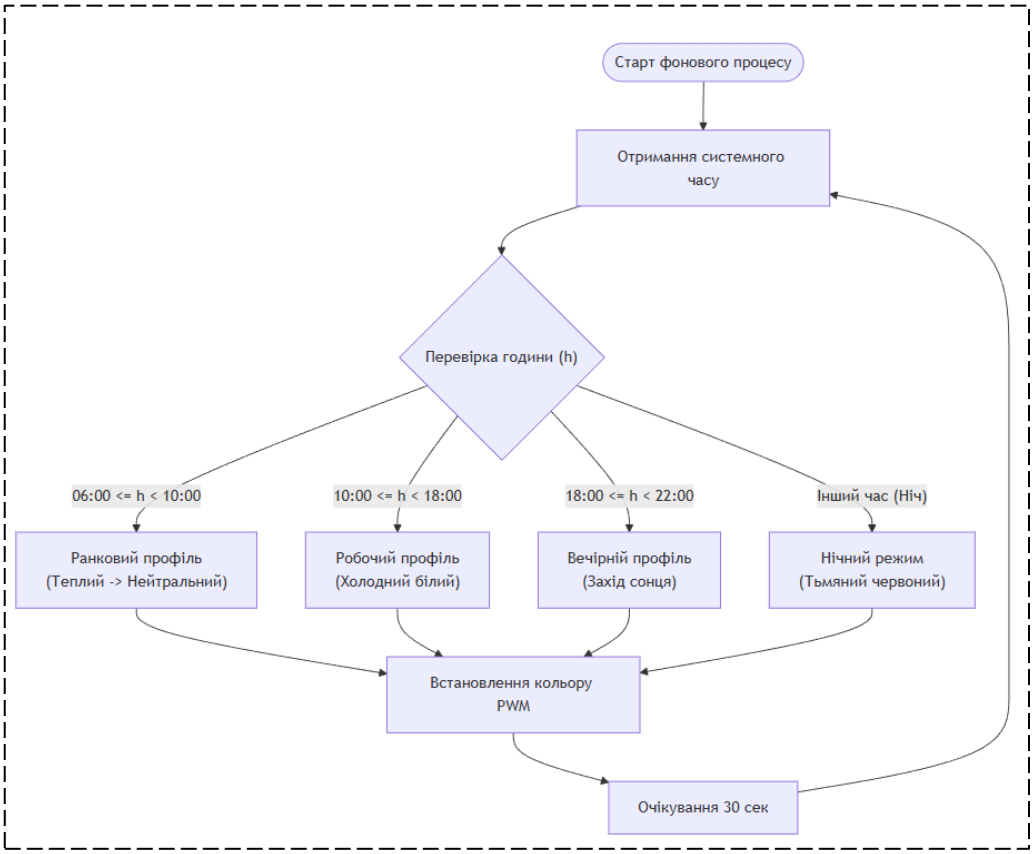


Рисунок 3.24 Блок-схема роботи алгоритму зі зміни освітлення за часом

3. Веб-сервер та API (Web Interface), реалізовані на базі фреймворку Flask. Він забезпечує REST API інтерфейс для прийому команд від користувача (зміна режиму, ручне налаштування кольору). Веб-інтерфейс є адаптивним і доступний з будь-якого пристрою в локальній мережі за IP-адресою мікрокомп'ютера.

Основний цикл керування реалізовано у класі *LightingController*. Для уникнення конфліктів при одночасному доступі до стрічки (наприклад, коли автоматичний режим намагається змінити колір одночасно з ручною командою користувача), використано механізм блокування потоків (*threading.Lock*). Такий підхід забезпечує автономність системи після запуску мікрокомп'ютера скрипт автоматично

завантажується службою *systemd* і починає керування освітленням без участі оператора.

3.4 Техніко-економічне обґрунтування розробки

Важливим етапом проєктування автоматизованих систем є оцінка їх економічної ефективності. Для обґрунтування доцільності впровадження розробленої системи адаптивного освітлення на базі Raspberry Pi проведено порівняльний аналіз вартості компонентів із вартістю аналогічних готових рішень, представлених на ринку. Як об'єкт для порівняння обрано екосистему Philips Hue, яка є лідером у сегменті розумного освітлення, але характеризується високою вартістю та закритим програмним кодом.

Розрахунок собівартості розробленого прототипу, розрахунок капітальних витрат на апаратну частину системи виконано на основі середніх роздрібних цін на компоненти в Україні станом на момент написання роботи. До кошторису включено всі елементи, необхідні для побудови автономного модуля (див. табл. 3.1).

Таблиця 3.1

Кошторис витрат на розробку системи автоматизації керування освітленням

№	Найменування компонента	Кількість	Ціна за од., грн	Сума, грн	Примітка
1	Мікрокомп'ютер Raspberry Pi 4 Model B (4GB)	1	3271	3271	Основний контролер
2	Блок живлення імпульсний 5V 10A (50W)	1	320	320	Живлення системи
3	Адресна LED-стрічка WS2812B (IP30, 60 led/m)	2	159	318	Виконавчий елемент

Продовження таблиці 3.1

№	Найменування компонента	Кількість	Ціна за од., грн	Сума, грн	Примітка
4	Карта пам'яті MicroSD 32GB Class 10	1	115	115	Носій ОС
5	Перетворювач логічних рівнів (3.3V-5V)	1	14	14	Узгодження сигналів
6	Витратні матеріали	-	104	104	Монтаж
Загальна вартість			**4 142 грн**		

Аналіз вартості аналогічного рішення (Philips Hue). Для реалізації аналогічного функціоналу (керування через Wi-Fi, адаптивні сценарії, довжина стрічки 2 метри) у екосистемі Philips Hue необхідно придбати наступний комплект обладнання:

1. Philips Hue Bridge – шлюз керування (обов'язковий для роботи сценаріїв автоматизації та віддаленого доступу).
2. Philips Hue Lightstrip Plus v4 (2m Base Kit) – базовий комплект стрічки з блоком живлення.

Зведення кошторису аналогічного рішення приведено в таблиці 3.2.

Таблиця 3.2

Кошторис витрат на закупівлю готової системи керування освітленням

№	Найменування компонента	Ціна, грн	Функціональні обмеження
1	Philips Hue Bridge 2.1	3 290	Працює лише з пристроями ZigBee
2	Philips Hue Lightstrip Plus v4 (2m)	2 990	Аналогове керування (вся стрічка одного кольору)
Загальна сума		**6 280 грн**	

Порівняння двох підходів дозволяє виділити економічну та функціональну вигоду розробки. Порівняльна характеристика двох рішень наведені у таблиці 3.3.

Таблиця 3.3

Порівняльний аналіз кошторису та гнучкості систем

Параметр	**Розроблена система (Raspberry Pi)**	**Philips Hue (Vendor Solution)**
Вартість стартового комплекту	4 142	6 280
Тип освітлення	Адресне (керування кожним діодом)	Аналогове RGB (один колір на всю довжину)
Вартість масштабування (+1 метр стрічки)	~159 грн	~1 400 грн (фірмове подовження)
Залежність від інтернету/хмари	Ні (Повна локальна автономність)	Так (Для віддаленого керування потрібна хмара)
Додаткові можливості	Необмежені (Медіа-сервер, Блокувальник реклами тощо)	Тільки освітлення

За результатами проведеного аналізу встановлено, що розроблена система є економічно вигіднішою за існуючі ринкові аналоги (зокрема, Philips Hue) на приблизно 34%, що у грошовому еквіваленті становить економію близько 2138 грн лише на стартовому комплекті обладнання. Така суттєва різниця зумовлена відсутністю маркетингової націнки за бренд та використанням універсальних компонентів відкритої архітектури замість спеціалізованих пропрієтарних рішень.

Окрім прямої фінансової вигоди на етапі впровадження, система на базі Raspberry Pi демонструє значно нижчу сукупну вартість володіння (TCO — Total Cost of Ownership) у довгостроковій перспективі, особливо при масштабуванні проєкту. Додавання нових зон освітлення або заміна компонентів, що вийшли з ладу (наприклад, блоку живлення), вимагає купівлі стандартних радіодеталей, доступних

на ринку за низькими цінами. Натомість, у закритих екосистемах користувач потрапляє у технологічну залежність від виробника (Vendor Lock-in) і змушений купувати коштовні брендові модулі розширення або спеціалізовані аксесуари, що не мають альтернатив. Важливим техніко-економічним аспектом є також коефіцієнт використання обладнання. Обчислювальна потужність мікрокомп'ютера Raspberry Pi 4 Model B у реалізованому проєкті задіяна лише на 5-10% (дані отримані з аналізу навантажень які було виведено через команду «htop»[16]). Цей надлишковий ресурс створює значний потенціал для функціонального розширення системи без додаткових капітальних інвестицій. На базі того ж контролера паралельно можна розгорнути сервер відеонагляду, систему голосового керування, блокувальник мережевої реклами або MQTT-брокер для інших датчиків розумного будинку, що робить питому вартість автоматизації однієї функції ще нижчою у порівнянні з вузькоспеціалізованими шлюзами від вендорів на ринку.

3.5 Можлива модернізація адаптивності системи освітлення

Розроблена у попередніх підрозділах система працює за принципом розімкненого циклу керування (Open Loop Control) в якому параметри освітлення змінюються виключно на основі часового розкладу або ручним керуванням, без урахування реальних умов у приміщенні. Однак, у сонячний день штучне освітлення може бути зайвим, а у похмуру погоду – недостатнім, навіть у денний час. Для реалізації повноцінної системи Human-Centric Lighting, пропонується модернізація апаратної частини шляхом введення контуру зворотного зв’язку. Ключовим елементом модернізації є інтеграція цифрового датчика зовнішньої освітленості.

Для точного вимірювання світлового потоку обрано цифровий сенсор BH1750 (див. рис. 3.25) або як його аналог можна обрати датчик TSL2561.

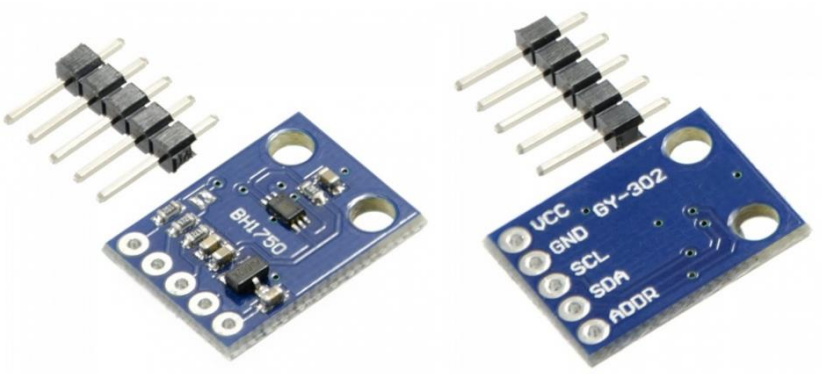


Рисунок 3.25 Датчик освітленості цифровий BH1750, вид зверху та знизу

На відміну від дешевих фоторезисторів, які видають нелінійний аналоговий сигнал і потребують зовнішнього аналого-цифрового перетворювача (оскільки Raspberry Pi не має аналогових входів), BH1750 є цифровим пристроєм[17]. Він видає значення безпосередньо у Люксах (Lux), має спектральну чутливість, близьку до людського ока, та підключається по шині I2C.

Технічні характеристики датчика:

– діапазон вимірювання: 1 - 65535 лк;
– інтерфейс: I2C (Inter-Integrated Circuit);
– напруга живлення: 3,3В - 5В.

Схема підключення та апаратна інтеграція здійснюється до стандартного інтерфейсу I2C на гребінці GPIO Raspberry Pi (див рис. 3.26). Це вимагає лише чотирьох проводів і не заважає роботі світлодіодної стрічки, оскільки використовуються інші піни.

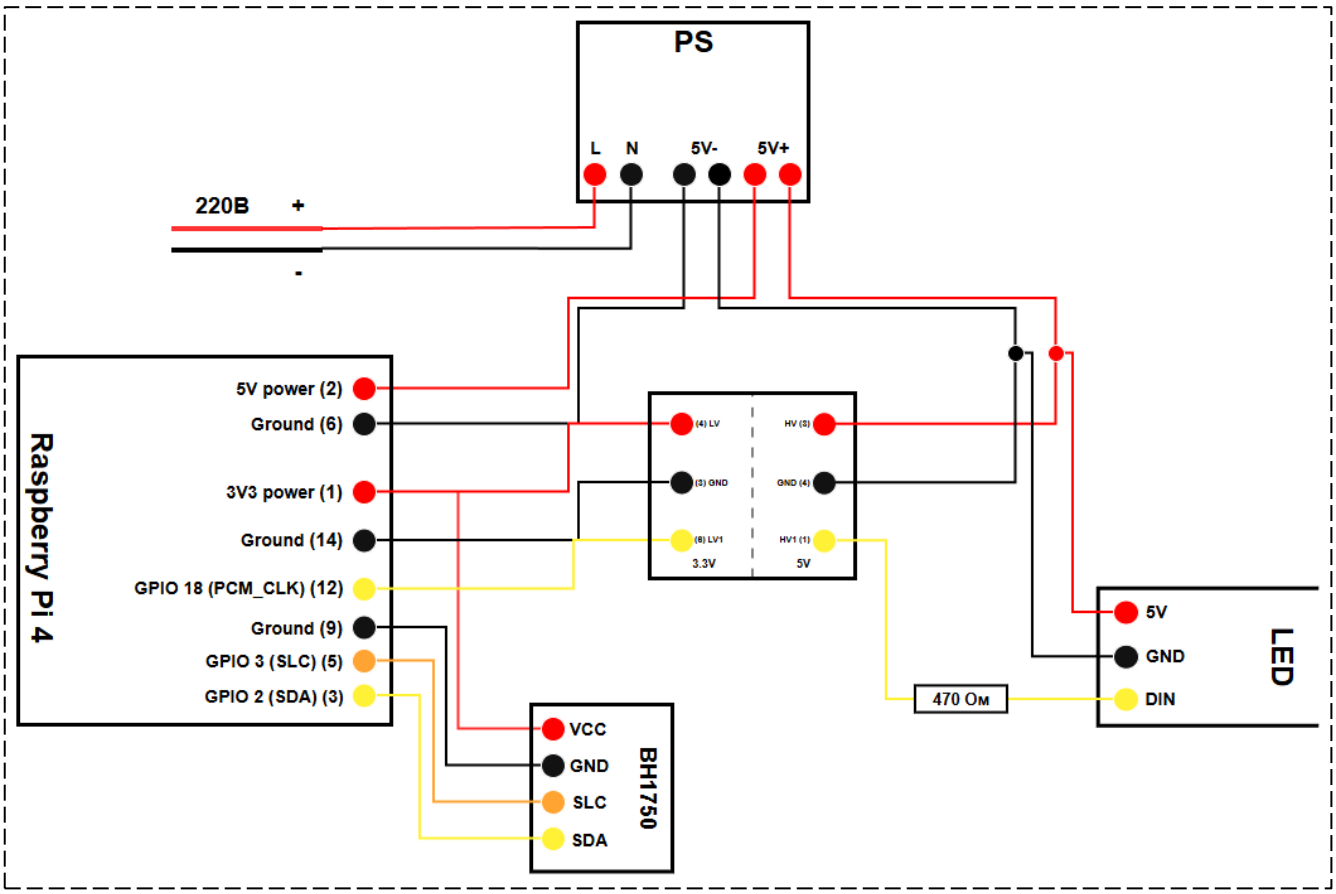


Рисунок 3.26 Схема підключення датчика BH1750 до інтерфейсу I2C

Введення датчика змінює логіку роботи головного скрипта. Тепер яскравість стрічки залежить не від жорстко заданого значення, а від формули компенсації (3.1):

$$L_{LED} = L_{Target} - L_{Ambient}, \quad (3.1)$$

де: L_{Target} – цільовий рівень освітленості згідно з санітарними нормами;

$L_{Ambient}$ – поточний рівень природного світла, виміряний датчиком;

L_{LED} – необхідна яскравість стрічки.

Нова логіка програмного коду полягає в тому, що система працює в режимі постійного моніторингу (Polling). В цій системі отримання даних відбувається так, що раз на 5 секунд зчитуються значення з датчика BH1750. Як і в попередній версії,

відбувається визначення час доби для вибору колірної температури «теплої чи холодної». Корекція яскравості відбувається за наступним принципом:

– якщо датчик показує більше 500 лк (сонячно), то стрічка вимикається або переходить у режим мінімального підсвічування для економії енергії.

– якщо датчик показує менше 100 лк (похмуро/вечір), то стрічка плавно піднімає яскравість до 100%, компенсуючи нестачу світла.

Нижче на рисунку 3.27, наведено фрагмент коду для інтеграції роботи з датчиком:

Smart-light-local.py > ...

```python
import smbus # Бібліотека для I2C

def read_light():
    # Адреса датчика BH1750 за замовчуванням 0x23
    DEVICE = 0x23
    bus = smbus.SMBus(1)
    # Читання даних (режим High Resolution)
    data = bus.read_i2c_block_data(DEVICE, 0x11)
    # Конвертація байтів у число Люксів
    lux = (data[1] + (256 * data[0])) / 1.2
    return lux

def adaptive_smart_loop():
    while True:
        current_lux = read_light()
        hour = datetime.datetime.now().hour

        # Базова логіка температури світла (Колір)
        if 8 <= hour < 18: color = (255, 255, 255) # День
        else: color = (255, 140, 50)               # Вечір

        # Логіка компенсації яскравості (Brightness)
        if current_lux > 500:
            target_brightness = 0  # Світла достатньо
        elif current_lux < 100:
            target_brightness = 100 # Темно, вмикаємо на повну
        else:
            # Лінійна інтерполяція для проміжних значень
            target_brightness = 100 - ((current_lux - 100) / 4)

        set_led_state(color, target_brightness)
        time.sleep(5)
```

Рисунок 3.27 Додатковий код для роботи датчика в системі освітлення

Запропонована модернізація дозволяє перетворити систему на повністю автономний комплекс клімат-контролю освітлення. Використання датчика BH1750

вартістю близько 55 грн дозволяє додатково заощадити в електроенергії у денний час, уникаючи роботи освітлення, коли воно не потрібне, та гарантує стабільний рівень освітленості на робочому місці незалежно від погодних умов за вікном.

Для того, щоб все працювало потрібно встановити бібліотеку для роботи з шиною I2C:

```
sudo apt-get install -y i2c-tools python3-smbus
```

Код налаштований так, щоб система шукала підключений датчик освітленості, у випадку якщо датчик відсутній, скрипт показує, що датчик освітленості недоступний.

Примітка №2. Повний код модифікованого скрипту під датчик освітленості, розташований у додатку Б.

Модифікований під інтеграцію датчика освітлення інтерфейс показано на рисунку 3.28.

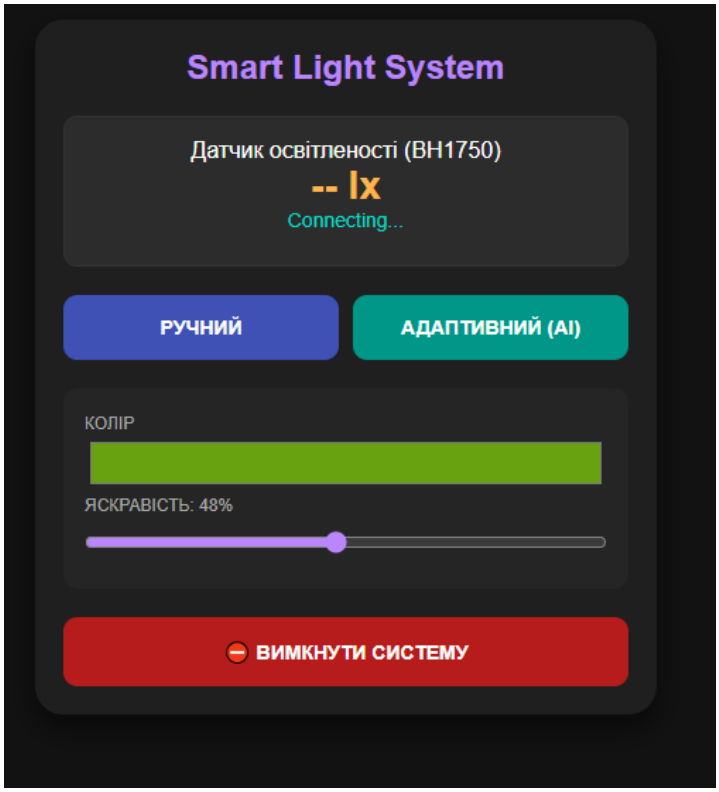


Рисунок 3.28 Модифікований під датчик освітлення інтерфейс веб-додатку

ВИСНОВКИ

У кваліфікаційній роботі проведено комплексне дослідження та практичну реалізацію автоматизованої системи адаптивного освітлення в екосистемі «Smart Home» із застосуванням мікрокомп’ютера Raspberry Pi. Результати роботи дозволяють зробити наступні висновки відповідно до поставлених завдань.

Теоретичний аналіз предметної області показав, що автоматизація освітлення є ключовим елементом сучасних інтелектуальних будівель, який поєднує в собі задачі енергозбереження та забезпечення комфорту. Проаналізовано класифікацію автоматизованих систем, принципи організації світлового середовища та сучасні протоколи керування (ZigBee, DALI, KNX, Matter). Встановлено, що найбільш перспективним напрямом є впровадження адаптивних систем, які динамічно змінюють параметри освітлення (яскравість, колірну температуру) відповідно до біоритмів людини, що позитивно впливає на самопочуття та продуктивність праці. Аналіз апаратних платформ у сфері технологій розумного дому виявив суттєві обмеження готових комерційних рішень, такі як закритість екосистеми, обмежена функціональність та висока вартість масштабування. Проведено порівняльний аналіз ринку одноплатних мікрокомп’ютерів (SBC), розглянуто еволюцію лінійки Raspberry Pi та альтернативних рішень (Orange Pi, ASUS Tinker Board, NVIDIA Jetson). Обґрунтовано вибір Raspberry Pi 4 Model B як оптимальної платформи для побудови центрального контролера освітлення. Дана модель забезпечує найкращий баланс між обчислювальною потужністю, розвиненою периферією (GPIO, Gigabit Ethernet), стабільністю програмного забезпечення та вартістю, залишаючи значний запас ресурсів для подальшого розширення функціоналу. Практична реалізація системи підтвердила життєздатність концепції побудови модульних рішень на базі компонентів відкритої архітектури. Розроблено структурно-принципову схему пристрою, що включає мікрокомп’ютер, адресну світлодіодну стрічку WS2812B,

імпульсний блок живлення та модуль узгодження логічних рівнів. Виконано фізичний монтаж компонентів із забезпеченням гальванічної розв'язки сигнальних ліній та стабілізації живлення. Застосування прямого керування через інтерфейс GPIO з використанням DMA (Direct Memory Access) дозволило досягти стабільної роботи системи реального часу без використання спеціалізованих мікроконтролерів-посередників. Програмна складова, реалізована мовою Python, забезпечує повну автономність роботи системи. Розроблені алгоритми адаптивного керування успішно імітують природний добовий цикл освітлення, автоматично змінюючи спектр від теплих тонів вранці та ввечері до холодного білого світла вдень. Реалізація багатопотокової архітектури дозволила поєднати фонове виконання алгоритмів автоматизації з миттєвою реакцією на команди користувача через розроблений веб-інтерфейс. Техніко-економічне обґрунтування довело високу ефективність власної розробки. Розрахунок собівартості показав, що стартовий комплект системи на базі Raspberry Pi коштує на 34% менше за функціональний аналог від провідного вендора (Philips Hue), що у грошовому еквіваленті становить економію понад 2000 грн. Окрім прямої економії, розроблена система демонструє значно нижчу вартість володіння (TCO) у довгостроковій перспективі завдяки відсутності прив'язки до бренду та можливості використання недорогих стандартних компонентів для ремонту чи розширення. Аналіз продуктивності виявив, що реалізований функціонал адаптивного освітлення задіює лише 5–10% обчислювальних ресурсів Raspberry Pi 4. Цей факт підтверджує доцільність використання мікрокомп'ютера як універсального сервера розумного будинку, здатного паралельно виконувати функції відеоспостереження, голосового асистента, медіа-сервера або шлюзу для інших IoT-пристроїв без додаткових капітальних витрат. Було розроблено програмний комплекс мовою Python із використанням фреймворку Flask, що забезпечує повну автономність системи. Реалізовано клієнт-серверну архітектуру, де серверна частина керує фізичними процесами у фоновому режимі, а клієнтська надає зручний доступ до керування через REST API у локальній мережі. Базовий алгоритм адаптивності успішно імітує

природний добовий цикл, автоматично змінюючи колірну температуру від 2700К (теплий спектр) вранці та ввечері до 6500К (холодний спектр) у робочі години, що відповідає санітарно-гігієнічним нормам освітлення робочих місць. Модернізація системи була виконана для підвищення енергоефективності та рівня адаптивності, систему було модернізовано шляхом інтеграції цифрового датчика освітленості BH1750. Це дозволило перейти від розімкненого циклу керування (лише за таймером) до замкнутого контуру зі зворотним зв’язком (Closed Loop Control). Також було розроблено та імплементовано алгоритм компенсації природного світла, який автоматично регулює яскравість світлодіодної стрічки залежно від фактичного рівня освітленості у приміщенні. Таке рішення дозволяє досягти додаткової економії електроенергії у світлий час доби та гарантує стабільний рівень освітленості незалежно від погодних умов.

Таким чином, у магістерській роботі вирішено актуальне науково-прикладне завдання створення доступної, енергоефективної та гнучкої системи освітлення, яка за своїми певними технічними та економічними показниками перевершує існуючі пропрієтарні аналоги.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шевченко В. В., Тимчик Г. С. Основи автоматизації технологічних процесів : курс лекцій. Київ : Нац. Техн. Ун-т України «Київ. Політехн. Ін-т Ім. Ігоря Сікорського», 2023. 111 с.
2. Охорона праці - Бібліотека BukLib.net. Головна - Бібліотека BukLib.net. URL: https://buklib.net/books/35165/ (дата звернення: 15.10.2025).
3. ДБН В.2.5-28:2018. Природне І Штучне Освітлення. Чинний від 2019-03-01. Вид. офіц. Київ : Портал Єдин. держ. електрон. системи у сфері буд-ва, 2018. 137 с.
4. Belikov A., Zhurbenko V. RESEARCH AREAS TO IMPROVE LIGHTING COMFORT OF THE ENVIRONMENT. *JOURNAL of Donetsk Mining Institute*. 2022. № 1. С. 7–15. URL: https://doi.org/10.31474/1999-981x-2022-1-7-15 (дата звернення: 17.10.2025).
5. Arun M., Gopan G. Effects of natural light on improving the lighting and energy efficiency of buildings: toward low energy consumption and CO2 emission. International Journal of Low-Carbon Technologies. 2025. Т. 20. С. 1047–1056. URL: https://doi.org/10.1093/ijlct/ctaf057 (дата звернення: 10.10.2025).
6. Інтернет-речей в Розумному Домі (Smart Home). PandaTeman.net. URL: https://pandateam.net.ua/razrabotka-iot-smart-home/ (дата звернення: 17.10.2025).
7. Raspberry Pi 4 Model B 4GB купити в Києві та Україні. Arduino в Україні. URL: https://arduino.ua/prod3308-raspberry-pi-4-4gb?srsltid=AfmBOoqd43sHqmgbaNFV02-f7eeL5JvYucZMeYuXc0roYCb1pV76xQfi (дата звернення: 17.10.2025).
8. Кончаківський Р. Що таке домашня автоматизація, і як вона працює | Блог Ajax. Ajax Systems. URL: https://ajax.systems/ua/blog/what-is-home-automation-and-how-does-it-work/ (дата звернення: 15.10.2025).

9. Що таке протокол для розумного будинку? 6 популярних видів. Інтернет-магазин WiseHome. URL: https://wisehome.ua/ua/scho-take-protokol-dlya-rozumnogo-budinku-6-populyarnih-vidiv/?srsltid=AfmBOor0nBXT8B3NW_fjb2swynJ7EPMAlpsRyAxoA3o1NCBMr2s2B3pL (дата звернення: 17.10.2025).

10. Hostragons Global Limited. Найкращі операційні системи та посібник з встановлення для Raspberry Pi. Hostragons. URL: https://www.hostragons.com/uk/блог/посібник-з-встановлення-операційних/ (дата звернення: 20.10.2025).

11. Що таке мікрокомп'ютери? Розповідаємо на прикладі Raspberry Pi. Мережеве обладнання EServer. URL: https://e-server.com.ua/uk/poradi/shho-take-mikrokompiuteri-rozpovidajemo-na-prikladi-raspberry-pi (дата звернення: 20.10.2025).

12. Покроковий посібник із живлення Raspberry Pi від літієвої батареї. Wonderful PCB. URL: https://www.wonderfulpcb.com/uk/blog/raspberry-pi-battery-power-tutorial-models-setup-guide/ (дата звернення: 21.10.2025).

13. Провід силіконовий 18 AWG Опис. Regis.com.ua. URL: https://regis.com.ua/ua/silicone-wire-18-awg-black?srsltid=AfmBOop3Cc8sbxi8EGVMSfZdHKGiE47h9Uj58Hf0uAEhpjGDRnY0H3tT (дата звернення: 21.10.2025).

14. Raspberry Pi Configuration Documentation. Raspberry Pi. URL: https://www.raspberrypi.com/documentation/computers/configuration.html#configuring-a-user (дата звернення: 22.10.2025).

15. Arduino урок 4 - PWM електроніка та програмування. IT Master. URL: https://itmaster.biz.ua/electronics/arduino/arduino-pwm.html (дата звернення: 22.10.2025).

16. Могильний С. Б. Мікрокомп’ютер Raspberry Pi - інструмент дослідника : Навч. посіб. Київ : Талком, 2014. 340 с.

17. Raspberry Pi. Цифровий датчик інтенсивності світла BH1750FVI I2C. MikroTik KPI. URL: https://mikrotik.kpi.ua/index.php/courses-list/category-raspberry/35-digital-sensor-light-intensity-bh1750fvi-i2c (дата звернення: 24.10.2025).

18. Mitsukova A. Y., Onykienko Y. O. Automated System for Adjusting Lighting Equipment with the Possibility of Further Editing. Electronic and Acoustic Engineering. 2020. Т. 3, № 3. С. 59–63. URL: https://doi.org/10.20535/2617-0965.2020.3.3.199025 (дата звернення: 24.10.2025).

19. Pajankar A., Kakkar A. Raspberry Pi By Example : навч. посіб. Birmingham-Mumbai : PACKT PUBLISHING, 2016. 107 с.

20. The Raspberry Pi GPIO pinout guide. Raspberry Pi GPIO Pinout. URL: https://pinout.xyz/ (дата звернення: 24.10.2025).

21. Berry R. Everything You Need to Know About the Raspberry Pi 4. PiCockpit. URL: https://picockpit.com/raspberry-pi/everything-you-need-to-know-about-the-raspberry-pi-4/ (дата звернення: 25.10.2025).

22. Михайло. Світлодіодна (led) стрічка 5v: особливості і поради вибору. Новини Ужгород. URL: https://news.uzhgorod.ua/blog/technology/svitlodiodna-led-strichka-5v-osoblyvosti-i-porady-vyboru/ (дата звернення: 25.10.2025).

23. Як обрати світлодіодну стрічку? Основні правила вибору світлодіодних стрічок. PROLUM. URL: https://prolum.com.ua/yak-pravylno-vybraty-svitlodiodnu-strichku-dlia-domu/?srsltid=AfmBOooa4muKesmu0dhOFrTLthueUSEtUWjkzFn8zBkTntezaJeOFK_X (дата звернення: 27.10.2025).

24. Таблиця потужності світлодіодної стрічки: Як вибрати відповідний варіант. PROLUM. URL: https://prolum.com.ua/tablytsia-potuzhnosti-svitlodiodnoi-strichky-yak-vybraty-vidpovidnyi-variant/?srsltid=AfmBOooPHpPcXdryqYBp2_XI4vtWiSDVY-2oKDBOUjUiNnaSwtDr6-pW (дата звернення: 27.10.2025).

25. ASTM B258 - 14. Standard Specification for Standard Nominal Diameters and Cross-Sectional Areas of AWG Sizes of Solid Round Wires Used as Electrical Conductors. Чинний від 2015-03-21. Вид. офіц. ASTM INTERNATIONAL, 2015. 10 с.

26. Підключення та маркування світлодіодних стрічок. BRILLE. URL: https://brille.ua/pidkliuchennia-ta-markuvannia-svitlodiodnykh-strichok/ (дата звернення: 28.10.2025).

27. Встановлення та підключення світлодіодної стрічки до блока живлення 12,24 В. OLEDIM. URL: https://oledim.com.ua/vstanovlennia-ta-pidkliuchennia-svitlodiodnoi-strichky-do-bloka-zhyvlennia-1224-v/ (дата звернення: 31.10.2025).

28. Вулична світлодіодна стрічка – ефективність за будь-яких умов. PROLUM. URL: https://prolum.com.ua/vulychna-svitlodiodna-strichka-praktychni-porady-ta-rekomendatsii/ (дата звернення: 01.11.2025).

29. Як вибрати аксесуари для світлодіодних стрічок. Elekomp. URL: https://elekomp.com.ua/blog/kak-vybrat-aksessuary-dlya-svetodiodnyh-lent-rasskazhut-eksperty-elekomp?srsltid=AfmBOort_2MLb7DVQdHx3luTCzb2qebOrLYV88H-G4qiVWcVXGd4qpfa (дата звернення: 02.11.2025).

30. Яку модель Raspberry Pi купити відповідно до потреб і проєкту. El Output. URL: https://uk.eloutput.com/виробляти/домашньої-автоматизації/який-Raspberry-Pi-купити/ (дата звернення: 02.11.2025).

31. Вивчаємо Raspberry Pi. #2. Початок роботи. EVO.net. URL: https://evo.net.ua/izuchaem-raspberry-pi.-chast-2.-nachalo-raboty/?srsltid=AfmBOoqFE_XaXpOxP1YV_mWscqf9CA8S1_fF6TzeLJimyPGCW0RV7dLA (дата звернення: 02.11.2025).

32. Вивчаємо Raspberry Pi. #3. Використання Raspberry Pi. EVO.net. URL: https://evo.net.ua/izuchaem-raspberry-pi.-chast-3.-ispolzovanie-raspberry-pi/?srsltid=AfmBOopxSojZ3_BvTVz-fAH-uU1jUzR5f7Y9YoD2pUI8TRAJyaxBgbgQ (дата звернення: 02.11.2025).

33. Як почати програмувати Raspberry Pi на Python. IT Master електроніка та програмування. URL: https://itmaster.biz.ua/electronics/raspberry-pi/raspberry-python.html (дата звернення: 02.11.2025).

34. Починаємо програмувати на Python для Raspberry Pi | Заняття 1-2. MikroTik KPI. URL: https://mikrotik.kpi.ua/index.php/courses-list/category-python/81-we-begin-to-program-in-python-to-raspberry-pi-session-1-2 (дата звернення: 02.11.2025).

35. Як правильно вибрати карту пам'яті для своїх потреб. Kingston Technology Company. URL: https://www.kingston.com/ua/blog/personal-storage/choose-the-right-memory-card (дата звернення: 02.11.2025).

36. Посібник із типів карт пам'яті SD та microSD. Kingston Technology Company. URL: https://www.kingston.com/ua/blog/personal-storage/microsd-sd-memory-card-guide (дата звернення: 02.11.2025).

37. Які карти пам'яті кращі: розбираємо microSDXC, SD і флеш-карти. Artline. URL: https://artline.ua/uk/blogs/kakie-karty-pamyati-luchshe-razbiraem-microsdxc-sd-i-flesh-karty (дата звернення: 02.11.2025).

38. Як обрати якісний кард-рідер для швидкого перенесення даних. Ugreen. URL: https://ugreen.mk.ua/ua/a512802-kak-vybrat-kachestvennyj.html (дата звернення: 02.11.2025).

39. Flash OS images to SD cards & USB drives. balenaEtcher. URL: https://etcher.balena.io/ (дата звернення: 02.11.2025).

40. Адаптивне освітлення за допомогою інтеграції Circadian Lighting. Smarty Ninja. URL: https://www.smarty.ninja/hard/lights/nalashtovuiemo-adaptyvne-osvitlennia-u-vas-vdoma/ (дата звернення: 02.11.2025).

41. Tomanová A. Як розпочати роботу з розумним освітленням із підтримкою HomeKitu. Letem svetem Applem. URL: https://www.letemsvetemapplem.eu/uk/2024/04/09/jak-zacit-pouzivat-chytre-osvetleni-s-podporou-homekitu/ (дата звернення: 03.11.2025).

42. Розумне освітлення: як заощадити на електриці та створити затишок. SmartDim. URL: https://smartdim.com.ua/rozumne-osvitlennya-yak-zaoshhadyty-na-elektryczi-ta-stvoryty-zatyshok/ (дата звернення: 03.11.2025).

43. Smart Lighting System using Raspberry PI / A. Maslekar та ін. International Journal of Innovative Research in Science, Engineering and Technology. 2015. Т. 4, № 7. С. 1–9.

44. Adaptalight: An Inexpensive PAR Sensor System for Daylight Harvesting in a Micro Indoor Smart Hydroponic System / J. D Stevens та ін. Horticulturae. 2022. Т. 8, № 2. С. 105. URL: https://doi.org/10.3390/horticulturae8020105 (дата звернення: 4.11.2025).

45. Jolles J. W. Broad-scale applications of the Raspberry Pi: A review and guide for biologists. Methods in Ecology and Evolution. 2021. Т. 12, № 9. С. 1562–1579. URL: https://doi.org/10.1111/2041-210x.13652 (дата звернення: 05.11.2025).

46. Upton E., Halfacree G. Raspberry Pi User Guide : навч. посіб. Wiley, 2016.

47. How to add ambient lighting to your TV with Raspberry Pi. Raspberry Pi. URL: https://www.raspberrypi.com/tutorials/raspberry-pi-tv-ambient-lighting/ (дата звернення: 05.11.2025).

48. Jackk Tutorials. Raspberry Pi Home Automation - Control Lights, Computers, CCTV and More!. Instructables. URL: https://www.instructables.com/Raspberry-Pi-Home-Automation-Control-lights-comput/ (дата звернення: 05.11.2025).

49. Design and Application of a Smart Lighting System Based on Distributed Wireless Sensor Networks / Y. Cheng та ін. Applied Sciences. 2020. Т. 10, № 23. С. 8545. URL: https://doi.org/10.3390/app10238545 (дата звернення: 10.11.2025).

50. London D. Build a programmable light display on Raspberry Pi. Opensource.com. URL: https://opensource.com/article/21/1/light-display-raspberry-pi (дата звернення: 11.11.2025).

Додаток А

Повний код скрипта автоматизованої системи адаптивного освітлення на базі мікрокомп’ютера Raspberry Pi 4

Appendix A code.py > ...

```python
import time
import board
import neopixel
import threading
import datetime
import os
from flask import Flask, render_template_string, request, jsonify

# --- КОНФІГУРАЦІЯ АПАРАТНОЇ ЧАСТИНИ ---
PIXEL_PIN = board.D18  # Пін, до якого підключено DIN стрічки
NUM_PIXELS = 60        # Кількість світлодіодів (змініть під свою стрічку)
ORDER = neopixel.GRB   # Порядок кольорів

# Ініціалізація стрічки
pixels = neopixel.NeoPixel(
    PIXEL_PIN, NUM_PIXELS, brightness=1.0, auto_write=False, pixel_order=ORDER
)

# --- ГЛОБАЛЬНІ ЗМІННІ ---
current_state = {
    "mode": "manual",       # 'manual' або 'adaptive'
    "color": "#000000",     # Поточний колір HEX
    "brightness": 50,       # Яскравість 0-100
    "status": "System Ready"
}

app = Flask(__name__)

# --- КЛАС КОНТРОЛЕРА ---
class LightingController:
    def __init__(self):
        self.running = True
        self.lock = threading.Lock()

    def set_color_instant(self, r, g, b):
        """Миттєве встановлення кольору на стрічці"""
        with self.lock:
            # Застосування яскравості
            factor = current_state["brightness"] / 100.0
            r = int(r * factor)
            g = int(g * factor)
            b = int(b * factor)
            pixels.fill((r, g, b))
            pixels.show()

    def adaptive_loop(self):
        """Фоновий процес адаптивного освітлення"""
        while self.running:
```

```
            if current_state["mode"] == "adaptive":
                now = datetime.datetime.now()
                hour = now.hour + now.minute / 60.0

                # Логіка біоритмів
                if 6 <= hour < 10:
                    r, g, b = 255, 200, 150 # Ранок (Теплий)
                    current_state["status"] = "Adaptive: Morning Rise"
                elif 10 <= hour < 18:
                    r, g, b = 255, 255, 255 # День (Холодний)
                    current_state["status"] = "Adaptive: Day Work"
                elif 18 <= hour < 22:
                    r, g, b = 255, 147, 41  # Вечір (Захід сонця)
                    current_state["status"] = "Adaptive: Evening Relax"
                else:
                    r, g, b = 50, 0, 0      # Ніч (Тьмяний червоний)
                    current_state["status"] = "Adaptive: Night Mode"

                self.set_color_instant(r, g, b)
                # Оновлюємо колір у веб-інтерфейсі для відображення
                current_state["color"] = "#{:02x}{:02x}{:02x}".format(r, g, b)

            time.sleep(30) # Перевірка кожні 30 секунд

controller = LightingController()

# --- WEB ІНТЕРФЕЙС (HTML) ---
HTML_PAGE = """
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>RPi Light Control</title>
    <style>
        body { font-family: sans-serif; background-color: #121212; color: #e0e0e0; text-align: center; padding: 20px; }
        .card { background-color: #1e1e1e; border-radius: 15px; padding: 20px; max-width: 400px; margin: 0 auto; box-shadow: 0 4px 10px rgba(0,0,0,0.5); }
        h1 { color: #bb86fc; margin-bottom: 5px; }
        .status { color: #03dac6; font-size: 14px; margin-bottom: 20px; }

        .btn { width: 100%; padding: 15px; margin: 5px 0; border: none; border-radius: 8px; font-size: 16px; cursor: pointer; font-weight: bold; transition: 0.2s; }
        .btn:active { transform: scale(0.98); }
        .btn-manual { background-color: #3700b3; color: white; }
        .btn-auto { background-color: #03dac6; color: black; }

        /* Кнопка вимкнення */
        .btn-off { background-color: #cf6679; color: black; margin-top: 30px; border: 2px solid #b00020; }

        input[type=range] { width: 100%; margin: 15px 0; accent-color: #bb86fc; }
        input[type=color] { width: 100%; height: 50px; border: none; background: none; cursor: pointer; }

        .control-group { margin: 20px 0; text-align: left; }
        label { display: block; margin-bottom: 5px; font-size: 14px; color: #aaa; }
    </style>
</head>
<body>
    <div class="card">
        <h1>Smart Lighting</h1>
        <div class="status" id="statusText">System Ready</div>

        <button class="btn btn-manual" onclick="setMode('manual')">MANUAL MODE</button>
        <button class="btn btn-auto" onclick="setMode('adaptive')">AUTO ADAPTIVE</button>

        <div class="control-group">
            <label>Колір підсвітки</label>
            <input type="color" id="picker" value="#000000" oninput="sendColor()">
        </div>

        <div class="control-group">
            <label>Яскравість</label>
            <input type="range" id="bright" min="0" max="100" value="50" oninput="sendColor()">
        </div>

        <hr style="border-color: #333; margin: 25px 0;">

        <button class="btn btn-off" onclick="shutdownSystem()">⛔ SHUTDOWN SYSTEM</button>
        <p style="font-size: 10px; color: #666;">Натисніть, щоб безпечно вимкнути Raspberry Pi</p>
    </div>

    <script>
        // Оновлення статусу при завантаженні
        updateUI();
        setInterval(updateUI, 5000);

        function setMode(mode) {
            fetch('/api/set_mode', {
                method: 'POST',
                headers: {'Content-Type': 'application/json'},
                body: JSON.stringify({ mode: mode })
            }).then(updateUI);
        }

        function sendColor() {
            let hex = document.getElementById('picker').value;
```

```
            let bright = document.getElementById('bright').value;
            // Відправляємо дані на сервер
            fetch('/api/set_color', {
                method: 'POST',
                headers: {'Content-Type': 'application/json'},
                body: JSON.stringify({ color: hex, brightness: bright })
            });
        }

        function updateUI() {
            fetch('/api/status').then(r => r.json()).then(data => {
                document.getElementById('statusText').innerText = data.status;
            });
        }

        function shutdownSystem() {
            if (confirm("Ви дійсно хочете вимкнути систему? Після цього веб-сайт стане недоступним до перезавантаження живлення.")) {
                fetch('/api/shutdown', {method: 'POST'});
                alert("Система вимикається... Зачекайте 15-20 секунд, поки згасне зелений світлодіод на платі, а потім витягніть шнур живлення.");
                document.body.innerHTML = "<h1 style='color:white; margin-top:50px;'>Система вимкнена.<br>На добраніч! 🌙</h1>";
            }
        }
    </script>
</body>
</html>
"""

# --- FLASK МАРШРУТИ ---
@app.route('/')
def index():
    return render_template_string(HTML_PAGE)

@app.route('/api/set_color', methods=['POST'])
def set_color():
    data = request.json
    current_state["mode"] = "manual"
    current_state["color"] = data['color']
    current_state["brightness"] = int(data['brightness'])
    current_state["status"] = "Manual Control"

    # HEX to RGB
    hex_col = data['color'].lstrip('#')
    r, g, b = tuple(int(hex_col[i:i+2], 16) for i in (0, 2, 4))

    controller.set_color_instant(r, g, b)
    return jsonify(success=True)
```

```
@app.route('/api/set_mode', methods=['POST'])
def set_mode():
    data = request.json
    current_state["mode"] = data['mode']
    if data['mode'] == 'manual':
        current_state["status"] = "Manual Mode Activated"
    else:
        current_state["status"] = "Adaptive Mode Started"
    return jsonify(success=True)

@app.route('/api/status')
def get_status():
    return jsonify(current_state)

# --- МАРШРУТ ДЛЯ ВИМКНЕННЯ ---
@app.route('/api/shutdown', methods=['POST'])
def shutdown():
    # 1. Гасимо світло перед сном
    pixels.fill((0, 0, 0))
    pixels.show()

    # 2. Даємо команду системі на вимкнення
    # (Це спрацює, тільки якщо скрипт запущено через sudo)
    def delayed_shutdown():
        time.sleep(2)
        os.system("shutdown now")

    # Запускаємо в окремому потоці, щоб встигнути відповісти браузеру "OK"
    threading.Thread(target=delayed_shutdown).start()

    return jsonify(success=True)

if __name__ == '__main__':
    # Запуск фонового процесу адаптивного світла
    t = threading.Thread(target=controller.adaptive_loop, daemon=True)
    t.start()

    # Запуск сервера
    app.run(host='0.0.0.0', port=80, debug=False)
```

Додаток Б

Повний код модифікованого скрипта автоматизованої системи адаптивного освітлення на базі мікрокомп'ютера Raspberry Pi 4 під датчик освітленості

Appendix B code.py > LightingController > adaptive_algorithm

```python
import time
import board
import neopixel
import threading
import datetime
import os
import smbus  # Бібліотека для спілкування з датчиком I2C
from flask import Flask, render_template_string, request, jsonify

# --- КОНФІГУРАЦІЯ СИСТЕМИ ---
PIXEL_PIN = board.D18   # Пін стрічки
NUM_PIXELS = 60         # Кількість діодів
ORDER = neopixel.GRB    # Порядок кольорів
I2C_ADDR = 0x23         # Адреса датчика BH1750 (стандартна)

# Ініціалізація стрічки
pixels = neopixel.NeoPixel(
    PIXEL_PIN, NUM_PIXELS, brightness=1.0, auto_write=False, pixel_order=ORDER
)

# Глобальний стан системи
current_state = {
    "mode": "manual",       # 'manual' або 'adaptive'
    "color": "#000000",     # Колір
    "brightness": 50,       # Поточна яскравість (0-100)
    "lux": 0,               # Показник датчика
    "status": "System Ready"
}

app = Flask(__name__)

# --- КЛАС ДЛЯ РОБОТИ З ДАТЧИКОМ ---
class LightSensorBH1750:
    def __init__(self):
        try:
            self.bus = smbus.SMBus(1)  # Використовуємо шину I2C-1
            self.working = True
        except Exception as e:
            print(f"Помилка I2C шини: {e}")
            self.working = False

    def read_light(self):
        """Зчитує рівень освітленості у Люксах"""
        if not self.working:
            return 0
        try:
            # Читаємо 2 байти даних з регістру 0x11 (High Res Mode)
            data = self.bus.read_i2c_block_data(I2C_ADDR, 0x11)
```

```
            # Формула конвертації згідно даташиту BH1750
            lux = (data[1] + (256 * data[0])) / 1.2
            return round(lux, 1)
        except Exception as e:
            print(f"Помилка читання сенсора: {e}")
            return 0

# --- КЛАС КОНТРОЛЕРА ОСВІТЛЕННЯ ---
class LightingController:
    def __init__(self):
        self.running = True
        self.lock = threading.Lock()
        self.sensor = LightSensorBH1750()

    def set_leds(self, r, g, b, brightness_percent):
        """Фізичне встановлення кольору на стрічку"""
        with self.lock:
            # Застосовуємо яскравість до значень RGB
            factor = brightness_percent / 100.0
            r_final = int(r * factor)
            g_final = int(g * factor)
            b_final = int(b * factor)

            pixels.fill((r_final, g_final, b_final))
            pixels.show()

            # Оновлюємо глобальний стан для відображення на сайті
            current_state["brightness"] = brightness_percent

    def adaptive_algorithm(self):
        """Головний цикл: Час + Датчик -> Світло"""
        while self.running:
            # 1. Зчитуємо реальні дані з датчика
            lux_val = self.sensor.read_light()
            current_state["lux"] = lux_val

            # 2. Якщо увімкнено Автоматичний режим
            if current_state["mode"] == "adaptive":
                now = datetime.datetime.now()
                hour = now.hour + now.minute / 60.0

                # ЕТАП А: Вибір колірної температури (за часом)
                if 6 <= hour < 10:
                    base_color = (255, 200, 150) # Ранок: Теплий
                    period_name = "Ранок"
                elif 10 <= hour < 18:
                    base_color = (255, 255, 255) # День: Холодний
                    period_name = "Робочий час"
                elif 18 <= hour < 22:
                    base_color = (255, 147, 41)  # Вечір: Захід сонця
                    period_name = "Вечір"
                else:
                    base_color = (50, 0, 0)      # Ніч: Червоний
                    period_name = "Ніч"

                # ЕТАП Б: Розрахунок яскравості (за датчиком)
                # Логіка: Чим більше світла в кімнаті, тим менше світить стрічка

                target_bright = 50 # Базове значення

                if lux_val > 400:
                    target_bright = 0       # Дуже світло -> Вимкнути
                    msg = "Світла достатньо"
                elif lux_val < 50:
                    target_bright = 100     # Темно -> Максимум
                    msg = "Темно, компенсуємо"
                else:
                    # Плавна формула: від 100% до 0% в діапазоні 50-400 лк
                    target_bright = int(100 - ((lux_val - 50) / 3.5))
                    msg = "Корекція яскравості"

                # Застосування параметрів
                self.set_leds(*base_color, target_bright)

                current_state["status"] = f"Авто: {period_name} | {msg}"
                current_state["color"] = "#{:02x}{:02x}{:02x}".format(*base_color)

            # Затримка циклу (оновлюємо кожні 2 секунди)
            time.sleep(2)

controller = LightingController()

# --- WEB ІНТЕРФЕЙС ---
HTML_PAGE = """
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Smart Home Control</title>
    <style>
        body { font-family: 'Roboto', sans-serif; background-color: #121212; color: #fff; text-align: center; padding: 10px; }
```

```
        .container { background-color: #1f1f1f; border-radius: 20px; padding: 20px; max-width: 400px; margin: 0 auto; box-shadow: 0 10px 20px rgba(0,0,0,0.5); }

        h2 { color: #bb86fc; margin-top: 0; }
        .sensor-box { background: #2c2c2c; padding: 15px; border-radius: 10px; margin: 15px 0; border: 1px solid #333; }
        .lux-val { font-size: 28px; font-weight: bold; color: #ffb74d; }
        .status-text { color: #03dac6; font-size: 14px; margin-bottom: 5px; min-height: 20px;}

        .btn-group { display: flex; gap: 10px; margin: 20px 0; }
        .btn { flex: 1; padding: 15px; border: none; border-radius: 10px; font-size: 14px; font-weight: bold; cursor: pointer; transition: 0.3s; text-transform: uppercase;}
        .btn-man { background: #3f51b5; color: white; }
        .btn-auto { background: #009688; color: white; }
        .btn-off { background: #b71c1c; color: white; width: 100%; margin-top: 20px; }
        .active { border: 2px solid white; box-shadow: 0 0 10px rgba(255,255,255,0.2); }

        .controls { text-align: left; background: #252525; padding: 15px; border-radius: 10px; }
        label { font-size: 12px; color: #aaa; text-transform: uppercase; }
        input[type=range] { width: 100%; margin: 10px 0; accent-color: #bb86fc; }
        input[type=color] { width: 100%; height: 40px; border: none; background: none; cursor: pointer; }
    </style>
</head>
<body>
    <div class="container">
        <h2>Smart Light System</h2>

        <div class="sensor-box">
            <div>Датчик освітленості (BH1750)</div>
            <div class="lux-val"><span id="lux">--</span> lx</div>
            <div class="status-text" id="status">Connecting...</div>
        </div>

        <div class="btn-group">
            <button id="btnM" class="btn btn-man" onclick="setMode('manual')">Ручний</button>
            <button id="btnA" class="btn btn-auto" onclick="setMode('adaptive')">Адаптивний (AI)</button>
        </div>

        <div class="controls" id="manualPanel">
            <label>Колір</label>
            <input type="color" id="picker" value="#000000" oninput="sendData()">

            <label>Яскравість: <span id="brVal">50</span>%</label>
            <input type="range" id="bright" min="0" max="100" value="50" oninput="sendData()">
        </div>

        <button class="btn btn-off" onclick="shutdown()">⛔ Вимкнути систему</button>
    </div>

    <script>
        setInterval(updateState, 1500); // Оновлення кожні 1.5 сек

        function updateState() {
            fetch('/api/status').then(r => r.json()).then(data => {
                document.getElementById('lux').innerText = data.lux;
                document.getElementById('status').innerText = data.status;
                document.getElementById('brVal').innerText = data.brightness;

                // Візуальне перемикання кнопок
                if (data.mode === 'adaptive') {
                    document.getElementById('btnA').classList.add('active');
                    document.getElementById('btnM').classList.remove('active');
                    document.getElementById('manualPanel').style.opacity = '0.5';
                    document.getElementById('manualPanel').style.pointerEvents = 'none';
                } else {
                    document.getElementById('btnM').classList.add('active');
                    document.getElementById('btnA').classList.remove('active');
                    document.getElementById('manualPanel').style.opacity = '1';
                    document.getElementById('manualPanel').style.pointerEvents = 'auto';
                }
            });
        }

        function setMode(mode) {
            fetch('/api/set_mode', {
                method: 'POST',
                headers: {'Content-Type': 'application/json'},
                body: JSON.stringify({ mode: mode })
            }).then(updateState);
        }

        function sendData() {
            let hex = document.getElementById('picker').value;
            let br = document.getElementById('bright').value;
            document.getElementById('brVal').innerText = br;

            fetch('/api/set_color', {
                method: 'POST',
                headers: {'Content-Type': 'application/json'},
                body: JSON.stringify({ color: hex, brightness: br })
            });
        }

        function shutdown() {
            if(confirm("Вимкнути сервер повністю?")) {
                fetch('/api/shutdown', {method: 'POST'});
                alert("Система вимикається...");
            }
        }
    </script>
```

```
</body>
</html>
"""

# --- API ---
@app.route('/')
def home(): return render_template_string(HTML_PAGE)

@app.route('/api/status')
def status(): return jsonify(current_state)

@app.route('/api/set_mode', methods=['POST'])
def set_mode():
    current_state["mode"] = request.json['mode']
    return jsonify(success=True)

@app.route('/api/set_color', methods=['POST'])
def set_color():
    data = request.json
    current_state["mode"] = "manual" # Перехід в ручний при зміні кольору
    current_state["color"] = data['color']
    current_state["status"] = "Ручне керування"

    # HEX -> RGB
    h = data['color'].lstrip('#')
    rgb = tuple(int(h[i:i+2], 16) for i in (0, 2, 4))

    controller.set_leds(*rgb, int(data['brightness']))
    return jsonify(success=True)

@app.route('/api/shutdown', methods=['POST'])
def shutdown():
    pixels.fill((0,0,0)); pixels.show()
    def off():
        time.sleep(1)
        os.system("shutdown now")
    threading.Thread(target=off).start()
    return jsonify(success=True)

if __name__ == '__main__':
    # Запуск фонового процесу адаптивності
    t = threading.Thread(target=controller.adaptive_algorithm, daemon=True)
    t.start()
    app.run(host='0.0.0.0', port=80, debug=False)
```

Додаток В

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

на тему: «Автоматизована система адаптивного освітлення в Smart Home із застосуванням мікрокомп'ютера Raspberry Pi»

Виконав студент групи: КСДМ-61
Ткачук Владислав Олександрович
Керівник кваліфікаційної роботи:
Волохін Віталій Васильович,
доцент кафедри, к.т.н., доцент

Київ – 2025

Мета роботи – мінімізація вартості модуля автоматизованого контролю світлом, за рахунок використання мікрокомп'ютера Raspberry Pi та його модулів.

Об'єкт дослідження – процес автоматизації контролю рівня освітлення.

Предмет дослідження – технології автоматизованого контролю рівня освітлення на базі модулів мікрокомп'ютерів.

2

Актуальність обраної теми

Світовий тренд на енергозбереження та необхідність оптимізації витрат електроенергії в побутовому секторі зумовлює перехід до автоматизованих систем керування, які дозволяють скоротити енергоспоживання освітленням без втрати комфорту.

Використання відкритої платформи Raspberry Pi вирішує проблему вартості комплектуючих, дозволяючи створити доступний, автономний та гнучкий модуль керування. Це забезпечує можливість легкої інтеграції нових функцій та значно знижує вартість масштабування системи порівняно з пропрієтарними аналогами.

Компонентна база проєкту

Для вирішення питання зменшення економічної перемінної при формуванні екосистеми Smart Home, було вирішено використати мікрокомп'ютери та додаткові модулі (див. рис. 5.1). Таке рішення також відкриває можливості подальшого розширення та модифікацій системи в доступному ціновому діапазоні.


(а)


(б)


(в)


(г)


(д)


(е)

Рисунок 4.1 Компонентна база проєкту

В перелік комплектуючих для створення автоматизованої системи освітлення (див. рис. 4.1) увійшли такі компоненти як: *мікрокомп'ютер Raspberry Pi 4* (а), *блок живлення* (б), *перетворювач логічних рівнів* (в), *LED-стрічка* (г), *резистор 470 Ом* (д), *дроти AWG* (е).

Схематика апаратної частини системи

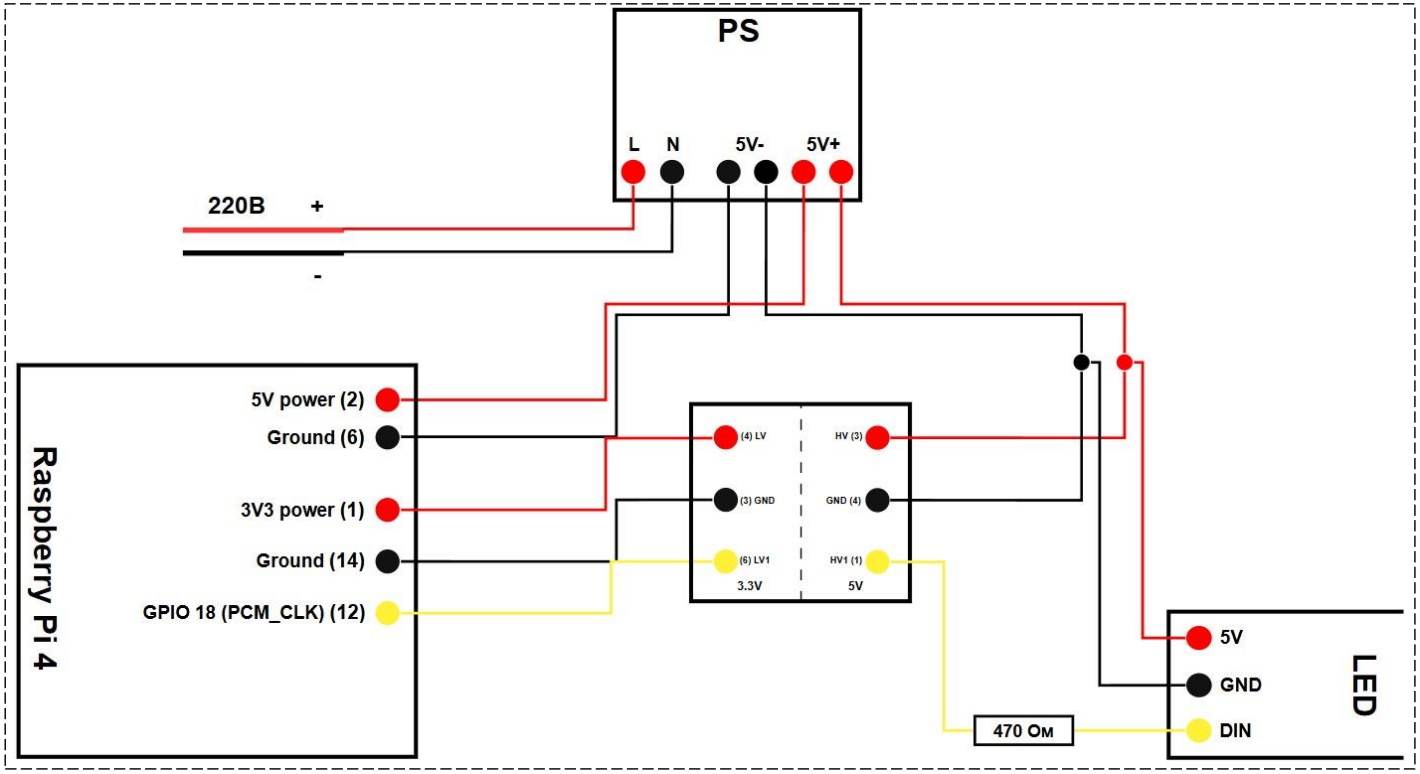


Рисунок 5.1 Схема підключень апаратної частини

Програмна частина системи

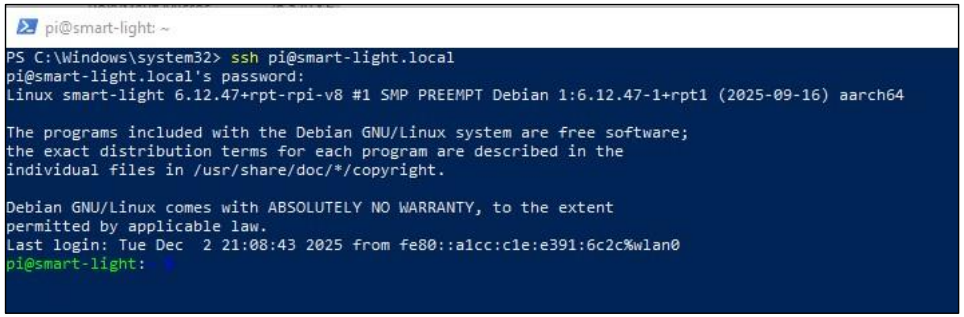


Рисунок 6.1 Підключення до системи через SSH

Програмний код було написано на мові Python для операційної системи на базі Linux, що була попередньо встановлена на мікрокомп’ютері Raspberry Pi.

За рахунок встановлення Lite (полегшеної) версії ОС на мікрокомп’ютер, з метою зменшення задіяних ресурсів, першочергове підключення та налаштування скрипту системи відбувалося через CLI та захищений протокол SSH (див. рис. 6.1).

Керування системою освітлення

Було розроблено скрипт для веб-застосунку на базі мікрофреймворку Flask.

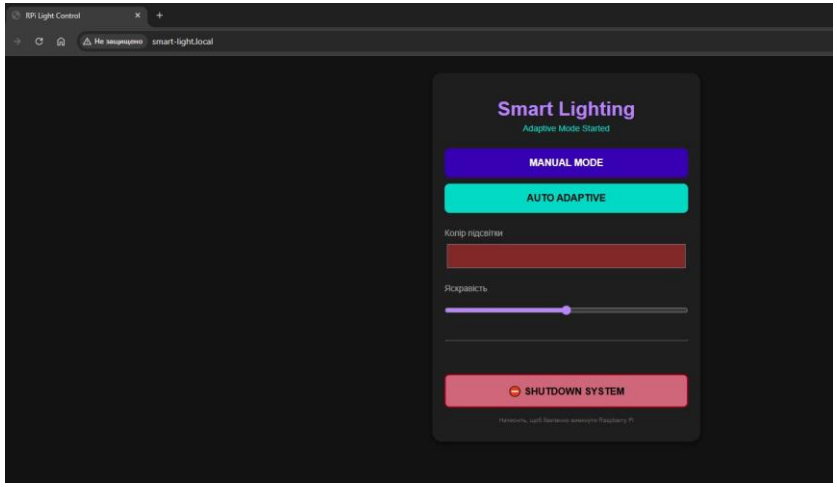


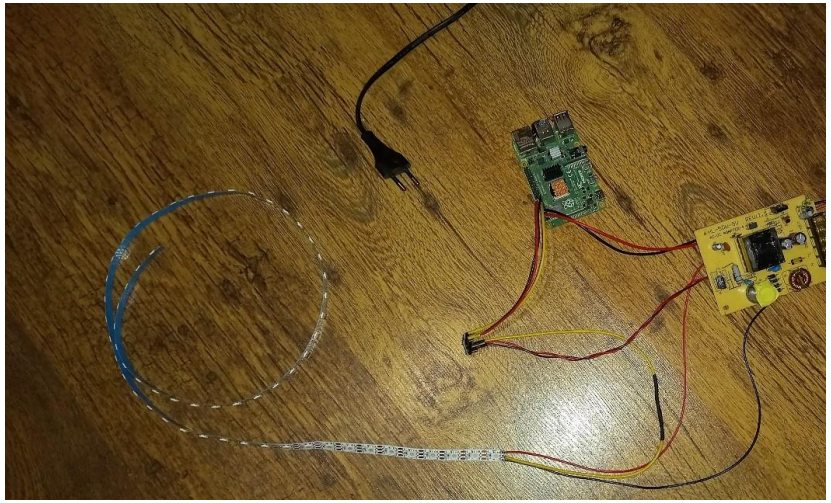
Рисунок 7.1 Інтерфейс веб-застосунку системи

Для керування системою освітлення, потрібно підключитися з будь-якого девайса та браузера до веб-застосунку за адресою: [smart-light.local] (див. рис. 7.1), знаходячись у одній мережі із системою (*це може бути як і ваша домашня мережа, так і спеціальна локальна мережа створена самим мікрокомп'ютером із Wi-Fi модулем*).

7

Втілення проєкту

Після виконання роботи зі збору апаратної та налаштування програмної частин системи, було проведено її монтування та виконано контрольний тестовий запуск (див. рис. 8.1-8.2).



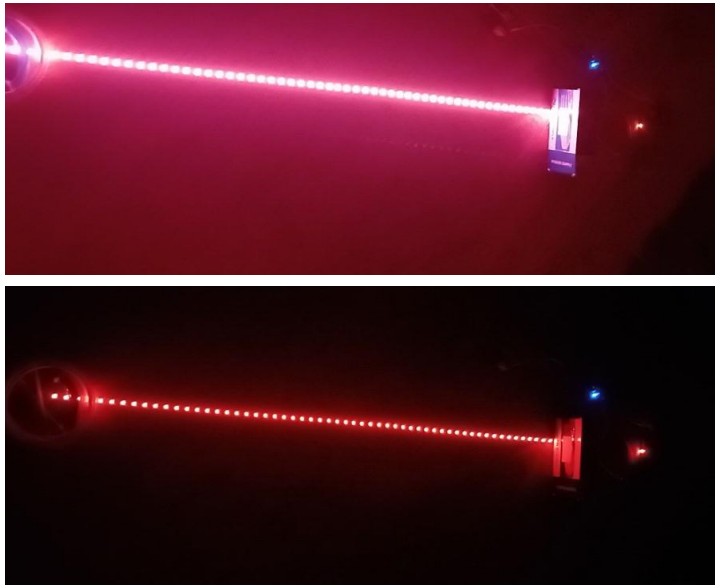


Рисунок 8.1-8.2 Фото зібраної системи та її функціонування

Аналіз економічного компонента

Для обґрунтування доцільності впровадження розробленої системи адаптивного освітлення на базі Raspberry Pi проведено порівняльний аналіз вартості компонентів та масштабування із вартістю аналогічних готових рішень (див. табл. 9.1-2), представлених на ринку. Як об'єкт для порівняння обрано екосистему Philips Hue, яка є лідером у сегменті розумного освітлення.

Таблиця 9.1. Кошторис базового комплекту

Вартість системи освітлення	**грн**
Система на базі Raspberry Pi 4	~ 4200 ₴
Система Philips Hue	~ 6300 ₴

Таблиця 9.2. Кошторис масштабування

Вартість масштабування системи	**грн**
Система на базі Raspberry Pi 4	~ 160 ₴
Система Philips Hue	~ 1400 ₴

Модернізація адаптивності (1)

З метою збільшення відсотка критерію адаптивності системи освітлення та покращення комфорту використання, схему було доопрацьовано (див. рис. 10.1) та оснащено додатковим датчиком рівня освітлення приміщення (див. рис. 10.2).

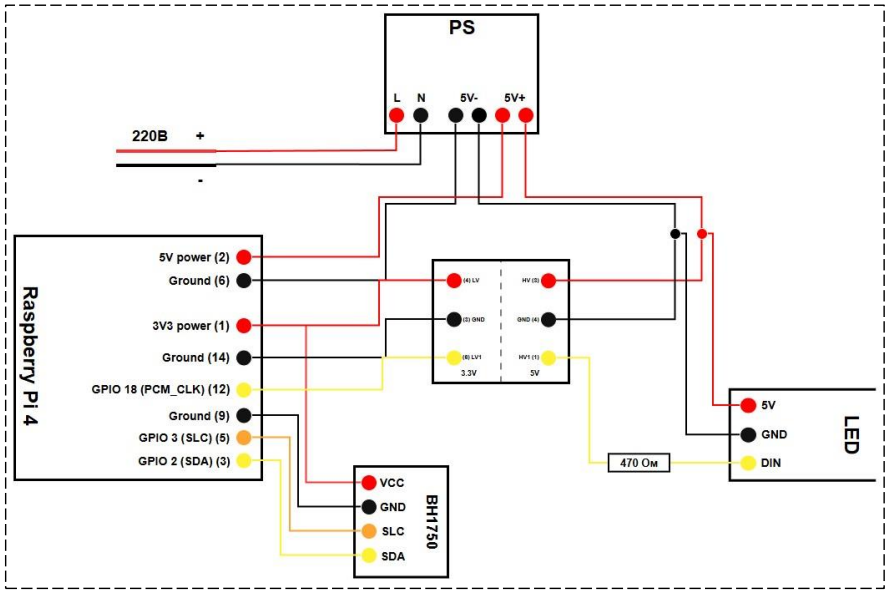


Рисунок 10.1 Доопрацьована схема системи

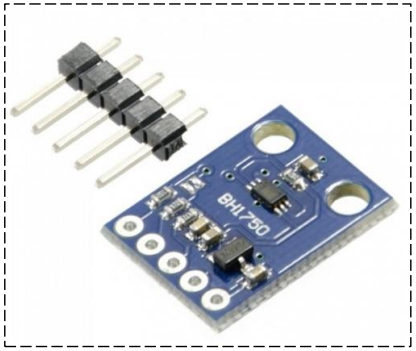


Рисунок 10.2 Датчик рівня освітлення

Модернізація адаптивності (2)

Використання датчика рівня освітлення, дозволяє системі зчитувати з певним інтервалом кількість Люксів, що дорівнює рівню освітленості кімнати в певний момент часу. Використання цих даних дозволяє автоматизованій системі адаптивно змінювати яскравість та колірну гаму освітлення, відштовхуючись від природнього освітлення в приміщенні та часу доби.



Рисунок 11.1 Оновлений інтерфейс системи

- оновлений функціонал та дизайн веб-застосунку системи під датчик освітленості (див. рис. 11.1).

ВИСНОВКИ

У ході роботи обґрунтовано вибір платформи Raspberry Pi 4 Model B як оптимального ядра системи, що дозволило уникнути обмежень закритих пропрієтарних рішень. Практична реалізація підтвердила ефективність модульної архітектури на базі адресної стрічки WS2812B та імпульсного блока живлення із застосуванням прямого керування через інтерфейс GPIO.

Розроблений програмний комплекс мовою Python із використанням фреймворку Flask забезпечує повну автономність системи. Реалізовані алгоритми адаптивності успішно імітують природний добовий цикл, автоматично змінюючи спектр освітлення від теплого до холодного залежно від часу доби, а клієнт-серверна архітектура надає зручний доступ до керування через веб-інтерфейс.

12

ПРОДОВЖЕННЯ ВИСНОВКІВ

Техніко-економічний аналіз довів, що розроблена система на 34% дешевша за функціональний аналог Philips Hue, що дозволяє заощадити понад 2000 грн на стартовому комплекті та значно знизити вартість подальшого масштабування. Система споживає лише від 5 до 10% обчислювальних ресурсів мікрокомп'ютера, залишаючи запас потужності для інтеграції інших функцій «Розумного дому».

Для підвищення енергоефективності та комфорту використання системи, було проведено модернізацію шляхом інтеграції датчика освітленості, що дозволило створити замкнутий контур керування. Це забезпечує автоматичну адаптацію яскравості залежно від рівня природного світла, гарантуючи стабільний рівень освітленості та додаткову економію електроенергії, при вартості модифікації в ~50 грн.

13

Результати впровадження

1. Волохін В. В., Ткачук В. О. АЛЬТЕРНАТИВА ІСНУЮЧИМ ПРОФІЛЬНИМ МОДУЛЯМ КЕРУВАННЯ В SMART HOME: АНАЛІЗ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ МІКРОКОМП'ЮТЕРІВ RASPBERRY PI. «Проблеми комп'ютерної інженерії» : VI науково-практична конференція, м. Київ, 3 грудня 2025 р. С. 1–3. URL: https://duikt.edu.ua/ua/news-1-641-14852-kafedra-kompyuternoi-inzhenerii-zaproshue-do-uchasti-u-vi-naukovo-praktichniy-konferencii-problemi-kompyuternoi-inzhenerii_kafedra-kompyuternoi-inzhenerii (дата звернення: 19.11.2025).

2. Торошанко Я. І., Торошанко О. С., Куфтеріна С. Р., Ткачук В. О. Стаття «МАСШТАБОВАНИЙ БАГАТОСТУПЕНЕВИЙ ШИФРАТОР». Журнал «ТЕЛЕКОМУНІКАЦІЙНІ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ» : номер 1, м. Київ, 2025 р. С. 201–209. URL: https://tit.dut.edu.ua/index.php/telecommunication/article/view/2600 (дата звернення: 17.10.2025).

14