

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій,
обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Луцков Олександр Вікторович
(прізвище, ім'я, по батькові здобувача освіти)

УДК 004.4:621.32

КВАЛІФІКАЦІЙНА РОБОТА

Система адаптивного управління освітленням на базі технологій ІОТ
(тема роботи)

126 «Інформаційні системи та технології»
(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

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

(підпис, ініціали та прізвище здобувача освіти)

Керівник роботи

(прізвище, ім'я, по батькові)

(науковий ступінь, вчене звання)

Житомир - 2025

Висновок кафедри _____
за результатами попереднього захисту: _____

Протокол засідання кафедри _____
№ _____ від “_____” _____ 20 _____ р.

Завідувач кафедри _____

(науковий ступінь, вчене звання)

(підпис)

(прізвище, ім'я, по батькові)

“_____” _____ 20 _____ р.

Результати захисту кваліфікаційної роботи

Здобувач освіти _____ захистив
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

(науковий ступінь, вчене звання)

(підпис)

(прізвище, ім'я, по батькові)

АНОТАЦІЯ

Луцков О. В. Система адаптивного управління освітленням на базі технологій IoT - Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра за спеціальністю 126 - Інформаційні системи та технології. - Поліський національний університет, Житомир, 2025.

У кваліфікаційній роботі було розглянуто питання підвищення ефективності роботи освітлення шляхом впровадження алгоритмів систем адаптивного управління освітленням. Було проведено аналіз існуючих рішень адаптивного керування освітленням, визначено основні функціональні вимоги до розробленої прошивки, що забезпечуватиме роботу системи, вибрано основні інструменти та технології, необхідні для її розробки. Серйозну увагу було надано впровадженню алгоритму ПІД-регуляції, який використовується системою для плавного та ефективного керування рівнем освітлення ламп. Практичне значення виконаної роботи полягає у тому, що запропоновану систему можна впровадити у житлових приміщеннях, що дозволить покращити та спростити взаємодію із освітленням та за рахунок алгоритмів керування, впроваджених у прошивці, зменшити споживання електроенергії.

Ключові слова: адаптивне освітлення, ПІД-регуляція, керування освітленням, Zigbee, IOT, ESP 32.

SUMMARY

Lutskov O. V. Adaptive lighting control system based on IoT. - Qualification work on the rights of the manuscript.

Qualification work for a bachelor's degree in specialty 126 - Information systems and technologies - Polissya National University, Zhytomyr, 2025.

The qualification work addressed the issue of improving the efficiency of lighting by implementing algorithms for adaptive lighting control systems. An analysis of existing solutions for adaptive lighting control was carried out, the main functional requirements for the developed firmware that will ensure the operation of the system were determined, and the main tools and technologies necessary for its development

were selected. Serious attention was paid to the implementation of the PID control algorithm, which is used by the system to smoothly and efficiently control the level of illumination of the lamps. The practical significance of this work is that the proposed system can be implemented in residential premises, which will improve and simplify interaction with lighting and, due to the control algorithms implemented in the firmware, reduce energy consumption.

Keywords: adaptive lighting, PID control, lighting control, Zigbee, IOT, ESP 32.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ.....	8
1.1 Аналіз предметної області дослідження.....	8
1.2 Визначення функціональних вимог до системи адаптивного управління освітленням на базі IoT.....	12
Висновки до першого розділу.....	15
РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ АДАПТИВНОГО УПРАВЛІННЯ ОСВІТЛЕННЯМ...16	
2.1 Системна архітектура та вибір апаратного забезпечення.....	16
2.2 Основні алгоритми функціонування системи адаптивного управління освітленням.....	18
Висновки до другого розділу.....	20
РОЗДІЛ 3 ОПИС РОЗРОБКИ.....	22
3.1 Реалізація прошивки системи адаптивного управління освітленням.....	22
3.2 Конфігурація інфраструктури системи.....	23
3.3 Тестування системи.....	26
Висновки до третього розділу.....	27
ВИСНОВКИ.....	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	31
ДОДАТКИ.....	34

ВСТУП

У сучасному світі, де населення стрімко зростає, особливої важливості набувають питання сталого розвитку, зокрема доступ до дешевої відновлюваної енергії, розвиток інфраструктури, боротьба зі змінами клімату. Досягнення цих цілей значною мірою залежить від впровадження енергоефективних технологій, зокрема адаптивного освітлення на базі інтернету речей (IoT).

Сучасні вимоги до енергоощадності, комфорту та безпеки зумовлюють потребу в інтелектуальних системах, здатних регулювати освітлення в реальному часі, аналізуючи дані з сенсорів (освітленості, руху, присутності тощо). Такі рішення знижують споживання електроенергії та викиди CO₂, підвищують зручність, безпеку й якість життя.

Крім того, ці системи дозволяють накопичувати дані про свою роботу, що відкриває можливості для оптимізації керування, прогнозування енергоспоживання та інтеграції зі смарт-інфраструктурою міст.

Таким чином, адаптивне освітлення на базі IoT є ключовим компонентом як глобальної стратегії сталого розвитку, так і сучасного енергоефективного управління в побуті та бізнесі.

Метою дипломної роботи є розробка прошивки для мікроконтролера яка забезпечуватиме роботу системи адаптивного управління освітленням, що дозволить користувачам керувати освітленням у приміщеннях та моніторити споживання електроенергії.

Об'єктом дослідження є процес управління штучним освітленням у приміщеннях.

Предметом дослідження є методи та алгоритми адаптивного регулювання освітлення на основі даних з датчиків освітленості та руху.

Методи дослідження:

За результатами виконання роботи було підготовлено публікації:

1. Луцков О. В. Огляд технологій для побудови рішення адаптивного управління освітлення // Збірник_БТІ_НГ_2024. - с. 62-64

2. Луцков О. В. Огляд протоколу Zigbee для систем адаптивного управління освітленням будинку на базі ІОТ // Збірник_ІттаМС_2025. с. 24-26

Практичне значення отриманих результатів: Розроблена система адаптивного управління освітленням може бути впроваджена в житлових приміщеннях для зменшення енергоспоживання, підвищення комфорту користувачів та зниження експлуатаційних витрат.

Структура роботи: робота складається з анотації, вступу, трьох розділи основної частини, висновків, списоку використаних джерел та додатків. У першому розділі розглянуто теоретичні основи адаптивного управління освітленням, проведено аналіз сучасних дротових і бездротових систем, визначено функціональні вимоги та побудовано UML-діаграми. Другий розділ присвячено проектуванню архітектури системи, вибору апаратного та програмного забезпечення, опису реалізації основних модулів і алгоритмів роботи. У третьому розділі наведено результати тестування, аналіз ефективності роботи системи та рекомендації щодо впровадження. Обсяг роботи становить 42 сторінки. Робота містить 2 рисунки, одну формулу, 8 додатків та 36 літературних джерел.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області дослідження

У сучасних умовах розвитку людства, кількість якого невпинно зростає із кожним роком, актуальними стали питання сталого розвитку людства. Це відображено у 17-ти цілях сталого розвитку, затверджених ООН. Серед них важливе місце займають: боротьба зі зміною клімату (ціль №13), сталий розвиток міст (ціль №11), інноваційна інфраструктура (ціль №9) та забезпечення доступною та дешевою електроенергією (ціль №7).

У контексті цих цілей сталого розвитку важливу роль відіграють системи адаптивного управління освітленням на базі IoT, які дозволяють ефективно використовувати електроенергію, знижувати навантаження на енергосистему, зменшувати викиди CO₂, підвищувати комфорт використання простору, підвищуючи якість життя населення.

У сучасних умовах зростаючого попиту на енергоефективні рішення, системи освітлення стають важливою складовою концепції «розумного дому». Рациональне використання електроенергії, оптимізація роботи освітлювальних пристроїв та підвищення комфорту проживання є ключовими завданнями таких систем.

Адаптивні системи управління освітленням дозволяють у режимі реального часу автоматично регулювати інтенсивність освітлення залежно від умов зовнішнього середовища, потреб користувачів та заданих параметрів роботи системи. Впровадження таких систем спрямоване на зменшення енергоспоживання, що позитивно впливає на екологію та знижує витрати користувачем електроенергії.

Ефективність роботи систем адаптивного управління освітленням досягається за допомоги використання різноманітних датчиків (освітленості, руху тощо), контролерів, що в залежності від отриманої інформації та попередньо заданих параметрів, можуть ефективно регулювати рівень освітленості. Основу

системи складають світлодіодні лампи, які завдяки підтримці технології димінгу, зміни температури світла та сценаріїв роботи, значно переважають традиційні джерела світла за енергоефективністю.

У ході аналізу існуючих рішень було визначено, що однією із основних характеристик систем адаптивного управління освітленням можна визначити спосіб підключення: дротовий та бездротовий. Він залежить від протоколу, який використовується для створення мережі, по якій будуть передаватися дані для контролю системи і є одним із ключових факторів при виборі яку систему встановлювати.

Із основних переваг дротового підключення для систем адаптивного управління освітленням можна виділити надійність з'єднання та відстань передачі керуючого сигналу. Основними недоліками ж є сама необхідність у прокладанні кабелів для передачі даних, необхідність його екранування якщо він прокладається біля кабелів передачі струму для перешкоджання появи електромагнітних завад. Також проблемою може бути складність доступу до кабелю при виникненні можливих несправностей.

Перевагами бездротових протоколів є простота встановлення із-за відсутності необхідності у прокладанні додаткових кабелів. Проте із недоліків можна зазначити відстань передачі сигналу, що може значно знижуватись із-за перешкод та електромагнітних завад від інших сигналів, та менша надійність з'єднання у порівнянні із провідним підключенням.

Відповідно до вище описаної інформації для порівняння було обрано наступні системи керування освітленням: серед провідних - PUSH (DIM), TRIAC (triode for alternating current), DALI (Digital addressable lightning interface), DMX-512, KNX; серед безпроводних - Philips HUE, Xiaomi Smart Home, Enlighted by Siemens та Casambi.

PUSH (DIM) - це проста аналогова система димінгу, за допомогою якої можна змінювати яскравість джерела світла за допомогою звичайного довгого або короткого натискання кнопки. Сигнал подається безпосередньо через фазний кабель до драйвера. Це зручно у житлових приміщеннях або невеликих об'єктах,

де не потрібна складна автоматика. Перевагою такого підходу є простота реалізації, низька вартість а надійність. Проте система не підтримує віддалене керування та обмежена у масштабуванні.

TRIAC (Triode for Alternating Current) - це електронний компонент, який використовується для керування змінним струмом, зокрема в системах димінгу у лампах розжарювання та деяких LED-лампах. Діє такий механізм керування за допомогою фазової модуляції напруги. TRIAC є доступним рішенням, сумісним із традиційними лампами. Обмеженнями виступає те, що цей механізм не сумісний із усіма світлодіодними лампами а також можливе спотворення сигналу, щ може призводити до мерехтіння світла.

DALI (Digital addressable lightning interface) - це цифровий протокол керування освітленням, який дозволяє точно налаштовувати кожен світильник у мережі. Система підтримує двосторонню передачу даних, що дає змогу не лише керувати освітленням, а й отримувати зворотний зв'язок про стан пристроїв. Застосовується у комерційних, офісних та промислових об'єктах. DALI вирізняється масштабованістю, точністю та високою функціональністю. Серед недоліків - потреба у додатковому кабелюванні та вищі початкові витрати.

DMX-512 - це цифровий протокол, що спочатку був розроблений для театрального і сценічного освітлення. Він забезпечує високошвидкісне керування великою кількістю освітлювальних пристроїв у реальному часі. У сучасних системах також використовується в архітектурному освітленні та великих комерційних проєктах. Основна перевага - точність і синхронізація. Проте система має обмежену кількість адрес і складна в налаштуванні для не спеціалізованих приміщень.

KNX - це універсальний протокол автоматизації будівель, який включає в себе керування освітленням, опаленням, вентиляцією тощо. Система працює через різні носії - провід, бездротову передачу, інфрачервоні або радіосигнали. KNX широко застосовується у великих розумних будівлях та житлових комплексах. До її переваг належать масштабованість, стабільність і підтримка широкого спектру

пристроїв. Серед недоліків можна зазначити високу складність конфігурації та вартість реалізації.

Philips Hue - популярна бездротова система освітлення для «розумного дому», яка використовує протокол Zigbee та підтримує керування через мобільний додаток або за допомогою голосових помічників. Система дозволяє змінювати колір, яскравість і створювати сценарії освітлення. Основною сферою застосування є житлові приміщення. Сильними сторонами цього рішення є зручність, інтеграція з іншими IoT-пристроями та просте встановлення. До мінусів можна віднести високу ціну та залежність від хабу.

Xiaomi Smart Home - це екосистема для автоматизації дому, до якої входить і система освітлення. Вона підтримує Wi-Fi, Zigbee та Bluetooth, що дозволяє з'єднувати різні пристрої у мережу та керувати ними через єдиний додаток. Система орієнтована на побутове використання і забезпечує дистанційне керування, автоматизацію сценаріїв та інтеграцію з голосовими асистентами. Основними перевагами є доступна ціна, широка сумісність і гнучкість. Недоліки - залежність від сервісів Xiaomi та іноді нестабільна робота з деякими сторонніми пристроями.

Enlighted by Siemens - це інтелектуальна система керування освітленням, орієнтована на комерційні та офісні простори. Вона використовує мережу бездротових датчиків і штучний інтелект для адаптації освітлення до присутності людей, природного світла та часу доби. Enlighted забезпечує детальну аналітику використання приміщень, енергозбереження та підвищення комфорту. Головною перевагою є масштабованість та аналітичні можливості. Недоліком є висока вартість впровадження.

Casambi - бездротова система керування освітленням на базі Bluetooth Low Energy, яка не потребує централізованого хаба. Всі пристрої утворюють mesh-мережу, що забезпечує стабільність і гнучкість у побудові. Сфера застосування - від офісів до музеїв і приватних будинків. Серед переваг це простота встановлення, висока адаптивність і підтримка різних виробників. Недоліками є обмеження дальності роботи Bluetooth у великих просторах і складність інтеграції

зі сторонніми системами. Порівняльну таблицю для вище наведених систем можна знайти у додатку А.1

На основі вище описаних систем можна дійти до висновку, що основними інформаційними потоками є дані від датчиків, налаштування користувача, керуючі команди та дані зворотного зв'язку (діагностичні дані, підтвердження виконання тощо). Структурно процес обробки інформаційних потоків можна поділити на такі етапи: збір даних, обробка сигналу, передача команд та зворотній зв'язок.

На першому етапі системи отримують команди або налаштування від користувача через пульт керування, додаток або перемикач. Також вони можуть враховувати показники із датчиків (світла, присутності тощо), пору року, температуру тощо.

Далі системою відбувається обробка сигналу, де контролер або шлюз аналізує отримані дані та робить наступне: визначає необхідність зміни освітлення та/або застосовує запрограмовані сценарії чи адаптивні алгоритми.

Наступним кроком відбувається передача команд: контролер надсилає керуючий сигнал або команду на відповідні освітлювальні пристрої (лампи, димери, драйвери) через відповідний провідний чи безпроводний інтерфейс.

Останнім кроком може бути зворотній зв'язок, де світильники чи модулі керування надсилають підтвердження виконання або діагностичні дані.

1.2 Визначення функціональних вимог до системи адаптивного управління освітленням на базі ІоТ.

У результаті аналізу, що було виконано у попередньому розділі було визначено, що система адаптивного управління освітленням має задовольняти наступні функціональні вимоги:

- Отримання даних із сенсорів - система має збирати дані з різноманітних датчиків в режимі реального часу аби мати можливість виконувати керування освітленням;
- Адаптивне керування освітленням - система повинна вміти автоматично змінювати яскравість та режим роботи освітлювальних пристроїв в

залежності від аналізу даних та попередньо заданих налаштувань користувача;

- Віддалений доступ - користувач повинен мати змогу віддалено керувати освітленням через інтернет;
- Можливість ручного керування системою - користувач повинен мати змогу вручну змінювати налаштування системи та освітлення через мобільний застосунок чи WEB-інтерфейс;
- Налаштування сценаріїв роботи освітлення - система повинна підтримувати можливість створення сценаріїв роботи, наприклад: “нічний режим”, “енергозбереження” тощо;
- Логування та зберігання історії роботи системи - вона повинна вести журнал подій, зберігати історію подій, спрацювань, налаштувань, вмикання/вимикання світла тощо;
- Інтеграцію з іншими існуючими системами - система повинна мати змогу простої інтеграції з уже існуючими пристроями і сервісами;
- Підтримку повідомлень та сповіщень про роботу системи - вона повинна вміти надсилати сповіщення, наприклад повідомлення про несправності у роботі системи, повідомлення про перевищення енергоспоживання тощо;
- Захист даних користувача - система повинна гарантувати безпечну передачу та зберігання даних, їх шифрування та захист від несанкціонованого доступу;

На основі наведеної вище інформації можна визначити, що у системі адаптивного управління освітленням присутні два головних актори: користувач і сама система. Кожен із них має доступ до певних варіантів використання системи. Користувач може налаштовувати сценарії освітлення, переглядати історію подій, виконувати налаштування системи, отримувати сповіщення, керувати освітленням вручну та інтегрувати систему управління освітленням із іншими. Система ж може збирати дані із датчиків, аналізувати дані та приймати рішення щодо роботи системи на основі чого здійснюється керування освітленням та логувати події. UML-діаграма прецедентів наведена у додатку А.2.

Із вище викладеної інформації одразу можна виділити декілька основних елементів у системі, які будуть взаємодіяти між собою, а саме: користувач, сенсори (датчики), контролер, IoT шлюз, прилади освітлення та система журналювання (журнал подій). Загалом система адаптивного управління освітлення має два глобальних сценарії роботи, де користувач обирає те як вона працюватиме вручну або ж система автоматично регулює роботу освітлення на основі попередніх налаштувань та даних із датчиків.

У першому випадку використання користувач через мобільний застосунок або веб-інтерфейс обирає режим освітлення. Ця команда передається до шлюзу, який передає її далі до контролера, який після її обробки, посилає команду на пристрої освітлення. Останні передають статус виконання команди (успішний або ні) на контролер який у свою чергу передає його на шлюз, який надсилає повідомлення про виконання команди та виконує запис події у журнал.

У другому сценарії використання системи вона періодично отримує дані із сенсорів про поточний стан освітлення у приміщеннях на контролер, який звіряє дані із налаштуваннями. У випадку, якщо освітлення не достатнє відбувається передача команди на освітлювальні пристрої, які повертають статус виконання, який передається хабу, що записує це у журнал подій. Схожим чином система діє і у випадку недостатнього освітлення. Може бути таке, що рівень освітлення у нормі, тоді система записує у логи, що зміни не були потрібні.

Також система адаптивного управління освітленням постійно збирає та записує дані із датчиків та про свій власний стан. Це корисно для виявлення помилок у її роботі та надає можливість оптимізації житлового приміщення, наприклад, якщо у світлу пору доби у якомусь із приміщень дуже яскраво це може свідчити про велику кількість сонячного світла у ньому, що може призводити до не бажаного збільшення температури повітря у приміщенні.

UML-діаграму послідовності вище описаного процесу можна знайти у додатку А.3.

Також було зроблено діаграму класів, яку можна знайти у додатку А.4. Вона відображає основні структурні компоненти системи та взаємозв'язки між ними,

моделюючи ролі користувача, шлюзу (Gateway), контролера, Zigbee-модуля, сенсорів, освітлювальних пристроїв та журналу подій. Користувач ініціює налаштування або керування, яке передається до контролера через шлюз. Контролер, у свою чергу, здійснює аналіз отриманих від сенсорів даних, приймає рішення щодо регулювання освітлення та надсилає відповідні команди через Zigbee-модуль до виконавчих пристроїв. Відповіді про статус виконання команд передаються зворотним каналом до шлюзу, де відбувається логування подій у відповідному журналі. Така структура забезпечує підтримку всіх ключових функціональних вимог до системи, включаючи автоматизацію, ручне керування, сценарії, логування, безпеку та інтеграцію з іншими системами.

Висновки до першого розділу

У першому розділі було проведено аналіз предметної області дослідження. Було проаналізовано уже існуючі рішення, визначено їх основні переваги і недоліки, що допомогло визначити основні критерії важливі при проектуванні та розробці системи адаптивного управління освітленням.

Також було визначено основні функціональні вимоги до системи адаптивного управління освітленням, побудовано UML-діаграми прецедентів, послідовностей і класів, які дозволять спростити розуміння та розробку системи адаптивного управління освітленням.

Було дійдено до висновків, що розроблювана нами система буде працювати на основі безпроводного протоколу зв'язку із-за відсутності у необхідності прокладання дротів для передачі даних, що полегшує встановлення системи та прибирає додаткові витрати пов'язані із цим процесом.

РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ АДАПТИВНОГО УПРАВЛІННЯ ОСВІТЛЕННЯМ

2.1 Системна архітектура та вибір апаратного забезпечення

Для ефективного керування освітленням у приміщенні система працює за принципом зонування освітлення. Це дозволяє ефективно адаптувати інтенсивність освітлення в різних частинах приміщення залежно від реальних умов - таких як наявність людей або рівень природного освітлення. Кожна зона функціонує як незалежна логічна одиниця, яка включає щонайменше один сенсор освітленості, один сенсор присутності та один або декілька освітлювальних приладів. Такий підхід дозволяє зменшити енергоспоживання, підвищити комфорт користувача та забезпечити гнучке масштабування системи при розширенні кількості зон або приміщень.

Ключовим технологічним компонентом для реалізації запропонованої системи є протокол зв'язку Zigbee, який визначає топологію та принципи побудови системи. Zigbee - це енергоефективний бездротовий протокол, що працює у мережах типу "точка-точка", "зірка" або "mesh", дозволяючи легко масштабувати систему. Саме ця особливість протоколу дозволяє логічно розділити приміщення на зони освітлення, де до кожної зони можуть належати декілька сенсорів присутності та освітленості, а також один або більше світильників. Контролер взаємодіє з кожним із цих пристроїв через Zigbee-модуль, що спрощує фізичне з'єднання та підвищує гнучкість системи загалом.

У центрі системи знаходиться контролер (координатор), що виконує основну обчислювальну логіку. Саме на ньому розгортається прошивка, яка відповідає за аналіз показників з сенсорів, прийняття рішень, розрахунок рівня яскравості та надсилання команд на освітлювальні прилади. Це дозволяє забезпечити стабільну взаємодію з брокером повідомлень та системою автоматизації.

Для збирання даних про навколишнє середовище використовуються два типи сенсорів: цифрові сенсори освітленості, що дозволяють вимірювати рівень

люксів у приміщенні (наприклад сенсори на базі TSL2561); сенсори присутності, який реєструє появу людей у зоні дії (наприклад датчик на базі HiLink LD2410).

Ці датчики підключаються до Zigbee-модуля (на базі CC2530 або аналогів) і передають дані контролеру у вигляді повідомлень. Такі пристрої були змодельовані як логічні компоненти, що імітують реальну поведінку сенсорів у середовищі розробки.

Вся взаємодія між координатором і системою автоматизації OpenHAB здійснюється через протокол MQTT. Контролер публікує повідомлення про стан зон та освітлення, а також підписується на команди зміни режимів. MQTT-брокер Mosquitto, що працює у середовищі Docker, виконує роль посередника у передачі даних між компонентами системи.

OpenHAB, своєю чергою, виконує функції інтерфейсу користувача: дозволяє налаштовувати режими освітлення (низький, середній, високий, автоматичний), відстежувати події, логувати активність системи. Дані передаються між OpenHAB і контролером у вигляді структурованих MQTT-повідомлень, що дає змогу реалізувати автоматичне виявлення пристроїв (авто-дискавері) та швидке налаштування інтеграції.

Рішення про ввімкнення освітлення приймається контролером на основі двох факторів: рівня освітлення у зоні та наявності присутності. Для розрахунку необхідного рівня яскравості використовується ПД-регулятор, налаштований за методом Зіглера-Нікольса.

Система визначає цільовий рівень освітленості для приміщення у 300 люкс згідно з ДБН В.2.5-28:2018. Контролер аналізує дані із датчиків освітленості про його поточний рівень відносно цільового значення, категоризуючи його як дуже низький (<20%), низький (20-50%), середній (50-80%), високий (80-120%) або дуже високий (>120%). При фіксації присутності людини та активного руху система застосовує відповідні правила її роботи для визначення цільового значення оптимального рівня штучного освітлення (0-100% від максимальної потужності ламп) з урахуванням часу доби, після чого надсилає цей рівень до ПД-регулятора, який займається керуванням для регулювання яскравості ламп.

На етапі розробки архітектура системи реалізується із застосуванням віртуалізованого середовища. Основні компоненти - MQTT-брокер Mosquitto та система автоматизації OpenHAB - розгортаються у вигляді окремих Docker-контейнерів, що забезпечує ізолюваність, легкість конфігурації та зручність масштабування. Прошивка контролера (ESP32) виконується у середовищі QEMU для емуляції апаратної платформи, що дозволяє розробляти, налагоджувати та тестувати систему без необхідності фізичного доступу до обладнання. У фінальній версії проєкту передбачається розгортання прошивки безпосередньо на апаратному контролері, тоді як вся інфраструктура зв'язку залишається сумісною із реальною мережею IoT-пристроїв.

На додатку Б.1 зображено функціональну схему взаємодії основних компонентів системи: від сенсорів та керування лампами - до OpenHAB і зворотного контролю станів.

У якості мінімальних апаратних вимог для повноцінної реалізації проєкту рекомендується використовувати контролер на базі ESP32 з Ethernet або Wi-Fi інтерфейсом, Zigbee-модуль CC2530 або CC2652, а також цифрові сенсори (освітленості та присутності). Серверна частина (де розміщено Docker і OpenHAB) може працювати на пристрої типу Raspberry Pi 4 або повноцінному ПК, обладнаному процесором з двома ядрами, 4-8 ГБ оперативної пам'яті та SSD-диском для стабільної роботи MQTT та контейнерів OpenHAB.

2.2 Основні алгоритми функціонування системи адаптивного управління освітленням

Логіку роботи системи адаптивного управління освітленням умовно можна поділити на кілька етапів: ініціалізація режиму роботи, періодичне опитування сенсорів, аналіз отриманих даних, прийняття рішення про керування освітленням у зонах та відправка результату до системи OpenHAB через MQTT. Основна відповідальність за алгоритмічну обробку покладається на контролер, який, з урахуванням вибраного режиму та сенсорних даних, динамічно регулює інтенсивність освітлення.

Уся взаємодія між компонентами системи - користувачем, OpenHAB, MQTT-брокером, контролером, сенсорами, Zigbee-модулем та освітлювальними приладами - ілюструється на UML-діаграмі послідовності (рисунок 2.1).

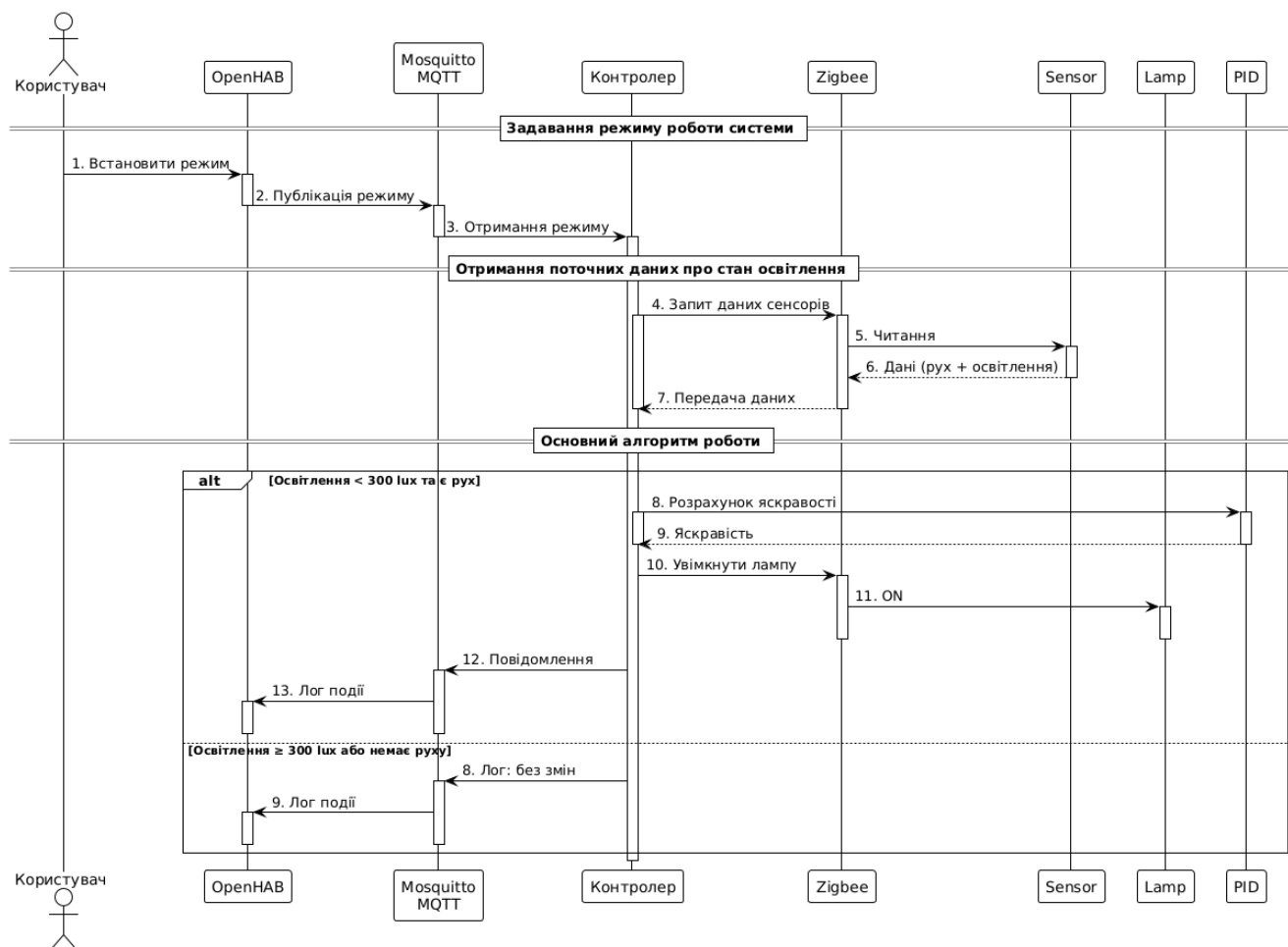


Рисунок 2.1 - UML-діаграма послідовності роботи системи адаптивного управління освітленням

Основна логіка функціонування контролера реалізується за допомогою циклічного опитування логічних зон, у яких розміщено сенсори руху та освітлення. Для кожної зони обчислюється середній рівень освітленості, після чого на основі заданого режиму та сенсорної активності контролер приймає рішення про необхідність увімкнення освітлення.

Ця логіка відображена у діаграмі алгоритму роботи контролера відображена у додатку Б.2.

Контролер надсилає повідомлення до OpenHAB лише у разі зміни стану (наприклад, увімкнення лампи або зміна режиму користувачем), що дозволяє знизити мережеве навантаження та зменшити кількість зайвих логів.

Також контролер системи адаптивного управління освітленням реалізує механізм автоматичного регулювання рівня освітлення на основі ПІД-регулятора - класичного інструменту керування, що дозволяє системі гнучко реагувати на зміни зовнішніх умов. Його використання забезпечує не лише плавність переходів між рівнями яскравості, а й стабільність освітлення у зоні при зміні інтенсивності природного світла або при появі людей у приміщенні.

Контролер аналізує поточне значення освітленості, порівнює його з цільовим і на основі цієї різниці (похибки) формує керувальний вплив - рівень яскравості, який потрібно подати на освітлювальний прилад. У процесі розрахунку враховується не тільки поточна похибка, але й швидкість її зміни та накопичене відхилення у часі. Це дозволяє забезпечити адаптивну поведінку системи - з одного боку, вона швидко реагує на зміни, а з іншого - не допускає коливань або перерегулювання, що характерні для систем із жорсткою логікою вмикання/вимикання.

Основою алгоритму є диференціальне рівняння ПІД-регулятора, яке описується наступною формулою:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de(t)}{dt}$$

де $u(t)$ - керувальний вплив, тобто яскравість лампи в момент часу t , а $e(t)$ - похибка, що визначається як різниця між бажаним рівнем освітлення в зоні (заданим системою) та фактичним значенням, отриманим від сенсора освітленості. Коефіцієнти K_p , K_i та K_d відповідають за пропорційний, інтегральний і диференціальний впливи відповідно та підбираються емпірично відповідно до динаміки кожної конкретної зони.

У межах даної системи налаштування параметрів регулятора (K_p , K_i , K_d) здійснюється за методом Зіглера-Нікольса, що дозволяє забезпечити узгодженість між швидкістю реакції та стабільністю освітлення. Завдяки цьому підходу

адаптивне керування яскравістю не лише покращує користувацький комфорт, а й дозволяє досягти оптимального балансу між енергоспоживанням і якістю освітлення.

Висновки до другого розділу

У цьому розділі було сформовано цілісне уявлення про архітектуру системи адаптивного управління освітленням, обґрунтовано вибір апаратного та програмного забезпечення, а також проаналізовано основні алгоритми функціонування системи. Запропонована архітектура базується на принципах модульності, масштабованості та енергоефективності, що досягається за рахунок використання протоколу Zigbee для побудови бездротової мережі пристроїв та MQTT як основного засобу зв'язку між контролером і системою автоматизації OpenHAB.

Зональний підхід до організації освітлення дозволяє індивідуалізувати керування у кожному сегменті приміщення, враховуючи локальні умови - присутність людей та рівень природного освітлення. Це сприяє як підвищенню комфорту, так і зниженню загального енергоспоживання.

Центральна роль у системі відводиться контролеру на базі ESP32, який реалізує ключову логіку аналізу сенсорних даних, прийняття рішень і управління виконавчими пристроями. Алгоритм роботи контролера базується на циклічному опитуванні зон, динамічному обчисленні середніх значень освітленості та оцінці умов для активації освітлення. Взаємодія з зовнішніми компонентами (зокрема з користувачем через OpenHAB) забезпечується уніфікованими MQTT-повідомленнями.

Особливу увагу було приділено адаптивному регулюванню яскравості, що реалізується за допомогою ПІД-регулятора. Застосування цього підходу забезпечує стабільність освітлення навіть за умов змінної освітленості та динаміки присутності, дозволяючи уникати перерегулювань або частих перемикачів. Параметри регулятора налаштовано за методом Зіглера-Нікольса, що дає змогу збалансувати швидкодію системи з її стійкістю.

У результаті розроблена системна архітектура демонструє високий потенціал для реалізації у реальних умовах, із можливістю подальшого масштабування, інтеграції з іншими IoT-компонентами та застосування у складніших сценаріях автоматизованого керування освітленням.

РОЗДІЛ 3 ОПИС РОЗРОБКИ

3.1 Реалізація прошивки системи адаптивного управління освітленням

Прошивка системи адаптивного управління освітленням є ключовим програмним компонентом, що реалізує основну логіку роботи контролера на базі ESP32. Вона забезпечує автоматичне керування освітленням у приміщенні на основі даних від сенсорів руху та освітленості, підтримуючи як автоматичний режим роботи, так і можливість зміни режимів користувачем через OpenHAB. Реалізація виконана мовою C з використанням фреймворку ESP-IDF, що дозволяє забезпечити високу продуктивність, ефективне використання апаратних ресурсів і стабільність роботи в режимах багатозадачності завдяки використанню FreeRTOS.

Основна відповідальність прошивки полягає у зборі даних від сенсорів, аналізі отриманих значень, прийнятті рішень щодо керування освітленням і формуванні команд для виконавчих пристроїв через Zigbee, а також обміні даними з OpenHAB через MQTT.

Архітектура прошивки реалізована у вигляді модульної структури з чітким розділенням відповідальності між компонентами: `main.c` (головний файл прошивки, містить реалізацію логіки роботи прошивки), `sensors.c/h` (робота із сенсорами), `lightning.c/h` (керування освітленням), `mqtt_client.c/h` (робота з протоколом MQTT), `zigbee.c/h` (робота з протоколом zigbee), `pid_controller.c/h` (модуль ПІД-регуляції), `config.h` (параметри конфігурації прошивки).

У `main.c` описана уся основна логіка роботи системи адаптивного управління освітленням освітлення. Модуль `sensors.c` забезпечує збір і оновлення даних від емуляторів сенсорів (TSL2561 для освітленості та HiLink LD2410 для руху). Модуль `lighting.c` реалізує логіку керування лампами відповідно до обраного режиму або розрахунків ПІД-регулятора. `mqtt_client.c` відповідає за публікацію статусів і прийом команд від OpenHAB, тоді як `zigbee.c` забезпечує обмін даними з мережевими пристроями. `pid_controller.c` реалізує алгоритм ПІД-

регуляції, налаштований за методом Зіглера-Нікольса для плавного керування яскравістю.

Обмін даними з системою OpenHAB організовано через протокол MQTT, що дозволяє публікувати поточний стан системи й отримувати команди керування. Формат повідомлень - JSON, що спрощує інтеграцію та підтримує auto-discovery для OpenHAB. Zigbee використовується для локального керування пристроями освітлення та отримання даних від сенсорів.

Для забезпечення точного регулювання яскравості використовується ПІД-регулятор, який формує керувальний вплив на основі похибки між цільовим і фактичним рівнями освітленості. Такий підхід забезпечує плавну зміну яскравості, стабільність освітлення й оптимальне енергоспоживання навіть у змінних умовах навколишнього середовища.

Прошивка розроблялась із використанням QEMU для емуляції ESP32, що дозволило виконувати налагодження та тестування без фізичного обладнання. Такий підхід забезпечив прискорення розробки, спрощення інтеграції з Docker-контейнерами OpenHAB і MQTT-брокера та відтворюваність сценаріїв тестування. У фінальній версії проєкту передбачено розгортання прошивки на фізичному контролері, тоді як архітектура залишається сумісною з реальною IoT-інфраструктурою.

Основними перевагами реалізованої прошивки є модульність, гнучкість налаштувань, можливість масштабування системи та простота інтеграції з іншими IoT-рішеннями.

3.2 Конфігурація інфраструктури системи

Важливим елементом проєкту системи адаптивного управління освітленням є використання Docker як середовища для розгортання основних сервісів. Завдяки контейнеризації забезпечується ізолюваність компонентів системи, спрощується процес їх налаштування та розгортання, а також підвищується зручність тестування та масштабування. Конфігурацію docker-compose файлу, який використовувався при розробці прошивки та тестуванні мережевої взаємодії

можна знайти у додатку В.1. Контейнери Mosquitto MQTT та OpenHAB розгортаються автоматично із заздалегідь підготовленою конфігурацією, що дозволяє швидко відтворити середовище системи на будь-якому сервері або робочій станції. На рисунку 3.1 наведено конфігурацію mosquitto MQTT брокера у файлі mosquitto.conf

```
listener 1882
allow_anonymous true

# налаштування логування mosquitto
log_dest file /mosquitto/log/mosquitto.log
log_type error
log_type warning
log_type notice
log_type information
log_timestamp true

# конфігурація збереження стану брокера
autosave_interval 1799
persistence true
persistence_file mosquitto.db
persistence_location /mosquitto/data/

mosquitto/config/mosquitto.conf
```

Рис. 3.1 - файл конфігурації MQTT брокера

Mosquitto MQTT виконує роль центрального брокера повідомлень у системі. Його конфігурація максимально спрощена для цілей розробки та тестування, що дозволяє приймати анонімні з'єднання та обмінюватися повідомленнями між контролером та OpenHAB без додаткової авторизації а також зберігати повідомлення логів та стану брокера, що може бути корисно якщо він різко припинив свою роботу. Брокер запущено у Docker-контейнері з використанням стандартного порту 1883 та мінімальних налаштувань.

Наступним важливим елементом нашої системи є OpenHAB - універсальна платформа автоматизації, яка виконує одразу кілька ключових функцій. Вона забезпечує централізоване керування системою освітлення, виступає у ролі інтерфейсу користувача та відповідає за обробку вхідних і вихідних MQTT-повідомлень. Завдяки OpenHAB користувач має можливість взаємодіяти з

системою через веб-інтерфейс або мобільний додаток, змінювати режими роботи освітлення, відстежувати стан датчиків і світильників, а також переглядати лог подій.

Конфігурація OpenHAB побудована на основі текстових файлів і структурно поділяється на кілька основних частин, кожна з яких виконує свою роль у забезпеченні функціональності системи:

- опис підключення до MQTT-брокера;
- визначення пристроїв та їхніх каналів (Things);
- налаштування елементів інтерфейсу (Items);
- побудова візуального інтерфейсу для користувача (Sitemap);
- реалізація простих правил для логування станів системи (Rules).

Перший елемент конфігурації відповідає за налаштування з'єднання з брокером Mosquitto. У ньому вказуються IP-адреса, порт та інші параметри з'єднання. OpenHAB виступає клієнтом MQTT та підписується на топіки для прийому даних від контролера й публікації команд.

Другий елемент визначає абстрактні представлення пристроїв (наприклад, контролер освітлення ESP32), а також канали для обміну даними. Кожен канал відповідає за окремий параметр: поточний режим освітлення, рівень яскравості, показник освітленості, стан руху тощо. Кожен канал пов'язується з конкретним топіком MQTT для обміну інформацією.

Далі йдуть елементи інтерфейсу (Items). Це логічні змінні в OpenHAB, які відображають стан каналів пристроїв і використовуються для побудови інтерфейсу. Вони також можуть об'єднуватися в групи для зручного керування (наприклад, група сенсорів, група освітлення). Через Items реалізується двосторонній обмін даними: зчитування стану та надсилання команд.

Наступним у списку йде sitemap. Цей елемент відповідає за відображення даних на веб-інтерфейсі або у мобільному додатку. В інтерфейсі користувач бачить поточний стан системи (рівень освітлення, активність датчиків, режим роботи), може обирати режими освітлення через випадаючі списки або перемикачі, переглядати історію подій і так далі.

Останніми йдуть правила (Rules). Вони дозволяють налаштувати поведінку системи при зміні станів. Наприклад, фіксуються зміни режиму освітлення, появи руху або зміни рівня освітленості. Такі правила допомагають вести журнал подій, який можна використовувати для діагностики та оптимізації роботи системи.

Уся ця конфігурація OpenHAB дозволяє реалізувати гнучку, масштабовану систему автоматизації, яка легко адаптується під зміну потреб користувача або розширення кількості зон керування.

3.3 Тестування системи

Для перевірки працездатності розробленої системи адаптивного управління освітленням було проведено комплексне тестування основних функціональних модулів. Зокрема, тестуванню підлягали модулі збору даних із сенсорів, обробки даних контролером, реалізації алгоритмів прийняття рішень, керування освітленням, а також взаємодії з OpenHAB через MQTT-протокол.

Тестування здійснювалося в умовах локального середовища з використанням Docker-контейнерів для Mosquitto MQTT брокера та OpenHAB. Такий підхід дозволив забезпечити ізолюваність компонентів, відтворюваність умов тестування та стабільну роботу системи на різних конфігураціях апаратного забезпечення. Крім того, використання QEMU для емуляції ESP32 контролера дало можливість тестувати прошивку без фізичного обладнання, що значно прискорило розробку та налагодження.

Перевірка функціональності системи охоплювала наступні аспекти роботи системи:

- Коректність обробки даних сенсорів - емуляція даних про рух та рівень освітленості з наступним прийняттям рішення щодо вмикання або вимикання освітлення.
- Робота алгоритму ПІД-регуляції - перевірка точності розрахунку яскравості освітлення відповідно до заданого рівня (наприклад, 300 lx) та стійкості до коливань вхідних даних.

- Взаємодія з MQTT-брокером та OpenHAB - публікація статусів системи та прийом керуючих команд, перевірка коректності форматів повідомлень та їх обробки.
- Відображення стану системи в OpenHAB - оновлення статусів освітлення, сенсорів і режимів роботи на веб-інтерфейсі без потреби ручного оновлення сторінки.

Для моделювання навантаження використовувалась емуляція роботи кількох віртуальних зон з різними сценаріями: поява руху при низькому освітленні, відсутність руху при достатньому освітленні, різкі зміни рівня освітлення тощо. Це дозволило перевірити стійкість алгоритмів до типових змін у середовищі та відсутність помилок при швидкій зміні даних.

Усі ключові модулі системи продемонстрували стабільну роботу відповідно до технічного завдання. Контролер своєчасно реагував на зміни стану сенсорів, коректно розраховував рівень яскравості за ПІД-регулятором і надсилав відповідні команди до пристроїв освітлення через Zigbee. Взаємодія між OpenHAB і контролером відбувалася згідно з протоколом MQTT, із коректною обробкою команд користувача та логуванням подій.

Тестування підтвердило готовність системи до подальшого розгортання на реальному обладнанні та її здатність ефективно керувати освітленням в умовах реального середовища з можливістю масштабування та інтеграції з іншими IoT-компонентами.

Висновки до третього розділу

У третьому розділі було описано основні аспекти реалізації та тестування системи адаптивного управління освітленням. Розроблена прошивка для контролера ESP32 забезпечує автоматичне керування освітленням на основі даних від сенсорів руху та освітленості. Її модульна структура дозволяє легко масштабувати систему, інтегрувати нові функції та адаптувати рішення під потреби користувача. Використання мови програмування C та фреймворку ESP-

IDF забезпечує високу продуктивність та стабільність роботи прошивки в умовах багатозадачності.

Для розгортання інфраструктури системи застосовано технології контейнеризації за допомогою Docker, що спростило налаштування, тестування та запуск основних сервісів - Mosquitto MQTT і OpenHAB. Mosquitto виконує роль надійного брокера обміну повідомленнями, тоді як OpenHAB забезпечує зручний інтерфейс користувача та централізоване керування системою освітлення.

Тестування системи підтвердило її працездатність і відповідність технічному завданню. Контролер демонструє коректну обробку даних сенсорів, стабільну роботу алгоритму ПІД-регуляції та точність керування освітленням. Інтеграція з OpenHAB через MQTT забезпечує своєчасну передачу статусів і обробку команд користувача. Проведене моделювання навантаження засвідчило стійкість системи до змін зовнішніх умов та готовність до подальшого розгортання у реальному середовищі з можливістю інтеграції з іншими IoT-компонентами.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено повнофункціональну систему адаптивного управління освітленням на базі мікроконтролера ESP32, яка поєднує модулі збору даних від сенсорів, аналізу даних, прийняття рішень та керування освітлювальними приладами. Вимоги до системи були сформовані на основі аналізу сучасних підходів до енергоефективного освітлення, особливостей роботи IoT-пристроїв та актуальних проблем автоматизації освітлювальних систем у приміщеннях. На основі цього аналізу були визначені технічні та функціональні критерії, що лягли в основу архітектури системи.

У процесі реалізації створено модульну архітектуру прошивки з використанням мови C і фреймворку ESP-IDF, що забезпечує ефективну багатозадачну роботу завдяки FreeRTOS. Для організації комунікації застосовано протокол MQTT для обміну даними з OpenHAB і Zigbee для взаємодії з сенсорами та виконавчими пристроями. Використання Docker-контейнерів для розгортання Mosquitto та OpenHAB значно спростило процес налаштування, тестування й масштабування системи, забезпечивши ізольованість компонентів та можливість гнучкої адаптації під різні середовища.

Особливістю розробленої системи є впровадження адаптивного алгоритму регулювання яскравості на основі ПІД-регулятора, налаштованого за методом Зіглера–Нікольса. Це дозволяє автоматично підтримувати заданий рівень освітленості з урахуванням реальних умов приміщення, таких як зміна природного освітлення або поява людей у зоні. Такий підхід сприяє підвищенню комфорту користувачів і одночасно оптимізує енергоспоживання. Реалізоване рішення демонструє ефективність застосування класичних алгоритмів управління у поєднанні з сучасними IoT-технологіями.

Додатково під час розробки було створено інфраструктуру для тестування системи з використанням QEMU для емуляції ESP32 та Docker-контейнерів для супутніх сервісів. Це забезпечило можливість проведення комплексного тестування у локальному середовищі, що дозволило підтвердити працездатність

системи, стабільність її роботи при динамічних змінах даних сенсорів і правильність реалізації алгоритмів прийняття рішень.

Загалом, результати виконаної роботи свідчать про доцільність і ефективність впровадження подібних систем у сучасну інфраструктуру розумного будинку та комерційних приміщень. Запропоноване рішення є масштабованим, гнучким, готовим до інтеграції з іншими IoT-платформами й здатним до подальшого розвитку, зокрема шляхом додавання нових типів сенсорів, розширення аналітичних можливостей або використання штучного інтелекту для прогнозування потреб в освітленні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Організація Об'єднаних Націй. Цілі сталого розвитку URL: <https://sdgs.un.org/goals> (дата звернення: 25.03.2025).
2. Zigbee Alliance. Zigbee Specification. - Version 3.0. Zigbee Document 05-3474-21. - Zigbee Alliance, 2022. - 312 с.
3. Козаченко Т. П., Бондар С. М. Аналіз сучасних бездротових протоколів для розумного освітлення // Системи управління, навігації та зв'язку. - 2021. - №2(62). - С. 110-115.
4. Дмитренко І. В., Смирнов В. О. Порівняльна характеристика дротових та бездротових систем управління освітленням // Світлотехніка та електроенергетика. - 2020. - Т. 53, № 1. - С. 29-36.
5. DALI Alliance. Introduction to the Digital Addressable Lighting Interface (DALI) URL: <https://www.dali-alliance.org/> (дата звернення: 22.04.2025).
6. KNX Association. KNX System Description. - Brussels: KNX International, 2021. - 152 с.
7. Philips Hue. Official Website URL: <https://www.philips-hue.com/> (дата звернення: 15.04.2025).
8. Xiaomi Smart Home. Product Documentation URL: <https://www.mi.com/global/smart-home/> (дата звернення: 16.04.2025).
9. Enlighted. Siemens Smart Lighting Platform URL: <https://www.enlightedinc.com/> (дата звернення: 14.04.2025).
10. Casambi. Bluetooth Low Energy Lighting Control URL: <https://www.casambi.com/> (дата звернення: 15.04.2025).
11. Digi International. What is Zigbee? Learn About Zigbee Wireless Mesh Technology URL: <https://www.digi.com/solutions/by-technology/zigbee-wireless-standard> (дата звернення: 29.04.2025).
12. Agcled Blog. Zigbee: The Ideal Wireless Protocol for Industrial Lighting Systems URL: - 2023.

<https://www.agcled.com/blog/zigbee-ideal-wireless-protocol-industrial-lighting-systems>
(дата звернення: 08.05.2025).

13. Multitech IoT Wiki. Zigbee URL: <https://www.multitech.com/iot-wiki/zigbee>
(дата звернення: 30.04.2025).

14. Community OpenHAB. Auto configuration of MQTT ESP32 devices in OpenHAB - Bindings URL: - 12.05.2024. [оновлений].

15. EMQX Technologies. MQTT with openHAB: A Step-by-Step Tutorial URL: <https://emqx.medium.com/mqtt-with-openhab-a-step-by-step-tutorial-53d68f622b13>
(дата звернення: 15.05.2025).

16. Hackaday.io. QEMU-ESP32 - lightweight QEMU-based ESP32 emulation with Docker URL: - 2024. <https://hackaday.io/project/201859-qemu-esp32> (дата звернення: 19.05.2025).

17. Alfian E., Ma'arif A., Chotikunnan P., Abougarair A. Optimizing Light Intensity with PID Control // Control Systems and Optimization Letters, Vol. 1, No 3, 2023. - С. 124-130. - DOI: 10.59247/csol.v1i3.38.

18. Obioma P., Agbodike O., Chen J., Wang L. ISLS: An IoT-Based Smart Lighting System for Improving Energy Conservation in Office Buildings URL: - 2025. <https://arxiv.org/abs/2503.13474> (дата звернення: 23.05.2025).

19. Espressif Systems. ESP-IDF Programming Guide URL: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/> (дата звернення: 15.05.2025)

20. Eclipse Foundation. Eclipse Mosquitto - An open source MQTT broker URL: <https://mosquitto.org/> (дата звернення: 02.05.2025)

21. OpenHAB Foundation. openHAB Documentation URL: <https://www.openhab.org/docs/> (дата звернення: 04.05.2025)

22. Howard M., Irvin R. B. ESP32: QEMU emulation within a Docker container // arXiv preprint arXiv:2303.12345 URL: - 2023. <https://arxiv.org/abs/2303.12345>

23. Zandberg K., Baccelli E. Femto-Containers: DevOps on Microcontrollers // arXiv preprint arXiv:2103.06739 URL: - 2021. <https://arxiv.org/abs/2103.06739>

24. Espressif Systems. Zigbee2MQTT with ESP32-C6: Pairing tutorial URL: <https://community.home-assistant.io/t/zigbee2mqtt-with-esp32-c6/412099>
25. MDPI Sensors. Design and implementation of ESP32-based IoT devices // Sensors. - 2023. - Т. 23, № 6. - С. 2715. - DOI: <https://doi.org/10.3390/s23062715>
26. Home Assistant Community. Zigbee2MQTT with adaptive lighting URL: <https://community.home-assistant.io/t/zigbee2mqtt-with-adaptive-lighting/389503>
27. Contextual Electronics. Need QEMU help: TLS handshake failing (ESP-IDF) URL: <https://forum.contextualelectronics.com/t/need-qemu-help-tls-handshake-failing-esp-idf/5147>
28. Docker Inc. Docker Documentation URL: <https://docs.docker.com/> (дата звернення: 07.05.2025).
29. FreeRTOS. FreeRTOS Documentation URL: <https://www.freertos.org/> (дата звернення: 10.05.2025).
30. MQTT.org. MQTT Version 5.0 Specification URL: <https://mqtt.org/mqtt-specification/> (дата звернення: 03.05.2025).
31. IEEE. IEEE 802.15.4 Standard for Low-Rate Wireless Networks URL: https://standards.ieee.org/standard/802_15_4-2020.html (дата звернення: 19.04.2025).
32. Docker Hub. Eclipse Mosquitto Docker Image URL: https://hub.docker.com/_/eclipse-mosquitto (дата звернення: 03.05.2025).
33. Arduino Project Hub. ESP32 and Zigbee Integration URL: – Режим доступу: <https://create.arduino.cc/projecthub> (дата звернення: 05.05.2025).
34. Docker Inc. Docker Compose Documentation URL: – Режим доступу: <https://docs.docker.com/compose/> (дата звернення: 12.05.2025).
35. Eclipse Foundation. Eclipse Mosquitto Configuration Examples URL: – Режим доступу: <https://mosquitto.org/documentation/> (дата звернення: 09.04.2025).
36. PID Control: Ziegler-Nichols Tuning URL: MathWorks. – Режим доступу: <https://www.mathworks.com/help/control/ug/pid-controller-tuning.html> (дата звернення: 02.03.2025).
37. ДБН В.2.5-28:2018. Природне і штучне освітлення. - Чинний від 01.10.2018. - К.: Мінрегіон України, 2018. - 106 с.

ДОДАТКИ

ДОДАТОК А.1

“Описова таблиця для існуючих систем адаптивного управління освітленням”

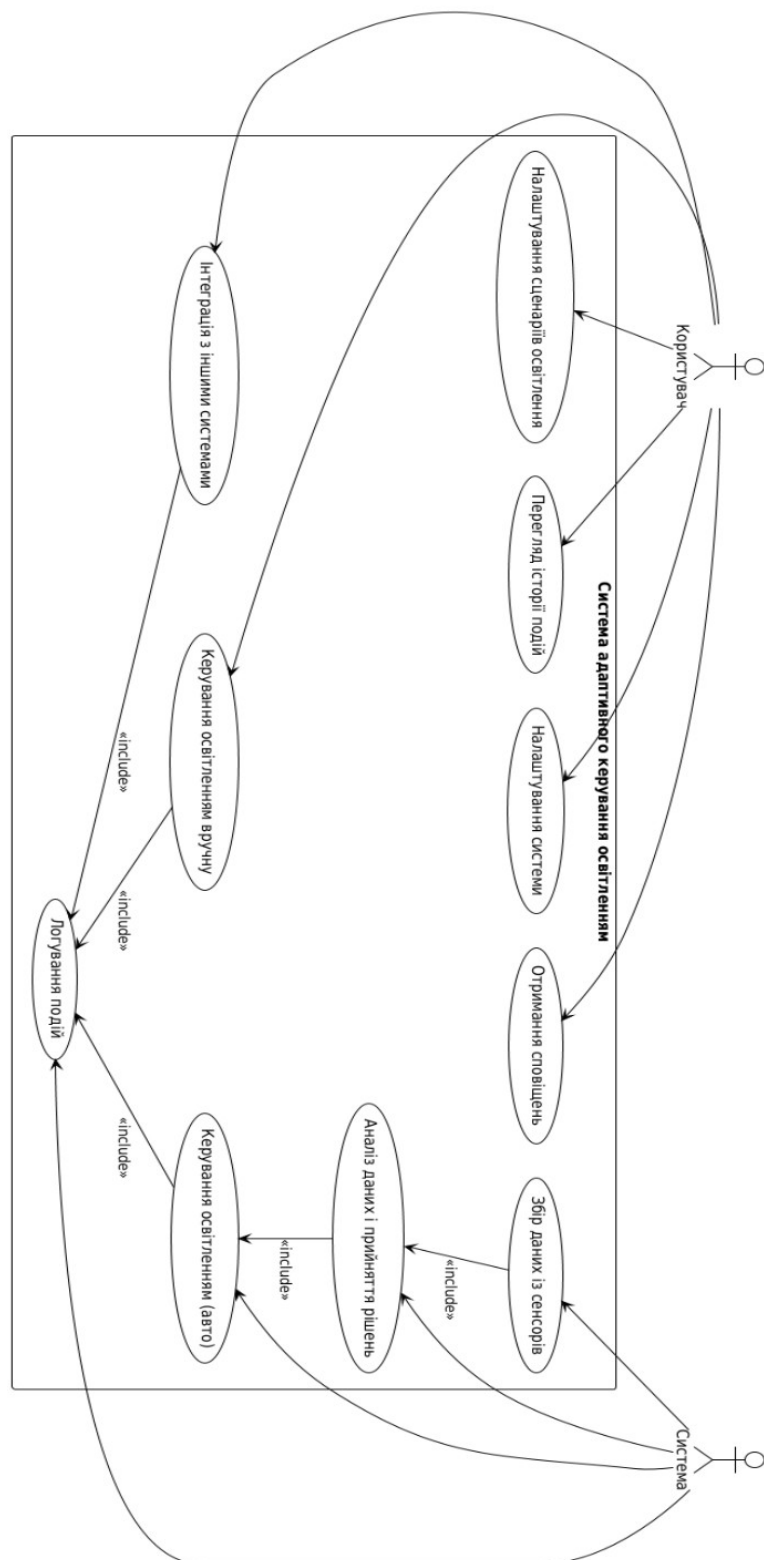
Система	Тип з'єднання	Протокол	Принцип роботи	Сфери застосування	Переваги	Недоліки
PUSH (DIM)	Провідна	-	Аналогове керування яскравістю через коротке/довге натискання кнопки	Кімнати, локальні рішення	Простота реалізації, низька вартість, надійність	Відсутність віддаленого керування, обмежене масштабування
TRIAC	Провідна	-	Фазове регулювання напруги для димінгу ламп	Побутове, просте освітлення	Доступність, сумісність з традиційними лампами	Не сумісна з усіма LED-лампами, можливі спотворення сигналу
DALI	Провідна	DALI	Цифровий протокол з двостороннім зв'язком для точного керування освітленням	Комерційні об'єкти	Масштабованість, точність, зворотний зв'язок	Потреба в додатковому кабелюванні, вищі початкові витрати
DMX-512	Провідна	DMX	Цифровий протокол для високошвидкісного керування освітленням у реальному часі	Сцени, архітектурне підсвічування	Точність, синхронізація, підтримка великої кількості пристроїв	Обмежена кількість адрес, складність налаштування
KNX	Провідна/ Бездротова	KNX/ KNX RF	Універсальний протокол автоматизації будівель з підтримкою різних носіїв	Інтелектуальні будівлі	Масштабованість, стабільність, підтримка широкого спектру пристроїв	Висока складність конфігурації, висока вартість впровадження
Philips HUE	Бездротова	Zigbee/ Bluetooth	Керування через мобільний додаток або голосових помічників, використовує Zigbee	Розумні домогосподарства	Зручність, інтеграція з іншими IoT-пристроями, просте встановлення	Висока ціна, залежність від хабу
Xiaomi Smart Home	Бездротова	Zigbee/ Wi-Fi	Екосистема з підтримкою Wi-Fi, Zigbee та Bluetooth, керування через додаток	Розумні домогосподарства	Доступна ціна, широка сумісність, гнучкість	Залежність від сервісів Xiaomi, іноді нестабільна робота з деякими сторонніми пристроями

Продовження таблиці додатку А.1

Enlighted by Siemens	Бездротова	Zigbee/ Wi-Fi	Інтелектуальна система з мережею бездротових датчиків та штучним інтелектом	Бізнес, комерційні будівлі	Масштабованість, аналітичні можливості, енергозбереження	Висока вартість впровадження
Casambi	Бездротова	Zigbee/ Wi-Fi	Керування на базі Bluetooth Low Energy, пристрої утворюють mesh-мережу	Архітектурне/ офісне освітлення	Простота встановлення, висока адаптивність, підтримка різних виробників	Обмеження Bluetooth у великих просторах, складність інтеграції зі сторонніми системами

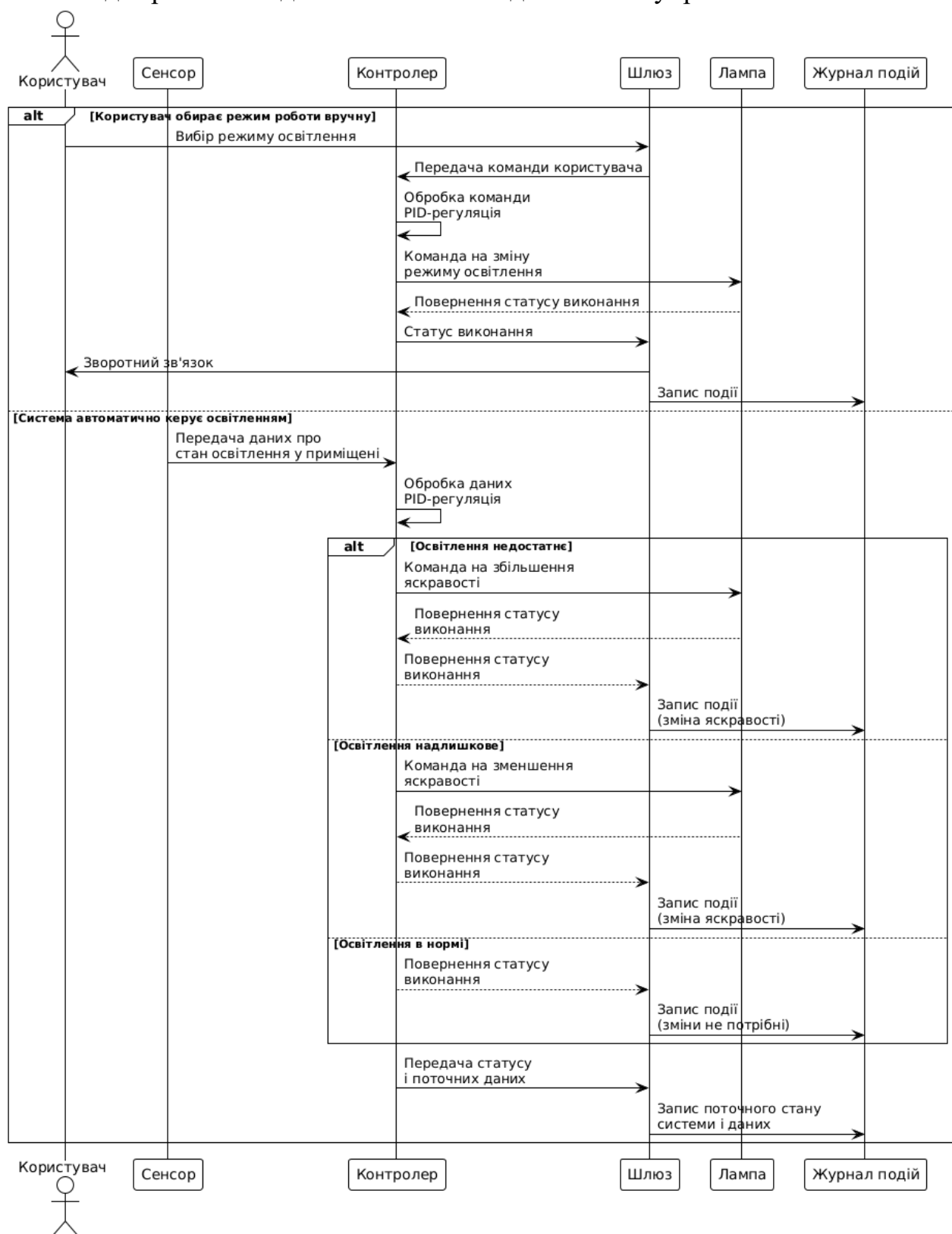
ДОДАТОК А.2

UML-діаграма прецедентів



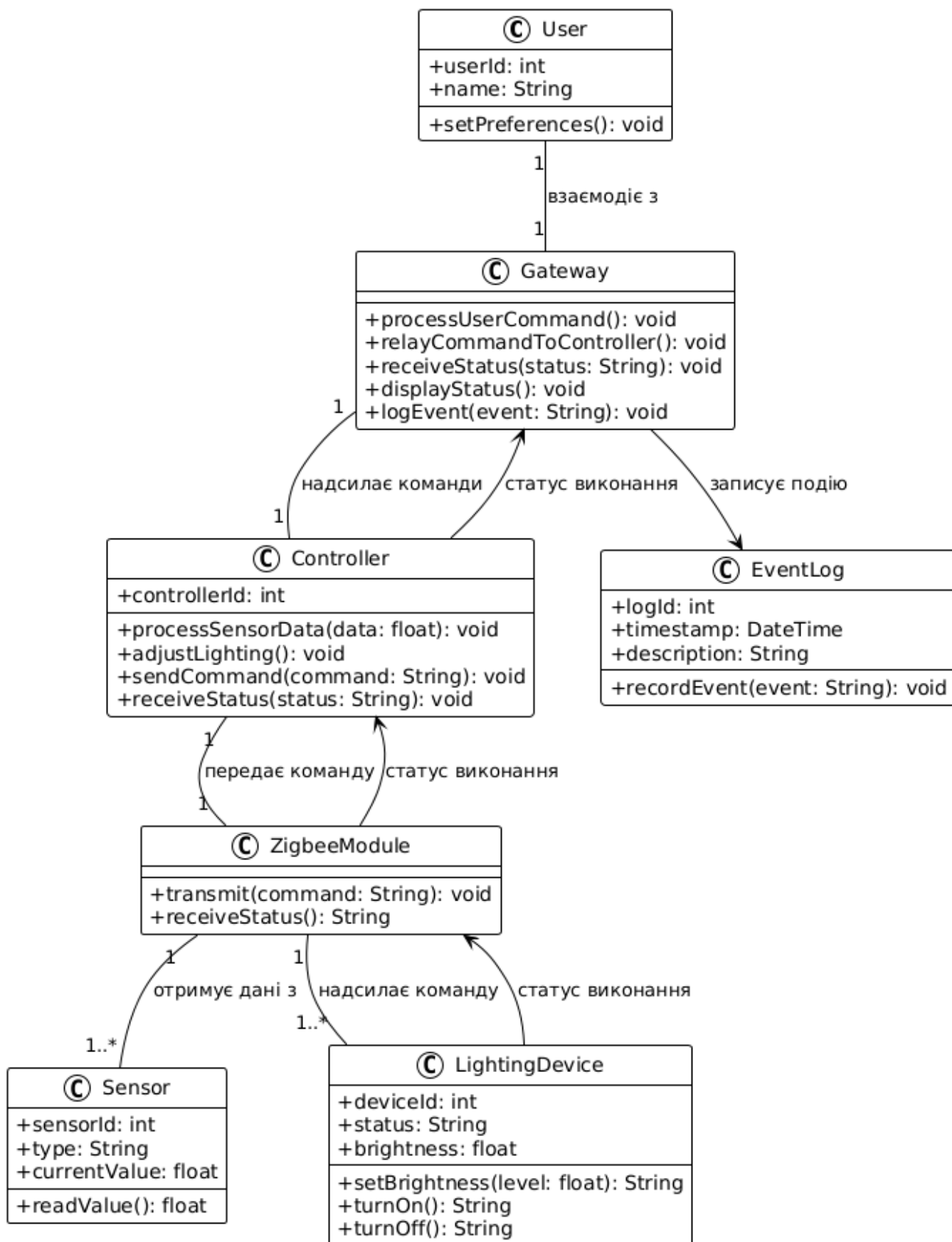
ДОДАТОК А.3

UML-діаграма послідовності системи адаптивного управління освітленням



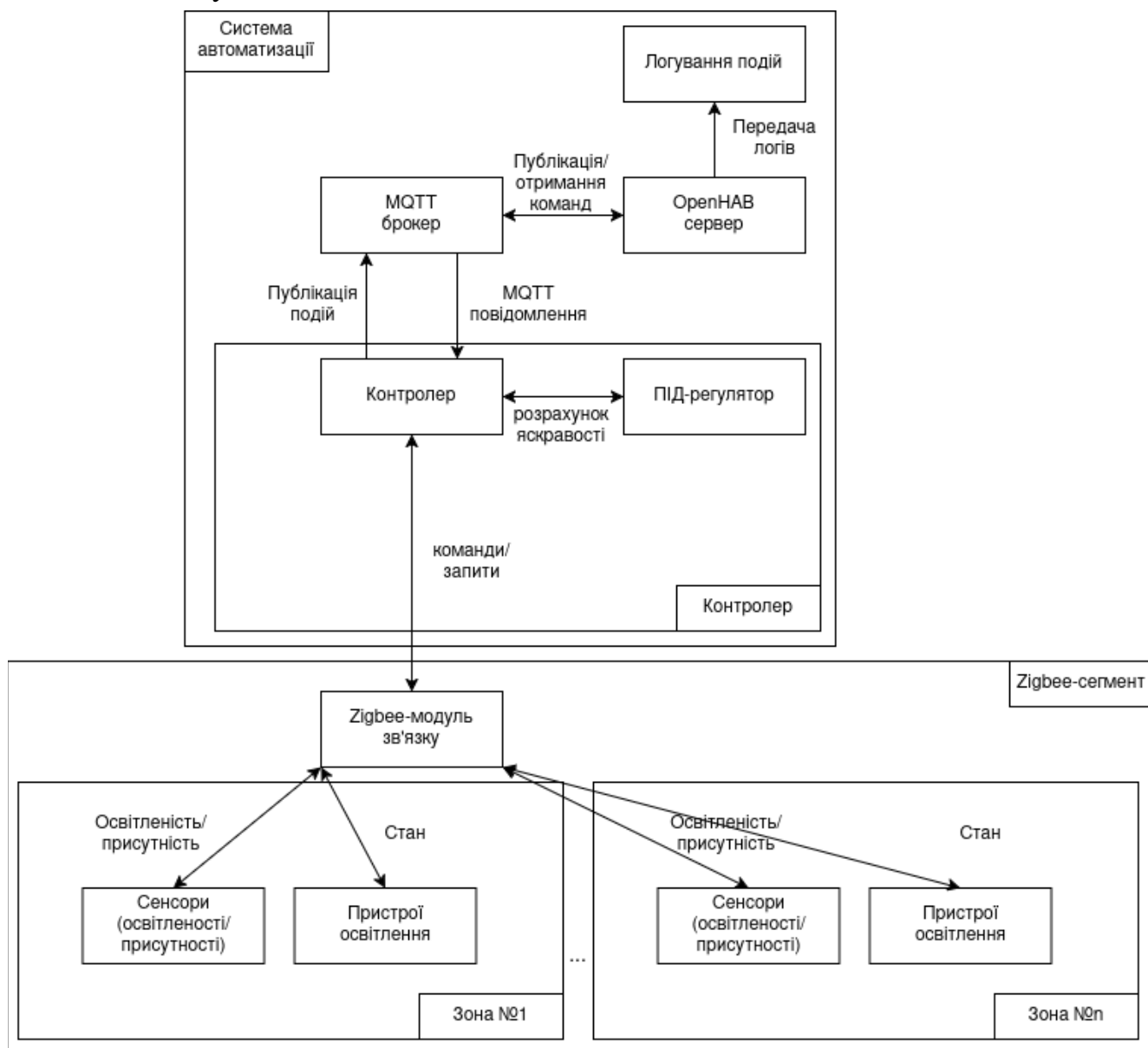
ДОДАТОК А.4

UML-діаграма класів системи адаптивного управління на базі IoT



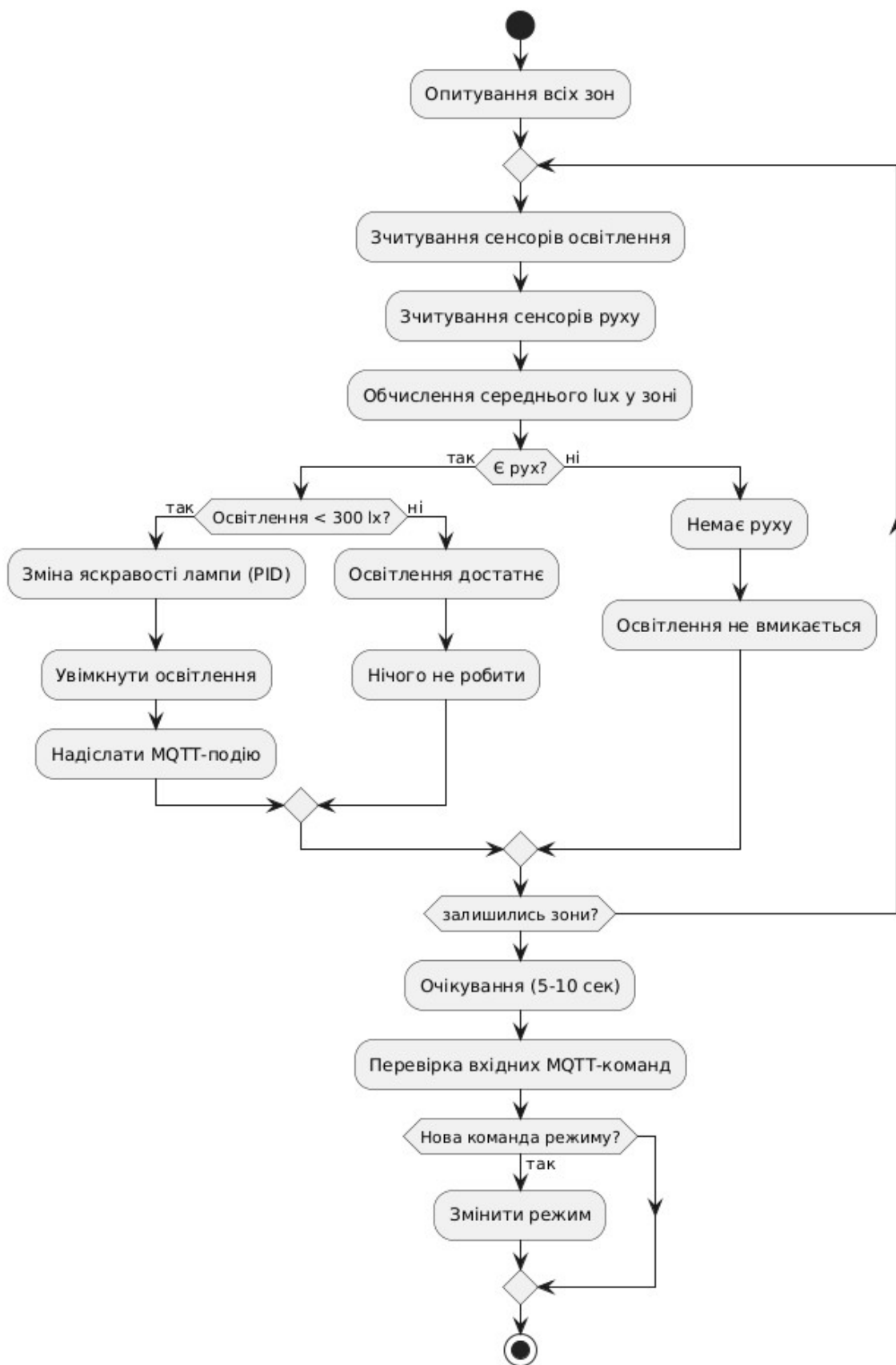
ДОДАТОК Б.1

Функціональна схема взаємодії основних компонентів системи



ДОДАТОК Б.2

Діаграма алгоритму роботи контролера



ДОДАТОК В.1
docker-compose файл

```
services:
  mosquitto:
    image: eclipse-mosquitto:latest
    container_name: mosquitto
    restart: always
    ports:
      - "1883:1883"
      - "9001:9001"
    volumes:
      - ./mosquitto/config:/mosquitto/config:z
      - ./mosquitto/data:/mosquitto/data:z
      - ./mosquitto/log:/mosquitto/log:z
    networks:
      - iotnet

  openhab:
    container_name: ${COMPOSE_PROJECT_NAME}-server
    image: openhab/openhab:latest
    restart: unless-stopped
    ports:
      - "8080:8080"
      - "8443:8443"
    group_add:
      - tty
    volumes:
      - $OPENHAB_ADDONS:/openhab/addons:z
      - $OPENHAB_CONF:/openhab/conf:z
      - $OPENHAB_USERDATA:/openhab/userdata:z
    environment:
      - CRYPTO_POLICY=unlimited
      - EXTRA_JAVA_OPTS=${EXTRA_JAVA_OPTS}
    networks:
      - iotnet

networks:
  iotnet:
    driver: bridge
    driver_opts:
      com.docker.network.bridge.name: br-iotnet
    ipam:
      config:
        - subnet: 172.18.1.0/24
          gateway: 172.18.1.1
```