

Я, Іванов Максим Олександрович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу з академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

"11" червня 2025 р.



Максим ІВАНОВ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет _____ АКТ
Кафедра _____ КІТАР
Рівень вищої освіти _____ перший (бакалаврський)
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
(код і повна назва)
Тип програми _____ Освітньо-професійна
Освітня програма _____ Системна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____ (підпис)
« ____ » _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві ІВАНОВУ Максиму Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення автоматичної системи розумного будинку на Node-RED
- Затверджена наказом по університету від 19.05.2025 р. № 391 Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 09.06.2025 р
3. Вихідні дані до роботи _____
 - 3.1 Одноплатний комп'ютер Raspberry PI 4B WiFi
 - 3.2 Датчик рівня освітлення BH1750
 - 3.3 Wi-Fi розетка TuYa Gosund EP2
 - 3.4 Мова програмування Python
 - 3.5 Бібліотека python-smbus2 для керування датчиком освітлення BH1750
 - 3.6 JSON, як спосіб створення потоку (flow) в дашборді Node-RED
4. Перелік питань, що потрібно опрацювати в роботі
 - 4.1 Аналіз технічного завдання
 - 4.2 Розроблення структури системи автоматизації освітлення
 - 4.3 Практична частина
 - 4.4 Розширення та масштабування системи
 - 4.5 Реалізація бездротового керування освітленням
 - 4.6 Потенціал розроблюваної системи і порівняння з існуючими рішеннями

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

5.1 Презентація “Розроблення автоматичної системи розумного будинку на Node-RED” 8 ст

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз технічного завдання	28.04-04.05.2025	виконано
2	Розробка структури системи	05.05-10.05.2025	виконано
3	Робота над практичною частиною	11.05-17.05.2025	виконано
4	Реалізація бездротового керування системою	18.05-25.05.2025	виконано
5	Дослідження потенціалу розроблюваної системи	26.05-31.05.2025	виконано
6	Оформлення пояснювальної записки	11.06-14.06.2025	виконано
7	Подання роботи на перевірку Інтернет-сервісом StrikePlagiarism	14.06-17.06.2025	
8	Подання роботи на рецензію	17.06-18.06.2025	
9	Подання роботи на підпис зав. кафедри	18.06-20.06.2025	
11	Подання кваліфікаційної роботи в ЕК	20.06-26.06.2025	

Дата видачі завдання 19.05.2025 р.

Здобувач _____ Максим ІВАНОВ
(підпис) (посада, власне ім'я, прізвище)

Керівник роботи _____ Микола СТАРОДУБЦЕВ
(підпис) (власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 60 с., 6 рис., 2 дод., 10 джерел.

ІНТЕРФЕЙС, ІНФОРМАЦІЙНА СИСТЕМА, ПРОГРАМНИЙ ЗАСІБ,
БІЗНЕС-ПРОЦЕСИ, АЛГОРИТМ.

Об'єкт розробки – система автоматизації процесів у розумному будинку.

Предмет розробки – інтелектуальна система автоматизованого керування освітленням на базі Node-RED і Raspberry Pi.

Мета – створення гнучкої, масштабованої системи розумного будинку

У першому розділі проаналізовано технічне завдання, актуальність теми, принципи автоматизації освітлення, Node-RED, Raspberry Pi, сенсор BH1750.

У другому розділі розроблено структуру системи, схему потоків Node-RED і веб-інтерфейс для зручного керування.

У третьому розділі описано практичну реалізацію: налаштування Raspberry Pi та Node-RED, підключення BH1750..

У четвертому розділі висвітлено масштабування: підключення кількох зон освітлення, бездротові реле, архітектура з ESP8266/ESP32 через MQTT, централізоване керування.

У п'ятому розділі представлено інтеграцію розумної Wi-Fi розетки Gosund EP2 у систему через MQTT.

У шостому розділі розглянуто перспективи розширення системи, порівняно з комерційними рішеннями та визначено ключові переваги.

У результаті створено функціональну, масштабовану систему з низькою вартістю, відкритою архітектурою та широкими можливостями індивідуальної автоматизації

ABSTRACT

Explanatory Note: 60 pages, 6 figures, 2 appendices, 10 sources.

INTERFACE, INFORMATION SYSTEM, SOFTWARE, BUSINESS PROCESSES, ALGORITHM.

Object of development – a system for automating processes in a smart home.

Subject of development – an intelligent system for automated control of lighting based on Node-RED and Raspberry Pi.

Objective – creation of a flexible, scalable smart home system.

In the first chapter, the technical specifications, relevance of the topic, principles of lighting automation, Node-RED, Raspberry Pi, and the BH1750 sensor are analyzed.

In the second chapter, the system structure, Node-RED flow diagram, and a web interface for convenient control are developed.

In the third chapter, the practical implementation is described: configuring Raspberry Pi and Node-RED, connecting the BH1750 sensor.

In the fourth chapter, system scaling is addressed: connecting multiple lighting zones, wireless relays, architecture with ESP8266/ESP32 via MQTT, centralized control.

In the fifth chapter, the integration of the smart Wi-Fi socket Gosund EP2 into the system via MQTT is presented.

In the sixth chapter, prospects for system expansion are considered, compared with commercial solutions, and key advantages are identified.

As a result, a functional, scalable system with low cost, open architecture, and extensive capabilities for individual home automation has been developed.

ЗМІСТ

Перелік умовних скорочень	9
Вступ	10
1 Аналіз технічного завдання	13
1.1 Актуальність розробки системи розумного будинку	13
1.2 Загальні принципи систем автоматизації освітлення	15
1.3 Платформа Node-RED: можливості та переваги	16
1.4 Raspberry Pi як апаратна основа	18
1.5 Сенсор освітленості BH1750: аналіз параметрів	20
1.6 Модуль реле: принцип дії, особливості	23
1.7 Протоколи зв'язку в IoT-системах: I2C, MQTT	26
1.8 Вибір логіки автоматизації в Node-RED	27
1.9 Структурна схема системи	29
1.10 Приклад сценарію автоматизації	30
1.11 Потенціал масштабування системи	32
1.12 Висновки по технічному завданню	33
2 Розроблення структури системи автоматизації освітлення	34
2.1 Загальна структура програмного забезпечення	34
2.2 Логічна структура потоку Node-RED	34
2.3 Веб-інтерфейс користувача (Node-RED Dashboard)	36
3 Практична реалізація та конфігурація системи	38
3.1 Установка та налаштування Raspberry Pi	38
3.2 Встановлення та запуск Node-RED	39
3.3 Підключення сенсора BH1750 до Raspberry Pi	40
3.4 Розробка базового потоку автоматизації в Node-RED	42
3.5 Веб-інтерфейс користувача: створення та налаштування	43
3.6 Ручне та автоматичне керування світлом	44
4 Розширення та масштабування системи	46
4.1 Підключення кількох зон освітлення	46

	8
4.2 Використання бездротових реле (Wi-Fi / Bluetooth)	48
4.3 Розподілена архітектура на ESP8266/ESP32 з MQTT	50
4.4 Централізоване керування з Node-RED	52
4.5 Інтеграція з голосовими помічниками або мобільними застосунками	53
5 Реалізація бездротового керування освітленням.....	55
5.1 Характеристика та вибір розумної розетки Gosund EP2.....	55
5.2 Інтеграція розетки Gosund EP2 у систему автоматизації через MQTT.....	56
6 Потенціал розроблюваної системи і її порівняння з існуючими рішеннями.....	58
6.1 Потенціал масштабування системи розумного будинку.....	58
6.2 Порівняння з готовими рішеннями та конкурентні переваги.....	59
Висновки	61
Перелік посилань	62
Додаток А Апробація результатів роботи.....	67
Додаток Б Демонстраційний матеріал.....	69

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – база даних

БЗ – база знань

СК – система керування

ШИМ – широтно-імпульсна модуляція

LED – світлодіод (Light Emitting Diode)

IoT – Інтернет речей (Internet of Things)

MQTT – протокол передавання телеметричних даних (Message Queuing Telemetry Transport)

Node-RED – інструмент потокового програмування для інтеграції пристроїв та сервісів

SCADA – система диспетчерського контролю та збору даних (Supervisory Control and Data Acquisition)

TTS –технологія перетворення тексту в мову (Text-to-Speech)

TTS/ESP – мікроконтролери ESP8266 / ESP32 (для IoT-пристроїв)

UI – інтерфейс користувача (User Interface)

Wi-Fi – бездротова технологія передачі даних

ВСТУП

На сьогоднішній день стрімкий розвиток цифрових технологій і необхідність підвищення енергоефективності, безпеки та комфорту в житлі призводить до зростання попиту на автоматизовані системи керування. Особливої актуальності набувають рішення, пов'язані з впровадженням розумних будинків, які забезпечують інтеграцію різних пристроїв у єдину систему керування. Завдяки таким технологіям можна досягти оптимізації використання ресурсів, підвищення рівня безпеки, автоматизації побутових процесів та дистанційного керування обладнанням. У цьому контексті платформа Node-RED, яка дозволяє створювати гнучкі та масштабовані системи автоматизації без глибоких знань програмування, є особливо корисною для реалізації концепції «розумного дому».

Однією з ключових форм підготовки висококваліфікованих фахівців у сфері автоматизації є виконання кваліфікаційної роботи бакалавра. Ця робота спрямована на узагальнення та практичне застосування знань і навичок, здобутих у процесі навчання, розвиток вміння самостійного вирішення інженерних задач, а також підготовку до професійної діяльності у відповідній галузі.

Кваліфікаційна робота бакалавра дозволяє продемонструвати здатність студента аналізувати технічні проблеми, розробляти та впроваджувати ефективні рішення, а також оформляти результати своєї роботи згідно з вимогами сучасних стандартів. Виконання такої роботи є важливим етапом у підготовці фахівців для ринку праці та подальшого професійного зростання.

Практична частина кваліфікаційної роботи передбачає розробку та дослідження реальної автоматизованої системи, що відповідає актуальним вимогам галузі. Робота виконувалась на кафедрі комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки Харківського національного університету радіоелектроніки.

Метою кваліфікаційної роботи є розвиток уміння самостійної роботи з літературними джерелами, закріплення навичок розрахункової, експериментальної та дослідницької діяльності, а також практичне застосування цих навичок при розробці сучасної автоматизованої системи.

Мета роботи – розробка автоматичної системи розумного будинку з використанням платформи Node-RED, яка забезпечуватиме моніторинг та керування побутовими процесами.

Об'єкт розробки – система керування елементами розумного будинку.

Предмет розробки – автоматизована система на базі Node-RED для інтеграції та керування пристроями у середовищі розумного будинку.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз існуючих систем керування розумним будинком;
- розробити структурну схему системи;
- провести підбір елементної бази;
- реалізувати прототип автоматизованої системи;
- протестувати функціонування системи за різними сценаріями роботи;
- оформити кваліфікаційну роботу згідно ДСТУ 3008:2015 [1], а також з методичними вказівками з підготовки й оформлення кваліфікаційної роботи здобувачами першого (бакалаврського) рівня вищої освіти спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» [2].

Отримані результати роботи можна віднести до Цілі сталого розвитку 9 “Промисловість, інновації та інфраструктура”, а саме п. 9.4 “Сприяти прискореному розвитку високо- та середньо-високотехнологічних секторів переробної промисловості, які формуються на основі використання ланцюгів «освіта – наука – виробництво» та кластерного підходу за напрямками розвиток інноваційної екосистеми”, індикатор 9.4.1.

1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Актуальність розробки системи розумного будинку

У сьогоденному цифровому світі питання енергоефективності, зручності та захищеності житла стає дедалі важливішим. Саме тому концепція "розумного будинку" зайняла провідне місце в сфері автоматизації. Споживачі, згідно з останніми тенденціями, прагнуть оптимізувати керування електроприладами, мінімізувати витрати енергоресурсів та зробити взаємодію з оселею комфортною і зрозумілою.

Під поняттям "розумний дім" (Smart Home) мається на увазі комплекс інженерних рішень і технологій, що дозволяють автоматизувати процеси керування різними системами в будинку. Головною метою є підвищення рівня комфорту, забезпечення безпеки, а також зменшення витрат на енергоресурси.

До складу системи розумного будинку можуть входити:

- система автоматизації освітлення;
- кліматичне обладнання (контроль температури, вологості, вентиляції);
- система відеоспостереження та охоронна сигналізація;
- датчики руху, відкриття, задимлення, витоку газу або води;
- інтелектуальні розетки та вимикачі;
- система мультимедійного керування (аудіо, відео, домашній кінотеатр);
- інтеграція з голосовими помічниками (Google Assistant, Alexa, Siri);
- сценарії для побутової техніки (пральна машина, кавоварка, пилосос тощо);
- віддалений доступ і моніторинг через Інтернет або мобільний додаток.

Однією з ключових переваг сучасних систем є можливість гнучкої конфігурації, що дозволяє поступово додавати нові елементи відповідно до потреб. Замість дорогих пропрієтарних рішень дедалі частіше використовуються відкриті програмні платформи, які не вимагають глибоких знань програмування.

Особливої популярності в цьому контексті набуває Node-RED – візуальний інструмент потокового програмування, який дає змогу інтегрувати сенсори, реле, розумні пристрої та хмарні сервіси в єдину автоматизовану систему. Завдяки простому інтерфейсу, великій кількості модулів і підтримці стандартних протоколів (MQTT, HTTP, WebSocket), Node-RED забезпечує швидку розробку сценаріїв автоматизації та є ідеальним інструментом для створення локального або гібридного розумного будинку.

Отже, розробка системи автоматизації освітлення на основі Node-RED (рис. 1.1) є актуальною, оскільки:

- вона відповідає світовим тенденціям у галузі енергоефективного житла;
- дозволяє гнучко налаштовувати сценарії управління без складного програмування;
- є доступною для впровадження як у нових, так і в наявних помешканнях;
- сприяє набуттю практичних навичок інтеграції програмно-апаратних рішень в рамках сучасної спеціальності.

1.2 Загальні принципи систем автоматизації освітлення

Системи автоматизації освітлення – один із найбільш доступних, але дієвих шляхів впровадження технологій розумного дому. Їхня головна мета – створення ідеального освітлення приміщень залежно від навколишнього середовища, пори доби, присутності людей та інших параметрів.

Ключові аспекти автоматизації освітлення:

а) керування яскравістю світла – базується на значеннях освітленості, які збираються з датчиків, на зразок BH1750, TSL2561 та подібних;

б) сенсори присутності або руху – дають можливість автоматично вмикати освітлення в разі появи особи в кімнаті (PIR-датчики);



Рисунок 1.1 – Приклад структури системи розумного будинку

- в) графіки або сценарії – налаштування часу автоматичного увімкнення/вимкнення освітлення (скажімо, при заході сонця);
- г) веб- або мобільне управління – інтеграція з інтерфейсом користувача для ручного регулювання з використанням смартфона чи браузера;
- д) взаємодія з іншими системами – наприклад, автоматичне вимкнення світла при ввімкненні телевізора або відкритті штор.

Ця система не тільки допомагає скоротити споживання електроенергії, а й робить використання житлового простору більш комфортним та зручним. Вона може бути реалізована як з використанням високоякісних промислових контролерів, так і на базі відкритих програмно-апаратних рішень, наприклад, Raspberry Pi з Node-RED.

Автоматизація освітлення – це також перший крок до побудови цілісного розумного будинку, оскільки вона дає змогу зрозуміти принципи збирання, опрацювання та застосування даних у системах IoT.

1.3 Платформа Node-RED: можливості та переваги

Node-RED – це інструмент для програмування потоків, розроблений в IBM для створення логіки обробки подій в системах Інтернету речей (IoT). Його головна перевага – використання візуального середовища для розробки сценаріїв автоматизації, що дає змогу будувати складну логіку без необхідності ручного написання коду. Це робить платформу особливо привабливою для проєктів автоматизації в домашніх умовах, зокрема, в системах розумного будинку.

Функціональні можливості Node-RED:

- інтуїтивний графічний інтерфейс, що працює через веб-браузер. Дозволяє створювати, налагоджувати та запускати потоки обробки даних методом "drag & drop";
- вбудована підтримка різноманітних протоколів обміну даними, таких як MQTT, HTTP, WebSocket, TCP, UDP, що забезпечує гнучку інтеграцію з різними пристроями та сервісами;
- можливість обробки даних за допомогою вбудованих JavaScript-функцій, що дає змогу реалізовувати як просту, так і складну логіку взаємодії з датчиками та виконавчими пристроями;
- підключення до популярних хмарних сервісів, як-от Telegram, Google Sheets, Amazon AWS, Microsoft Azure, а також до локальних серверів, зокрема Home Assistant;
- наявність надбудови Node-RED Dashboard для створення графічного інтерфейсу користувача, що дозволяє реалізувати візуальне керування системою автоматизації освітлення через браузер;
- гнучке розширення функціональності шляхом встановлення додаткових

вузлів (node packages) з офіційного каталогу, яких на сьогодні є понад 4000;

– підтримка створення подієвих сценаріїв автоматизації, заснованих на логіці "якщо – то", з використанням таймерів, умов, фільтрації даних та інших засобів програмної логіки.

Переваги Node-RED у контексті автоматизації освітлення:

– простота розробки та низький поріг входження, що дозволяє створити повноцінну систему навіть користувачам без глибоких знань програмування;

– кросплатформенність, що забезпечує сумісність Node-RED з Windows, macOS, Linux та вбудованими платформами, зокрема Raspberry Pi;

– відкритий вихідний код та активна підтримка спільноти, що гарантує постійне оновлення функціоналу та наявність великої кількості готових прикладів;

– зручна інтеграція з апаратними платформами, як-от Arduino, ESP8266, ESP32, а також з API сторонніх сервісів;

– висока швидкість розробки, що дозволяє в короткі строки реалізувати та налаштувати логіку управління освітленням, включаючи автоматичні та ручні режими управління.

Таким чином, використання Node-RED для побудови системи автоматизації освітлення є виправданим з точки зору зручності, функціональності та перспективності. Ця платформа дозволяє створювати масштабовані та адаптивні рішення, які можна інтегрувати в більші системи розумного будинку.

1.4 Raspberry Pi як апаратна основа

Для втілення системи автоматизації освітлення доцільно застосувати одноплатний комп'ютер Raspberry Pi, що виконуватиме роль головного контролера. Raspberry Pi поєднує в собі прийнятну ціну, компактні розміри, широкий вибір апаратних інтерфейсів та повноцінну підтримку операційної системи Linux, що дає можливість ефективно реалізовувати локальні

IoT-рішення.

Raspberry Pi являє собою універсальну платформу для створення інтелектуальних систем керування, зокрема, систем автоматизації побутових процесів. У контексті цієї роботи використовується модель Raspberry Pi 4 Model B, яка вирізняється збільшеною продуктивністю та розширеним набором периферійних портів, забезпечуючи сумісність з різноманітними сенсорами та виконавчими пристроями.

Ключові технічні характеристики Raspberry Pi 4 Model B:

- а) центральний процесор: чотирьохядерний ARM Cortex-A72, 64-біт, з тактовою частотою 1,5 ГГц;
- б) оперативна пам'ять: 2 ГБ / 4 ГБ / 8 ГБ LPDDR4 SDRAM (залежить від конфігурації);
- в) графічний процесор: VideoCore VI (підтримка OpenGL ES 3.0);
- г) зовнішні інтерфейси:
 - 1) 2 порти USB 3.0 та 2 порти USB 2.0;
 - 2) 2 micro-HDMI порти з підтримкою 4K;
 - 3) Ethernet, Bluetooth 5.0, Wi-Fi 802.11ac;
 - 4) 40-контактний роз'єм GPIO для під'єднання сенсорів та модулів;
 - 5) накопичувач: microSD (підтримка системного завантаження та зберігання даних).

Платформа підтримує повнофункціональну операційну систему Raspberry Pi OS (на базі Debian Linux), що дозволяє використовувати її як сервер, шлюз даних або програмований контролер. Завдяки відкритій архітектурі, велика кількість бібліотек та готових проєктів доступна для спрощення розробки систем автоматизації.

В межах цієї розробки Raspberry Pi виконує наступні функції:

- збір та обробка даних з сенсорів (зокрема, датчика освітленості BH1750);
- керування виконавчими пристроями (реле-модулями);
- робота Node-RED-серверу для розробки логіки автоматизації;

- хостинг веб-інтерфейсу для управління освітленням (Node-RED Dashboard);
- локальне зберігання та передача даних.

Застосування Raspberry Pi (рис. 1.2) надає можливість розробляти масштабовані рішення з можливістю розширення за рахунок під'єднання додаткових пристроїв або інтеграції з хмарними платформами. Крім того, завдяки компактним розмірам та низькому енергоспоживанню, пристрій легко інтегрується в різноманітні середовища – від настінних корпусів до прихованих монтажних блоків.

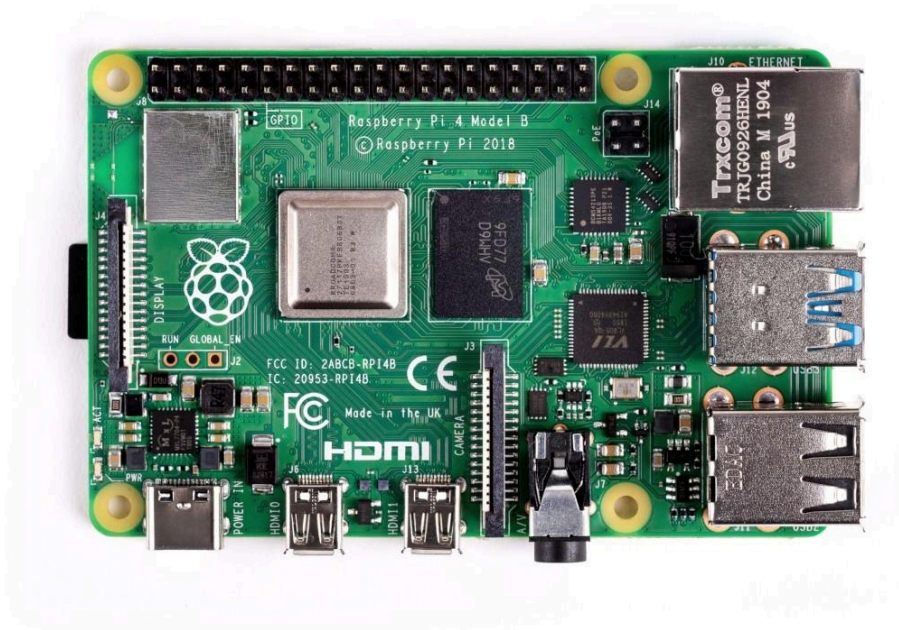


Рисунок 1.2 – Raspberry Pi 4 Model B

Таким чином, Raspberry Pi виступає як ефективна, доступна та функціональна апаратна платформа для реалізації локальних систем автоматизації, у тому числі для автоматизації освітлення в межах розумного будинку.

1.5 Сенсор освітленості BH1750: аналіз параметрів

Для реалізації функції автоматичного керування освітленням необхідне використання датчика, що дозволяє точно визначати рівень освітленості навколишнього середовища. У межах даної розробки було обрано сенсор BH1750 (рис. 2.3), який є одним із найбільш популярних і ефективних пристроїв для вимірювання освітленості у системах автоматизації.



Рисунок 2.3 – Сенсор освітленості BH1750

BH1750 – це цифровий датчик освітленості, який вимірює світло в люксах та відправляє дані через інтерфейс I2C. Завдяки простоті використання, високій точності та апаратному аналізу сигналу, цей модуль є відмінним вибором для розумних будинків, особливо для автоматизації світла.

Основні параметри сенсора BH1750:

- діапазон вимірювання: від 1 люксів до 65535 люксів;
- інтерфейс обміну даними: I2C;
- робоча напруга живлення: від 2,4 В до 3,6 В (можлива робота від 5 В завдяки вбудованому стабілізатору на деяких модулях);
- споживання струму: близько 0,12 мА під час вимірювання;

- вбудовані режими чутливості: H-Resolution Mode, H-Resolution Mode2, L-Resolution Mode;

- час вимірювання: залежить від режиму (від 16 мс до 120 мс);

- компактні розміри та низьке енергоспоживання.

Сенсор BH1750 працює в безперервному режимі вимірювання, що дозволяє постійно отримувати актуальні дані про освітлення без потреби складної обробки на контролері. Це значно полегшує створення автоматичних сценаріїв у Node-RED.

Принцип функціонування сенсора BH1750.

Датчик перетворює інтенсивність світла на цифрове значення, яке передається контролеру через інтерфейс I2C. Завдяки високій точності та широкому динамічному діапазону, BH1750 добре працює як при слабкому, так і при яскравому сонячному світлі.

Вибір робочого режиму датчика дозволяє оптимально балансувати між швидкістю вимірювання та точністю. Наприклад:

- H-Resolution Mode – висока точність, час вимірювання близько 120 мс;

- H-Resolution Mode2 – подвійна точність, триваліше вимірювання;

- L-Resolution Mode – менша точність, швидкий відгук (приблизно 16 мс).

Переваги BH1750 у системі автоматизації освітлення:

- висока точність та стабільність вимірювань;

- простота інтеграції через стандартний протокол I2C;

- низьке енергоспоживання, що дозволяє довго працювати автономно від батарей;

- наявність великої кількості бібліотек для Raspberry Pi та Node-RED;

- компактність та можливість встановлення в будь-яких умовах.

Отже, використання датчика BH1750 в автоматизації освітлення забезпечує точне визначення рівня зовнішнього освітлення, що дозволяє ефективно налаштувати автоматичне ввімкнення та вимкнення світильників.

1.6 Модуль реле: принцип дії, особливості

Модулі реле широко застосовуються у системах автоматизації для керування електричними навантаженнями, зокрема для комутації освітлювальних приладів. Вони дозволяють з невеликого сигналу управління, який генерує мікроконтролер або одноплатний комп'ютер, замикати або розмикати силові кола змінного або постійного струму.

У межах даної роботи використовується стандартний одноканальний реле-модуль, призначений для керування освітлювальними пристроями через інтерфейс Raspberry Pi.

Основні характеристики реле-модуля:

- напруга керування: 5 В постійного струму;
- струм спрацювання: до 70 мА;
- тип реле: електромеханічне, з нормально відкритими (NO) та нормально закритими (NC) контактами;
- максимальні параметри комутації: до 250 В змінного струму при струмі до 10 А або до 30 В постійного струму при струмі до 10 А;
- інтерфейс керування: цифровий вхід (активний низький або активний високий рівень залежно від версії модуля);
- захисні елементи: оптоізоляція для електричного відділення силової частини від логічної.

Принцип роботи реле-модуля.

Модуль працює за наступною схемою: при подачі логічного сигналу керування від Raspberry Pi активується внутрішній транзистор модуля, який замикає ланцюг живлення обмотки реле. Під дією електромагнітного поля відбувається перемикування контактів реле – вмикається або вимикається навантаження.

Важливою особливістю є наявність оптоізолятора у деяких версіях модулів, що дозволяє ізолювати логічні кола керування від силових ланцюгів, тим самим підвищуючи безпеку системи.

Переваги використання реле-модуля у системі автоматизації освітлення:

- простота інтеграції з мікроконтролерами та одноплатними комп'ютерами завдяки стандартному цифровому інтерфейсу;
- можливість керування освітлювальними приладами великої потужності безпосередньо з малопотужних виходів контролера;
- високий рівень надійності при правильному підключенні;
- наявність оптоізоляції для захисту апаратної частини системи;
- відносно низька вартість та велика доступність на ринку.

Реле-модуль (рис. 1.4) забезпечує необхідний рівень безпеки та надійності для побудови систем автоматичного керування освітленням. Використання даного модуля у поєднанні з Raspberry Pi та Node-RED дозволяє реалізувати як прості сценарії керування вручну, так і складні багаторівневі логіки автоматизації залежно від даних з сенсорів.

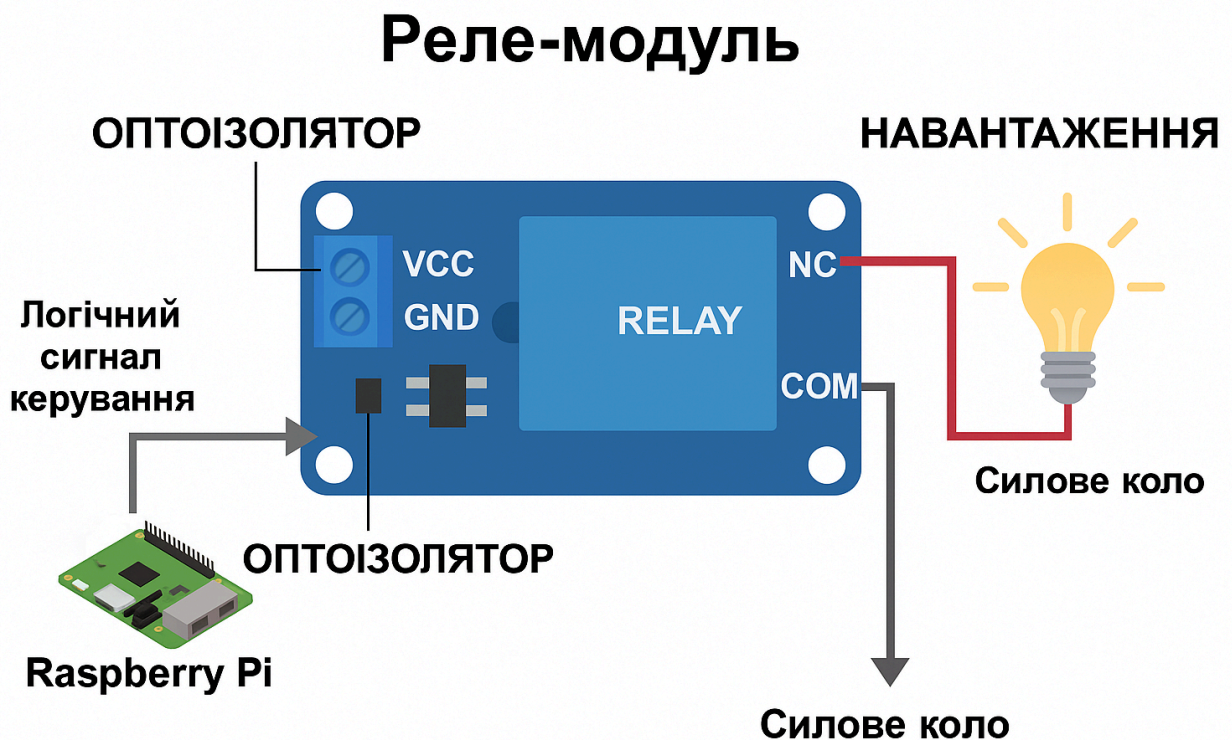


Рисунок 1.4 – Проста схема реле-модуля

1.7 Протоколи зв'язку в IoT-системах: I2C, MQTT

Одним з ключових аспектів при розробці систем автоматизації є налагодження ефективної взаємодії між окремими модулями та пристроями. В системах Інтернету речей (IoT) широко використовуються різні протоколи обміну даних, котрі дозволяють збирати, передавати та обробляти інформацію в реальному часі. В межах даної роботи основними протоколами, що використовуються, є I2C та MQTT.

Інтерфейс I2C.

I2C (Inter-Integrated Circuit) – це синхронний послідовний протокол, розроблений для швидкого обміну даними між мікроконтролерами та периферійними пристроями.

Основні характеристики I2C:

- обмін даними відбувається через дві лінії: SCL (Serial Clock Line) та SDA (Serial Data Line);
- підтримка декількох головних (master) і підлеглих (slave) пристроїв на одній шині;
- використовується в системах з невеликими відстанями між компонентами та невисокою швидкістю передачі даних;
- простота підключення завдяки використанню лише двох сигналів плюс загального провідника (GND);
- адресація пристроїв за допомогою унікальних ідентифікаторів (зазвичай 7-бітові або 10-бітові адреси).

У даній роботі інтерфейс I2C використовується для підключення датчика освітленості BH1750 до Raspberry Pi. Використання I2C забезпечує стабільний та швидкий обмін даними з мінімальною кількістю фізичних з'єднань.

Протокол MQTT.

MQTT (Message Queuing Telemetry Transport) – легкий мережевий протокол, що працює за принципом публікації-підписки, використовується для передачі повідомлень між пристроями в середовищах з обмеженими ресурсами

та ненадійним зв'язком.

Основні властивості MQTT:

- архітектура базується на моделі "брокер-клієнт", де всі повідомлення передаються через центральний брокер;
- підтримує механізм підписки на теми (topics) та публікації повідомлень;
- мінімальне використання пропускну здатності мережі завдяки невеликому розміру повідомлень;
- підтримка різних рівнів гарантії доставки повідомлень (QoS 0, 1, 2);
- простота реалізації на вбудованих системах та у великих розподілених IoT-мережах.

Хоча в даному проєкті MQTT не є основним інструментом для взаємодії всередині однієї локальної системи (оскільки управління здійснюється безпосередньо через Node-RED), його використання перспективне при розширенні системи на декілька пристроїв або інтеграції з хмарними платформами для дистанційного керування.

Переваги застосування протоколів I2C та MQTT:

- спрощення апаратного та програмного забезпечення системи;
- зменшення витрат на розробку та розгортання мережі IoT-пристроїв;
- можливість гнучкого масштабування системи без зміни базової архітектури;
- надійність передачі даних навіть за нестабільного з'єднання або високих вимог до енергоспоживання.

Таким чином, правильний вибір та інтеграція протоколів I2C та MQTT забезпечують ефективну та безпечну роботу системи автоматизації освітлення в розроблюваному розумному будинку.

1.8 Вибір логіки автоматизації в Node-RED

У системах автоматизації освітлення ключовим аспектом є розробка коректної логіки роботи, яка базується на даних з сенсорів, часових параметрах,

умовах довкілля та командах користувача. Платформа Node-RED, задіяна в даному проєкті, дозволяє ефективно реалізовувати сценарії автоматизації, застосовуючи побудову потоків даних та обробку подій в режимі реального часу.

Основні принципи вибору логіки автоматизації в Node-RED включають:

- застосування потокової обробки даних, де кожний вузол відповідає за конкретну функцію (отримання даних, обробку, прийняття рішення, керування виконавчим пристроєм);
- реалізація простих умовних операторів для керування освітленням на основі даних з датчика освітленості;
- використання таймерів для затримки увімкнення або вимкнення освітлення за визначеним графіком чи часовими умовами;
- створення інтерфейсу користувача через Node-RED Dashboard для ручного керування освітленням з можливістю пріоритетності ручного режиму над автоматичним;
- реалізація функцій аварійного відключення освітлення у випадку несправностей датчика або втрати зв'язку.

Основні компоненти логіки автоматизації:

- вхідні вузли: отримання даних з датчика BH1750 через інтерфейс I2C;
- обробка даних: порівняння отриманого рівня освітленості з заданими пороговими значеннями;
- прийняття рішення: визначення необхідності увімкнення або вимкнення світильників в залежності від порогу;
- виконавчі вузли: передача керуючого сигналу на вихід Raspberry Pi для активації реле-модуля;
- інтерфейс користувача: створення панелі управління з кнопками для ручного керування світлом та відображення стану системи.

Для забезпечення належної роботи системи передбачено також збереження останнього стану пристроїв при перезавантаженні Raspberry Pi та налаштування повторного встановлення зв'язку у разі втрати з'єднання з

сенсором.

Сценарії автоматизації у системі:

- автоматичне вмикання освітлення при падінні рівня освітленості нижче встановленого порогу;
- автоматичне вимикання освітлення при досягненні заданого рівня природного освітлення;
- можливість ручного управління світильниками через веб-інтерфейс, незалежно від поточних показників освітленості;
- встановлення тимчасових режимів (наприклад, режим «нічного освітлення» зі зниженою яскравістю або обмеженим часом роботи);
- аварійне вимикання освітлення у разі втрати з'єднання з датчиком.

Node-RED забезпечує гнучке налаштування та модифікацію логіки роботи системи без необхідності перепрограмування усієї системи, що є суттєвою перевагою при розробці адаптивних та масштабованих рішень для розумного будинку.

1.9 Структурна схема системи

Для наочної демонстрації взаємодії складових системи автоматизованого освітлення була створена структурна схема, яка ілюструє принцип побудови та обміну інформацією між вузлами.

Структурна схема містить такі ключові компоненти:

- датчик світла BH1750, що проводить вимірювання рівня освітленості в кімнаті та надсилає ці дані через інтерфейс I2C;
- одноплатний комп'ютер Raspberry Pi, який виконує задачі збирання, обробки та аналізу даних з сенсора, а також приймає рішення щодо управління світлом;
- програмна платформа Node-RED, встановлена на Raspberry Pi, що відповідає за обробку даних, керування релейним модулем і створення веб-інтерфейсу для користувача;

- релейний модуль, з'єднаний з Raspberry Pi через GPIO-виводи, який фізично замикає або розмикає електричне коло освітлення;
- світильник або інший освітлювальний пристрій, що підключається до реле та управляється відповідно до команд системи;
- користувач, який має можливість регулювати освітлення через веб-інтерфейс Node-RED Dashboard з будь-якого пристрою, що має доступ до локальної мережі або Інтернету.

Отже, всі елементи системи співпрацюють між собою у єдиній інформаційно-керуючій мережі, що забезпечує як автоматичне, так і ручне регулювання освітленням.

Умовна структурна схема системи представлена на рисунку 2.5.

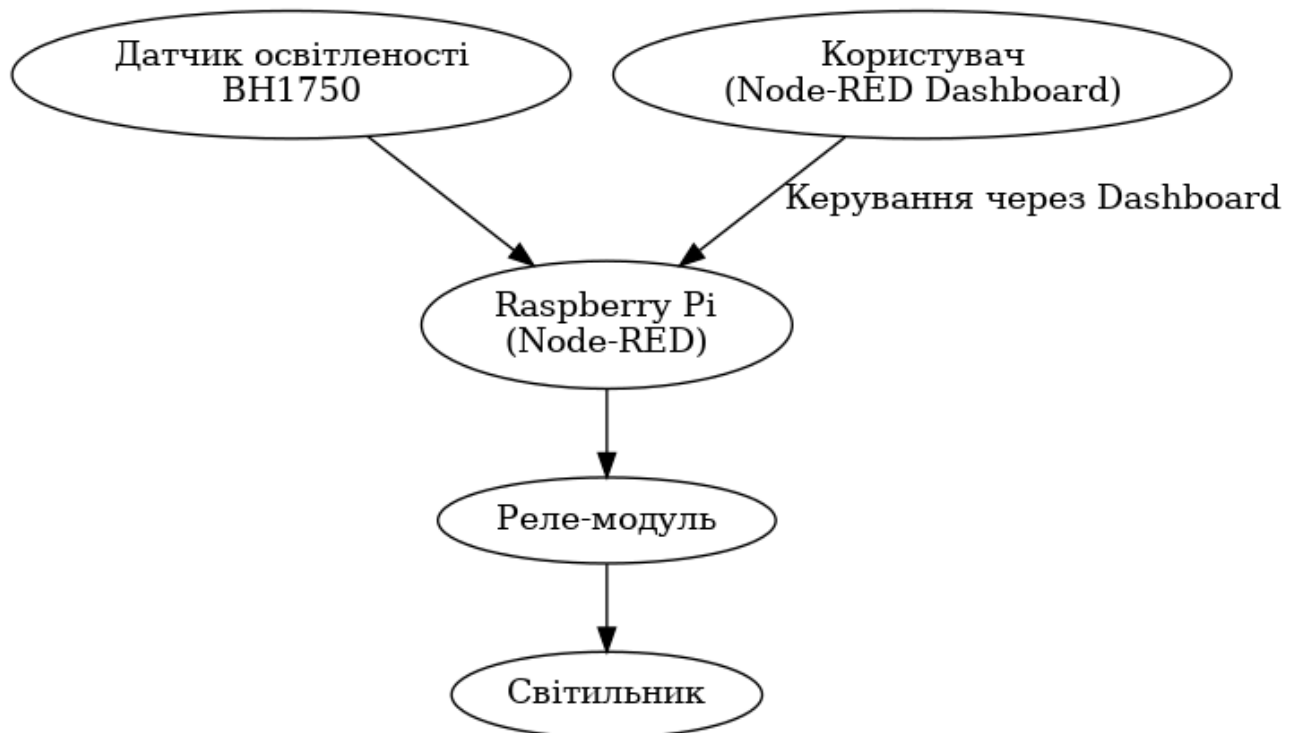


Рисунок 1.5 – Структурна схема системи

1.10 Приклад сценарію автоматизації

Для демонстрації роботи автоматизованої системи освітлення, яку було розроблено, пропонується типовий сценарій, реалізований на базі Node-RED.

Сценарій реалізує автоматичне керування включенням та виключенням

освітлення, що залежить від інтенсивності природного світла у приміщенні. Одночасно з цим, користувач має можливість у будь-який час вручну змінювати стан освітлення, використовуючи веб-інтерфейс Node-RED Dashboard.

Умови, за яких працює сценарій:

- коли рівень освітленості, визначений сенсором BH1750, опускається нижче визначеного порогу (наприклад, 200 люкс), система автоматично активує освітлення;
- якщо рівень освітленості перевищує встановлений поріг (наприклад, 300 люкс), система автоматично вимикає світло;
- користувач може примусово включати чи виключати світло за допомогою веб-інтерфейсу, незалежно від показань датчика;
- при примусовому управлінні автоматичне регулювання тимчасово припиняється до наступного циклу перевірки.

Етапи виконання сценарію:

- а) датчик BH1750 постійно надсилає інформацію про поточну освітленість на Raspberry Pi через інтерфейс I2C;
- б) Node-RED отримує ці дані, обробляє їх за допомогою функціональних вузлів і порівнює з заданим пороговим значенням;
- в) якщо рівень освітленості не відповідає встановленим умовам, система створює керуючий сигнал;
- г) Raspberry Pi, використовуючи вихід GPIO, активує або деактивує релейний модуль, що, відповідно, включає або вимикає світильник;
- д) користувач має змогу спостерігати за поточним станом освітлення через веб-інтерфейс та здійснювати ручне управління.

Переваги представленого сценарію (рис. 1.6):

- автоматичне регулювання освітлення з врахуванням реальних умов оточення;
- економія енергії завдяки відключенню освітлення при достатньому рівні природного світла;
- збільшення комфорту користувача завдяки можливості ручного

керування системою;

– гнучкість налаштування порогових значень і поведінки системи без потреби в зміні апаратної частини.

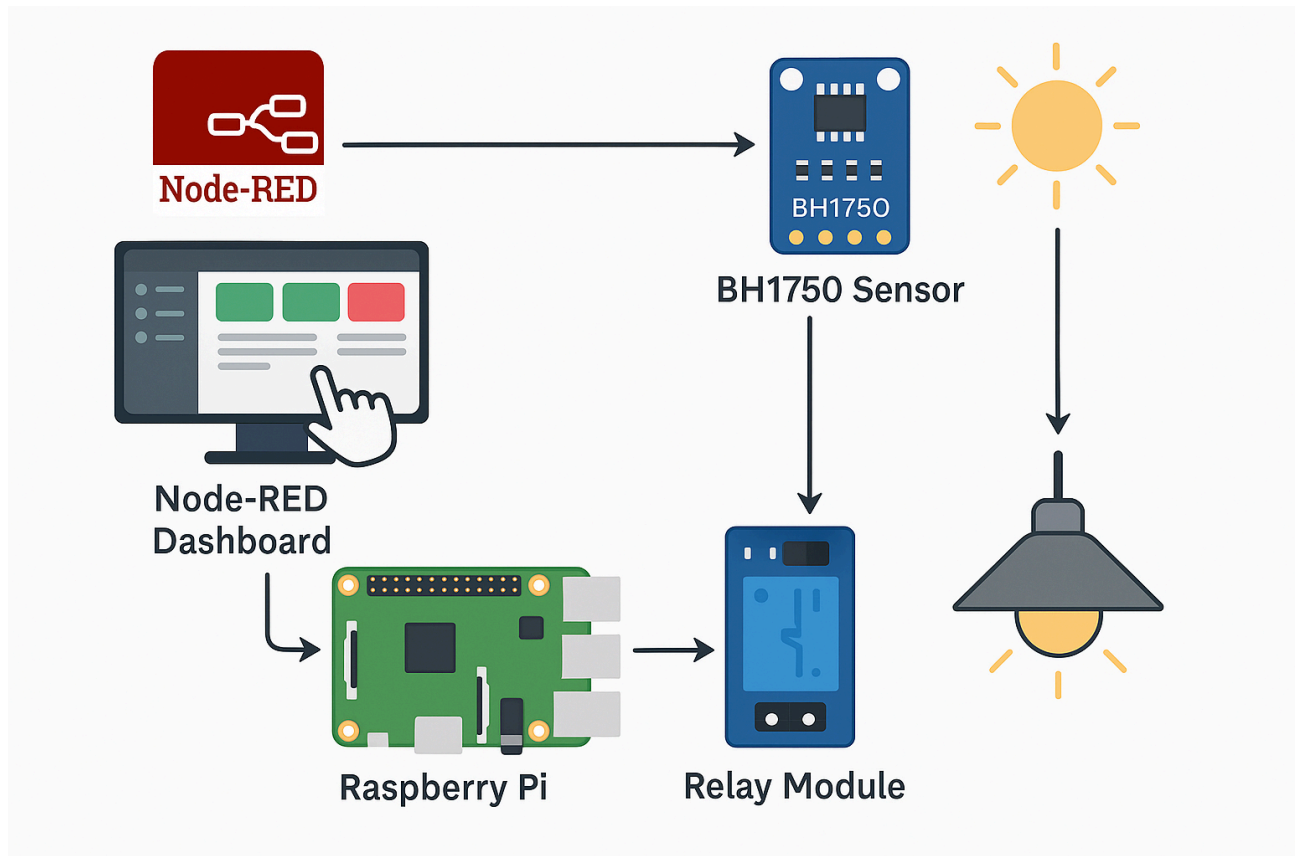


Рисунок 1.6 – Приклад сценарію системи

Таким чином, розроблений сценарій автоматизації дозволяє забезпечити оптимальне керування освітленням у реальному часі відповідно до змін зовнішніх умов та побажань користувача.

1.11 Потенціал масштабування системи

Розроблювана система автоматизації освітлення на базі Raspberry Pi та Node-RED відкриває широкі перспективи для подальшого розширення і масштабування. Архітектура системи спроектована таким чином, що дозволяє

легко збільшувати її функціональність, додаючи нові пристрої, сенсори та логічні блоки, не змінюючи суттєво основні принципи роботи.

Основні шляхи масштабування системи включають:

- збільшення кількості контрольованих освітлювальних зон через підключення додаткових датчиків світла та реле;
- інтеграцію сенсорів руху для автоматичного ввімкнення світла при виявленні присутності людей;
- додавання функції регулювання інтенсивності освітлення за допомогою керованих драйверів або світлодіодних димерів;
- використання додаткових сенсорів довкілля (температури, вологості, якості повітря) для створення складних сценаріїв комфорту;
- реалізацію централізованого керування всіма освітлювальними зонами через єдиний веб-інтерфейс Node-RED Dashboard;
- впровадження віддаленого моніторингу та керування системою через Інтернет з використанням протоколу MQTT та хмарних серверів;
- інтеграцію з голосовими помічниками (Google Assistant, Amazon Alexa) для голосового керування освітленням;
- використання баз даних для збереження історії подій, рівнів освітленості та стану системи з можливістю подальшого аналізу.

Завдяки відкритості програмного середовища Node-RED та широкій підтримці апаратної платформи Raspberry Pi, реалізація вказаних функціональних розширень не потребує значних інженерних зусиль або переробки базової системи.

Масштабування системи дозволить не лише збільшити рівень автоматизації житлового простору, але й забезпечити гнучке налаштування відповідно до індивідуальних потреб користувача, що є важливим аспектом сучасних розумних будинків.

Отже, запропонована система має значний потенціал для подальшого розвитку та інтеграції в комплексні рішення автоматизації побутових процесів.

1.12 Висновки по технічному завданню

Внаслідок проведеного дослідження технічного завдання було встановлено ключові вимоги до розробки системи автоматизації освітлення на основі Raspberry Pi та Node-RED.

Обґрунтовано вибір апаратної платформи Raspberry Pi як центрального вузла системи, що забезпечує необхідну обчислювальну потужність, стійкість та гнучкість. Проаналізовано специфіку застосування сенсора освітленості BH1750 для точного вимірювання інтенсивності світла в приміщенні, а також реле-модуля для комутації електричних кіл освітлювальних приладів.

Проаналізовано використання протоколів обміну даними I2C для локальної взаємодії між пристроями та MQTT як перспективного варіанту для подальшої інтеграції системи в розподілені мережі Інтернету речей. Описано принципи побудови логіки автоматизації з використанням інструменту Node-RED, що дає змогу реалізувати сценарії автоматичного та ручного керування освітленням через зручний веб-інтерфейс.

Структурна схема системи ілюструє взаємозв'язок ключових апаратних та програмних компонентів і визначає логіку їхньої взаємодії.

Окремо було розглянуто приклад сценарію автоматизації, який дозволяє регулювати роботу освітлення в залежності від рівня природного освітлення та потреб користувача. Також було проаналізовано потенціал масштабування системи шляхом інтеграції нових сенсорів, розширення зон управління та впровадження віддаленого доступу.

Отже, проведений аналіз підтвердив актуальність розробки системи автоматизації освітлення на основі Node-RED та Raspberry Pi, визначив ключові технічні вимоги та заклав основу для подальшої реалізації та вдосконалення даного рішення.

2 РОЗРОБЛЕННЯ СТРУКТУРИ СИСТЕМИ АВТОМАТИЗАЦІЇ ОСВІТЛЕННЯ

2.1 Загальна структура програмного забезпечення

Node-RED використовує концепцію потоків (flows), у яких інформація передається від одного вузла до іншого. Для реалізації системи автоматизації освітлення було створено окремий потік, що обробляє дані з датчика освітленості BH1750, визначає стан освітлення і формує відповідні керуючі сигнали.

У межах цього потоку використовуються такі типи вузлів:

- Inject (Ініціалізація) – періодично запускає цикл збору даних;
- Function (Обробка логіки) – обробляє значення освітленості, приймає рішення;
- I2C input (Зчитування з BH1750) – отримує значення рівня освітленості з сенсора;
- GPIO (Керування реле) – вмикає або вимикає реле через вихід Raspberry Pi;
- Dashboard nodes (UI вузли) – створюють веб-інтерфейс для ручного керування та відображення стану системи.

Весь функціонал реалізований у Node-RED у вигляді візуального потоку, який працює у циклі з інтервалом 1–2 секунди.

2.2 Логічна структура потоку Node-RED

Логічна архітектура потоку в Node-RED ґрунтується на парадигмі подій, де кожен вузол виконує власну роль у загальній системі автоматизації освітлення. Потік обробки даних будується з послідовно з'єднаних вузлів, що забезпечують зчитування показників з датчика, прийняття рішення, керування виконавчим пристроєм та взаємодію з користувачем.

Послідовність виконання потоку.

Inject – створює таймер, який регулярно (наприклад, кожні 2 секунди) ініціює запуск процесу зчитування даних з датчика освітленості. Цей вузол гарантує безперервну роботу системи.

Function: Зчитування BH1750 – використовує скрипт або готовий вузол для зчитування значення освітленості з датчика BH1750 через інтерфейс I2C. Отримане значення передається далі у потоці як змінна `msg.payload`.

Function: Аналіз значення освітленості – порівнює рівень освітленості із заданим порогом (наприклад, 200 люкс). Генерує керуючий сигнал (1 або 0) в залежності від того, чи потрібно ввімкнути або вимкнути освітлення.

GPIO out – отримує керуючий сигнал від попереднього вузла та передає його на відповідний вихід Raspberry Pi, до якого підключено реле. Відповідно до сигналу, реле замикає або розмикає електричне коло освітлення.

UI Switch (ручне керування) – відображає перемикач на веб-інтерфейсі Node-RED Dashboard. Дає змогу користувачеві вмикати або вимикати освітлення вручну за потреби.

UI Text / Gauge (візуалізація даних) – показує поточне значення освітленості, стан світильника та інші дані в режимі реального часу на панелі керування користувача.

Logic merge (пріоритети управління) – додатковий вузол, що об'єднує автоматичне та ручне управління, наприклад, тимчасово відключає автоматичний режим при ручному втручанні.

Особливості реалізації:

- усі вузли об'єднані в компактний та зрозумілий потік;
- потік працює циклічно, не потребуючи втручання користувача;
- використані вузли Node-RED Dashboard забезпечують зручний доступ до керування з будь-якого пристрою в локальній мережі.

Така структура дозволяє реалізувати адаптивне та масштабоване керування освітленням, що легко розширюється у разі додавання нових зон або

пристроїв.

2.3 Веб-інтерфейс користувача (Node-RED Dashboard)

Node-RED Dashboard – це вмонтований модуль в платформу Node-RED, що дозволяє будувати графічний інтерфейс користувача для візуалізації даних та управління системою в реальному часі. У рамках даної розробки веб-інтерфейс відіграє важливу роль, забезпечуючи зручне та інтуїтивно зрозуміле управління освітленням.

Інтерфейс доступний через веб-браузер з будь-якого пристрою, з'єднаного з тією ж мережею, що й Raspberry Pi, і не вимагає встановлення додаткового програмного забезпечення.

Основні елементи інтерфейсу:

- цифровий індикатор рівня освітленості – демонструє актуальні значення, отримані з датчика BH1750;
- стан реле – інформує користувача про те, чи включене на даний момент освітлення;
- перемикач режиму управління – дозволяє перемикатися між автоматичним і ручним режимом;
- кнопки ручного вмикання / вимикання – забезпечують пряме управління реле незалежно від рівня освітленості;
- графіки або історія значень (за потреби) – можуть бути реалізовані для відображення змін рівня освітленості протягом певного часу.

Функціональні можливості інтерфейсу:

- негайна реакція на дії користувача без потреби перезавантаження сторінки;
- відображення поточного стану системи та її основних параметрів;
- можливість перегляду та управління з комп'ютера, планшета чи смартфона;
- доступна локально або через тунель (наприклад, з використанням

сервісів ngrok або VPN) для віддаленого доступу.

Node-RED Dashboard реалізується за допомогою спеціальних вузлів типу `ui_switch`, `ui_text`, `ui_gauge` та інших, які додаються до основного потоку. Розробка інтерфейсу не вимагає знань HTML чи CSS, що значно спрощує створення зручного середовища управління.

Застосування веб-інтерфейсу в системі автоматизації освітлення значно збільшує її функціональність та зручність для кінцевого користувача, дозволяючи легко перемикати режими, змінювати параметри або одержувати зворотній зв'язок щодо роботи системи.

3 ПРАКТИЧНА ЧАСТИНА

3.1 Установка та налаштування Raspberry Pi

Для реалізації системи автоматизації на базі Node-RED було використано одноплатний комп'ютер Raspberry Pi 3 Model B+. Це доступний, енергоефективний та функціональний пристрій, який підтримує роботу з різноманітними сенсорами, протоколами зв'язку та відкритими програмними платформами.

Підготовка Raspberry Pi до роботи. Вибір операційної системи.

Для роботи було обрано офіційну операційну систему Raspberry Pi OS (Lite) – легку версію Debian, яка не містить графічного інтерфейсу, але забезпечує стабільну роботу сервісів у фоновому режимі.

Завантаження образу: <https://www.raspberrypi.com/software/operating-systems/>

Запис образу на карту пам'яті:

Образ ОС записується на microSD карту (не менше 8 ГБ, краще 16–32 ГБ) за допомогою утиліти Raspberry Pi Imager або balenaEtcher.

Перший запуск та базові налаштування:

Після встановлення та запуску необхідно виконати базову конфігурацію:

- підключитися до Raspberry Pi по SSH або через HDMI + клавіатура;
- запустити конфігуратор командою:

```
sudo raspi-config;
```

- увімкнути I2C у розділі Interface Options/I2C;
- налаштувати Wi-Fi у розділі System Options/Wireless LAN;
- за потреби змінити мову, час і розкладку клавіатури.

Оновлення системи.

Після першого запуску рекомендується оновити систему:

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

Встановлення додаткових утиліт.

Для роботи з Node-RED та сенсорами рекомендовано встановити:

- Git – для керування конфігурацією;
- I2C-tools – для перевірки підключення сенсорів;
- Python 3 (якщо планується робота з Python-скриптами).

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

Для перевірки наявності підключених I2C пристроїв (наприклад, BH1750) використовують команду:

```
i2cdetect -y 1
```

Таким чином, Raspberry Pi повністю підготовлено для встановлення Node-RED та реалізації логіки автоматизації.

3.2 Встановлення та запуск Node-RED

Node-RED – це інструмент потокового програмування, який ідеально підходить для створення систем автоматизації. Платформа розроблена на базі Node.js та має веб-інтерфейс для побудови логіки керування пристроями, обробки даних з сенсорів та взаємодії з користувачем.

У рамках даного проекту Node-RED встановлюється безпосередньо на Raspberry Pi, що дозволяє використовувати його ресурси для локального оброблення даних і керування периферією.

Встановлення Node.js.

Перед встановленням Node-RED необхідно інстальювати середовище виконання Node.js. Найпростіший спосіб – через офіційний інсталяційний скрипт:

```
sudo apt install -y nodejs npm
```

Альтернативно (рекомендовано Node-RED спільнотою):

```
bash <(curl -sL https://deb.nodesource.com/setup_18.x)
```

```
sudo apt install -y nodejs
```

Встановлення Node-RED.

Node-RED встановлюється глобально через менеджер пакетів npm:

```
sudo npm install -g --unsafe-perm node-red
```

Після інсталяції перевіряється доступність команди:

```
node-red --version
```

Запуск Node-RED.

Перший запуск виконується так:

```
node-red
```

Після цього в консолі буде виведено URL-адресу, за якою доступна веб-панель Node-RED, наприклад <http://192.168.1.100:1880>.

Цю адресу можна відкрити в браузері зі смартфона або комп'ютера, підключеного до тієї ж локальної мережі.

Автоматичний запуск Node-RED при включенні Raspberry Pi.

Щоб система запускалася автоматично, Node-RED можна зареєструвати як сервіс:

```
sudo systemctl enable nodered.service
```

Для керування:

```
sudo systemctl start nodered
```

```
sudo systemctl status nodered
```

3.3 Підключення сенсора BH1750 до Raspberry Pi

Сенсор BH1750 є цифровим датчиком освітленості, що вимірює рівень освітлення в одиницях люкс. Завдяки підтримці протоколу I2C, він легко інтегрується з Raspberry Pi без потреби у додаткових компонентах.

Схема підключення.

Сенсор BH1750 має чотири основні контакти:

- VCC – живлення (+3,3 В або +5 В);
- GND – земля;
- SCL – тактова лінія I2C;
- SDA – лінія даних I2C.

Примітка: для більшості модулів BH1750 допускається живлення як від

3,3 В, так і від 5 В.

Активація інтерфейсу I2C.

Щоб увімкнути I2C на Raspberry Pi, використовується конфігуратор:

```
sudo raspi-config
```

Далі: Interface Options/I2C/Enable.

Після перезавантаження перевіряється наявність підключеного пристрою:

```
i2cdetect -y 1
```

У виводі має з'явитися адреса пристрою (наприклад, 23 с або 5 с), що підтверджує правильне підключення сенсора.

Встановлення бібліотеки BH1750 (для Node-RED).

Щоб Node-RED міг зчитувати дані з сенсора, використовуються спеціальні бібліотеки або власні функціональні вузли.

Встановлення i2c-bus:

```
cd ~/.node-red  
npm install i2c-bus
```

Встановлення bh1750 (готова npm-бібліотека):

```
npm install bh1750
```

Або можна використовувати вузол node-red-contrib-i2c, який додається через Palette Manager у Node-RED.

Таким чином, після підключення, активації I2C і встановлення потрібних бібліотек, Raspberry Pi може повноцінно отримувати дані з сенсора освітленості BH1750 для подальшої обробки у Node-RED.

3.5 Веб-інтерфейс користувача: створення та налаштування

Node-RED має вбудований модуль Dashboard, що може створювати простий веб-інтерфейс для моніторингу та керування пристроями. Це дуже зручно для систем автоматизації, де користувач має бачити стан сенсорів і мати можливість вручну вмикати і вимикати обладнання.

Встановлення Dashboard виконується через меню «Manage palette»:

У веб-інтерфейсі Node-RED: меню → Manage palette.

У вкладці Install знайти node-red-dashboard,

Натиснути Install, або через термінал:

```
cd ~/.node-red
npm install node-red-dashboard
```

Після встановлення з'являються нові вузли в категорії dashboard.

Створення елементів інтерфейсу.

Інтерфейс складається з таких елементів:

- ui_text / ui_gauge – відображення значення освітленості;
- ui_switch – перемикач ручного керування світлом;
- ui_button (за потреби) – пряме вмикання/вимикання;
- ui_group / ui_tab – групування елементів по вкладках.

Приклад:

Вкладка: Освітлення.

Група: Вітальня.

Елементи: Освітленість (графік), Перемикач, Стан реле.

Кожен вузол має поля для назви, кольору, мін/макс значення (наприклад, 0–1000 люкс), а також налаштування вигляду інтерфейсу.

Доступ до інтерфейсу.

Після розгортання Dashboard буде доступний за адресою:

<http://<IP-адреса Raspberry Pi>:1880/ui>

Наприклад:

<http://192.168.1.100:1880/ui>

Його можна відкрити з будь-якого пристрою в локальній мережі – смартфона, планшета або комп'ютера.

Приклад інтерфейсу:

- поточна освітленість: 250 люкс;
- стан світла: Ввімкнено;
- перемикач ручного режиму: [ON/OFF];
- кнопка: "Увімкнути примусово".

Інтерфейс динамічно оновлюється й відображає реальні дані, отримані з сенсора ВН1750 та стан реле.

3.6 Ручне та автоматичне керування світлом

Система автоматизації освітлення має забезпечувати не лише автономну роботу, а й можливість ручного втручання користувача. Це особливо важливо в ситуаціях, коли автоматичний режим не враховує всіх обставин або користувач бажає тимчасово змінити поведінку системи.

Основна ідея:

- у звичайному режимі система керує освітленням автоматично на основі даних з датчика ВН1750;
- користувач має змогу вручну змінити стан світла через веб-інтерфейс;
- після ручного втручання автоматичний режим тимчасово блокується, щоб уникнути конфліктів;

Реалізація логіки в Node-RED.

Використані вузли:

- `ui_switch` – елемент керування на панелі Dashboard;
- `function` – блок логіки для обробки ручного вводу;
- `rbe (report by exception)` – запобігає повторному надсиланню однакових значень;
- `change` – очищення прапора ручного втручання після тайм-ауту;
- `delay` – затримка повернення до автоматичного режиму.

Алгоритм роботи.

Кожні 2 секунди автоматичний сценарій перевіряє рівень освітленості й формує команду для реле.

Якщо користувач вручну перемикає стан світла:

- система запам'ятовує цей факт (встановлюється змінна `manualOverride = true`);
- автоматичне керування блокується.

Через певний час (наприклад, 5 хвилин) система автоматично повертається в автоматичний режим

Код функції (приклад):

```
if (flow.get("manualOverride")) {  
    return null; // блокуємо авто  
}  
const lux = msg.payload;  
if (lux < 200) {  
    msg.payload = 1;  
} else if (lux > 300) {  
    msg.payload = 0;  
} else {  
    return null;  
}  
return msg;
```

Коли користувач перемикає вручну:

```
flow.set("manualOverride", true);  
return msg;
```

Таймер очищує прапор:

```
flow.set("manualOverride", false);
```

Переваги такої реалізації:

- користувач завжди має контроль над системою;
- автоматичне керування не перешкоджає діям людини;
- після втручання автоматизація відновлюється автоматично, не потребуючи повторного налаштування.

Такий підхід дозволяє поєднати зручність автоматизації з гнучкістю ручного втручання – як у класичних комерційних рішеннях "розумного дому".

4 РОЗШИРЕННЯ ТА МАСШТАБУВАННЯ СИСТЕМИ

4.1 Підключення кількох зон освітлення

На попередньому етапі було втілено автоматизоване управління однією ділянкою освітлення. Однак для повної реалізації системи розумного дому необхідно забезпечити можливість контролю кількох незалежних зон.

Приклади зон:

- вітальня;
- кухня;
- коридор;
- спальня.

Принцип масштабування.

Втілення кількох зон освітлення базується на принципі розширення логіки автоматизації для кожної зони окремо. Для кожної зони додаються:

- окремий сенсор (або спільний з уточненням логіки);
- окремий вузол керування реле (GPIO або бездротове реле);
- власний набір порогових значень;
- власні елементи інтерфейсу Dashboard.

В Node-RED реалізується:

- окремий потік або секція в потоці для кожної зони;
- всі вузли підписані з назвами зон (наприклад: "BH1750_кухня", "Реле_вітальня");
- інтерфейс користувача розділений на вкладки або групи.

Схематичне представлення:

```
[Inject] → [BH1750_вітальня] → [Логіка] → [Реле_вітальня]
      → [BH1750_кухня]   → [Логіка] → [Реле_кухня]
      → [BH1750_коридор] → [Логіка] → [Реле_коридор]
```

Варіанти реалізації:

- на одному Raspberry Pi – використовується кілька GPIO та сенсорів;
- з віддаленими модулями – з використанням ESP8266/ESP32 для кожної зони (MQTT, Wi-Fi).

Переваги масштабування:

- кожна зона функціонує незалежно, але під централізованим управлінням через єдину систему;
- система легко масштабується: інтеграція нової зони – це лише кілька нових вузлів у Node-RED;
- здатність розробляти сценарії, що враховують дані з декількох зон одночасно.

Такий підхід відкриває шлях від концепції "однієї лампочки" до повноцінного розумного дому з багатозонним освітленням, що адаптується до умов у кожному приміщенні індивідуально.

4.2 Використання бездротових реле (Wi-Fi / Bluetooth)

Для зручного масштабування системи освітлення та уникнення прокладки дротів до кожної зони можна використовувати бездротові реле. Найбільш поширеними варіантами є Wi-Fi реле на базі ESP8266/ESP32 або Bluetooth-реле.

Підхід на основі Wi-Fi (ESP8266/ESP32).

Мікроконтролери ESP8266 або ESP32 є доступними пристроями, які підтримують Wi-Fi та здатні керувати реле. Вони ідеально підходять для створення віддалених керованих модулів, які можна розмістити в будь-якому приміщенні.

Архітектура:

- Raspberry Pi з Node-RED працює як центральний сервер;
- ESP8266/ESP32 з прошивкою (наприклад, ESPHome або власна Arduino-програма) підключається до домашнього Wi-Fi;
- кожен ESP модуль підписується на MQTT-брокер та виконує команди з Node-RED;

- MQTT як канал зв'язку;
- Node-RED публікує команду: "room1/light/set" → "on";
- ESP приймає повідомлення та вмикає реле;
- зворотній зв'язок: ESP надсилає "room1/light/state" → "on" для підтвердження.

Приклад конфігурації (ESP8266 + MQTT).

Код для ESP на Arduino IDE:

```
#include <ESP8266WiFi.h>
```

```
#include <PubSubClient.h>
```

```
const char* ssid = "HomeWiFi";
```

```
const char* password = "password";
```

```
const char* mqtt_server = "192.168.1.100";
```

```
WiFiClient espClient;
```

```
PubSubClient client(espClient);
```

```
const int relayPin = D1;
```

```
void setup() {
```

```
  pinMode(relayPin, OUTPUT);
```

```
  digitalWrite(relayPin, LOW);
```

```
  WiFi.begin(ssid, password);
```

```
  client.setServer(mqtt_server, 1883);
```

```
  client.setCallback(callback);
```

```
  while (!client.connected()) {
```

```
    client.connect("esp8266_relay");
```

```
  }
```

```
  client.subscribe("room1/light/set");
```

```
}
```

```

void callback(char* topic, byte* payload, unsigned int length) {
  if (strcmp((char*)payload, "on", length) == 0) {
    digitalWrite(relayPin, HIGH);
  } else {
    digitalWrite(relayPin, LOW);
  }
}

void loop() {
  client.loop();
}

```

Альтернатива: Bluetooth-реле.

Bluetooth-реле теж присутні, але:

- вимагають з'єднання з Raspberry Pi;
- радіус їх дії менший;
- складніше конфігурувати, ніж Wi-Fi аналоги;
- не вміють працювати з централізованим керуванням через MQTT.

Отже, для надійної багатозонної системи Wi-Fi-реле на основі ESP – найпрактичніший варіант.

Плюси бездротового підключення:

- відсутність кабелів від Raspberry Pi до кімнат;
- легке додавання нових зон без потреби фізичного втручання;
- гнучкість розміщення модулів (в шафі, на стелі, за вимикачем, і т.д.);
- низька вартість модулів ESP8266 (від 3-5\$).

Таким чином, бездротові реле забезпечують можливість розширення системи автоматизації у повноцінний багатозонний розумний дім з централізованим керуванням через Node-RED.

4.3 Розподілена архітектура на ESP8266/ESP32 з MQTT

Для побудови масштабованої, стабільної та гнучкої системи автоматизації освітлення доцільно реалізувати розподілену архітектуру, в якій Raspberry Pi виконує функції центрального сервера, а ESP8266/ESP32 працюють як віддалені вузли керування реле.

Загальна схема взаємодії:

а) Raspberry Pi:

- 1) запущено Node-RED та MQTT-брокер (наприклад, Mosquitto);
- 2) приймає дані з сенсорів, формує логіку керування;
- 3) публікує команди для ESP-вузлів;

б) ESP8266 / ESP32:

- 1) підключені до локальної Wi-Fi мережі;
- 2) підписані на певні MQTT-топіки (topics);
- 3) отримують команду → вмикають/вимикають реле;
- 4) надсилають зворотний зв'язок (статус реле).

MQTT: основа обміну повідомленнями.

MQTT (Message Queuing Telemetry Transport) – легкий протокол обміну повідомленнями, ідеальний для IoT. Він працює за принципом "видавець – брокер – підписник".

Структура топіків (приклад):

- home/room1/light/set → команда на ввімкнення/вимкнення;
- home/room1/light/state → зворотній зв'язок від ESP;
- home/room1/lux → значення освітленості (опціонально, якщо є сенсор на ESP).

Переваги розподіленої архітектури:

- відмова однієї зони не зупиняє всю систему;
- проста заміна або додавання нових вузлів;
- мінімальне навантаження на Raspberry Pi;
- гнучка логіка керування – всі правила формуються централізовано;

– можливість розміщення вузлів у різних кімнатах чи навіть будівлях.

Рекомендоване програмне забезпечення.

На Raspberry Pi:

– Node-RED – основна логіка керування;

– Mosquitto MQTT Broker – приймає та передає повідомлення

```
sudo apt install mosquitto mosquitto-clients
```

На ESP8266/ESP32:

– Arduino IDE або ESPHome (як альтернатива без коду);

– підключення до Wi-Fi і підписка на MQTT.

Приклад конфігурації (Node-RED – MQTT – ESP):

У Node-RED:

a) mqtt out вузол:

1) topic: home/room1/light/set;

2) payload: "on" або "off".

На ESP:

– підписка на цей топик;

– при "on" – активується реле;

– зворотне повідомлення надсилається на home/room1/light/state.

Ця архітектура відкриває двері до побудови високоякісної, стійкої та здатної до зростання системи автоматизації. Вона слугуватиме надійним фундаментом не лише для керування освітленням, а й для інтеграції з іншими підсистемами, такими як опалення, безпека чи вентиляція.

4.4 Централізоване керування з Node-RED

Node-RED дозволяє об'єднати всі компоненти розумного будинку в одну єдину систему керування – незалежно від того, де фізично розташовані пристрої. За допомогою MQTT та Node-RED Dashboard реалізується централізоване керування всіма зонами освітлення та зворотний моніторинг їхнього стану.

Архітектура централізації:

- усі ESP-вузли (вітальня, кухня, спальня тощо) працюють незалежно;
- усі вони підключені до одного MQTT-брокера, який запущено на Raspberry Pi;
- Node-RED отримує дані від кожної зони та відображає їх на веб-інтерфейсі;
- користувач керує освітленням з одного місця: зі смартфона, планшета або ПК.

На панелі керування можна реалізувати:

- вкладки за кімнатами (вітальня, кухня, спальня);
- перемикачі (ui_switch) – для вмикання/вимикання кожної зони;
- індикатори стану (ui_led / ui_text) – показують, увімкнено чи ні;
- графіки освітленості (ui_chart) – історія показників сенсорів;
- автоматичні сценарії – наприклад, “нічний режим” або “економія енергії”.

Обробка даних в Node-RED:

- усі MQTT-повідомлення з ESP надходять у відповідні mqtt in вузли;
- дані відображаються в UI та можуть зберігатися у базу даних;
- у mqtt out вузлах користувач передає команди до кожної зони.

Приклад:

Topic: home/bedroom/light/set → payload: "on"

Node-RED відображає "home/bedroom/light/state" у вигляді індикатора

Приклад структури інтерфейсу:

а) вкладка "Вітальня":

- 1) освітленість: 180 люкс;
- 2) стан: Вимкнено;
- 3) перемикач: [Увімкнути].

б) вкладка "Кухня":

- 1) освітленість: 250 люкс;
- 2) стан: Увімкнено;

3) перемикач: [Вимкнути].

в) вкладка "Усі зони":

загальні режими: ніч / день / авто.

Переваги централізованого управління:

- єдина точка входу до всіх пристроїв;
- зручний та зрозумілий інтерфейс для кожного користувача;
- можливість масштабування без зміни поточної логіки чи структури;
- повний контроль та моніторинг в одному місці.

Такий підхід дозволяє користувачеві не тільки керувати освітленням, але й повністю бачити картину в домі, діяти вручну або покласти на автоматизацію.

4.5 Інтеграція з голосовими помічниками або мобільними застосунками

Інтеграція розумного будинку з голосовими асистентами (Google Assistant, Alexa) та мобільними додатками значно збільшує зручність використання системи. Користувач отримує змогу контролювати освітлення за допомогою голосу або з мобільного пристрою з будь-якої точки світу.

Варіанти інтеграції.

Інтеграція з Google Assistant через посередницькі платформи.

Найпростіший варіант – використовувати платформи, що підтримують Node-RED або MQTT.

Home Assistant – популярна система для розумного дому, має інтеграцію з Node-RED та Google Home.

IFTTT – дозволяє зв'язати Google Assistant з HTTP-запитами, які можна обробити в Node-RED.

Nora (Node-RED + Google Home) – спеціальний модуль для Node-RED з інтеграцією Google Smart Home.

Приклад інтеграції через IFTTT.

Створюється аплет "If you say: 'увімкни світло в кімнаті'"

Then → Webhook → надсилає HTTP POST на Node-RED (через ngrok або публічну IP-адресу)

Node-RED отримує запит та виконує необхідну дію: надсилає команду через MQTT

Наприклад:

→ "Ok Google, увімкни світло на кухні";

→ POST-запит;

→ Node-RED → MQTT → ESP8266 → реле.

Мобільні додатки для керування.

Node-RED Dashboard:

- веб-інтерфейс працює на будь-якому смартфоні через браузер;
- доступ через локальну мережу або через VPN/портфорвардинг;
- можна закріпити як іконку на головному екрані.

MQTT Dash (Android):

- додаток, що напямую підключається до MQTT-брокера;
- кнопки, індикатори, графіки;
- працює незалежно від Node-RED.

Home Assistant Companion App:

– повна інтеграція, якщо використовувати Home Assistant паралельно з Node-RED;

- push-повідомлення, автоматизація, геолокація.

Переваги інтеграції:

- голосове управління – зручно та сучасно;
- віддалений доступ – керування з будь-якого місця;
- автоматизація за подіями – наприклад, увімкнення світла при наближенні до будинку;
- використання звичних інструментів – Google Home, смартфон, голосові команди.

Інтеграція з голосовими помічниками та мобільними додатками дозволяє розширити функціональність системи, підвищити комфорт і відповідати

сучасним очікуванням користувачів. Система автоматизації освітлення перетворюється з простої логіки на інтелектуальне середовище, яким зручно керувати будь-де і будь-коли.

5 РЕАЛІЗАЦІЯ БЕЗДРОТОВОГО КЕРУВАННЯ ОСВІТЛЕННЯМ

5.1 Характеристика та вибір розумної розетки Gosund EP2

У процесі створення автоматизованої системи освітлення критично важливо зосереджуватися не лише на методах обробки інформації та алгоритмах управління, а й на типах виконавчих механізмів. Якщо у традиційній схемі управління освітленням використовуються дротові реле-модулі, то при розширенні системи та необхідності контролю над освітленням у декількох кімнатах, раціонально використовувати бездротові виконавчі пристрої.

Одним з найдоступніших та найзручніших рішень є застосування розумної Wi-Fi розетки Gosund EP2. Цей пристрій дає змогу вмикати чи вимикати підключене до нього електричне навантаження (скажімо, лампу) за допомогою мобільного додатку або дистанційно через локальну мережу.

Основні параметри розетки Gosund EP2:

- з'єднання через Wi-Fi (2,4 ГГц), без потреби в центральному хабі;
- максимальна потужність навантаження: до 3680 Вт / 16 А;
- компактні розміри – дозволяють використовувати кілька розеток одночасно;
- підтримка мобільних додатків Tuuya Smart та Smart Life;
- сумісність з Google Assistant та Amazon Alexa;
- можливість використання розкладу, таймера, або віддаленого керування.

Через відсутність фізичного інтерфейсу, підключення розетки до системи автоматизації Node-RED відбувається не напряму, а з використанням хмарних сервісів, як-от MQTT-шлюз (наприклад, tuuya-mqtt або через Home Assistant).

Додатковою перевагою даної розетки є її доступність на ринку без необхідності перепошивки. Деякі моделі Wi-Fi розеток потребують

заміни заводської прошивки на Tasmota або ESPHome для інтеграції в локальні системи, однак Gosund EP2 забезпечує управління навіть на стандартній прошивці через MQTT, що значно полегшує процес впровадження.

Відтак, вибір Gosund EP2 обґрунтований її функціональністю, доступністю, надійністю та простотою у використанні в контексті системи автоматизації на основі Node-RED.

5.2 Інтеграція розетки Gosund EP2 у систему автоматизації через MQTT

Для втілення бездротового контролю освітлення через Node-RED обрано протокол MQTT (Message Queuing Telemetry Transport), який є легким і широко застосовується в індустрії Інтернету речей (IoT). MQTT дозволяє ефективно організувати миттєвий обмін повідомленнями між пристроями.

Щоб інтегрувати розетку Gosund EP2 в систему, потрібно:

- з'єднати розетку з домашньою Wi-Fi мережею;
- застосувати шлюз MQTT (наприклад, `tuya-mqtt`), що встановлюється на Raspberry Pi або окремому сервері;
- здійснити авторизацію в `tuya-mqtt`, використовуючи обліковий запис TuYa, для доступу до пристроїв;
- встановити з'єднання Node-RED з локальним брокером MQTT (наприклад, Mosquitto);
- застосувати вузли Node-RED `'mqtt in'` та `'mqtt out'` для відправлення команд на розетку та отримання її статусу.

Сценарій функціонування системи:

- Node-RED отримує показник рівня освітленості від сенсора BH1750;
- у випадку, якщо освітленість опускається нижче встановленої межі,

Node-RED відправляє MQTT-повідомлення в топик розетки з командою `ON`;

– якщо освітленість перевищує поріг, надсилається команда `OFF`;

– користувач може у будь-який час керувати розеткою вручну через веб-інтерфейс Node-RED Dashboard.

Приклад MQTT-повідомлення:

```
{  
  "id": "розетка_вітальня",  
  "command": "switch",  
  "value": 1  
}
```

Такий підхід сприяє створенню масштабованої, безпечної та стабільної системи освітлення в кількох кімнатах, уникаючи провідних з'єднань між Raspberry Pi та виконавчими пристроями.

6 ПОТЕНЦІАЛ РОЗРОБЛЮВАНОЇ СИСТЕМИ І ПОРІВНЯННЯ З ІСНУЮЧИМИ РІШЕННЯМИ

6.1 Потенціал масштабування системи розумного будинку

Розроблена система управління освітленням, що базується на Raspberry Pi та Node-RED, демонструє лише початковий рівень автоматизації. Однак її архітектура спроектована з можливістю масштабування, що дозволяє інтегрувати нові можливості та розширювати систему відповідно до потреб конкретного користувача. У центрі розробки – відкритість, гнучкість та модульність, які є ключовими характеристиками сучасного підходу до створення IoT-систем.

Напрями розширення функціоналу:

- сенсори руху (PIR) – для автоматичного ввімкнення/вимкнення освітлення залежно від присутності людини;
- датчики температури і вологості (DHT22, BME280) – дозволяють створити інтелектуальну систему клімат-контролю;
- розумні жалюзі та штори – автоматичне регулювання світлового потоку протягом доби, враховуючи погодні умови;
- сенсори якості повітря (MQ-135, CO₂ датчики) – важливо для підтримки здорового мікроклімату в приміщеннях;
- сценарії поведінки (Node-RED Flows) – створення сценаріїв на кшталт «Ранок», «Ніч», «Я не вдома» з автоматичною зміною освітлення, температури тощо;
- інтеграція з сервісами сповіщень – надсилання повідомлень у Telegram, email або push-повідомлення при зміні стану пристроїв;
- голосове керування – взаємодія з Google Assistant, Alexa або офлайн-рішеннями на базі Raspberry Pi;
- камери відеоспостереження – інтеграція IP-камер у систему безпеки;
- багатокімнатна система – підключення додаткових датчиків у різних

кімнатах, як дротовим способом, так і через бездротові модулі (наприклад, ESP32 через MQTT).

Приклади логік, які можна реалізувати:

- ввімкнення світла у коридорі лише у темний час доби та при виявленні руху;
- зменшення яскравості освітлення вночі у спальні;
- автоматичне вимкнення всіх пристроїв при виході з дому;
- включення обігрівача, якщо температура нижче заданої межі.

Отже, система не є «закритою коробкою», як часто буває з готовими комерційними рішеннями. Її гнучка архітектура дозволяє користувачеві самостійно обирати обладнання, типи керування, способи обміну даними (дротові чи бездротові) та розширювати функціонал без додаткових витрат на підписки чи спеціалізовані хаби.

6.2 Порівняння з готовими рішеннями та конкурентні переваги

На сучасному ринку представлено безліч комерційних рішень для розумного дому, як-от Google Nest, Amazon Alexa, Philips Hue, Xiaomi Smart Home, TuYa Smart, Tapo, Aqara та інші. Вони пропонують готові до використання рішення для автоматизації простих задач: освітлення, безпека, контроль клімату. Однак, ці рішення мають як свої сильні сторони, так і значні обмеження.

Основні риси готових рішень:

- легке початкове встановлення та налаштування;
- власні мобільні додатки з зручним інтерфейсом;
- часто потребують постійного з'єднання з хмарними сервісами;
- висока вартість додаткових компонентів;
- залежність від політики виробника (оновлення, сумісність, підписки);
- прив'язаність до одного бренду, що ускладнює інтеграцію сторонніх пристроїв.

Переваги запропонованої системи на базі Raspberry Pi + Node-RED:

Повна кастомізація.

Користувач має повну свободу у визначенні логіки роботи системи, може створювати власні потоки, сценарії та налаштовувати взаємодію між пристроями. Це дає змогу реалізувати унікальні сценарії, які недоступні у фірмових системах.

Відсутність залежності від хмари.

Вся система функціонує локально, не потребує постійного підключення до інтернету чи зовнішніх сервісів. Це забезпечує більшу безпеку, конфіденційність та швидкість реакції.

Низька вартість.

Компоненти системи доступні на ринку, підтримуються різними виробниками та не вимагають регулярних платежів чи підписок. Багато рішень є повністю безкоштовними та з відкритим кодом.

Гнучкість у виборі обладнання.

Система не обмежує вибір пристроїв одним брендом. Можна використовувати будь-які датчики, реле, модулі ESP32, Wi-Fi розетки та інше.

Освітня цінність.

Самостійна розробка системи надає цінні практичні навички у сфері IoT, мереж, автоматизації та програмування, що може бути корисним у професійній діяльності.

Можливість повної автономності.

Система може працювати навіть без доступу до інтернету, наприклад, у сільській місцевості або як незалежна мережа.

Отже, хоча комерційні системи мають свою нішу та підходять для недосвідчених користувачів, запропонована система є більш вигідним та універсальним варіантом для ентузіастів, студентів, дослідників та розробників. Вона не тільки дає змогу заощадити кошти, але й відкриває широкі можливості для творчості, навчання та гнучкого керування власним розумним середовищем.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було досягнуто закріплення теоретичних знань та отримано практичні вміння, необхідні для розробки та впровадження сучасних систем автоматизації у сфері комп'ютерно-інтегрованих технологій. Виконання роботи сприяло формуванню навичок самостійного технічного мислення, досвіду роботи з апаратно-програмними платформами, а також вміння аналізувати поставлені завдання та розробляти ефективні інженерні рішення.

У ході розробки було спроектовано систему автоматизації освітлення на базі платформи Node-RED та мікрокомп'ютера Raspberry Pi. Проведено дослідження апаратної та програмної складової системи, обґрунтовано вибір основних компонентів, зокрема сенсора освітленості BH1750 та розумної Wi-Fi розетки Gosund EP2 для реалізації бездротового керування. Створено логіку автоматизованого контролю освітлення з урахуванням показників навколишнього середовища, а також забезпечено можливість ручного керування системою через веб-інтерфейс Node-RED Dashboard.

У результаті виконання кваліфікаційної роботи:

- сформульовано технічні вимоги до системи автоматизації освітлення;
- розроблено структурну та логічну схему побудови системи;
- реалізовано обробку даних сенсора освітленості та керування виконавчими пристроями у середовищі Node-RED;
- створено зручний графічний веб-інтерфейс для локального керування системою;
- забезпечено можливість інтеграції системи з бездротовими виконавчими пристроями через протокол MQTT;
- проведено аналіз потенціалу масштабування системи та подальшого її розвитку.

Розроблена система підтвердила доцільність використання відкритих апаратно-програмних платформ для побудови ефективних, масштабованих та

економічно вигідних систем розумного будинку. Результати виконаної роботи можуть бути використані як основа для подальшого вдосконалення системи автоматизації, її розширення функціональності (керування кліматом, безпекою, мультимедійними пристроями), а також для практичного впровадження у реальних умовах побутового та промислового застосування.

ПЕРЕЛІК ПОСИЛАНЬ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. Структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. – 29 с.
2. Навчальний посібник з підготовки кваліфікаційної роботи бакалавра для здобувачів вищої освіти денної і заочної форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології» та 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» освітньої програми «Системна інженерія» [Електронний ресурс] : навчальний посібник / І. Ш. Невлюдов, О. В. Токарева, О. М. Цимбал, А. І. Бронніков ; М-во освіти і науки України, ХНУРЕ. Харків : Видавництво Іванченка І. С., 2023. 218 с. ISBN 978-617-8332-16-7; DOI: 10.30837/978-617-8332-16-7 / Упоряд.: І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, С.П. Новоселов, О.В. Сичова. – Харків: ХНУРЕ, 2022. – 55 с.
3. Node-RED. Low-code programming for event-driven applications [Електронний ресурс]. – Режим доступу: [www / URL: https://nodered.org](http://www.nodered.org)
4. Raspberry Pi Documentation [Електронний ресурс]. – Режим доступу: [www / URL: https://www.raspberrypi.com/documentation](http://www.raspberrypi.com/documentation)
5. BH1750 Digital Light Sensor Module [Електронний ресурс]. – Режим доступу: [www / URL: https://components101.com/sensors/bh1750-light-sensor](http://components101.com/sensors/bh1750-light-sensor)
6. MQTT Version 3.1.1 Protocol Specification [Електронний ресурс]. – Режим доступу: [www / URL: https://mqtt.org/mqtt-specification](http://mqtt.org/mqtt-specification)
7. Arduino [Електронний ресурс]. – Режим доступу: [www / URL: https://www.arduino.cc](http://www.arduino.cc)
8. Денисюк В.Ю. Електронний посібник з дисципліни «Автоматизація виробничих процесів в приладобудуванні» / В.Ю. Денисюк – Луцьк: Луцький національний технічний університет, 2017 [Електронний ресурс]. – Режим доступу: [www / URL: https://elib.lntu.edu.ua/sites/default/files/elib_upload/...](https://elib.lntu.edu.ua/sites/default/files/elib_upload/...)
9. Кожухаренко С.О. Аналіз структурних та функціональних особливостей комплексу завдань обліку відвідуваності занять у ВНЗ / С.О.

Кожухаренко // «Automation and Development of Electronic Devices» ADED-2022: збірник студентських наукових статей. – Харків: ХНУРЕ, 2022. – Вип. 2. – С. 196–199.

10. Іванов, М. О. Розроблення автоматичної системи розумного будинку на Node Red / М. О. Іванов // Collection of Students' Scientific Paper «Automation and Development Of Electronic Devices» ADED-2025 Part 1 (Key infrastructure 2025) - Kharkiv/ The Editorial.: Nevlyudov I.Sh. (head), that all. Kharkiv: Kind of Kharkiv National University of Radio Electronics [electronic edition], 2025. – Vol. 1. – P. 72-76