

КВАЛІФІКАЦІЙНА РОБОТА

Програмно-апаратна система освітлення приміщення з датчиком присутності та регулюванням яскравості

Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КвРКІ 02280.22.01.02 ПЗ

Виконав здобувач IV курсу, група KI2-22-1


Підпис

Петро СВІНЦІ

Ініціали, прізвище

Керівник канд. техн. наук, доцент
Науковий ступінь, учене звання


Підпис

Дмитро МЕДЗАТИЙ

Ініціали, прізвище

Нормоконтролер канд. фіз.-мат. наук, доц.
Науковий ступінь, учене звання


Підпис

Тетяна КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІІС


Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

«01» червня 2026 р.

дата

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІІС

 Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Свінцу Петру Віталійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-апаратна система освітлення приміщення з датчиком присутності та регулюванням яскравості

Керівник проекту (роботи) Медзятий Дмитро Миколайович, к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

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

Аналіз предметної області та огляд існуючих рішень

Проектування програмно-апаратної системи освітлення приміщення з датчиком присутності та регулюванням яскравості

Симуляція програмно-апаратної системи освітлення приміщення з датчиком присутності та регулюванням яскравості

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Структурна схема програмно-апаратної системи освітлення приміщення

Схема потоку обробки даних у Node-RED

Результати симуляції

6. Консультанти розділів кваліфікаційної роботи

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

7. Дата видачі завдання « 10 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2026	виконано
4	Робота над розділом 2 – проектування структури програмно-апаратної системи	01.04.2026	виконано
5	Робота над розділом 3 – реалізація та симуляція програмно-апаратної системи	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	виконано
7	Попередній захист ВКР	25.05.2026	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач


Підпис

Петро СВІНЦ

Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи


Підпис

Дмитро МЕДЗАТИЙ

Імя, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-апаратна система освітлення приміщення з датчиком присутності та регулюванням яскравості».

Автор роботи: Петро СВІНЦ.

Керівник роботи: Дмитро МЕДЗАТИЙ.

Пояснювальна записка: 60 с., 23 рис., 3 табл., 3 дод., 40 джерел.

Графічна частина: 3 креслення.

ОСВІТЛЕННЯ, ІНТЕРНЕТ РЕЧЕЙ, NODE RED, WOKWI.

Сучасний розвиток технологій інтернету речей та автоматизації будівель створює передумови для широкого впровадження систем "розумного" освітлення, здатних адаптуватися до реальних умов використання приміщення без постійної участі людини. Освітлення є однією з найбільш енерговитратних статей споживання електроенергії в офісних, навчальних та побутових приміщеннях, тому будь-яке скорочення часу роботи освітлювальних приладів за відсутності реальної потреби має безпосередній економічний ефект. Водночас в умовах нестабільності енергопостачання та зростання тарифів на електроенергію питання енергоефективного керування освітленням набуває особливої актуальності не лише для великих об'єктів, а й для окремих приміщень.

Метою дипломної роботи є проектування та реалізація програмно-апаратної системи автоматизованого керування освітленням приміщення на базі мікроконтролера ESP32, що забезпечує виявлення присутності людини за допомогою датчика руху, адаптивне регулювання яскравості освітлення залежно від рівня природного освітлення та підтримку дистанційного моніторингу й керування через протокол MQTT і веб-орієнтовану інформаційну панель.







Підпис здобувача

16.06.2026

Дата

ЗМІСТ

Вступ.....	4
1 Аналіз предметної області та огляд існуючих рішень	6
1.1 Актуальність автоматизації освітлення приміщень	6
1.2 Огляд технологій та протоколів для побудови IoT-систем освітлення.....	7
1.3 Датчики для систем автоматичного керування освітленням	10
1.4 Методи регулювання яскравості штучного освітлення	12
1.5 Аналіз відомих рішень та засобів автоматизації освітлення	14
1.6 Огляд мікроконтролерної платформи ESP32	18
1.7 Огляд програмного забезпечення для побудови інформаційних панелей	20
1.8 Постановка задачі.....	22
2 Проєктування програмно-апаратної системи освітлення приміщення з датчиком присутності та регулюванням яскравості.....	25
2.1 Вимоги до програмно-апаратної системи.....	25
2.2 Структурна схема програмно-апаратної системи.....	26
2.3 Алгоритм функціонування мікроконтролера.....	28
2.4 Аналіз компонентів апаратної складової.....	31
2.5 Оцінка вартості компонентів	36
2.9 Висновки до другого розділу	37
3 Симуляція програмно-апаратної системи освітлення приміщення з датчиком присутності та регулюванням яскравості.....	39
3.1 Мета і завдання проведення симуляції	39
3.2 Реалізація інформаційної панелі за допомогою Node-RED клієнта	41
3.3 Створення цифрового двійника програмно-апаратної системи у Wokwi	53
3.6 Висновки до третього розділу.....	57
Висновки	59

КВРКІ. 022080.22.01.02 ПЗ										
Зм.	Арк.	Молокум.	Підпис	Дата	Програмно-апаратна система освітлення приміщення з датчиком присутності та регулюванням яскравості			Літера	Аркуш	Аркушів
Виконав	Петро СВІНЦ							у	2	67
Перевір.	Дмитро МЕДЗАТИЙ				ХНУ КІ2-22-1					
Н.контр.	Тетяна КИСІЛЬ									
Затвер.	Ольга ПАВЛОВА			01.06						

Перелік джерел посилань	61
Додаток А Структурна схема програмно-апаратної системи освітлення приміщення	65
Додаток Б Схема потоку обробки даних у Node-RED	66
Додаток В Результати симуляції	67

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 3
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Сучасний розвиток технологій інтернету речей та автоматизації будівель створює передумови для широкого впровадження систем "розумного" освітлення, здатних адаптуватися до реальних умов використання приміщення без постійної участі людини. Освітлення є однією з найбільш енерговитратних статей споживання електроенергії в офісних, навчальних та побутових приміщеннях, тому будь-яке скорочення часу роботи освітлювальних приладів за відсутності реальної потреби має безпосередній економічний ефект. Водночас в умовах нестабільності енергопостачання та зростання тарифів на електроенергію питання енергоефективного керування освітленням набуває особливої актуальності не лише для великих об'єктів, а й для окремих приміщень.

Метою дипломної роботи є проєктування та реалізація програмно-апаратної системи автоматизованого керування освітленням приміщення на базі мікроконтролера ESP32, що забезпечує виявлення присутності людини за допомогою датчика руху, адаптивне регулювання яскравості освітлення залежно від рівня природного освітлення та підтримку дистанційного моніторингу й керування через протокол MQTT і веб-орієнтовану інформаційну панель.

Об'єктом дослідження є процес автоматизованого керування освітленням приміщення на основі даних, отриманих від датчика присутності та датчика освітленості, у поєднанні з дистанційним керуванням з боку оператора.

Предметом дослідження є методи та засоби проєктування програмно-апаратних систем керування освітленням на базі мікроконтролера ESP32, включно з алгоритмами адаптивного регулювання яскравості за допомогою широтно-імпульсної модуляції, протоколом обміну даними MQTT, платформою візуалізації та керування Node-RED, а також засобами симуляційного моделювання апаратної частини у середовищі Wokwi.

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 4
Зм.	Арк.	№ докум.	Підпис	Дата		

У межах кваліфікаційної роботи планується розробка та дослідження прототипу системи освітлення, що поєднує апаратну частину на базі ESP32 з датчиком присутності, датчиком освітленості та світлодіодним джерелом світла, керованим за допомогою апаратного PWM-модуля, із програмною частиною, що включає локальний MQTT-брокер Mosquitto, потоки обробки даних Node-RED та веб-інтерфейс інформаційної панелі оператора. Реалізація передбачає підтримку чотирьох режимів роботи системи: автоматичного режиму, в якому яскравість визначається на основі комбінації показань датчика присутності та датчика освітленості; ручного режиму з повним override керуванням з боку оператора; нічного режиму з обмеженою комфортною яскравістю; та режиму енергозбереження з найбільш консервативною логікою активації освітлення.

Для досягнення поставленої мети застосовуються методи структурного проектування програмно-апаратних систем, методи мережевої взаємодії на основі архітектури публікації та підписки, методи імітаційного моделювання сенсорних даних на основі математичних моделей денного циклу освітленості та стохастичних моделей присутності людини, а також методи розробки вбудованого програмного забезпечення для мікроконтролерів сімейства ESP32 у середовищі Arduino.

Практична цінність роботи полягає у створенні працездатного прототипу програмно-апаратної системи автоматизованого керування освітленням, яка може бути впроваджена в реальних приміщеннях навчального або офісного призначення для скорочення непродуктивного часу роботи освітлення та підвищення комфорту користувачів за рахунок адаптивного регулювання яскравості.

					КВРКІ. 022080.22.01.02 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Актуальність автоматизації освітлення приміщень

Питання раціонального використання електричної енергії набуває дедалі більшої актуальності в умовах зростання її вартості та підвищення вимог до енергоефективності будівель. Освітлення є однією з найбільш енергоємних статей споживання як у житловому, так і в комерційному секторі. За даними Міжнародного енергетичного агентства, на освітлення припадає близько п'ятнадцяти відсотків від загального споживання електроенергії у будівлях, причому значна частина цього споживання є нераціональною – освітлення продовжує працювати у відсутність людей або при достатньому рівні природного освітлення.

Традиційні системи керування освітленням, засновані на ручному вмиканні та вимиканні через настінні вимикачі, не забезпечують жодної автоматизації і повністю покладаються на свідомість користувача. Такий підхід є неефективним з точки зору енергоспоживання, особливо у громадських приміщеннях, коридорах, санвузлах та інших зонах із нерегулярним перебуванням людей. Дослідження показують, що впровадження автоматичних систем керування освітленням із датчиками присутності дозволяє скоротити споживання електроенергії на освітлення від двадцяти до шістдесяти відсотків залежно від типу приміщення та сценарію використання.

Паралельно із проблемою енергоефективності існує запит на підвищення комфорту та зручності користування освітленням. Сучасні користувачі очікують від систем освітлення не лише простого вмикання та вимикання, а й адаптивного регулювання яскравості залежно від часу доби, рівня природного освітлення та характеру виконуваних завдань. Комфортне освітлення безпосередньо впливає на продуктивність праці, концентрацію уваги та самопочуття людини, що

					КВРКІ. 022080.22.01.02 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

підтверджується численними дослідженнями у галузі ергономіки та архітектурного середовища.

Розвиток концепції Інтернету речей відкрив принципово нові можливості для побудови інтелектуальних систем керування освітленням. Поява доступних мікроконтролерів із вбудованою підтримкою WiFi, зокрема платформи ESP32, суттєво знизила поріг входу у розробку таких систем і зробила їх доступними не лише для великих комерційних об'єктів, а й для приватних користувачів та освітніх проєктів. Сьогодні побудова повноцінної системи автоматичного керування освітленням може бути реалізована з мінімальними апаратними витратами при використанні відкритого програмного забезпечення та стандартизованих протоколів передачі даних.

Саме тому розробка програмно-апаратної системи освітлення приміщення з датчиком присутності та регулюванням яскравості є актуальним завданням, що поєднує практичну цінність з точки зору енергозбереження та інженерний інтерес з точки зору інтеграції сучасних IoT-технологій і протоколів передачі даних.

1.2 Огляд технологій та протоколів для побудови IoT-систем освітлення

Побудова сучасної системи автоматичного керування освітленням передбачає вибір технологічного стеку, що складають сукупності апаратних платформ, комунікаційних протоколів та програмних інструментів, які у своїй взаємодії забезпечують функціонування системи від датчика до інтерфейсу оператора. Розгляд наявних технологій доцільно здійснювати на кількох рівнях: рівні мікроконтролерних платформ, рівні комунікаційних протоколів та рівні програмного забезпечення для обробки і візуалізації даних.

На рівні мікроконтролерних платформ для IoT-застосунків найбільшого поширення набули три сімейства. Платформа Arduino на базі мікроконтролерів AVR є класичним рішенням для навчальних та прототипних проєктів завдяки

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

простоті програмування та великій спільноті. Проте відсутність вбудованого WiFi-модуля у більшості плат Arduino змушує використовувати зовнішні мережеві модулі, що ускладнює схему та збільшує вартість. Платформа Raspberry Pi є одноплатним комп'ютером із повноцінною операційною системою Linux, що відкриває ширші можливості для обробки даних, проте є надлишковою для задач простого керування периферією та має значно вищу вартість. Платформа ESP32 від компанії Espressif Systems займає проміжне положення: це повноцінний мікроконтролер із двоядерним процесором Xtensa LX6, вбудованими модулями WiFi та Bluetooth, багатоканальним АЦП та апаратним ШІМ-контролером при вартості близько п'яти доларів США за модуль. Саме поєднання обчислювальних можливостей, вбудованої мережевої підтримки та низької вартості робить ESP32 оптимальним вибором для IoT-систем керування освітленням.

На рівні комунікаційних протоколів для IoT виділяють кілька основних підходів. Протокол HTTP є найбільш звичним для веб-розробників, однак його використання у вбудованих системах пов'язане зі значними накладними витратами: кожен запит вимагає встановлення TCP-з'єднання, передачі заголовків та очікування відповіді, що є неефективним для телеметричних даних, що оновлюються кожні одну-дві секунди. Протокол WebSocket вирішує проблему постійного перевстановлення з'єднань, однак також орієнтований на браузерні застосунки і не є оптимальним для мікроконтролерів з обмеженими ресурсами. Протокол Zigbee та Z-Wave орієнтовані на mesh-мережі пристроїв «розумного будинку» та забезпечують низьке споживання енергії, проте вимагають окремого координатора мережі та спеціалізованих радіомодулів. Протокол MQTT (Message Queuing Telemetry Transport) є найбільш поширеним вибором для IoT-застосунків завдяки мінімальним накладним витратам, моделі публікації та підписки та підтримці різних рівнів гарантії доставки (QoS). MQTT працює поверх TCP/IP, що дозволяє використовувати стандартну WiFi-інфраструктуру без будь-яких спеціалізованих радіомодулів.

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

Протокол MQTT, розроблений компанією IBM наприкінці 1990-х років для моніторингу нафтопроводів через супутниковий зв'язок, був стандартизований організацією OASIS у 2014 році. Його ключова особливість це модель «брокер – видавець – підписник»: пристрої не обмінюються даними безпосередньо, а публікують повідомлення у визначені топіки через центральний брокер, а інші пристрої отримують ці повідомлення, підписавшись на відповідні топіки. Така архітектура забезпечує слабке зв'язування компонентів системи: видавець не знає нічого про підписників, і навпаки. MQTT-брокер зберігає підписки та маршрутизує повідомлення, забезпечуючи асинхронну доставку навіть за тимчасової відсутності підписника у мережі. Мінімальний розмір заголовка MQTT-паketу складає лише два байти, що робить протокол ідеальним для пристроїв з обмеженими ресурсами та каналами зв'язку з низькою пропускнуою здатністю.

На рівні програмного забезпечення для обробки та візуалізації IoT-даних широкого застосування набула платформа Node-RED, розроблена компанією IBM і передана у відкритий доступ у 2016 році. Node-RED реалізує концепцію потокового програмування: логіка обробки даних визначається не традиційним кодом, а графічними потоками, де вузли з'єднуються між собою і передають повідомлення. Це суттєво знижує поріг входу для розробників, не знайомих з серверним програмуванням, і водночас забезпечує достатню гнучкість для реалізації складних сценаріїв обробки. Розширення node-red-dashboard додає до Node-RED готові UI-компоненти для побудови інформаційних панелей оператора безпосередньо у браузері без необхідності написання HTML та JavaScript-коду.

Для симуляції апаратної частини IoT-систем сьогодні існують спеціалізовані інструменти. Середовище Wokwi є провідним онлайн-симулятором мікроконтролерів, що підтримує повноцінну симуляцію ESP32 включаючи WiFi-стек, периферійні пристрої та більшість бібліотек Arduino. На відміну від простих емуляторів, Wokwi виконує реальний скомпільований

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 9
Зм.	Арк.	№ докум.	Підпис	Дата		

машинний код прошивки, що забезпечує максимальну відповідність між симуляцією та поведінкою реального пристрою. Можливість підключення симульованого ESP32 до реального MQTT-брокера через Wokwi-мережу дозволяє проводити наскрізне тестування всього ланцюжка передачі даних без жодного фізичного обладнання.

1.3 Датчики для систем автоматичного керування освітленням

Ключовим елементом будь-якої системи автоматичного керування освітленням є сенсорна підсистема, що забезпечує вхідні дані для прийняття рішень про вмикання, вимикання та регулювання яскравості. У залежності від реалізованих функцій система може використовувати датчики присутності, датчики освітленості, а також їх комбінацію для більш інтелектуального керування.

Датчики виявлення присутності людини поділяються на кілька принципово різних типів. Пасивні інфрачервоні датчики є найбільш поширеним рішенням завдяки низькій вартості, простоті підключення та задовільній надійності. Принцип їх дії ґрунтується на реєстрації змін теплового інфрачервоного випромінювання у зоні чутливості: людина, переміщуючись у полі зору датчика, спричиняє зміну теплової картини, яку піроелектричний сенсор перетворює на електричний сигнал. Типові PIR-датчики, зокрема HC-SR501 та AM312, забезпечують кут огляду до 120 градусів і дальність до семи метрів, що є достатнім для більшості приміщень. Істотним обмеженням PIR є те, що вони реагують виключно на рух: нерухома людина у приміщенні може не виявлятися датчиком після закінчення часу утримання. Для задач керування освітленням це обмеження є прийнятним, оскільки метою є виявлення перебування активної людини у приміщенні.

Ультразвукові датчики присутності, наприклад HC-SR04, використовують принцип відлуння, що полягає у тому, що датчик випромінює ультразвуковий

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

імпульс і вимірює час до повернення відбитого сигналу, що дозволяє визначити відстань до об'єкта. На відміну від PIR, ультразвукові датчики здатні реагувати на нерухомі об'єкти, оскільки реєструють сам факт наявності перешкоди в зоні чутливості, а не її рух. Проте ультразвукові датчики чутливі до акустичних перешкод та мають вузьку діаграму спрямованості. У контексті симуляції HC-SR04 активно використовується як функціональний заміник PIR у середовищі Wokwi, де власне PIR-датчик відсутній у бібліотеці компонентів.

Мікрохвильові датчики присутності працюють на основі ефекту Доплера і реєструють зміщення частоти відбитого мікрохвильового сигналу при русі об'єкта. Вони здатні виявляти мінімальні рухи, зокрема дихання нерухомої людини, і не залежать від теплового контрасту між людиною та навколишнім середовищем. Проте вища вартість та можливість спрацювання через перешкоди (наприклад, рух за тонкою стіною) обмежують їх застосування у простих системах освітлення.

Датчики освітленості є другим ключовим елементом адаптивних систем керування яскравістю. Фоторезистори, зокрема серія GL5516, змінюють свій опір залежно від інтенсивності падаючого світла: від кількох сотень Ом при яскравому освітленні до декількох мегаОм у повній темряві. Підключення фоторезистора через дільник напруги до аналогового входу мікроконтролера дозволяє отримати числове значення рівня освітленості. Фоторезистори є недорогими та надійними, однак не мають стандартизованої одиниці виміру і вимагають калібрування для кожного конкретного розташування. Фотодіоди та фототранзистори забезпечують вищу точність і швидший відгук, проте для задач вимірювання фонового освітлення приміщення надмірна точність є зайвою. Цифрові датчики освітленості, наприклад BH1750 з інтерфейсом I2C, надають вимірне значення безпосередньо у люксах, проте їх вища вартість виправдана лише у системах із суворими метрологічними вимогами.

У пропонованій системі використовується поєднання PIR-датчика HC-SR501 та фоторезистора GL5516, що забезпечує достатню функціональність для

реалізації адаптивного керування яскравістю при мінімальній вартості компонентів.

1.4 Методи регулювання яскравості штучного освітлення

Регулювання яскравості штучного освітлення є фундаментальною технічною задачею, вирішення якої можливе кількома принципово різними методами. Вибір методу залежить від типу джерела світла, вимог до точності регулювання, допустимого рівня втрат потужності та складності реалізації.

Резистивний метод регулювання є найпростішим з точки зору схемотехніки: змінний опір, включений послідовно з навантаженням, змінює напругу на джерелі світла і, відповідно, його яскравість. Цей метод широко використовувався у ранніх системах dimming для ламп розжарювання. Проте його принциповий недолік полягає у тому, що потужність, яка «зрізається» за допомогою резистора, розсіюється у вигляді тепла, а не повертається до мережі. При зниженні яскравості вдвічі половина споживаної потужності перетворюється на тепло в регулювальному елементі, що робить метод неприйнятним з точки зору енергоефективності.

Фазове регулювання є стандартним методом для мережевих систем dimming у житловому та комерційному освітленні. Симістор «зрізає» частину синусоїди змінного струму, зменшуючи середню потужність, що подається на навантаження. Метод є відносно енергоефективним, оскільки у закритому стані симістор не споживає потужності. Проте він потребує обережного поводження з мережевою напругою та несумісний із більшістю сучасних LED-драйверів без спеціальної адаптації.

Широтно-імпульсна модуляція є найбільш поширеним методом регулювання яскравості у сучасних LED-системах низьковольтного живлення. Принцип ШІМ полягає у комутації навантаження з постійною частотою при змінній шпаруватості сигналу. Оскільки людське око не здатне сприймати

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

мерехтіння з частотою вище ста Гц як окремі імпульси, а не постійне світло, при частоті ШІМ-сигналу від одного кілогерця і вище людина сприймає усереднену яскравість.

Ключова перевага ШІМ полягає у відсутності резистивних втрат: силовий транзистор, що комутує навантаження, знаходиться або у повністю відкритому стані (падіння напруги мінімальне), або у повністю закритому (струм відсутній). Тому ККД регулятора на основі ШІМ наближається до теоретичного максимуму незалежно від встановленого рівня яскравості.

ESP32 має вбудований апаратний ШІМ-контролер LEDC, що підтримує до шістнадцяти незалежних каналів із налаштовуваною частотою та роздільною здатністю від одного до двадцяти біт. Апаратна реалізація означає, що після початкового налаштування та задання шпаруватості мікроконтролер формує ШІМ-сигнал автономно без залучення процесора, що звільняє обчислювальні ресурси для виконання інших задач.

У пропонованій системі канал LEDC налаштований на частоту 5000 Гц та 8-бітну роздільну здатність, що забезпечує 256 рівнів яскравості без видимого мерехтіння та є більш ніж достатнім для задач керування освітленням приміщення.

Оскільки GPIO ESP32 може віддавати не більше 12 мА, що є явно недостатнім для живлення LED-стрічки, між мікроконтролером та навантаженням необхідний силовий ключ.

У низьковольтних схемах (5 В або 12 В) стандартним рішенням є N-канальний MOSFET-транзистор із логічним рівнем керування (logic-level MOSFET), наприклад IRLZ44N або IRFZ44N. Такий транзистор повністю відкривається при напрузі затвор-витік від 2,5 до 4 В, що є сумісним з 3,3 В логікою ESP32. ШІМ-сигнал від піна мікроконтролера GPIO25 подається на затвор транзистора, а стрічка живиться від зовнішнього джерела через ланцюг стік-витік транзистора.

1.5 Аналіз відомих рішень та засобів автоматизації освітлення

На ринку систем автоматизованого керування освітленням сьогодні представлені рішення різного рівня складності та вартості від простих таймерних вимикачів до повноцінних платформ «розумного будинку». Аналіз цих рішень дозволяє виявити їх переваги та недоліки і обґрунтувати доцільність розробки власної системи на базі ESP32.

Система Philips Hue є одним із найбільш відомих комерційних рішень у сегменті інтелектуального освітлення (рис. 1.1).



Рисунок 1.1 – Philips - LED RGBW вбудований точковий світильник PERIDOT з регулюванням яскравості

Вона складається з «розумних» LED-ламп або стрічок, центрального хабу (Hue Bridge) та мобільного застосунку. Лампи комунікують із хабом через протокол Zigbee, а хаб підключається до домашньої мережі через Ethernet. Система підтримує регулювання яскравості, зміну колірної температури та повний спектр RGB, інтеграцію з голосовими асистентами та автоматизацію за сценаріями. Проте вартість базового комплекту (хаб плюс дві лампи) становить від восьмидесяти доларів США і вище, а розширення системи датчиками

присутності вимагає придбання окремих аксесуарів за ціною двадцять-тридцять доларів кожен. Прошивка пристроїв є закритою, і більшість функцій автоматизації залежать від хмарних сервісів Philips, що означає ризик припинення підтримки або деградації функціональності у разі зміни політики компанії.

Платформа Xiaomi Yeelight пропонує більш доступний варіант інтелектуального освітлення: LED-лампи та стрічки з підтримкою WiFi без необхідності окремого хабу (рис. 1.2).



Рисунок 1.2 – Розумна лампа Xiaomi Yeelight Smart LED Filament Gold E27
Apple HomeKit SmartThings

Пристрої підключаються безпосередньо до домашньої WiFi-мережі та керуються через застосунок Mi Home або Yeelight. Вартість базових пристроїв починається від десяти-п'ятнадцяти доларів, що є суттєво нижчим порогом входу порівняно з Philips Hue. Система частково підтримує MQTT та інтеграцію з Home Assistant, що відкриває певні можливості для кастомізації. Проте датчики

присутності не входять до базового рішення, а управління залишається переважно дистанційним, без реалізації повноцінного адаптивного алгоритму на основі сенсорних даних.

Система DALI (Digital Addressable Lighting Interface) є промисловим стандартом керування освітленням, широко застосовуваним у комерційних будівлях, офісах та виробничих приміщеннях. DALI забезпечує індивідуальну адресацію до шістдесяти чотирьох пристроїв на одній шині, підтримку груп та сцен освітлення, двосторонній зв'язок із можливістю зворотного читання стану. Протокол є стандартизованим (IEC 62386) і підтримується більшістю провідних виробників світлотехніки. Проте реалізація DALI вимагає спеціалізованих контролерів, сертифікованих DALI-драйверів для кожного світильника та окремої шини керування, що робить систему надлишковою та невиправдано дорогою для невеликих приміщень та індивідуального використання.

Рішення на базі Arduino широко представлені у сегменті DIY та навчальних проєктів. Численні відкриті проєкти демонструють реалізацію автоматичного керування освітленням із PIR-датчиком на базі Arduino Uno або Nano. Такі системи мають мінімальну вартість компонентів і дозволяють повністю контролювати логіку роботи. Проте відсутність вбудованого WiFi обмежує можливості дистанційного керування та моніторингу: для підключення до мережі необхідний зовнішній модуль ESP8266 або Ethernet-шилд, що ускладнює схему. Крім того, Arduino-рішення, як правило, не передбачають повноцінної інформаційної панелі оператора та обмежуються локальним керуванням без можливості дистанційного моніторингу.

Платформа Home Assistant є найбільш функціональним рішенням у сегменті відкритого програмного забезпечення для автоматизації будинку. Вона підтримує сотні інтеграцій із різними протоколами та пристроями, включаючи MQTT, Zigbee, Z-Wave, Philips Hue та багато інших. Home Assistant дозволяє реалізувати складні сценарії автоматизації з умовами, часовими тригерами та реакцією на стан датчиків. Проте для розгортання Home Assistant необхідний

окремий сервер, як правило, Raspberry Pi або аналогічний одноплатний комп'ютер, що збільшує як вартість, так і складність системи. Для вирішення конкретної задачі керування освітленням одного приміщення Home Assistant є надлишковим рішенням.

Порівняльний аналіз розглянутих рішень наведено у таблиці 1.1.

Таблиця 1.1 – Порівняльний аналіз існуючих рішень та пропонованої системи

Критерій	Philips Hue	Xiaomi Yeelight	Власна система ESP32
Відкритий код прошивки	Ні	Частково	Так
Гнучкість налаштування	Обмежена	Середня	Повна
Залежність від хмари виробника	Висока	Висока	Відсутня
Підтримка MQTT	Через міст	Так	Нативна
Адаптивне регулювання яскравості	Так	Так	Так
Датчик присутності (PIR)	Окремий аксесуар	Окремий аксесуар	Інтегрований
Вартість базового комплекту	Висока (від \$80)	Середня (від \$30)	Низька (від \$10)
Можливість симуляції фізичного обладнання без	Ні	Ні	Так (Wokwi)

Аналіз показує, що жодне з існуючих готових рішень не задовольняє повному переліку вимог одночасно: доступна вартість, відкритий код, нативна підтримка MQTT, інтегрований датчик присутності та можливість повноцінної симуляції без фізичного обладнання. Розробка власної системи на базі ESP32 із використанням MQTT, Node-RED та Wokwi дозволяє усунути ці обмеження і побудувати гнучке рішення, повністю адаптоване до конкретних вимог.

1.6 Огляд мікроконтролерної платформи ESP32

Мікроконтролер ESP32, розроблений компанією Espressif Systems та випущений у 2016 році, став одним із найбільш поширених рішень для IoT-застосунків завдяки вдалому поєднанню обчислювальних можливостей, інтегрованих периферійних пристроїв та доступної вартості.

Обчислювальне ядро ESP32 побудоване на базі двох 32-бітних процесорів Xtensa LX6, що працюють на тактовій частоті до 240 МГц. Наявність двох ядер дозволяє розподілити задачі між ними: одне ядро може обслуговувати мережевий стек WiFi, тоді як друге виконує прикладний код керування периферією. Обсяг вбудованої оперативної пам'яті складає 520 КБ SRAM, чого достатньо для виконання програм середньої складності з буферизацією даних. Для зберігання прошивки використовується зовнішня Flash-пам'ять об'ємом зазвичай від 4 до 16 МБ, підключена через інтерфейс SPI.

Вбудований WiFi-модуль підтримує стандарти IEEE 802.11 b/g/n у діапазоні 2,4 ГГц та може працювати у режимах Station (підключення до точки доступу), Access Point (власна точка доступу) та одночасно в обох режимах. Реалізація TCP/IP-стеку виконана на апаратному рівні, що суттєво знижує навантаження на основний процесор при мережевих операціях. Підтримка протоколів TLS/SSL дозволяє реалізовувати захищені з'єднання, що є важливим для комерційних застосунків. Вбудований модуль Bluetooth 4.2 із підтримкою

BLE відкриває додаткові можливості для безпроводного зв'язку на короткій відстані.

Аналогово-цифровий перетворювач ESP32 має 12-бітну роздільну здатність та підтримує до вісімнадцяти аналогових входів. 12-бітний АЦП забезпечує 4096 рівнів квантування, що відповідає кроку вимірювання близько 0,8 мВ при напрузі живлення 3,3 В. Для задачі вимірювання рівня освітленості через фоторезистор ця роздільна здатність є цілком достатньою.

Слід зазначити, що АЦП ESP32 має нелінійність у крайніх діапазонах входної напруги, що може вимагати калібрування при застосуванні у метрологічно відповідальних задачах, однак для відносного вимірювання рівня освітленості ця особливість є несуттєвою.

Контролер LEDC підтримує до шістнадцяти незалежних ШІМ-каналів із роздільною здатністю від 1 до 20 біт та частотою від кількох герців до десятків мегагерців. Апаратна реалізація ШІМ забезпечує точне дотримання заданих параметрів незалежно від завантаженості процесора та відсутність джиттера, характерного для програмної реалізації. Саме контролер LEDC використовується у пропонованій системі для формування ШІМ-сигналу керування яскравістю LED-стрічки.

Серед інших периферійних пристроїв ESP32 слід виділити: два 12-бітних ЦАП (цифрово-аналоговий перетворювач) для виведення аналогових сигналів; чотири апаратних таймери для точного відліку часових інтервалів; інтерфейси I2C, SPI, UART, I2S та CAN для підключення зовнішніх пристроїв; контролер сенсорних входів із можливістю реалізації емнісних кнопок без зовнішніх компонентів; та апаратний блок шифрування з підтримкою алгоритмів AES, RSA та SHA.

Модуль ESP32 DevKit V1, що використовується у пропонованій системі, є найбільш поширеним форм-фактором для розробки та прототипування.

Плата розміром 55×28 мм містить мікросхему ESP32-WROOM-32, USB-to-UART перетворювач CP2102 для програмування через USB, стабілізатор

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

напруги 3,3 В та виводить усі GPIO-піни на два ряди штирьових контактів із кроком 2,54 мм, сумісних зі стандартними макетними платами.

1.7 Огляд програмного забезпечення для побудови інформаційних панелей

Інформаційна панель оператора є критично важливим елементом будь-якої системи моніторингу та керування: вона забезпечує наочне відображення поточного стану системи та надає інструменти для дистанційного керування. Вибір платформи для побудови такої панелі суттєво впливає на складність розробки, можливості налаштування та вимоги до інфраструктури.

Node-RED, розроблений лабораторією IBM Emerging Technologies у 2013 році та переданий під управління фонду OpenJS Foundation, є де-факто стандартом для побудови IoT-застосунків у сегменті промислового інтернету речей та «розумного будинку». Середовище реалізує парадигму потокового програмування: розробник створює потоки обробки даних, з'єднуючи вузли (nodes) у редакторі з перетягуванням. Кожен вузол виконує певну функцію: підписку на MQTT-топік, перетворення JSON, порівняння значень, публікацію у топік, відображення на дашборді тощо. Логіку обробки даних визначає топологія з'єднань між вузлами, а не традиційний послідовний код. Node-RED розгортається як Node.js-застосунок і доступний через веб-браузер на порту 1880. Розширення node-red-dashboard додає понад двадцять UI-компонентів: кнопки, перемикачі, слайдери, gauge-індикатори, текстові поля, графіки та інші, що дозволяє побудувати повноцінну інформаційну панель без написання жодного рядка HTML або JavaScript.

Grafana є більш потужною платформою для візуалізації часових рядів даних, широко застосовуваною у корпоративному середовищі для моніторингу серверів, мереж та промислових систем. Grafana підтримує десятки джерел даних, включаючи InfluxDB, Prometheus, Elasticsearch та інші, і надає широкі можливості для побудови багатопанельних дашбордів із гнучким

налаштуванням. Проте для задач керування освітленням одного приміщення Grafana є надлишковою: вона орієнтована переважно на відображення та аналіз даних, а не на двосторонню взаємодію – реалізація елементів керування вимагає написання плагінів або інтеграції з зовнішніми системами.

ThingsBoard є повноцінною IoT-платформою з відкритим кодом, що підтримує MQTT, CoAP та HTTP як протоколи передачі даних та надає інструменти для побудови дашбордів, управління пристроями та реалізації правил обробки даних. Платформа підходить для масштабних промислових застосунків із десятками тисяч підключених пристроїв. Для задачі керування освітленням одного приміщення ThingsBoard також є надлишковим вибором через складність розгортання та конфігурування.

Порівняльний аналіз відомого програмного забезпечення для побудови інформаційних панелей наведений у таблиці 2.1.

Таблиця 1.2 – Порівняльний аналіз відомого програмного забезпечення для побудови інформаційних панелей

Параметр	Node-RED	Grafana	ThingsBoard
Основне призначення	Потокове програмування для IoT та інтеграцій	Візуалізація часових рядів та моніторинг	Повноцінна IoT-платформа
Тип взаємодії	Двостороння (керування + моніторинг)	Переважно одностороння (візуалізація)	Двостороння (керування + аналітика)
Складність розгортання	Низька	Середня	Висока
Підтримка MQTT	Так (вбудовано через вузли)	Через джерела/плагіни	Так (нативно)

Кінець таблиці 1.2

UI/дашборди	node-red- dashboard (кнопки, слайдери, графіки)	Потужні графічні панелі	Готові IoT- дашборди
Гнучкість налаштування	Висока (через flow-логіку)	Дуже висока у візуалізації	Висока, але складна
Масштабованість	Середня	Висока	Дуже висока
Основна перевага	Простота та швидка розробка	Сильна аналітика та графіки	Повний IoT- екосистемний підхід
Недолік для задачі	Обмежена аналітика	Слабка інтерактивність	Надмірна складність

Порівняльний аналіз показує, що Node-RED із розширенням node-red-dashboard є оптимальним вибором для запропонованої системи: він забезпечує нативну підтримку MQTT, простоту розробки потоків обробки даних, готові UI-компоненти для дашборду та двосторонню взаємодію з користувачем без необхідності встановлення додаткової інфраструктури або написання значного обсягу коду.

1.8 Постановка задачі

За результатами проведеного аналізу предметної області, а також огляду існуючих підходів до автоматизації керування освітленням приміщень та особливостей побудови програмно-апаратних систем на базі мікроконтролерів сімейства ESP32 можна сформулювати конкретне технічне завдання. Відповідно до мети роботи, якою є проєктування, реалізація та симуляційне тестування

програмно-апаратної системи освітлення приміщення з датчиком присутності та регулюванням яскравості, можна визначити наступний перелік завдань, які потрібно вирішити:

1. Провести аналіз предметної області, розглянути існуючі підходи до автоматизації систем освітлення приміщень, визначити переваги поєднання логіки присутності та адаптивного регулювання яскравості порівняно з традиційними системами дискретного увімкнення та вимкнення світла.

2. Дослідити архітектуру цільової платформи мікроконтролера ESP32, проаналізувати її апаратні ресурси, зокрема канали аналого-цифрового перетворення, апаратний PWM-контролер та вбудований модуль Wi-Fi, а також оцінити їх достатність для реалізації поставлених функцій.

3. Сформулювати повний перелік вимог до програмно-апаратної системи, включаючи функціональні вимоги до логіки чотирьох режимів роботи (автоматичного, ручного, нічного та енергозбереження), а також нефункціональні вимоги щодо часу реакції, енергоефективності та надійності мережевого з'єднання.

4. Розробити структурну схему системи, визначивши розподіл функціональних компонентів між апаратним рівнем (ESP32, датчики, світлодіодне освітлення), мережевим рівнем (MQTT-брокери Mosquitto та HiveMQ) та програмним рівнем (платформа Node-RED, веб-інтерфейс інформаційної панелі).

5. Обґрунтувати вибір апаратних компонентів системи, зокрема датчика присутності HC-SR501, фоторезистора як датчика освітленості та схеми керування світлодіодним навантаженням засобами широтно-імпульсної модуляції, з урахуванням їх характеристик та вартості.

6. Розробити алгоритм функціонування мікроконтролера у вигляді блок-схеми та реалізувати відповідну прошивку мовою C++ у середовищі Arduino, що забезпечує читання сенсорів, каскадну логіку перевірки активного режиму роботи та публікацію телеметрії через протокол MQTT.

7. Реалізувати програмний симулятор сенсорних даних мовою Python, що відтворює математичну модель денного циклу природного освітлення та стохастичну модель присутності людини для верифікації логіки системи без використання фізичного обладнання.

8. Спроекувати та реалізувати цифровий двійник апаратної частини у середовищі Wokwi, відтворивши схему підключення мікроконтролера до датчиків та виконавчих пристроїв, а також інформаційну панель оператора на платформі Node-RED з елементами керування, індикаторами та графіками динаміки параметрів.

9. Провести симуляційне тестування розробленої системи, перевірити коректність роботи всіх чотирьох режимів освітлення, двосторонній обмін даними між компонентами через MQTT-брокер, а також виконати оцінку орієнтовної вартості апаратних компонентів прототипу.

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ПРОГРАМНО-АПАРАТНОЇ СИСТЕМИ ОСВІТЛЕННЯ ПРИМІЩЕННЯ З ДАТЧИКОМ ПРИСУТНОСТІ ТА РЕГУЛЮВАННЯМ ЯСКРАВОСТІ

2.1 Вимоги до програмно-апаратної системи

Перед переходом до безпосереднього проєктування системи необхідно чітко сформулювати вимоги, яким вона має відповідати. Ці вимоги формуються з огляду на умови експлуатації, особливості середовища розгортання, обмеження щодо вартості компонентів та функціональне призначення системи в цілому.

З функціональної точки зору система повинна забезпечувати автоматичне керування освітленням приміщення залежно від наявності руху людини та поточного рівня природного освітлення. Для цього необхідно передбачити чотири чітко розмежовані режими роботи: автоматичний, ручний, нічний та режим енергозбереження. В автоматичному режимі яскравість штучного освітлення визначається адаптивно, залежно від свідчень датчика освітленості та датчика присутності. Ручний режим передбачає повне підпорядкування системи командам оператора, що надходять через панель керування. Нічний режим обмежує яскравість до мінімально комфортного рівня, достатнього для орієнтування у приміщенні. Режим енергозбереження активує освітлення лише за одночасного виконання двох умов, виявлення руху та недостатнього природного освітлення.

Окремою функціональною вимогою є реалізація таймера автоматичного вимкнення освітлення після зникнення людини з зони чутливості датчика. Тривалість таймауту відрізняється залежно від режиму: тридцять секунд у автоматичному та нічному режимах і п'ятнадцять секунд у режимі енергозбереження. Така диференціація дозволяє гнучко балансувати між зручністю користувача та економією електроенергії.

З технічної точки зору система повинна використовувати мікроконтролер із вбудованою підтримкою WiFi, оскільки передача телеметрії та отримання

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

команд від оператора здійснюється через протокол MQTT. Регулювання яскравості має виконуватися апаратним ШІМ-сигналом з достатньою роздільною здатністю, не менше восьми біт для забезпечення плавного переходу між рівнями освітлення. Зчитування аналогового сигналу з датчика освітленості потребує АЦП з роздільною здатністю не менше дванадцяти біт. Система повинна автоматично відновлювати підключення до MQTT-брокера після втрати мережевого зв'язку.

Щодо програмного забезпечення, прошивка мікроконтролера має реалізовувати двосторонній обмін повідомленнями з брокером: публікацію телеметричних даних кожні дві секунди та прийом команд керування в режимі реального часу. Формат передачі даних JSON, Він забезпечує сумісність з різними платформами обробки. На рівні хмарної обробки та візуалізації необхідно реалізувати інформаційну панель оператора з відображенням поточного стану датчиків, динамічними графіками та елементами ручного керування.

До нефункціональних вимог належать: бюджет компонентів апаратної частини не більше п'ятисот гривень; можливість повноцінного тестування усієї системи в середовищі симуляції без фізичного обладнання; використання виключно програмного забезпечення з відкритим кодом або безкоштовних інструментів розробки.

2.2 Структурна схема програмно-апаратної системи

Структурна Структурна схема системи охоплює два рівні: апаратну частину, розташовану безпосередньо у приміщенні, та програмну частину, що функціонує на стороні керуючого вузла.

Апаратна частина системи побудована навколо мікроконтролера ESP32, який є центральним елементом обробки даних та керування. На його входи надходять сигнали від двох датчиків. PIR-датчик фіксує факт присутності

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

людини у приміщенні та передає на мікроконтролер дискретний сигнал виявлення руху. LDR-датчик вимірює поточний рівень природного освітлення та передає аналоговий сигнал, який мікроконтролер перетворює через вбудований АЦП на числове значення освітленості. На основі отриманих даних ESP32 реалізує логіку одного з чотирьох режимів роботи та формує ШІМ-сигнал, що надходить на вихідний елемент, який представляє LED-освітлення з регульованою яскравістю у діапазоні від 0 до 255 одиниць.

Зв'язок між апаратною частиною та програмним рівнем здійснюється через протокол MQTT версії 3.1.1 по мережі WiFi. ESP32 кожні дві секунди публікує зведений JSON-пакет у топик `lighting/status`, що містить поточні значення яскравості, стану PIR-датчика, рівня освітленості LDR, активного режиму роботи та стану живлення освітлення. Окрім зведеного пакету, окремі значення дублюються у спеціалізовані топіки `lighting/pir`, `lighting/ldr`, `lighting/brightness` та `lighting/mode`, що дозволяє підписникам отримувати лише ті дані, які їм необхідні, без розбору повного JSON-об'єкта. У зворотному напрямку ESP32 підписується на три командні топіки: `lighting/cmd/mode` для перемикання між режимами роботи, `lighting/cmd/brightness` для встановлення цільового рівня яскравості в ручному режимі та `lighting/cmd/power` для вмикання і вимикання освітлення незалежно від поточного режиму. Отримання повідомлення у будь-якому з цих топіків негайно змінює внутрішній стан мікроконтролера і відображається у наступному циклі розрахунку яскравості.

Програмна частина складається з MQTT-брокера Mosquitto, що розгорнутий локально за адресою 127.0.0.1, та Node-RED дашборду. Брокер маршрутизує повідомлення між мікроконтролером і Node-RED у обох напрямках. Node-RED, у свою чергу, відображає отримані дані у вигляді графіків та індикаторів і надає оператору елементи керування для зміни режиму роботи та рівня яскравості. Команди, сформовані оператором через дашборд, повертаються через брокер до мікроконтролера, замикаючи контур керування. Структурну схему системи наведено на рисунку 2.1.

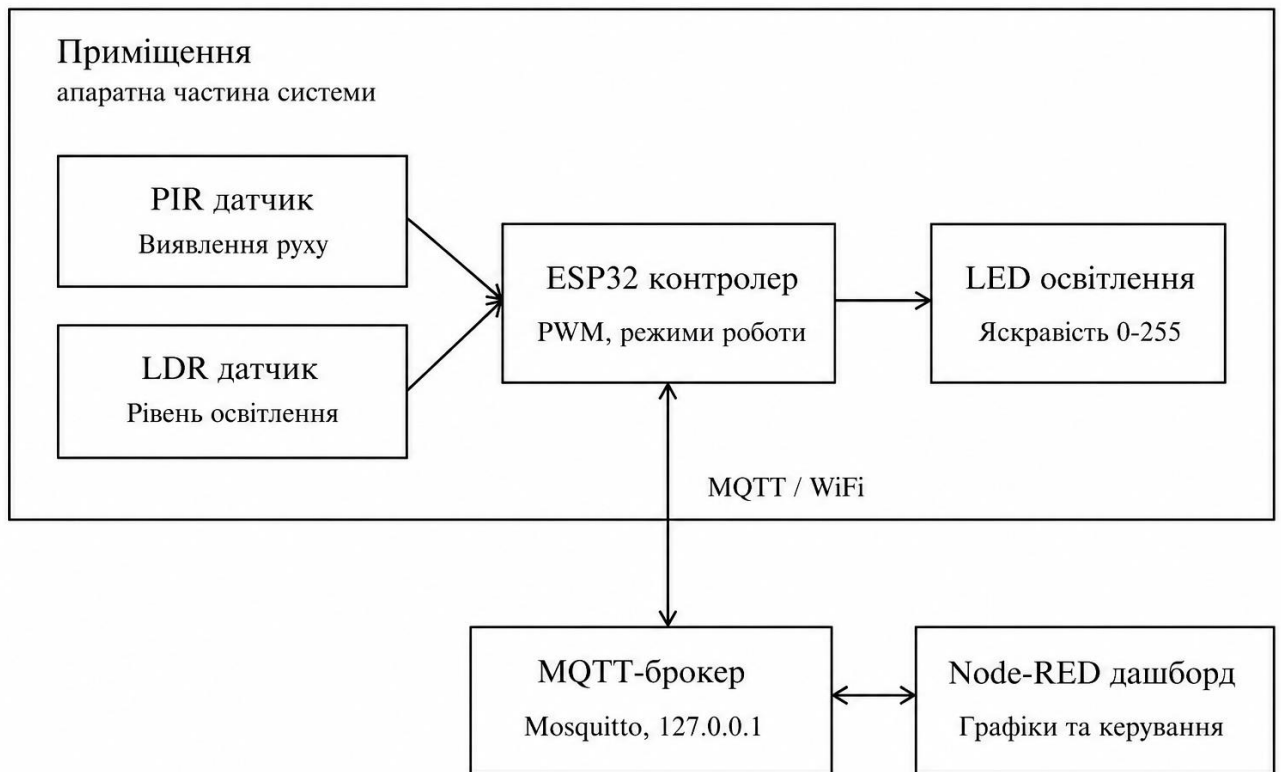


Рисунок 2.1 – Структурна схема програмно-апаратної системи

2.3 Алгоритм функціонування мікроконтролера

Алгоритм роботи мікроконтролера ESP32 відображає повний цикл виконання прошивки, починаючи від ініціалізації апаратної частини до циклічної обробки сенсорних даних та передачі телеметрії. Блок-схему алгоритму наведено на рисунку 2.2.

Після подачі живлення мікроконтролер виконує ініціалізацію апаратної частини. На цьому етапі налаштовуються GPIO-піни, зокрема пін датчика PIR переводиться у режим цифрового входу, пін LED-освітлення прив'язується до ШІМ-каналу, а пін індикатора WiFi налаштовується як цифровий вихід. ШІМ-канал ініціалізується з частотою 5000 Гц та 8-бітною роздільною здатністю, що забезпечує 256 рівнів яскравості без видимого мерехтіння. Одночасно задаються початкові значення змінних стану: режим роботи встановлюється як

автоматичний, яскравість – нульова, прапорець живлення – вимкнено, часова мітка останнього виявленого руху – поточний момент.

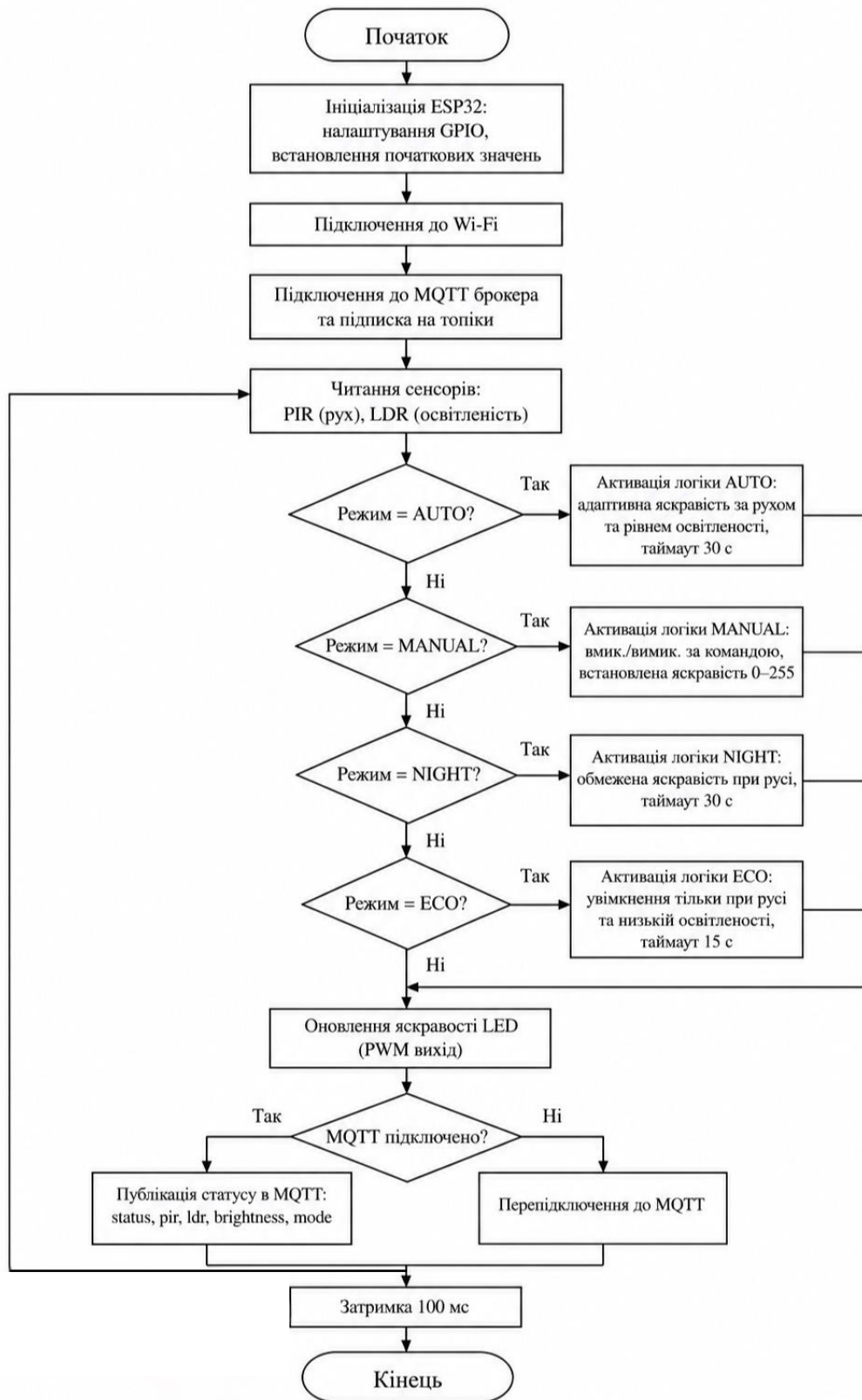


Рисунок 2.2 – Блок-схема роботи алгоритму мікроконтролера

Наступним кроком є підключення до мережі WiFi. Мікроконтролер надсилає запит на підключення до заданої точки доступу та очікує на отримання IP-адреси. Після успішного встановлення WiFi-з'єднання виконується підключення до MQTT-брокера за вказаною адресою та портом. При підключенні мікроконтролер одразу підписується на три командні топіки, такі як `lighting/cmd/mode`, `lighting/cmd/brightness` та `lighting/cmd/power`. Після цього реєструє функцію зворотного виклику, яка буде автоматично викликатися при надходженні нового повідомлення у будь-який з цих топіків.

Основний цикл виконання розпочинається зі зчитування стану датчиків. Цифровий сигнал PIR зчитується безпосередньо з відповідного GPIO-піна і повертає значення HIGH при виявленні руху або LOW за його відсутності. Аналоговий сигнал фоторезистора LDR зчитується через 12-бітний АЦП, що повертає значення у діапазоні від 0 до 4095, яке потім перетворюється на відсоток освітленості у діапазоні від 0 до 100 за допомогою функції масштабування. Зчитані дані надходять до блоку визначення поточного режиму роботи, де система послідовно перевіряє значення змінної режиму і передає керування відповідній гілці логіки.

В автоматичному режимі яскравість визначається адаптивно залежно від двох вхідних параметрів. При виявленому русі та рівні освітленості нижчому за пороговий рівень 30 відсотків система встановлює максимальну яскравість 255 одиниць, забезпечуючи повноцінне освітлення в умовах недостатнього природного світла. Якщо ж рівень природного освітлення перевищує поріг, яскравість штучного освітлення знижується пропорційно показникам LDR за формулою адаптивного розрахунку, але не нижче мінімального порогу 60 одиниць, що гарантує достатню видимість навіть вдень. Після зникнення руху фіксується часова мітка події, і система очікує тридцять секунд перед вимкненням освітлення. Такий порядок дій унеможливує небажане мерехтіння при короткочасному виході людини із зони чутливості датчика.

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

В ручному режимі система повністю ігнорує показники датчиків і встановлює яскравість виключно відповідно до значення, отриманого через топик `lighting/cmd/brightness`, а стан живлення відповідно до команди з топіку `lighting/cmd/power`. Нічний режим обмежує яскравість до фіксованого значення 40 одиниць при виявленні руху, що відповідає приблизно 16 відсоткам від максимуму і є достатнім для комфортного орієнтування у приміщенні без надмірного засвічення. Логіка таймауту вимкнення тривалістю тридцять секунд зберігається і в нічному режимі. Режим енергозбереження є найбільш обмежувальним: освітлення активується лише при одночасному виконанні двох умов, таких як наявності руху та рівня природного освітлення нижчого за 30 відсотків. Яскравість у цьому режимі фіксована на рівні 120 одиниць, а таймаут вимкнення скорочено до п'ятнадцяти секунд для більш агресивної економії електроенергії.

Після розрахунку цільового значення яскравості система передає його до ШІМ-контролера, який негайно оновлює вихідний сигнал на піні LED-освітлення. Далі виконується перевірка стану MQTT-підключення. Якщо підключення активне, мікроконтролер формує JSON-пакет із поточними значеннями всіх параметрів та публікує його у топик `lighting/status`, а також окремі значення у топіки `lighting/pir`, `lighting/ldr`, `lighting/brightness` та `lighting/mode`. У разі виявлення втрати підключення виконується процедура перепідключення до брокера з повторною підпискою на командні топіки, після чого основний цикл продовжується у штатному режимі. Цикл завершується програмною затримкою 100 мс, яка задає частоту опитування датчиків на рівні десяти ітерацій на секунду, після чого виконання повертається на початок циклу.

2.4 Аналіз компонентів апаратної складової

Апаратна складова системи формується навколо мікроконтролерного модуля ESP32 DevKit V1, який є центральним елементом усієї схеми.

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 2.3 – Мікроконтролерний модуль ESP32 DevKit V1

Вибір саме цього модуля обумовлений кількома ключовими чинниками. По-перше, ESP32 має вбудований двоядерний процесор Xtensa LX6 з тактовою частотою до 240 МГц, що є надмірним запасом для задач керування освітленням, однак забезпечує стабільне одночасне виконання мережевого стеку та логіки керування без ризику пропустити критичні події від датчиків. По-друге, наявність вбудованого WiFi-модуля стандарту 802.11 b/g/n дозволяє реалізувати MQTT-комунікацію без будь-яких зовнішніх мережевих компонентів. По-третє, ESP32 має багатоканальний 12-бітний АЦП, необхідний для точного вимірювання аналогового сигналу фоторезистора, та апаратний ШІМ-контролер з підтримкою 8-16 бітної роздільної здатності для плавного регулювання яскравості LED.

Для виявлення присутності людини у приміщенні використовується пасивний інфрачервоний датчик HC-SR501 (рис. 2.4).

Цей датчик реагує на зміну інфрачервоного випромінювання у зоні чутливості, яка охоплює кут близько 120 градусів та відстань до семи метрів. Датчик формує цифровий сигнал HIGH на виході протягом часу, що регулюється потенціометром на його корпусі.



Рисунок 2.4 – Пасивний інфрачервоний датчик HC-SR501

Для підключення до ESP32 використовується будь-який цифровий GPIO-пін у режимі вхідного читання. У середовищі Wokwi замість HC-SR501 використовується ультразвуковий модуль HC-SR04, оскільки власне PIR-датчик відсутній у бібліотеці компонентів симулятора, однак з точки зору програмної логіки обидва компоненти надають ідентичний дискретний сигнал на відповідний GPIO-пін.

Вимірювання рівня природного освітлення здійснюється за допомогою фоторезистора GL5516, опір якого змінюється від кількох сотень Ом при яскравому освітленні до кількох мегаОм у темряві.



Рисунок 2.5 – Фоторезистор GL5516

Для перетворення зміни опору на напругу, придатну для зчитування АЦП мікроконтролера, фоторезистор включається у дільник напруги разом із резистором 10 кОм. Вихід дільника підключається до GPIO34, який налаштований як аналоговий вхід. Отримане 12-бітне значення АЦП перетворюється прошивкою на відсоток освітленості за допомогою функції масштабування.

Для освітлення приміщення у пропонованій програмно-апаратній системі використовується LED-стрічка, що підключається до GPIO25 через силовий MOSFET-транзистор. GPIO25 налаштований як ШІМ-вихід з 8-бітною роздільною здатністю і подає керуючий сигнал на затвор транзистора, який у свою чергу комутує живлення стрічки від зовнішнього джерела. Таким чином забезпечується регулювання яскравості у 256 рівнях без перевантаження виходів мікроконтролера. Для даної роботи було обрано MOSFET-транзистор IRFZ44N. Вибір IRFZ44N обумовлений його низьким опором каналу у відкритому стані, що забезпечує мінімальні втрати потужності та нагрів під час комутації навантаження. Крім того, транзистор має достатній запас за струмом і напругою, що робить його надійним рішенням для керування LED-стрічками середньої та підвищеної потужності.



Рисунок 2.6 – MOSFET-транзистор IRFZ44N

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 34
Зм.	Арк.	№ докум.	Підпис	Дата		

Принцип роботи полягає у тому, що мікроконтролер подає на затвор MOSFET-транзистора імпульсний сигнал із фіксованою частотою 5000 Гц. Яскравість визначається шпаруватістю цього сигналу, тобто співвідношенням тривалості активного імпульсу до загального періоду. При значенні 255 (максимум 8-бітного діапазону) сигнал увесь час HIGH, транзистор повністю відкритий і LED-стрічка світить на повну потужність. При значенні 0, сигнал увесь час LOW, транзистор закритий і стрічка вимкнена. Проміжні значення, наприклад 120, означають, що транзистор відкритий рівно половину кожного періоду, і людське око сприймає це як половинну яскравість, оскільки частота перемикання 5000 Гц значно вища за поріг зорового сприйняття мерехтіння. Таким чином, змінюючи лише цифрове значення шпаруватості від 0 до 255, система забезпечує плавне безступінчасте регулювання яскравості LED-стрічки без жодних втрат потужності, характерних для резистивного методу регулювання.

Схема підключення MOSFET-транзистора IRFZ44N для керування стрічкою світлодіодів у пропонованій програмно-апаратній системі наведено на рис. 2.7.

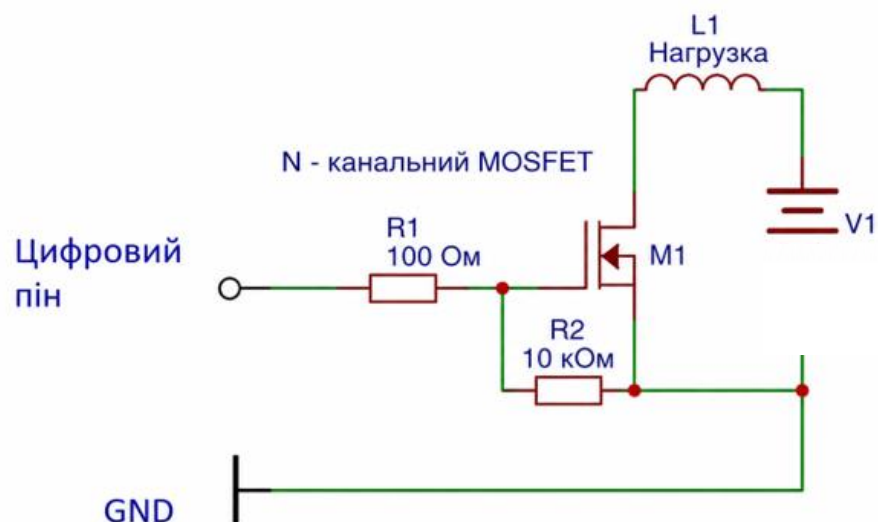


Рисунок 2.7 – Використана схема підключення MOSFET-транзистора 1N5819 для керування стрічкою світлодіодів

Додатково у схемі передбачено зелений LED-діод, підключений до GPIO2 через обмежувальний резистор, який виконує функцію індикатора стану WiFi-підключення: світиться постійно при активному з'єднанні і блимає при спробах відновлення зв'язку.

Регулювання яскравості освітлення здійснюється через апаратний ШІМ (широко-імпульсна модуляція) контролер, вбудований у ESP32. Це дозволяє формувати стабільний високочастотний сигнал без навантаження на мікроконтролер, забезпечуючи плавну зміну яскравості світлодіодної стрічки.

2.5 Оцінка вартості компонентів

Оцінка вартості апаратних компонентів системи проводилася на підставі актуальних роздрібних цін у спеціалізованих магазинах електронних компонентів станом на першу половину 2026 року. Перелік компонентів, їх кількість та вартість наведено у таблиці 2.1.

Загальна вартість компонентів апаратної частини системи складає 379 гривень, що відповідає встановленому бюджетному обмеженню у п'ятсот гривень. Найбільшу частку у структурі вартості займає мікроконтролерний модуль ESP32 DevKit V1 – близько 47 % від загальної суми.

Решта компонентів є доступними масовими виробами, ціна яких не перевищує кількох десятків гривень за одиницю.

Слід зазначити, що наведені ціни є орієнтовними та можуть відрізнятись залежно від конкретного постачальника, способу доставки та курсу валют на момент придбання (оцінка проводилась на початок червня 2026 року).

Практична реалізація проєкту дозволила перевірити схему без закупівлі фізичних компонентів завдяки повноцінній симуляції у середовищі Wokwi, що є додатковою перевагою обраного підходу.

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 – Перелік та вартість компонентів апаратної частини програмно-апаратної системи керування освітленням

Компонент	Кількість	Ціна за од., грн	Вартість, грн
ESP32 DevKit V1	1	180	180
PIR-датчик HC-SR501	1	45	45
Фоторезистор GL5516 + резистор 10 кОм	1	12	12
LED жовтий 5 мм	2	3	6
LED зелений 5 мм (WiFi- індикатор)	1	3	3
Резистор 220 Ом (для LED)	3	1	3
Макетна плата (breadboard 830)	1	55	55
Набір з'єднувальних проводів (dupont)	1 компл.	35	35
Кабель USB–microUSB	1	40	40
Разом			379

2.9 Висновки до другого розділу

У другому розділі виконано проектування програмно-апаратної системи керування освітленням приміщення з датчиком присутності та регулюванням яскравості. Визначено функціональні та технічні вимоги до системи, що стали основою для прийняття конкретних рішень щодо вибору компонентів та архітектури програмного забезпечення.

Розроблена структурна схема системи відображає її поділ на два рівні. Апаратна частина, розташована безпосередньо у приміщенні, включає

мікроконтролер ESP32, PIR-датчик виявлення руху, LDR-датчик рівня освітленості та LED-стрічку як виконавчий елемент. Програмна частина складається з MQTT-брокера Mosquitto, що забезпечує маршрутизацію повідомлень, та Node-RED дашборду, який виконує обробку телеметрії і надає оператору інтерфейс керування. Зв'язок між рівнями здійснюється через протокол MQTT по мережі WiFi, що забезпечує повне розмежування між рівнями збору даних, їх транспортування, обробки та відображення.

Алгоритм функціонування мікроконтролера реалізує чотири режими роботи – автоматичний, ручний, нічний та режим енергозбереження з диференційованою логікою розрахунку яскравості та таймерами автоматичного вимкнення. Регулювання яскравості LED-стрічки здійснюється через апаратний ШІМ-контролер ESP32 та силовий MOSFET-транзистор, що забезпечує плавне керування у 256 рівнях без втрат потужності.

Аналіз апаратних компонентів підтвердив відповідність обраної елементної бази технічним вимогам системи. Оцінка вартості показала, що загальна вартість апаратної частини складає 379 гривень, що не перевищує встановленого бюджетного обмеження і підтверджує економічну доцільність запропонованого рішення.

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

3 СИМУЛЯЦІЯ ПРОГРАМНО-АПАРАТНОЇ СИСТЕМИ ОСВІТЛЕННЯ ПРИМІЩЕННЯ З ДАТЧИКОМ ПРИСУТНОСТІ ТА РЕГУЛЮВАННЯМ ЯСКРАВОСТІ

3.1 Мета і завдання проведення симуляції

Розробка програмно-апаратної системи керування освітленням передбачає не лише проектування схемотехнічної частини та написання прошивки, а й обов'язкове верифікаційне тестування логіки роботи системи до її розгортання на реальному обладнанні. Саме тому в межах даної дипломної роботи передбачено повний цикл симуляції, що охоплює три взаємопов'язані компоненти: цифровий двійник апаратної частини у середовищі Wokwi, програмний симулятор потоку сенсорних даних на мові Python та інформаційну панель реального часу на базі платформи Node-RED.

Метою симуляції була перевірка функціональної коректності системи освітлення в умовах, що були б максимально наближені до реального використання, без залучення фізичного обладнання. Це дозволило виявити логічні помилки у режимах роботи, перевірити правильність передачі даних через протокол MQTT та переконатися у коректній відображенні інформації на дашборді оператора.

Завдання симуляції охоплюють кілька ключових аспектів. По-перше, необхідно відтворити поведінку датчика руху присутності та фоторезистора в умовах динамічної зміни вхідних параметрів. По-друге, перевірити логіку всіх чотирьох режимів роботи системи: автоматичного, ручного, нічного та режиму енергозбереження. По-третє, підтвердити коректність двостороннього обміну даними між симульованим контролером та хмарною частиною системи через MQTT-брокер. По-четверте, валідувати роботу таймера автоматичного вимкнення освітлення після зникнення руху у приміщенні.

Середовище Wokwi є провідним онлайн-симулятором мікроконтролерів, що підтримує повноцінну симуляцію ESP32, включаючи WiFi-стек, периферійні

пристрої та бібліотеки Arduino. Саме завдяки підтримці WiFi та MQTT-бібліотеки PubSubClient стає можливим підключення симульованого мікроконтролера до реального MQTT-брокера та відтворення повного ланцюжка передачі даних. Wokwi дозволяє налаштовувати параметри компонентів у реальному часі, зокрема змінювати рівень освітленості на фоторезисторі та стан цифрового виходу датчика руху, що є критично важливим для тестування реактивної поведінки системи.

Python-симулятор виконує роль програмного замітника апаратної частини у сценаріях, де використання Wokwi є надлишковим або незручним. Він генерує реалістичні потоки сенсорних даних відповідно до математичних моделей денного циклу освітленості та стохастичної моделі присутності людини, після чого публікує їх через MQTT безпосередньо до брокера. Це забезпечило можливість тривалого тестування системи без ручного управління симуляцією.

Node-RED виступає як середовище розробки потоків обробки даних та одночасно як платформа для побудови інформаційної панелі оператора. Завдяки модульній архітектурі потоків Node-RED дозволяє наочно відстежувати маршрути проходження даних від MQTT-брокера до елементів інтерфейсу, що суттєво спрощує відлагодження та верифікацію логіки обробки. Розширення node-red-dashboard надає готові UI-компоненти, такі як gauge-індикатори, графіки, перемикачі та слайдери, які формують повноцінний інтерфейс оператора системи освітлення. Вони й були використані у створенні потоків та інформаційної панелі.

Mosquitto виступає локальним MQTT-брокером, що забезпечує маршрутизацію повідомлень між усіма компонентами системи. Його використання замість хмарного брокера дозволяє проводити симуляцію повністю в автономному режимі, без залежності від доступності мережі Інтернет та без затримок, обумовлених географічною відстанню до серверів.

Технологічний стек симуляції формується таким чином. Апаратний рівень представлений середовищем Wokwi із мікроконтролером ESP32 DevKit V1,

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

ультразвуковим модулем HC-SR04 у ролі замітника PIR-датчика, модулем фоторезистора та світлолідом, що моделює освітлення приміщення. Рівень транспортування даних реалізований протоколом MQTT версії 3.1.1 через брокер Eclipse Mosquitto 2.0. Рівень обробки та візуалізації побудований на Node-RED 3.1 із розширенням node-red-dashboard. Рівень програмного моделювання реалізований скриптом Python 3 із використанням бібліотеки paho-mqtt для взаємодії з брокером.

Схема проведення симуляції наведено на рис. 3.1. Можна відмітити два рівня проведення симуляції, що включали хмарний та локальний рівень (як мок для хмарного рівня).

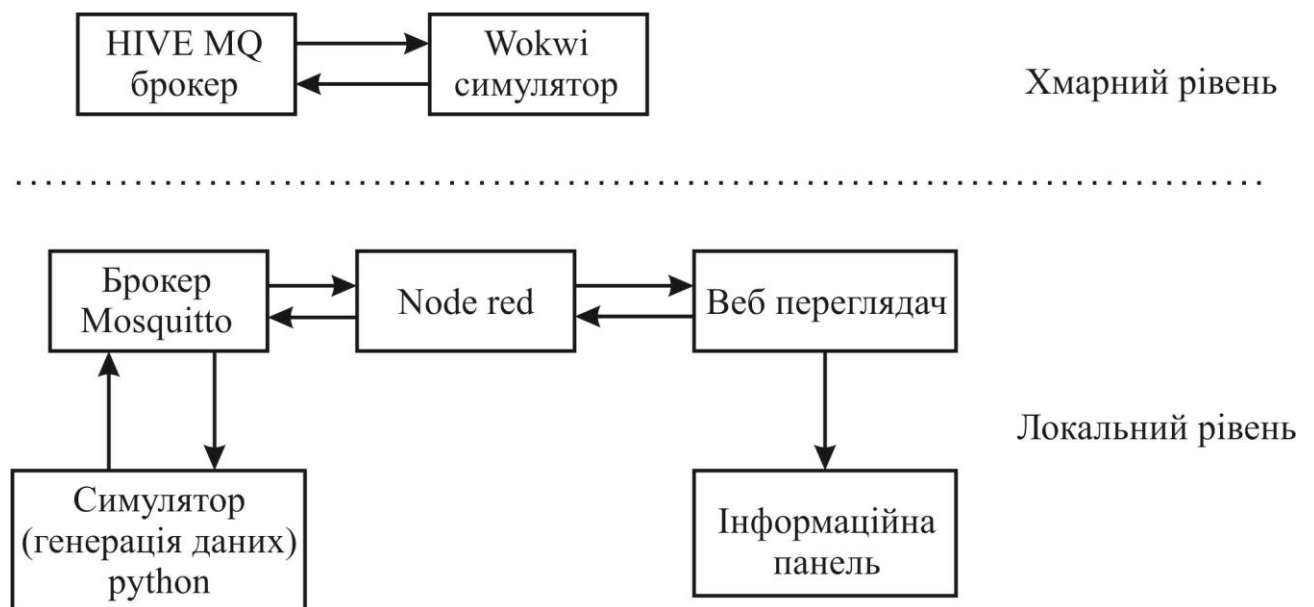


Рисунок 3.1 – Схема проведення симуляції

3.2 Реалізація інформаційної панелі за допомогою Node-RED клієнта

Перед описом побудови інформаційної панелі необхідно розглянути програмний симулятор, що забезпечує генерацію вхідних даних для всієї системи. Симулятор реалізований у вигляді автономного скрипту Python і виконує функцію замітника реального мікроконтролера ESP32 у тих сценаріях тестування, де не потрібна повна апаратна симуляція у Wokwi.

Головним елементом симулятора є математична модель денного циклу природного освітлення. Рівень освітленості приміщення залежить від часу доби і описується синусоїдальною функцією, що відтворює реальний характер зміни сонячного освітлення протягом дня. Значення освітленості E у відсотках обчислюється за формулою, де аргументом є поточний час у годинах:

$$E(t) = \max\left(0, 100 \sin\left(\frac{\pi \cdot (t - 6)}{12}\right)\right) + \varepsilon, \quad (1)$$

де t – поточний час у годинах (від 0 до 24),

ε – випадкова величина рівномірного розподілу на інтервалі $[-5; 5]$, що моделює природні флуктуації освітленості.

Функція визначена лише для годин від 6 до 18, оскільки в нічний час природне освітлення вважається нульовим. Значення обмежується знизу нулем операцією \max , що запобігає появі від'ємних значень при накладанні шуму в граничних умовах.

Модель присутності людини реалізована як стохастичний процес із двома станами. Симулятор відстежує тривалість поточного стану руху за допомогою лічильника `pir_timer`. Якщо лічильник більший за нуль, система перебуває у стані виявленого руху і декрементує лічильник на кожному кроці. Після обнуління лічильника з імовірністю 0,3 генерується нова серія руху тривалістю від 5 до 20 секунд. Така модель відтворює реалістичну поведінку людини у приміщенні із чергуванням активних та пасивних фаз.

Алгоритм розрахунку яскравості ШІМ у симуляторі відповідає логіці прошивки ESP32 і враховує поточний режим роботи, стан PIR-датчика та рівень природного освітлення. В автоматичному режимі при виявленому русі яскравість визначається адаптивно: якщо рівень природного освітлення нижчий за порогове значення у 30 відсотків, система встановлює максимальну яскравість 255 одиниць. При вищому рівні природного освітлення яскравість знижується пропорційно відповідно до виразу:

$$B = \max\left(60, 255 \cdot \left(1 - \frac{E}{200}\right)\right), \quad (1)$$

де B – значення яскравості у діапазоні від 0 до 255, E – поточний рівень природного освітлення у відсотках. Мінімальне обмеження у 60 одиниць забезпечує достатню видимість навіть при яскравому денному освітленні. Після зникнення руху система очікує 30 секунд і лише після закінчення цього таймауту вимикає освітлення, щоб уникнути миготіння при короткочасній відсутності людини у зоні чутливості датчика.

Симулятор підключається до MQTT-брокера за допомогою бібліотеки `raho-mqtt` і кожні дві секунди публікує JSON-пакет у топик `lighting/status`, а також окремі значення у топіки `lighting/pir`, `lighting/ldr`, `lighting/brightness` та `lighting/mode`. Одночасно симулятор підписується на командні топіки `lighting/cmd/mode`, `lighting/cmd/brightness` та `lighting/cmd/power` і реагує на зміни режиму роботи в режимі реального часу. Саме ця двостороння взаємодія перетворює симулятор на повноцінний цифровий замітник мікроконтролера для тестування локальної частини системи.

Перевірка доступності MQTT-брокера проводилась за допомогою утиліти командного рядка `mosquitto_pub`, що входить до складу дистрибутиву Eclipse Mosquitto. Публікація тестових повідомлень у топик `lighting/pir` з дискретними значеннями 0 та 1 підтвердила коректність роботи брокера та правильність підписки Node-RED на відповідний топик (рис. 3.2).

```
C:\Program Files\mosquitto>mosquitto_pub -t "lighting/pir" -m "1"
C:\Program Files\mosquitto>mosquitto_pub -t "lighting/pir" -m "0"
```

Рисунок 3.2 – Тестування брокера Mosquitto через командний рядок

Інформаційна панель системи освітлення розроблена у середовищі Node-RED з використанням розширення `node-red-dashboard`. Панель організована у три

функціональні групи: керування, датчики та графіки, що розміщені горизонтально на вкладці Освітлення (рис. 3.3). Така структура дозволяє оператору одночасно спостерігати за станом системи та здійснювати управляючі впливи без необхідності перемикатися між різними екранами.

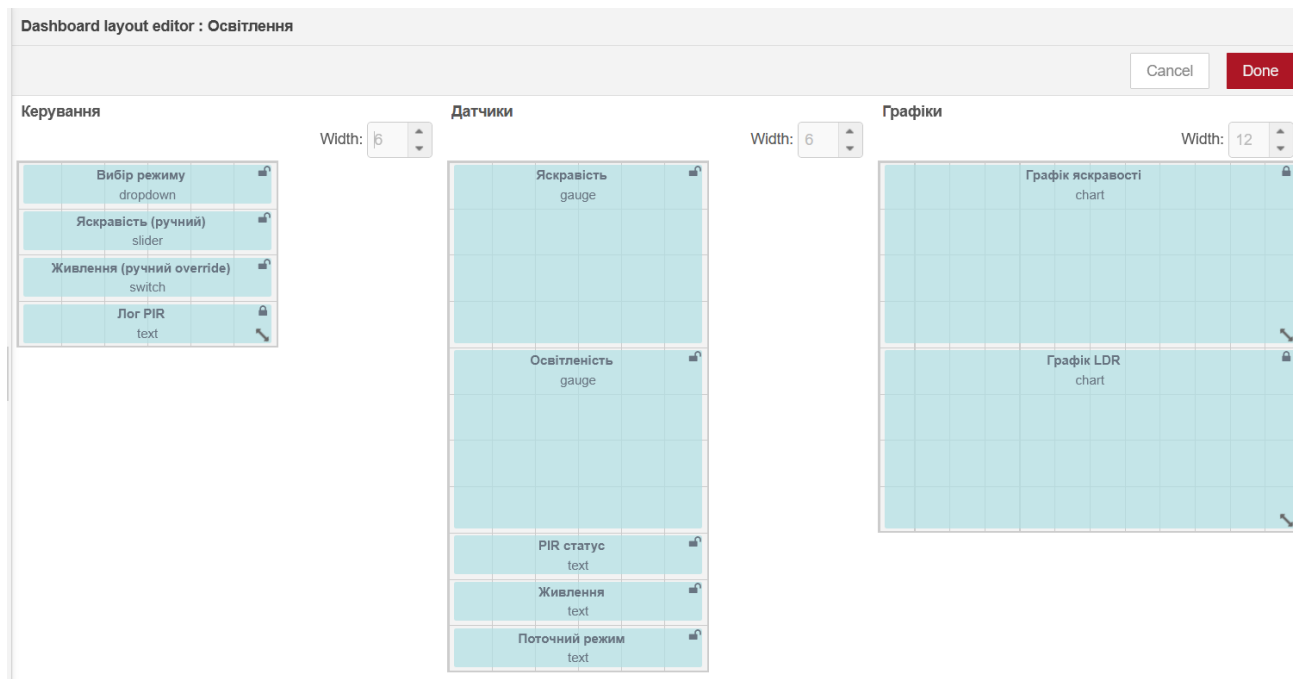


Рисунок 3.3 – Макет інформаційної панелі у редакторі Dashboard Layout Editor

Група Керування містить чотири елементи управління. Випадний список Режим роботи дозволяє обирати один із чотирьох режимів функціонування системи. Слайдер Яскравість забезпечує ручне встановлення рівня яскравості у діапазоні від 0 до 255 одиниць і передає значення у топик `lighting/cmd/brightness`. Перемикач Увімк/Вимк здійснює ручний `override` живлення освітлення і публікує значення `on` або `off` у топик `lighting/cmd/power`. Текстовий елемент Остання подія відображає часову мітку та тип останньої події, отриманої від PIR-датчика.

Група Датчики відображає поточний стан всіх вимірюваних параметрів. Два `gauge`-індикатори показують поточне значення яскравості LED у діапазоні 0–255 та рівень природного освітлення у відсотках від 0 до 100. Три текстові поля відображають відповідно стан датчика руху у вигляді символічних позначень

ТАК або НІ, стан освітлення УВІМК або ВІМК, а також назву поточного активного режиму.

Група Графіки містить два часових графіки лінійного типу. Перший відображає динаміку яскравості LED у часі та дозволяє візуально оцінити характер реакції системи на зміну режимів (рис.3.4). Другий графік показує зміну рівня природного освітлення, що дає змогу аналізувати кореляцію між показниками фоторезистора та поведінкою автоматичного регулювання яскравості.

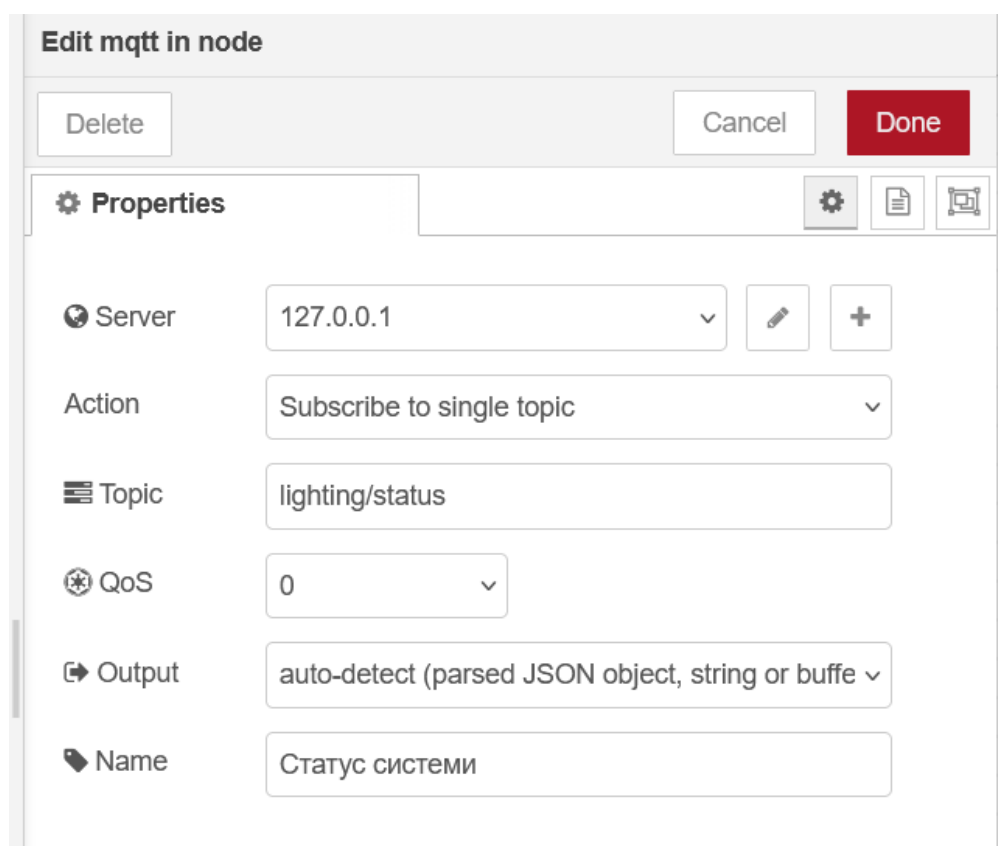


Рисунок 3.4 – Налаштування MQTT вузла Статус системи у Node-RED

Потік обробки даних у Node-RED організований таким чином, що MQTT-вузол Статус системи підписується на топик lighting/status і отримує JSON-пакети від симулятора або від реального ESP32. Вузол налаштований на підключення до брокера за адресою 127.0.0.1 порт 1883, що відповідає локальному екземпляру Mosquitto. Рівень QoS встановлений на значення 0, оскільки для поточкових

телеметричних даних допускається можлива втрата окремих пакетів без суттєвого впливу на якість відображення.

Повна схема потоку обробки даних у клієнті Node-RED наведена на рис. 3.5.

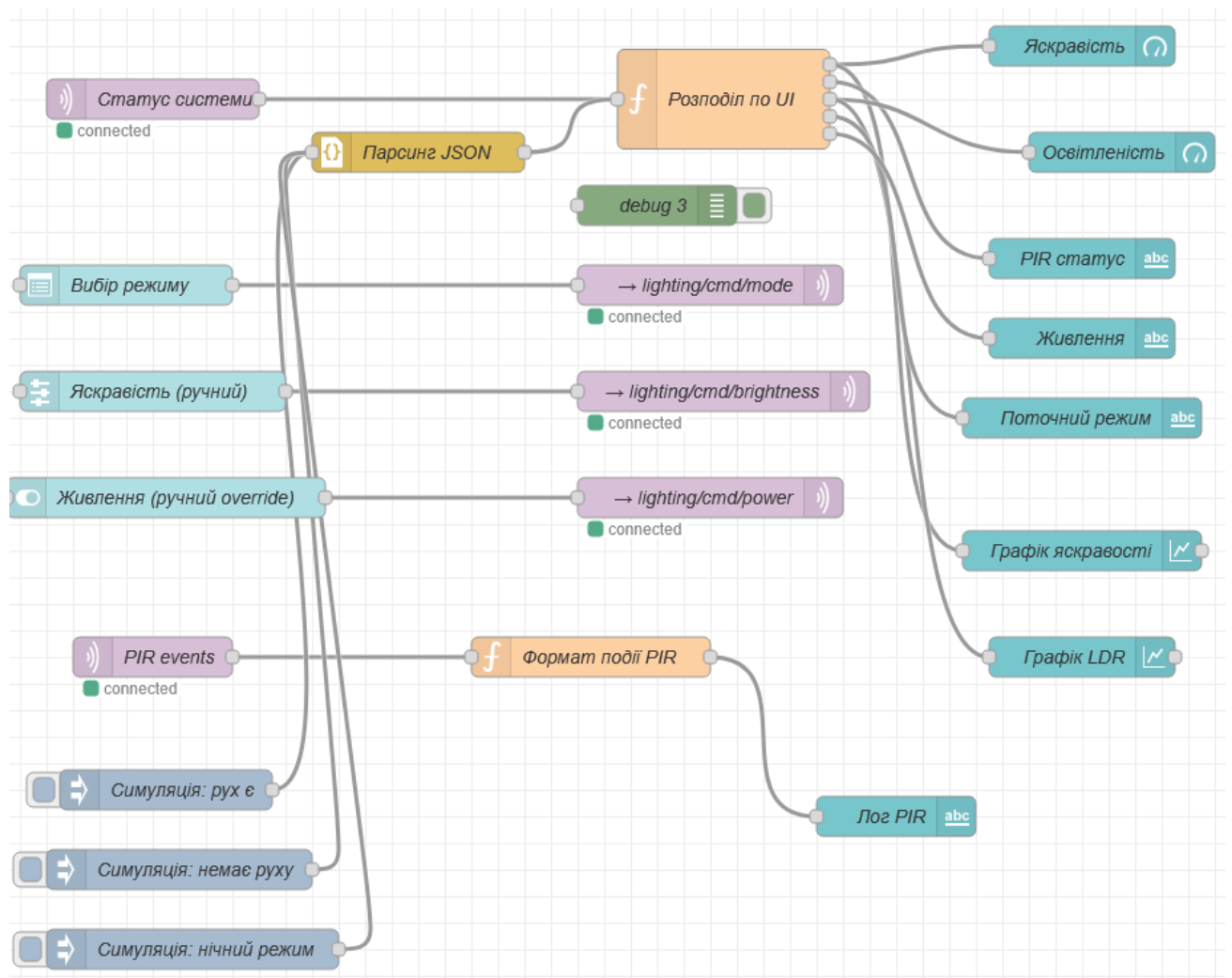


Рисунок 3.5 – Повна схема потоку обробки даних у Node-RED

Функціональний вузол Розподіл по UI отримує розпарсований JSON-об'єкт і формує п'ять паралельних потоків повідомлень для кожного поля пакету. Вузол налаштований на п'ять виходів і використовує масив повідомлень для паралельного надсилання даних до відповідних UI-елементів. Це рішення дозволяє уникнути дублювання підписок на MQTT та забезпечує єдину точку входу для всіх телеметричних даних. Код вузлу реалізований таким чином:

```
node.send([ { payload: msg.payload.brightness }, { payload: msg.payload.pir },  
{ payload: msg.payload.ldr }, { payload: msg.payload.mode }, { payload:  
msg.payload.power ? "ON" : "OFF" } ]);
```

Важливою деталлю реалізації є те, що вузол JSON-парсингу у ланцюжку обробки виявився надлишковим після налаштування MQTT-вузла на режим автоматичного визначення формату "auto-detect (parsed JSON object, string or buffer)". У цьому режимі Node-RED самостійно розпізнає JSON-рядок та перетворює його на об'єкт JavaScript, тому повторний парсинг спричиняв помилку обробки. Після видалення зайвого вузла потік почав коректно передавати дані до всіх елементів інтерфейсу.

Інформаційна панель у початковому стані, до підключення симулятора, відображає нульові значення на всіх індикаторах. Gauge яскравості показує значення 0, gauge природного освітлення показує 0%, стан датчика руху відображається як НІ, стан освітлення як ВИМК. Графіки знаходяться у стані очікування даних. Це підтверджує коректність ініціалізації всіх елементів інтерфейсу (рис. 3.6).

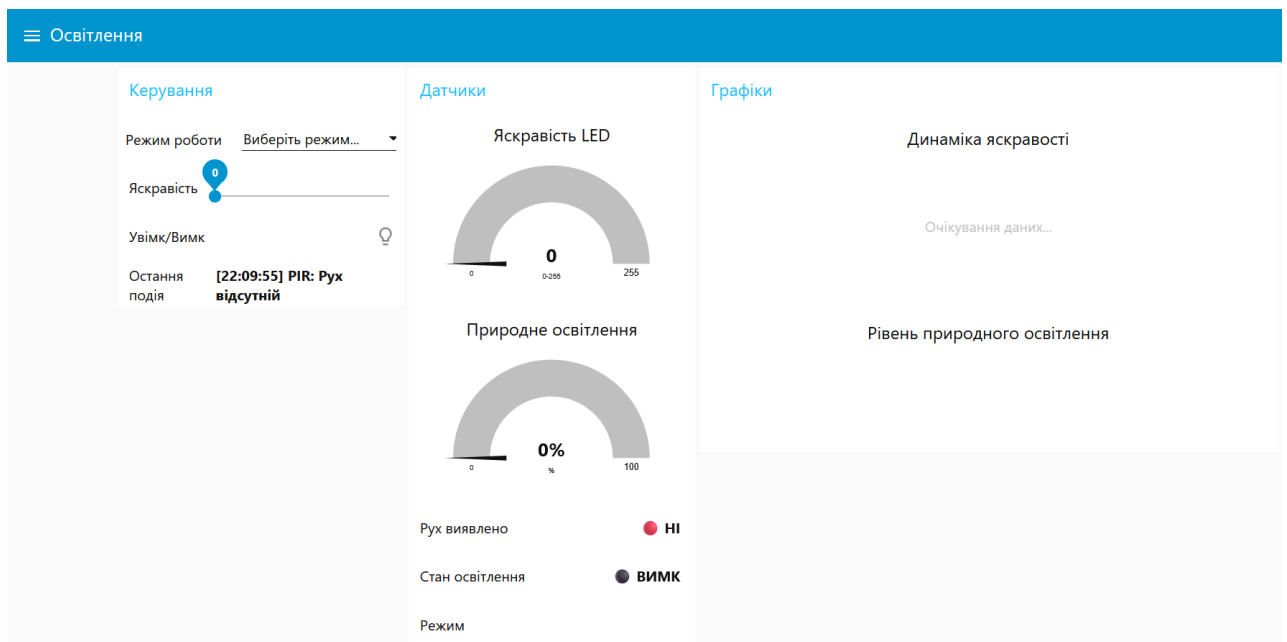


Рисунок 3.6 – Початковий стан інформаційної панелі до підключення симулятора

Після запуску симулятора командою `python simulator.py` з параметром `--broker 127.0.0.1` та `--port 1883` система починала отримувати телеметричні дані і відображати їх на панелі в режимі реального часу. На графіку динаміки яскравості формується характерна ступінчаста крива, що відображає цикли виявлення руху та подальшого вимкнення після таймауту. Рівень природного освітлення в нічний час тестування залишається близьким до нуля, що пояснюється математичною моделлю денного циклу, де після 18 та до 6 години ранку базове значення функції дорівнює нулю.

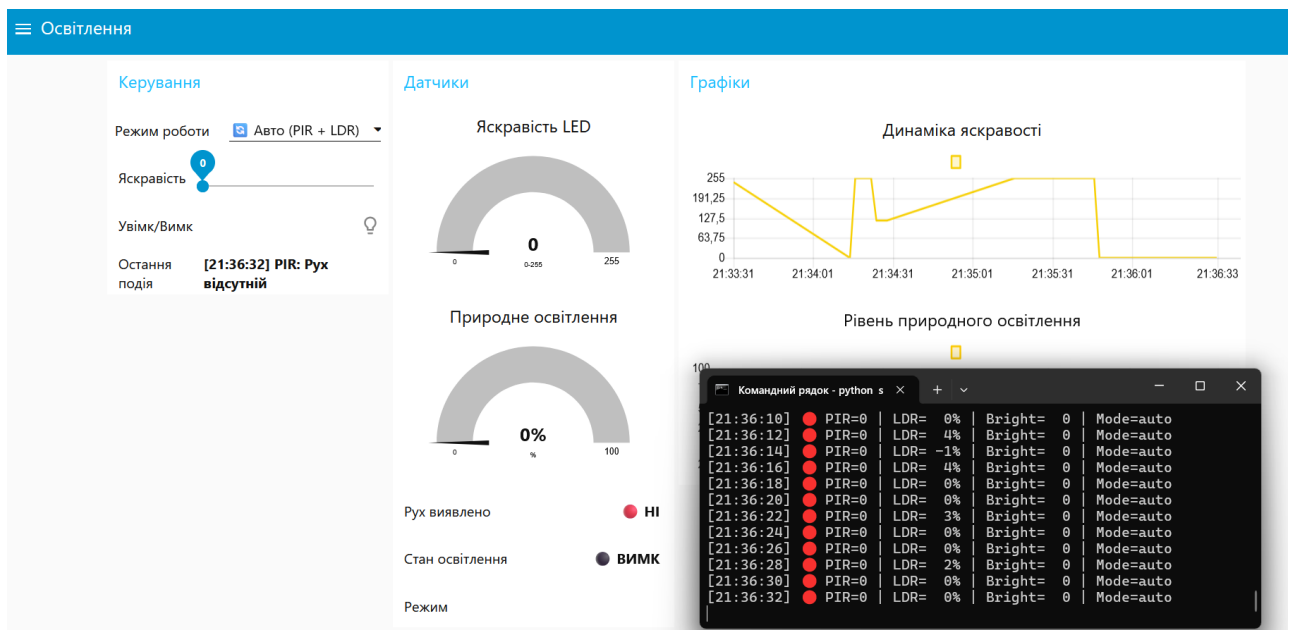


Рисунок 3.7 – Робота системи в автоматичному режимі: дані симулятора та панель

Тестування режимів роботи проводилося послідовно із фіксацією реакції системи на кожну зміну. Для перемикання між режимами використовувався випадний список на панелі керування, що публікує відповідне значення у топик `lighting/cmd/mode`. Симулятор підписаний на цей топик і при отриманні команди оновлює поле `state["mode"]`, що відразу впливає на алгоритм розрахунку яскравості у наступному циклі.

При перемиканні в ручний режим симулятор отримує повідомлення `"[lighting/cmd/mode] manual"` і з наступного кроку починає застосовувати логіку

ручного керування. Автоматична реакція на PIR припиняється, яскравість переходить під повний контроль слайдера та перемикача живлення на панелі. На графіку цей момент відображається характерним спадом яскравості до нуля, оскільки при першому переході в ручний режим живлення вимкнено за замовчуванням (рис. 3.8).

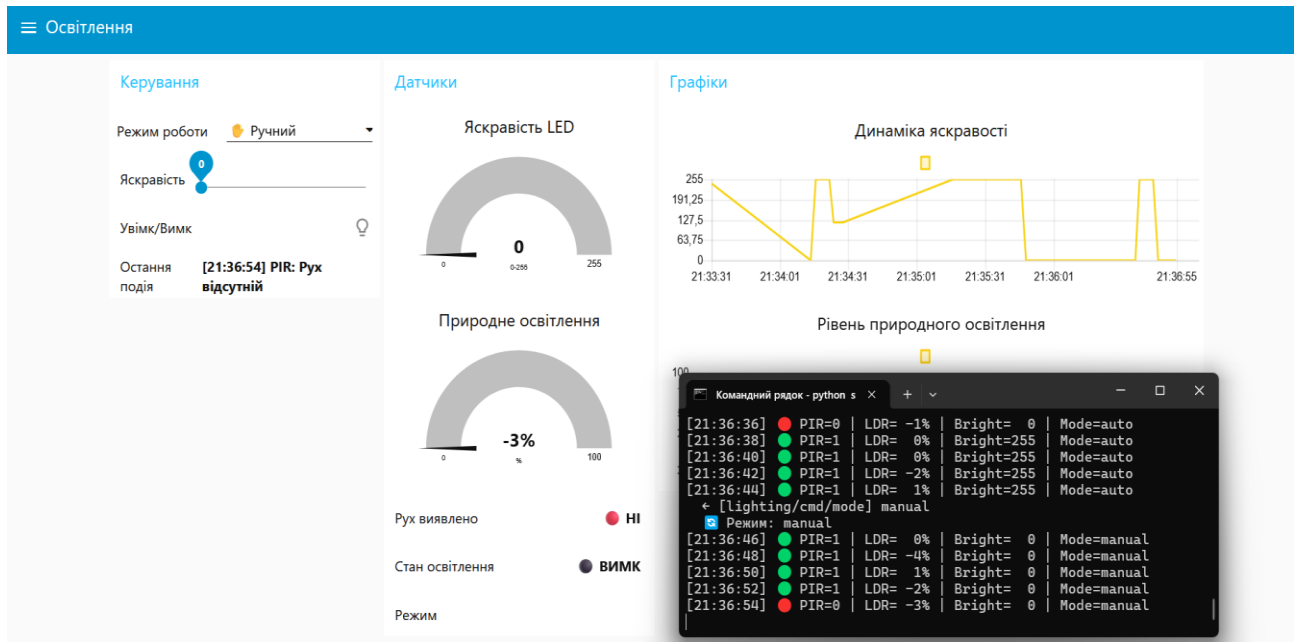


Рисунок 3.8 – Перемикання в ручний режим та реакція системи

Нічний режим активується передачею значення night у командний топик lighting/cmd/mode. Симулятор отримує повідомлення, оновлює внутрішній стан і з наступного циклу починає застосовувати логіку нічного режиму. Система переходить до обмеженої яскравості 40 одиниць при виявленні руху датчиком присутності, що становить приблизно 16 відсотків від максимального рівня 255 одиниць. Таке значення обрано не випадково: воно відповідає комфортному рівню підсвічування для нічного орієнтування у приміщенні, достатньому для безпечного пересування, але без надмірного засвічення, яке могло б порушити сон або дискомфортно вплинути на зір.

На gauge-індикаторі яскравості значення 40 відображається у синій зоні шкали, що наочно підкреслює суттєву відмінність від повного освітлення та дозволяє оператору миттєво ідентифікувати активний нічний режим без

необхідності звертатися до текстового індикатора режиму. Таймер автоматичного вимкнення продовжує діяти і в нічному режимі: після зникнення руху система очікує 30 секунд і лише після спрацювання таймауту вимикає освітлення, що запобігає дратівливому миготінню при короткочасному виході людини з зони чутливості датчика.

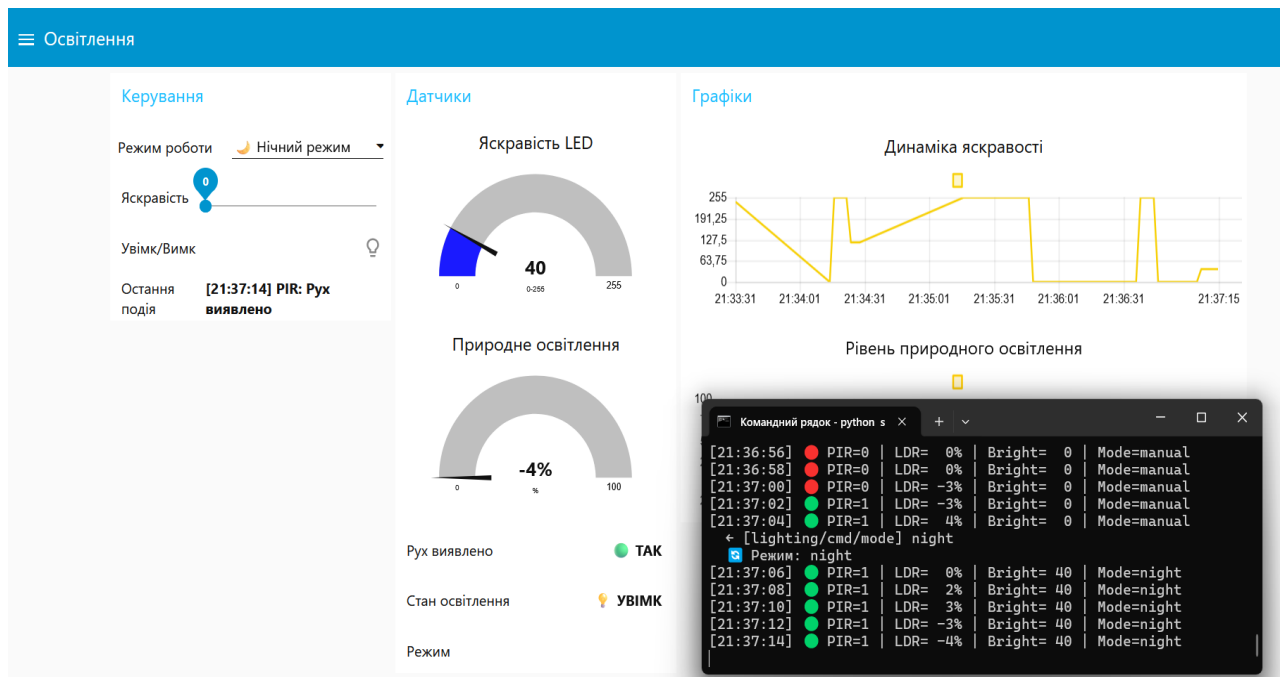


Рисунок 3.9 – Нічний режим (яскравість обмежена до 40 одиниць)

Режим енергозбереження відрізняється від нічного додатковою умовою активації: освітлення вмикається лише при одночасному виконанні двох умов – наявності руху та низькому рівні природного освітлення, що не перевищує порогового значення у 30 відсотків. Яскравість у цьому режимі фіксована на рівні 120 одиниць, що забезпечує достатнє освітлення при мінімальному енергоспоживанні.

Таймаут вимкнення скорочений до 15 секунд для більш агресивної економії електроенергії. Можна побачити, що на gauge-індикаторі значення освітленості на рівні 120 відображається в помаранчевій зоні, що сигналізує про проміжний рівень.

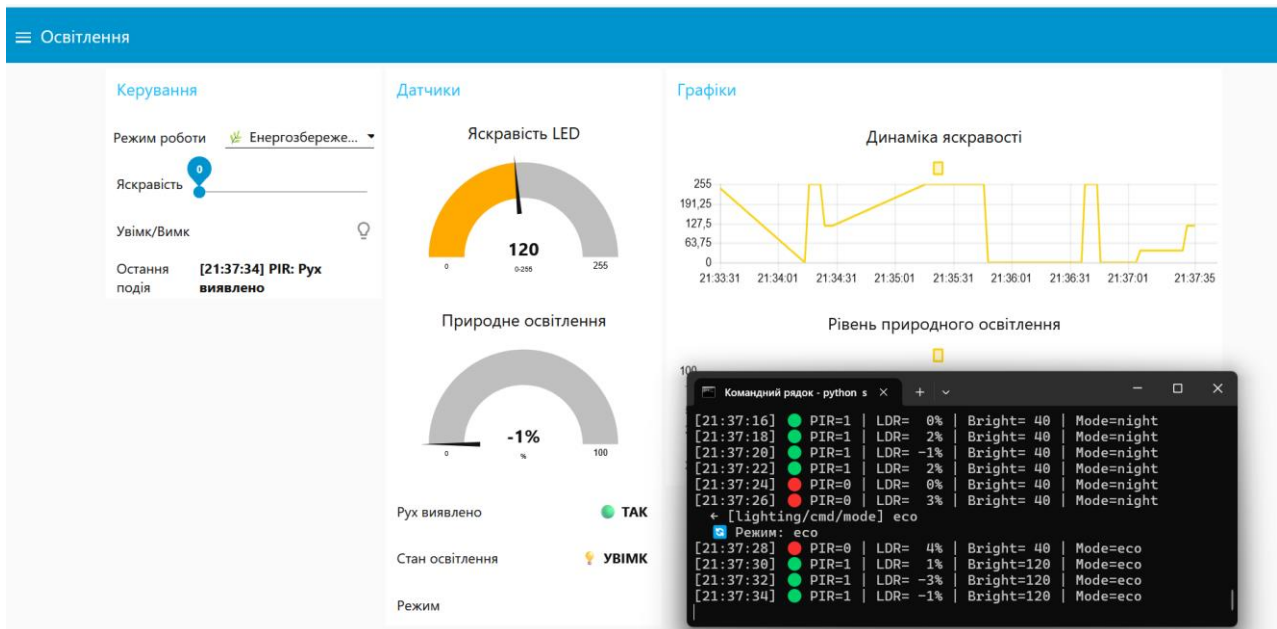


Рисунок 3.10 – Режим енергозбереження

Тестування ручного управління яскравістю проводилось в умовах активного режиму енергозбереження. Переміщення слайдера яскравості генерувало повідомлення у топик `lighting/cmd/brightness` зі значеннями 107, 111 та 151, що видно з логу симулятора. Проте gauge-індикатор яскравості не змінював свого значення, залишаючись на рівні 120, оскільки симулятор зберігав режим енергозбереження і ігнорував командні значення яскравості поза ручним режимом. Це підтвердило коректну пріоритизацію режимів у логіці симулятора.

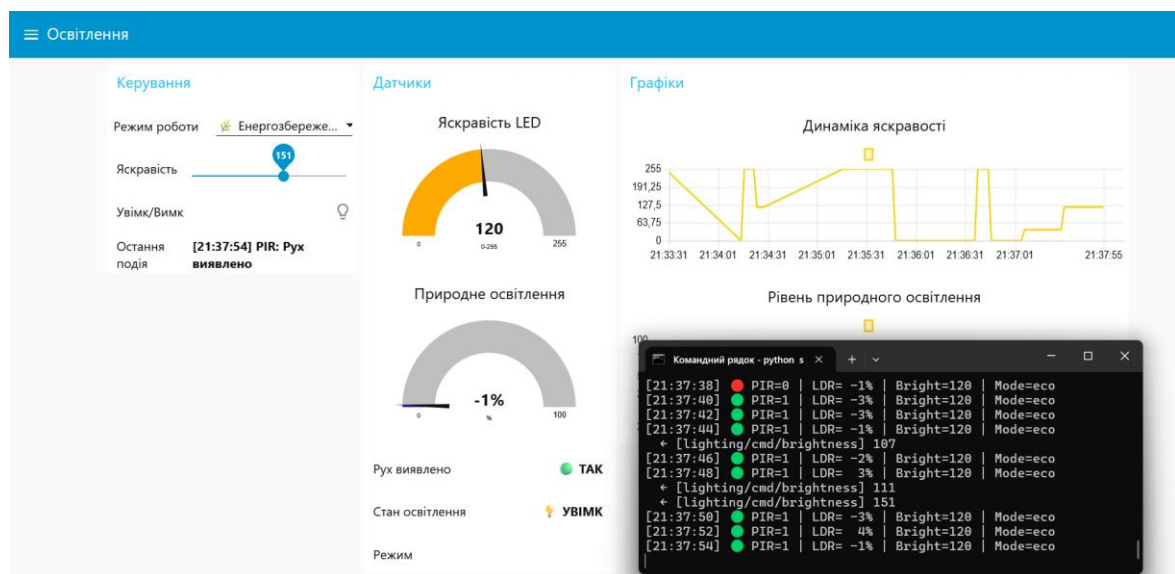


Рисунок 3.11 – Тестування слайдера яскравості

Після перемикання в ручний режим та активації живлення перемикачем Увімк/Вимк система почала відображати задані значення яскравості. Симулятор отримав послідовно команди `lighting/cmd/power` зі значенням `on` та `lighting/cmd/mode` зі значенням `manual`, після чого gauge яскравості відобразив значення 151, що відповідало положенню слайдера. Це підтвердило повноцінну роботу ланцюжка передачі команд від панелі до симулятора та назад.

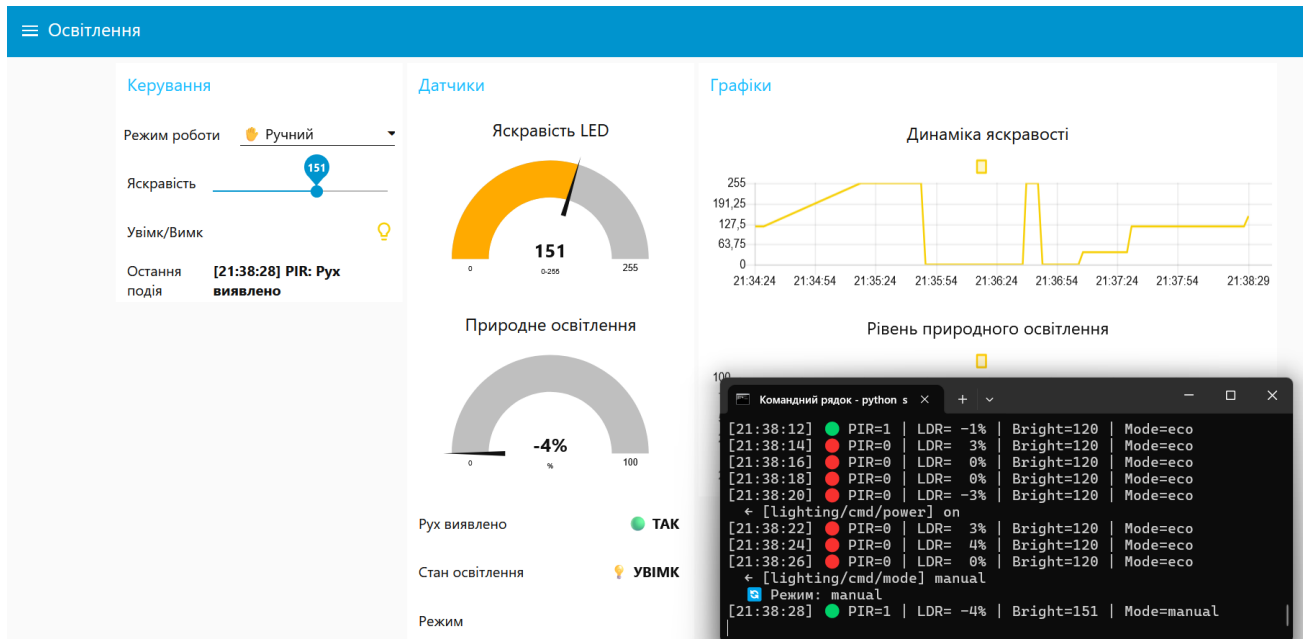


Рисунок 3.12 – Ручний режим з увімкненим живленням та яскравістю 151

Зведений графік динаміки яскравості за весь сеанс тестування подано на рис. 3.13. Він демонструє характерну картину послідовної зміни режимів. На початку сеансу видно ділянку автоматичного режиму з яскравістю 255 при виявленому русі. Подальший різкий спад до нуля відповідає моменту перемикання в ручний режим з вимкненим живленням. Короткий пік до 255 відображає короткочасне спрацювання у попередньому режимі перед перемиканням. Ступінчастий підйом до 40, а потім до 120 відображає послідовну активацію нічного режиму та режиму енергозбереження. Фінальне зростання до 127 та вище відображає ручне керування яскравістю через слайдер.



Рисунок 3.13 – Зведений графік динаміки яскравості за весь сеанс тестування

Рівень природного освітлення на нижньому графіку залишається стабільно близьким до нуля протягом усього сеансу тестування. Це пояснюється тим, що тестування проводилось у нічний час доби, тому математична модель денного циклу генерує значення в межах випадкового шуму ϵ навколо нуля. Окремі незначні підйоми до 1–4 відсотків обумовлені саме шумовою складовою функції освітленості. Для повноцінного тестування адаптивного зниження яскравості в автоматичному режимі слід запускати симулятор у денний час або вручну встановити значення `ldr` через ін'єкційний вузол Node-RED.

3.3 Створення цифрового двійника програмно-апаратної системи у Wokwi

Цифровий двійник програмно-апаратної системи освітлення реалізований у середовищі онлайн-симуляції Wokwi, яке забезпечує повноцінне відтворення поведінки мікроконтролера ESP32 разом із підключеними периферійними пристроями. На відміну від програмного симулятора, цифровий двійник у Wokwi

виконує реальний машинний код прошивки, що дозволяє верифікувати не лише логіку алгоритму, а й коректність роботи бібліотек, таймінгів та апаратних периферій мікроконтролера.

Середовище Wokwi обране з кількох причин. По-перше, воно підтримує повноцінну симуляцію ESP32 DevKit V1 із реальним WiFi-стеком, що дозволяє підключатися до MQTT-брокера через мережу. По-друге, Wokwi надає бібліотеку компонентів, достатню для моделювання всіх елементів схеми системи освітлення. По-третє, проєкт Wokwi зберігається у форматі JSON та може бути відтворений без встановлення додаткового програмного забезпечення, що суттєво спрощує відтворюваність результатів.

Схема цифрового двійника системи освітлення у середовищі Wokwi подана на рис. 3.14.

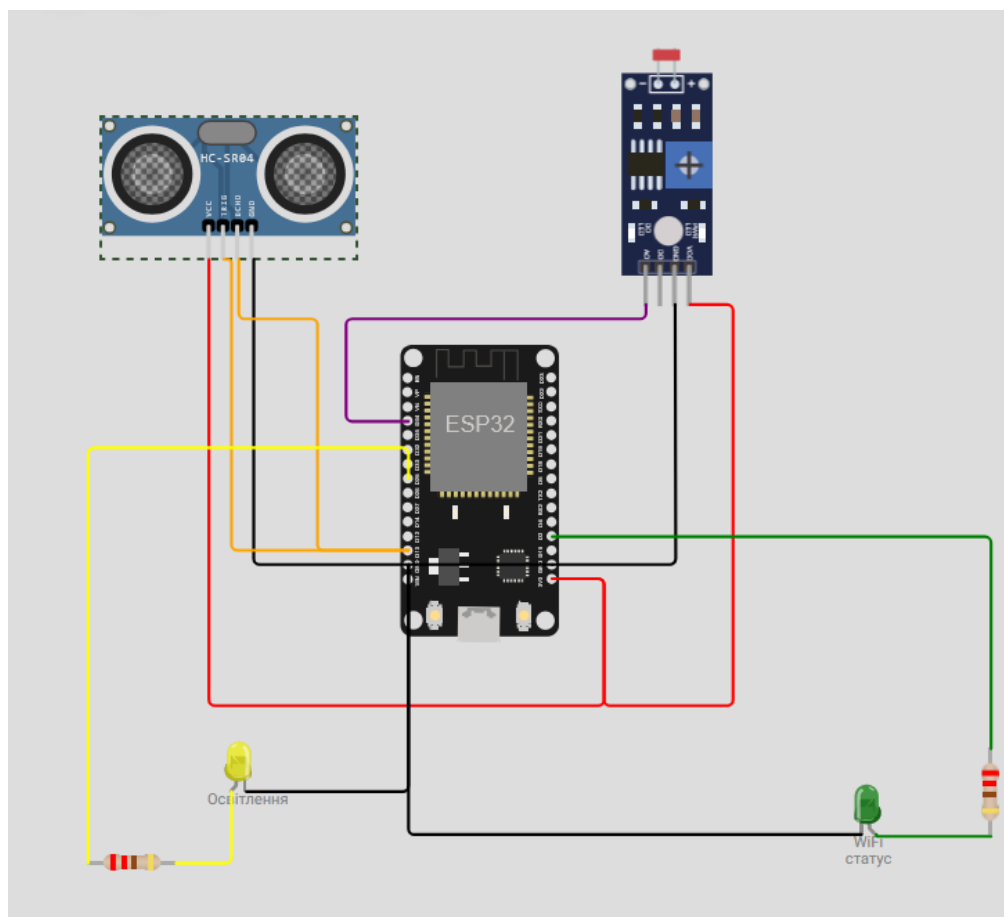


Рисунок 3.14 – Схема цифрового двійника системи освітлення у середовищі Wokwi

Зм.	Арк.	№ докум.	Підпис	Дата

Схема цифрового двійника включає мікроконтролер ESP32 DevKit V1 як центральний елемент, до якого підключені чотири периферійні пристрої. Модуль HC-SR04 виконує роль функціонального заміника PIR-датчика присутності, оскільки власне PIR-датчик відсутній у бібліотеці компонентів Wokwi. З точки зору програмного забезпечення обидва компоненти надають дискретний сигнал на GPIO-пін мікроконтролера, тому прошивка не потребує змін для роботи з обома варіантами датчика. Модуль фоторезистора підключений до аналогового входу GPIO34 через дільник напруги і забезпечує аналоговий сигнал пропорційний рівню освітленості. Жовтий LED-діод імітує основне освітлення приміщення і підключений через обмежувальний резистор 220 Ом до GPIO25, що налаштований як PWM-вихід. Зелений LED-діод виконує функцію індикатора стану WiFi-підключення і підключений до GPIO2 через аналогічний резистор.

Прошивка мікроконтролера реалізована на мові C++ із використанням фреймворку Arduino для ESP32. Вона включає три основні функціональні блоки: підсистему управління WiFi та MQTT-підключенням, підсистему зчитування сенсорів та підсистему реалізації режимів освітлення. Бібліотека PubSubClient забезпечує MQTT-комунікацію, бібліотека ArduinoJson використовується для формування та розбору JSON-пакетів телеметрії. Апаратний ШІМ реалізований через функції ledcSetup та ledcWrite фреймворку ESP32 Arduino, що дозволяє керувати яскравістю LED з роздільною здатністю 8 біт.

Налаштування ШІМ здійснюється під час ініціалізації системи. Канал ШІМ номер 0 налаштовується на частоту 5000 Гц із 8-бітною роздільною здатністю, що забезпечує 256 рівнів яскравості від 0 до 255. Частота 5 кГц обрана з міркувань відсутності видимого мерехтіння, яке сприймається людським оком при частотах нижче 50–100 Гц, та відсутності акустичного шуму від котушок при частотах у чутному діапазоні від 20 до 20 000 Гц. Зчитування АЦП фоторезистора здійснюється з 12-бітною роздільною здатністю, що дає значення

від 0 до 4095, які перетворюються на відсотки освітленості за допомогою функції `map`.

Логіка автоматичного режиму в прошивці реалізована ідентично до логіки Python-симулятора, що забезпечило узгоджену поведінку обох компонентів системи. При виявленні руху через давач руху або HC-SR04 система фіксувала часову мітку у змінній `lastMotionTime`. Якщо рівень природного освітлення нижчий за порогове значення `LDR_DARK_THRESHOLD` рівне 30 відсоткам, встановлюється максимальна яскравість. В іншому випадку яскравість розраховується адаптивно пропорційно рівню `LDR`. Після зникнення руху перевіряється різниця між поточним часом і часом останнього виявленого руху. Якщо вона перевищує константу `AUTO_TIMEOUT_MS`, яка рівна 30 000 мілісекундам, яскравість встановлюється на нуль.

Публікація телеметрії у Wokwi відбувається кожні дві секунди через MQTT-брокер `broker.hivemq.com`, що є публічним безкоштовним брокером із підтримкою Wokwi-симуляції. Це продиктовано технічним обмеженням Wokwi: симульований мікроконтролер не має прямого доступу до локального мережевого адаптера хоста, тому для зв'язку з локальним Mosquitto необхідно використовувати тунелювання або перенаправлення портів через `wokwi.toml`. Використання публічного брокера спрощує налаштування і є прийнятним для симуляційних цілей.

Відповідність цифрового двійника реальній системі забезпечується на кількох рівнях. На рівні схемотехніки використовуються ті самі номінали резисторів та ті самі GPIO-піни, що передбачені для реальної збірки.

На рівні програмного забезпечення виконується ідентичний бінарний код прошивки. На рівні протоколів використовується той самий набір MQTT-топиків та формат JSON-пакетів, що й у Node-RED дашборді. Це забезпечило взаємозамінність між режимом симуляції та режимом роботи з реальним обладнанням без будь-яких змін у коді серверної частини.

Цифровий двійник демонструє поведінку системи в умовах, що відрізняються від тих, що моделюються Python-симулятором. Зокрема, у Wokwi можна вручну змінювати інтенсивність світла на фоторезисторі в режимі реального часу та спостерігати миттєву реакцію PWM-сигналу на LED. Також можна перевірити поведінку системи при нестабільному WiFi-з'єднанні, симулюючи переривання мережі та верифікуючи логіку автоматичного перепідключення до MQTT-брокера, реалізовану у функції connectMQTT.

3.6 Висновки до третього розділу

У третьому розділі виконано симуляцію програмно-апаратної системи освітлення приміщення з датчиком присутності та регулюванням яскравості. Симуляція охоплювала три взаємодоповнювальні компоненти: Python-симулятор, інформаційну панель Node-RED та цифровий двійник у Wokwi.

Розроблений Python-симулятор забезпечує моделювання двох ключових фізичних процесів. Математична модель денного циклу на основі синусоїдальної функції відтворює характер зміни природного освітлення протягом доби з урахуванням випадкових флуктуацій. Стохастична модель присутності людини генерує реалістичні послідовності виявлення руху з довільною тривалістю та паузами між сеансами активності. Разом ці моделі забезпечили тестове навантаження, яке було достатнім для перевірки всіх гілок логіки системи.

Інформаційна панель Node-RED успішно реалізована із повним набором необхідних елементів управління та відображення. Двосторонній зв'язок між панеллю та симулятором підтверджено в ході тестування: панель коректно відображає телеметрію від симулятора, а команди від панелі негайно змінюють поведінку симулятора. Всі чотири режими роботи перевірено з фіксацією характерних значень яскравості: автоматичний режим з яскравістю до 255 одиниць, ручний режим з довільним встановленням від 0 до 255, нічний режим з

фіксованою яскравістю 40 одиниць та режим енергозбереження з яскравістю 120 одиниць.

Цифровий двійник у Wokwi відтворює апаратну частину системи на рівні виконання реального машинного коду прошивки. Схема включає ESP32 DevKit V1, датчик присутності на базі HC-SR04, модуль фоторезистора для вимірювання природного освітлення та два LED-діоди для індикації освітлення та стану WiFi-підключення. Прошивка реалізує всі чотири режими роботи, двосторонній MQTT-зв'язок та адаптивне регулювання яскравості залежно від показників LDR.

В ході симуляції виявлено та усунуто ряд технічних особливостей реалізації. Встановлено, що MQTT-вузол Node-RED у режимі автоматичного розпізнавання формату самостійно перетворює JSON-рядки на об'єкти JavaScript, що робить зайвим окремий вузол парсингу. Підтверджено, що Node-RED коректно резолвує лише адресу 127.0.0.1 для локального брокера на Windows через особливості резолюції IPv4 та IPv6. Виявлено, що значення яскравості в автоматичному режимі залишаються максимальними в нічний час тестування через природне обнуління функції освітленості, що є коректною поведінкою системи.

Проведена симуляція підтвердила функціональну відповідність розробленої системи поставленим вимогам. Система коректно реагує на зміну присутності людини, адаптує яскравість до рівня природного освітлення, обробляє команди управління від оператора та відображає актуальний стан у реальному часі на інформаційній панелі. Реалізований комплекс симуляційних інструментів може бути безпосередньо використаний при переході до тестування на реальному апаратному забезпеченні без внесення змін у серверну та клієнтську частини системи.

ВИСНОВКИ

У результаті виконання дипломної роботи здійснено проектування, реалізацію та симуляційне тестування програмно-апаратної системи освітлення приміщення з датчиком присутності та регулюванням яскравості, що повністю відповідає поставленій меті та сформульованим завданням.

У першому розділі роботи проведено аналіз предметної області та розглянуто існуючі підходи до автоматизації систем освітлення приміщень. Встановлено, що поєднання логіки присутності з адаптивним регулюванням яскравості забезпечує суттєву перевагу порівняно з традиційними системами дискретного увімкнення та вимкнення світла, оскільки дозволяє підтримувати комфортний рівень освітлення лише за фактичної потреби, мінімізуючи непродуктивні витрати електроенергії.

У другому розділі сформульовано перелік функціональних та нефункціональних вимог до системи. Визначено логіку чотирьох режимів роботи, що включали автоматичний, ручний, нічний та режим енергозбереження, кожен з яких реалізує власну комбінацію реакції на датчик присутності та рівень природного освітлення. Розроблено структурну схему системи, що відображає узгоджену взаємодію апаратного рівня на базі ESP32 з датчиками та світлодіодним освітленням, мережевого рівня на основі MQTT-брокерів Mosquitto та HiveMQ, а також програмного рівня, реалізованого засобами платформи Node-RED. Обґрунтовано вибір апаратних компонентів.

Розроблено алгоритм функціонування мікроконтролера у вигляді блок-схеми, що відображає послідовність ініціалізації апаратних ресурсів, підключення до мережі Wi-Fi та MQTT-брокера, а також каскадну логіку перевірки активного режиму роботи в основному циклі. На основі цього алгоритму реалізовано прошивку мовою C++ у середовищі Arduino, яка забезпечує читання показань датчиків присутності та освітленості, прийняття рішень щодо яскравості освітлення відповідно до активного режиму та

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 59
Зм.	Арк.	№ докум.	Підпис	Дата		

публікацію телеметричних даних через протокол MQTT з періодичністю, достатньою для відображення динаміки процесів на інформаційній панелі.

У третьому розділі реалізовано програмний симулятор сенсорних даних мовою Python, що відтворює математичну модель денного циклу природного освітлення на основі синусоїдальної залежності та стохастичну модель присутності людини з випадковою тривалістю активних і пасивних фаз. Це дозволило здійснити тривалу верифікацію логіки системи без використання фізичного обладнання. Спроектовано та реалізовано цифровий двійник апаратної частини у середовищі Wokwi, що відтворює схему підключення мікроконтролера ESP32 до датчика присутності, фоторезистора та світлодіодного освітлення з використанням реального машинного коду прошивки. Розроблено інформаційну панель оператора на платформі Node-RED, що включає елементи керування режимами роботи, слайдер яскравості, перемикач живлення, gauge-індикатори стану датчиків та графіки динаміки яскравості й рівня природного освітлення в реальному часі.

Проведене симуляційне тестування підтвердило коректність роботи чотирьох режимів освітлення. Зафіксовано характерні значення яскравості для кожного режиму: максимальна яскравість 255 одиниць в автоматичному режимі при виявленому русі та низькому рівні природного освітлення, довільне встановлення яскравості в ручному режимі, обмежена яскравість 40 одиниць у нічному режимі та яскравість 120 одиниць у режимі енергозбереження. Підтверджено двосторонній обмін даними між програмним симулятором, MQTT-брокером Mosquitto та інформаційною панеллю Node-RED: команди оператора, надіслані через панель, негайно змінювали поведінку симулятора, а телеметричні дані симулятора в реальному часі відображалися на gauge-індикаторах та графіках панелі.

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Створення системи управління автономним освітленням: комфорт та економія у кожному будинку. *Ledstorm*. URL: <https://ledstorm.ua/ua/statti/stvorennja-sistemi-keruvannja-avtonomnim-osvitlennjam/> (дата звернення: 14.06.2026).
2. Системи керування освітленням: види та призначення. *MiLight*. URL: <https://milight.com.ua/systemy-keruvannia-osvitlenniam-vydy-ta-pryznachennia/> (дата звернення: 14.06.2026).
3. Nair G. B., Dhoble S. J. The Fundamentals and Applications of Light-Emitting Diodes. Cambridge: Woodhead Publishing, 2020. P. 253–270.
4. Tran D., Tan Y. K. Sensorless illumination control of a networked LED-lighting system using feedforward neural network. *IEEE Trans. Ind. Electron.* 2014. Vol. 61. P. 2113–2121.
5. Lohaus L., Rossius A., Dietrich S., Wunderlich R., Heinen S. A Dimmable LED Driver With Resistive DAC Feedback Control for Adaptive Voltage Regulation. *IEEE Trans. Ind. Appl.* 2015. Vol. 51. P. 3254–3262.
6. Boscarino G., Moallem M. Daylighting Control and Simulation for LED-Based Energy-Efficient Lighting Systems. *IEEE Trans. Ind. Inf.* 2016. Vol. 12. P. 301–309.
7. Wagiman K. R., Abdullah M. N., Hassan M. Y., Mohammad Radzi N. H., Abu Bakar A. H., Kwang T. C. Lighting system control techniques in commercial buildings: Current trends and future directions. *J. Build. Eng.* 2020. Vol. 31. P. 101342.
8. Byun J., Hong I., Lee B., Park S. Intelligent Household and LED Lighting and System and Considering Energy and Efficiency and User and Satisfaction. *IEEE Trans. Consum. Electron.* 2013. Vol. 59. P. 70–76.
9. Mahdavi A., Mathew P., Kumar S., Hartkopf V., Loftness V. Effects of Lighting, Zoning, and Control Strategies on Energy Use in Commercial Buildings. *J. IES.* 1995. Vol. 24. P. 25–35.

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

10. Gilani S., O'Brien W. A preliminary study of occupants use of manual lighting controls in private offices: A case study. *Energy Build.* 2018. Vol. 159. P. 572–586.

11. ESP32 Web Server: Charts with Historical Data (load .csv file). *Random Nerd Tutorials*. URL: <https://randomnerdtutorials.com/esp32-web-server-charts-historical-data/> (дата звернення: 14.06.2026).

12. Інструкція по підключенню ультразвукового датчика HC-SR04 з Arduino. *Мій Проект*. URL: <https://myproject.com.ua/instruktsiia-po-vykorystanni-u-ultrazvukovoho-datchyka-hc-sr04-z-arduino.html> (дата звернення: 14.06.2026).

13. Bao J., Song X., Li Y., Bai Y., Zhou Q. Effect of lighting illuminance and colour temperature on mental workload in an office setting. *Sci. Rep.* 2021. Vol. 11. P. 15284.

14. Chen R., Tsai M., Tsay Y. Effect of Color Temperature and Illuminance on Psychology, Physiology, and Productivity: An Experimental Study. *Energies*. 2022. Vol. 15. P. 4477.

15. Górecki K., Ptak P., Wnuczko S. The Embedded System to Control the Illuminance of an Office Workplace with LED Light Sources. *Energies*. 2022. Vol. 15. P. 2406.

16. Kandasamy N. K., Karunagaran G., Spanos C., Tseng K. J., Soong B. H. Smart lighting system using ANN-IMC for personalized lighting control and daylight harvesting. *Build. Environ.* 2018. Vol. 139. P. 170–180.

17. Jia B., Li W., Chen G., Sun W., Wang B., Xu N. Optimized Design of Skylight Arrangement to Enhance the Uniformity of Indoor Sunlight Illumination. *Sustainability*. 2023. Vol. 15. P. 11257.

18. Cuttle C. A fresh approach to interior lighting design: The design objective—direct flux procedure. *Light. Res. Technol.* 2018. Vol. 50. P. 1142–1163.

19. Peña-García A., Salata F. Indoor Lighting Customization Based on Effective Reflectance Coefficients: A Methodology to Optimize Visual Performance and Decrease Consumption in Educative Workplaces. *Sustainability*. 2021. Vol. 13. P. 119.

					КВРКІ. 022080.22.01.02 ПЗ	Арк. 62
Зм.	Арк.	№ докум.	Підпис	Дата		

20. Park J. Y., Dougherty T., Fritz H., Nagy Z. LightLearn: An adaptive and occupant centered controller for lighting based on reinforcement learning. *Build. Environ.* 2019. Vol. 147. P. 397–414.

21. Golasi I., Salata F., de LietoVollaro E., Peña-García A. Influence of lighting colour temperature on indoor thermal perception: A strategy to save energy from the HVAC installations. *Energy Build.* 2019. Vol. 185. P. 112–122.

22. Minoli D., Sohraby K., Occhiogrosso B. IoT Considerations, Requirements, and Architectures for Smart Buildings-Energy Optimization and Next-Generation Building Management Systems. *IEEE Internet Things J.* 2017. Vol. 4. P. 269–283.

23. Wen Y. J., Agogino A. M. Control of wireless-networked lighting in open-plan offices. *Light. Res. Technol.* 2011. Vol. 43. P. 235–248.

24. Ha S., Tae S., Kim R. A Study on the Limitations of South Korea's National Roadmap for Greenhouse Gas Reduction by 2030 and Suggestions for Improvement. *Sustainability.* 2019. Vol. 11. P. 3969.

25. Han J., Jeon S. H., Lee G. H., Park S., Choi J. K. Power and Frequency Band Allocation Mechanisms for WPT System with Logarithmic-Based Nonlinear Energy Harvesting Model. *Sustainability.* 2023. Vol. 15. P. 10567.

26. Kahn J. M., Barry J. R. Wireless infrared communications. *Proc. IEEE.* 1997. Vol. 85. P. 265–298.

27. Yang H., Bergmans J. M., Schenk T. C. W., Linnartz J.-P. M. G., Rietman R. An analytical model for the illuminance distribution of a power LED. *Opt. Express.* 2008. Vol. 16. P. 21641–21646.

28. Chen C., Chen Y., Han Y., Lai H. Q., Zhang F., Liu K. J. R. Achieving Centimeter-Accuracy Indoor Localization on WiFi Platforms: A Multi-Antenna Approach. *IEEE Internet Things J.* 2017. Vol. 4. P. 122–134.

29. Pan M. S., Yeh L. W., Chen Y. A., Lin Y. H., Tseng Y. C. A WSN-based intelligent light control system considering user activities and profiles. *IEEE Sens. J.* 2008. Vol. 8. P. 1710–1721.

30. Yeh L., Lu C., Kou C., Tseng Y., Yi C. Autonomous Light Control by Wireless Sensor and Actuator Networks. *IEEE Sens. J.* 2010. Vol. 10. P. 1029–1041.
31. Wen Y. J., Agogino A. M. Personalized dynamic design of networked lighting for energy-efficiency in open-plan offices. *Energy Build.* 2011. Vol. 43. P. 1919–1924.
32. Pandharipande A., Caicedo D. Adaptive illumination rendering in LED lighting systems. *IEEE Trans. Syst. Man Cybern. Syst.* 2013. Vol. 43. P. 1052–1062.
33. Meugheuvél N., Pandharipande A., Caicedo D., Van Den Hof P. P. J. Distributed lighting control with daylight and occupancy adaptation. *Energy Build.* 2014. Vol. 75. P. 321–329.
34. Caicedo D., Li S., Pandharipande A. Smart lighting control with workspace and ceiling sensors. *Light. Res. Technol.* 2016. Vol. 49. P. 446–460.
35. Caicedo D., Pandharipande A. Distributed illumination control with local sensing and actuation in networked lighting systems. *IEEE Sens. J.* 2013. Vol. 13. P. 1092–1104.
36. DIY IoT Implementation with ESP32. *HackMD*. URL: https://hackmd.io/@pramho05/basic_iot (дата звернення: 15.06.2026).
37. ESP32: The Ultimate Guide to the Versatile IoT Microcontroller. *Evelta*. URL: <https://evelta.com/blog/esp32-the-ultimate-guide-to-the-versatile-iot-microcontroller/> (дата звернення: 15.06.2026).
38. Low-code programming for event-driven applications. *Node-RED*. URL: <https://nodered.org> (дата звернення: 15.06.2026).
39. HiveMQ – Stream Data. Build Intelligence. Activate AI. *HiveMQ*. URL: <https://www.hivemq.com> (дата звернення: 15.06.2026).
40. Wokwi - World's most advanced ESP32 Simulator. *Wokwi*. URL: <https://wokwi.com> (дата звернення: 15.06.2026).

Зав. кафедри КІПС
д-р. філософії Ользі ПАВЛОВІЙ

Петро СВІНЦ

ІІІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-22-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

8 червня 2026 року



Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Петро СВІНЦ

Співавтор:

Назва: Програмно-апаратна система освітлення приміщення з датчиком присутності та регулюванням яскравості

Експерт: Дмитро МЕДЗАТИЙ

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 5.64%

Коефіцієнт подібності 2: 2.94%

Мікропробіли: 3

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-06-15 08:44:19.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2026-06-15



Доцент Андрій Нічепорук

Дата

експерт

Tue Jun 16 08:10:06 EEST 2026, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 11%**

ID: 275399 Назва: БКР Програмно-апаратна система освітлення приміщення з датчиком присутності та регулюванням яскравості Додано в БД: 2026-06-16 Автора: Петро СВІНЦ Керівники: Дмитро МЕДЗАТИЙ Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	86047	589	1104 (1%)	15 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-апаратна система освітлення приміщення з датчиком присутності та регулюванням яскравості

Автор Петро СВІНЦ

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: к.т.н. доцент. Дмитро МЕДЗАТИЙ

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 5,64% і адресується до 26 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

16.06.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи


Підпис


Підпис

Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК
Ім'я, ПРІЗВИЩЕ

Дмитро МЕДЗАТИЙ
Ім'я, ПРІЗВИЩЕ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Петро СВІНЦ

Тема: Програмно-апаратна система освітлення приміщення з датчиком присутності та регулюванням яскравості

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 60

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано програмно-апаратну систему освітлення приміщення з датчиком присутності та регулюванням яскравості

2. Висновок про відповідність роботи дипломному завданню _____
Дипломний проект відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз актуальності задачі автоматизованого керування освітленням приміщень, розглянуто концепцію програмно-апаратних систем на базі мікроконтролерів з підтримкою бездротового зв'язку, проаналізовано існуючі методи виявлення присутності людини та адаптивного регулювання яскравості і виконано порівняльний огляд відомих рішень у сфері розумного освітлення приміщень. У другому розділі здійснено проектування програмно-апаратної системи освітлення приміщення з датчиком присутності та регулюванням яскравості на основі мікроконтролера ESP32. Розглянуто функціональні та нефункціональні вимоги до системи, розроблено її структурну схему з описом MQTT-топиків та розподілом компонентів між хмарним і локальним рівнями, розроблено алгоритм функціонування мікроконтролера та його блок-схему, виконано аналіз апаратних компонентів. У третьому розділі представлено процес симуляційного тестування розробленої системи. Описано мету і завдання симуляції та застосований технологічний стек, реалізовано програмний симулятор сенсорних

даних мовою Python, побудовано цифровий двійник апаратної частини у симуляторі Wokwi, розгорнуто локальну MQTT-інфраструктуру з брокером Mosquitto та інформаційну панель оператора на платформі Node-RED.

4. Позитивні сторони роботи: Розроблена система поєднує доступну апаратну базу на основі мікроконтролера ESP32 з повноцінною програмною інфраструктурою у вигляді локального MQTT-брокера Mosquitto та інформаційної панелі Node-RED.

5. Негативні сторони роботи: Математична модель денного циклу природного освітлення у Python-симуляторі є спрощеною і не враховує сезонних змін тривалості світлового дня, хмарності та локальних умов освітленості, що вимагатиме повторного калібрування порогових значень АЦП при переході до фізичного прототипу в реальних умовах експлуатації.

6. Оцінка графічного оформлення та пояснювальної записки роботи: пояснювальна записка та листи креслення виконані згідно діючих вимог

7. Відгук про роботу в цілому: В загальному робота виконана на задовільному рівні.

8. Інші зауваження: —

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «задовільно» 70 (D)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Петлюк Наталія Сергіївна, доцент кафедри КБ

“ 17 ” червня 2026р.