

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“___” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему:

**Програмне забезпечення системи автоматизованого освітлення
з дистанційним керуванням через мобільний додаток**

Виконав здобувач вищої освіти
IV курсу, групи КІ-22-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Дощенко Ю.Д.
«___» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Кислун О.А.
«___» _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет

Факультет Механіко-технологічний

Кафедра Кібербезпеки та програмного забезпечення

Рівень вищої освіти бакалавр

Галузь знань 12 "Інформаційні технології"

Спеціальність 123 "Комп'ютерна інженерія"

Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

_____ Олексій СМІРНОВ

_____ " " 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Доценку Юрію Дмитровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи автоматизованого освітлення з дистанційним керуванням через мобільний додаток

2. Керівник роботи Кислун Олег Андрійович, канд. техн. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від "27" лютого 2026 року №136-02

3. Строк подання роботи до захисту 10.06.2026 р.

4. Мета та завдання випускної кваліфікаційної роботи. Метою роботи є розробка програмного забезпечення системи управління освітленням з дистанційним керуванням

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію

6. Висновки.

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

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи системи керування 2 аркуша

7. Дата видачі завдання «27» лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем керування	10.03.2026	
2.	Постановка задачі, оформлення ТЗ	15.03.2026	
3.	Розробка моделі компонента	20.03.2026	
4.	Розробка структур даних	25.03.2026	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026	
6.	Програмування алгоритмів	10.04.2026	
7.	Оформлення ПЗ	17.05.2026	
8.	Попередній захист роботи	23.05.2026	

Дата видачі завдання
«__»_____2026р.

Підпис керівника

(прізвище та ініціали)

Завдання прийнято до виконання
«__»_____2026р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Дощенко Ю.М. Програмне забезпечення системи автоматизованого освітлення з дистанційним керуванням через мобільний додаток. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

У випускній кваліфікаційній бакалаврській роботі розглянуто розробку програмного забезпечення системи автоматизованого освітлення з можливістю дистанційного керування через мобільний додаток. Проаналізовано сучасні підходи до побудови систем розумного освітлення та технології Інтернету речей (IoT), що використовуються для їх реалізації.

У роботі спроектовано архітектуру системи, яка включає мікроконтролерний модуль, програмне забезпечення керування та мобільний додаток для користувача. Реалізовано функціонал дистанційного увімкнення та вимкнення освітлення, налаштування режимів роботи, а також моніторинг стану системи в режимі реального часу.

Особливу увагу приділено розробці програмного забезпечення, взаємодії між клієнтською та серверною частинами, а також забезпеченню стабільності та безпеки передачі даних.

Проведено тестування системи та оцінено її ефективність у реальних умовах використання.

Результати роботи можуть бути використані для впровадження у системах «розумного будинку» з метою підвищення енергоефективності та зручності керування освітленням.

Ключові слова: втоматизоване освітлення, мобільний додаток, IoT, мікроконтролер, дистанційне керування, розумний будинок.

ABSTRACT

Doschenko Y. Software for automated lighting systems with remote control via a mobile app. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

The qualification bachelor's thesis considers the development of software for an automated lighting system with the possibility of remote control via a mobile application. Modern approaches to building smart lighting systems and the Internet of Things (IoT) technologies used for their implementation are analyzed.

The work designs the system architecture, which includes a microcontroller module, control software, and a mobile application for the user.

The functionality of remote switching on and off of lighting, setting operating modes, and monitoring the system status in real time are implemented.

Particular attention is paid to software development, interaction between the client and server parts, as well as ensuring the stability and security of data transmission.

The system is tested and its effectiveness is assessed in real-world conditions.

The results of the work can be used for implementation in "smart home" systems in order to increase energy efficiency and convenience of lighting control.

Keywords: automated lighting, mobile application, IoT, microcontroller, remote control, smart home.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	4
1.1 Призначення системи.....	4
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським рівнем вищої освіти)	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	17
2.3 Розгорнута постановка завдання	19
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	20
3.1 Опис функціонування системи	22
3.2 Розробка структурної схеми.....	30
3.3 Розробка функціональної схеми	32
3.4 Розробка діаграми процесів.....	34
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	36
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	36
4.2 Алгоритм функціонування системи	45
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	56
6 ОСНОВНІ ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	71

					ВКРБ-123.26.0041.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи автоматизованого освітлення з дистанційним керуванням через мобільний додаток</i>	Лім.	Аркуш	Аркушів
Розроб.	Доценко Ю.Д.					Б	1	75
Перев.	Кислун О.А.					Ков		
Н.контр.	Коваленко А.С.					ЦНТУ КІ-22-2		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

APM	—	автоматизоване робоче місце
АСУ	—	автоматизована система управління
МК	—	мікроконтролер (microcontroller)
БС	—	дистанційне управління
Arduino Uno	—	апаратно-програмна платформа на МК ATmega328
AVR	—	лінійка мікроконтролерів фірми Atmel
LCD	—	рідкокристалічний індикатор/дисплей
RAM (ОЗП)	—	перативний запам'ятовуючий пристрій
ПДУ	—	пульти дистанційного управління
ПЗ	—	програмне забезпечення
ПЛК	—	програмувальні логічні контролери
USART	—	універсальний синхронний/асинхронний приймач та передавач
I ² C	—	послідовна шина даних для зв'язку інтегральних схем, що використовує дві двонаправлені лінії зв'язку (SDA і SCL)

ВСТУП

У сучасних умовах концепція «розумного дому» швидко переходить від теоретичних ідей до практичного застосування. Інтелектуальні системи автоматизації дедалі ширше впроваджуються у повсякденне життя, підвищуючи комфорт, рівень безпеки та енергоефективність житлових і комерційних об'єктів. Одним із прикладів таких рішень є автоматизоване керування підсвіткою, зокрема освітленням сходів, яке виконує не лише декоративну функцію, а й істотно підвищує безпеку пересування у темний період доби.

Кваліфікаційна робота присвячена створенню комп'ютерної системи інтелектуального керування підсвіткою сходів із підтримкою дистанційного моніторингу та керування. Основна увага зосереджена на розробці енергоефективної та адаптивної системи освітлення, що реагує на рух, рівень навколишнього освітлення та надає користувачеві можливість змінювати режими підсвітки віддалено - через мобільний застосунок або Telegram-бот.

Побудована на базі мікроконтролера ESP32, система поєднує функції локального керування за допомогою кнопок із дистанційним контролем через хмарні сервіси та бездротові технології. Інтеграція датчиків руху, присутності й освітленості забезпечує адаптивну роботу пристрою: підсвітка автоматично вмикається за потреби та вимикається, коли сходи не використовуються.

Актуальність теми обумовлена підвищеними вимогами до безпеки та комфорту сучасного житла, а також необхідністю раціонального й енергоефективного використання електроенергії. У межах роботи реалізовано повний цикл створення системи - від аналізу вимог і проєктування апаратної частини до розробки програмного забезпечення та проведення випробувань.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Комп'ютерна система підсвітки сходів із функцією віддаленого моніторингу має бути розроблена як інтерактивне автоматизоване рішення, що забезпечує повний контроль і керування адресними світлодіодними стрічками. Запропоновано реалізацію системи на базі мікроконтролера ESP32 із підтримкою Wi-Fi та Bluetooth, а також інтеграцію з популярним месенджером Telegram для створення зручного та інтуїтивно зрозумілого мобільного інтерфейсу.

Головною метою розробки є формування адаптивної системи освітлення, яка реагує як на локальні дії користувача (через фізичні кнопки), так і на команди, отримані з віддаленого пристрою. Важливою особливістю є застосування адресної світлодіодної стрічки WS2812B, що дозволяє програмно керувати кожним світлодіодом окремо. Це відкриває можливість реалізації різноманітних світлових ефектів, кольорових схем і режимів індикації.

На відміну від традиційних освітлювальних систем, розроблювана комп'ютерна система повинна мати здатність адаптуватися до різних сценаріїв освітлення відповідно до команд користувача, зберігати налаштування у хмарному середовищі та здійснювати обмін даними із сервером у реальному часі через WebSocket-з'єднання.

Особливу увагу доцільно приділити забезпеченню доступності та надійності системи. Telegram-бот залишається доступним у будь-який час, тоді як фізичні кнопки дублюють основні функції у разі втрати мережевого з'єднання. Пристрій також повинен підтримувати автоматичне повторне підключення до Wi-Fi і зберігати останній активний стан освітлення, що забезпечує стабільність роботи та зручність у щоденному використанні.

Отже, система призначена не лише для створення візуального комфорту чи декоративного підсвічування, але й для демонстрації практичного застосування

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

підходів IoT у побутових умовах, поєднуючи апаратну частину, програмну логіку та хмарну інтеграцію в єдину функціональну платформу. Крім того, вона є масштабованою та передбачає можливість керування кількома пристроями з одного інтерфейсу.

Головна мета розробки системи моніторингу освітлення сходів полягає у підвищенні гнучкості, зручності та функціональності сучасних освітлювальних рішень завдяки поєднанню локального й дистанційного керування, інтеграції з мобільним інтерфейсом (Telegram-ботом) і реалізації двостороннього моніторингу через хмарну інфраструктуру.

Проект орієнтований на створення системи, що дозволяє користувачеві у реальному часі змінювати параметри підсвітки на базі WS2812B, формувати індивідуальні сценарії освітлення та отримувати актуальну інформацію про стан пристрою незалежно від відстані до нього.

Досягнення поставленої мети передбачає виконання таких завдань:

- проведення аналітичного огляду існуючих систем керування освітленням і рішень із інтеграцією Telegram;
- розроблення структурної та функціональної архітектури комп'ютеризованої системи керування адресною світлодіодною стрічкою на базі ESP32;
- обґрунтування вибору апаратних компонентів (мікроконтролера, кнопок керування, стрічки WS2812B, джерела живлення);
- проектування схеми підключення апаратних елементів з урахуванням енергетичних і комунікаційних параметрів;
- створення програмного забезпечення, що забезпечує обробку команд із Telegram, взаємодію з WebSocket-сервером та керування світлодіодною стрічкою;
- проведення тестування готової системи, вимірювання затримки відгуку, перевірку стабільності мережевого з'єднання та оцінювання надійності роботи в різних режимах експлуатації.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Об’єктом дослідження у кваліфікаційній роботі виступає комп’ютерна система дистанційного моніторингу освітлення, реалізована з використанням адресної світлодіодної стрічки WS2812B, мікроконтролера ESP32 та технологій Інтернету речей (IoT). Система забезпечує як локальне, так і віддалене керування підсвіткою в реальному часі, що робить її універсальним рішенням для автоматизації різних сценаріїв освітлення.

Ключовою особливістю об’єкта є можливість інтерактивної зміни параметрів освітлення - кольору, яскравості та режиму - через два паралельні канали керування: фізичні кнопки, підключені до ESP32, і Telegram-бот, що функціонує через хмарну інфраструктуру (API Gateway, WebSocket, Lambda).

Система побудована на архітектурі реального часу із використанням FreeRTOS, що забезпечує паралельне виконання завдань. Кожен світлодіод стрічки програмується окремо, що дозволяє реалізовувати складні світлові ефекти та сценарії з високою точністю.

Особлива увага приділена адаптивності освітлення завдяки адресному керуванню WS2812B, дистанційному контролю через Telegram-інтерфейс і двосторонній синхронізації з сервером за допомогою WebSocket-з’єднання.

У разі втрати зв’язку з мережею або сервером система автоматично переходить у локальний режим роботи без втрати поточних налаштувань. Крім того, всі ключові дії супроводжуються зворотними повідомленнями в Telegram, що дозволяє користувачеві постійно контролювати стан пристрою. Завдяки модульній архітектурі система може бути розширена шляхом підключення додаткових компонентів або впровадження нових сценаріїв, зокрема додавання датчиків освітленості для автоматичного вмикання підсвітки у темний час доби чи інтеграції з голосовими помічниками, такими як Google Assistant або Alexa.

Окрім базових функцій, об’єкт дослідження передбачає інтелектуальний моніторинг стану пристрою через WebSocket-з’єднання з AWS, автоматичне відновлення підключення до Wi-Fi, а також реалізацію захисту від несанкціонованого доступу із використанням токенів і шифрування даних.

Комп'ютерна система має проєктуватися як багатокомпонентне IoT-рішення, що поєднує сучасні підходи до інтеграції мікроконтролерів, хмарних сервісів і зручних мобільних інтерфейсів керування в єдину платформу, придатну для використання у побутових, освітніх або комерційних умовах.

Система підсвітки сходів із функцією дистанційного моніторингу є багатофункціональною IoT-системою, яка забезпечує керування освітленням на основі подій, пов'язаних із присутністю користувача, а також надає можливість моніторингу й конфігурування у реальному часі. Загальні вимоги до системи поділяються на функціональні та нефункціональні, охоплюючи як апаратні, так і програмні аспекти її роботи.

До основних функціональних можливостей системи належать:

- автоматичне вмикання та вимикання світлодіодної підсвітки при виявленні руху на сходах;
- плавне ввімкнення освітлення знизу вгору або зверху вниз відповідно до напрямку руху користувача;
- локальне керування режимами підсвітки за допомогою фізичних кнопок (наприклад, вибір сценарію освітлення);
- можливість дистанційного керування підсвіткою через мобільний застосунок або месенджер (зокрема Telegram-бот);
- збереження активного режиму освітлення та його автоматичне відновлення після перезапуску системи;
- передавання інформації про стан пристрою на сервер моніторингу через WebSocket у режимі реального часу;
- підтримка кількох режимів підсвітки (анімаційні ефекти, статичне освітлення, нічний режим).
- можливість ручного керування системою за відсутності підключення до мережі Wi-Fi.

Програмне забезпечення системи повинно забезпечувати:

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

- обробку сигналів від датчиків руху (HC-SR501) та освітленості з мінімальною затримкою (до 1 секунди);
- реалізацію анімаційного сценарію підсвітки з урахуванням напрямку переміщення користувача;
- передавання даних про стан системи (увімкнення, активний режим, рівень освітлення) через WebSocket до хмарної платформи (наприклад, AWS);
- обробку запитів із Telegram-бота з часом відгуку до 3 секунд;
- ведення журналу подій із фіксацією часу активації, активного режиму, стану датчиків і дій користувача;
- підтримку багатозадачності завдяки використанню операційної системи реального часу (наприклад, FreeRTOS на ESP32).

До нефункціональних характеристик системи належать:

- зручність і ергономічність, що досягаються завдяки інтуїтивно зрозумілому інтерфейсу Telegram-бота;
- надійність - система повинна стабільно працювати без збоїв у безперервному циклі вмикання та вимикання щонайменше протягом 72 годин;
- доступність - дублювання ключових функцій через локальне керування, яке має працювати незалежно від наявності Інтернет-з'єднання;
- захищеність - використання авторизаційних токенів під час взаємодії з Telegram-ботом і шифрування даних між пристроєм та сервером;
- масштабованість - можливість додавання нових сценаріїв освітлення, підключення додаткових сенсорів або розширення світлодіодної стрічки;
- енергоефективність - автоматичне зменшення яскравості у нічний час і оптимізація тривалості роботи підсвітки.

1.2 Область застосування

Сучасні тенденції автоматизації житлових і комерційних приміщень спрямовані на підвищення комфорту, енергоефективності та рівня безпеки. Одним

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

із перспективних напрямів є інтелектуальні системи освітлення, зокрема автоматизовані системи підсвітки сходів із можливістю дистанційного керування та моніторингу. Такі рішення не лише покращують естетичний вигляд приміщень, а й забезпечують необхідну видимість на сходах у темний час доби або за умов недостатнього освітлення.

Дослідження свідчать, що використання адаптивних світлодіодних підсвіток дає змогу зменшити споживання електроенергії на 30–50% порівняно з традиційними системами освітлення. Крім того, інтеграція засобів віддаленого моніторингу забезпечує оперативний контроль роботи системи та своєчасне виявлення можливих несправностей.

Системи підсвітки сходів можна класифікувати за такими критеріями:

- тип керування - застосовується локальне керування з використанням датчиків руху, кнопок, таймерів або дистанційне керування із використанням Wi-Fi, Bluetooth, Zigbee, GSM та інших бездротових технологій передавання даних;
- тип освітлення - може бути постійне або динамічне освітлення, що реагує на рух, час доби чи рівень навколишнього освітлення;
- рівень автоматизації - від простих систем увімкнення/вимкнення до інтелектуальних рішень із алгоритмічним керуванням або використанням елементів штучного інтелекту;
- наявність зворотного зв'язку та моніторингу - системи можуть мати функції контролю стану освітлення або працювати без них.

У загальному випадку архітектура сучасної системи підсвітки сходів може бути представлена як взаємодія таких основних компонентів:

- контролер керування, наприклад Arduino або ESP32;
- датчики руху, освітленості та присутності;
- світлодіодні елементи, зокрема світлодіодні стрічки або модулі;
- комунікаційні модулі, такі як Wi-Fi або GSM;
- хмарні сервіси чи мобільні застосунки;
- модуль живлення.

У загальному випадку архітектуру системи дистанційного керування підсвіткою сходів можна подати у вигляді структурної схеми, як показано на рис. 1.1.

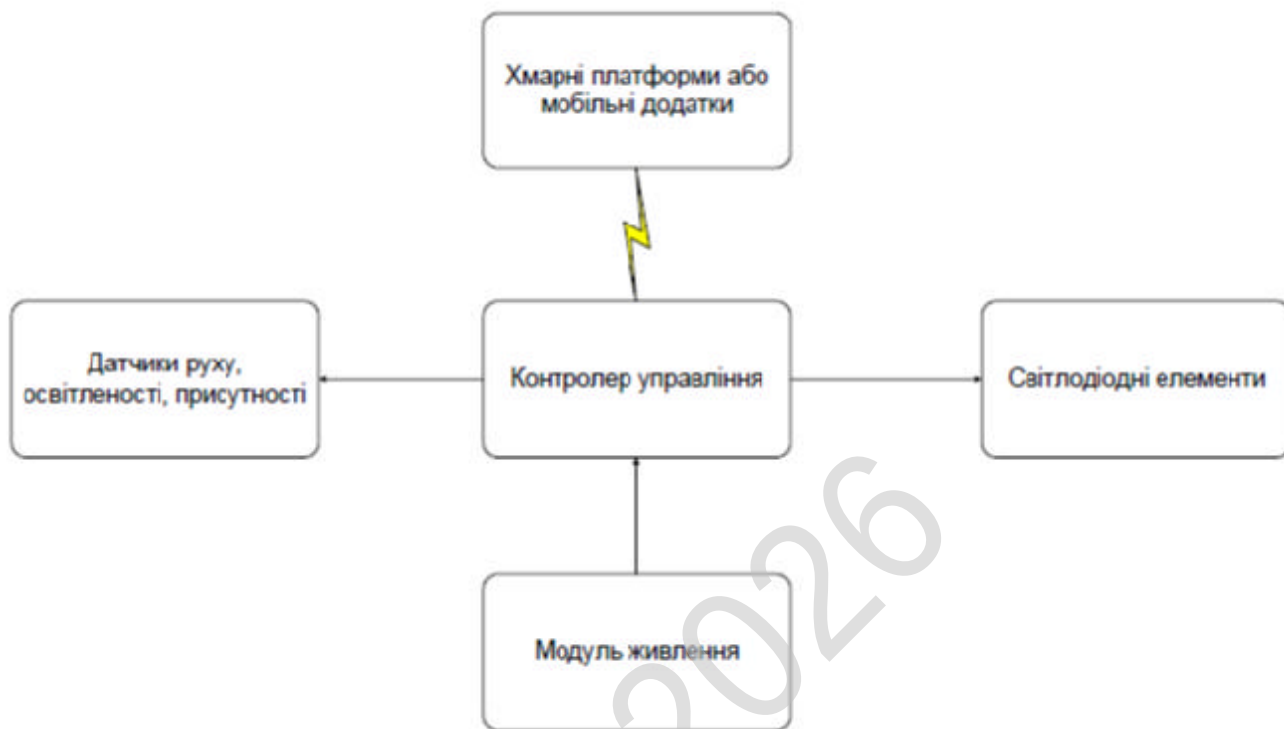


Рисунок 1.1 – Узагальнена структурна схема систем віддаленого керування підсвіткою сходів

Контролер керування є центральним елементом системи, який забезпечує взаємодію із сенсорами та обмін даними з мобільним застосунком або хмарним сервісом. До датчиків руху належить PIR-сенсор. Як датчики освітленості можуть застосовуватися фоторезистори, а для визначення присутності - інфрачервоні або ультразвукові сенсори. У ролі світлодіодних елементів можуть використовуватися як звичайні, так і адресні світлодіодні стрічки.

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським рівнем вищої освіти)

Одним із найпоширеніших підходів є застосування модулів ESP8266 або ESP32 із підтримкою Wi-Fi. Такі пристрої зазвичай працюють разом із протоколом MQTT для передавання даних до хмарного сервера. MQTT є легким протоколом типу «публікація-підписка», що забезпечує ефективний обмін інформацією навіть за обмеженої пропускної здатності мережі.

У випадках, коли підключення до локальної мережі неможливе, використовують GSM-модулі (SIM800L, A6, SIM900), які дозволяють передавати SMS-повідомлення або підключатися до Інтернету через мобільні мережі. Такий підхід забезпечує автономність роботи, проте потребує наявності SIM-карти та додаткових витрат на мобільний зв'язок.

Технології Bluetooth і Zigbee застосовуються переважно в локальних бездротових мережах. Зокрема, Zigbee є енергоефективним рішенням для багатозонних систем із кількома контролерами, тоді як Bluetooth частіше використовується для керування через мобільний застосунок на невеликій відстані.

Системи моніторингу підсвітки сходів дають змогу відображати стан кожного світлодіодного елемента, виявляти несправності (наприклад, перегрів або втрату зв'язку), а також фіксувати статистику роботи - час і тривалість увімкнення підсвітки під час переміщення сходами.

До найпоширеніших платформ моніторингу стану підсвітки належать:

– **Blynk** - хмарна платформа для створення інтерфейсів керування IoT-пристроями з підтримкою обміну даними через Wi-Fi або GSM;

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

- **ThingSpeak** - сервіс для обробки, зберігання та візуалізації телеметричних даних;
- **Home Assistant** - потужна система з відкритим кодом для реалізації комплексної автоматизації «розумного дому».

Одним із можливих рішень для автоматичної підсвітки сходів є використання мікроконтролера Arduino. Розробки на базі Arduino Uno або Nano із застосуванням і використання датчиків HC-SR501 (інфрачервоних сенсорів руху) дає змогу реалізувати просту та економічно доступну систему з поступовим увімкненням світлодіодної стрічки (рис. 2.1). Однак подібні рішення, як правило, не підтримують дистанційний доступ і не забезпечують функцій моніторингу.

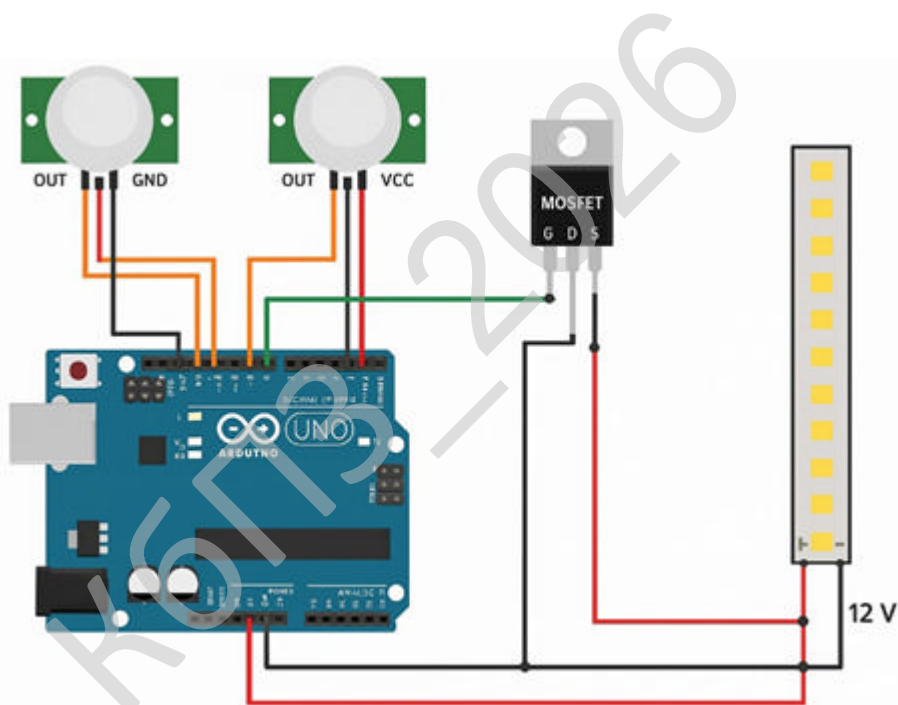


Рисунок 2.1 – Схема організації підсвітки сходів на базі Arduino

Принцип функціонування такої системи передбачає використання двох датчиків HC-SR501, встановлених на початку та в кінці сходового прольоту. У разі спрацювання будь-якого з них система визначає напрямок руху користувача (вгору або вниз), після чого контролер Arduino поетапно активує сегменти світлодіодної стрічки, формуючи ефект «хвилі».

Після завершення переміщення сходами освітлення автоматично вимикається у зворотній послідовності. Час затримки вимкнення може задаватися користувачем, наприклад, становити 10 секунд за відсутності руху.

Спрощений алгоритм керування підсвіткою в такій системі включає послідовне виконання таких дій: безперервне опитування стану обох датчиків HC-SR501; у разі спрацювання одного з них - визначення напрямку руху та запуск процедури поступового ввімкнення світлодіодів, наприклад із використанням програмної функції `analogWrite()` із плавним підвищенням рівня яскравості.

Після завершення проходу таймер відраховує 10 - 15 секунд, після чого відбувається плавне вимкнення світлодіодної стрічки, а система повертається у режим очікування наступного спрацювання.

До переваг такого підходу можна віднести низьку вартість, простоту реалізації та створення естетичного ефекту плавного освітлення. Водночас подібні рішення мають певні обмеження, зокрема відсутність дистанційного моніторингу, обмежену точність визначення напрямку руху, відсутність мобільного інтерфейсу та неможливість зберігання журналу подій.

Іншим варіантом реалізації системи дистанційного керування підсвіткою сходів є рішення на базі ESP32 із використанням платформи Blynk. Мікроконтролер ESP32 забезпечує підключення до Wi-Fi, а мобільний застосунок Blynk надає користувачеві можливість керувати підсвіткою, відстежувати стан системи в реальному часі та отримувати повідомлення про можливі відхилення в роботі. Таке рішення є більш функціональним, однак вимагає налаштування мережевого підключення, проходження авторизації на сервері та постійного доступу до Інтернету. На рисунку 2.2 наведено типову архітектуру системи, побудованої на основі ESP32 та Blynk.

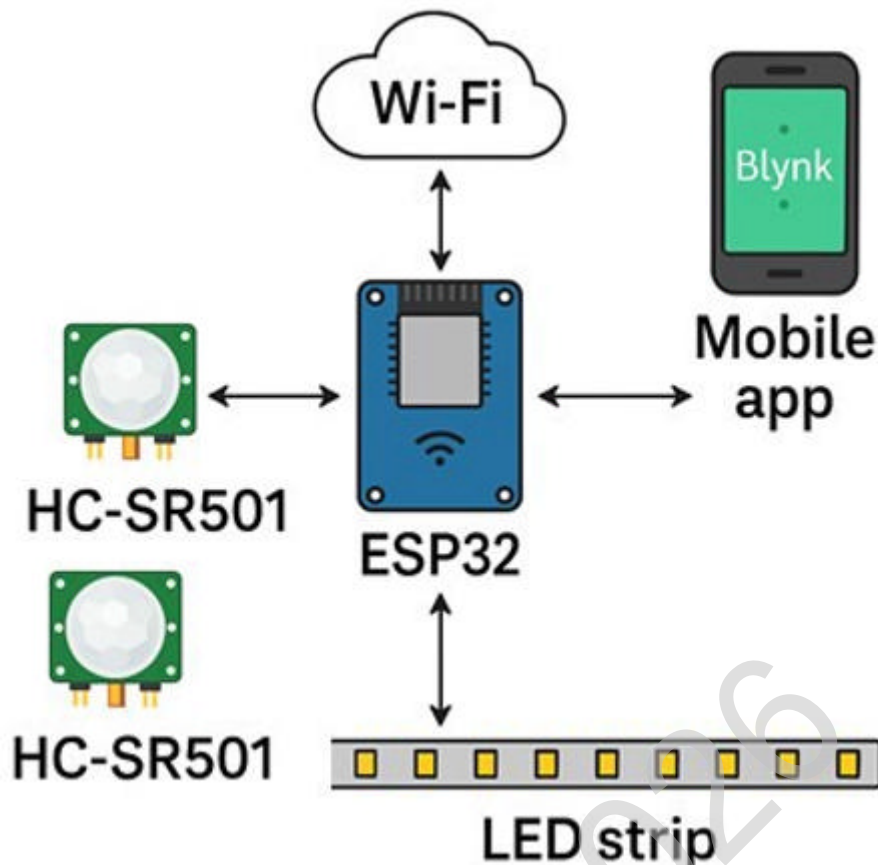


Рисунок 2.2 - Архітектура системи, побудованої на основі ESP32 та Blynk.

Система, представлена на рисунку 2.2, призначена для автоматичного вмикання світлодіодної підсвітки при виявленні руху на сходах, а також для дистанційного керування, моніторингу та отримання сповіщень через мобільний застосунок.

Першим етапом алгоритму роботи системи є ініціалізація: мікроконтролер ESP32 підключається до локальної мережі Wi-Fi та проходить авторизацію на сервері Blynk за допомогою унікального токена. Після цього налаштовуються вхідні та вихідні контакти: входи - два піни для датчиків HC-SR501, вихід - ШІМ-пін для регулювання яскравості світлодіодної стрічки.

Другий етап передбачає постійне опитування датчиків, тобто безперервний контроль стану кожного HC-SR501. У разі спрацювання нижнього датчика система запускає плавне ввімкнення підсвітки, а при активації верхнього - аналогічний процес для руху у зворотному напрямку.

Плавне ввімкнення світлодіодної стрічки забезпечується за допомогою

широтно-імпульсної модуляції: яскравість поступово зростає, після чого освітлення підтримується протягом заданого часу. Після завершення тайм-ауту світло плавно зменшується до повного вимкнення.

У мобільному застосунку користувач може вручну вмикати або вимикати підсвітку сходів, переглядати стан датчиків, а також отримувати повідомлення про зафіксований рух або можливі збої в роботі. Крім того, передбачена можливість перегляду журналу подій (через Virtual Pins або webhooks).

За такої організації системи можуть бути реалізовані додаткові функції, зокрема:

- налаштування тривалості роботи підсвітки за допомогою слайдера у Blynk;
- нічний режим - автоматичне ввімкнення освітлення лише при низькому рівні зовнішнього освітлення з використанням фоторезистора;
- виявлення аномалій, наприклад у разі безперервної активації датчика протягом тривалого часу;
- збір статистики руху - підрахунок кількості спрацювань за добу з відображенням у застосунку.

На ринку представлені готові комерційні рішення для підсвітки сходів від відомих виробників, зокрема Philips Hue та LIFX Z, які можуть інтегруватися у системи «розумного дому». Проте їхня вартість суттєво перевищує собівартість розглянутих раніше варіантів, а функціональні можливості часто обмежуються рамками фірмових екосистем. Приклад реалізації комерційних рішень для освітлення сходів і поручнів наведено на рисунку 2.3.

Гнучкі світлодіодні стрічки Hue, розміщені під сходами, забезпечують рівномірне підсвічування кожної сходинки та створюють ефект м'якого світлового переливу. Такий підхід є популярним для оновлення інтер'єру завдяки гармонійному поєднанню естетики та функціональності.

Варіант із синьою RGB-підсвіткою під поручнями демонструє декоративне рішення для формування атмосферного освітлення, яке може змінюватися

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

відповідно до обраних колірних сценаріїв.

Інші приклади показують роботу системи через Hue Bridge, датчики руху Hue або світлодіодні стрічки Hue Strip, що дає змогу автоматично активувати підсвітку під час наближення до сходів.

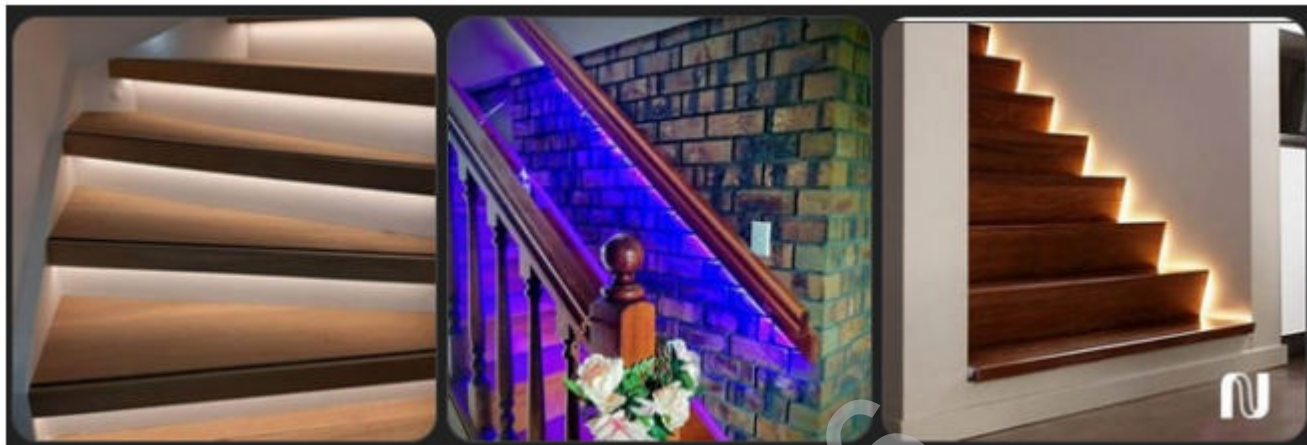


Рисунок 2.3 – Інтегровані у розумний будинок комерційні рішення підсвітки сходових кліток

Платформа Philips Hue пропонує готові, естетично привабливі рішення для підсвітки сходів із широкими можливостями налаштування, автоматизації та інтеграції у систему «розумного дому». Освітлювальні модулі можна персоналізувати за кольором, плавністю переходів і сценаріями роботи, а датчики руху забезпечують безконтактне ввімкнення підсвітки.

Розглянуті у цьому розділі системи керування підсвіткою мають також певні недоліки та обмеження, зокрема:

- необхідність стабільного підключення до мережі Інтернет;
- підвищені вимоги до енергоефективності при автономному режимі роботи;
- складність налаштування мережевих протоколів для користувачів без достатнього досвіду;
- потенційну вразливість до кіберзагроз за відсутності належного шифрування.

Попри зазначені недоліки, системи дистанційного керування підсвіткою

сходів мають значний потенціал розвитку. Використання алгоритмів машинного навчання може забезпечити адаптивну роботу освітлення з урахуванням поведінкових особливостей користувача.

До перспективних напрямів розвитку також належать:

- інтеграція з голосовими помічниками, такими як Google Assistant та Alexa;
- застосування енергоефективних драйверів і самовідновлюваних схем живлення;
- розширення функціональності завдяки модульному підходу.

Системи підсвітки сходів із віддаленим моніторингом демонструють широкий спектр застосування у сфері побутової автоматизації. Найбільш доцільним рішенням для сучасних проєктів є використання платформи ESP32 у поєднанні з мобільним застосунком Blynk або альтернативними хмарними сервісами, що дозволяє реалізувати не лише базові функції освітлення, а й моніторинг, керування та аналітику. Подальші дослідження можуть бути спрямовані на підвищення адаптивності та безпеки системи. У межах кваліфікаційної роботи запропоновано проєкт системи дистанційного керування підсвіткою сходів через Telegram-бот із використанням адресної світлодіодної стрічки та мікроконтролера ESP32.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для реалізації системи дистанційного керування підсвіткою сходів було обрано мікроконтролер ESP32, адресну світлодіодну стрічку типу WS2812B та програмне середовище Arduino IDE з використанням мови програмування C/C++. Такий вибір зумовлений технічними, економічними та функціональними перевагами зазначених засобів.

Мікроконтролер ESP32 є сучасною високопродуктивною платформою, що

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

має вбудовані модулі Wi-Fi та Bluetooth, достатній обсяг оперативної пам'яті та високу обчислювальну потужність. Наявність Wi-Fi дозволяє реалізувати віддалене керування системою без використання додаткових мережевих модулів, що зменшує складність та вартість пристрою. Крім того, ESP32 підтримує роботу в режимі реального часу, має низьке енергоспоживання та велику кількість периферійних інтерфейсів, що робить його оптимальним вибором для систем автоматизації та пристроїв Інтернету речей (IoT).

Використання адресної світлодіодної стрічки **WS2812B** дозволяє керувати кожним світлодіодом окремо, що забезпечує можливість створення різноманітних світлових ефектів, плавної зміни яскравості та кольору, а також реалізації інтелектуальних алгоритмів підсвітки сходів (послідовне вмикання, адаптивна яскравість, індикація руху тощо). Стрічка має простий цифровий інтерфейс керування, що значно спрощує апаратну та програмну реалізацію системи.

Для організації дистанційного керування було обрано платформу Telegram, оскільки вона є безкоштовною, кросплатформною та широко розповсюдженою. Telegram-бот забезпечує зручний користувацький інтерфейс, можливість надсилання команд, отримання стану системи та реалізацію сценаріїв автоматизації без створення окремого мобільного застосунку. Крім того, Telegram API є простим у використанні та добре документованим.

Як середовище розробки було використано **Arduino IDE**, яке має простий інтерфейс, велику кількість бібліотек і прикладів, а також підтримує програмування ESP32. Це дозволяє значно скоротити час розробки та спростити налагодження системи. Для роботи зі світлодіодною стрічкою застосовуються спеціалізовані бібліотеки (наприклад, FastLED або Adafruit NeoPixel), а для реалізації Telegram-бота - бібліотеки для роботи з мережевими протоколами та API.

Мова програмування **C/C++** була обрана через її високу швидкодію, ефективне використання ресурсів мікроконтролера та повну сумісність із середовищем Arduino. Вона дозволяє реалізувати обробку мережевих запитів,

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

керування світлодіодами в реальному часі, роботу з периферією та виконання алгоритмів автоматизації з мінімальними затримками. Крім того, C/C++ є стандартом для програмування вбудованих систем, що забезпечує переносимість та масштабованість програмного забезпечення.

Отже, вибір мікроконтролера ESP32, адресної світлодіодної стрічки WS2812B, платформи Telegram, середовища Arduino IDE та мови програмування C/C++ є технічно обґрунтованим і забезпечує надійність, функціональність, простоту реалізації та можливість подальшого розширення системи дистанційного керування підсвіткою сходів.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну роботу, реалізації розробка програмного забезпечення для управління підсвіткою сходів. В процесі розробки кваліфікаційної роботи необхідно виконати наступний обсяг проекту:

- а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;
- б) вибрати та обґрунтувати методику розробки відповідної системи та програмного забезпечення. Розробити функціональну та структурну схеми системи;
- в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;
- г) організувати інтерфейс користувача з метою формування та виводу на екран результатів аналізу та тестування;
- д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;
- е) сформулювати висновки про виконаний обсяг робіт та результати.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

Склад системи

Система дистанційного керування підсвіткою сходів побудована за принципом мікроконтролерної автоматизованої системи та складається з апаратної і програмної частин, що взаємодіють між собою через бездротову мережу Wi-Fi.

Апаратна частина системи

До складу апаратної частини входять такі основні елементи:

- **Мікроконтролерний модуль (ESP32)**

Виконує функції центрального керуючого пристрою: обробляє команди від користувача, формує сигнали керування світлодіодною стрічкою, забезпечує підключення до мережі Wi-Fi та обмін даними з Telegram-ботом.

- **Адресна світлодіодна стрічка (WS2812B)**

Служить виконавчим елементом системи освітлення. Дозволяє керувати кожним світлодіодом окремо, змінювати яскравість, колір та формувати світлові ефекти підсвітки сходів.

- **Блок живлення**

Забезпечує живлення мікроконтролера та світлодіодної стрічки. Як правило, використовується імпульсний блок живлення постійної напруги 5 В достатньої потужності, розрахованої на кількість світлодіодів.

- **Рівнеперетворювач (за потреби)**

Застосовується для узгодження логічних рівнів сигналу керування між ESP32 (3.3 В) та світлодіодною стрічкою (5 В), що підвищує стабільність роботи системи.

- **Захисні та допоміжні елементи**

Резистор у лінії даних, конденсатор згладжування живлення, з'єднувальні провідники, монтажна плата, корпус тощо.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Програмна частина системи

Програмна частина забезпечує логіку роботи та взаємодію компонентів системи і включає:

Прошивку мікроконтролера ESP32

Реалізує:

- підключення до Wi-Fi мережі;
- обмін даними з Telegram-ботом;
- прийом та обробку команд користувача;
- керування світлодіодною стрічкою;
- виконання світлових сценаріїв.

Telegram-бот

Забезпечує інтерфейс дистанційного керування системою. Дає можливість:

- вмикати та вимикати підсвітку;
- змінювати режим роботи;
- задавати колір та яскравість;
- отримувати стан системи.

Бібліотеки програмного забезпечення

- для роботи з Wi-Fi та мережевими протоколами;
- для взаємодії з Telegram API;
- для керування адресною світлодіодною стрічкою (FastLED або NeoPixel).

Функціональні модулі системи

У структурі системи можна виділити такі модулі:

- модуль живлення;
- модуль керування;
- модуль освітлення (WS2812B);
- модуль зв'язку (Wi-Fi + Telegram);
- програмний модуль обробки команд;

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

- модуль світлових ефектів.

Запропонований склад системи забезпечує надійну роботу, простоту реалізації та можливість подальшого розширення функціоналу, зокрема додавання датчиків руху, автоматичних сценаріїв освітлення, регулювання яскравості за часом доби або інтеграції з іншими системами «розумного будинку»

3.1 Опис функціонування системи

Архітектура системи віддаленого керування та моніторингу підсвіткою сходів

У роботі запропоновано розглядати систему підсвітки сходів із віддаленим моніторингом як розподілену інтелектуальну архітектуру, орієнтовану на керування адресною світлодіодною підсвіткою у режимі реального часу. В основі системи лежить мікроконтролер ESP32, використання Telegram-бота як віддаленого інтерфейсу керування та хмарної інфраструктури AWS для обробки запитів, збереження конфігурацій і забезпечення двостороннього обміну даними через WebSocket-з'єднання.

Архітектура об'єднує апаратний рівень (edge) із хмарними обчисленнями (cloud), що дозволяє забезпечити масштабованість, безперервність функціонування, а також реалізувати віддалений моніторинг і керування системою. Загальну структуру архітектури наведено на рисунку 3.1.

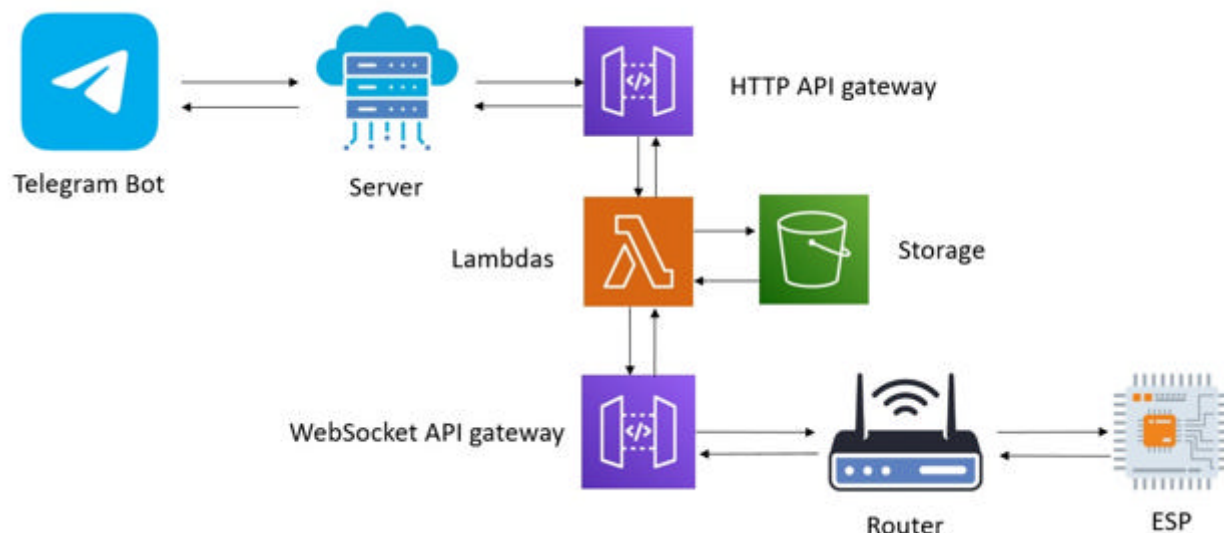


Рисунок 3.1 – Архітектура системи віддаленого моніторингу освітлення сходів

Telegram-бот виступає основним інтерфейсом взаємодії з користувачем. За його допомогою можна надсилати команди на вмикання або вимикання підсвітки, змінювати кольори світлодіодів, обирати режими освітлення (анімаційні, статичні, реактивні). Крім того, месенджер надає можливість отримувати актуальну інформацію про стан пристрою, зокрема поточний режим роботи, вибрані кольори, статус підключення, а також запускати процедури діагностики. Використання Telegram API забезпечує доступність керування з будь-якого пристрою, підключеного до Інтернету.

Центральний сервер виконує обробку повідомлень, отриманих від Telegram-бота, та передає запити через HTTP API Gateway. Він може функціонувати як проксі або як логічний рівень, що забезпечує авторизацію, зберігання токенів і ведення журналу сесій користувачів.

Хмарна інфраструктура AWS, зокрема HTTP API Gateway, виконує роль шлюзу для обробки вхідних HTTP-запитів від Telegram-бота чи інших зовнішніх джерел. До його основних функцій належать:

- перевірка коректності запитів;
- авторизація за допомогою токена або ключа;

- маршрутизація запитів до відповідної Lambda-функції;
- контроль доступу;
- ведення журналу транзакцій.

API Gateway забезпечує масштабовану обробку запитів без необхідності розгортання власної серверної інфраструктури.

Lambda-функції реалізують серверless-логіку - це обчислювальні модулі, які автоматично активуються у відповідь на запити. Вони виконують такі задачі:

- обробку команд /on, /off, /color, /mode;
- перевірку коректності параметрів;
- формування повідомлень для ESP32 у відповідному форматі;
- підготовку відповідей для Telegram;
- взаємодію з WebSocket API Gateway для передавання команд на пристрій;
- журналювання дій користувача;
- керування збереженням налаштувань у S3.

Перевагою використання Lambda є можливість автоматичного масштабування, відсутність необхідності у постійно працюючому сервері та оплата лише за фактичний час обробки запитів.

Сервіс Amazon S3, що виконує роль сховища конфігурацій, відповідає за збереження стану пристрою та користувацьких налаштувань. У межах системи віддаленого моніторингу підсвітки сходів у сховищі можуть зберігатися:

- останній активний режим підсвітки;
- обраний колір або світловий ефект;
- історія взаємодій із системою;
- конфігураційні файли для попередньо заданих сценаріїв.

Використання S3 забезпечує надійне, безпечне та доступне зберігання параметрів, що дозволяє зберігати працездатність системи незалежно від локальних збоїв.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

На відміну від протоколу HTTP, який працює за принципом «запит–відповідь», WebSocket забезпечує постійне з’єднання між пристроєм (ESP32) і серверною частиною. Це дає змогу оперативно передавати команди на пристрій та отримувати зворотний зв’язок, наприклад підтвердження виконання. Крім того, WebSocket підтримує канал моніторингу (heartbeat, keep-alive) і зменшує затримки під час частих подій, таких як швидка зміна режимів освітлення. Саме цей канал використовується для обміну даними між Lambda та ESP32 у режимі реального часу.

Маршрутизатор забезпечує локальне підключення ESP32 до мережі Інтернет, що дозволяє приймати WebSocket-повідомлення, виконувати оновлення прошивки (OTA), передавати телеметричні дані тощо.

ESP32 виступає центральним виконавчим модулем системи: він отримує команди із WebSocket-сервера, керує адресною світлодіодною стрічкою WS2812B, реагує на натискання локальних кнопок і передає зворотну інформацію, зокрема підтвердження виконання команд. Крім того, контролер у реальному часі керує анімаціями, кольорами та світловими ефектами.

На ESP32 реалізовано прошивку з використанням операційної системи реального часу FreeRTOS, що забезпечує паралельне виконання задач - обробку команд, керування світлодіодною стрічкою та взаємодію з мережею.

До переваг такої побудови комп’ютерної системи підсвітки сходів із віддаленим моніторингом можна віднести:

- розподіленість - розмежування логіки обробки (Lambda), зберігання конфігурацій і стану (S3) та інтерфейсу взаємодії (Telegram) підвищує надійність і забезпечує масштабованість системи;
- гнучкість - користувач має змогу керувати системою як локально, так і дистанційно, а розширення функціональних можливостей може здійснюватися без необхідності перепрошивання ESP32;
- надійність і безпека - інфраструктура AWS підтримує механізми авторизації, захищені WebSocket-з’єднання та зберігання резервних копій

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

конфігурацій;

– масштабованість - один Telegram-бот може взаємодіяти з кількома пристроями, кожен із яких використовує окремий канал WebSocket.

Архітектура комп'ютерної системи, наведена на рис. 3.1, має потенціал для подальшого розширення. Зокрема, можливими напрямками розвитку є інтеграція з голосовими помічниками Google Assistant або Alexa, додавання датчика освітленості для реалізації нічного режиму, впровадження автоматичного оновлення прошивки (OTA) через сервер, а також створення окремого мобільного застосунку замість Telegram-інтерфейсу.

Пристрій керування комп'ютерною системою підсвітки сходів

Мікроконтролер ESP32 є сучасним високопродуктивним вбудованим обчислювальним модулем, розробленим компанією Espressif Systems. Він поєднує значну обчислювальну потужність, енергоефективність і широкий набір інтерфейсів, що робить його придатним для реалізації проєктів у сфері Інтернету речей (IoT), автоматизованих систем керування, «розумного дому», інтелектуального моніторингу та, зокрема, систем підсвітки сходів із функціями дистанційного доступу.

У центральній частині ESP32 розміщений 32-бітний двоядерний процесор Tensilica Xtensa LX6, який працює на частоті до 240 МГц. Це забезпечує достатній рівень продуктивності для паралельного виконання кількох задач у режимі реального часу. Зовнішній вигляд модуля ESP32 наведено на рисунку 3.2.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

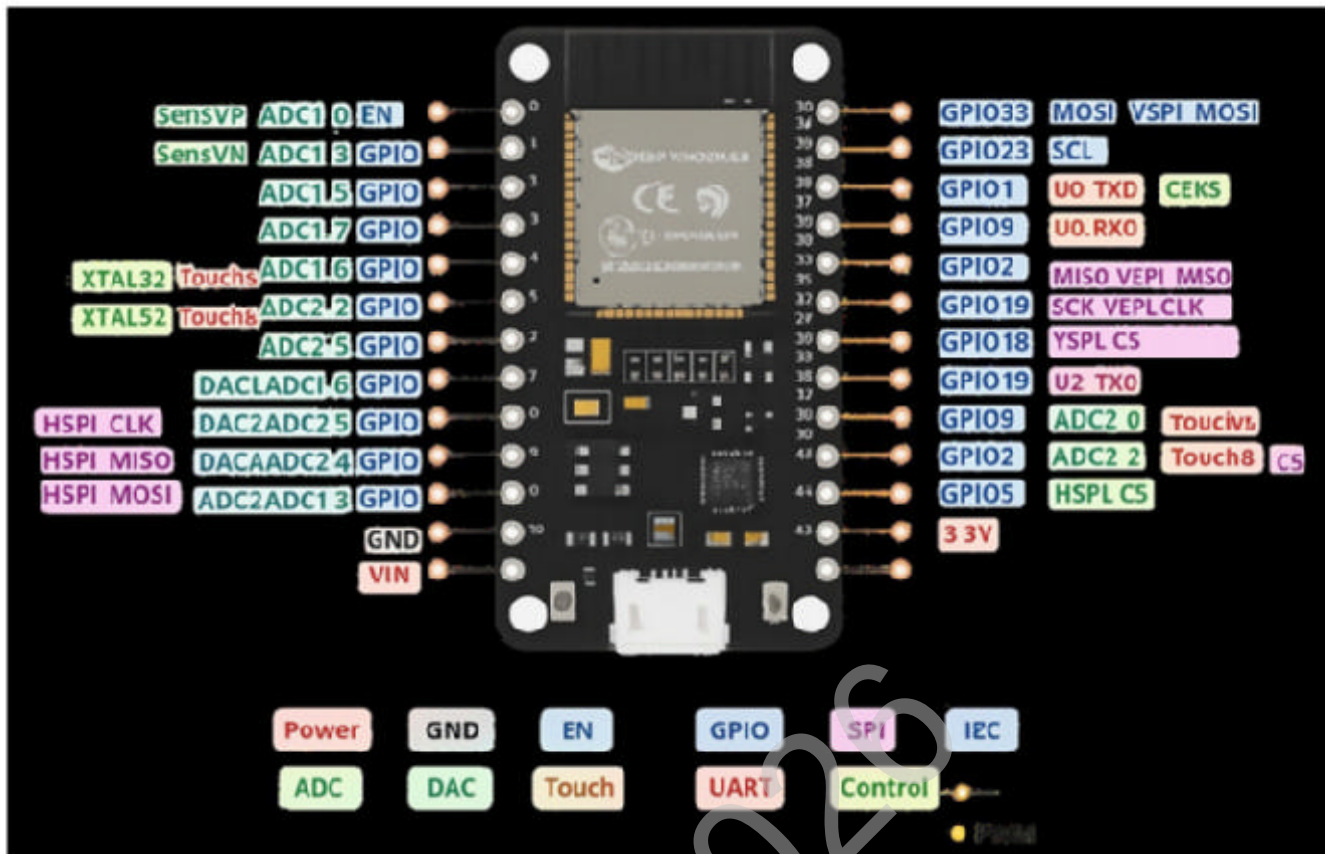


Рисунок 3.2 – Пристрій керування системи ESP32

Операційна система реального часу (наприклад, FreeRTOS), що може бути розгорнута на ESP32, забезпечує ефективне керування багатозадачними обчисленнями та дозволяє організувати паралельне виконання процесів.

Окрім обчислювального ядра, ESP32 оснащений вбудованим модулем Wi-Fi стандарту 802.11 b/g/n, а також підтримкою Bluetooth. Це дає змогу здійснювати бездротовий обмін даними з іншими пристроями, хмарними сервісами або мобільними застосунками без використання додаткового мережевого обладнання. Така інтеграція створює оптимальні умови для систем, у яких необхідне поєднання локального керування (наприклад, через кнопки) та дистанційного доступу (через Telegram-бот або вебінтерфейс).

ESP32 підтримує до 34 програмованих GPIO-контактів, які можуть використовуватися для зчитування сигналів із сенсорів (руху, освітленості, присутності), а також для керування виконавчими пристроями, такими як світлодіодні стрічки, реле чи звукові індикатори. Завдяки вбудованим апаратним

модулям. Завдяки наявності апаратних інтерфейсів PWM, SPI, I²C, ADC, DAC, UART і CAN мікроконтролер ESP32 легко інтегрується як у цифрові, так і в аналогові системи.

У задачах автоматизованої підсвітці сходів ESP32 дозволяє реалізувати гнучку логіку керування освітленням, зокрема плавне ввімкнення світлодіодної стрічки при виявленні руху, регулювання яскравості залежно від рівня освітленості або активацію освітлення відповідно до часу доби. Використання ESP32 також надає можливість зберігати та обробляти дані про події, передавати інформацію до хмарних сервісів (AWS, Blynk, Firebase) чи локальних серверів для подальшої аналітики та ведення журналу.

Важливою особливістю ESP32 є гнучке керування енергоспоживанням: мікроконтролер підтримує кілька режимів енергозбереження, зокрема deep sleep, що є актуальним для автономних систем із живленням від акумуляторів. Це дозволяє суттєво зменшити споживання енергії у періоди неактивності, зберігаючи здатність реагувати на зовнішні події, наприклад сигнал від датчика руху.

Ще однією перевагою ESP32 є активна підтримка спільноти розробників, наявність великої кількості бібліотек, SDK (зокрема ESP-IDF), відкритої документації та сумісність із популярними середовищами розробки, такими як Arduino IDE, PlatformIO, MicroPython і NodeMCU. Це значно спрощує створення прототипів і впровадження готових рішень.

Отже, ESP32 виступає ключовим елементом системи керування підсвіткою сходів, об'єднуючи апаратні й програмні компоненти в єдиний інтелектуальний комплекс із підтримкою автоматизації, дистанційного керування, мережевого обміну даними та високого рівня адаптивності до вимог користувача.

На рівні «розумного» пристрою мікроконтролер ESP32 безпосередньо взаємодіє зі світлодіодною підсвіткою сходів і сенсорами (наприклад, інфрачервоними датчиками руху, встановленими на початку та в кінці сходового прольоту). Прошивка, що виконується на ESP32, відстежує сигнали датчиків (виявлення руху, рівень освітлення тощо) та у реальному часі керує вмиканням і

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

вимиканням LED-підсвітки. Вбудований модуль Wi-Fi забезпечує підключення до локальної мережі через домашній маршрутизатор.

Маршрутизатор у цій системі виконує функцію шлюзу до Інтернету, забезпечуючи обмін даними між пристроєм і хмарними сервісами. Для коректної роботи необхідне стабільне підключення до мережі через Wi-Fi, що дозволяє ESP32 встановлювати зв'язок із хмарною інфраструктурою AWS. Таким чином, фізичний пристрій працює у межах локальної мережі, але використовує захищений канал зв'язку із сервером, що усуває потребу у прямому зовнішньому доступі до домашньої мережі.

У разі спрацювання датчика або зміни стану освітлення ESP32 формує повідомлення з відповідними даними та надсилає його до хмари для подальшої обробки. У цій архітектурі для комунікації застосовується служба AWS API Gateway, яка надає два типи інтерфейсів: HTTP (REST) і WebSocket. HTTP-інтерфейс використовується для обробки одноразових запитів або подій, зокрема для приймання повідомлень від Telegram-бота (вхідні webhooks), а також може застосовуватися самим пристроєм для періодичної передачі інформації.

WebSocket-інтерфейс API Gateway забезпечує постійне двостороннє з'єднання між ESP32 і хмарною частиною системи, що є критично важливим для роботи в режимі реального часу. Такий канал зв'язку дозволяє як пристрою надсилати повідомлення на сервер, так і серверу ініціювати передавання даних або команд на клієнт. Це дає змогу хмарному сервісу оперативно передавати команди на пристрій без додаткових запитів, наприклад негайно активувати освітлення за командою користувача. ESP32 підтримує постійне WebSocket-з'єднання, передаючи телеметричні дані (стан системи, події сенсорів) і очікуючи вхідні керувальні команди.

На стороні AWS усі взаємодії обробляються безсерверними функціями AWS Lambda. Кожен HTTP-запит або повідомлення WebSocket, що надходить через API Gateway, автоматично передається до відповідної Lambda-функції для подальшої обробки. AWS Lambda є serverless-сервісом виконання коду, який

дозволяє запускати програмну логіку у відповідь на події без необхідності адміністрування серверної інфраструктури.

У системі підсвітки сходів із дистанційним керуванням AWS Lambda виконує роль центрального елемента хмарної логіки, тобто «мозку» серверної частини. Саме тут реалізується основна бізнес-логіка автоматизації. Після отримання даних від ESP32 (наприклад, повідомлення про виявлення руху на сходах або факт увімкнення освітлення) Lambda-функція може виконати низку операцій: зберегти отриману інформацію, проаналізувати її та прийняти рішення щодо подальших дій. Наприклад, у разі фіксації руху система може надіслати користувачеві сповіщення через Telegram-бот або занести подію до журналу. Якщо ж отримано команду від користувача, Lambda обробляє її та формує відповідний сигнал для передачі на ESP32.

Безсерверна архітектура AWS забезпечує високу масштабованість і швидкодію таких функцій: середовище виконання автоматично підлаштовується під поточне навантаження, що дозволяє системі одночасно обробляти декілька подій без помітних затримок. Важливою перевагою є також відсутність необхідності підтримувати постійно працюючі сервери - обчислювальні ресурси надаються динамічно лише у момент виникнення подій, що підвищує ефективність використання ресурсів і надійність роботи системи.

3.2 Розробка структурної схеми

Структурна схема системи автоматизованого освітлення з дистанційним керуванням відображає взаємодію основних апаратних і програмних компонентів, що забезпечують збір даних, їх обробку, керування освітленням та обмін інформацією з користувачем.

Центральним елементом системи є мікроконтролер (наприклад, ESP32), який виконує функції обробки сигналів, реалізації логіки керування та забезпечення бездротового зв'язку. До його входів підключаються сенсори руху,

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

освітленості або присутності, які формують сигнали про зміну стану навколишнього середовища. Отримані дані аналізуються мікроконтролером, після чого приймається рішення щодо активації або деактивації освітлення.

Виконавчим блоком системи є освітлювальні елементи (світлодіодні стрічки або світильники), керування якими здійснюється через вихідні порти мікроконтролера або через силові ключі (реле, MOSFET-драйвери). Модуль бездротового зв'язку (Wi-Fi), інтегрований у мікроконтролер, забезпечує підключення до локальної мережі та взаємодію з хмарним сервером або мобільним застосунком. Через мобільний додаток користувач може змінювати режими роботи освітлення, налаштовувати яскравість, обирати сценарії або отримувати інформацію про стан системи в режимі реального часу.

Також структурна схема може містити блок живлення, який забезпечує стабілізоване електроживлення мікроконтролера та освітлювальних елементів, а також блок збереження даних (флеш-пам'ять), що використовується для збереження налаштувань системи. Таким чином, структурна схема відображає логічну організацію системи та інформаційні зв'язки між її складовими, забезпечуючи реалізацію автоматизованого та дистанційного керування освітленням.

На рисунку 3.3 зображено структурну схему запропонованої системи.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31



Рисунок 3.3 - Структурна схема системи

3.3 Розробка функціональної схеми

Функціональна схема системи автоматизованого освітлення відображає послідовність обробки сигналів і логіку взаємодії між окремими модулями під час її роботи.

Система функціонує за принципом подієвого керування. Вхідними елементами є датчики руху, присутності та освітленості, які формують сигнали про зміну умов у зоні контролю. Отримані сигнали надходять до мікроконтролера, де здійснюється їх аналіз відповідно до заданого алгоритму. На основі оброблених даних визначається необхідність увімкнення або вимкнення освітлення, вибір режиму роботи та рівня яскравості.

У разі виявлення руху за недостатнього рівня освітленості мікроконтролер формує керуючий сигнал на виконавчий блок (світлодіодну стрічку або світильник). Освітлення може працювати в одному з передбачених режимів: статичному, анімаційному або з можливістю зміни кольору. Після завершення

заданого інтервалу часу або за відсутності руху система автоматично вимикає освітлення для забезпечення енергоефективності.

Паралельно реалізується функція дистанційного керування. Через Wi-Fi-модуль мікроконтролер підтримує зв'язок із хмарним сервером або мобільним застосунком. Користувач може змінювати налаштування, активувати або деактивувати освітлення, обирати сценарії роботи та отримувати інформацію про поточний стан системи. Вхідні команди з мобільного додатка обробляються програмною логікою та впливають на параметри функціонування системи.

Таким чином, функціональна схема відображає алгоритмічну взаємодію між сенсорним блоком, модулем обробки даних, виконавчим блоком та модулем дистанційного зв'язку, забезпечуючи автоматизовану та гнучку роботу системи освітлення.

На рисунку 3.4 зображено функціональну схему системи.

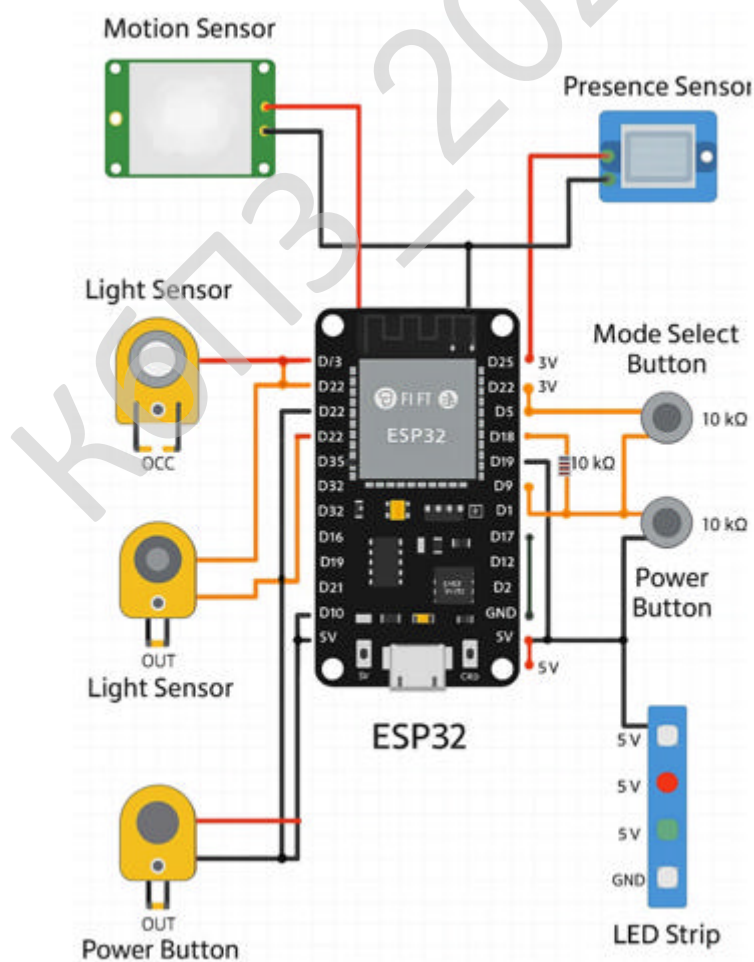


Рисунок 3.4 - Функціональна схема системи керування

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів відображає послідовність обміну даними між основними програмними та апаратними компонентами системи автоматизованого освітлення, а також логіку їх спільної роботи у часі.

Після подачі живлення ініціалізується мікроконтролер ESP32, запускаються основні програмні процеси та виконується перевірка підключених модулів: сенсорів руху, освітленості, кнопок керування і виконавчого блоку освітлення. Паралельно встановлюється з'єднання з мережею Wi-Fi та ініціюється зв'язок із хмарним сервером або мобільним застосунком.

У процесі роботи сенсорний модуль постійно формує події, пов'язані з виявленням руху або зміною рівня освітленості. Ці дані передаються до процесу обробки, де виконується аналіз умов і прийняття рішення щодо активації або деактивації освітлення. Після цього формується керуючий сигнал, який передається до виконавчого модуля - світлодіодної стрічки або світильника.

Одночасно функціонує процес дистанційної взаємодії. Користувач через мобільний застосунок або інший інтерфейс надсилає команди керування, які через мережу передаються до мікроконтролера. Програмний модуль приймає ці команди, обробляє їх і змінює параметри роботи системи (режим освітлення, яскравість, стан увімкнення). У зворотному напрямку передається інформація про поточний стан системи, включаючи активний режим, стан сенсорів та параметри освітлення.

Також у системі працює процес моніторингу і журналювання, який фіксує події, зміни станів і помилки, забезпечуючи можливість діагностики та аналізу роботи системи. Взаємодія всіх процесів здійснюється синхронно з використанням механізмів планування задач операційної системи реального часу, що гарантує стабільну, передбачувану та безперебійну роботу системи.

Таким чином, діаграма взаємодії процесів демонструє послідовність обміну повідомленнями між сенсорним, обчислювальним, виконавчим і

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

комунікаційним модулями, забезпечуючи узгоджену роботу всієї системи автоматизованого освітлення.

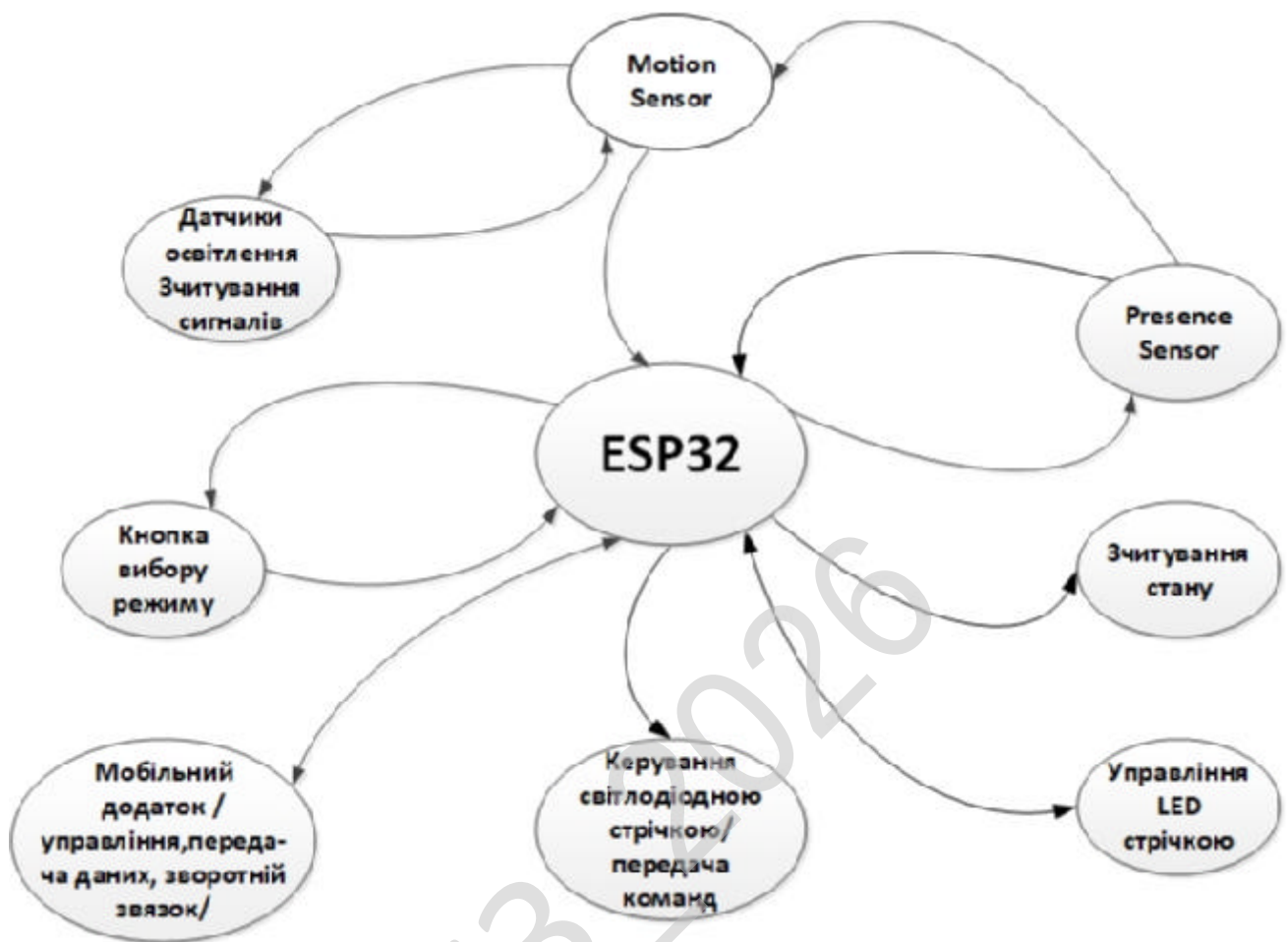


Рисунок 3.5 – Діаграма взаємодії процесів

Отже, після аналізу опису системи, структурної та функціональної схем, а також діаграми взаємодії процесів, переходимо до опису блок-схем основної програми та підпрограм, що використовуються для реалізації системи.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Розробка блок-схем та опис алгоритмів функціонування системи

Апаратно-програмане забезпечення

Сенсори системи підсвітки сходів з віддаленим моніторингом

Інфрачервоний модуль HC-SR501 є одним із найпоширеніших засобів автоматичного виявлення руху, що широко застосовується у вбудованих системах керування. Його робота базується на пасивній інфрачервоній технології, яка дозволяє фіксувати теплове випромінювання людини або інших об'єктів із температурою, що відрізняється від навколишнього середовища. Основним елементом конструкції є високочутливий сенсор LHI778, розміщений під напівсферичною лінзою Френеля, яка формує зону спостереження (рис. 4.1).

Однією з ключових переваг цього сенсора є низьке енергоспоживання, що дає змогу використовувати його у системах із живленням від акумуляторів або резервних джерел. Модуль широко застосовується у проєктах «розумного дому», охоронних системах, а також у побутовій електроніці, зокрема в системах автоматичної підсвітці сходів. Аналізуючи рівень інфрачервоного випромінювання у зоні спостереження, сенсор фіксує появу об'єкта та активує цифровий вихід, сигналізуючи про виявлення руху.



Рисунок 4.1 – Сенсор руху HC-SR501

До основних функціональних можливостей модуля HC-SR501 належать регулювання рівня чутливості, налаштування часу утримання вихідного сигналу після спрацювання (затримка вимкнення), а також перемикання режимів роботи між одиничним і повторним (циклічним) тригером. Пристрій може функціонувати як у режимі одноразової активації, коли сигнал на виході зберігається протягом заданого часу, так і в режимі безперервного утримання, за якого наявність руху підтримує вихід у стані логічної одиниці. Електрична принципова схема сенсора наведена на рисунку 4.2.

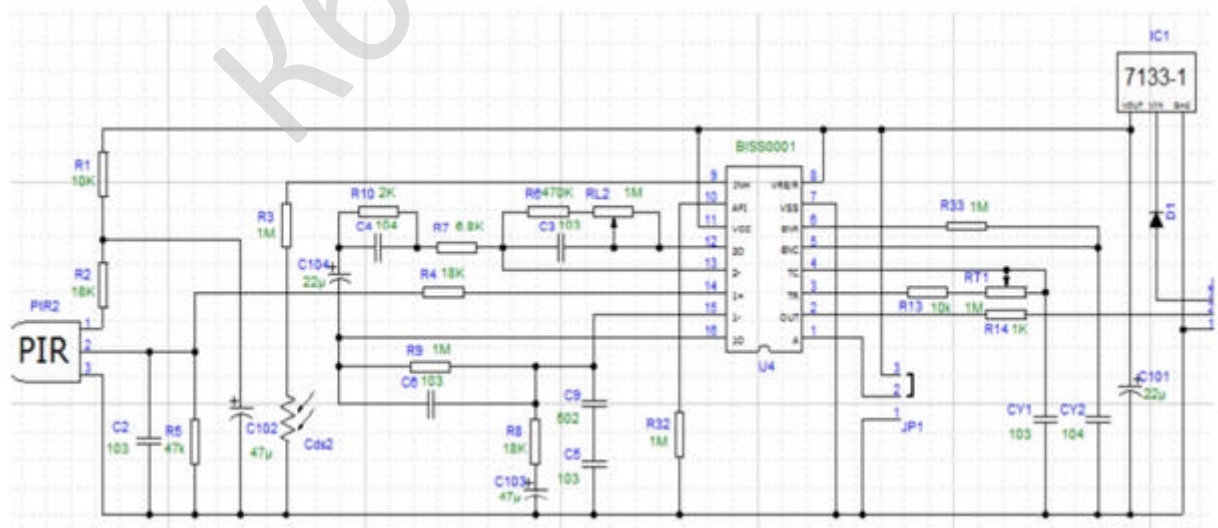


Рисунок 4.2 – Схема електрична принципова HC-SR501

Модуль також оснащений функцією температурної компенсації, що дозволяє сенсору зберігати ефективність навіть за підвищених температур навколишнього середовища. Це особливо важливо в літній період, коли чутливість стандартних PIR-датчиків може знижуватися.

Серед конструктивних особливостей варто виділити наявність потенціометрів для точного налаштування дальності спрацювання (до 7 м) і часу затримки (до 5 хвилин), контакти живлення (VCC, GND) та цифровий вихід (OUT), який формує TTL-сумісний сигнал для підключення до мікроконтролерів, зокрема ESP32. Розташування виводів HC-SR501 наведено на рисунку 4.3.

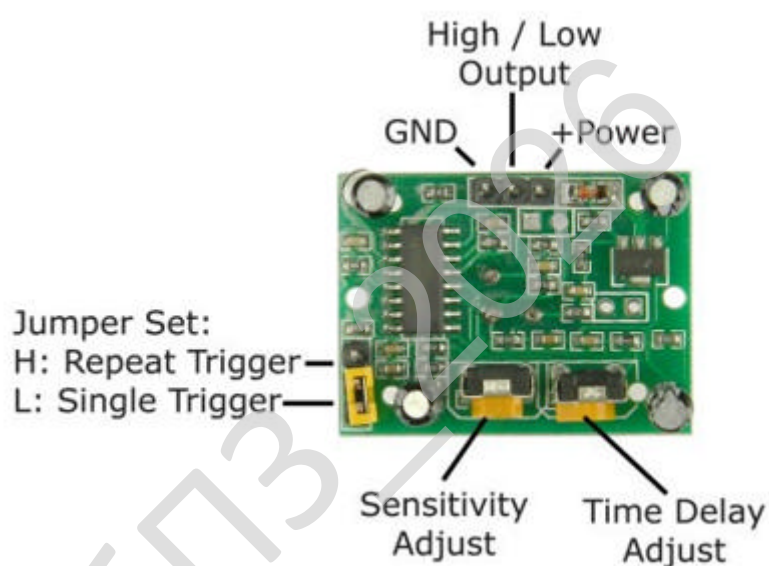


Рисунок 4.3 – Виводи HC-SR501

До основних технічних характеристик HC-SR501 належить:

- робоча напруга – від 5 до 20 В постійного струму;
- струм споживання – до 65 мА\$;
- цифровий вихід: рівень логічної одиниці - 3,3 В; нульовий рівень - 0В;
- регульований час затримки – від 0,3 с до 300 с;
- час блокування між спрацюваннями – 0,2 с (за замовчуванням);
- два режими роботи – одиничне спрацювання (режим “L”) та повторне спрацювання (режим “H”);
- кут огляду – приблизно 120°;

- дальність виявлення - до 7 м;
- робочий температурний діапазон - від -15 до +70 °С;
- струм у режимі очікування - менше 50 мкА.

Цей сенсор добре сумісний із мікроконтролером ESP32, який безпосередньо сприймає його цифровий вихідний сигнал, що дозволяє створювати надійні та енергоефективні рішення для систем підсвітки або безпеки. Схему підключення сенсора до ESP32 наведено на рисунку 4.4.

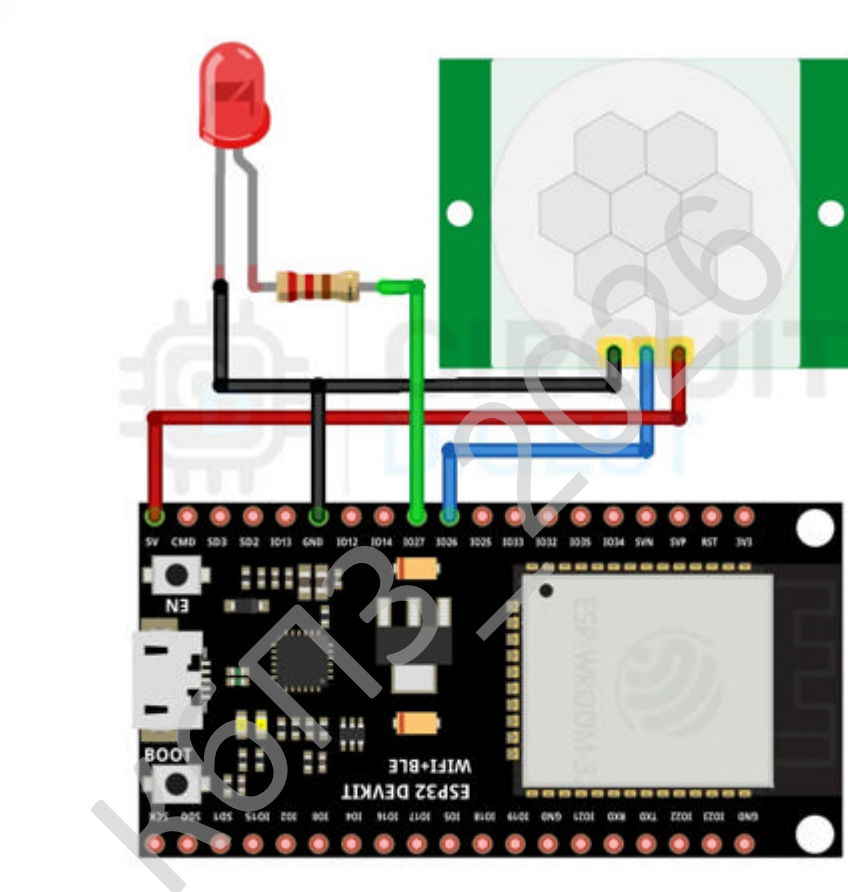


Рисунок 4.4 – Схема підключення HC-SR501 до ESP32

Сенсор присутності є інтелектуальним елементом автоматизованих систем керування, що дозволяє визначати наявність об'єкта (зазвичай людини) у заданій зоні спостереження. На відміну від традиційних датчиків руху, які реагують лише на зміну положення об'єкта, сенсор присутності здатний фіксувати навіть мінімальну активність або статичну присутність, наприклад перебування людини без руху чи незначні переміщення.

Саме ця властивість робить його важливим компонентом у системах освітлення, де важливо підтримувати освітлення увімкненим доти, доки користувач перебуває у зоні дії системи, навіть за відсутності активного руху.

Сенсори присутності можуть базуватися на різних принципах роботи. У побутових і вбудованих системах найчастіше застосовуються два типи: інфрачервоні (IR) та мікрохвильові (радар Доплера). У проєктах на основі ESP32 перевагу часто надають саме мікрохвильовим датчикам, зокрема модулям типу RCWL-0516, завдяки їх здатності працювати крізь неметалеві перешкоди (дерево, пластик, скло) та високій чутливості навіть до незначних переміщень (рис. 4.5).

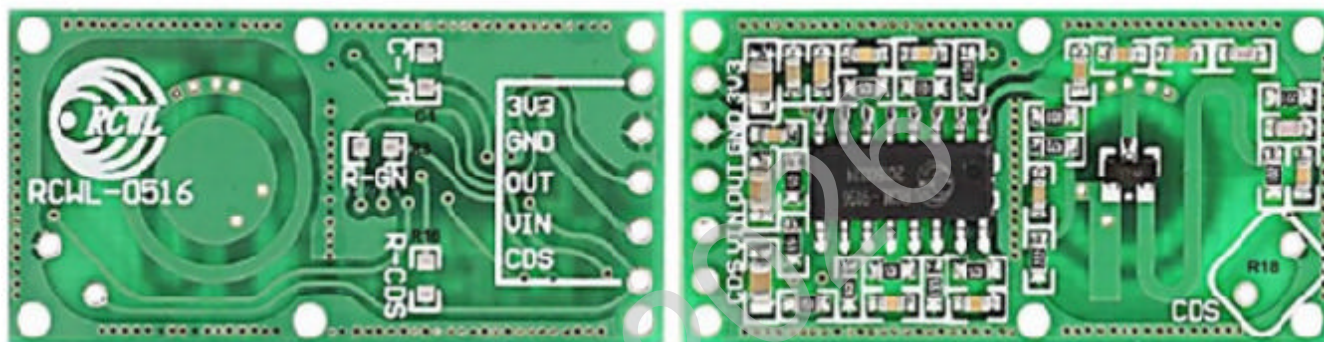


Рисунок 4.5 – Датчик присутності RCWL-0516

Принцип роботи мікрохвильового сенсора ґрунтується на ефекті Доплера: модуль генерує електромагнітні хвилі високої частоти (близько 3,2 ГГц) та аналізує сигнал, відбитий від об'єктів у зоні дії. У разі наявності об'єкта навіть незначні його переміщення змінюють частоту відбитого сигналу, що фіксується мікросхемою сенсора як ознака присутності.

Модулі цього типу мають цифровий вихід, який може безпосередньо зчитуватися мікроконтролером ESP32, що значно спрощує їх інтеграцію у системи керування освітленням або сигналізацією. Вихідний сигнал утримується на високому логічному рівні доти, доки об'єкт перебуває в зоні чутливості сенсора, що забезпечує безперервне функціонування освітлення чи інших керованих пристроїв без повторних спрацювань.

Завдяки компактним розмірам, низькому енергоспоживанню та високій точності виявлення сенсор присутності добре підходить для встановлення у

сходових прольотах, коридорах або технічних приміщеннях. Схему підключення модуля RCWL-0516 до ESP32 наведено на рисунку 4.6.

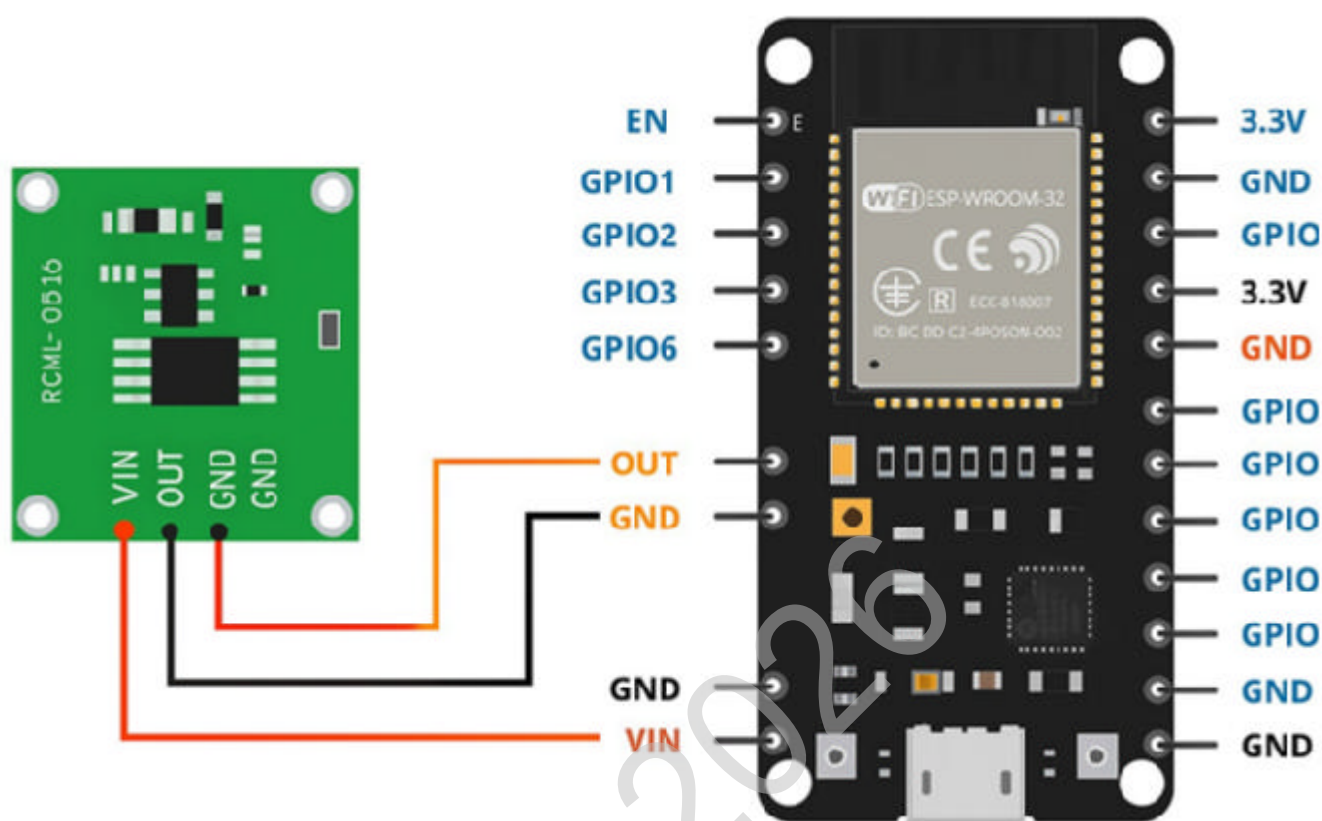


Рисунок 4.6 – Схема підключення RCWL-0516 до ESP32

Основні технічні характеристики модуля RCWL-0516 такі:

- робоча напруга живлення – 4 - 28 В постійного струму;
- споживаний струм — близько 3 мА;
- робоча частота — приблизно 3,2 ГГц (мікрохвильовий радар);
- час утримання вихідного сигналу — близько 2 секунд (може змінюватися залежно від налаштувань або прошивки);
- дальність виявлення - 5–7 м;
- кут покриття - до 360° у горизонтальній площині;
- вихідний сигнал - TTL-рівень (логічна «1» - 3,3 В, логічний «0» - 0 В);
- здатність працювати крізь нетеплопровідні матеріали, такі як скло, дерево або пластик.

У межах інтелектуальної системи підсвітки сходів і моніторингу сенсор

присутності забезпечує більш гнучку реакцію на поведінку користувача, підтримуючи освітлення увімкненим не лише під час короткочасного руху, а й у разі тривалого перебування в зоні контролю. Це підвищує зручність користування, зменшує кількість помилкових вимкнень і загалом покращує рівень комфорту та безпеки автоматизованої системи.

У системах автоматизованого керування освітленням, зокрема в інтелектуальних підсвітках сходів, датчики освітленості відіграють важливу роль у визначенні поточного рівня зовнішнього світла та прийнятті рішення щодо ввімкнення або вимкнення LED-підсвітки. Для реалізації цієї функції можуть використовуватися як прості аналогові компоненти, наприклад фоторезистори, так і спеціалізовані цифрові сенсори, зокрема BH1750.

Фоторезистор є світлочутливим резистивним елементом, опір якого змінюється залежно від рівня освітленості: зі зростанням інтенсивності світла опір зменшується. Для інтеграції LDR із мікроконтролером ESP32 застосовується простий аналоговий подільник напруги, де фоторезистор з'єднаний послідовно з резистором фіксованого номіналу. Напруга з точки поділу подається на аналоговий вхід ESP32 (наприклад, GPIO34) і перетворюється в цифрове значення за допомогою вбудованого АЦП, що дозволяє оцінити рівень освітленості. Такий метод є простим і недорогим, однак має певні обмеження щодо точності, стабільності вимірювань і температурної залежності.

Альтернативою фоторезистору є цифровий сенсор освітленості BH1750 (рис. 4.7), спеціально розроблений для вимірювання інтенсивності світла у фізичних одиницях - люксах. Сенсор працює через інтерфейс I²C, що дозволяє отримувати точні дані без використання зовнішнього аналого-цифрового перетворення.

Модуль BH1750 містить вбудований фотодіод, підсилювач і АЦП, що забезпечують високу точність вимірювання (до 1 люкса) при мінімальних похибках.

Завдяки використанню стандартного інтерфейсу I²C (підключення через

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

лінії SDA та SCL, наприклад до GPIO21 і GPIO22 мікроконтролера ESP32), сенсор легко інтегрується у систему. Живлення модуля може здійснюватися як від 3,3 В, так і від 5 В, що забезпечує його сумісність із більшістю сучасних мікроконтролерів.



Рисунок 4.7 – Цифровий сенсор BH1750

Сенсор BH1750 підтримує програмне налаштування чутливості та роботу в кількох режимах вимірювання - від швидкого до високоточного. Завдяки цьому він добре підходить для систем, у яких необхідно динамічно регулювати рівень підсвітки залежно від умов зовнішнього освітлення.

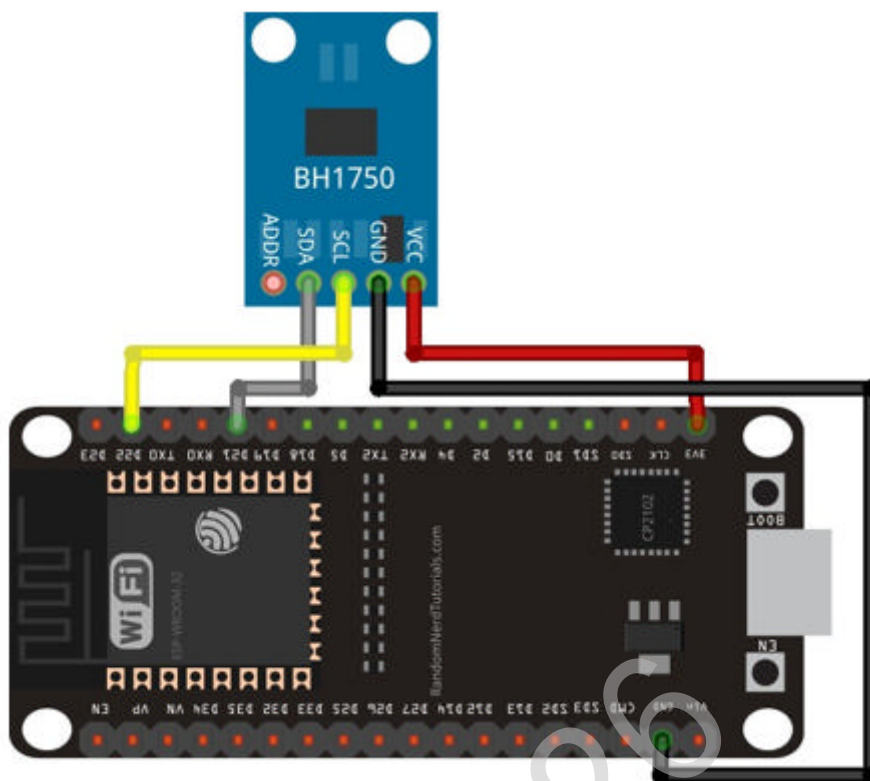


Рисунок 4.8 – Схема підключення BH1750 з ESP32

Фоторезистор у поєднанні з подільником напруги є простим і недорогим рішенням для базових застосувань, де важливі швидкість реалізації та мінімальна вартість. Водночас сенсор BH1750 забезпечує вищу точність і стабільність вимірювань у складніших умовах. У системах «розумної» підсвітці сходів датчики освітленості дозволяють активувати підсвітку лише за недостатнього природного освітлення (наприклад, у темний час доби або в затемнених приміщеннях), що сприяє зниженню енергоспоживання та підвищенню зручності користування.

Схема, наведена на рисунку 4.8, є типовою та надійною для більшості IoT-проектів. Підключення модуля BH1750 до ESP32 відповідно до описаної конфігурації забезпечує простий і точний спосіб визначення рівня освітленості у системі підсвітці сходів.

Типову схему підключення світлодіодних стрічок показано на рисунку 4.9.

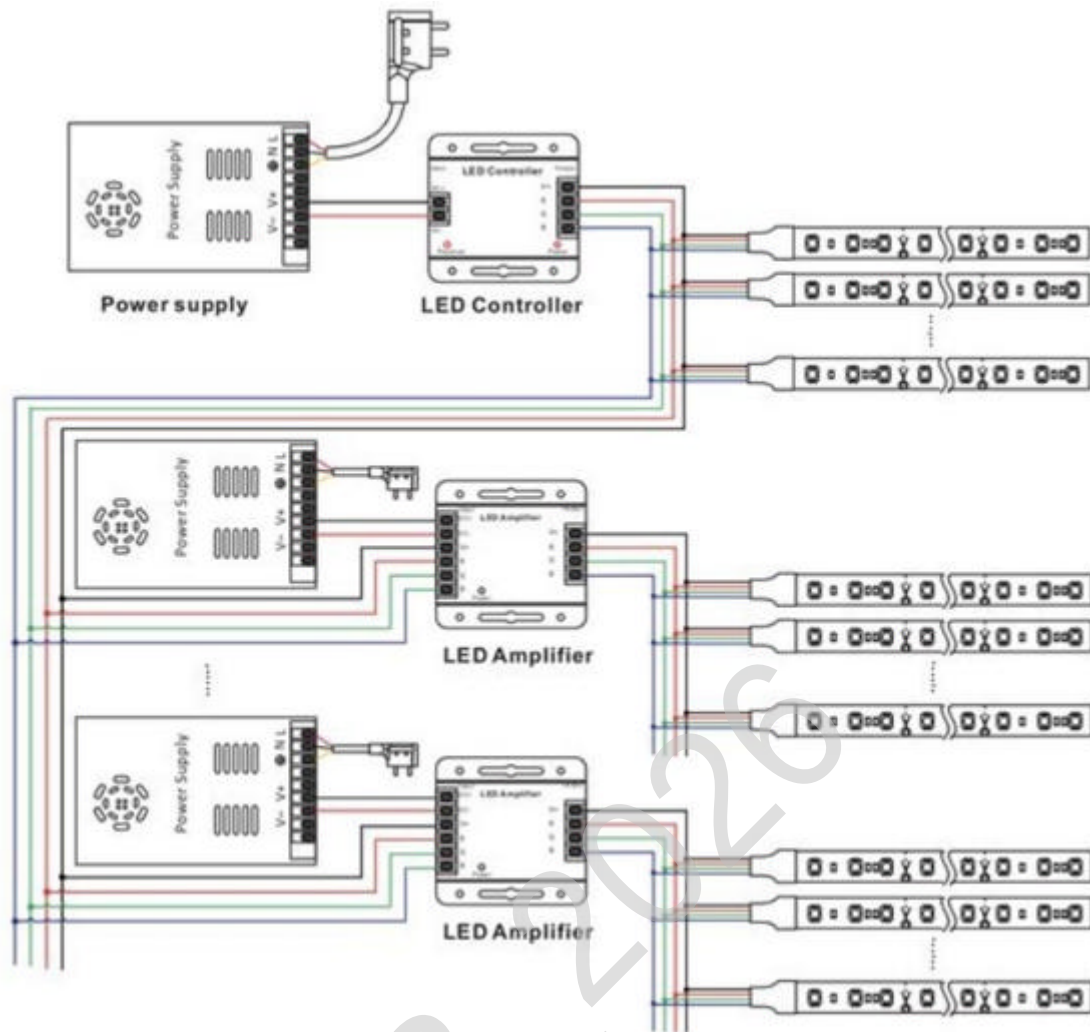


Рисунок 4.9 – Типова схема підключення світлодіодних стрічок

4.2 Алгоритм функціонування системи

Алгоритм функціонування системи на локальному програмному рівні передбачає виконання послідовності дій і перевірку умов, що визначають поточний режим і стан компонентів. Після подачі живлення мікроконтролер ESP32 здійснює ініціалізацію внутрішніх і зовнішніх модулів (датчиків, світлодіодної стрічки, кнопок), а також налаштовує режими роботи GPIO. Далі встановлюється з'єднання з мережею Wi-Fi і виконується підключення до сервера моніторингу через MQTT, WebSocket або API, після чого активується Telegram-бот.

Після успішної ініціалізації системи запускається безперервний процес

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

зчитування даних із сенсорів. Контролер із заданою періодичністю виконує:

- вимірювання рівня освітленості за допомогою цифрового сенсора ВН1750;
- зчитування сигналів із датчика руху HC-SR501 для виявлення руху;
- визначення присутності людини за допомогою сенсора RCWL-0516;
- контроль стану кнопок увімкнення та перемикання режимів;
- аналіз активного режиму роботи системи (автоматичний, ручний, нічний тощо).

Логіка керування підсвіткою базується на аналізі даних сенсорів і поточного режиму функціонування. Якщо зафіксовано рух або присутність людини та рівень освітленості нижчий за встановлений поріг, система активує світлодіодну стрічку, забезпечуючи освітлення сходів.

У разі активації нічного режиму система використовує знижений рівень яскравості освітлення. Якщо протягом заданого інтервалу часу рух або присутність не фіксуються, автоматично запускається процедура поступового вимкнення підсвітки. Загальний алгоритм роботи системи на локальному рівні керування наведено на рисунку 4.10.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

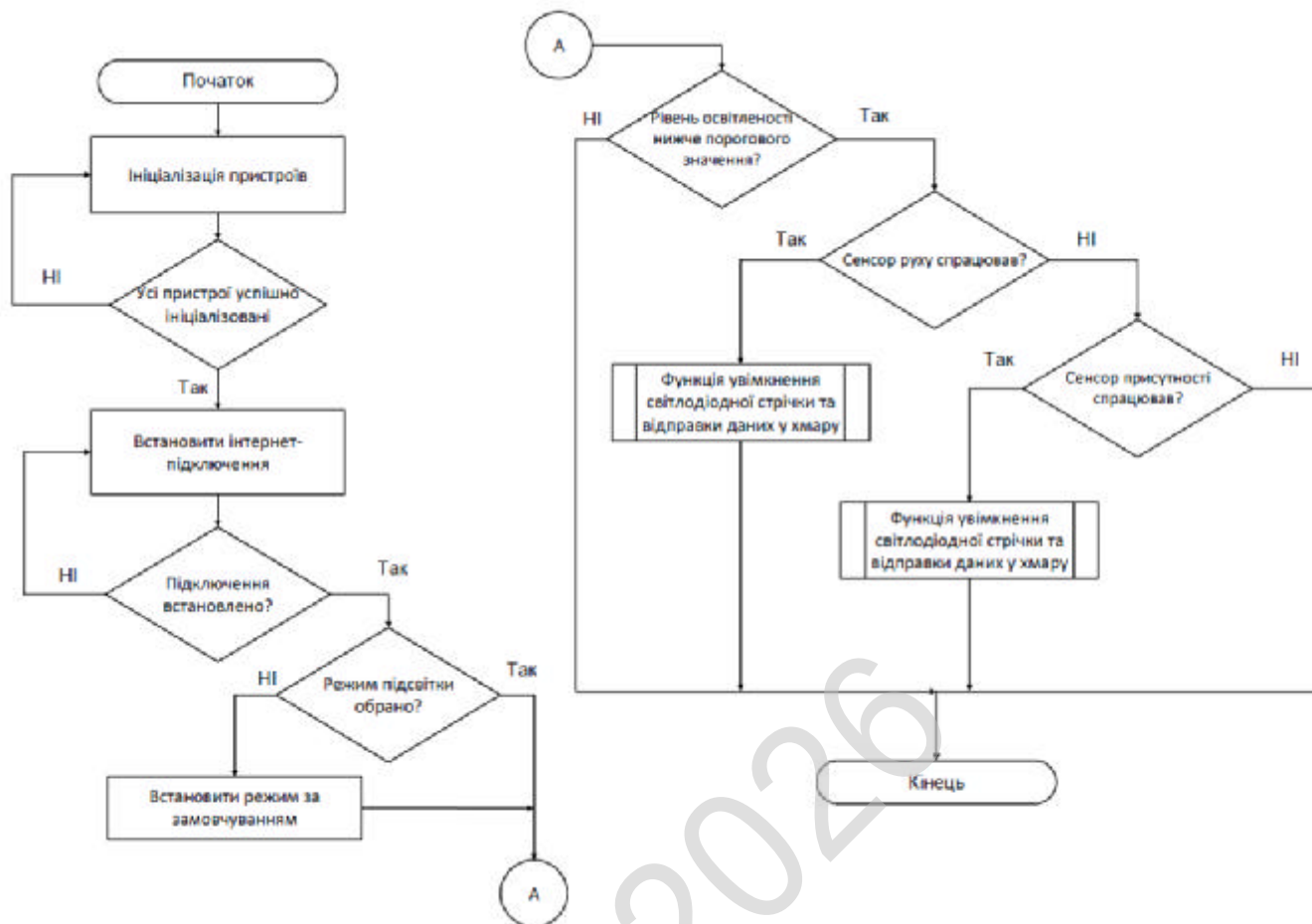


Рисунок 4.10 – Алгоритм роботи системи на локальному рівні

За допомогою кнопок локального керування користувач може вручну вмикати підсвітку та перемикає режими роботи системи: автоматичний, нічний, постійний або вимкнений.

Взаємодія з віддаленим сервером і Telegram-ботом відбувається під час подій, пов'язаних зі спрацюванням сенсорів або зміною режиму підсвітки. У такому разі формується повідомлення, яке надсилається в Telegram, а відповідні дані зберігаються у хмарному сховищі або на локальному сервері. Сервер, своєю чергою, може передати команду на ESP32 (наприклад, увімкнути освітлення), обробити запит користувача через Telegram-бота або виконати автоматичний сценарій, наприклад активацію нічного режиму.

Для підвищення рівня безпеки та надійності системи в перспективі можна

передбачити реалізацію додаткових захисних механізмів. Зокрема, доцільно впровадити звукову індикацію у випадку виникнення несправностей або тривалого перебування людини у зоні дії сенсорів, можливість аварійного вимкнення системи за допомогою фізичної кнопки, а також функцію програмного перезавантаження - як через Telegram-інтерфейс, так і за допомогою апаратної кнопки.

Зберігання даних у хмарі (AWS S3)

Важливою складовою хмарної інфраструктури є сховище даних AWS S3 (Amazon Simple Storage Service), яке використовується для надійного зберігання історичної та статичної інформації, пов'язаної з функціонуванням системи. Це високодоступне й масштабоване об'єктне сховище забезпечує безпечну та довготривалу платформу для збереження і отримання даних.

У системі підсвітки сходів у сховищі S3 можуть накопичуватися журнали подій (наприклад, часові мітки спрацювання датчиків руху та моментів увімкнення підсвітки), статистичні дані використання, конфігураційні файли (режими роботи освітлення, налаштування яскравості тощо) та інша інформація, необхідна для моніторингу й аналізу.

S3 виступає централізованим репозиторієм таких даних, гарантуючи їхню збереженість і доступність для подальшої обробки. Завдяки автоматичній реплікації даних у кількох зонах доступності досягається висока відмовостійкість і постійна доступність інформації. Це дозволяє користувачеві у будь-який момент переглянути історію подій або змін налаштувань, а також використовувати ці дані іншими хмарними сервісами, наприклад для аналітики чи візуалізації.

Крім того, на базі S3 можна реалізувати простий вебінтерфейс: розмістивши статичну вебсторінку (HTML/JavaScript) у бакеті, можна відображати дані IoT у графічному вигляді та отримувати доступ до них з будь-якого пристрою. Такий підхід розширює можливості віддаленого моніторингу, дозволяючи користувачеві спостерігати за станом підсвітки через веббраузер або мобільний застосунок із зручним графічним інтерфейсом, що доповнює текстові

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

повідомлення Telegram-бота.

Telegram-бот як інтерфейс користувача

Для забезпечення зручної взаємодії користувача із системою інтегровано Telegram-бот, який виконує роль посередника між користувачем і хмарною інфраструктурою. Telegram є хмарним месенджером, доступним на мобільних пристроях і комп'ютерах, що підтримує створення ботів — автоматизованих облікових записів, керування якими здійснюється через API. Взаємодія з ботом відбувається у форматі звичайного чату: користувач надсилає текстові команди або використовує інтерактивні кнопки, а бот відповідає повідомленнями з інформацією чи підтвердженням виконання дій.

З технічної точки зору Telegram-бот працює таким чином, що всі повідомлення користувача перенаправляються серверами Telegram на визначену URL-адресу, яким у даній системі виступає ендпойнт AWS API Gateway (HTTP). Тобто Telegram передає запит до хмарної інфраструктури щоразу, коли бот отримує нове повідомлення. Далі цей запит обробляється API Gateway і передається до Lambda-функції, відповідальної за обробку команд. Такий підхід дозволяє реалізувати роботу бота на безсерверній платформі AWS, обробляючи запити лише у момент їх надходження, без необхідності підтримки постійно працюючого сервера, що підвищує ефективність і зменшує витрати ресурсів.

Lambda-функція, яка реалізує логіку бота, аналізує вхідне повідомлення та визначає отриману команду або запит. Наприклад, користувач може надіслати команду /lights_on для примусового ввімкнення підсвітки сходів. Після отримання такого запиту функція перевіряє права доступу, переконуючись, що команда надійшла від авторизованого користувача, і формує відповідне повідомлення для пристрою.

Через WebSocket-канал API Gateway ця команда миттєво передається на ESP32, де її обробляє прошивка мікроконтролера. Контролер виконує команду - у даному випадку вмикає світлодіодну стрічку - та може надіслати зворотне повідомлення про результат виконання. Після отримання підтвердження від

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

ESP32 Lambda-функція через Telegram Bot API надсилає користувачеві повідомлення в чаті, інформуючи про успішне виконання команди.

Обмін даними відбувається практично миттєво завдяки постійному з'єднанню: хмарний сервіс у будь-який момент може передати інформацію до клієнтського застосунку, тобто в Telegram, без необхідності періодичних запитів з боку користувача. Крім того, користувач має можливість у будь-який час отримати поточний стан системи. Для цього використовується команда /state: після її надсилання Lambda-функція звертається до ESP32 через WebSocket із запитом надати актуальні дані. Отримавши відповідь від пристрою, хмарний сервіс передає інформацію користувачеві, і бот формує текстове повідомлення зі звітом про стан системи. Такий підхід забезпечує інтерактивний контроль, дозволяючи користувачеві в будь-який момент дізнатися про поточний стан системи, просто надіславши запит у чаті.

Сценарії взаємодії та сповіщення

Архітектура системи підсвітки сходів із функцією віддаленого моніторингу підтримує різні сценарії обміну даними між пристроєм і користувачем, охоплюючи як керувальні запити з боку користувача, так і автоматичні сповіщення від пристрою.

По-перше, користувач може дистанційно керувати освітленням сходів. Надіслана через Telegram команда передається боту і майже миттєво виконується пристроєм. При цьому немає необхідності перебувати в межах локальної мережі - достатньо мати доступ до Інтернету зі смартфона. Такий підхід забезпечує глобальний доступ: незалежно від місця перебування власник може керувати своєю системою освітлення.

По-друге, система здатна самостійно інформувати користувача про певні події. Наприклад, якщо датчик руху фіксує активність на сходах за відсутності господаря, ESP32 формує тривожне повідомлення і через хмарний шлюз ініціює його передачу користувачеві. Lambda-функція, налаштована на обробку таких подій, викликає API Telegram і надсилає попередження на мобільний пристрій.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Таким чином, система поєднує функції безпеки - сповіщення про небажану активність - і автоматизації, спрямованої на підвищення комфорту користування.

Користувач отримує сповіщення на смартфон практично миттєво незалежно від свого місцезнаходження, за умови наявності підключення до Інтернету. Це підвищує зручність і впевненість у роботі системи: власник може оперативно реагувати на події або просто відстежувати, коли і як функціонує освітлення.

Основним користувацьким інтерфейсом є мобільний застосунок Telegram, оскільки саме в ньому інтегровано керувального бота. Такий підхід робить взаємодію з системою простою та зрозумілою, адже керування здійснюється через звичайний чат без необхідності встановлення спеціалізованого програмного забезпечення. З погляду користувацького досвіду це зручно: достатньо відкрити Telegram, щоб отримувати повідомлення від системи або надсилати їй команди, подібно до звичайного спілкування у месенджері.

Водночас архітектура системи передбачає можливість підключення розширеного інтерфейсу за потреби. Наприклад, може бути розроблений окремий мобільний застосунок або вебсторінка, що взаємодіятиме з тим самим API Gateway через HTTPS-запити або WebSocket-з'єднання для отримання даних у реальному часі. Такий клієнтський інтерфейс може відображати графічні індикатори стану підсвітки, статистику використання, надавати розширені налаштування та інші функції, забезпечуючи більш глибоку взаємодію з системою.

У підсумку користувач отримує кілька варіантів керування: простий текстовий інтерфейс через Telegram-бот або більш функціональний графічний інтерфейс. Обидва підходи можуть використовуватися паралельно, доповнюючи один одного та підвищуючи зручність експлуатації системи.

Роль хмарної інфраструктури та переваги архітектури

Хмарна інфраструктура на базі AWS відіграє визначальну роль у забезпеченні віддаленого доступу, обробки даних і взаємодії між різними компонентами системи. Вона виступає посередником між фізичним пристроєм і

користувачем, виконуючи функції обробки логіки роботи, зберігання інформації та інтеграції із зовнішніми сервісами, зокрема Telegram. Такий підхід забезпечує низку переваг, серед яких автоматизація процесів, висока масштабованість, підвищений рівень безпеки та гнучкість у розвитку системи.

По-перше, автоматизація процесів суттєво спрощується завдяки програмованості хмарної логіки. Використовуючи AWS Lambda, можна реалізувати складні алгоритми реагування на події, які виконуються автоматично без участі користувача. Система здатна самостійно приймати рішення - наприклад, вмикати підсвітку лише у темний час доби або після одночасного спрацювання кількох датчиків, надсилати повторні сповіщення при тривалому виявленні руху тощо. Достатньо один раз задати необхідні правила, після чого хмарна інфраструктура забезпечує їх стабільне виконання, оскільки Lambda автоматично запускається при кожній відповідній події. У результаті досягається високий рівень автоматизації: система самостійно керує освітленням за заданими умовами та інформує користувача про важливі події, підвищуючи комфорт і функціональність.

По-друге, використання хмарної інфраструктури забезпечує практично необмежену масштабованість рішення. Платформа AWS дозволяє легко адаптувати систему до зростання кількості пристроїв або обсягу даних. За потреби можна підключити додаткові сходові прольоти, нові сенсори чи навіть розгорнути аналогічні системи у кількох будівлях. Для цього достатньо додати новий пристрій на базі ESP32 з унікальним ідентифікатором до вже існуючої хмарної інфраструктури.

Сервіси API Gateway і Lambda автоматично масштабуються відповідно до навантаження, а сховище S3 забезпечує збереження додаткових даних без обмежень. Завдяки використанню серверless-архітектури система може розширюватися від одного пристрою до сотень без суттєвих змін у структурі, причому масштабованість стосується не лише кількості пристроїв, а й рівня навантаження на систему.

У разі раптового зростання кількості подій, наприклад коли кілька користувачів одночасно надсилають команди або численні датчики генерують сигнали, AWS Lambda автоматично запускає паралельні екземпляри функцій у міру потреби. Це дозволяє обробляти запити практично без затримок, забезпечуючи стабільну швидкодію. У результаті користувачі не відчують погіршення якості обслуговування, а система зберігає високу продуктивність і швидкість реагування.

По-третє, важливою перевагою обраної архітектури є високий рівень безпеки. Насамперед, усі з'єднання в системі є захищеними: ESP32 взаємодіє з AWS через протокол HTTPS із використанням TLS-шифрування, а обмін даними між Telegram-ботом і хмарою здійснюється через зашифровані вебхуки. Зі свого боку, Telegram шифрує передачу повідомлень між клієнтом і сервером, що забезпечує захист каналу зв'язку від смартфона користувача до Lambda-функції. AWS надає розвинені механізми автентифікації, шифрування та керування доступом для безпечної взаємодії IoT-пристроїв із хмарною інфраструктурою.

У системі дистанційного моніторингу підсвітки сходів можуть використовуватися токени або API-ключі для авторизації запитів ESP32 до API Gateway, що гарантує передачу даних лише від довірених пристроїв. Аналогічно, Telegram-бот може бути налаштований на обробку повідомлень тільки від визначеного користувача за допомогою унікального chat ID, що запобігає несанкціонованому доступу. Дані, які зберігаються у сховищі S3, можуть бути зашифровані як на рівні сервісу, так і на рівні застосунку, а політики доступу забезпечують можливість їх перегляду лише авторизованим користувачем. Відсутність прямого зовнішнього доступу до домашньої мережі (відкритих портів) означає, що сторонні особи не можуть безпосередньо підключитися до ESP32 — усі запити проходять через контрольований шлюз AWS, який фільтрує небажаний трафік.

Таким чином, архітектура та алгоритм роботи системи поєднують фізичну безпеку (застосування сенсорів і автоматичного освітлення, що підвищує

захищеність у темний час доби) та кібербезпеку (захищені канали зв'язку, механізми автентифікації й авторизації). Це дозволяє користувачеві бути впевненим у надійному захисті даних і безпечному керуванні пристроями за допомогою сучасних технологій шифрування та контролю доступу.

По-четверте, гнучкість і модульність архітектури забезпечують простоту подальшого розвитку системи. У разі появи нових вимог, наприклад інтеграції з іншими компонентами «розумного дому», реалізації додаткових сценаріїв (реакція на пожежну сигналізацію, взаємодія з голосовими помічниками тощо), це можна здійснити без зміни базової структури. Хмарна архітектура дозволяє безперешкодно додавати нову функціональність: достатньо створити нову Lambda-функцію або розширити існуючу, додати відповідний ендпойнт у API Gateway чи організувати зберігання нового типу даних у S3.

У разі потреби, окрім Telegram, система може бути інтегрована з платформами Google Home або Alexa. Для цього відповідний сервіс підключається до тієї ж хмарної інфраструктури через додаткові Lambda-функції та API Gateway, що дозволяє обробляти голосові команди без впливу на існуючу логіку Telegram чи прошивку ESP32. Така гнучкість досягається завдяки слабкому зв'язуванню компонентів: пристрій, шлюз, обчислювальні модулі та бот взаємодіють через стандартизовані інтерфейси й не залежать від внутрішньої реалізації один одного, що значно спрощує обслуговування та оновлення системи. Крім того, serverless-підхід AWS спрощує процес розгортання оновлень і внесення змін.

У підсумку поєднання апаратної платформи ESP32, хмарної інфраструктури AWS і месенджера Telegram формує цілісну комп'ютерну систему автоматизації освітлення, що характеризується надійністю, ефективністю та гнучкістю. Вона демонструє переваги використання хмарних технологій у IoT-проектах: забезпечується безперервний зв'язок і якісний обмін даними між пристроєм і користувачем, водночас зберігається можливість масштабування та подальшого вдосконалення.

Система автоматично реагує на зміни зовнішніх умов, підвищуючи

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

комфорт і безпеку, а користувач отримує зручні інструменти керування через звичний мобільний інтерфейс. Завдяки хмарній архітектурі рішення є масштабованим, захищеним від несанкціонованого доступу та відмов окремих компонентів, а також легко адаптується до нових вимог і сценаріїв використання.

Запропонована архітектура є прикладом сучасного підходу до побудови IoT-систем, оскільки поєднує автономну роботу кінцевого пристрою з можливостями хмарних сервісів. Такий підхід забезпечує оптимальний баланс між локальним керуванням і дистанційним моніторингом, підвищуючи рівень автоматизації, надійність та гнучкість функціонування системи загалом.

К6ПЗ_2026

5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Схема підключення пристроїв комп'ютерної системи підсвітки сходів з віддаленим керуванням

Схему підключення компонентів комп'ютерної системи керування підсвіткою сходів на базі мікроконтролера ESP32 наведено на рисунку 5.1. Вона охоплює сенсори виявлення руху, присутності та рівня освітленості, а також кнопки, призначені для ручного керування і вибору режимів роботи системи.

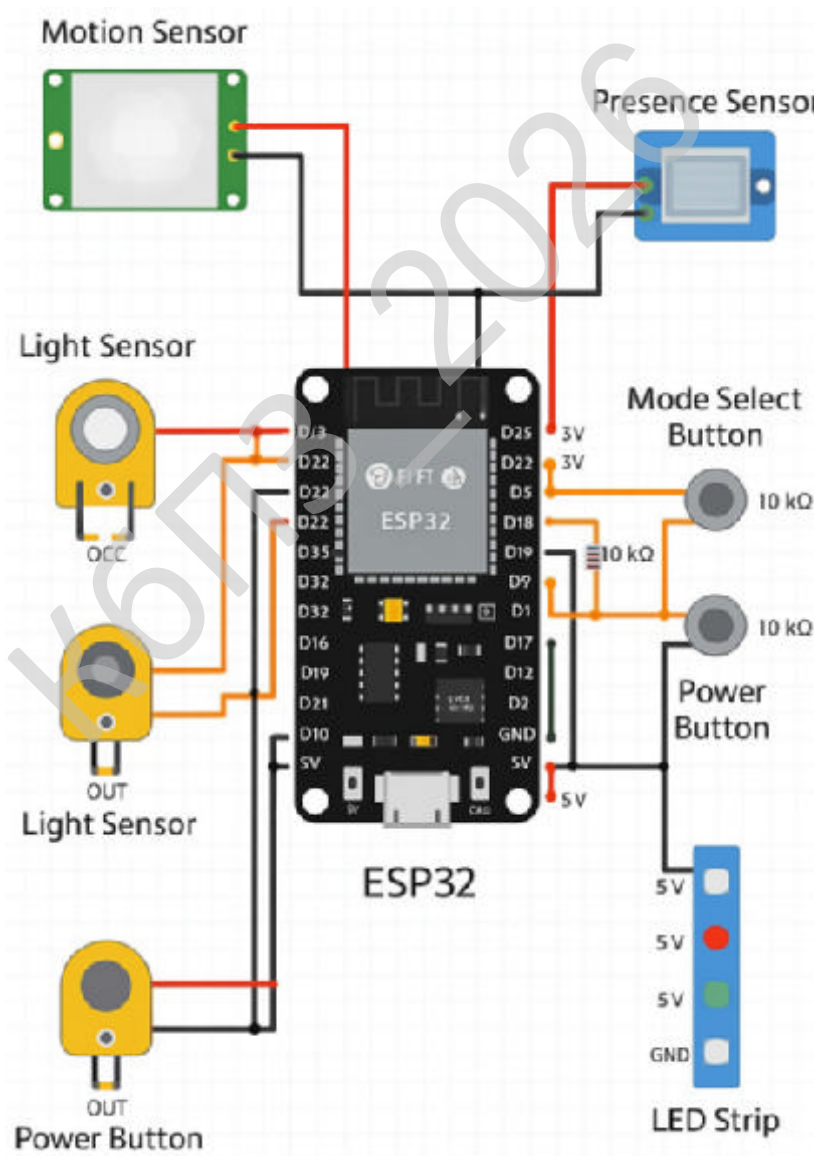


Рисунок 5.1 – Схема підключення компонентів системи підсвітки сходів

Мікроконтролер ESP32 виконує функцію центрального вузла керування, будучи з'єднаним із сенсорами та виконавчими елементами системи. Завдяки вбудованим модулям Wi-Fi і Bluetooth він забезпечує можливість дистанційного моніторингу та керування підсвіткою.

Сенсор руху використовується для фіксації появи людини на початку або в кінці сходового прольоту та підключається таким чином:

- VCC - до 3.3 V ESP32;
- GND - до GND;
- OUT - до одного з цифрових входів ESP32 (наприклад, GPIO14).

Сенсор присутності RCWL-0516 визначає наявність людини у зоні сходів протягом тривалішого часу. Його підключення до ESP32 здійснюється так:

- VCC - до 5 V ESP32;
- GND - до GND;
- OUT - до цифрового входу ESP32 (наприклад, GPIO27).

Сенсор освітленості BH1750 вимірює рівень зовнішнього освітлення, що дозволяє вмикати підсвітку лише за умов недостатнього світла. Схема підключення до ESP32:

- SDA - до GPIO21;
- SCL - до GPIO22;
- VCC і GND - відповідно до живлення модуля.

Кнопка увімкнення системи дає змогу вручну керувати її роботою (вмикання/вимикання). Вона підключається до одного з GPIO (наприклад, GPIO13) і до GND. Рекомендується використання резистора 10 кОм або внутрішнього підтягувального резистора (pull-up) ESP32.

Кнопка вибору режиму підсвітки забезпечує перемикання між різними режимами роботи (наприклад, статичне освітлення, анімація, реакція на рух) і підключається аналогічно до кнопки увімкнення.

Адресна світлодіодна стрічка WS2812B є основним виконавчим елементом підсвітки сходів. Схема її підключення така:

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

- DIN - до одного з цифрових виходів ESP32 (наприклад, GPIO18);
- VCC - до зовнішнього джерела живлення 5 V (відповідно до довжини стрічки);

GND - спільний провід (земля) із ESP32.

Рекомендується встановити резистор номіналом 330–470 Ом між виходом GPIO мікроконтролера та входом DIN світлодіодної стрічки, а також електролітичний конденсатор ємністю близько 1000 мкФ між лініями VCC і GND для стабілізації живлення.

Живлення основних компонентів системи може здійснюватися від USB або через вхід VIN (5 V) мікроконтролера ESP32. Світлодіодна стрічка повинна живитися від окремого джерела з достатнім струмовим запасом, відповідно до її довжини та споживаної потужності. Отже, спроектовано комп'ютерну систему підсвітки сходів із функцією дистанційного моніторингу та можливістю локального керування. У разі спрацювання одного із сенсорів, наприклад датчика руху, ESP32 аналізує рівень освітленості й за необхідності активує підсвітку. Кнопки дозволяють вручну вмикати освітлення або змінювати режим роботи. За наявності підключення до Wi-Fi мікроконтролер передає телеметричні дані до хмарного сервісу чи Telegram-бота, а також приймає команди дистанційного керування.

Налаштування і програмна реалізація системи підсвітки сходів з віддаленим моніторингом

Перед початком реалізації програмної логіки системи підсвітки сходів із функцією віддаленого моніторингу необхідно виконати початкове налаштування параметрів і створити відповідні сервіси у хмарній інфраструктурі та месенджері Telegram. Процедура створення Telegram-бота є відносно простою і детально описана в офіційній документації.

У результаті конфігурації хмарного середовища було розгорнуто API Gateway (рис. 5.2), HTTP API для взаємодії з Telegram-ботом (рис. 5.3) та WebSocket API для зв'язку з мікроконтролером (рис. 5.4).

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

	Name	Description	ID	Protocol	API endpoint type	Created
	botControlAPI		dbuvukg9t	HTTP	Regional	2024-11-04
	botControlAPIWebSoc		bye1jg889	WebSocket	Regional	2024-11-06

Рисунок 5.2 – Розгорнуті API Gateway

Routes for botControlAPI Search /botControlFunction POST AWS Lambda	Integration details for route POST /botControlFunction (fw490q) Lambda function botControlFunction (eu-central-1) Integration ID: 1kr59uh Description	Detach integration Manage integration
---	---	---

Рисунок 5.3 – HTTP API для зв’язку з телеграм ботом:

Create route Route selection expression info \$request.body.action \$connect \$disconnect sendMessage	WebSocket API для зв’язку з мікроконтролером \$connect ARN arn:aws:apigateway:eu-central-1:962382527220:bye1jg889:/\$connect Client Route request Integration request Lambda integration Enable two-way communication Delete
--	---

Рисунок 5.4 – WebSocket API для зв’язку з мікроконтролером:

Фрагмент програмного коду для мікроконтролера ESP32, що реалізує функціонування інтелектуальної системи керування підсвіткою сходів, наведено на рисунку 5.5.

```

#include <Adafruit_NeoPixel.h>
#include <SPIFFS.h>
#include <WiFiManager.h>
#include <WebSocketsClient.h>
#include <ArduinoJson.h>

//#define DEBUG

#ifdef DEBUG
    #define LOG(text) Serial.println(text);
#else
    #define LOG(text)
#endif

#define PIXEL_PIN 27

#define NUMPIXELS 60

#define BTN_WIFI 4
#define BTN_MODE 18
#define BTN_COLOR 19
#define BTN_POWER 21
#define BTN_R 33
#define BTN_G 32
#define BTN_B 35

#define websocketServer "bye1jg8i89.execute-api.eu-central-1.amazonaws.com"

#define getChipId() ((uint32_t)ESP.getEfuseMac())

#define saveDataFileName "/data.json"

SemaphoreHandle_t xMutexColor;
SemaphoreHandle_t xMutexStates;

```

Рисунок 5.5 – Фрагмент коду контролю ESP32

Під час реалізації функцій керування мікроконтролера ESP32 були використані такі основні компоненти та бібліотеки:

- **Adafruit_NeoPixel** - забезпечує керування світлодіодною стрічкою, сумісною з WS2812.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

- **SPIFFS** - використовується для збереження і зчитування параметрів кольору, режимів роботи та стану живлення.
- **WiFiManager** - забезпечує налаштування підключення до Wi-Fi через режим точки доступу.
- **WebSocketsClient** - відповідає за встановлення та підтримку з'єднання із сервером через WebSocket.
- **ArduinoJson** - використовується для обробки вхідних і вихідних команд у форматі JSON.
- **FreeRTOS** - дозволяє організувати багатозадачну роботу за допомогою механізму xTaskCreatePinnedToCore.

Програмна реалізація системи керування та моніторингу підсвітки передбачає оголошення кнопок керування:

- **BTN_WIFI** - активує режим точки доступу для налаштування Wi-Fi.
- **BTN_POWER** - вмикає або вимикає світлодіодну стрічку.
- **BTN_MODE** - перемикає режим роботи підсвітки.
- **BTN_COLOR** - активує режим зміни кольору.
- **BTN_R, BTN_G, BTN_B** - регулюють значення складових кольору R/G/B.

У програмі також визначені сценарії освітлення, які обираються залежно від значення змінної newState. Серед них:

- статичне освітлення;
- анімація «Pong»;
- ефект імітації завантаження (load);
- випадкове підсвічування;
- «шахівниця» (чергування двох кольорів);
- анімація з використанням двох кольорів.

Функції збереження і відновлення стану передбачають запис параметрів кольору, режиму, chatId і статусу повідомлень у файл /data.json. Логіка взаємодії з хмарою реалізована таким чином, що кожна команда, отримана через WebSocket,

аналізується та виконується — зокрема зміна кольору, увімкнення або вимкнення підсвітки, перемикання режимів - після чого відповідні повідомлення надсилаються назад на сервер через WebSocket.

Задачі FreeRTOS забезпечують обслуговування Wi-Fi та WebSocket-з'єднання, виконання світлових ефектів, обробку натискань кнопок і виклик відповідних функцій керування.

Використання механізмів синхронізації xSemaphoreTake та xSemaphoreGive забезпечує коректний доступ до спільних ресурсів, запобігаючи пошкодженню даних кольору та стану при одночасному виконанні кількох задач. До основних переваг такої реалізації належать: повноцінне локальне керування, можливість дистанційного керування через WebSocket, гнучка система світлових ефектів, збереження поточного стану після перезавантаження, а також підтримка зміни кольору за допомогою двох незалежних наборів RGB.

Повний програмний код керування мікроконтролером ESP32 наведено у додатку Б. На рисунку 5.6 представлено фрагмент коду, що відповідає за підключення мікроконтролера.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

```

import json
import boto3
from botocore.exceptions import ClientError

s3 = boto3.client('s3')
bucket_name = "esp-connection"

def lambda_handler(event, context):
    connection_id = event["requestContext"]["connectionId"]
    devices = []
    devices_states = {}
    default_device_state = {}

    query_params = event.get("queryStringParameters", {})
    device_id = query_params.get("device_id", "unknown_device")
    chat_id = query_params.get("chat_id")
    device_state = query_params.get("state")
    color_r = query_params.get("r", -1)
    color_g = query_params.get("g", -1)
    color_b = query_params.get("b", -1)

    print(f"Device id : {device_id}")
    print(f"Chat id : {chat_id}")
    print(f"State : {device_state}")
    print(f"Color : R={color_r}, G={color_g}, B={color_b}")

    if not chat_id:
        print("Chat id missing!")

        return {
            "statusCode" : 200
        }

    print("Pass chat check!")

```

Рисунок 5.6 – Фрагмент коду підключення ES

Код, наведений на рисунку 5.6, є реалізацією AWS Lambda-функції, яка обробляє HTTP-запити, пов'язані з WebSocket-з'єднанням пристроїв (наприклад, ESP32) із хмарним сервером через API Gateway. Основне призначення функції - керування підключеннями, збереження стану пристроїв і параметрів кольору у сховищі S3, а також обслуговування запитів від ESP-контролерів у режимі

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

реального часу.

Цей програмний модуль забезпечує збереження у S3 інформації про підключені пристрої ESP, зокрема їхні device_id, поточний стан (state) та значення кольору (color), ведення обліку WebSocket-з'єднань (connectionId) і підтримку одночасної роботи кількох підключень у межах одного чату.

Серед особливостей реалізації - використання WebSocket-з'єднання з отриманням connectionId для керування підключеннями, а також застосування сховища S3 у ролі простої NoSQL-структури для зберігання даних. Для підвищення надійності та безпеки обробки запитів у коді використано механізми обробки винятків (ClientError, Exception). Крім того, реалізовано динамічне оновлення списку пристроїв, що дозволяє додавати або змінювати параметри пристроїв у конфігурації відповідного chat_id. Ідентифікатори пристрою, кольору та стану світлодіодної стрічки передаються як параметри та зберігаються у сховищі.

Типовий вміст файлів у S3, що містять ці дані, наведено на рис. 5.7 та рис. 5.8.

```
{
  "chat_id": "123456",
  "devices": ["esp_001", "esp_002"],
  "devicesStates": {
    "esp_001": {
      "state": "3",
      "color": { "r": "255", "g": "0", "b": "0" }
    }
  }
}
```

Рисунок 5.7 – Вміст /chats/{chat_id}.json

```
{
  "connectionId": "abcd1234",
  "device_id": "esp_001"
}
```

Рисунок 5.8 – Вміст /connections/{device_id}.json

Для реалізації програмного вимкнення мікроконтролера використано окрему функцію, фрагмент якої наведено на рис. 5.9.

```
import json
import boto3

s3 = boto3.client('s3')
bucket_name = "esp-connection"

def lambda_handler(event, context):
    connection_id = event["requestContext"]["connectionId"]

    response = s3.list_objects_v2(Bucket = bucket_name, Prefix = "connections/")
    for obj in response.get("Contents", []):
        file_content = s3.get_object(Bucket = bucket_name, Key = obj["Key"])["Body"].read().decode('utf-8')
        data = json.loads(file_content)
        if data.get("connectionId") == connection_id:
            s3.delete_object(Bucket=bucket_name, Key=obj["Key"])
            break

    return {
        "statusCode": 200,
        "body": json.dumps({"message": "Disconnected"})
    }
```

Рисунок 5.9 – Функція відключення мікроконтролера

Фрагмент функції **botControlFunction**, наведений на рисунку 5.10, призначений для організації взаємодії з Telegram-ботом і передачі команд на мікроконтролер. Даний скрипт реалізує AWS Lambda-функцію, що забезпечує керування пристроями ESP32 через Telegram у межах системи «розумної» підсвітки сходів. Обмін командами здійснюється через WebSocket API із використанням хмарного сховища S3, сервісів AWS API Gateway і Telegram API.

У програмному коді реалізовано функцію, яка надає користувачеві

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

можливість надсилати через Telegram команди для зміни стану пристроїв.

Користувачеві доступні такі можливості керування системою:

- увімкнення та вимкнення підсвітки сходів;
- вибір режимів світлових ефектів;
- встановлення кольору у форматі RGB;
- перегляд переліку підключених пристроїв;
- видалення пристроїв із системи;
- отримання інформації про поточний стан усіх підключених пристроїв.

```
import json
import requests
import boto3
from botocore.exceptions import ClientError
import enum

class commandCodes(enum.Enum):
    OFF = 1
    ON = 2
    STATIC = 3
    PONG = 4
    BREATHING = 5
    FIRE = 6
    START = 100
    SHOW_DEVICES = 101
    REMOVE_DEVICE = 102
    GET_STATES = 103
    ERROR = -1

s3 = boto3.client('s3')
bucket_name = "esp_connection"

BOT_TOKEN = "7751291125:AAHtmRtTAPBGdNIomz4-g06aTkRgD3q5gCs"
url = f"https://api.telegram.org/bot{BOT_TOKEN}/sendMessage"

def command_handler(message) -> int:
    match message:
        case "/on":
            return commandCodes.ON.value
        case "/off":
            return commandCodes.OFF.value
        case "/static_light":
            return commandCodes.STATIC.value
        case "/pong_light":
            return commandCodes.PONG.value
        case "/breath_light":
```

Рисунок 5.10 - Фрагмент функції botControlFunction ()

Перелік команд, доступних у Telegram-боті для моніторингу та керування підсвіткою сходів, наведено у таблиці 5.1.

Таблиця 5.1 – Команди Telegram-бота

Команда	Опис
/ on <device_id>	Увімкнути пристрій
/ off <device_id>	Вимкнути пристрій
/ statistic_light <device_id> R G B	Увімкнути статичне світло вказаного кольору
/ pong_light <device_id>	Візуальний ефект “pong”
/ breath_light <device_id>	Дихаючий ефект світла
/ fire_light <device_id>	Імітація полум'я
/remove_device <device_id>	Видалити пристрій із чату
/ shov_devices	Список пристроїв у чаті
/ get_states	Стани пристроїв (on/off + RGB)
/ start	Вивід ID чату

Інтеграцію Telegram-бота реалізовано з використанням таких основних компонентів:

- BOT_TOKEN — застосовується для надсилання повідомлень через Telegram; API;
- send_feedback(chat_id, reply) — функція, що забезпечує відправлення відповіді користувачеві.

Для приймання повідомлень від мікроконтролера та передачі зворотного зв'язку користувачеві використовується модуль sendMessage, фрагмент програмного коду якого наведено на рис. 5.11.

```

import json
import urllib3
import boto3
import requests

BOT_TOKEN = "7751291125:AAHtmRtTAP8GdMIomz4-g06aTkRgD3q6gCs"

s3 = boto3.client('s3')
bucket_name = "esp-connection"

url = f'https://api.telegram.org/bot{BOT_TOKEN}/sendMessage'

def lambda_handler(event, context):
    print("Event received:", event)

    chat_id = 0
    device_id = ""
    message = ""

    reply = ""

    state = -1

    color = {"r": 0, "g": 0, "b": 0}

    connectionId = event["requestContext"]["connectionId"]

    body = {}

    if "body" in event:
        body_content = event["body"]
        print("Raw body content:", body_content)
        try:
            if isinstance(body_content, str) and body_content.strip():
                body = json.loads(body_content)
            else:

```

Рисунок 5.11 – Фрагмент модуля sendMessage

Команди в Telegram-боті можуть вводитися вручну, наприклад: /static_light 1111111 180 32 76. У цьому прикладі:

- /static_light - команда активації режиму статичного освітлення;
- 1111111 - ідентифікатор пристрою (device_id);

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

- 180 - значення інтенсивності для червоного каналу (R);
- 32 - значення інтенсивності для зеленого каналу (G);
- 76 - значення інтенсивності для синього каналу (B).

Команди `/static_light`, `/pong_light`, `/load_light`, `/random_light`, `/checkers_light` та `/two_colors` обов'язково потребують зазначення ідентифікатора пристрою, який можна отримати за допомогою команди `/show_devices`.

Параметри кольору є необов'язковими, тому команду можна подати у спрощеному вигляді, наприклад: `/static_light 1111111`. Водночас доступна передача значень для двох палітр кольорів, наприклад:

`/static_light 1111111 45 65 23 98 100 32`,

де 45 65 23 — значення RGB для першої палітри, а 98 100 32 — для другої.

Якщо під час виконання команди з'являється повідомлення «*Waiting for response from <device id>*», це означає, що пристрій втратив з'єднання з мережею або не має живлення. Повідомлення «*Device is not connected*» свідчить про відсутність підключення пристрою до мережі або його вимкнений стан. Команда `/get_states` дозволяє переглянути поточний режим роботи та колір світлодіодів.

Отже, було розроблено й налаштовано програмне забезпечення для керування та моніторингу підсвітки сходів із використанням IoT-пристроїв, хмарної інфраструктури та месенджера Telegram.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

6 ОСНОВНІ ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було розроблено та реалізовано комп'ютерну систему автоматизованого керування підсвіткою сходів із підтримкою дистанційного моніторингу та налаштування.

Ключовим апаратним компонентом системи є мікроконтролер ESP32, який забезпечує обробку даних від сенсорів руху, присутності та освітленості, а також керування адресною світлодіодною стрічкою. Завдяки інтеграції з хмарною платформою Amazon Web Services (AWS) і використанню WebSocket-з'єднання реалізовано можливість віддаленого керування освітленням через Telegram-бота та синхронізацію станів між пристроєм і сервером.

У межах роботи створено багатозадачне програмне забезпечення, що включає обробку локального керування за допомогою кнопок, збереження конфігурацій у файловій системі SPIFFS, обробку команд із хмари, зміну кольору підсвітки та перемикання між різними режимами відображення (статичне освітлення, ефекти «pong», «random», «checkers», «two colors» тощо).

Отримані результати підтверджують, що запропонована система є гнучким і масштабованим рішенням для освітлення житлових приміщень, яке може бути адаптоване до індивідуальних потреб користувача. Вона підвищує рівень комфорту та безпеки, а також сприяє зменшенню споживання електроенергії завдяки автоматичному вмиканню освітлення лише за необхідності.

Таким чином, поставлену мету досягнуто: створено функціональний, економічний та ефективний прототип комп'ютерної системи підсвітки сходів із можливістю віддаленого моніторингу, який має потенціал для подальшого розвитку та впровадження у практичне використання.

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Koval V., Adamiv O., Proc. of the Third IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2005). – Sofia (Bulgaria). – 2005. – P. 120- 124.
2. Yasin, J.N., Mohamed, S.A.S., Haghbayan, MH. et al. Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots. Int. j. inf. tecnol. 13, 97–107 (2021). <https://doi.org/10.1007/s41870-020-00513-w>.
3. Sabry F. Self Driving Car: Solving Full Self-driving Need Solving Real-world artificial Intelligence. One Billion Knowledgeable. 2022 . 293 p.
4. Shapiro D. G. Three Anecdotes from the DARPA Autonomous Land Vehicle Project. AI Magazine, 2008. 29(2), 40. <https://doi.org/10.1609/aimag.v29i2.2108>
5. Young R. Critical Analysis of Prototype Autonomous Vehicle Crash Rates Six Scientific Studies from 2015-2018 SAE International. 2021. 252 p.
6. Рудик А. В. Наукові основи та принципи побудови приладової системи вимірювання прискорення мобільного робота : дис. ... д-ра техн. наук : 05.11.01– Прилади та методи вимірювання механічних величин. Київ, 2018. – 460 с.
7. Поліщук М. М., Ткач М.М. Робототехнічні системи: проектування і моделювання: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2021. 112 с.
8. Мигаль В. Д. Інтелектуальні системи в технічній експлуатації автомобілів: монографія. Х.: Майдан, 2018. 262 с.
9. Staron M. Automotive Software Architectures: An Introduction. Second Edition Springer International Publishing A&G. 2022. 274 p.
10. Vargas J. et al. An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions. Sensors. 2021; 21(16):5397.
11. Wevolver 2020 Autonomous Vehicle Technology Report. URL:<https://www.coursehero.com/file/62601304/Wevolver2020AutonomousVehicleTechnologyReportpdf/> (accessed 2023 June 1).

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

12. React.js [Електронний ресурс] – Режим доступу до ресурсу:
<https://uk.reactjs.org/>
13. AngularJS MVC [Електронний ресурс] – Режим доступу до ресурсу:
https://www.tutorialspoint.com/angularjs/angularjs_mvc_architecture.htm
14. Vue.js [Електронний ресурс] – Режим доступу до ресурсу:
<https://vuejs.org/>
15. A Look Inside ADAS Modules. URL: <https://amkor.com/semiconductor-story/alook>.
16. AV and ADAS Sensors. URL [https://community.sw.siemens.com/s/article/AVand- ADAS-Sensors](https://community.sw.siemens.com/s/article/AVand-ADAS-Sensors) (accessed 2023 June 1).
17. Гуржій А. М. Основи автоматики та робототехніки: Навчальний посібник/ А.М. Гуржій, А. Т. Нельга, В. М. Співак, О. С. Ітякін:—Дніпро:«Гарант СВ», 2021.- 243с.
18. Lim, B.S. et al. Autonomous Vehicle Ultrasonic Sensor Vulnerability and Impact Assessment. In Proceedings of the IEEE World Forum on Internet of Things (WFIoT), Singapore, 5–8 February 2018; pp. 231–236.
19. Саліхов М. М. Arduino – перспективний інструмент протитопування у робототехніці // Концептуальні шляхи розвитку науки та освіти: матеріали VIII Міжнародної науково-практичної конференції м. Львів, 9-10 червня 2023 року.— Львів: Львівський науковий форум, 2023. – с. 73-79.
20. Bräunl T. Embedded Robotics: From Mobile Robots to Autonomous Vehicles with RaspberryPi and Arduino Springer Nature, 2022. 519 p.
21. Навігаційні системи [Електронний ресурс] : навч. посіб. для студ. Спеціальності 151 «Автоматизація та комп’ютерно-інтегровані технології» / С.Л. Лакоза; КПІ ім. Ігоря Сікорського, 2021. — 80 с.
22. Imad M. et al. Navigation system for autonomous vehicle: A survey. JCSTS., vol. 2, no. 2, pp. 20–35, 2020.

23. Ковалець І. В. та ін. Технологія планування траєкторій руху мобільних об'єктів з урахуванням перешкод на складній місцевості. Енергетика і автоматика. - 2017. - № 1. - С. 110-122.

24. HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/uk/docs/Web/HTML>

25. Aguilo I. et al. Artificial Intelligence Research and Development Hardcover by IOS Press. 2003. 500 p.

26. Гребенюк Б. А. Розробка підсистеми управління інтелектуальним роботом / Б. А. Гребенюк // «Automation and Development of Electronic Devices» ADED-2023: Collection of Students' Scientific Paper. – Kharkiv : Kind of Kharkiv National University of Radio Electronics [electronic edition], 2023. – Part 1. –336p. P. 263-269.

27. MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/MongoDB>.

28. Becker M. et al Obstacle avoidance procedure for mobile robots. ABCM Symposium series in Mechatronics. Vol. 2, 2006. P. 250-257.

29. CORS, XSS [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.to/maleta/cors-xss-and-csrf-with-examples-in-10-minutes-35k3>.

30. AJAX [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/AJAX>.

31. Fetch API [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/ru/docs/Web/API/Fetch_API.

32. Звенігородський О.С. Інтелектуальна система планування тактики руху автономного робота в квазістаціонарному середовищі: Дис... канд. техн. наук: 05.13.23. — Д., 2002. — 127 с.

33. Адамів, О.П. Моделі та інтелектуальні засоби адаптивного керування автономним мобільним роботом. Дис. канд.техн. наук. Одеса, 2007. 124 с.

34. Koval V., Adamiv O., Proc. of the Third IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2005). – Sofia (Bulgaria). – 2005. – P. 120- 124.

35. npm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.npmjs.com/>

36. Install MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.mongodb.com/manual/administration/install-on-linux/>

37. Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://www.digitalocean.com/community/tutorials/node-js-ubuntu-18-04-ua>.

38. Linux [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Linux>.

39. npm-audit [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.npmjs.com/cli/audit>.

40. Kojima T. et al. Intelligent Technology for More Advanced Autonomous Driving. 2018. Hitachi Review Vol. 67, No. 1. p. 58–63.

41. SQL [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/SQL>.

42. MongoDB Compass [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/products/compass>.

43. Postman [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postman.com/>

44. Саліхов М. М. Від автоматизованих до автономних транспортних засобів // Наукові досягнення та інновації: шлях до успіху. X Всеукраїнська мультидисциплінарна науково-практична Інтернет-конференція, 31 травня 2023, Україна, Київ : зб. матеріалів — Електрон. дан. — Київ : Яроченко Я. В., 2023. — с. 136-142.

45. SPA (Single-page application) [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Glossary/SPA>

46. Саліхов М.М. Самокеровані автомобілі та системи їх навігації// Практичні та теоретичні питання розвитку науки та освіти: матеріали VIII Міжнародної науково-практичної конференції: м. Львів, 19-20 червня 2023 року. – Львів: Львівський науковий форум, 2023. – С.53-59.

47. SPA (Single-page application) [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Glossary/SPA>.

48. Vargas J. et al. An Overview of Autonomous Vehicles Sensors and Their

49. Vulnerability to Weather Conditions. Sensors. 2021; 21(16):5397.

50. <https://www.coursehero.com/file/62601304/Wevolver2020AutonomousVehiclesTechnologyReportpdf/> (accessed 2024 June 1).

К6ПЗ_2026

					ВКРБ-123.26.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75