

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти перший (бакалаврський)

Система керування режимами світлодіодної стрічки через мобільний  
додаток  
(тема)

Виконав: здобувач 4 року навчання,  
групи КІУКІ-21-8

Протасенко А.С  
(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Комп'ютерна інженерія  
(повна назва освітньої програми)

Керівник проф. Немченко В.П.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_ Чумаченко С.В.  
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія  
(шифр і назва)

Тип програми Освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерна інженерія  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Протасенко Артему Сергійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Система керування режимами світлодіодної стрічки через мобільний додаток \_\_\_\_\_

затверджена наказом по університету від "21" \_\_\_\_\_ 05 \_\_\_\_\_ 2025 р. № 403 Ст. \_\_\_\_\_

2. Термін подання студентом роботи до екзаменаційної комісії 12.06.2025

3. Вихідні дані до роботи \_\_\_\_\_

Мікроконтролер ESP32-D0WDQ6

Плата ESP32-DevKitC V4

Мова програмування C++

Середовище розробки Arduino IDE 2.3.5

Датчик освітленості: Фотоелектричний датчик (PhotoresistorModule)

Датчик руху (PIR): HC-SR501

RGB-стрічка: 5050 RGB LED Strip (12V, 4-pin)

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

Системи керування режимами світлодіодної стрічки

Технологічна платформа реалізації пристрою керування

Апаратне проектування системи керування освітленням


Середовище розробки програмного забезпечення

Програмування мікроконтролерного пристрою

Реалізація зв'язку між мікроконтролером і мобільним додатком

. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій  
(п.5 включається до завдання за рішенням випускової кафедри) 15 слайдів \_\_\_\_\_


6. Консультанти розділів роботи (проекту)


Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
	Проф. Немченко В.П.		15.06.25

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання етапів проекту (роботи)	Примітка
1	Видача теми проекту, узгодження і затвердження теми	5.05.2025 - 6.05.2025	Виконано
2	Аналіз предметної області, постановка задачі	7.5.2025 - 15.5.2025	Виконано
3	Обґрунтування вибору програмних засобів для розробки системи керування світлодіодною стрічкою	16.5.2025 - 19.5.2025	
4	Розробка концептуальної схеми моделі пристрою керування	20.05.2025 - 22.5.2025	Виконано
5	Проектування та розробка електричної схеми моделі пристрою керування світлодіодною	23.05.2025 - 24.5.2025	Виконано
6	Обґрунтування вибору апаратних засобів для розробки системи керування світлодіодною	25.5.2025 - 27.05.2025	Виконано
7	Розробка та тестування системи керування світлодіодною стрічкою	28.05.2025 - 01.06.2025	Виконано
8	Оформлення пояснювальної записки	02.06.2025 - 08.05.2025	Виконано
9	Перевірка виконаного проекту керівником	09.06.2025 - 09.06.2025	Виконано
10	Захист проекту	12.06.2025 - 22.06.2025	Виконано

Дата видачі завдання 05 05 2025\_р.

Студент   
(підпис)

Керівник роботи (проекту) проф. Немчен   
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка складається з: 60 сторінки, 28 рисунків, 20 джерел за переліком посилань.

### МІКРОКОНТРОЛЕР, ARDUINO, СИСТЕМА КЕРУВАННЯ РЕЖИМАМИ ОСВІТЛЕННЯ, ОСВІТЛЕННЯ, МОБІЛЬНИЙ ДОДАТОК.

Метою роботи є розробка системи керування режимами світлодіодної стрічки через мобільний додаток. Система дозволяє автоматично змінювати освітлення за заданими сценаріями, що враховують час доби та дані з датчиків руху і освітленості. Користувач також може змінювати колір і яскравість підсвітки вручну через мобільний додаток.

Розроблено електричну схему системи, програму для мікроконтролера ESP32 і мобільний додаток на Flutter. Зв'язок між додатком та пристроєм здійснюється через Bluetooth LowEnergy (BLE).

Апаратна частина містить , RGB-стрічку 5050 (12V), датчик руху HC-SR501 та фоторезисторний модуль MH-Sensor-Series. Програмна частина написана мовами Dart (Flutter) та C++ (Arduino IDE).

## ABSTRACT

The explanatory note consists of:60 pages,28 figures, 20 sources according to the list of references.

MICROCONTROLLER, ARDUINO, AUTOMATIC LIGHTING MODES CONTROL SYSTEM, LIGHTING, MOBILE APPLICATION.

The purpose of the work is to develop a system for controlling RGB LED strip modes via a mobile application. The system allows for automatic lighting adjustments based on predefined scenarios, which take into account the time of day and data from motion and light sensors. The user can also manually control the color and brightness of the lighting through the mobile application.

The electrical schematic of the system, firmware for the ESP32 microcontroller, and a mobile application using Flutter were developed. Communication between the application and the device is established via Bluetooth Low Energy (BLE).

The hardware part includes the ESP32-DevKitC V4 development board, a 5050 RGB LED strip (12V), a HC-SR501 motion sensor, andan MH-Sensor-Series light sensor module. The software part is implemented in Dart (Flutter) and C++ (Arduino IDE).

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП.....	9
1 СИСТЕМА КЕРУВАННЯ РЕЖИМАМИ СВІТЛОДІОДНОЇ СТРІЧКИ ЧЕРЕЗ МОБІЛЬНИЙ ДОДАТОК .....	10
1.1 Система освітлення та її параметри .....	10
1.2 Методи контролю режимів освітлення .....	17
1.2.1 Датчик освітлення приміщення .....	18
1.2.2 Датчик руху приміщення .....	20
1.2.3 Світлодіодна стрічка RGB (5050) .....	21
1.3 Процесорний блок .....	23
1.3.1 Вибір мікропроцесора .....	23
1.3.2 Платформа ESP32 .....	25
1.3.3 Апаратна частина платформи ESP32 .....	25
1.3.4 Версії платформи ESP32 .....	27
1.3.5 Огляд мікроконтролера ESP32-WROOM-32 .....	29
1.3.6 Середовище програмування – ArduinoIDE .....	33
2 РОЗРОБКА СИСТЕМИ КЕРУВАННЯ РЕЖИМАМИ СВІТЛОДІОДНОЇ СТРІЧКИ ЧЕРЕЗ МОБІЛЬНИЙ ДОДАТОК .....	36
2.1 Розробка функціональної схеми пристрою .....	36
2.1.1 Транзисторний драйвер ULN2003A .....	38
2.2 Розробка електричної схеми системи керування .....	39
2.3 Програмний код та його оптимізація .....	40
3 РОЗРОБКА ТА ФУНКЦІОНУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ КЕРУВАННЯ СВІТЛОДІОДНОЮ СТРІЧКОЮ .....	45
3.1 Призначення та можливості мобільного додатку .....	45
3.2 Структура та інтерфейс користувача .....	45
3.2.1 Головний екран .....	46
3.2.2 Екран Modes .....	48
3.2.3 Екран SensorSettings .....	50

3.2.4 Екран Timer.....	51
3.2.5 Екран керування Bluetooth-з'єднанням .....	52
3.3 Технічна реалізація та використані програмні засоби .....	54
3.4 Алгоритм взаємодії додатку з апаратною частиною системи .....	56
ВИСНОВКИ .....	57
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	58
ДОДАТОК А.....	61
ДОДАТОК Б.....	69
ДОДАТОК В.....	78

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ESP32 – одноплатний мікроконтролер із Wi-Fi та Bluetooth;  
BLE – Bluetooth Low Energy (низькоенергетичний Bluetooth);  
RGB – Red Green Blue (кольорова модель світла);  
PIR – пасивний інфрачервоний датчик руху;  
ULN2003A – транзисторний драйвер;  
MH-Sensor-Series – фотоперетворювальні датчики освітленості;  
Dart – мова програмування;  
Flutter – кросплатформенний фреймворк;  
RSSI – індекс рівня прийнятого сигналу;  
MAC – унікальна апаратна мережна адреса;  
SMD – компоненти поверхневого монтажу;  
ADC – аналогово-цифровий перетворювач;  
DAC – цифро-аналоговий перетворювач;  
Wi-Fi – бездротова мережа;  
VIN – вхідна напруга;  
USB – універсальна послідовна шина;  
GPIO – універсальні входи/виходи;  
QFN – тип корпусу мікросхеми;  
MHz – мегагерц (одиниця частоти);  
kB – кілобайт (одиниця пам'яті);  
MB – мегабайт (одиниця пам'яті);  
LED – світлодіод;  
ESP-IDF – офіційне середовище розробки ESP32.

## ВСТУП

Актуальність кваліфікаційної роботи полягає у важливості освітлення як невід'ємної складової життя людини. З давніх часів використовували різні засоби освітлення — від відкритого вогню до сучасних електричних ламп і складних систем управління. Незмінною залишається мета забезпечення комфорту, безпеки та раціонального використання енергії, особливо в умовах зростання тарифів і обмежених ресурсів.

В останні роки в Україні зростає популярність автоматизованих систем керування освітленням у побуті та громадських місцях. Це пов'язано не лише з прагненням підвищити комфорт, а й зі споживчою потребою знизити витрати на електроенергію та оптимізувати її використання.

Сучасні технології дозволяють створювати індивідуальні сценарії освітлення, враховуючи час доби та присутність людей у приміщенні. Замість складних панелей та кнопок достатньо смартфона для керування кольором, режимом або увімкненням освітлення.

Такі системи сприяють комфортному середовищу для роботи та відпочинку, а також економії енергії за рахунок автоматичного регулювання підсвітки відповідно до потреб.

Впровадження дистанційного керування режимами освітлення через мобільний додаток відкриває нові можливості, дозволяючи налаштовувати графік роботи, вибирати кольори та активувати автоматичні сценарії або керувати вручну.

Отже, розробка сучасної системи освітлення з простим управлінням і гнучкими налаштуваннями є актуальним напрямом, що підвищує комфорт, зменшує енергоспоживання та робить керування доступним для користувача.

# 1 СИСТЕМА КЕРУВАННЯ РЕЖИМАМИ СВІТЛОДІОДНОЇ СТРІЧКИ ЧЕРЕЗ МОБІЛЬНИЙ ДОДАТОК

## 1.1 Система освітлення та її параметри

Перед розглядом технічної реалізації системи керування режимами світлодіодної стрічки через мобільний додаток важливо визначити основні поняття та етапи розвитку технологій освітлення.

Кероване освітлення — це система, що забезпечує можливість регулювання параметрів світлового потоку: яскравості, кольору, температури світла, а також створення індивідуальних режимів роботи залежно від часу доби, умов навколишнього середовища або визначених сценаріїв. Основне призначення таких систем полягає у створенні адаптивного, енергоефективного та комфортного середовища в приміщенні.

Активний розвиток керованого освітлення почався наприкінці 1990-х — на початку 2000-х років, коли на ринку з'явилися перші модульні рішення: димери, багатоканальні контролери, релейні блоки. Проте ці системи потребували окремих пультів або дротових інтерфейсів, що ускладнювало їх використання та обмежувало можливості інтеграції в побуті.

Справжній прорив став можливим завдяки поширенню бездротових технологій, зокрема Bluetooth, Wi-Fi та розвитку мобільних додатків. У 2012 році компанія Philips представила на ринку першу комерційну систему керування освітленням — PhilipsHue, яка дозволяла через мобільний додаток змінювати яскравість, колір світла та створювати сценарії роботи освітлювальних приладів. Такий підхід відкрив нові можливості для автоматизації освітлення у побутових умовах, оскільки керування стало простим, доступним і зрозумілим навіть без спеціальних знань.

Сучасні системи освітлення поєднують апаратні компоненти (контролери, датчики, виконавчі пристрої) та програмне забезпечення для

гнучкого налаштування параметрів освітлення. Наприклад, у системах освітлення передбачено сценарії на кшталт «нічний режим»: у визначений часовий проміжок активується тепле м'яке світло, а датчик руху контролює присутність людини у приміщенні. Якщо протягом заданого часу рух не виявляється, підсвітка автоматично вимикається, що дозволяє економити електроенергію. У разі появи руху система вмикає освітлення на необхідний рівень, підтримуючи комфорт у приміщенні.

Системи керованого освітлення сучасного покоління відрізняються простотою інтеграції, можливістю налаштування за допомогою мобільного додатку, а також високим рівнем енергоефективності. Регулювання освітлення здійснюється автоматично або вручну через інтерфейс додатку, що забезпечує повну відповідність вимогам користувача та умовам експлуатації.



Рисунок 1.1 – Загальне представлення концепції «Кероване освітлення»

Розглянемо основні функціональні компоненти системи керування режимами світлодіодної стрічки через мобільний додаток.

1. Побудова системи має забезпечувати стабільну та безперервну роботу всіх елементів освітлення, включаючи регулювання кольору, яскравості та режимів підсвітки, а також взаємодію з датчиками руху й освітленості для реалізації адаптивних сценаріїв.

2. Автоматизація керування освітленням здійснюється без постійного втручання користувача: система налаштовується один раз і далі працює автономно, визначаючи оптимальні умови для обраних сценаріїв.

3. Передбачено можливість оперативного вмикання та вимикання режимів підсвітки через мобільний додаток, що дозволяє змінювати параметри освітлення у будь-який момент.

4. Наявність бездротового каналу зв'язку (BLE) забезпечує надійний обмін даними між додатком та контролером, що керує роботою підсвітки.

5. Функціонал системи передбачає збереження попередніх налаштувань режимів та їх автоматичне відновлення після перезавантаження або тимчасової втрати живлення.

Система керування режимами освітлення включає такі об'єкти автоматизації:

- світлодіодна стрічка з можливістю зміни кольору та яскравості;
- датчик руху для активації освітлення за потреби;
- датчик освітленості для аналізу умов навколишнього середовища;
- мобільний додаток для дистанційного керування;
- контролер на базі Arduino для обробки команд і управління підсвіткою.

Реалізація такої системи дозволяє забезпечити автоматичне керування освітленням відповідно до встановлених умов: часу доби, активності у приміщенні або необхідного рівня яскравості. Система підтримує плавне регулювання яскравості залежно від обраного сценарію чи зовнішніх чинників. Наприклад, у вечірній час підсвітка переходить у режим теплового приглушеного освітлення, що створює затишну атмосферу, тоді як вдень активується яскраве біле світло, яке підходить для роботи та повсякденних справ.

Ключовим елементом системи є датчик освітленості, який аналізує рівень природного освітлення та передає дані до контролера. На основі отриманої інформації система визначає, коли необхідно увімкнути або

вимкнути підсвітку, а також регулює її яскравість у режимі реального часу для досягнення оптимального комфорту та економії електроенергії. Наприклад, при настанні сутінок система активує підсвітку, а при достатньому природному світлі — вимикає її, запобігаючи надмірному споживанню енергії. Ця функціональність реалізується завдяки використанню датчика освітленості, приклад якого зображено на рисунку 1.2.



Рисунок 1.2 – Датчик освітленості Aqara Light Sensor T1

Система постійно відстежує наявність руху в приміщенні, отримуючи дані від датчика активності, і автоматично вмикає або вимикає підсвітку залежно від поточної ситуації. Це забезпечує комфортне використання системи та сприяє економії електроенергії, запобігаючи її зайвому витрачанням.

Передбачено декілька режимів роботи освітлення: нічний, денний та універсальний. Перемикання між ними відбувається відповідно до заданого розкладу або на основі сигналів від датчика. Наприклад, у нічний час підсвітка вмикається лише за наявності руху, а вдень вона працює в активному режимі протягом заданого інтервалу часу.

Достатньо лише одного налаштування параметрів освітлення для кожного з режимів, після чого система автоматично виконує свої функції згідно з обраними сценаріями. При тривалій відсутності активності підсвітка вимикається для зменшення енергоспоживання та збереження ресурсу

обладнання. Для виконання цих функцій у системі застосовується датчик руху, приклад якого наведено на рисунку 1.3.



Рисунок 1.3 – Контролер для управління руху Philips Hue Motion Sensor

Фіксування подій у приміщенні здійснюється системою на основі даних, що надходять від датчиків руху та освітленості. Завдяки цьому система може автоматично вмикати або вимикати підсвітку залежно від поточного стану: часу доби, рівня природного освітлення чи наявності активності в кімнаті. Такий підхід сприяє створенню комфортного середовища та дозволяє зменшити споживання електроенергії. Наприклад, у вечірній час вмикається тепле м'яке світло, що створює затишну атмосферу, тоді як удень підсвітка переходить у режим яскравого освітлення для виконання повсякденних завдань.

Конструкція системи керування складається з наступних компонентів:

- контролер на основі Arduino або ESP32, який обробляє сигнали від датчиків і керує підсвіткою;
- датчики освітленості та руху, що аналізують умови в приміщенні;
- світлодіодна стрічка з можливістю регулювання кольору та яскравості;
- мобільний додаток для налаштування параметрів освітлення та вибору сценаріїв роботи;

- бездротовий інтерфейс (BLE) для обміну даними між контролером і додатком.

Загальний алгоритм роботи системи включає кілька послідовних етапів, які забезпечують чітку взаємодію всіх елементів та виконання завдань у реальному часі.

- зчитування даних із датчиків освітленості та руху;
- передача інформації на контролер за допомогою бездротового інтерфейсу;
- обробка отриманих даних контролером та формування команд для керування підсвіткою;
- виконання команд: зміна кольору, яскравості або активація/деактивація підсвітки відповідно до встановлених сценаріїв;
- відображення поточного стану системи та можливість коригування параметрів через мобільний додаток.

Основна функція контролера полягає в аналізі отриманих даних, обробці вхідних сигналів та передачі команд для зміни параметрів підсвітки. Контролер забезпечує стабільну роботу системи, зберігає налаштування режимів, плавно регулює яскравість освітлення та своєчасно реагує на зміну умов у приміщенні.

Алгоритми, що закладені в систему, дають змогу підтримувати необхідний рівень освітлення, автоматично адаптуючи параметри світла до вибраного сценарію, умов у приміщенні та часу доби. Це забезпечує комфорт, енергоефективність і зменшення необхідності ручного керування підсвіткою.

Серед основних параметрів системи освітлення можна виділити наступні:

- інтенсивність світлового потоку (яскравість);
- кольорова температура підсвітки;
- рівень освітленості в приміщенні;
- наявність або відсутність активності (руху).

Сприйняття освітлення формується сукупністю параметрів підсвітки та характеристиками навколишнього середовища. Низький або надмірно високий рівень яскравості може спричиняти втому очей, знижувати концентрацію уваги, впливати на продуктивність виконання завдань і загальний стан здоров'я. Оптимальне поєднання яскравості та кольору світла дозволяє створювати сприятливі умови для виконання різних видів діяльності та водночас сприяє економному споживанню електроенергії.

Для підтримання належного рівня комфорту система автоматично регулює яскравість і колірну температуру підсвітки залежно від часу доби та обраного режиму роботи. У вечірній час активується тепле приглушене освітлення, що сприяє розслабленню, а в денний період вмикається яскраве біле світло для забезпечення продуктивності. Система враховує рівень активності у приміщенні: якщо протягом певного часу датчик руху не фіксує активність, подається команда на вимкнення підсвітки для зниження енергоспоживання.

Керування режимами освітлення здійснюється через мобільний додаток, який забезпечує зміну яскравості, кольору, активацію або деактивацію датчиків, а також вибір доступних сценаріїв та контроль поточного стану системи. Передбачена можливість попередньої підготовки приміщення за заданими параметрами: наприклад, активація приглушеного освітлення у вечірній час або увімкнення яскравої підсвітки у робочій зоні. Для оптимізації енергоспоживання система передбачає режими, які обмежують роботу підсвітки або зменшують яскравість у разі відсутності активності в приміщенні.

Функціонування системи забезпечується комплексом датчиків (руху та освітленості) та контролером, який обробляє отримані дані та формує команди на зміну параметрів освітлення. Завдяки цьому досягається адаптація підсвітки до змін у навколишньому середовищі та автоматичне підтримання оптимального режиму без потреби в ручному втручанні.

Висновок: застосування системи керування освітленням дозволяє створити комфортні умови в приміщенні, зменшити споживання електроенергії та забезпечити автоматизовану роботу підсвітки відповідно до заданих параметрів і поточних умов

## 1.2 Методи контролю режимів освітлення

Контроль параметрів освітлення здійснюється за допомогою спеціалізованих датчиків, що забезпечують визначення рівня освітленості в приміщенні та наявності активності (руху). Ці датчики можна поділити на цифрові та аналогові залежно від принципу роботи. Цифрові датчики передають дані у вигляді готових значень, наприклад, рівень освітленості або стан активності, що дозволяє швидко та без додаткової обробки використовувати їх у системі. Аналогові датчики формують сигнал у вигляді змінної напруги, який потребує перетворення у цифровий формат для подальшої обробки, наприклад, у діапазоні 0...1023. У рамках реалізації системи керування освітленням обрано цифрові датчики, оскільки вони дозволяють спростити підключення та забезпечують більш точне зчитування параметрів.

Центральний контролер, побудований на базі Arduino або ESP32, виконує обробку отриманих даних, аналізує їх та формує команди для зміни стану підсвітки. Завдяки цьому система автоматично реагує на зміни зовнішніх умов, коригуючи яскравість або колір підсвітки відповідно до часу доби, сценарію чи сигналів від датчиків. Наприклад, у разі недостатньої природної освітленості подається команда на активацію підсвітки, а в разі виявлення руху в приміщенні – увімкнення або зміна режиму роботи.

Сигнали для управління освітленням формуються контролером через спеціальні виходи, що підключаються до драйверів або актуаторів (наприклад, драйверів світлодіодних стрічок). Для плавного регулювання яскравості застосовується широтно-імпульсна модуляція (PWM). У разі керування

потужними джерелами освітлення передбачено використання транзисторів або електромагнітних реле для забезпечення необхідного рівня потужності та захисту системи від перевантажень.

Для належної роботи системи визначено допустимі діапазони параметрів освітлення:

- інтенсивність світлового потоку (яскравість): 10–100%;
- колірна температура: від 2700К (тепле світло) до 6500К (холодне світло);
- рівень освітленості в приміщенні: 0–500 лк;
- затримка перед автоматичним вимкненням підсвітки у разі відсутності активності: 5–30 хвилин (налаштовується у додатку).

Під час розробки системи важливо враховувати сумісність датчиків із контролером, а також потужність підключених освітлювальних елементів для забезпечення надійної роботи підсвітки у всіх режимах.

Детальний огляд датчиків і виконавчих компонентів, що використовуються у системі, а також їхній функціонал, буде розглянуто у наступних підрозділах.

### 1.2.1 Датчик освітлення приміщення

Датчик освітленості (або сутінковий датчик) – це пристрій, призначений для автоматичного керування джерелами світла залежно від рівня освітленості в приміщенні. Основна функція такого датчика полягає в тому, щоб фіксувати кількість світла в навколишньому середовищі та, у разі потреби, подавати сигнал на включення або вимкнення освітлення. Це дозволяє оптимізувати споживання електроенергії, запобігаючи непотрібній роботі світлових приладів у яскравий час доби.

Принцип роботи датчика базується на використанні фоторезистора, який змінює свій електричний опір залежно від інтенсивності світла. При підвищенні рівня освітленості опір фоторезистора зменшується, а при

зниженні – збільшується. У складі модуля також є компаратор, який обробляє сигнал від фоторезистора і формує вихідний сигнал для подальшої передачі на контролер.

Коли освітленість у приміщенні знижується до встановленого порогу, ланцюг замикається, активуючи освітлення. Якщо ж рівень світла достатній, наприклад, у денний час або при штучному освітленні, ланцюг розмикається, і підсвітка вимикається. На рисунку 1.8 зображено датчик освітленості, який буде застосовано в даній системі керування освітленням.

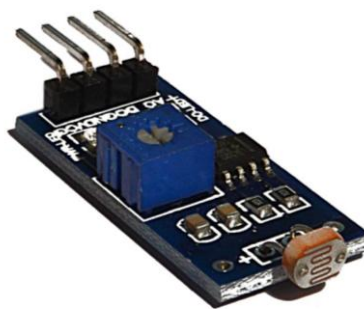


Рисунок 1.4 – Модуль датчика освітленості з фоторезистором (MH-Sensor-Series)

Цифровий датчик освітленості MH-Sensor-Series на основі фоторезистора призначений для вимірювання рівня освітленості у приміщенні. Завдяки високій чутливості забезпечується точна реєстрація змін інтенсивності світла, що дозволяє ефективно керувати режимами освітлення залежно від природних умов. Вихідний сигнал аналоговий, що потребує додаткового перетворення мікроконтролером у цифровий формат.

Датчик простий у використанні, компактний за розмірами та сумісний з мікроконтролерами типу Arduino. Таблиця 1.1 з технічними характеристиками датчика MH-Sensor-Series наведена в ДОДАТКУ В.

На рисунку 1.5 зображена схема підключення датчику освітленості MH-Sensor-Series до Arduino.

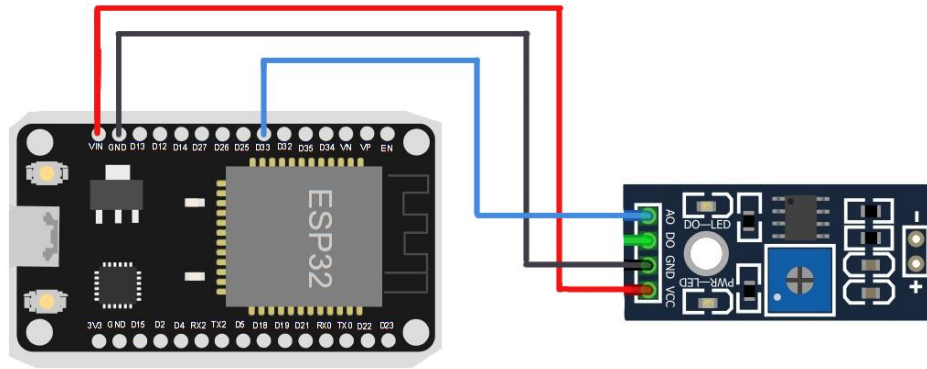


Рисунок 1.5 – Підключення датчика освітленості MH-Sensor-Series до Arduino.

### 1.2.2 Датчик руху приміщення

Датчик руху (PIR-датчик, passive infrared sensor) – це електронний пристрій, призначений для автоматичного виявлення руху людини або інших теплокровних об'єктів у контрольованій зоні. Основною функцією датчика руху є визначення зміни теплового інфрачервоного випромінювання, яке генерується живими істотами під час їхнього переміщення, і подача відповідного сигналу для керування освітленням або іншими електронними пристроями. Використання датчика руху дозволяє підвищити комфорт експлуатації приміщень і суттєво зменшити енергоспоживання, автоматично вимикаючи світло за відсутності активності у приміщенні.

Принцип роботи датчика руху базується на піроелектричному ефекті. Сенсорний елемент пристрою складається з піроелектричного кристалу, який чутливо реагує на зміну інтенсивності інфрачервоного випромінювання в зоні контролю. Коли людина або інший теплокровний об'єкт рухається, виникає різниця теплового випромінювання, що фіксується сенсором. Електронна схема датчика аналізує отриманий сигнал і при виявленні руху генерує цифровий вихідний імпульс для подальшої обробки контролером або керування електричними пристроями. На рисунку 1.6 зображено датчик руху, який, який буде використано для проектування системи.



Рисунок 1.6– Датчик руху інфрачервоний (PIR Sensor) HC-SR501

У даній системі використовується датчик руху моделі HC-SR501, який призначений для швидкого і надійного визначення наявності руху в приміщенні. Перевагами моделі є наявність можливості налаштування часу утримання вихідного сигналу після виявлення руху, що дозволяє гнучко адаптувати пристрій до різних умов експлуатації. Датчик має цифровий вихід, завдяки чому легко інтегрується з мікроконтролерами та іншими пристроями автоматизації без додаткових схем перетворення сигналу.

Таблиця 1.2 з технічними характеристиками датчика руху HC-SR501 наведена у ДОДАТКУ В

### 1.2.3 Світлодіодна стрічка RGB (5050)

Світлодіодна стрічка RGB – це гнучкий світловий елемент, який складається з послідовно розташованих світлодіодів типу SMD 5050, здатних генерувати різні кольори шляхом змішування трьох основних кольорів: червоного (Red), зеленого (Green) та синього (Blue). Така стрічка дозволяє створювати багатобарвне підсвічування, змінювати відтінки, яскравість і режими світіння залежно від завдань або обраного сценарію. Основна перевага RGB-стрічок полягає у можливості плавної зміни кольору та створення динамічних ефектів, що робить їх популярними у системах декоративного освітлення та підсвічування.

Принцип роботи RGB-стрічки ґрунтується на керуванні рівнем яскравості кожного з кольорових каналів через широтно-імпульсну модуляцію

(PWM). За допомогою мікроконтролера (наприклад, ESP32) можна регулювати подачу сигналу на кожен з каналів стрічки, змінюючи тим самим колір світіння в режимі реального часу. Стрічка живиться від джерела постійного струму 12 В і має високий рівень яскравості завдяки використанню потужних світлодіодів типу 5050. На рисунку 1.8 зображена RGB-стрічка, що використовується в проектуванні.



Рисунок 1.8 –Світлодіодна стрічка RGB (5050)

Світлодіодна стрічка сумісна з різними системами керування освітленням, легко монтується завдяки клейкому шару на звороті та може бути розрізана на секції для створення необхідної довжини підсвітки. Таблиця 1.3 з основними технічними характеристиками стрічки RGB 5050, що використовується в системі керування освітленням, наведена в ДОДАТКУ В.

Схематичне підключення світлодіодної стрічки RGB (5050)показано на рисунку 1.9.

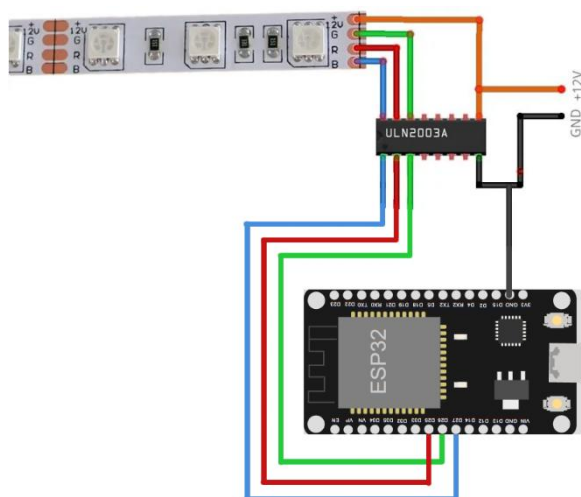


Рисунок 1.9 – Підключення світлодіодної стрічки RGB (5050) до ESP32

### 1.3 Процесорний блок

#### 1.3.1 Вибір мікропроцесора

Для розробки системи керування режимами світлодіодної стрічки через мобільний додаток необхідно обрати мікропроцесорний блок, який буде виконувати обробку вхідних сигналів від датчиків та формувати керуючі команди для освітлювальних елементів. При виборі мікроконтролера враховуються такі фактори, як доступність, вартість, технічні характеристики, наявність вбудованих інтерфейсів зв'язку (наприклад, Bluetooth або Wi-Fi) та зручність програмування.

Розглянемо кілька популярних варіантів мікроконтролерів, що можуть бути використані для реалізації даного проекту:

1. ArduinoNano – компактна плата на базі 8-бітного мікроконтролера ATmega328P з тактовою частотою 16 МГц і 32 КБ флеш-пам'яті. Плата підтримує програмування через середовище Arduino IDE, що робить її зручною для навчальних і тестових проектів. Завдяки широкій спільноті користувачів, відкритому коду та великій кількості бібліотек,

Arduino Nano залишається одним із найпопулярніших рішень для початківців. Вартість плати складає 2–3 долари.

2. ESP8266 (NodeMCU) – мікроконтролер, побудований на 32-бітній архітектурі Tensilica Xtensa LX106 із тактовою частотою 80 МГц. Особливістю плати є вбудований модуль Wi-Fi, що дозволяє реалізовувати бездротовий обмін даними без додаткових пристроїв. Обсяг флеш-пам'яті складає 4 МБ, оперативної – близько 20 КБ. Ціна Node MCU невисока (приблизно 3–4 долари), а розробка програмного забезпечення може здійснюватися через Arduino IDE або інші платформи.

3. ESP32 – це вдосконалений варіант мікроконтролера ESP, який використовує 32-бітний двоядерний процесор Tensilica Xtensa LX6 з тактовою частотою до 240 МГц. ESP32 має вбудовані модулі Wi-Fi і Bluetooth Low Energy (BLE), що дозволяє використовувати його для реалізації різноманітних систем керування та IoT-проектів. Обсяг оперативної пам'яті досягає 520 КБ, а флеш-пам'яті – до 4 МБ. Плата забезпечує високий рівень продуктивності, підтримує широкий спектр периферійних інтерфейсів, включаючи ШІМ, АЦП, DAC, SPI, I2C, UART. Ціна модуля ESP32 – від 5 до 7 доларів.

4. STM32F103C8T6 – мікроконтролер серії STM32, побудований на базі ядра ARM Cortex-M3 із тактовою частотою 72 МГц. Має 64 КБ флеш-пам'яті та 20 КБ оперативної пам'яті. Плата Blue Pill, на якій встановлений даний мікроконтролер, відома серед розробників завдяки своїй доступності (близько 3–4 доларів) і підтримці середовищ розробки Keil, STM32CubeIDE та Arduino IDE (через відповідні бібліотеки). Особливістю STM32 є висока продуктивність при низькому енергоспоживанні, що робить його привабливим для багатьох проектів.

На основі проведеного аналізу вирішено використовувати ESP32 як основний мікропроцесорний блок для реалізації системи керування режимами світлодіодної стрічки через мобільний додаток. Такий вибір обумовлений високою продуктивністю ESP32, наявністю вбудованого модуля Bluetooth Low

Energy для зв'язку з мобільним додатком, а також широкими можливостями інтеграції з іншими сенсорами та виконавчими пристроями.

### 1.3.2 Платформа ESP32

Мікроконтролер ESP32 був розроблений компанією Espressif Systems у 2016 році як універсальне рішення для розробки вбудованих систем і проектів Інтернету речей. Архітектура мікроконтролера побудована на основі 32-бітного двоядерного процесора Tensilica Xtensa LX6 з тактовою частотою до 240 МГц, що дозволяє обробляти дані значно швидше, ніж класичні 8-бітні рішення. Основною метою створення ESP32 було забезпечення широких можливостей підключення до бездротових мереж, тому мікроконтролер оснащений модулями Wi-Fi та Bluetooth Low Energy, що робить його ідеальним вибором для реалізації проектів дистанційного керування та моніторингу. Плата ESP32 дозволяє підключати різноманітні зовнішні сенсори для збору даних про навколишнє середовище, а також керувати різними виконавчими пристроями, такими як світлодіодна стрічка, реле та серводвигуни. Програмування ESP32 здійснюється за допомогою мови програмування C++, яка використовується у середовищі Arduino IDE, що робить платформу доступною навіть для користувачів з базовими знаннями в програмуванні. Для завантаження коду на плату використовується стандартний USB-роз'єм, що спрощує процес розробки та тестування проекту без необхідності додаткового програмного обладнання. Завдяки своїм технічним характеристикам, широким можливостям підключення та зручності використання, ESP32 отримав велику популярність серед розробників усього світу.

### 1.3.3 Апаратна частина платформи ESP32

На сьогоднішній день існує кілька популярних платформ на базі ESP32, серед яких найбільш поширеними є ESP32-DevKitC, ESP32-WROOM-32 та ESP32-WROVER (з додатковою пам'яттю). Завдяки широким можливостям підключення, високій обчислювальній потужності та вбудованим модулям бездротового зв'язку, ESP32 став універсальним рішенням для створення систем автоматизації, управління освітленням та інших проектів Інтернету речей. Зазвичай на платі ESP32 розміщені всі необхідні компоненти для роботи:

- мікроконтролер;
- пам'ять;
- порти підключення;
- специфічні для кожної версії модулі (наприклад, Wi-Fi, Bluetooth, додаткова PSRAM);
- індикація живлення та статусу роботи.

Плати ESP32 підтримують живлення як від 5 В (через USB або контакт VIN), так і безпосередньо від джерела 3,3 В, що дозволяє гнучко інтегрувати їх у різноманітні електронні системи. Різні версії ESP32 розроблені для різних завдань: наприклад, ESP32-DevKitC підходить для базових проектів, тоді як ESP32-WROVER з розширеною пам'яттю використовується у складніших проектах з обробкою великого обсягу даних або графічним інтерфейсом.

Плата ESP32 включає в себе 32-бітний мікроконтролер Tensilica Xtensa LX6, а також необхідні елементи обв'язки для програмування та підключення зовнішніх пристроїв. Багато плат мають вбудований стабілізатор живлення 3,3 В, що дозволяє зручно підключати сенсори та інші модулі, які працюють від цього рівня напруги. Тактування мікроконтролера здійснюється від вбудованого кварцового резонатора з частотою 40 МГц, що забезпечує точну синхронізацію обчислювальних процесів.

Концептуально плати ESP32 програмуються через інтерфейс UART (послідовне з'єднання), однак на практиці більшість моделей підтримують завантаження коду через USB завдяки інтегрованим модулям USB-to-Serial, таким як CP2102 або CH340G. Це дозволяє здійснювати пряме підключення

плати до комп'ютера для програмування без використання додаткового обладнання. Деякі версії плат підтримують емуляцію пристроїв вводу, наприклад, клавіатури або миші, що відкриває додаткові можливості для розробки інтерактивних проєктів. У випадках, коли використовується окрема версія плати ESP32 без вбудованого USB-конвертера, для завантаження прошивки необхідно підключати зовнішній програматор або модуль USB-to-Serial.

Плати ESP32 надають широкий доступ до виводів мікроконтролера для підключення зовнішніх сенсорів, модулів і виконавчих пристроїв. Наприклад, на платі ESP32-DevKitC доступно до 34 універсальних цифрових входів/виходів, 18 з яких можуть генерувати ШІМ - сигнали, а також 12-бітні аналогові входи, що дозволяють зчитувати дані з аналогових датчиків. Виводи доступні через штирові роз'єми на платі, що спрощує підключення периферійних пристроїв. Крім того, для ESP32 існують численні плати розширення, відомі як «shields», які можуть додавати нові функціональні можливості — наприклад, модулі дисплеїв, датчиків, реле або драйвери для керування двигунами.

#### 1.3.4 Версії платформи ESP32

Платформа ESP32 представлена широким спектром модулів та плат розробки, які відрізняються за функціональністю, розмірами та призначенням. Нижче наведено огляд найбільш поширених версій, що використовуються в проєктах Інтернету речей, автоматизації та вбудованих системах.

1. ESP32-DevKitC – базова плата розробки з компактними розмірами та виведеними пінхедами для підключення периферійних пристроїв. Оснащена вбудованим USB-to-Serial конвертером, що спрощує завантаження прошивок через USB-кабель. Підходить для загальних проєктів керування та прототипування.

2. ESP32-WROOM-32 – популярний модуль з вбудованими Wi-Fi та Bluetooth Low Energy модулями, флеш-пам'яттю обсягом до 4 МБ та розширеними можливостями підключення. Часто використовується в готових платах або інтегрується безпосередньо в кінцеві пристрої.

3.ESP32-WROVER – модифікація з додатковою PSRAM пам'яттю обсягом до 8 МБ, що дозволяє використовувати модуль для більш ресурсомістких задач, таких як обробка аудіо, зображень або робота з графічними дисплеями.

4.ESP32-S2 – спрощена версія з підтримкою лише Wi-Fi (без Bluetooth), але з розширеними можливостями безпеки, вбудованим USB-інтерфейсом та покращеним енергоспоживанням. Призначена для проектів, де критично важлива низька потужність споживання та робота з підключенням по USB.

5.ESP32-C3 – плата на базі архітектури RISC-V, що підтримує Wi-Fi та Bluetooth Low Energy. Відрізняється покращеною енергоефективністю та підходить для проектів, де важлива сумісність з новими стандартами безпеки та зменшення витрат енергії.

6.ESP32-S3 – вдосконалена версія модуля з підтримкою AI-функцій, таких як розпізнавання голосу та обробка зображень. Оснащена додатковою PSRAM пам'яттю та спеціалізованими функціями для роботи з машинним навчанням.

7.ESP32-H2 – модуль, орієнтований на мережі Zigbee та Thread, без підтримки Wi-Fi. Призначений для проектів, де важлива енергоефективність та підтримка мереж з низьким енергоспоживанням.

8. ESP32-PICO-D4 – система в пакеті (SiP), яка об'єднує ESP32 чіп, кварцовий резонатор, флеш-пам'ять та інші компоненти в одному корпусі розміром 7×7 мм. Ідеально підходить для проектів з обмеженим простором.

Ці модулі та плати розробки на базі ESP32 забезпечують широкий спектр можливостей для реалізації різноманітних проектів, від простих пристроїв до складних систем з підтримкою штучного інтелекту та мережевих протоколів нового покоління.

### 1.3.5 Огляд мікроконтролера ESP32-WROOM-32

ESP32-WROOM-32 — це компактний та універсальний модуль, розроблений компанією Espressif Systems у 2016 році. Він є повноцінною платформою для розробки вбудованих систем і проєктів Інтернету речей, що поєднує в собі мікроконтролер, пам'ять та комунікаційні модулі. ESP32-WROOM-32 забезпечує високу продуктивність завдяки 32-бітному двоядерному процесору Tensilica Xtensa LX6, що працює на частоті до 240 МГц, а також підтримує підключення до мереж Wi-Fi та Bluetooth Low Energy (BLE) без необхідності додаткових модулів.

Модуль оснащено стандартним виводом для підключення до плати розробки (наприклад, ESP32-DevKitC) або до кінцевих пристроїв. Підключення та програмування модуля здійснюється через стандартний інтерфейс USB або послідовний порт UART. Для написання та завантаження програм використовується середовище Arduino IDE або спеціалізоване середовище ESP-IDF, що дозволяє працювати як онлайн, так і офлайн. Живлення модуля може здійснюватися від зовнішнього джерела 5 В або через роз'єм USB на платі розробки. Зовнішній вигляд модуля ESP32-WROOM-32 представлено на рис. 1.10.



Рисунок 1.10 – Мікроконтролер ESP32-WROOM-32

Модуль має до 34 універсальних контактів цифрових входів і виходів, а також кілька спеціалізованих контактів для підключення інтерфейсів, таких як SPI, I2C, UART, PWM, ADC та DAC. Деякі контакти можуть виконувати багато функціональні завдання в залежності від налаштування мікроконтролера в програмному коді. Виводи модуля зазвичай розташовані по краях корпусу і доступні через пінхеда або спеціальні роз'ємні площадки для монтажу на друкованій платі або підключення через паяльні з'єднання. Додатково на модулі можуть бути присутні зовнішні елементи обв'язки, такі як кварцовий резонатор для забезпечення стабільного тактування, а також резистори, конденсатори та інші компоненти, необхідні для стабільної роботи системи. Усі ці елементи та контакти дозволяють мікроконтролеру ESP32-WROOM-32 обробляти сигнали від датчиків, керувати різними електронними пристроями та виконувати завдання обробки даних у режимі реального часу. Додаткові характеристики модуля ESP32-WROOM-32 наведені в таблиці 1.4, що розміщена в ДОДАТКУ В.

Далі розглянемо особливості живлення модуля ESP32-WROOM-32. Живлення цієї плати може здійснюватися двома основними способами:

- через стандартний USB-роз'єм при підключенні до комп'ютера або адаптера живлення;
- через зовнішнє джерело живлення, підключене до контактів VIN і GND, з напругою в межах від 5 В до 12 В (при наявності вбудованого стабілізатора на платі).

Слід враховувати технічні обмеження за напругою та струмом, характерні для мікроконтролера ESP32. Усі входи та виходи модуля працюють на рівні логіки 3,3 В. Подача більшої напруги на пін може призвести до пошкодження плати, тому перед підключенням зовнішніх компонентів важливо передбачати захисні схеми або застосовувати дільники напруги. Для захисту входів використовуються вбудовані діоди Шотткі, що запобігають перевищенню допустимого рівня напруги. Максимальний струм, який можна подавати на один вихід, становить близько 40 мА, а загальний струм для всіх

виходів не повинен перевищувати 120...160 мА, залежно від конкретної моделі плати.

Більшість плат ESP32 обладнані світлодіодними індикаторами, серед яких PWR для відображення подачі живлення та один або декілька світлодіодів, підключених до цифрових виходів (зазвичай GPIO2 або GPIO5) для перевірки роботи програми. Крім того, на багатьох моделях присутні індикатори TX і RX для відображення активності послідовного порту під час обміну даними.

Типова схема живлення модуля ESP32-WROOM-32 наведена на рисунку 1.11.

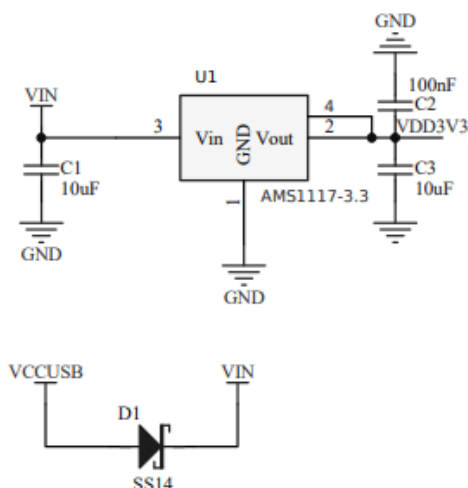


Рисунок 1.11 – Схема живлення ESP32-WROOM-32

Існує кілька модифікацій модуля ESP32-WROOM-32, які відрізняються технічними характеристиками, такими як обсяг пам'яті, доступні інтерфейси та додаткові функції. Вибір конкретної моделі залежить від вимог проєкту, але загальна структура контактів та розпіновка модуля залишаються схожими.

На рисунку 1.12 представлено детальну розпіновку модуля ESP32-WROOM-32, яка демонструє призначення основних контактів для підключення периферійних пристроїв та реалізації зв'язку.

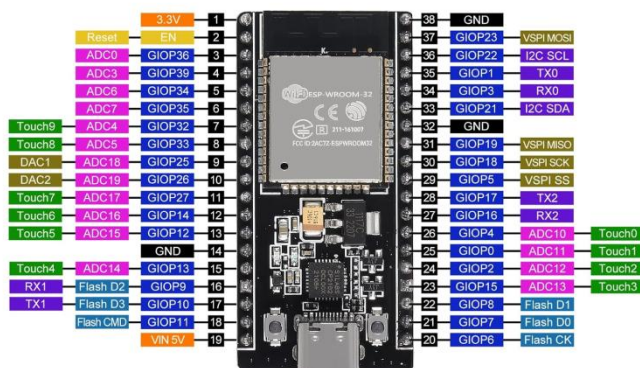


Рисунок 1.12 – Деталізована розпіновка модуля ESP32-WROOM-32

Нижче наведено основні піни живлення модуля ESP32-WROOM-32:

1. VIN – вхідний пін для підключення зовнішнього джерела живлення в діапазоні від 5 до 12 В. Через цей пін напруга подається на стабілізатор плати розробки.
2. 3.3V – вихідний пін стабілізатора напруги, що забезпечує стабільне живлення 3,3 В для зовнішніх модулів та датчиків. Максимальний струм залежить від конкретної плати, зазвичай до 500 мА.
3. GND – загальний контакт для підключення заземлення всіх зовнішніх компонентів.

Основні порти введення та виведення модуля ESP32-WROOM-32:

1. Цифрові входи/виходи (GPIO0–GPIO39) – до 34 універсальних пінів, що працюють на логічному рівні 3,3 В. Максимальний вихідний струм на один пін – до 40 мА. Деякі порти підтримують внутрішні підтягуючі резистори, які можна активувати програмно за потреби.
2. ШІМ (PWM) – до 18 пінів можуть генерувати широтно-імпульсні сигнали. Розрядність PWM регулюється в коді та може досягати 16 біт, що дозволяє створювати точні аналогові рівні.
3. АЦП – до 18 аналогових входів з розрядністю 12 біт. Діапазон напруги – від 0 до 3,3 В. Підключення вищої напруги може пошкодити мікроконтролер, тому варто використовувати захисні схеми або дільники напруги.

4. DAC – два 8-бітових цифро-аналогових перетворювачі для створення аналогових сигналів, наприклад, для керування яскравістю освітлення або генерації аудіосигналів.

5. UART – кілька апаратних портів для серійного зв'язку, серед яких основний – TX0, RX0, а також додаткові TX2, RX2. Використовується для обміну даними з комп'ютером, іншими модулями або зовнішніми пристроями.

6. SPI – до трьох апаратних шин SPI (VSPI, HSPI, FSPI), що дозволяють підключати датчики, пам'ять, дисплеї та інші пристрої.

7. I2C – два інтерфейси для підключення зовнішніх сенсорів та модулів, наприклад, датчиків температури, руху або освітленості.

Основні переваги ESP32-WROOM-32:

- високий рівень інтеграції, включаючи Wi-Fi та Bluetooth;
- потужний двоядерний процесор з частотою до 240 МГц;
- велика кількість універсальних пінів та підтримка різних інтерфейсів (SPI, I2C, UART, ADC, DAC).

Основні недоліки ESP32-WROOM-32:

- робота на рівні 3,3 В вимагає використання сумісних компонентів або додаткових захисних схем;
- складність налаштування для початківців через велику кількість функцій та гнучкість налаштувань.

ESP32-WROOM-32 – це потужний та гнучкий модуль, що дозволяє реалізовувати як прості проекти, так і складні системи автоматизації та керування в рамках Інтернету речей.

### 1.3.6 Середовище програмування – Arduino IDE

Для програмування ESP32-WROOM-32 широко використовується Arduino IDE — кросплатформне середовище для Windows, macOS і Linux, що дозволяє створювати, компілювати та завантажувати код. Воно має простий інтерфейс із редактором, консоллю повідомлень і меню навігації.

Редактор підтримує C/C++ з підсвічуванням синтаксису, автодоповненням і пошуком. Код компілюється і завантажується одним кліком. Для роботи з ESP32 потрібно встановити бібліотеки ESP32 Arduino Core, які додають підтримку Wi-Fi, Bluetooth, GPIO, ADC, DAC, PWM та інших функцій.

Скетчі – це програми, написані за допомогою Arduino IDE, що зберігаються у вигляді текстових файлів із розширенням .ino.

1. Setup(): Ця функція викликається один раз після запуску або перезавантаження мікроконтролера. Вона використовується для ініціалізації змінних, налаштування режимів пінів (вхід/вихід), підключення бібліотек та інших операцій, які виконуються на початку роботи програми.

2. Loop(): Після завершення роботи функції setup(), функція loop() виконується безперервно у циклі та визначає основну логіку роботи пристрою. Вона керує обробкою вхідних даних, обчисленнями, керуванням виконавчими пристроями та іншими операціями в реальному часі. Зовнішній вигляд Arduino IDE наведено на рисунку 1.13.

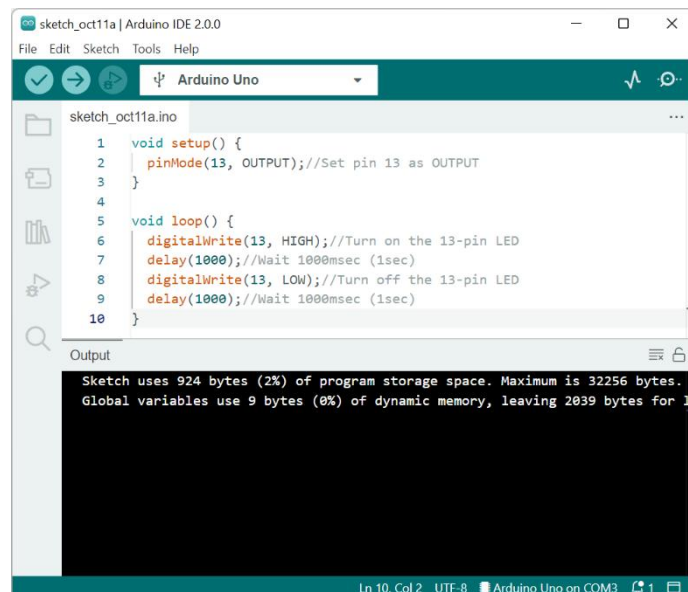


Рисунок 1.13 – Зовнішній вигляд Arduino IDE

14 вересня 2022 року було офіційно представлено стабільну версію Arduino IDE 2.0, яка стала значним кроком уперед у розвитку інструментів для

розробників мікроконтролерів. Ця версія побудована на основі фреймворку Eclipse Theia, що забезпечує сучасний та інтуїтивно зрозумілий інтерфейс користувача, а також підтримує розширення функціональності через модулі та плагіни.

Однією з ключових особливостей Arduino IDE 2.0 є інтеграція з Arduino CLI (Command Line Interface), що дозволяє ефективно компілювати та завантажувати скетчі на мікроконтролери. Це поєднання забезпечує гнучкість у розробці, дозволяючи як графічне, так і командне управління проектами.

## 2 РОЗРОБКА СИСТЕМИ КЕРУВАННЯ РЕЖИМАМИ СВІТЛОДІОДНОЇ СТРІЧКИ ЧЕРЕЗ МОБІЛЬНИЙ ДОДАТОК

### 2.1 Розробка функціональної схеми пристрою

Враховуючи особливості сучасних систем автоматизації освітлення та інтелектуального керування, для створення ефективної системи керування режимами світлодіодної стрічки було прийнято рішення використовувати централізовану архітектуру на базі мікроконтролера ESP32. Такий підхід забезпечує спрощення побудови системи, зменшує витрати на апаратні компоненти завдяки використанню компактних і недорогих датчиків, а також дозволяє централізовано керувати усіма елементами через єдиний обчислювальний блок. Функціональна структура системи зображена на рисунку 2.1.

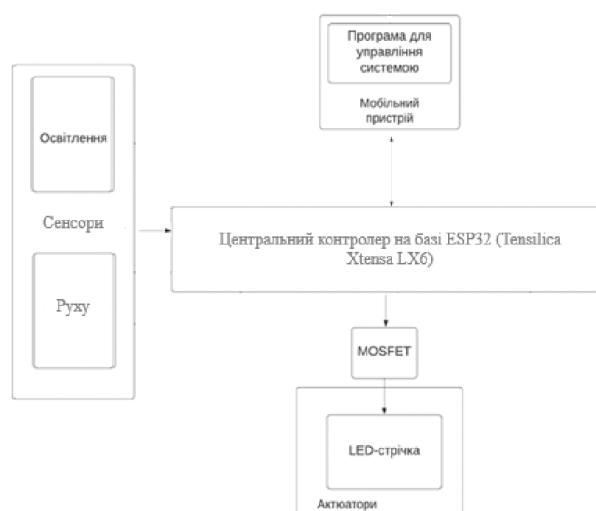


Рисунок 2.1 – Структурна схема системи керування мікрокліматом приміщення

Усі основні елементи системи, що відповідають за обробку даних від датчиків, керування світлодіодною стрічкою та обмін інформацією з мобільним додатком, зображені на функціональній схемі. Також на схемі наведені додаткові модулі, які можуть бути інтегровані для розширення можливостей системи.

Зв'язок між окремими компонентами організовано безпосередньо через ESP32, який виступає як центральний вузол керування. Через обмеження струму виходів ESP32 (приблизно 20 мА), підключення потужних виконавчих пристроїв, таких як світлодіодна стрічка, безпосередньо до контролера є неможливим. Для цього застосовується силова частина на базі драйвера ULN2003A, що дозволяє комутувати більші струми та надійно захищає ESP32 від перевантажень.

Живлення системи здійснюється від зовнішнього джерела постійного струму 12 В, яке подається на драйвер транзисторів і далі — на світлодіодну стрічку. Живлення мікроконтролера ESP32 організовано окремо через мікро-USB із напругою 5 В. Така конфігурація дозволяє зменшити навантаження на сам контролер і забезпечити стабільну роботу системи.

На відміну від реле, транзисторні ключі дозволяють реалізувати плавне регулювання сигналу, що особливо важливо для створення ефектів освітлення, наприклад, режиму «Світанок», коли яскравість стрічки змінюється поступово, імітуючи природне освітлення.

Датчики, підключені до системи, включають PIR-сенсор для виявлення руху (цифровий вхід ESP32, пін 13) та датчик освітленості, який підключено до аналогового входу ESP32 (пін 33) через резистивний подільник напруги з резистором 10 кОм. Це забезпечує вимірювання рівня освітленості в приміщенні та дозволяє системі реагувати на зміну умов освітлення.

Інтерфейс I2C у даній системі не використовується, оскільки всі необхідні підключення реалізовані безпосередньо через порти ESP32, що дозволяє спростити конструкцію та зменшити кількість компонентів.

Наступним кроком є детальний опис основних модулів системи, крім уже розглянутих сенсорів та виконавчих механізмів.

### 2.1.1 Транзисторний драйвер ULN2003A

Транзисторний драйвер ULN2003A – це електронний компонент, призначений для комутації електричних навантажень, які потребують значно більшого струму, ніж здатні забезпечити стандартні виходи мікроконтролера. Принцип роботи драйвера базується на використанні каскаду транзисторів Дарлінгтона, що дозволяє підсилювати слабкі сигнали керування від мікроконтролера та комутувати живлення навантаження. На відміну від реле, транзисторний драйвер не має механічних елементів, що забезпечує безшумну роботу, високу швидкодію та тривалий термін експлуатації.

Драйвер ULN2003A часто використовується у системах автоматизації для керування пристроями з живленням 12 В, зокрема світлодіодними стрічками, двигунами та іншими виконавчими механізмами. Його основні переваги включають компактний розмір, наявність вбудованих захисних діодів, здатність комутувати струм до 500 мА на кожен з семи каналів, а також підтримку стандартної напруги керуючого сигналу 3,3–5 В, що сумісна з ESP32. Основні відмінні риси транзисторного драйвера ULN2003A порівняно з іншими пристроями комутації такі:

- мінімальний рівень шуму під час роботи, завдяки відсутності механічних елементів;
- значно менше енергоспоживання порівняно з електромагнітними реле (економія до 95%);
- висока швидкодія (порядку мілісекунд), що дозволяє створювати плавні ефекти освітлення та миттєво реагувати на керуючі сигнали.

Для розробки системи керування режимами світлодіодної стрічки у даній роботі використовується саме транзисторний драйвер ULN2003A, який дозволяє ефективно комутувати живлення RGB-стрічки через сигнали

мікроконтролера ESP32. Зовнішній вигляд транзисторного драйвера представлений на рисунку 2.2. Основні технічні параметри транзисторного драйвера ULN2003A подано в таблиці 2.1 у ДОДАТКУ В.

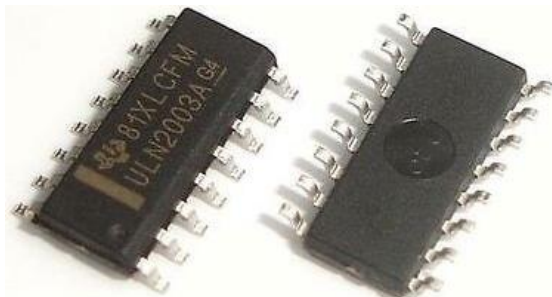


Рисунок 2.2 – Транзисторний драйвер ULN2003A

## 2.2 Розробка електричної схеми системи керування

Для розробки електричної схеми системи керування режимами світлодіодної стрічки було обрано середовище Easy EDA, що дозволяє створювати зручні графічні схеми підключення компонентів. Перед побудовою схеми необхідно визначити тип джерела живлення, що буде використовуватись у системі. У даній роботі застосовується імпульсний блок живлення моделі 1230, призначений для підключення до мережі змінного струму 220 В та формування стабільного вихідного струму 12 В із максимальним навантаженням до 3 А. Цей блок забезпечує живлення силових елементів системи, зокрема світлодіодної стрічки.

Для підключення блока живлення до електричної схеми використовується стандартний роз'єм типу DC Jack (як показано на рисунку 2.4), що забезпечує зручність монтажу та можливість підключення зовнішнього джерела живлення.

У таблиці 2.2 у ДОДАТКУ В наведено основні характеристики блока живлення, що використовується у системі керування.

У системі підключення основних компонентів здійснюється безпосередньо через стандартні роз'єми, пайку та пінові з'єднання на платі ESP32, що забезпечує надійність з'єднань і спрощує монтаж. Для живлення системи використовується роз'єм типу DC Jack, що дозволяє підключати блок живлення 12 В без необхідності додаткових перетворювачів або стабілізаторів.

Усі елементи системи пов'язані загальною шиною «GND» та відповідними лініями живлення: 5 В для мікроконтролера ESP32 (через мікро-USB) і 12 В для світлодіодної стрічки, підключеної через драйвер ULN2003A. Така схема дозволяє забезпечити чітку логіку роботи системи без перевантаження ESP32, а також залишає можливість для подальшого розширення функціоналу за рахунок підключення додаткових датчиків або модулів.

### 2.3 Програмний код та його оптимізація

Після завершення розробки електричної схеми системи керування режимами світлодіодної стрічки виникла необхідність створення програмного забезпечення, яке забезпечує інтеграцію всіх компонентів системи та керування ними. Розробка програмного коду здійснювалася у середовищі Arduino IDE, яке є зручним для роботи з мікроконтролерами ESP32 та підтримує всі необхідні бібліотеки.

Для реалізації програмної частини пристрою використовувалися такі бібліотеки:

- `#include <Wire.h>` – забезпечує взаємодію з модулями через інтерфейс I<sup>2</sup>C (наприклад, для OLED-дисплея);

- `#include <BLEDevice.h>` – використовується для реалізації бездротового зв'язку за допомогою Bluetooth Low Energy (BLE) між ESP32 та мобільним додатком;

- `#include <Preferences.h>` – дозволяє зберігати параметри налаштувань (колір освітлення, таймер, дні роботи) в постійній пам'яті ESP32;

– `#include <driver/ledc.h>` – бібліотека для керування широтно-імпульсною модуляцією (PWM), що використовується для плавної зміни кольорів та яскравості світлодіодної стрічки.

Зокрема, для керування кольором стрічки використовуються PWM-канали, налаштовані за допомогою функцій бібліотеки `ledc.h`, що забезпечує точність та плавність керування освітленням.

На рисунку 2.3 представлений фрагмент програмного коду, що відповідає за прийом команд через Bluetooth та подальше керування кольорами RGB-світлодіодної стрічки

```
class MyCallbacks : public BLECharacteristicCallbacks {
void onWrite(BLECharacteristic* pCharacteristic) override {
std::string value = pCharacteristic->getValue();
// Motion effect
if (value.length() == 1 && (uint8_t)value[0] == 0xA1) {
motionEffectEnabled = true;
return;
}
if (value.length() == 1 && (uint8_t)value[0] == 0xA0) {
motionEffectEnabled = false;
return;
}
// Timer enable/disable
if (value.length() == 1 && (uint8_t)value[0] == 0xF0) {
timerEnabled = true;
prefs.putBool("timer", true);
Serial.println("Timer ON via BLE");
return;
}
if (value.length() == 1 && (uint8_t)value[0] == 0xF1) {
timerEnabled = false;
prefs.putBool("timer", false);
Serial.println("Timer OFF via BLE");
return;
}
// Set active days
if (value.length() == 2 && value[0] == 0xF4) {
activeDays = value[1];
prefs.putUChar("days", activeDays);
return;
}
// Colour from the user
if (value.length() == 3 && !inMotionEffect) {
currentR = value[0];
currentG = value[1];
currentB = value[2];
manualOverride = true;
// Keep it if it's not pure black
if (!(currentR == 0 && currentG == 0 && currentB == 0)) {
prefs.putUChar("r", currentR);
prefs.putUChar("g", currentG);
prefs.putUChar("b", currentB);
}
// Update lastR/G/B always
lastR = currentR;
lastG = currentG;
lastB = currentB;
// Block the application of **only** between offtime and next ontime
if (timerEnabled && !activeTimerWindow) {
Serial.println("Colour saved but NOT applied (the timer has turned the light off)");
} else {
applyColor(currentR, currentG, currentB);
Serial.println(" Applied the colour manually");
}
}
};
```

Рисунок 2.3 – Фрагмент коду для прийому BLE-команд та керування кольором світлодіодної стрічки

Для налагодження роботи програмного забезпечення системи та моніторингу стану роботи пристрою використовується метод `Serial.println()`, який забезпечує виведення інформації в послідовний порт (Serial Monitor). Це дозволяє здійснювати контроль отриманих значень від датчиків, статусу

Bluetooth-з'єднання, а також перевіряти виконання логіки роботи таймера і зміни кольору світлодіодної стрічки.

Код розробленого пристрою структурований у вигляді великої кількості окремих функцій, що дозволило зробити його більш зручним для читання і значно скоротити розмір основного циклу. Основними функціями, які використовуються у програмі, є:

- `setup()` – для початкової конфігурації системи, включаючи налаштування пінів, ініціалізацію BLE-з'єднання, завантаження збережених налаштувань з пам'яті (Preferences), а також налаштування PWM для LED-стрічки;

- `checkTimerTrigger()` – для керування автоматичним включенням та виключенням світлодіодної стрічки за таймером, що враховує поточний час та дні тижня;

- `applyColor()` – для застосування заданого кольору до стрічки шляхом налаштування PWM-значень.

Додатково реалізовано спеціальні функції для обробки подій, які виникають при отриманні BLE-команд від мобільного додатку (наприклад, увімкнення та вимкнення ефекту руху, зміна налаштувань таймера). Ці функції забезпечують швидку реакцію пристрою на дії користувача та підтримують стабільну роботу системи.

Далі, розглянемо приклад програмного коду, що відповідає за керування таймером включення та виключення світлодіодної стрічки, який представлено на рисунку 2.4.

```

void checkTimerTrigger() {
  if (!TimerEnabled) {
    Serial.println("Timer off");
    return;
  }
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo)) {
    Serial.println("Failed to get the time");
    return;
  }
  int hour = timeinfo.tm_hour;
  int minute = timeinfo.tm_min;
  int weekday = (timeinfo.tm_wday + 6) % 7;
  // calculate the timer window (on-off)
  activeTimerWindow =
    (hour > onHour || (hour==onHour && minute>=onMinute)) &&
    (hour < offHour || (hour==offHour && minute<=offMinute));
  Serial.printf(" CNow %02d:%02d, day of the week: %d\n", hour, minute, weekday);
  Serial.printf(" Active days: 0b" BYTE_TO_BINARY_PATTERN "\n", BYTE_TO_BINARY(activeDays));
  Serial.printf(" time ON: %02d:%02d | time OFF: %02d:%02d\n", onHour, onMinute, offHour, offMinute);
  if (!(activeDays & (1<<weekday))) {
    Serial.println(" Today is not active day");
    return;
  }
  // Timer switch on
  if (hour==onHour && minute==onMinute && !isLightOnByTimer) {
    manualOverride = false;
    applyColor(lastR, lastG, lastB);
    isLightOnByTimer = true;
    Serial.printf("Turn on the colour by timer: R=%d G=%d B=%d\n", lastR, lastG, lastB);
  }
  // Timer switch off
  } else if (hour==offHour && minute==offMinute && isLightOnByTimer) {
    manualOverride = false;
    lastR = currentR; lastG = currentG; lastB = currentB;
    applyColor(0, 0, 0);
    isLightOnByTimer = false;
    Serial.printf("OFF by timer. The colour has been preserved: R=%d G=%d B=%d\n", lastR, lastG, lastB);
  }
}

```

Рисунок 2.4 – Логіка таймера автоматичного вмикання/вимикання світлодіодної стрічки

Як видно з рисунку 2.4, для реалізації автоматичного увімкнення та вимкнення світлодіодної стрічки використовується функція `checkTimerTrigger()`. У цій функції реалізовані умовні конструкції, які контролюють поточний час, задані параметри увімкнення (`onHour`, `onMinute`) та вимкнення (`offHour`, `offMinute`), а також активні дні тижня. При досягненні відповідних часових меж пристрій автоматично змінює стан світлодіодної стрічки, зберігаючи при цьому попередньо обраний користувачем колір.

На рисунку 2.5 представлений фрагмент програмного коду, який відповідає за обробку подій датчика руху (PIR). Під час виявлення руху датчиком система тимчасово змінює колір світлодіодної стрічки на білий для створення візуального ефекту, після чого автоматично повертається до попередньо встановлених параметрів освітлення через заданий інтервал часу. Це дозволяє створити додатковий комфорт і підвищити функціональність системи керування освітленням приміщення.

```
int motion = digitalRead(pirPin);
if (motionEffectEnabled && motion == HIGH && !inMotionEffect) {
  inMotionEffect = true;
  savedR = currentR; savedG = currentG; savedB = currentB;
  applyColor(255, 255, 255);
  motionStartTime = millis();
}
if (inMotionEffect && millis() - motionStartTime > motionEffectDuration) {
  applyColor(savedR, savedG, savedB);
  inMotionEffect = false;
}
```

Рисунок 2.6 – Обробка подій від датчика руху PIR у системі освітлення

Як видно з рисунку 2.6, для обробки роботи датчика руху використовується перевірка стану входу, підключеного до піну 13 ESP32. При виявленні руху вмикається спеціальний режим освітлення, який тимчасово змінює колір світлодіодної стрічки. Для контролю часу активації ефекту використовується функція `millis()`, що дозволяє точно визначати момент початку та завершення ефекту без блокування основного коду.

Також у коді реалізовано обробку флагів, які активуються або деактивуються при отриманні відповідних команд через Bluetooth-з'єднання. Наприклад, флаг `motion EffectEnabled` визначає, чи буде активовано ефект руху при спрацюванні PIR-сенсора. Таким чином, система дозволяє гнучко налаштовувати режими освітлення та забезпечує високий рівень інтерактивності між користувачем і пристроєм.

## 3 РОЗРОБКА ТА ФУНКЦІОНУВАННЯ МОБІЛЬНОГО ДОДАТКУ ДЛЯ КЕРУВАННЯ СВІТЛОДІОДНОЮ СТРІЧКОЮ

### 3.1 Призначення та можливості мобільного додатку

Мобільний додаток є головним засобом взаємодії користувача із системою керування світлодіодною стрічкою. Його мета — забезпечити зручний та інтуїтивний інтерфейс для налаштування, вибору режимів роботи та управління освітленням у приміщенні.

Додаток дозволяє змінювати колір і яскравість стрічки, обирати готові режими освітлення, встановлювати індивідуальні параметри або використовувати автоматичні сценарії на основі розкладу чи даних датчиків. Завдяки гнучкій структурі підтримується налаштування профілів освітлення для різних сценаріїв — роботи, відпочинку чи нічного режиму.

Інтерфейс додатку виконано англійською мовою з урахуванням простоти використання та доступності для різних користувачів.

### 3.2 Структура та інтерфейс користувача

Інтерфейс мобільного додатку побудовано за принципом швидкого доступу до всіх основних функцій керування системою освітлення. Головна сторінка містить інтерактивні елементи для зміни кольору та яскравості світлодіодної стрічки, а також перемикачі для активації основних датчиків. Внизу розміщене зручне меню навігації, яке дозволяє переходити до розширених налаштувань сценаріїв, роботи сенсорів, таймера та Bluetooth-з'єднання.

На рисунку 3.1 наведено вигляд головного екрана додатку, який є відправною точкою для подальшої навігації між функціональними розділами системи.

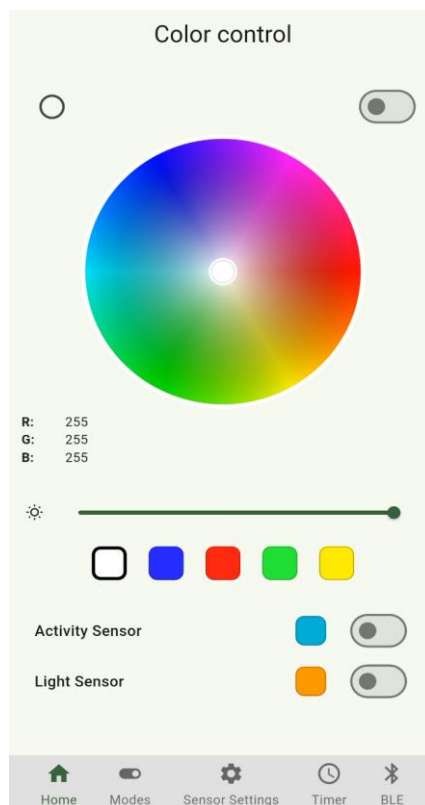


Рисунок 3.1–Головний екран мобільного додатку для керування освітленням

### 3.2.1 Головний екран

Головний екран мобільного додатку забезпечує швидке керування кольором і яскравістю світлодіодної стрічки та доступ до налаштувань датчиків і пресетів. Центральне місце займає інтерактивне коло вибору кольору, де можна обрати будь-який відтінок RGB, переміщуючи маркер. Зміна кольору відбувається автоматично з затримкою близько 200 мс, щоб уникнути перевантаження системи.

На екрані є два режими вибору кольору: стандартний і альтернативний. Вибраний колір чітко відображається на колі керування. В альтернативному режимі центральний круг має поточний колір освітлення, а в стандартному — кольорова обводка навколо кола показує вибраний відтінок. Це допомагає швидко зорієнтуватися у виборі кольору навіть при зміні режимів.

Стандартний режим дозволяє обирати будь-які відтінки, включно з білими, завдяки центральній області кола, що додає білу складову. Альтернативний режим працює лише з насиченими відтінками, де одне зі значень RGB завжди нульове, що виключає білі та пастельні кольори і створює яскраві акцентні ефекти, оптимальні для декоративного освітлення.

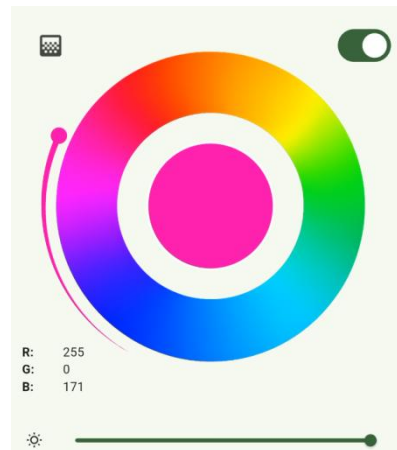


Рисунок 3.2 – Альтернативний спектр вибору кольору у мобільному додатку

Основна відмінність альтернативного спектра полягає у відсутності можливості вибору білого або близьких до нього відтінків. Кожен колір у цьому режимі буде максимально насиченим і контрастним, що підходить для підсвічування акцентних зон або створення яскравих ефектів. Стандартний режим дає змогу регулювати яскравість і тон, включаючи пастельні та білі відтінки.

Під кольоровим кругом розташовані числові значення RGB і слайдер для налаштування яскравості від 0 до 100%. Яскравість впливає на обраний колір, забезпечуючи точне налаштування освітлення.

У нижній частині екрана — пресети у вигляді кольорових квадратів для збереження та швидкого вибору улюблених комбінацій кольору і яскравості. Натискання на пресет встановлює збережені параметри, тривале — зберігає поточний колір.

Окремо розміщені перемикачі для активації датчика руху (ActivitySensor) та датчика освітленості (LightSensor) з міні-пресетами

кольору, які використовуються при спрацьовуванні сенсорів і можуть бути змінені.

Головний слайдер у верхній частині відповідає за повне увімкнення чи вимкнення світлодіодної стрічки. Якщо світло вимкнене, зміни кольору зберігаються, але не застосовуються до стрічки. При увімкненні встановлюється збережений колір.

Меню навігації внизу екрана дає доступ до налаштувань сценаріїв, сенсорів, таймера та Bluetooth-з'єднання для гнучкого керування системою.

### 3.2.2 ЕкранModes

Екран Modes призначено для організації гнучкого керування сценаріями роботи системи освітлення. На рисунку 3.3 демонструється перелік сценаріїв, кожен із яких оформлений у вигляді окремої картки на сторінці Modes.

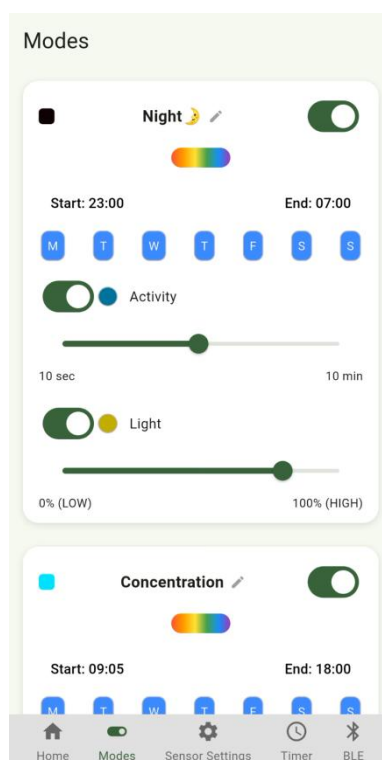


Рисунок 3.3 – загальний вигляд сторінки Modes.

Для розширення функціоналу передбачена кнопка додавання нового сценарію, що дозволяє налаштовувати освітлення відповідно до індивідуальних потреб.

Кожен сценарій містить низку елементів керування: загальний перемикач увімкнення або вимкнення сценарію, назву з можливістю редагування, кольоровий індикатор поточного кольору, а також діалогове вікно вибору кольору та яскравості, яке зображено на рисунку 3.4. Всі внесені зміни застосовуються миттєво, що забезпечує зручність і оперативність налаштувань.

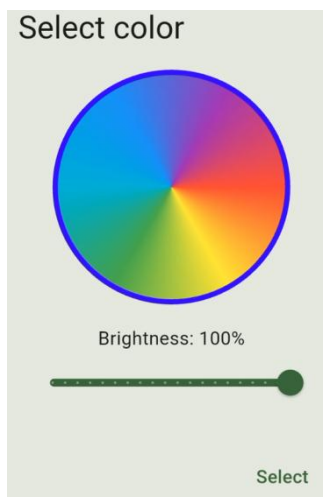


Рисунок 3.4 –вікно вибору кольору та яскравості

Система підтримує гнучке налаштування часу дії сценарію: визначається час початку і завершення, а також вибираються дні тижня для активації режиму. Для запобігання конфліктам між сценаріями реалізовано автоматичний контроль часових інтервалів. У випадку перекриття часових інтервалів система здійснює підсвічування відповідних полів та деактивує конфліктний сценарій, що продемонстровано на рисунку 3.5, запобігаючи одночасній активації взаємовиключних режимів.

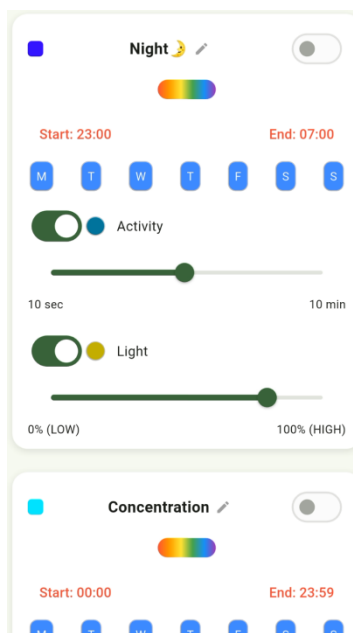


Рисунок 3.5 – Підсвічування та деактивація конфліктуючого сценарію через накладення таймерів

Кожен сценарій дозволяє незалежно увімкнути або вимкнути датчик руху (Activity) та датчик освітленості (Light). Для них можна вибирати колір, яскравість та налаштовувати час утримання ефекту після спрацювання руху (від 10 секунд до 10 хвилин) і рівень чутливості датчика освітленості в відсотках.

Приклад: сценарій «Night» активується з 23:00 до 07:00. Світло за замовчуванням вимкнене. Датчик руху увімкнений з утриманням ефекту 10 хвилин. При спрацюванні світло м'яке бірюзове з яскравістю 50%, що створює комфортне підсвічування. Якщо рух відсутній 10 хвилин, світло вимикається. Це зручно для пересування вночі без надмірного освітлення.

Інтерфейс Modes зроблено інтуїтивним для швидкої адаптації режимів освітлення.

### 3.2.3 Екран SensorSettings

Екран Sensor Settings дозволяє детально налаштувати два основні датчики системи — руху (ActivitySensor) та освітленості (Light Sensor). Тут

реалізовані базові глобальні параметри, які впливають на роботу системи незалежно від режимів.

Для Activity Sensor доступні три параметри: чутливість до відстані (Distance sensitivity), тривалість ефекту (Motion duration) та таймаут (Timeout), які визначають дальність виявлення руху, швидкість реакції та час утримання увімкненого стану.

Датчик освітленості має три ключові налаштування: поріг темряви (Darkness threshold), адаптацію до освітлення (Ambient adaptation) та криву чутливості (Sensitivity curve), що регулюють рівень спрацьовування та швидкість реагування.

Значення параметрів керуються слайдерами, що забезпечують зручне і точне налаштування системи під різні умови. Інтерфейс екрану наведено на рисунку 3.6.

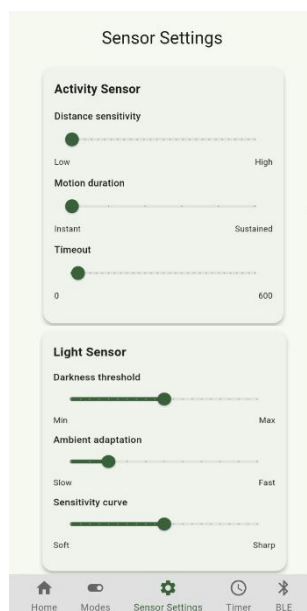


Рисунок 3.6 – Інтерфейс екрану Sensor Settings

### 3.2.4 Екран Timer

Екран Timer призначений для налаштування автоматичного увімкнення та вимкнення основного світла світлодіодної стрічки за розкладом, що

забезпечує зручність і енергоефективність. На екрані є два блоки: увімкнення світла (Turn On Light) і вимкнення світла (Turn Off Light), кожен із власними параметрами.

Керування здійснюється через перемикачі режимів. Для кожного встановлюється час увімкнення або вимкнення та вибір днів тижня, які підсвічуються для зручності.

Налаштування таймера впливають лише на основне світло і не залежать від сценаріїв у розділі Modes. При увімкненні за таймером активується останній збережений колір і яскравість.

Інтерфейс Timer забезпечує легке і швидке коригування параметрів автоматичного керування освітленням згідно з індивідуальними потребами.

Вигляд екрану Timer наведено на рисунку 3.7.

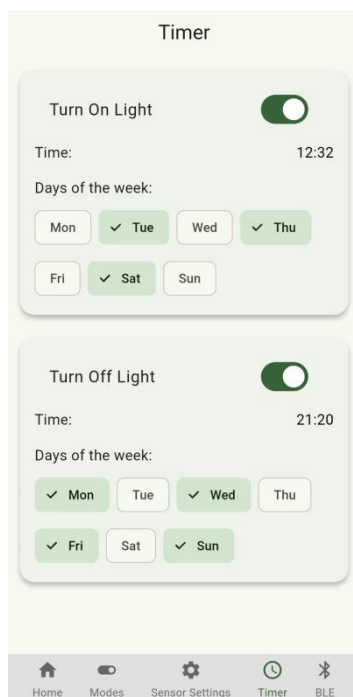


Рисунок 3.7 – Інтерфейс екрану Timer

### 3.2.5 Екран керування Bluetooth-з'єднанням

На екрані реалізовано керування з'єднанням з Bluetooth-пристроєм, який відповідає за управління світлодіодною стрічкою. Для зручності передбачено

автоматичне підключення до відомого пристрою ESP32\_RGB\_Control. Функцію автопідключення можна вмикати або вимикати за допомогою перемикача у верхній частині екрану.

При ввімкненні додатку або появі сигналу система автоматично намагається встановити зв'язок із пристроєм. У разі невдалих спроб відбувається до чотирьох повторних спроб, після чого з'являється повідомлення про помилку. Якщо пристрій доступний, підключення відбувається миттєво.

Загальний вигляд екрану наведено на рисунку 3.8.

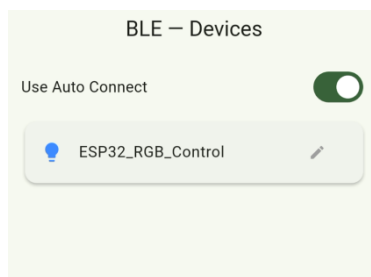


Рисунок 3.8 – Загальний вигляд екрану з'єднання

Основний екран відображає назву відомого пристрою та іконку статусу підключення. При натисканні на картку відкривається діалогове вікно з детальною інформацією — ім'ям пристрою, MAC-адресою, силою сигналу (RSSI) та кнопкою для ручного підключення або відключення. Детальний вигляд інформаційного діалогового вікна наведено на рисунку 3.9.

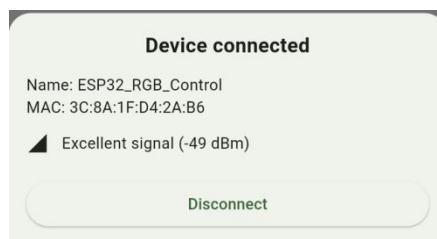


Рисунок 3.9 – Детальний вигляд інформаційного діалогового вікна

На відміну від попередньої версії, додано можливість перейменування пристрою в межах мобільного додатку, що забезпечує гнучкість у керуванні декількома пристроями або для зручності користувача.

Інтерфейс орієнтований на підтримку стабільного Bluetooth-з'єднання, що гарантує безперебійну роботу системи освітлення через бездротовий канал.

### 3.3 Технічна реалізація та використані програмні засоби

Для забезпечення ефективної взаємодії між мобільним додатком та апаратною частиною системи освітлення використовується протокол Bluetooth Low Energy (BLE). Це дозволяє реалізувати енергозберігаюче бездротове з'єднання з пристроєм на базі мікроконтролера ESP32, що керує світлодіодною стрічкою.

В якості основного інструменту розробки програмного забезпечення мобільного додатку використовується Flutter–кросплатформений фреймворк, який забезпечує зручний та швидкий спосіб створення інтерфейсів користувача. У коді додатку реалізовано логіку роботи з BLE, що передбачає сканування пристрою, підключення, відправку кольору та налаштувань, а також обробку відповідей від пристрою.

На рисунку 3.10 представлено фрагмент коду, що відповідає за вибір кольору світлодіодної стрічки та оновлення відповідного стану. На цьому ж екрані видно інтерфейс мобільного додатку, в якому користувач може обирати кольори, яскравість та керувати увімкненням світла.

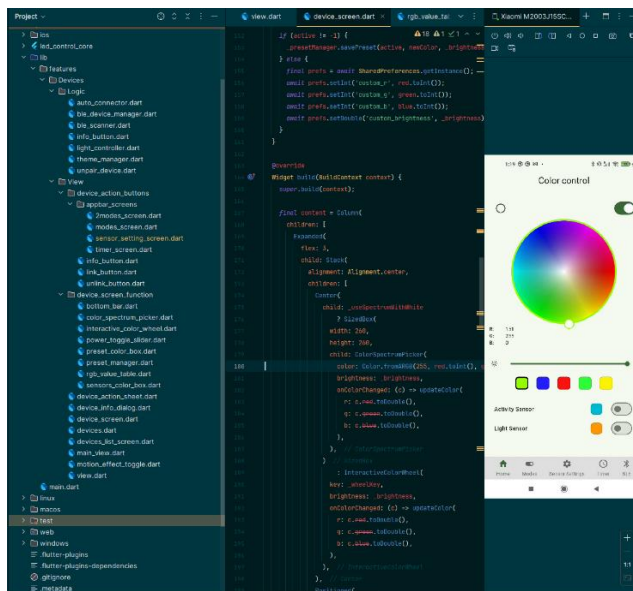


Рисунок 3.10 –Фрагмент структури Flutter-проекту з реалізацією вибору кольору та яскравості світлодіодної стрічки

Для відлагодження та контролю роботи зв'язку додаток веде детальний лог обміну BLE-командами. Рисунок 3.11 ілюструє приклад такого логу в середовищі розробки Android Studio. Тут відображаються повідомлення про успішне підключення до пристрою, обробку службових команд та передачу кольорів у вигляді RGB-значень.

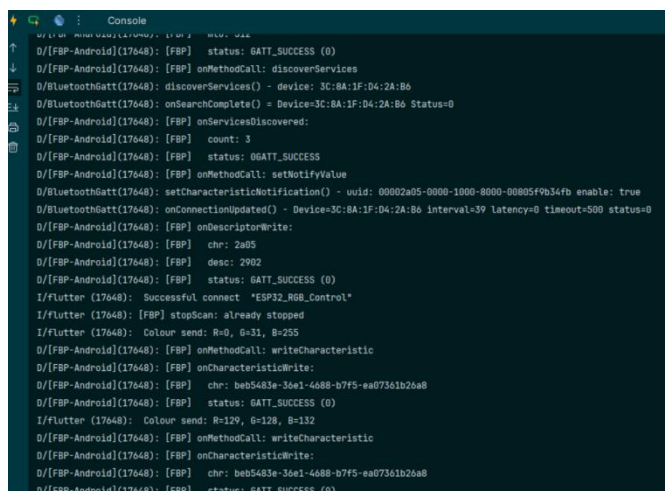


Рисунок 3.11 –Фрагмент консолі відлагодження з повідомленнями про Bluetooth-з'єднання та передачу кольору

Використання BLE забезпечує стабільне та швидке оновлення параметрів освітлення в реальному часі. Для збереження останніх вибраних кольорів і режимів застосовується бібліотека Shared Preferences, що дозволяє зберігати налаштування навіть після перезапуску додатку.

Таким чином, поєднання Flutter, BLE та системи збереження налаштувань створює зручний, адаптивний і ефективний інструмент для керування світлодіодною стрічкою, який відповідає вимогам сучасних мобільних додатків.

### 3.4 Алгоритм взаємодії додатку з апаратною частиною системи

Мобільний додаток взаємодіє з апаратною частиною через бездротовий протокол (BLE), який забезпечує енергоефективний і надійний зв'язок між пристроями. При запуску додатку відбувається автоматичне сканування навколишніх BLE-пристроїв для пошуку ESP32\_RGB\_Control.

Після виявлення пристрою встановлюється з'єднання, яке підтримується активним під час роботи додатку. У разі втрати зв'язку додаток робить кілька спроб повторного підключення. З'єднання також може бути ініційоване вручну користувачем.

Обмін даними здійснюється шляхом читання і запису характеристик BLE-сервісу. Додаток відправляє пакети команд для встановлення кольору, яскравості, активації режимів і налаштування таймера. ESP32 приймає, обробляє команди і виконує відповідні дії, а також може надсилати статусні повідомлення назад.

Для ідентифікації інформації використовуються унікальні UUID сервісу і характеристики. Операції відбуваються асинхронно, що забезпечує швидкий відгук і плавну роботу інтерфейсу.

Таким чином, взаємодія дозволяє ефективно керувати світлодіодною стрічкою в реальному часі та отримувати актуальну інформацію про стан системи.

## ВИСНОВКИ

Отже, на основі розробки системи керування режимами світлодіодної стрічки через мобільний додаток можна зробити такі висновки:

1. У першому розділі проведено аналіз предметної області, розглянуто сучасні технології керованого освітлення, визначено основні параметри системи, зокрема роль датчиків руху та освітленості. Обґрунтовано вибір мікроконтролера ESP32 як центрального елемента системи завдяки його потужності та бездротовому зв'язку.

2. У другому розділі розроблено функціональну та електричну схеми системи, здійснено вибір і підключення основних компонентів: світлодіодної стрічки RGB, драйвера ULN2003A, датчиків руху та освітленості. Оптимізовано програмний код для ESP32 для обробки сигналів та керування підсвіткою.

3. Третій розділ присвячено розробці мобільного додатку на Flutter з інтуїтивним інтерфейсом для керування кольором, яскравістю, режимами та таймерами. Забезпечено надійний зв'язок із апаратною частиною через Bluetooth Low Energy з автоматичним підключенням і збереженням налаштувань.

Для подальшого вдосконалення доцільно розглянути інтеграцію Wi-Fi для дистанційного керування, розширення функціоналу додатку складнішими сценаріями та підтримкою голосового управління.

Розроблена система є ефективним і сучасним рішенням, що підвищує комфорт, сприяє енергоощадженню та забезпечує гнучкість налаштувань згідно з потребами користувача.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ArduinoNano V3.0 AVR ATmega328P з розпаяними роз'ємами. Arduino. [Електронний ресурс] / arduino.ua – Режим доступу: www / URL: <https://arduino.ua/prod166-arduino-nano-v3-0-avr-atmega328p-s-raspayannimi-razemami> – 24.12.2024 – Загол. з екрану.
2. ESP32 DevKit v1. Документація. Espressif Systems. [Електронний ресурс] / espressif.com – Режим доступу: www / URL: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkit-v1.html> – 10.01.2025 – Загол. з екрану.
3. Bluetooth Low Energy. Основи та застосування. Nordic Semiconductor. [Електронний ресурс] / nordicsemi.com – Режим доступу: www / URL: <https://www.nordicsemi.com/Products/Low-power-short-range-wireless/Bluetooth-low-energy> – 20.03.2025 – Загол. з екрану.
4. Flutter. Cross-platform mobileapp development. Flutter.dev. [Електронний ресурс] / flutter.dev – Режим доступу: www / URL: <https://flutter.dev/> – 15.02.2025 – Загол. з екрану.
5. Light-EmittingDiode (LED) Fundamentals. Electronics Tutorials. [Електронний ресурс] / electronics-tutorials.ws – Режим доступу: www / URL: [https://www.electronics-tutorials.ws/io/io\\_2.html](https://www.electronics-tutorials.ws/io/io_2.html) – 22.04.2025 – Загол. з екрану.
6. Принцип роботи транзисторного драйвера ULN2003A. Electronics Hub. [Електронний ресурс] / electronicshub.org – Режим доступу: www / URL: <https://www.electronicshub.org/uln2003-datasheet/> – 12.05.2025 – Загол. з екрану.
7. Управління світлодіодною стрічкою RGB через мікроконтролер ESP32. RandomNerdTutorials. [Електронний ресурс] / randomnerdtutorials.com – Режим доступу: www / URL: <https://randomnerdtutorials.com/esp32-rgb-led-strip/> – 05.06.2025 – Загол. з екрану.

8. Протокол Bluetooth Low Energy (BLE). Texas Instruments. [Електронний ресурс] / ti.com – Режим доступу: www / URL: <https://www.ti.com/solution/bluetooth-low-energy> – 18.05.2025 – Загол. з екрану.

9. Основи розробки мобільних додатків на Flutter. Flutter Documentation. [Електронний ресурс] / flutter.dev – Режим доступу: www / URL: <https://docs.flutter.dev/get-started/codelab> – 23.05.2025 – Загол. з екрану.

10. Збереження даних у Flutter за допомогою shared\_preferences. Pub.dev. [Електронний ресурс] / pub.dev – Режим доступу: www / URL: [https://pub.dev/packages/shared\\_preferences](https://pub.dev/packages/shared_preferences) – 28.05.2025 – Загол. з екрану.

11. Margolis M. Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects / M. Margolis. — O'Reilly Media, 2011. — 480 p.

12. Порівняння технологій Bluetooth Classic та Bluetooth Low Energy. Bluetooth.com. [Електронний ресурс] / bluetooth.com – Режим доступу: www / URL: <https://www.bluetooth.com/bluetooth-resources/bluetooth-classic-vs-bluetooth-low-energy/> – 07.05.2025 – Загол. з екрану.

13. Побудова користувацьких інтерфейсів у Flutter. Medium. [Електронний ресурс] / medium.com – Режим доступу: www / URL: <https://medium.com/flutter/> – 12.05.2025 – Загол. з екрану.

14. Програмування мікроконтролерів ESP32 на Arduino IDE. EspressifSystems. [Електронний ресурс] / espressif.com – Режим доступу: www / URL: <https://docs.espressif.com/projects/arduino-esp32/en/latest/> – 18.05.2025 – Загол. з екрану.

15. Розробка системи управління світлом через мобільний додаток. ResearchGate. [Електронний ресурс] / researchgate.net – Режим доступу: www / URL: <https://www.researchgate.net/publication/> – 24.05.2025 – Загол. з екрану.

16. Підключення Bluetooth-пристроїв у Flutter. FlutterAwesome. [Електронний ресурс] / flutterawesome.com – Режим доступу: www / URL: <https://flutterawesome.com/> – 30.05.2025 – Загол. з екрану.

17. Основи роботи з таймерами в мікроконтролерах. Microcontroller Tutorials. [Електронний ресурс] / microcontrollertutorials.com – Режим доступу:

www / URL: <https://microcontrollertutorials.com/timers/> – 05.05.2025 – Загол. з екрану.

18. Gomez C., Oller J., Paradells J. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology / C. Gomez, J. Oller, J. Paradells // Sensors. — 2012. — Vol. 12, No. 9. — P. 11734-11753.

19. Програмне забезпечення для управління світлодіодними стрічками. Git Hub. [Електронний ресурс] / github.com – Режим доступу: www / URL: <https://github.com/topics/led-strip-controller> – 18.45.2025 – Загол. з екрану.

20. Основи розробки мобільних додатків для систем розумного будинку. IEEE Xplore. [Електронний ресурс] / ieeeexplore.ieee.org – Режим доступу: www / URL: <https://ieeeexplore.ieee.org/document/> – 24.03.2025 – Загол. з екрану.