

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та кібербезпеки

(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА  
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

СИСТЕМА РОЗУМНОГО ОСВІТЛЕННЯ З  
БАГАТОФАКТОРНИМ КЕРУВАННЯМ

SMART LIGHTING SYSTEM WITH MULTI-FACTOR CONTROL

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти

групи КІ-41

Хомін Віталій Русланович

(підпис)

Керівник:

к.т.н., доцент

Лавренчук Світлана Василівна

(підпис)

Кваліфікаційну роботу

допущено до захисту

« 11 » червня 2024 р.

Гарант освітньої програми:

к.т.н., доцент

Лавренчук Світлана Василівна

(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та кібербезпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф. Н.Черняшук

« 10 » 01 2024 р.

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*Хоміну Віталію Руслановичу*

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Система розумного освітлення з багатофакторним керуванням

Керівник роботи к.т.н., доцент Лавренчук Світлана Василівна

затверджені наказом закладу вищої освіти від «30» грудня 2023 року № 459/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 10.06.2024р.

3. Вихідні дані до роботи джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування

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

Вступ

Аналіз предметної області

Підбір деталей для апаратної частини проекту

Проектування та тестування системи освітлення

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Система розумного освітлення з багатофакторним керуванням. Структурна схема

Система розумного освітлення з багатофакторним керуванням. Структурна електрична схема

Алгоритм прошивки мікроконтролера

Зовнішній огляд спроектованого пристрою

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз проблемної області</i>	<i>Лавренчук С.В., доцент</i>		
<i>Підбір деталей для апаратної частини проекту</i>	<i>Лавренчук С.В., доцент</i>		
<i>Проектування та тестування системи освітлення</i>	<i>Лавренчук С.В., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>		_____%	
<i>Академічна доброчесність</i>	<i>Міскевич О.І., асистент</i>		

7. Дата видачі завдання 10.01.2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Розділ 1. Аналіз проблемної області</i>	до 15.02.2024 р.	Виконано
2.	<i>Розділ 2. Підбір деталей для апаратної частини проекту</i>	до 15.03.2024 р.	Виконано
3.	<i>Розділ 3. Проектування та тестування системи освітлення</i>	до 04.05.2024 р.	Виконано
4.	<i>Висновки та пропозиції</i>	до 07.05.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 10.05.2024 р.	Виконано
6.	<i>Формування додатків</i>	до 15.05.2024 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 20.05.2024 р.	Виконано
8.	<i>Нормоконтроль</i>	до 01.06.2024 р.	Виконано
9.	<i>Інструментальна перевірка на академічний плагіат</i>	до 04.06.2024 р.	Виконано
10.	<i>Представлення кваліфікаційної роботи бакалавра до захисту</i>	до 11.06.2024 р.	Виконано

Здобувач вищої освіти

(підпис)

Хомін В.Р.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Лавренчук С.В.

(прізвище, ініціали)

## АНОТАЦІЯ

Хомін В.Р. Система розумного освітлення з багатофакторним керуванням.  
Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2024. 65 с.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел, додатку.

Перший розділ присвячено обґрунтуванню актуальності обраної теми, розглянуто сучасні пристрої та системи розумного освітлення, фактори, які можуть впливати на функціонування системи освітлення.

В другому розділі здійснено вибір компонентів для проектування системи розумного освітлення з багатофакторним керуванням. Розглянуто будову та принцип роботи світлодіодів, розумних лампочок, сенсорів освітленості, інфрачервоних приймачів, а також обґрунтовано вибір моделі плати мікроконтролера.

Третій розділ присвячено детальному опису етапів проектування пристрою та створення прошивки мікроконтролера для системи освітлення.

Ключові слова: Arduino, світлодіод, фоторезистор, пульт дистанційного керування, Arduino IDE.

## ANNOTATION

Khomin V.R. Smart lighting system with multi-factor control. Manuscript.

Qualifying work of a bachelor of EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2024. 65 p.

Qualification work consists of an introduction, three sections, conclusions, a references, and an appendix.

The first chapter is devoted to substantiating the relevance of the chosen topic; modern devices and systems of intelligent lighting are considered, as well as factors that can influence the functioning of the lighting system.

In the second section, the selection of components for designing a smart lighting system with multi-factor control is made. The design and principle of operation of LEDs, smart light bulbs, light sensors, and infrared receivers are considered, and the choice of microcontroller board model is justified.

The third section is devoted to a detailed description of the stages of designing a device and creating microcontroller firmware for lighting.

Keywords: Arduino, LED, photoresistor, remote control, Arduino IDE.

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Обґрунтування актуальності обраної теми.....	8
1.2 Сучасні пристрої для освітлення та розумні системи освітлення.....	9
1.3 Фактори, що можуть впливати на функціонування системи освітлення	14
РОЗДІЛ 2 ПІДБІР ДЕТАЛЕЙ ДЛЯ АПАРАТНОЇ ЧАСТИНИ ПРОЄКТУ .....	16
2.1 Джерела світла .....	16
2.1.1 Світлодіоди.....	16
2.1.2 Розумні лампочки .....	23
2.2 Сенсор освітленості.....	25
2.3 Інфрачервоний приймач .....	27
2.4 Вибір мікроконтролера .....	28
РОЗДІЛ 3 ПРОЄКТУВАННЯ ТА ТЕСТУВАННЯ СИСТЕМИ ОСВІТЛЕННЯ.	33
3.1 Схема та монтаж пристрою.....	33
3.2 Програмування мікроконтролера.....	38
3.3 Тестування роботи спроектованої системи освітлення .....	46
ВИСНОВКИ.....	51
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ.....	55

## ВСТУП

Впровадження Інтернету речей у системи освітлення дає багато переваг, серед яких розширені можливості керування (наприклад, заплановане освітлення або затемнення) та зниження витрат на електроенергію, що особливо актуально для українців в наш час. Завдяки швидкому розвитку технології IoT, підключення кожного світильника до Інтернету стає економічно вигідним. Розумне освітлення може змінювати спектр світла та адаптуватися до природного освітлення. Саме різноманітність сценаріїв освітлення потребує багатьох факторів керування.

Метою роботи є пошук ефективних та економних методів керування параметрами освітлення.

Об'єкт дослідження – моніторинг та управління освітленням.

Предмет дослідження – система багатофакторного управління освітленням.

Завдання, які необхідно виконати:

- здійснити огляд сучасних пристроїв та сценаріїв освітлення;
- вибрати компоненти для проектування апаратної частини системи освітлення з багатофакторним керуванням;
- розробка електричної структурної схеми пристрою;
- створення прошивки мікроконтролера.

Ця кваліфікаційна робота має практичне значення, адже розумне освітлення можна використовувати в умовах зростаючих тарифів на електроенергію та за необхідності зменшення енергоспоживання, наразі це доволі актуально в Україні, де енергетичні ресурси обмежені. Розумні системи освітлення дозволять значно скоротити витрати на електроенергію, а наявність автономного джерела енергії та низьке її споживання дадуть змогу мати хоча б мінімальне освітлення під час вимкнень електроенергії.

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

#### 1.1 Обґрунтування актуальності обраної теми

Розумні системи освітлення є частиною більш широкого тренду розвитку інтернету речей (IoT). Використання сучасних технологій, таких як датчики, мікроконтролери та алгоритми обробки даних, сприяє розвитку нових рішень та інновацій у галузі автоматизації та управління. Це відкриває нові можливості для досліджень та вдосконалення систем освітлення.

Сучасні системи освітлення часто споживають значну кількість електроенергії. Використання розумних систем, які автоматично регулюють рівень освітлення залежно від зовнішніх умов (природного освітлення або присутності людей), дозволяє значно знизити енергоспоживання. Це, у свою чергу, сприяє зменшенню викидів вуглекислого газу та інших шкідливих речовин в атмосферу, що є важливим кроком у боротьбі з глобальним потеплінням та зміною клімату.

Розумні системи освітлення дозволяють забезпечити підвищений комфорт користувачів, автоматично адаптувати рівень освітлення до їх потреб. Комфорт передбачає налаштування яскравості та кольору світла залежно від часу доби та рівня природного освітлення. Розумні системи освітлення створюють комфортне середовище для роботи, відпочинку та інших видів діяльності, що позитивно впливає на продуктивність та самопочуття.

Система розумного освітлення може бути інтегрована з іншими розумними системами у рамках концепції «розумного дому» або «розумного офісу». Це дозволяє створювати складніші та функціональні екосистеми, де всі елементи працюють разом для забезпечення максимального комфорту та ефективності. Наприклад, освітлення можна синхронізувати з системою безпеки або опалення, вентиляції та кондиціонування.

Інвестиції в розумні системи освітлення передбачають швидку окупність за рахунок зниження витрат на електроенергію. Крім того, оптимальне

використання освітлення зменшує навантаження на електричні прилади та подовжує їх термін служби, що знижує витрати на заміну та обслуговування обладнання.

Використання бюджетного мікроконтролера (такого як Arduino Nano) у ролі платформи для розробки розумної системи освітлення забезпечує доступність та простоту впровадження проекту.

Тому створення проєкту «Система розумного освітлення з багатофакторним керуванням» є актуальною задачею, адже він дасть змогу забезпечити енергозбереження, комфорт та зручність, інтеграцію з іншими розумними системами, економічну ефективність, сприяння технічному прогресу та простоту впровадження. Дослідження та проєктування в цьому напрямку є важливим та перспективним напрямком у сучасних умовах (коли спостерігається швидкий розвиток технологій та зростають вимоги до ефективності та комфорту). Крім того, військові дії на території України також спонукають до пошуку нових та енергоефективних рішень в напрямку забезпечення балансу між комфортом та економією при енергоспоживанні.

Актуальність обраної теми також підтверджується наявністю великої кількості публікацій та досліджень в даному напрямку [9, 12, 14, 16, 17, 20].

## **1.2 Сучасні пристрої для освітлення та розумні системи освітлення**

Сучасні освітлювальні пристрої орієнтовані на підвищення комфорту, енергоефективності та естетичної привабливості освітлення.

Освітлення – один з основних споживачів електроенергії в побутових приміщеннях. Традиційним лампочкам більше ста років, в наш час вони є вкрай неефективними, адже перетворюють тільки біля 5% електроенергії, яку вони спожили, у видиме світло.

Порівняно з традиційними джерелами світла (звичайними лампами розжарювання) світлодіоди споживають менше енергії, мають довший термін служби, а також виділяють значно менше тепла, важливо що світлодіоди миттєво

(без затримок) вмикаються на повну яскравість. Завдяки своїй твердотільній конструкції світлодіоди міцні та стійкі до механічних впливів. Тому використання світлодіодів та ефективне керування ними за допомогою різних алгоритмів є одним із завдань проєкту.

Світлодіодні лампочки (LED) ефективніші, ніж традиційні лампочки, (перетворюють біля 20% електроенергії, яку вони споживають, у видиме світло). LED-лампи на даний час вважаються одними з найефективніших та довговічних джерел світла. Вони досить популярні, адже споживають менше енергії, при цьому мають довший термін служби і можуть забезпечити різні відтінки світла.

Розумні світлодіодні лампочки (рисунок 1.1) ще більш енергоефективні. Їх можна запрограмувати для регулювання яскравості та режимів роботи за допомогою мобільних пристроїв. Вони також можуть працювати в поєднанні з датчиками руху та освітлення, щоб вмикатися або вимикатися автоматично, зменшуючи надмірне освітлення та споживання енергії. Інтелектуальні LED-лампи (smart LED lamps) – один із трендів у світі розумних речей. Їх можна підключати за допомогою Wi-Fi або Bluetooth до смартфона, також можна використовувати голосового помічника. Такі smart-лампи зазвичай підтримують різні кольори світла та інтенсивність, їх також можна інтегрувати з іншими смарт-пристроями.

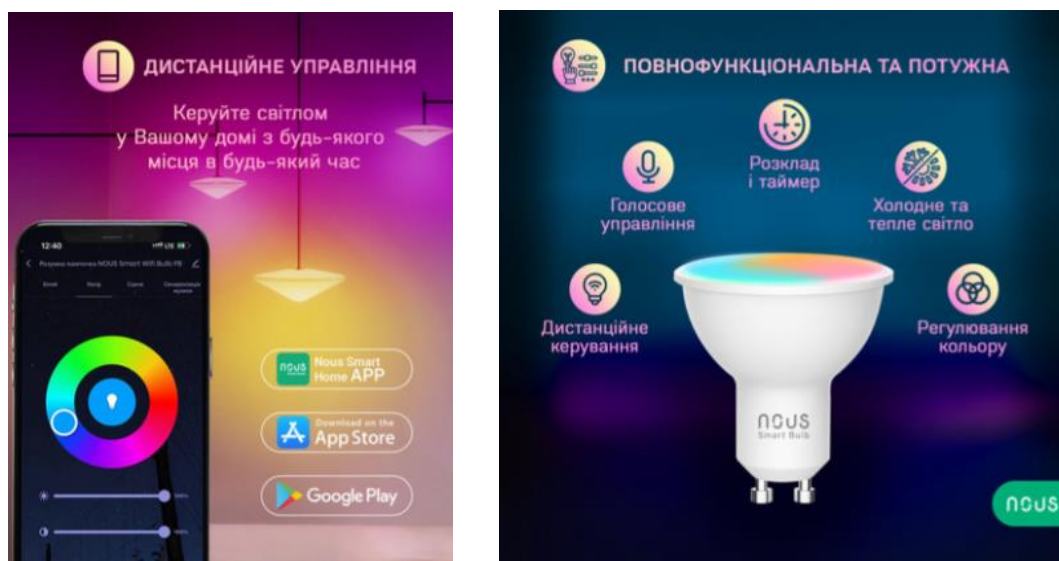


Рисунок 1.1 – Розумна лампочка NOUS Smart Wifi Bulb [19]

Одним із ефективних методів економії електроенергії є встановлення автоматичної системи управління освітленням.

Системи динамічного освітлення можуть змінювати колірну температуру та інтенсивність світла протягом дня для комфорту та підтримки біологічних ритмів людини. Такі системи здатні імітувати природне світло та допомагати покращити якість сну і продуктивність роботи.

Системи управління освітленням дають можливість автоматизувати управління освітленням, використовуючи таймери, сенсори або програмне забезпечення. Завдяки таким системам зручно налаштовувати та керувати освітленням залежно від потреб користувачів та їх вимог.

Енергоефективне освітлення з датчиками руху вмикає або зменшує інтенсивність світла залежно від присутності людей у приміщенні. Такі автоматизовані системи освітлення дозволяють заощадити енергію, вмикаючи світло лише там, де воно дійсно потрібне.

Щодня з'являються нові технології та інновації, які спрямовані на поліпшення якості і ефективності освітлення. Всі розумні пристрої можна підключити до однієї системи управління, а це дозволить користувачам контролювати своє споживання електроенергії, будувати складніші сценарії управління та призведе в кінцевому випадку до значної економії електроенергії.

Умовно автоматичні системи управління освітленням можна поділити на три види:

- автоматичне управління точкою освітлення. Зазвичай така система конструктивно входить до складу світильника, керувати можна одною лампою (світлодіодом) або їх системою (групою);

- автоматичне управління освітленням приміщення. Ця система трохи складніша, адже враховує більше характеристик та її складові можуть мати взаємовплив;

- автоматичне управління об'єднаного типу. Це комп'ютеризована установка (рисунок 1.2), яка крім освітлення може містити управління ще й іншими системами в приміщенні, або навіть в групі приміщень.

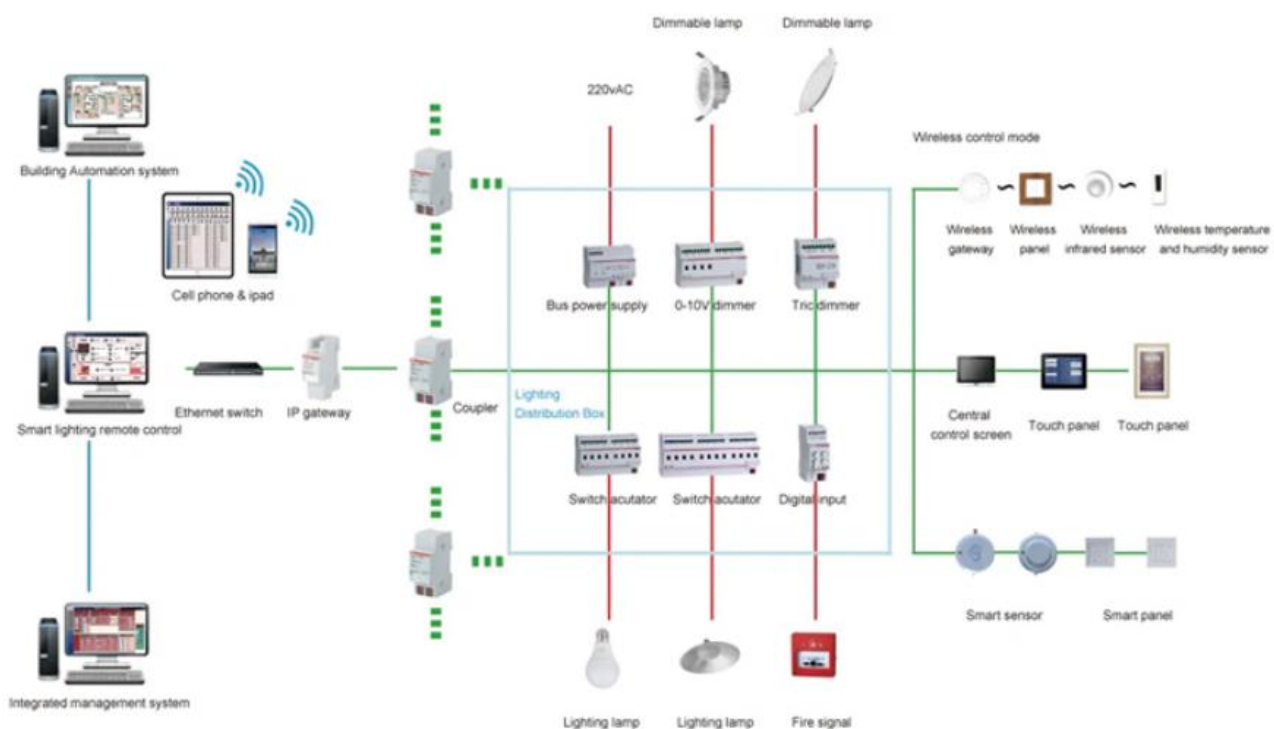


Рисунок 1.2 – Загальна структура розумної системи керування освітленням [7]

Досить часто існує потреба забезпечувати різну інтенсивність освітлення залежно від виду приміщень та способів їх використання (рисунок 1.3).



Рисунок 1.3 – Оптимальна освітленість для різних приміщень [5]

Розумні системи освітлення стають все більш популярними завдяки їх здатності підвищувати комфорт, енергоефективність і зручність управління.

Philips Hue [8] є однією з найпопулярніших розумних систем освітлення. Вона пропонує широкий спектр продуктів, серед яких лампи, світлодіодні стрічки, світильники та інші аксесуари для них. Вона пропонує дистанційне керування через мобільний додаток, має підтримку голосових асистентів

(Amazon, Google Assistant, Apple), дає можливість регулювання яскравості та кольорової температури. Можна робити розклади та різні сценарії освітлення для автоматизації різних ситуацій.

Китайська фірма Yeelight [18] (підрозділ Xiaomi) пропонує доступні розумні освітлювальні рішення з широким асортиментом продуктів. Вона пропонує ті ж самі функції, що й Philips Hue, але її перевагою є доступність, широкий асортимент та легка інтеграція з екосистемою Xiaomi.

LIFX [11] пропонує розумні лампи, які не потребують окремого хаба для роботи. Вони можуть підключатися безпосередньо до Wi-Fi.

Sylvania [6] пропонує розумні лампи, світильники та аксесуари, які підтримують Zigbee та Bluetooth, сумісні з багатьма платформами.

Nanoleaf [4] спеціалізується на декоративному освітленні, пропонує світлодіодні панелі, які можуть створювати унікальні візуальні ефекти, що дає можливість забезпечити унікальний дизайн та персоналізацію.

Переваги розумних систем освітлення:

- розумні системи освітлення дозволяють знижувати споживання енергії завдяки автоматичному регулюванню яскравості та використанню сенсорів освітленості;
- керування освітленням через додатки або голосові команди підвищує зручність для користувачів;
- можливість налаштування автоматичних сценаріїв освітлення підвищує комфорт та безпеку;
- різноманітні налаштування кольору та яскравості дозволяють створювати бажану атмосферу в приміщенні.

Існуючі розумні системи освітлення пропонують широкий спектр функцій і можливостей для покращення комфорту, енергоефективності та зручності управління освітленням. Вибір конкретної системи залежить від потреб користувача, бюджету та сумісності з іншими розумними пристроями. Розвиток IoT-технологій сприяє появі нових і вдосконалених рішень у цій галузі, що

дозволяє користувачам створювати більш індивідуалізовані та ефективні системи освітлення.

### **1.3 Фактори, що можуть впливати на функціонування системи освітлення**

Від розташування приміщення, режиму його роботи та від побажань користувачів залежить функціональність, якої вимагають від системи автоматизованого керування освітленням. Тому зручно ці системи класифікувати за основними функціями. В першу чергу така система передбачає підвищення енергоефективності та комфорту приміщень. Для цього використовують шість факторів (стратегій), які сприяють зниженню споживання енергії (рисунок 1.4).

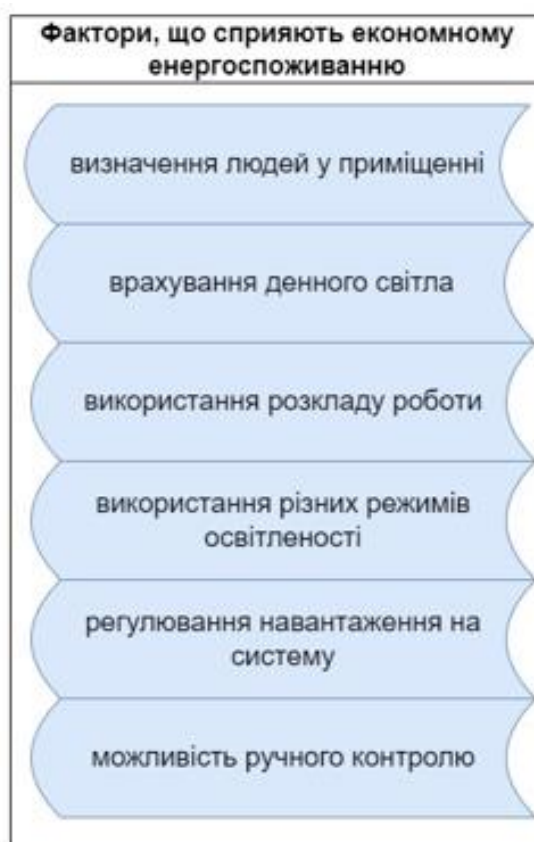


Рисунок 1.4 – Фактори, що використовуються для сценаріїв освітлення

Використання датчиків руху або присутності для виявлення людей у приміщенні дозволяє автоматично вмикати або вимикати освітлення, коли люди входять або виходять. Це призведе до зменшення споживання енергії, коли приміщення не використовується, що призводить до значної економії.

Система освітлення може автоматично регулювати яскравість на основі кількості природного світла, яке проникає у приміщення. Максимальне використання денного світла знижує потребу в штучному освітленні, що сприяє економії енергії.

Програмування освітлення за розкладом, відповідно до графіку роботи приміщення або офісу призведе до оптимізації часу ввімкнення та вимкнення освітлення відповідно до робочих годин або інших запланованих подій.

Налаштування системи для роботи в різних режимах освітленості залежно від потреб (наприклад, зниження яскравості ввечері або вночі) призведе до зменшення енергоспоживання за рахунок адаптації рівня освітленості до поточних умов і потреб.

Використання технологій для автоматичного регулювання навантаження на систему освітлення залежно від кількості ввімкнених світильників і їх інтенсивності оптимізує роботу системи, що дозволить зменшити витрати на електроенергію.

Надання користувачам можливості вручну керувати освітленням через додатки, пульти або інші інтерфейси підвищує гнучкість та зручність використання системи, що дозволяє додатково знижувати споживання енергії відповідно до потреб користувачів.

Комплексне чи хоча б часткове впровадження вказаних факторів у системи освітлення дозволяє значно знизити споживання енергії, підвищити ефективність роботи та покращити комфорт для користувачів. Розумні системи освітлення, що враховують ці фактори, сприяють стійкому використанню ресурсів та зниженню експлуатаційних витрат.

## РОЗДІЛ 2

### ПІДБІР ДЕТАЛЕЙ ДЛЯ АПАРАТНОЇ ЧАСТИНИ ПРОЄКТУ

#### 2.1 Джерела світла

##### 2.1.1 Світлодіоди

Світлодіод (Light Emitting Diode) – це напівпровідникове джерело світла, що має дві ніжки (контакти): катод (візуально коротший) та анод (довший контакт). Через ці електроди підключають світлодіод до електричної схеми: анод до позитивного полюса джерела живлення, а катод – до негативного.

Світло виникає завдяки руху електронів у кристалі, який виготовлений з напівпровідникового матеріалу (наприклад, з галій-арсеніду або галій-фосфіду). Колір світла залежить і від хімічного складу кристалу (таблиця 2.1), і від енергії фотонів, яка виникає коли електрони з'єднуються з електронними дірками під час подання напруги (рисунок 2.1).

Таблиця 2.1 – Матеріали та кольори світлодіодів

Колір	Матеріал	Довжина хвилі (нм)	Використання
Білий	індій-галій-нітрид + фосфор	широкий спектр	загальне освітлення, екрани
Жовтий	галій-арсенід-фосфід, галій-фосфід	570-600	індикатори, сигнальні пристрої
Зелений	галій-фосфід, галій-нітрид	500-570	індикатори, освітлення
Інфра-червоний	галій-арсенід	760-950	пульти дистанційного керування
Оранжевий	галій-арсенід-фосфід, галій-фосфід	600-630	індикатори, сигнальні пристрої
Синій	галій-нітрид, індій-галій-нітрид	450-500	освітлення, дисплеї, медичні прилади
Ультра-фіолетовий	алюміній-галій-нітрид	10-400	стерилізація, детектори, спеціалізоване освітлення
Червоний	галій-арсенід-фосфід, галій-фосфід	630-760	індикатори та дисплеї

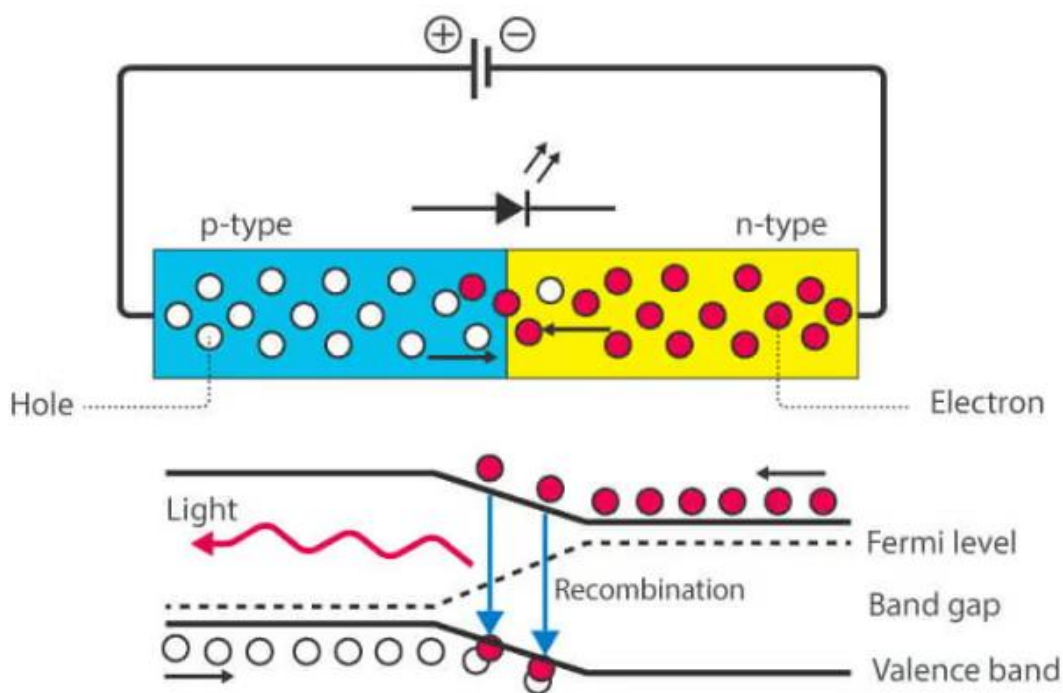


Рисунок 2.1 – Принцип роботи світлодіоду [15]

Процес випромінювання енергії у формі фотонів називається електролюмінесценцією. Колір світла залежить від ширини забороненої зони напівпровідника, яка відповідає енергії фотона.

Світлодіод виготовляється на основі звичайного діода, тому його структура містить базові елементи діода, а також інші елементи для розширеного функціоналу. Схематично будову світлодіода зображено на рисунку 2.2.



Рисунок 2.2 – Будова світлодіоду [13]

Світлодіод зазвичай розміщено в пластиковому або епоксидному чи скляному корпусі, який захищає внутрішні компоненти, а також сприяє фокусуванню світла. Деякі світлодіоди оснащені вбудованою лінзою для покращення розподілу світла. Можуть також бути різні форми корпусів для зміни характеристик світлодіода.

Світлодіод на схемі позначається подібно до звичайного діода, але доповнюється двома стрілками, напрямленими по діагоналі від нього. Може також обводитися колом, але це не є обов'язковим (рисунок 2.3). Позначення може бути: HL, VD або LED.

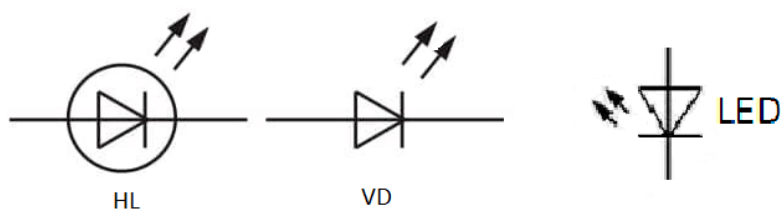


Рисунок 2.3 – Позначення на схемі

При підключенні світлодіоду обов'язково потрібно використовувати струмообмежуючий резистор (рисунок 2.4), інакше термін служби світлодіода буде в рази меншим, (інколи світлодіод може згоріти одразу після першої подачі живлення). Хоч він виділяє значно менше тепла, ніж звичайна лампа, але за відсутності додаткового навантаження можна легко досягти пікового значення струму, внаслідок чого кристал досягне стану теплового пробою. Тому робочий струм є важливим параметром світлодіоду. Світлодіоди працюють при низькій напрузі (від 2-3 В для червоних до 3-4 В для синіх та білих).

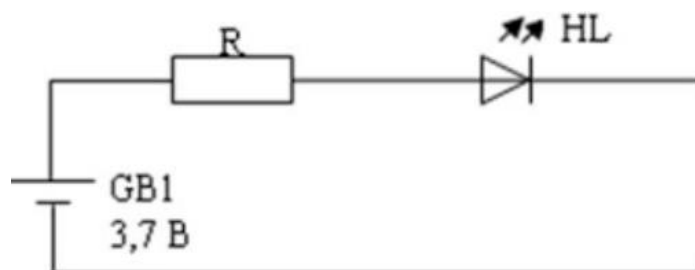


Рисунок 2.4 – Типова схема підключення світлодіоду

Для розрахунку опору резистора ( $R$ ) при приєднанні світлодіода необхідно знати напругу живлення ( $V_P$ ), напругу випередження світлодіода ( $V_f$ ) і, як вже зазначалося, максимально допустимий струм ( $I_f$ ) для світлодіода. Знаючи ці параметри можемо розрахувати опір резистора за законом Ома:

$$R = \frac{V_P - V_{LED}}{I_{LED}} \quad (2.1)$$

де  $V_P$  – напруга живлення (В);

$V_{LED}$  – напруга випередження світлодіода (В);

$I_{LED}$  – допустимий струм через світлодіод (А).

Важливо також врахувати потужність ( $P$ ) резистора, яку розраховуємо за формулою:

$$P = (V_P - V_{LED}) \cdot I_{LED} \quad (2.2)$$

Резистор обираємо з опором та потужністю не меншими за обчислені значення за формулами (2.1)-(2.2).

Світлодіоди зручно використовувати в схемах з мікроконтролерами. Для цього існує кілька способів підключення:

– анод (+) підключений до цифрового контакту (наприклад, D13) на платі мікроконтролера, катод (-) – до GND через резистор;

– катод (-) підключений до цифрового контакту на платі мікроконтролера, анод (+) – до живлення 5 В через резистор.

Приклад першого варіанту підключення світлодіода наведено на рисунку 2.5.

Варіанти (правильні та не правильні) під'єднання світлодіода наведено в таблиці 2.2. Тут також наведено стан світлодіода залежно від способу підключення живлення до його контактів.

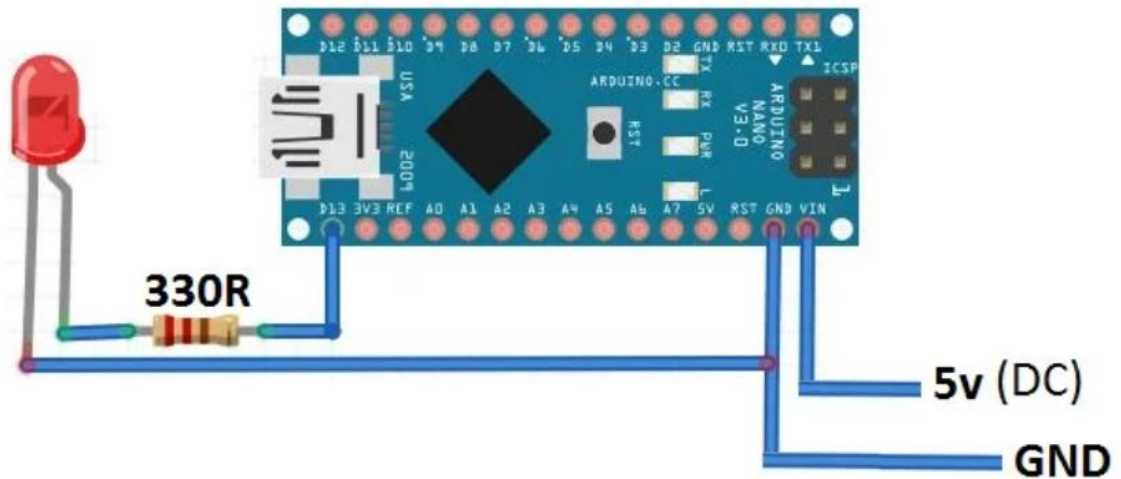


Рисунок 2.5 – Один із варіантів підключення світлодіоду до плати мікроконтролера

Таблиця 2.2 – Способи під'єднання світлодіоду

Катод (-)	Анод (+)	Резистор	Стан світлодіоду
GND	VCC	так	увімкнено
GND	PWM	так	увімкнено, можна керувати яскравістю
GND	GND	ні	вимкнено
VCC	VCC	ні	вимкнено
VCC	GND	ні	вийшов з ладу

Після приєднання світлодіоду (див. рисунок 2.5) нам потрібно налаштувати (ініціалізувати в програмі) контакт плати як цифровий вихід за допомогою функції `pinMode()`, наприклад: `pinMode(LED_WHT_D13, OUTPUT)`.

Щоб вимкнути світлодіод потрібно запрограмувати його контакт на GND за допомогою функції `digitalWrite()`, наприклад `digitalWrite(LED_WHT_D0, LOW)`.

Щоб увімкнути світлодіод, потрібно запрограмувати його контакт на VCC за допомогою функції `digitalWrite()`, наприклад `digitalWrite(LED_WHT_D0, HIGH)`.

Особливим видом світлодіоду є RGB-світлодіод, який по своїй суті містить три окремі світлодіоди: червоний (Red), зелений (Green) та синій (Blue), що об'єднані в одному корпусі. Кожним з них можна керувати окремо, змінюючи їх

яскравість. Це дає можливість отримувати широкий спектр кольорів завдяки змішуванню основних трьох. RGB-світлодіоди бувають зі спільним анодом або зі спільним катодом, вони зазвичай мають 4 піни: один загальний і три для кожного кольору (R, G, B). Анод чи катод у RGB-світлодіода завжди має найдовшу ніжку.

При підключенні до плати мікроконтролера світлодіода зі спільним анодом, його (анод) потрібно під'єднати до позитивного полюсу джерела живлення, а катоди світлодіода – до трьох цифрових пінів плати мікроконтролера, не забуваючи про опір.

При підключенні до плати мікроконтролера світлодіода зі спільним катодом, його (катод) потрібно заземлити (під'єднати до GND), а аноди світлодіода – до трьох цифрових виходів плати мікроконтролера через резистори (рисунок 2.6).

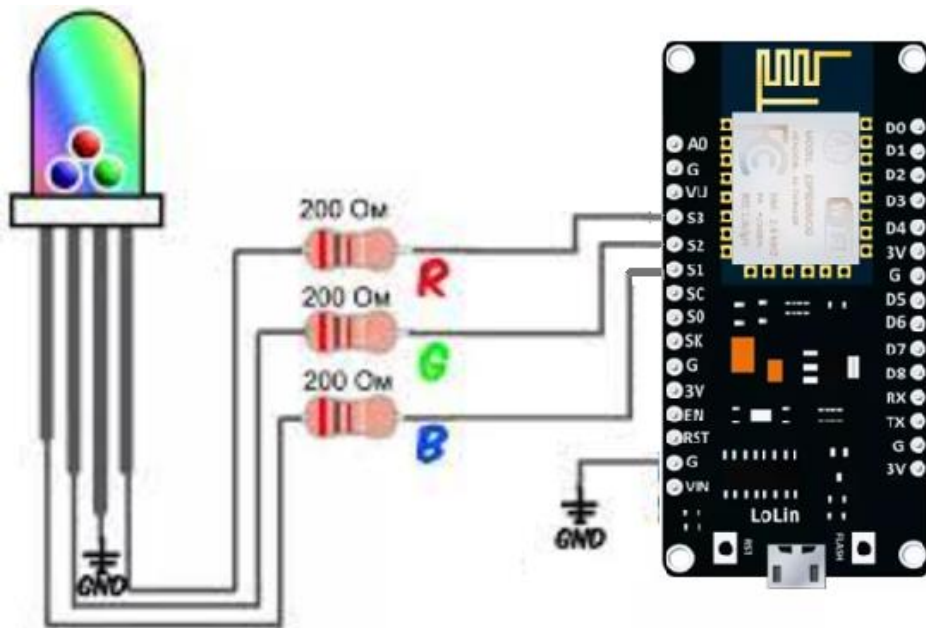


Рисунок 2.6 – Схема підключення для RGB-світлодіода з спільним катодом

Для розрахунку резисторів для кожного кольору, використовуємо формулу (2.1). Наприклад, якщо напруга живлення  $V_p$  становить 5 В (більшість плат мікроконтролерів), напруга випередження  $V_{LED}$  2 В (для червоного

світлодіода), максимально допустимий струм  $I_{LED}$  0.02 А, то підставивши ці значення в формулу (2.1) отримаємо:

$$R = \frac{V_P - V_{LED}}{I_{LED}} = \frac{5 - 2}{0,02} = \frac{3}{0,02} = 150 \text{ Ом.}$$

Тому варто обрати резистор, що має не менший опір, тобто 180 або 200 Ом.

При програмуванні роботи з світлодіодом використовують додаткову функцію setColor (рисунок 2.7), яка встановлює рівні яскравості для кожного кольору світлодіода за допомогою ШІМ (PWM).

```
// функція для встановлення кольору RGB-світлодіода
void setColor(int redValue, int greenValue, int blueValue) {
    analogWrite(redPin, redValue);
    analogWrite(greenPin, greenValue);
    analogWrite(bluePin, blueValue);
}
```

Рисунок 2.7 – Функція для встановлення кольору світлодіода

Щоб вимкнути світлодіод – потрібно задати всі три параметри цієї функції рівними нулю, щоб засвітити білим кольором – всі параметри 255, зелений колір можна отримати шляхом передачі в функцію параметрів 0, 255, 0.

Інші види світлодіодів:

- DIP світлодіоди мають малий кут розсіювання (до 60 °), різні кольори;
- Super Flux «Piranha» дає пучок білого світла, широкий кут розсіювання (40 ° – 120 °), має підвищену яскравість, можна вибрати температуру світіння, колір лише білий;
- Straw Hat – світлодіод з лінзою, близько до якої розташовано кристал, за рахунок цього зменшена висота світлодіода та збільшено кут розсіювання (100 ° – 140 °);
- SMD-світлодіоди мають широкий спектр кольорів та хорошу яскравість, проте головною їх перевагою є розмір (його корпус може бути навіть 1 мм і меншим). Такі світлодіоди використовують в сучасних освітлювальних приладах, в тому числі в розумних лампах.

### 2.1.2 Розумні лампочки

Розумна лампочка – це інтегроване освітлювальне рішення у вигляді пристрою, який поєднує в собі сучасні досягнення у сфері світлодіодного освітлення та бездротових технологій. Вона дає можливість створювати комфортні умови освітлення (керувати кольором, яскравістю та іншими характеристиками світла за допомогою мобільних додатків, пультів, голосових команд або ж автоматичних сценаріїв) та економити енергію, що є дуже важливо в наш час.

Вона складається з кількох основних компонентів:

- корпус для забезпечення захисту внутрішніх компонентів та їх належного охолодження. З метою уніфікації та стандартизації його зовнішня форма часто відповідає стандартним розмірам звичайних лампочок;
- світлодіодний модуль (LED-модуль) забезпечує високий рівень енергоефективності та довгий термін служби;
- драйвер світлодіодів який контролює подачу струму на світлодіоди, дає можливість ним стабільно працювати, а також регулювати яскравість, тут інтегровано схеми для управління напругою та струмом, який подається на світлодіоди;
- вбудований мікроконтролер керує всіма функціями лампочки, здійснює обробку сигналів від бездротових модулів, керування світлодіодами та взаємодією з користувачем;
- бездротовий модуль виконує функції зв'язку з іншими пристроями та мережами. Зазвичай цей модуль підтримує такі стандарти, як Wi-Fi, Bluetooth, Zigbee або Z-Wave і саме він дає змогу керувати лампочкою за допомогою мобільних додатків чи голосових асистентів;
- деякі розумні лампочки можуть мати вбудовані сенсори руху, освітленості чи температури. Такі сенсори дають можливість автоматично регулювати освітлення залежно від умов навколишнього середовища чи присутності людей;

– блок живлення забезпечує напругу та струм, необхідні для роботи всіх компонентів. Зазвичай блок живлення містить інтегровані схеми перетворення змінного струму (AC) в постійний (DC), які забезпечують стабільне живлення для світлодіодів та електроніки.

Функціональна схема підключення компонентів типової розумної лампи наведена на рисунку 2.8.

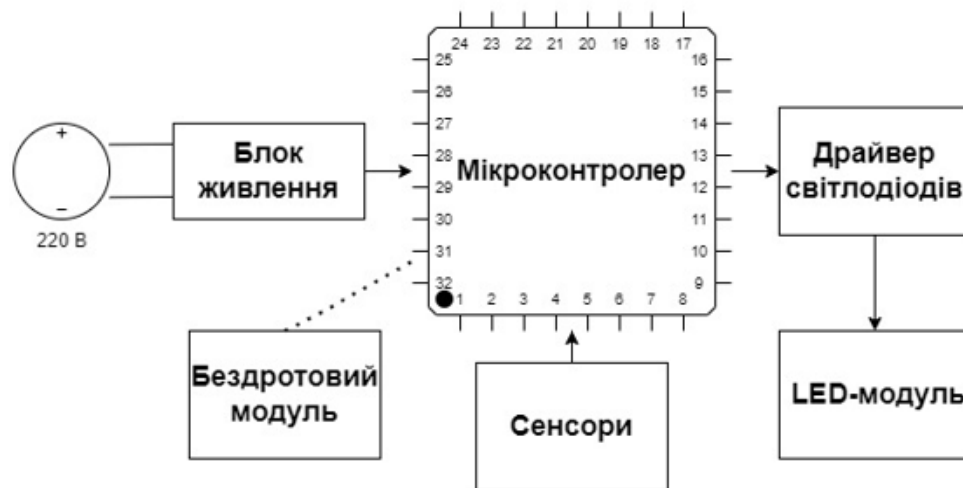


Рисунок 2.8 – Схема компонентів типової розумної лампи

Блок живлення підключають до мережі змінного струму (наприклад, 220В або 110В).

Мікроконтролер отримує живлення від блоку живлення і контролює роботу драйвера світлодіодів.

Світлодіодний модуль підключається до виходу драйвера світлодіодів, який регулює яскравість та колір світіння.

Бездротовий модуль зв'язується з мікроконтролером та забезпечує комунікацію з зовнішніми пристроями.

Сенсори (якщо вони є) підключаються до мікроконтролера для збору даних про навколишнє середовище.

Взаємодія лампи з користувачем може здійснюватися через мобільні додатки (через Wi-Fi або Bluetooth), за допомогою голосових асистентів (наприклад, Amazon Alexa, Google Assistant чи Apple HomeKit). Такі лампочки

можуть бути інтегровані в системи розумного дому, які дозволяють створювати незалежні автоматизовані сценарії залежно від часу доби, присутності людей чи інших параметрів.

Переваги розумних лампочок:

- завдяки використанню світлодіодів вони споживають значно менше енергії порівняно з традиційними лампами;
- можливість керування яскравістю та кольором світла через додатки або голосові команди;
- підтримка автоматизованих сценаріїв для створення комфортних умов освітлення;
- можливість інтеграції з іншими пристроями розумного дому, і, як наслідок, підвищення рівня зручності та безпеки.

## 2.2 Сенсор освітленості

Сенсор освітленості – це фотоелектричний пристрій, який вимірює рівень освітленості (або світлову енергію – фотони) в навколишньому середовищі і перетворює його на електричний сигнал (електрони), який може бути використаний для автоматичного регулювання освітлення або інших систем [10]. Існують різні типи датчиків світла, основні з них: фоторезистори, фотодіоди та фототранзистори (рисунок 2.9).

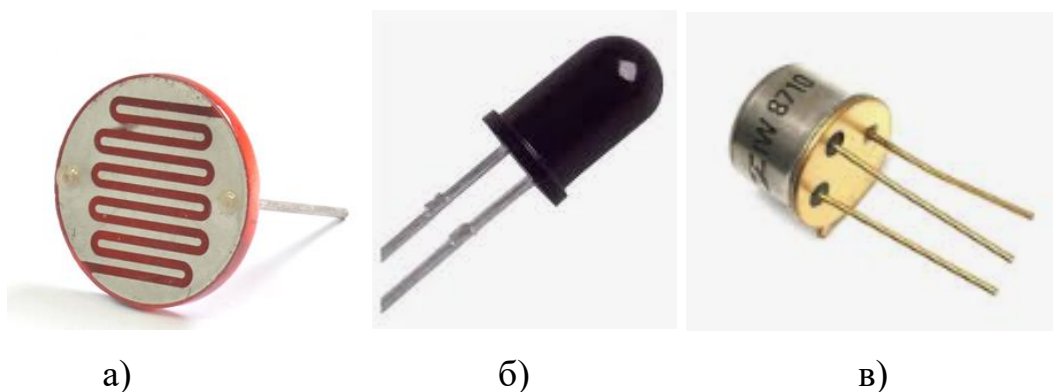


Рисунок 2.9 – Сенсори освітленості: а) – фоторезистор;  
б) – фотодіод; в) – фототранзистор

Найпоширенішим типом сенсора світла, який використовується, є фоторезистор, також відомий як світлозалежний резистор (Light Dependent Resistor – LDR). Фоторезистори виготовлені з напівпровідникового матеріалу з високим опором, який дуже чутливий до видимого світла.

Він працює так само, як звичайний резистор, але зміна опору залежить від кількості світла, яке падає на його чутливий елемент. Висока інтенсивність світла призведе до нижчого опору між елементами з сульфїду кадмію, тоді як низька інтенсивність світла призводить до більш високого опору між цими елементами.

LDR може використовуватися не лише для виявлення світла, але й для вимірювання його яскравості (вимірюючи опір можна знайти яскравість). Фоторезистор має два контакти, які є симетричними (їх не потрібно розрізняти).

Фотодіод (фотодетектор або фотодатчик) – ще один тип сенсора освітленості, який генерує струм пропорційно до інтенсивності світла, що падає на нього. Фотодіоди в основному виготовляються з кремнію та германію та містять оптичні фільтри, вбудовані лінзи та поверхні.

Фотодіоди працюють за принципом роботи, який називається внутрішнім фотоефектом; коли промінь світла потрапляє на сенсор, його електрони послаблюються, викликаючи електронні дірки, через які протікає електричний струм. Чим яскравіше світло, тим сильнішим буде електричний струм. Фотодіоди мають високу швидкість реакції та хорошу лінійність, але для їх використання потрібен підсилювач сигналу.

Фотодіоди дуже чутливі до інфрачервоного світла, тому часто використовуються в проєктах з пристроями дистанційного керування, сонячних панелях, медицині.

Фототранзистор працює як звичайний транзистор, але керується світлом замість електричного сигналу. Фототранзисторний датчик світла можна описати як фотодіод плюс підсилювач. Його перевагами є висока чутливість та швидка реакція, а недолік: потреба в калібруванні.

В промислових проєктах також використовують цифрові сенсори освітленості (наприклад, TSL2561, BH1750).

Цифрові сенсори освітленості містять фотодіоди або фототранзистори разом з вбудованою електронікою для перетворення світла у цифровий сигнал. Вони мають переваги: висока точність, зручний цифровий інтерфейс (наприклад, I2C або SPI), але разом з тим вища вартість порівняно з аналоговими сенсорами.

### **2.3 Інфрачервоний приймач**

Інфрачервоний (ІЧ) приймач (IR Receiver) – це електронний компонент, який приймає інфрачервоні сигнали, зазвичай від пульта дистанційного керування, та перетворює їх на електричні сигнали, які зрозумілі мікроконтролерам чи іншим електронним пристроям.

Інфрачервоні приймачі працюють на принципі детектування ІЧ-випромінювання, яке модулюється з певною частотою. Ці сигнали потім демодулюються і перетворюються на цифровий сигнал (0 і 1), який може бути оброблений мікроконтролером [3].

Основні компоненти ІЧ-приймача:

- інфрачервоний детектор – фотодіод або фототранзистор, який приймає ІЧ-сигнали;
- фільтр частот відсікає сигнали, що не відповідають заданій частоті модуляції (зазвичай 36-38 кГц);
- підсилювач підсилює сигнал для подальшої обробки;
- демодулятор відділяє корисний сигнал від несучої частоти.

ІЧ приймач має три контакти, для його підключення до мікроконтролера потрібно вихід VCC приймача підключити до живлення 5 В мікроконтролера, GND приймача – до GND мікроконтролера, а OUT приймача підключається до цифрового входу мікроконтролера (наприклад, D11) через резистор (рисунок 2.10).

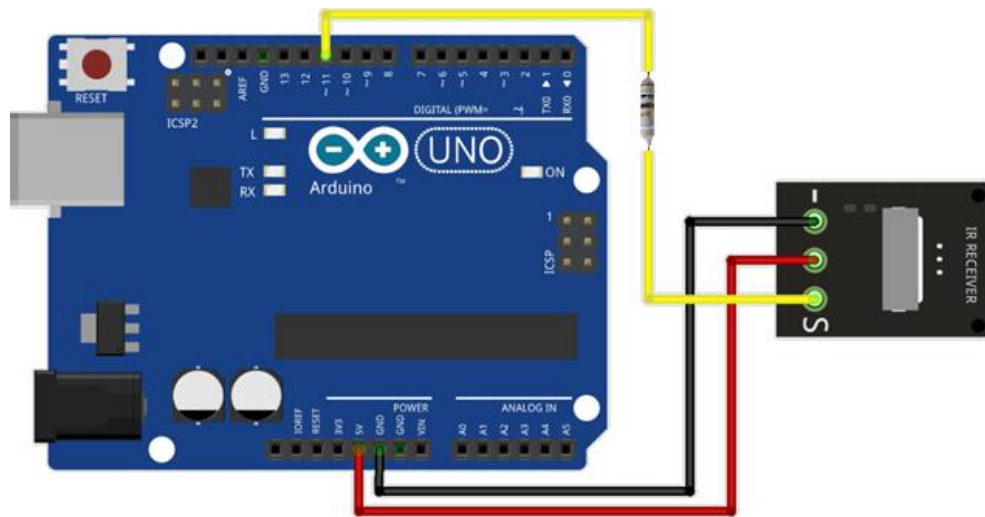


Рисунок 2.10 – Схема приєднання ІЧ-приймача до мікроконтролера

ІЧ приймачі часто використовуються для дистанційного керування системами освітлення. Вони дозволяють: вмикати та вимикати світло, регулювати яскравість, змінювати кольори світлодіодів (якщо це RGB-світлодіоди) або ж запускати попередньо налаштовані сценарії освітлення.

Існують різні протоколи передачі ІЧ сигналів, наприклад, NEC, Sony, RC5 тощо. Кожен протокол має свій спосіб модуляції сигналу та кодування команд.

ІЧ приймачі є важливою частиною систем дистанційного управління, в тому числі систем розумного освітлення. Вони забезпечують зручний спосіб керування пристроями на відстані, що дозволяє створювати більш інтерактивні та функціональні системи освітлення.

## 2.4 Вибір мікроконтролера

При проектуванні цифрової системи її основним елементом є мікроконтролер, тому важливо його правильно обрати, при цьому слід враховувати вимоги до системи і намагатися знизити її загальну вартість. Одними з найпопулярніших мікроконтролерів на даний час є мікроконтролери Arduino та ESP. Для вибору оптимального мікроконтролера складемо таблицю порівняння їх основних характеристик (таблиця 2.3).

Таблиця 2.3 – Порівняння популярних мікроконтролерів

Характеристика	Arduino			ESP		
	Uno	Nano	Mega	ESP8266	ESP32	
Процесор	ATmega328P	ATmega328P	ATmega2560	Tensilica L106 32-bit	Tensilica Xtensa 32-bit LX6	
Частота	16 MHz	16 MHz	16 MHz	80/160 MHz	до 240 MHz	
Пам'ять	Flash	32 KB	32 KB	256 KB	4 MB	до 16 MB
	SRAM	2 KB	2 KB	8 KB	160 KB	до 520 KB
	EEPROM	1 KB	1 KB	4 KB	-	-
Цифрові входи/виходи	14 (6 з них PWM)	14 (6 з них PWM)	54 (15 з них PWM)	17	до 36	
Аналогові входи	6	8	16	1	до 18	
Примітка	популярний, простий у використанні, багато навчальних матеріалів	компактний розмір, той самий процесор, що й у Uno, більша кількість аналогових входів	більше I/O пінів та пам'яті, підходить для великих проєктів з багатьма датчиками та виконавчими механізмами	вбудований Wi-Fi	вбудований Wi-Fi та Bluetooth, висока продуктивність, підходить для IoT проєктів	

Проаналізувавши вимоги до системи освітлення та характеристики плат, було вирішено обрати мікроконтролер Arduino Nano.

По-перше, він використовує той самий процесор (ATmega328P), що й Uno, тим самим забезпечує однакову продуктивність, але в більш компактному форм-факторі (45 мм x 18 мм), що дає змогу легко інтегрувати його в невеликі корпуси чи в проєкти з обмеженим простором.

По-друге, Nano має 14 цифрових входів/виходів і 8 аналогових входів, що достатньо для нашого проєкту розумного освітлення.

По-третє, Nano споживає менше енергії порівняно з більшими платами, що важливо для проєктів з автономним живленням або де необхідно мінімізувати енергоспоживання.

Крім того, Arduino Nano має велику спільноту користувачів, багато доступних навчальних матеріалів та бібліотек, що полегшить процес розробки та налагодження.

Тому Arduino Nano (рисунок 2.11) є оптимальним вибором для нашого проєкту, він дасть можливість легко інтегрувати різні сенсори та виконавчі механізми, а також забезпечить надійну роботу системи в компактному та зручному форм-факторі.

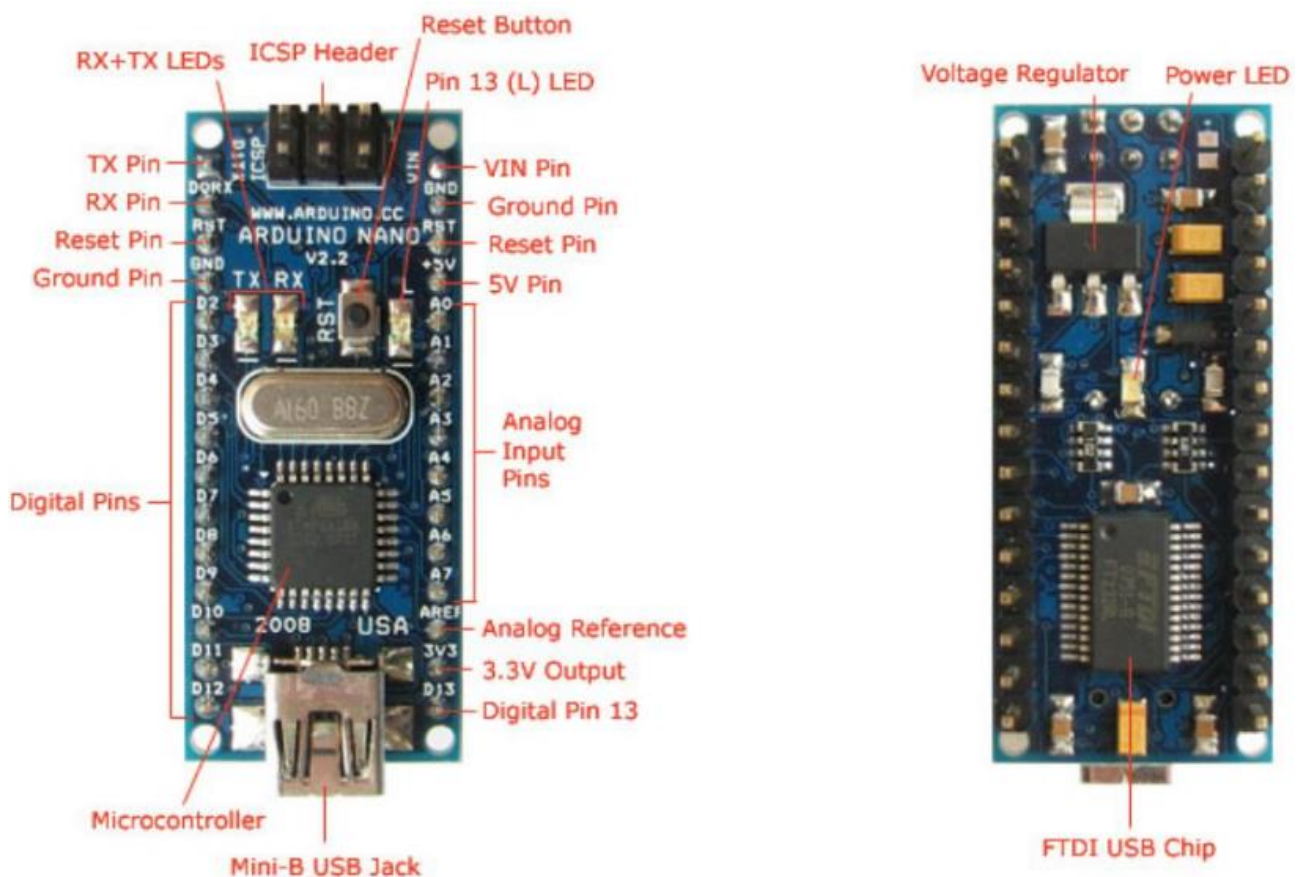


Рисунок 2.11 – Зовнішній вигляд Arduino Nano

Плати Arduino містять всі необхідні компоненти для розробки і реалізації широкого спектру електронних проектів:

- мікроконтролер – центральний процесор плати (для Nano це ATmega328);
- піни для підключення цифрових датчиків і пристроїв (14 цифрових пінів на Arduino Nano);
- піни для зчитування аналогових сигналів від датчиків (8 аналогових пінів на Arduino Nano);
- цифрові піни, які можуть забезпечувати широтно-імпульсну модуляцію (PWM) для регулювання яскравості світлодіодів, керування сервоприводами і т.д. (6 PWM виходів на Arduino Nano);
- роз'єм для підключення зовнішнього джерела живлення (зазвичай від 7 В до 12 В);
- USB порт, який використовується для програмування плати і для живлення від комп'ютера або повербанку;
- роз'єми та гнізда для підключення проводів і модулів, які дозволяють легко з'єднувати різні компоненти з платою;
- кварцовий резонатор забезпечує стабільну тактову частоту для мікроконтролера (16 MHz на Arduino Nano);
- перемикач живлення дає можливість робити вибір між джерелом живлення від USB або зовнішнього адаптера;
- індикаторні світлодіоди можуть використовуватися для LED-індикації стану живлення, є також вбудований світлодіод, підключений до цифрового піну 13 (на більшості плат Arduino);
- нульовий резистор захищає мікроконтролер від неправильного підключення або короткого замикання;
- енергонезалежна пам'ять (EEPROM) для зберігання даних між перезавантаженнями (1 KB на Arduino Nano);
- флеш-пам'ять для зберігання програмного коду (32 KB на Arduino Nano);

- оперативна пам'ять (SRAM) для виконання програм (2 КВ на Arduino Nano);
- штифти (headers) для підключення додаткових модулів і плат розширення (шилдів);
- тактова частота Arduino Nano становить 16 МГц, що означає виконання приблизно 300 000 рядків вихідного коду С за секунду.

Входи-виходи плати Arduino Nano наведено на рисунку 2.12, а входи та виходи процесора АТmega328 – на рисунку 2.13.

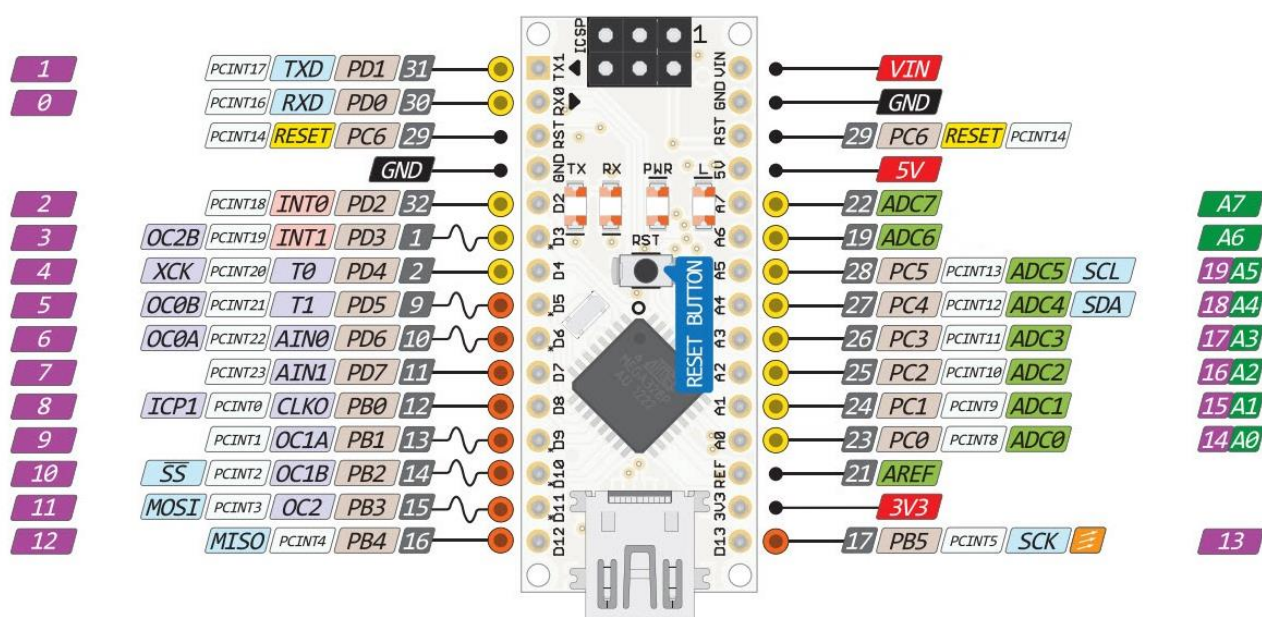


Рисунок 2.12 – Роз'єми (піни) Arduino Nano

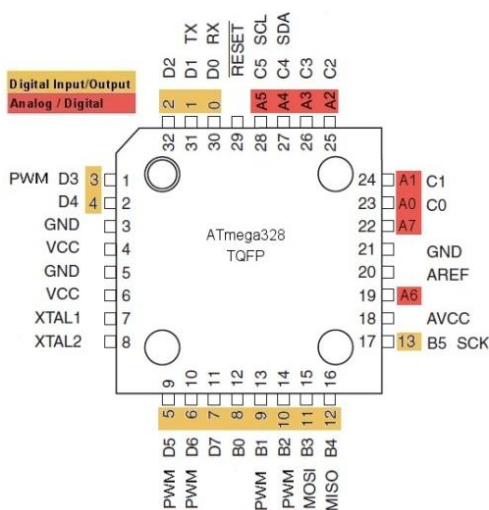


Рисунок 2.13 – Роз'єми (піни) АТmega328

## РОЗДІЛ 3

### ПРОЄКТУВАННЯ ТА ТЕСТУВАННЯ СИСТЕМИ ОСВІТЛЕННЯ

#### 3.1 Схема та монтаж пристрою

Найпростішою формою системи управління освітленням є один ручний перемикач для включення та вимкнення світла. Проте автоматизована система освітлення є системою з більш складною структурою та функціями. Метою такої системи є: підтримка необхідного рівня яскравості в залежності від часу та погодних умов; мінімізація енергоспоживання; надання можливості створення користувацьких сценаріїв для підтримки заданого рівня освітлення.

Автоматична системи управління освітленням складається як мінімум з двох приладів: виконавчий пристрій (джерело світла) та керуючий пристрій (контролер). На рисунку 3.1 структура такої системи з одним контролером, який містить вбудований алгоритм управління освітленням.

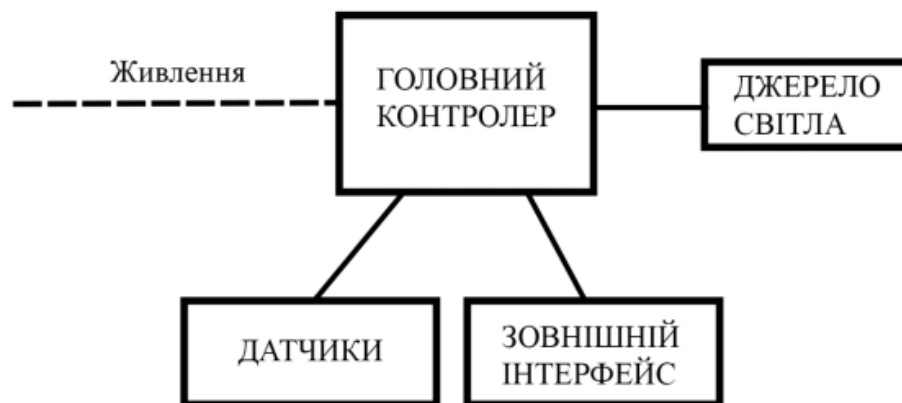


Рисунок 3.1 – Схема типової системи управління освітленням

До контролера підключаються інші пристрої (наприклад, сенсор для визначення рівня освітленості або сенсор знаходження людей у приміщенні). Контролер повинен мати зовнішній інтерфейс для взаємодії користувача з системою. Це може бути програма, кнопка, сенсорний дисплей або пульт.

Контролер, на основі даних датчиків та даних отриманих від користувача, надсилає сигнали до приладів штучного освітлення.

Після огляду літератури та підбору елементів для системи освітлення з багатофакторним керуванням було поставлено завдання: зібрати прототип системи освітлення на основі мікроконтролера Arduino Nano. Вимоги до системи:

- можливість дистанційного керування (пультом);
- економне енерговикористання та автоматичне вимкнення, якщо в приміщенні достатнє освітлення;

- не дорогі компоненти;
- надійність та довговічність.

Для реалізації проєкту було обрано компоненти:

- мікроконтролер Arduino Nano;
- фоторезистор із опором при номінальному освітленні 3,3 кОм (KY-018 PHOTORESISTOR MODULE або GL5547);
- інфрачервоний приймач (ресівер VS1838B);
- сім світлодіодів різних кольорів, які можуть змінювати яскравість освітлення;
- резистори, провідники, макетна плата або корпус пристрою;
- джерело живлення (для Arduino, ламп та пультів керування).

Потрібно поєднати ці всі компоненти в один проєкт та написати програму (скетч) для Arduino, яка керуватиме світлодіодами залежно від рівня освітленості та команд з інфрачервоних пультів.

Структурна схема проєкту (рисунок 3.2) відображає статичну організацію нашої системи освітлення, тобто як один об'єкт пов'язаний з іншим. Тобто структурна схема відображає склад та логічні зв'язки в системі, але для правильного поєднання електронних компонентів цього не достатньо. Потрібно розробити ще схему електричну структурну, враховуючи контакти складових частин проєкту (рисунок 3.3).

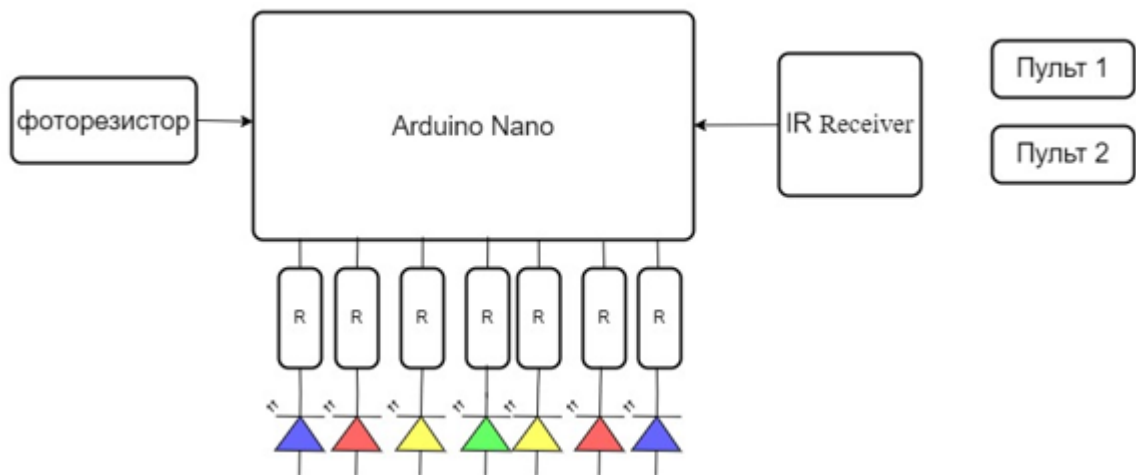


Рисунок 3.2 – Система розумного освітлення з багатофакторним керуванням.

Схема структурна

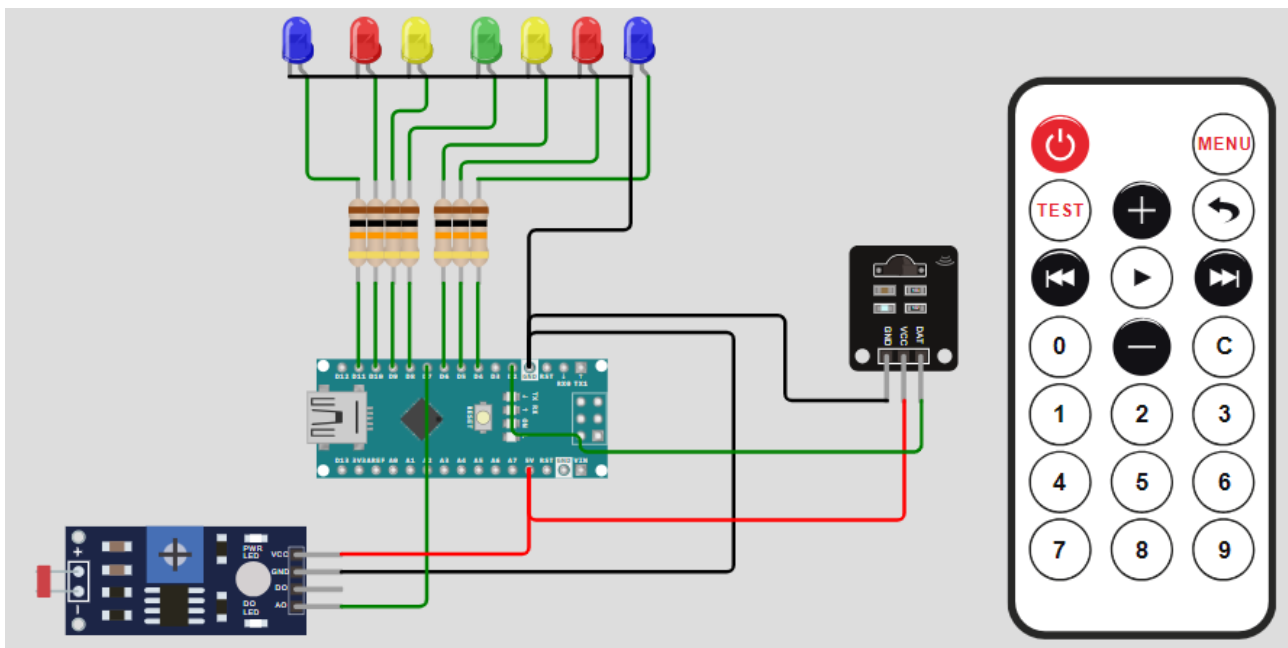


Рисунок 3.3 – Система розумного освітлення з багатофакторним керуванням.

Структурна електрична схема

Як видно з рисунку 3.3, центральним компонентом схеми є мікроконтролер Arduino Nano, який здійснює управління всією системою. До цифрових виходів Arduino (D3, D4, D5, D8, D9, D10, D11) через резистори для обмеження струму під'єднано 7 світлодіодів. Інфрачервоний приймач підключений до цифрового

виходу Arduino (D2) для прийому сигналів від інфрачервоних пультів. Сенсор освітленості (день/ніч) під'єднано до аналогового входу Arduino A7.

Функціональність спроектованої системи:

- інфрачервоний приймач отримує сигнали від пультів для керування світлодіодами;
- сенсор освітленості дозволяє системі автоматично регулювати яскравість світлодіодів залежно від рівня зовнішнього освітлення;
- Arduino Nano обробляє сигнали і контролює включення/виключення та яскравість світлодіодів.

Ця система дозволяє реалізувати багатофакторне керування освітленням з використанням як ручного (пульту), так і автоматичного (датчик освітленості) режимів керування.

Готовий пристрій, спроектований за наведеною вище схемою, наведено на рисунку 3.4, роль автономного джерела живлення виконують 3 сольові АА батарейки (рисунок 3.5).

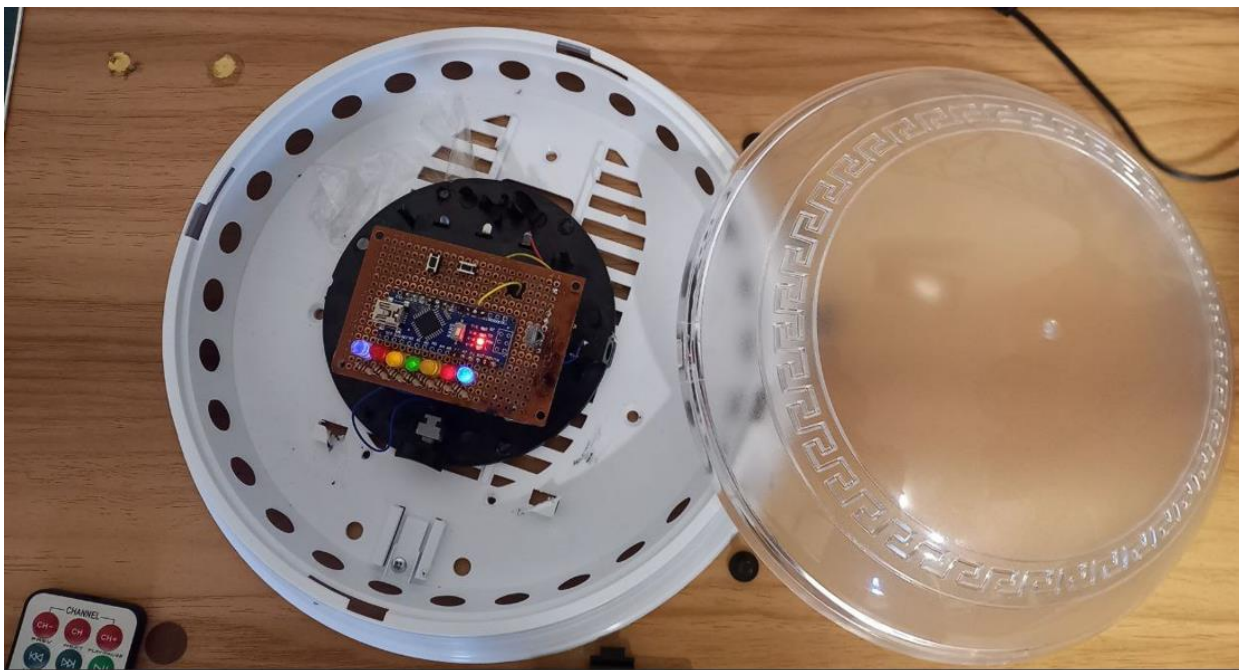


Рисунок 3.4 – Зовнішній вигляд пристрою

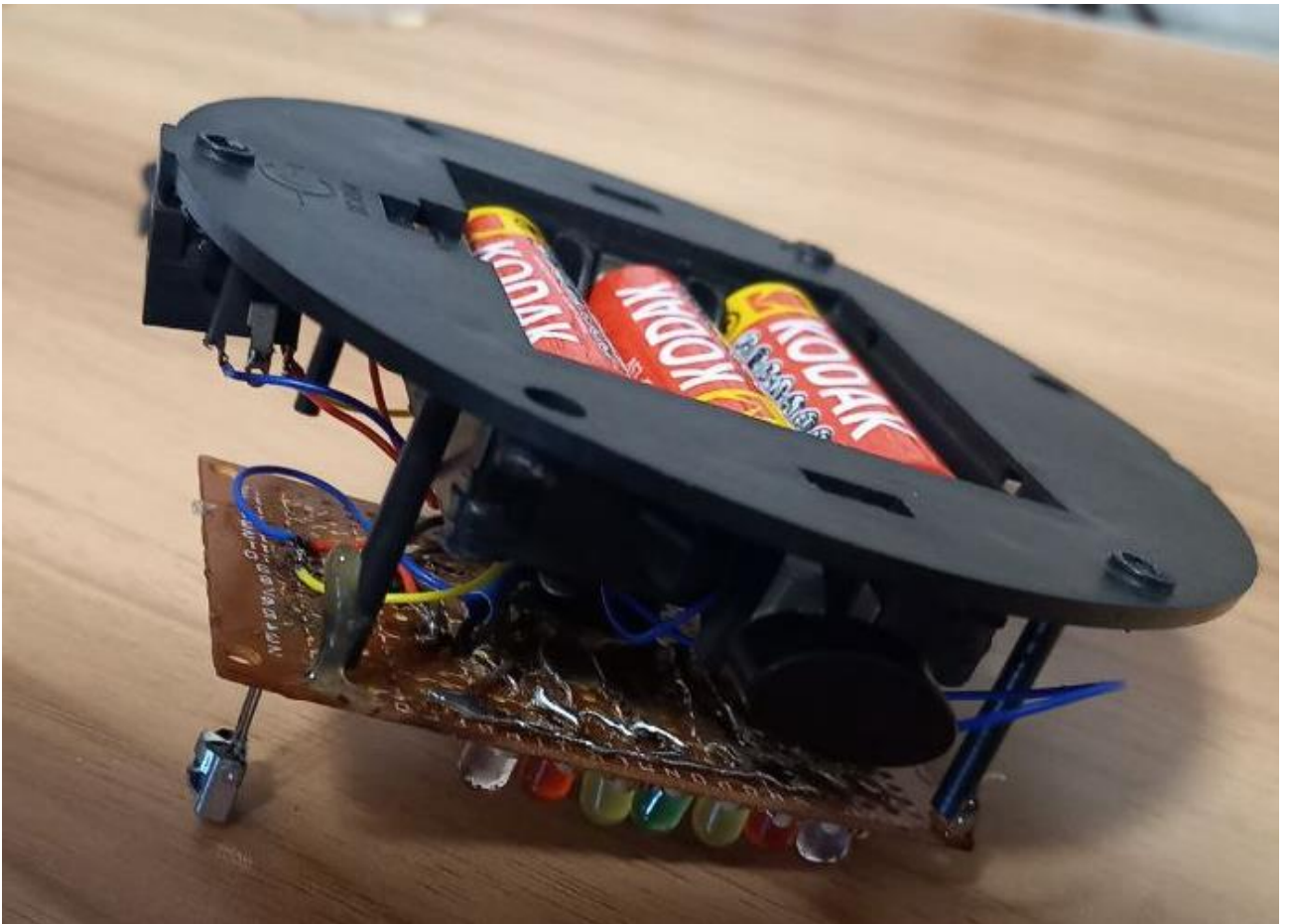


Рисунок 3.5 – Автономне джерело живлення пристрою

Живлення пультів також автономне:

- стандартний IR пульт для Arduino (MP3) – батарейка-таблетка CR2032 (може бути CR2025);
- універсальний пульт Hуау – дві АА батарейки.

В проєкті використано два пульти, бо вони мають різний радіус дії (в Hуау набагато потужніший діод, який випромінює інфрачервоне світло).

Окрім автономного блоку живлення також можна використовувати для під'єднання альтернативного джерела живлення роз'єм Mini-B USB, який призначений для прошивки мікроконтролера. Його, як і додаткове гніздо живлення DC-005 (рисунок 3.6) можна приєднати до USB-порта ноутбука або повербанка відповідними кабелями. На рисунку 3.6 також видно інфрачервоний приймач, який передає сигнали керування з пультів на плату мікроконтролера,

та фоторезистор, який керує ввімкненням або вимкненням освітлення залежно від світлового потоку.

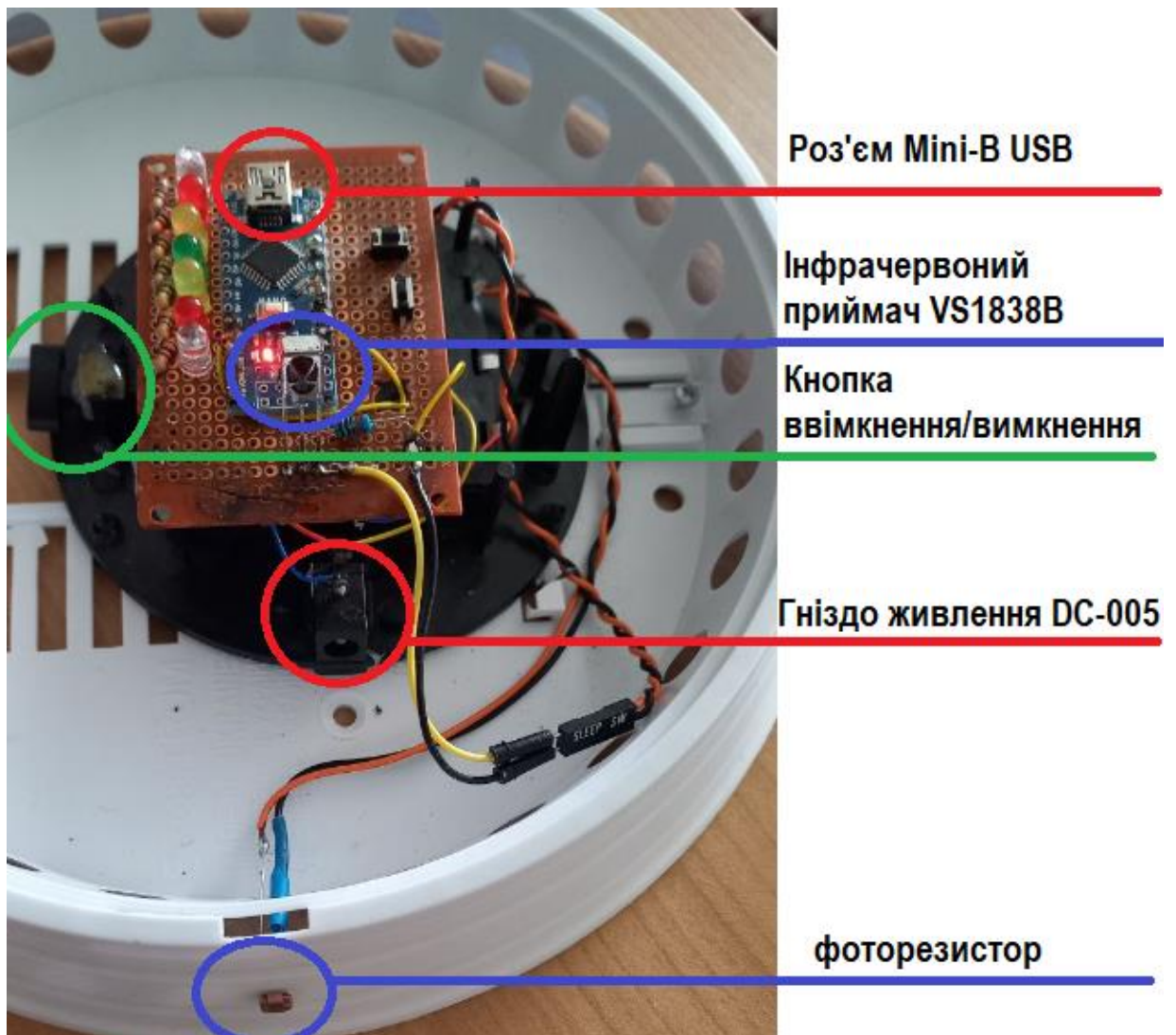


Рисунок 3.6 – Альтернативні джерела живлення пристрою

### 3.2 Програмування мікроконтролера

Після приєднання пристрою до ПК через інтерфейс micro USB/USB знаходимо в меню Інструменти/ Плата «Arduino Uno»/ Менеджер плат, та замість «Arduino Uno» вибираємо «Arduino nano», а в полі «Processor: ATmega 328P» – «ATmega 328P (Old Bootloader)» (рисунок 3.7), далі виберемо порт (в нашому випадку це COM3), перевіримо та завантажимо скетч на плату.

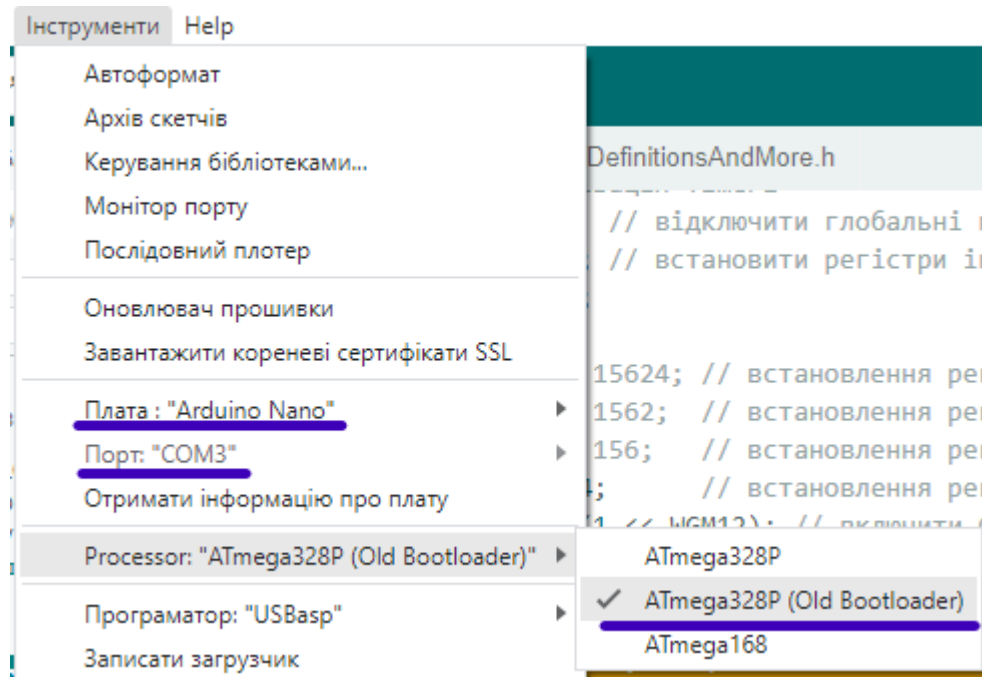


Рисунок 3.7 – Менеджер плат в середовищі Arduino IDE

Алгоритм роботи з мікроконтролером має кілька кроків:

– визначення констант відповідно до апаратної логіки проєкту (рисунки 3.8-3.9);

```

48  /* константи світлодіодів : номер - назва порта - тип IO - колір підключеного діода */
49  const int LED_WHT_D0 = 3; // 6 D3 I03 синій
50  const int LED_RED_D1 = 4; // 7 D4 I04 червоний
51  const int LED_YLW_D2 = 5; // 8 D5 I05 жовтий
52  const int LED_GRN_D3 = 8; // 11 D8 I08 зелений
53  const int LED_YLW_D4 = 9; // 12 D9 I09 жовтий
54  const int LED_RED_D5 = 10; // 13 D10 I010 червоний
55  const int LED_WHT_D6 = 11; // 14 D11 I011 синій

```

Рисунок 3.8 – Константи для світлодіодів

```

59  const int KEY_0 = 22; //
60  const int KEY_1 = 12; //
61  const int KEY_2 = 24; //
62  const int KEY_3 = 94; //
63  const int KEY_4 = 8; //
64  const int KEY_5 = 28; //
65  const int KEY_6 = 90; //
66  const int KEY_7 = 66; //
67  const int KEY_8 = 82; //
68  const int KEY_9 = 74; //
69
70  const int KEY_MINUS = 7; //
71  const int KEY_PLUS = 21; //
72  const int KEY_100 = 25; //
73  const int KEY_200 = 13; //
74  const int KEY_EQ = 9; //

```

Рисунок 3.9 – Константи для стандартного IR-пульта для Arduino

– ініціалізація портів, до яких приєднано світлодіоди, сенсор та інфрачервоний приймач (рисунок 3.10);

```

105     pinMode(LED_WHT_D0, OUTPUT);
106     pinMode(LED_RED_D1, OUTPUT);
107     pinMode(LED_YLW_D2, OUTPUT);
108     pinMode(LED_GRN_D3, OUTPUT);
109     pinMode(LED_YLW_D4, OUTPUT);
110     pinMode(LED_RED_D5, OUTPUT);
111     pinMode(LED_WHT_D6, OUTPUT);

```

Рисунок 3.10 – Ініціалізація портів, до яких приєднано світлодіоди

– основний цикл (`loop()`), в якому з заданою частотою перевіряється наявність сигналів від інфрачервоного приймача, при отриманні сигналу викликається функція обробки IR-даних (`handleReceivedIRData`);

– обробка IR-сигналу (зчитувати адреси і команди від пульта, залежно від отриманої команди, виконувати відповідні дії: вмикати/вимикати окремі світлодіоди або регулювати за допомогою ШІМ яскравість світлодіодів);

– читання значень з сенсора освітленості (періодично зчитувати значення з датчика освітленості, якщо рівень зовнішнього освітлення низький, вмикати світлодіоди, якщо рівень освітленості високий, вимикати світлодіоди);

– вмикання/вимикання світлодіодів (вмикати всі світлодіоди – виклик функції `ALL_LED_ON()`, встановлюючи відповідні пін-коди в низький стан; вимикати всі світлодіоди – виклик функції `ALL_LED_OFF()`, встановлюючи відповідні пін-коди в високий стан).

Загальний алгоритм прошивки мікроконтролера наведено на рисунку 3.11.

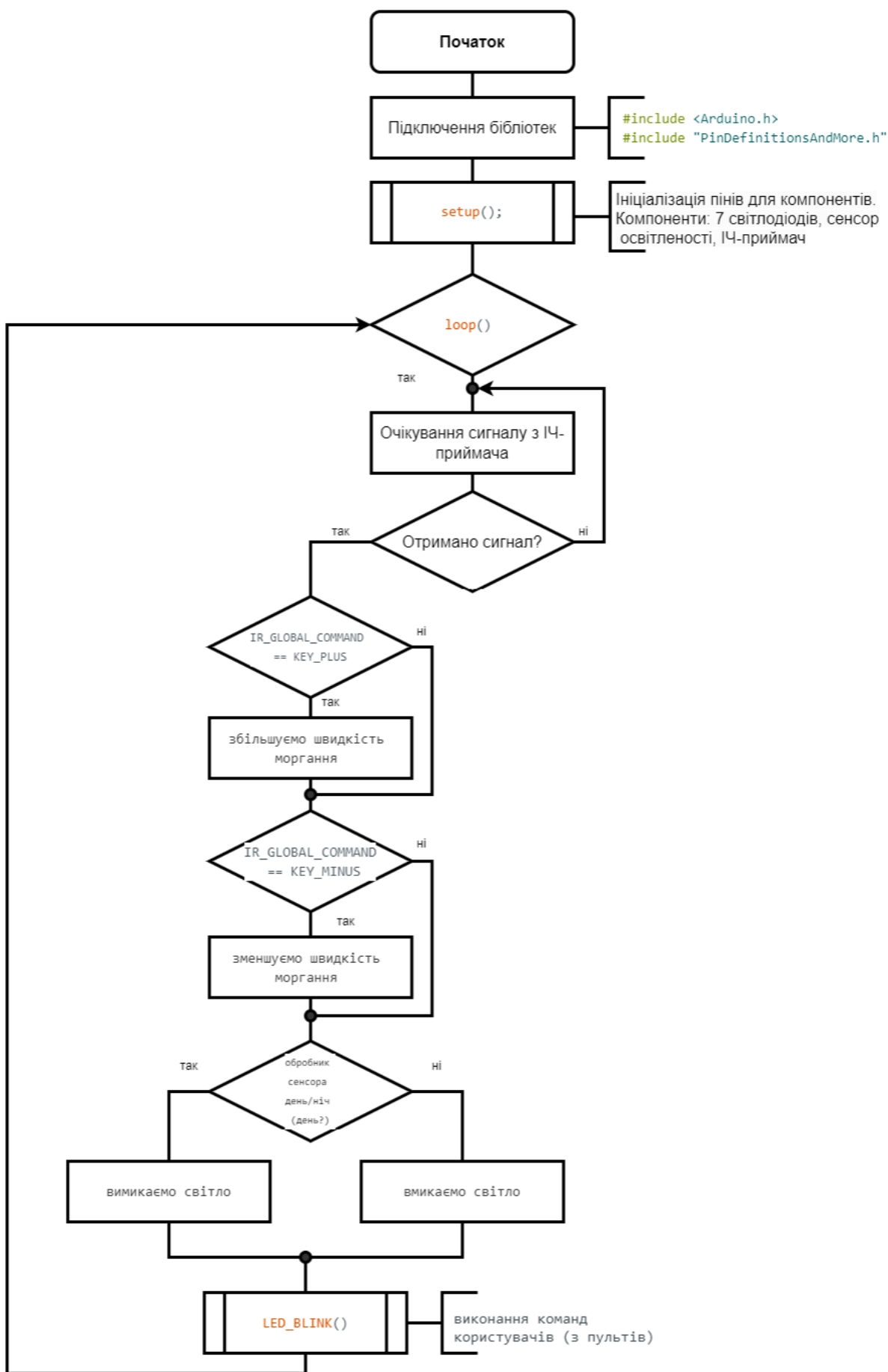


Рисунок 3.11 – Алгоритм прошивки для проєкту

В проєкті ми використовуємо два різних пульти, вони мають багато клавiш, тому потрібно систематизувати коди клавiш щоб нічого не переплутати. Для початку потрібно ці коди дізнатися. Для цього нам потрібно зібрати міні-проєкт за схемою, наведеною на рисунку 3.12.

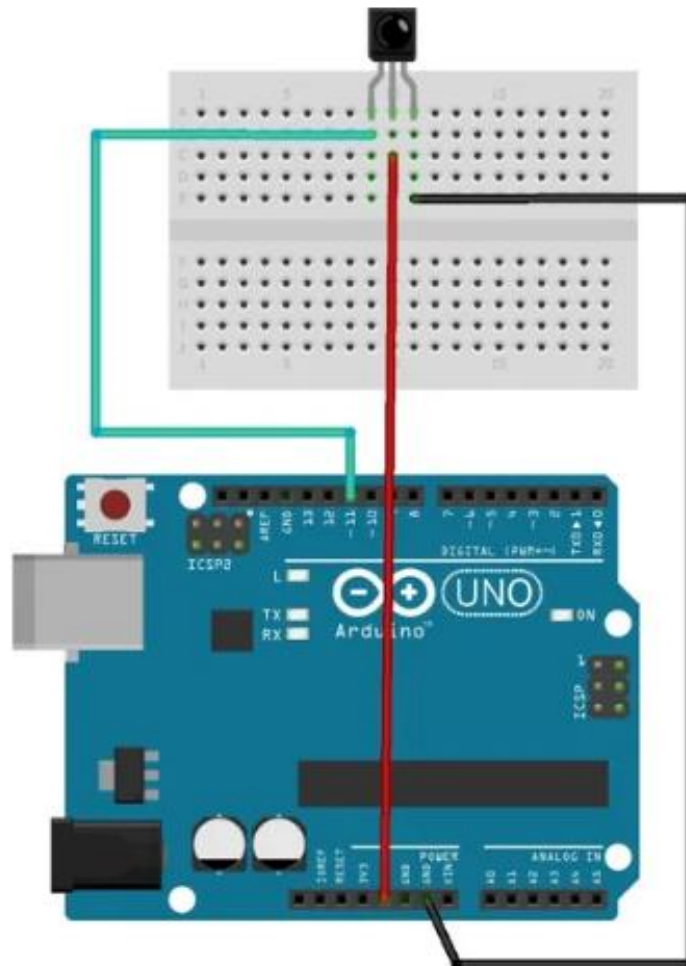


Рисунок 3.12 – Схема під'єднань для знаходження коду для кожної кнопки пульта дистанційного керування [2]

Наступний крок – потрібно під'єднати проєкт до комп'ютера, на якому встановлено середовище Arduino IDE, додати (встановити) бібліотеку IRRemote.h за допомогою менеджера бібліотек (виконати команду Інструменти/ Керування бібліотеками, далі в полі пошуку ввести потрібну нам бібліотеку та встановити її як на рисунку 3.13). Детальну документацію та приклад використання цієї бібліотеки можна знайти в репозиторії github [1].

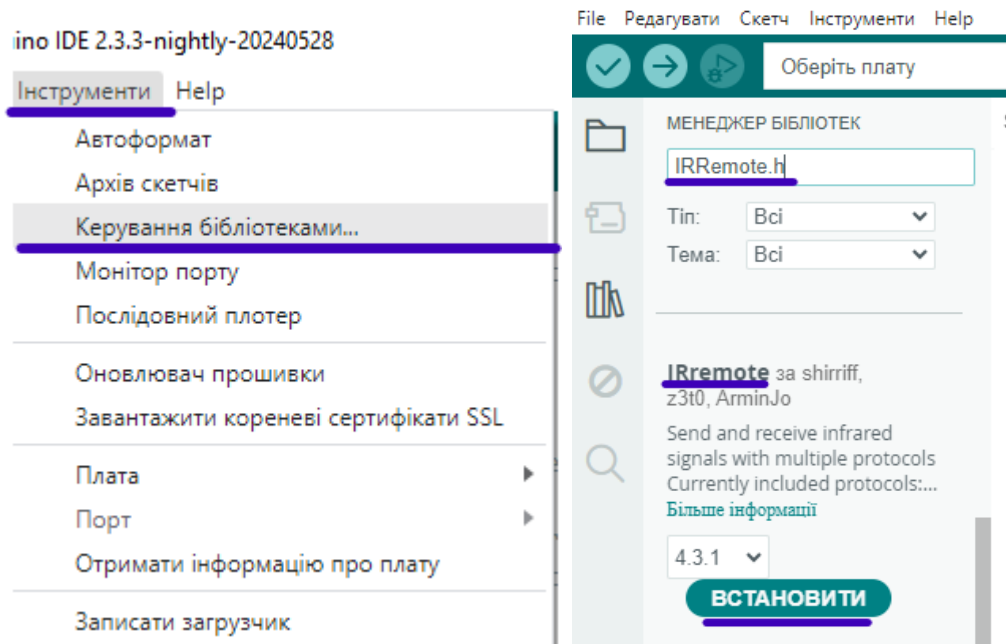


Рисунок 3.13 – Додавання бібліотеки в середовищі Arduino IDE

Щоб знайти ключові коди для пульта дистанційного керування, завантажимо готовий код з джерела [1] або [2] (рисунок 3.14) на мікроконтролер Arduino, відкриваємо монітор послідовного порту і під час натискання кнопок пульта бачимо результат (рисунок 3.15).

```

1  #include <IRremote.h>
2  int RECV_PIN = 11;
3  IRrecv irrecv(RECV_PIN);
4  decode_results results;
5
6  void setup()
7  {
8    Serial.begin(9600);
9    irrecv.enableIRIn(); // Запуск приймача
10 }
11
12 void loop()
13 {
14   if (irrecv.decode(&results))
15   {
16     Serial.println(results.value, HEX);
17     irrecv.resume(); // Отримання наступного значення
18   }
19 }

```

Рисунок 3.14 – Код, за допомогою якого дізнаємося коди клавіш пульта

```

FFFFFFFF
FD10EF
FFFFFFFF
FD906F
FFFFFFFF
FD50AF
FFFFFFFF
FD30CF
FFFFFFFF
FDB04F
FDB04F
FFFFFFFF
FD708F
FFFFFFFF
FD08F7
FFFFFFFF
FD8877
FD708F
FFFFFFFF
FD28D7
FDA857

```

Рисунок 3.15 – Приклад розкодування клавiш пульта

Код «FFFFFFFF» з'являється, коли ми двічі підряд натискаємо ту саму клавiшу (або задовго її утримуємо).

Таким чином, ми отримуємо коди всіх клавiш будь-якого ІЧ-пульта (для нашого проєкту потрібно дізнатися коди для двох пультів).

Коди кнопок для меншого пульта наведено в таблиці 3.1, а запрограмовані в цьому проєкті кнопки та їх функції – в таблиці 3.2. Більший пульт має подібні функції (таблиця 3.3).

Таблиця 3.1 – Коди кнопок стандартного ІР пульта для Arduino

Кнопка	Код	Кнопка	Код	Кнопка	Код
	22		28		7
	12		90		21
	24		66		25
	94		82		13
	8		74		9





Таблиця 3.2 – Функції запрограмованих кнопок меншого пульта

Кнопка	Змінна в програмі	Код	Запрограмовані функції
	KEY_0	22	засвітити сині світлодіоди (LED_WHT_D0, LED_WHT_D6)
	KEY_1	12	засвітити червоні світлодіоди (LED_RED_D1, LED_RED_D5)
	KEY_2	24	засвітити жовті світлодіоди (LED_YLW_D2, LED_YLW_D4)
	KEY_3	94	засвітити зелений світлодіод (LED_GRN_D3)
	KEY_MINUS	7	зменшити частоту мигання
	KEY_PLUS	21	збільшити частоту мигання
	KEY_EQ	9	засвітити всі світлодіоди

Таблиця 3.3 – Запрограмовані функції пульта Нуауу

Кнопка	Код	Запрограмовані функції
	76	засвітити сині світлодіоди (LED_WHT_D0, LED_WHT_D6)
	73	засвітити червоні світлодіоди (LED_RED_D1, LED_RED_D5)
	67	засвітити жовті світлодіоди (LED_YLW_D2, LED_YLW_D4)
	16	засвітити зелений світлодіод (LED_GRN_D3)

## Продовження таблиці 3.3

<p><b>BASS</b></p> 	85	збільшити частоту мигання всіх світлодіодів в шість разів $G\_DELAY = 6 * G\_DELTA$ ;
<p><b>TXT/TV</b></p> 	70	збільшити частоту мигання всіх світлодіодів в чотири рази $G\_DELAY = 4 * G\_DELTA$ ;
<p><b>S.SYS</b></p> 	19	збільшити частоту мигання всіх світлодіодів в два рази $G\_DELAY = 2 * G\_DELTA$ ;
<p><b>SURROUND</b></p> 	87	встановити початкову частоту мигання всіх світлодіодів $G\_DELAY = G\_DELTA$ ;

### 3.3 Тестування роботи спроектованої системи освітлення

Перед тим, як прикріпити мікроконтролер з блоком живлення до корпусу лампи, потрібно перевірити коректність роботи пристрою. Для цього приєднуємо пристрій до повербанку або вмикаємо вбудований блок живлення. Спочатку бачимо мигання червоного індикатора (рисунок 3.16) на платі мікроконтролера, що свідчить про його роботу, а приблизно через 1 секунду засвічуються всі світлодіоди (рисунок 3.17). Як вже було зазначено раніше, для управління системою освітлення використовуємо два пульти (рисунок 3.18).

На рисунку 3.16 плата живиться через USB від ПК, а на рисунку 3.17 – від повербанку через гніздо живлення DC-005.



Рисунок 3.16 – На плату подано живлення



Рисунок 3.17 – Ввімкнуто всі світлодіоди



Рисунок 3.18 – Пульти для керування освітленням

При натисканні відповідних кнопок, розглянутих вище, на пультах ми бачимо світіння або мерехтіння світлодіодів різних кольорів та з різною яскравістю (рисунок 3.19). При потраплянні прямого пучка світла на фоторезистор всі світлодіоди вимикаються (рисунок 3.20), що дає можливість економити енергію автоматично.

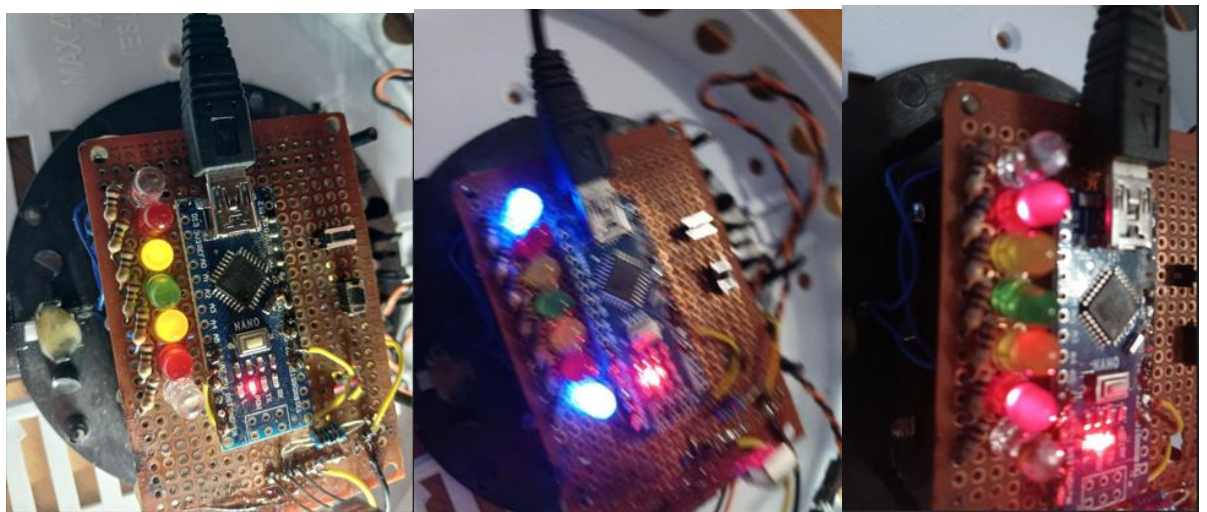


Рисунок 3.19 – Тестування роботи пульта

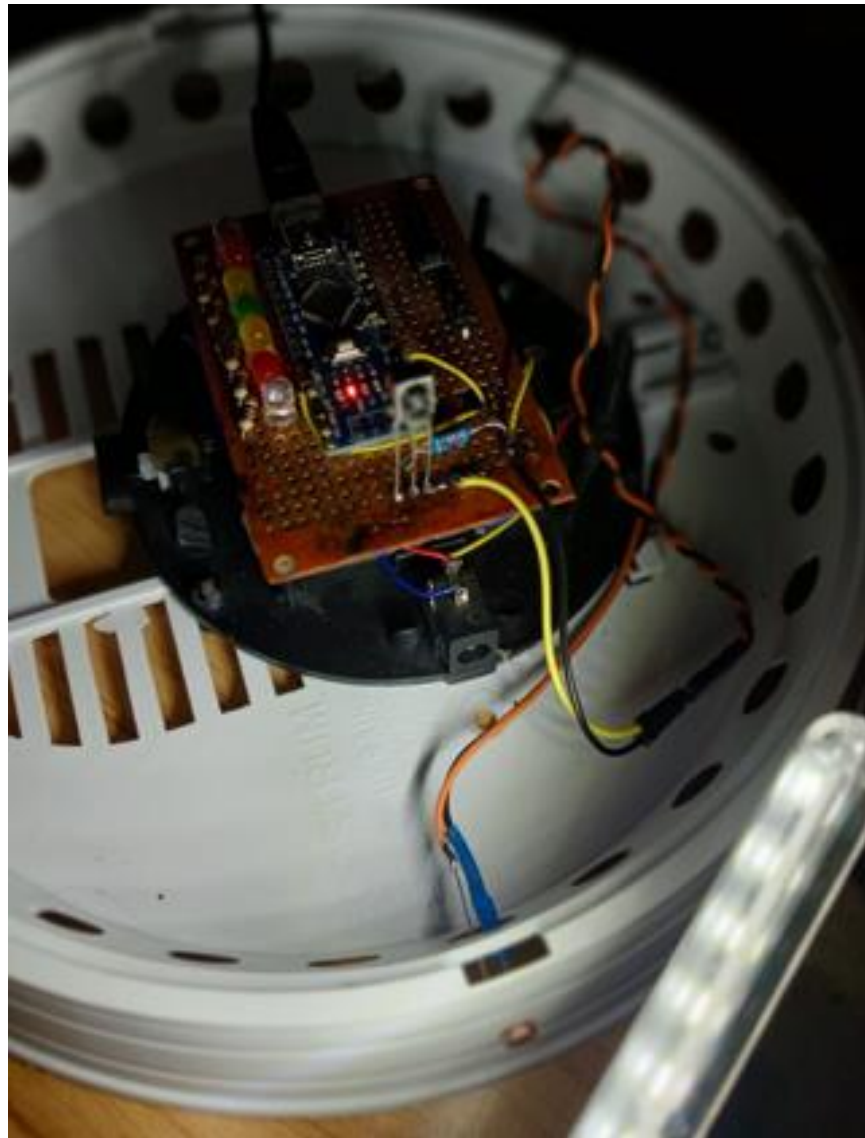


Рисунок 3.20 – Тестування роботи фоторезистора

При тестуванні роботи пультів також можна через монітор послідовного порту відслідковувати які команди надходять від пультів дистанційного керування (рисунки 3.21-3.23).

```

Монітор порту x
Повідомлення (Введіть повідомлення для відправки до 'Arduino Nano' на 'COM3')
Новий рядок 9600 бод
13:03:25.189 -> START D:\main\DIP\dip2024\DipArduinoLigght\SmartLightingV0\SmartLightingV0.ino from May 29 2024
13:03:25.271 -> Using library version 2.1.0
13:03:25.271 -> Ready to receive NEC IR signals at pin 2
  
```

Рисунок 3.21 – Плата підключена, очікує команд пультів  
(через пін 2 під'єднано ІЧ-приймач)



Монітор порту x

Повідомлення (Введіть повідомлення для відправки до 'Arduino Nano' на 'COM3')

Новий рядок 9600 бод

```
13:04:10.090 -> Command: 12
13:04:10.217 -> Command: 12
13:04:10.390 -> Command: 12
13:04:14.741 -> Command: 24
13:04:19.384 -> Command: 94
13:04:19.524 -> Command: 94
13:04:24.756 -> Command: 90
13:04:28.824 -> Command: 90
13:04:28.966 -> Command: 90
13:04:33.173 -> Command: 82
13:04:36.465 -> Command: 9
13:04:36.591 -> Command: 9
13:04:44.393 -> Command: 21
13:04:45.302 -> Command: 21
13:04:45.433 -> Command: 21
13:04:49.375 -> Command: 21
13:04:50.081 -> Command: 7
13:04:50.254 -> Command: 7
```

Рисунок 3.22 – Команди від меншого пульта



Монітор порту x

Повідомлення (Введіть повідомлення для відправки до 'Arduino Nano' на 'COM3')

Новий рядок 9600 бод

```
13:06:00.241 -> Command: 16
13:06:02.946 -> Command: 75
13:06:03.071 -> Command: 67
13:06:04.556 -> Command: 76
13:06:04.743 -> Command: 76
13:06:04.856 -> Command: 76
13:06:10.400 -> Command: 70
13:06:10.579 -> Command: 70
13:06:10.721 -> Command: 70
13:06:10.851 -> Command: 70
13:06:12.804 -> Command: 19
13:06:12.990 -> Command: 19
13:06:16.569 -> Command: 19
13:06:16.695 -> Command: 19
```

Рисунок 3.23 – Команди від більшого пульта

## ВИСНОВКИ

Під час виконання кваліфікаційної роботи був здійснений огляд сучасних пристроїв та сценаріїв освітлення, виконано обґрунтування актуальності обраної теми. Також розглянуто основні фактори, які беруть участь в реалізації різних сценаріїв освітлення, що дало змогу обрати ті критерії, які впливатимуть на роботу спроектованої в даній роботі системи розумного освітлення, а саме: наявність денного світла та дистанційне управління.

Здійснено огляд та вибір компонентів для проектування апаратної частини системи освітлення з багатофакторним керуванням. Розглянуто різні джерела світла, особливу увагу приділили світлодіодам та розумним лампочкам. Для реалізації проекту обрано плату Arduino Nano, світлодіоди, інфрачервоний ресівер VS1838B, фоторезистор, пульти.

Було розроблено електричну структурну схему пристрою системи управління освітленням, на основі якої зібрано прототип системи освітлення на базі мікроконтролера Arduino Nano, так як він разом з фоторезистором та інфрачервоним ресівером дасть змогу організувати багатофакторне керування.

Розроблено алгоритм скетчу мікроконтролера для ефективного керування пристроєм. Написано програми мовою C++. Прошивку мікроконтролера виконано в середовищі Arduino IDE.

Отже, в результаті виконання поставлених завдань продемонстровано створення системи розумного освітлення з використанням Arduino Nano, світлодіодів, датчика освітленості та інфрачервоних пультів. Проєкт дозволяє автоматично регулювати освітлення залежно від рівня природного освітлення та керувати ним за допомогою пультів, також зменшити енергоспоживання та значно скоротити витрати на електроенергію.

Є перспективи подальшого розширення проєкту шляхом додавання більшої кількості сенсорів або розширення алгоритмів керування та реалізації додаткових функцій, таких як керування через Bluetooth або Wi-Fi.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. GitHub – Arduino-IRremote/Arduino-IRremote: Infrared remote library for Arduino: send and receive infrared signals with multiple protocols. GitHub. URL: <https://github.com/Arduino-IRremote/Arduino-IRremote> (дата звернення: 10.04.2024)
2. Instructables. Arduino Infrared Remote Tutorial. *Instructables*. URL: <https://www.instructables.com/Arduino-Infrared-Remote-tutorial/> (дата звернення: 06.04.2024).
3. IR Receivers. *Vishay Intertechnology: Passives & Discrete Semiconductors*. URL: <https://www.vishay.com/en/ir-receiver-modules/> (дата звернення: 22.03.2024).
4. LED Lighting Products. *Nanoleaf*. URL: <https://nanoleaf.me/en-UK/> (дата звернення: 21.03.2024).
5. Lighting circuits guide. Simple solutions for control and protection of lighting circuits. URL: <https://nous.ua/smart/lamps/> [https://earecoebandung.id/online-resources/Electrical-Installation/2-EnergyManagement\\_web\\_gen\\_web/res/Lighting\\_circuits\\_guide.pdf](https://earecoebandung.id/online-resources/Electrical-Installation/2-EnergyManagement_web_gen_web/res/Lighting_circuits_guide.pdf) (дата звернення: 16.03.2024).
6. Smart Lamps – Feilo Sylvania. *Sylvania*. URL: <https://www.sylvania-lighting.com/en-gb/smart-lamps/> (дата звернення: 21.03.2024).
7. Smart Lighting Control System, Smart Energy Management Solutions. *Jiangsu Acrel Electrical Manufacturing*. URL: <https://www.acrel.gr/smart-lighting-control-system.html> (дата звернення: 26.03.2024).
8. Smart lighting. *Philips Hue*. URL: <https://www.philips-hue.com/en-us> (дата звернення: 09.04.2024).
9. V. Kumar, P. Sharma and K. Kamaldeep, «Smart Lighting System Using Arduino,» *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, Dehradun, India, 2021, pp. 1-5, doi: 10.1109/UPCON52273.2021.9667610.

10. What is a Light Sensor? Types, Uses, Arduino Guide. *Latest Open Tech From Seeed*. URL: <https://www.seeedstudio.com/blog/2020/01/08/what-is-a-light-sensor-types-uses-arduino-guide/> (дата звернення: 17.03.2024).
11. Wi-Fi enabled LED smart lights. LIFX. URL: <https://www.lifx.com/> (дата звернення: 21.03.2024).
12. Zdolbitska, N., Delyavskyy, M., Lishchyna, N., Lishchyna, V., Lavrenchuk, S., Sulim, V. (2023). DIY Smart Auxiliary Power Supply for Emergency Use. In: Hu, Z., Zhang, Q., He, M. (eds) *Advances in Artificial Systems for Logistics Engineering III. ICAILE 2023. Lecture Notes on Data Engineering and Communications Technologies*, vol 180. Springer, Cham. URL: [https://doi.org/10.1007/978-3-031-36115-9\\_35](https://doi.org/10.1007/978-3-031-36115-9_35) (дата звернення: 12.03.2024).
13. Будова та принцип роботи світлодіода. *Проба Пера*. URL: <https://probarpera.org/publication/13/55593/budova-ta-pryntsyp-roboty-svitlodioda.html> (дата звернення: 07.03.2024).
14. Василюк А.О. Управління зовнішнім освітленням на мікроконтролері Atmega. Кваліфікаційна робота на здобуття ступеня магістра спеціальності 171 «Електроніка». Національний авіаційний університет. Київ, 2023. 77 с.
15. Вичерпний посібник із світловипромінюючих діодів (світлодіодів). *LEDYi Lighting*. URL: <https://www.ledyilighting.com/uk/a-comprehensive-guide-to-led/> (дата звернення: 05.03.2024).
16. Гусев А., Шинкар В. Використання світлодіодних джерел світла в освітленні міст. *Проблеми охорони праці, промислової та цивільної безпеки*, 2023, с. 58-63.
17. Лавренчук С. В., Артемук А. В. Оптимізація системи вуличного освітлення на основі IoT -пристроїв. *Міжнародна науково-практична інтернет-конференція молодих учених та студентів «Actual Problems of Automation and Control»*. Вип. 11. Луцьк, 2023. С. 156-160.
18. Розумне освітлення всього будинку Yilai. URL: [https://cn.yeelight.com/zh\\_CN/](https://cn.yeelight.com/zh_CN/) (дата звернення: 18.03.2024).

19. Розумне освітлення. NOUS. URL: <https://nous.ua/smart/lamps/> (дата звернення: 03.04.2024).
20. Трубочанінова К. А., Черкашин Є. А. Система «розумного» освітлення. *Інформаційно-керуючі системи на залізничному транспорті*, 2023, 28.2: с. 14-21.

# ДОДАТКИ

## Додаток А

### Прошивка мікроконтролера

```

/*
 * Small memory footprint and no timer usage!
 *
 * Receives IR protocol data of NEC protocol using pin change interrupts.
 * On complete received IR command the function handleReceivedIRData(uint16_t
aAddress, uint8_t aCommand, uint8_t aFlags)
 * is called in Interrupt context but with interrupts being enabled to enable use of
delay() etc.
 * !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 * Functions called in interrupt context should be running as short as possible,
 * so if you require longer action, save the data (address + command) and handle it
in the main loop.
 * !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 *
 * The FAST protocol is a proprietary modified JVC protocol without address, with
parity and with a shorter header.
 * FAST Protocol characteristics:
 * - Bit timing is like NEC or JVC
 * - The header is shorter, 3156 vs. 12500
 * - No address and 16 bit data, interpreted as 8 bit command and 8 bit inverted
command,
 * leading to a fixed protocol length of  $(6 + (16 * 3) + 1) * 526 = 55 * 526 =$ 
28930 microseconds or 29 ms.
 * - Repeats are sent as complete frames but in a 50 ms period / with a 21 ms
distance.
 *
 *
 * This file is part of IRMP https://github.com/IRMP-org/IRMP.
 * This file is part of Arduino-IRremote https://github.com/Arduino-IRremote/Arduino-IRremote.
 */

#include <Arduino.h>

#include "PinDefinitionsAndMore.h" // Set IR_RECEIVE_PIN for different CPU's

//#define DEBUG // to see if attachInterrupt is used
//#define TRACE // to see the state of the ISR state machine

/*
 * Protocol selection
 */
//#define DISABLE_PARITY_CHECKS // Disable parity checks. Saves 48 bytes of program
memory.
//#define USE_EXTENDED_NEC_PROTOCOL // Like NEC, but take the 16 bit address as one
16 bit value and not as 8 bit normal and 8 bit inverted value.

```

```

// #define USE_ONKYO_PROTOCOL // Like NEC, but take the 16 bit address and command
each as one 16 bit value and not as 8 bit normal and 8 bit inverted value.
// #define USE_FAST_PROTOCOL // Use FAST protocol instead of NEC / ONKYO.
// #define ENABLE_NEC2_REPEATS // Instead of sending / receiving the NEC special
repeat code, send / receive the original frame for repeat.
/*
 * Set compile options to modify the generated code.
 */
// #define DISABLE_PARITY_CHECKS // Disable parity checks. Saves 48 bytes of program
memory.
// #define USE_CALLBACK_FOR_TINY_RECEIVER // Call the fixed function "void
handleReceivedTinyIRData()" each time a frame or repeat is received.

#include "TinyIRReceiver.hpp" // бібліотека інфрачервоного ресівера

/* константи світлодіодів : номер - назва порта - тип IO - колір підключеного діода
*/
const int LED_WHT_D0 = 3; // 6 D3 IO3 синій
const int LED_RED_D1 = 4; // 7 D4 IO4 червоний
const int LED_YLW_D2 = 5; // 8 D5 IO5 жовтий
const int LED_GRN_D3 = 8; // 11 D8 IO8 зелений
const int LED_YLW_D4 = 9; // 12 D9 IO9 жовтий
const int LED_RED_D5 = 10; // 13 D10 IO10 червоний
const int LED_WHT_D6 = 11; // 14 D11 IO11 синій

/* константи клавіатури маленького пульта Ардуїно */
/* скорочення після назви константи відповідають напису на кнопці пульта */
const int KEY_0 = 22; //
const int KEY_1 = 12; //
const int KEY_2 = 24; //
const int KEY_3 = 94; //
const int KEY_4 = 8; //
const int KEY_5 = 28; //
const int KEY_6 = 90; //
const int KEY_7 = 66; //
const int KEY_8 = 82; //
const int KEY_9 = 74; //

const int KEY_MINUS = 7; //
const int KEY_PLUS = 21; //
const int KEY_100 = 25; //
const int KEY_200 = 13; //
const int KEY_EQ = 9; // фіолетова кнопка

// максимальна затримка / загальна
const int DELAY_MAX = 1400;

// світловий сенсор
const int lightSensorPin = 7; // A7 - аналоговий вхід до якого
підключено сенсор день/ніч //
volatile bool lightSensorStatus = false; // false - не світить - день , true -
ніч, світають усі світлодіоди

```

```

volatile int LIGHT_SENSOR_GLOBAL_COMMAND = 0; // команда сенсора 1 - LIGHT_ON / 0 -
LIGHT_OFF
volatile byte seconds; // затримка для таймера - пропуск
тактів на спрацювання

// команда пульта
volatile int IR_GLOBAL_COMMAND = 0;
// попередня команда пульта - кеш
volatile int IR_GLOBAL_COMMAND_OLD = 0;

volatile int G_DELAY = 900; // час для затримки - milliseconds
int G_DELTA = 150; // крок-дельта для затримки delta-milliseconds

/*
 * Helper macro for getting a macro definition as string
 */
#ifdef STR_HELPER
#define STR_HELPER(x) #x
#define STR(x) STR_HELPER(x)
#else
#define STR_HELPER(x) #x
#define STR(x) STR_HELPER(x)
#endif

void setup() {
    Serial.begin(9600);

    // ініціалізуємо цифрові порти на вихід
    pinMode(LED_WHT_D0, OUTPUT);
    pinMode(LED_RED_D1, OUTPUT);
    pinMode(LED_YLW_D2, OUTPUT);
    pinMode(LED_GRN_D3, OUTPUT);
    pinMode(LED_YLW_D4, OUTPUT);
    pinMode(LED_RED_D5, OUTPUT);
    pinMode(LED_WHT_D6, OUTPUT);

    // високий рівень - гасимо усі світлодіоди
    ALL_LED_OFF();

    //-----
    // ініціалізація Timer1
    cli(); // відключити глобальні переривання контролера
    TCCR1A = 0; // встановити регістри ініціалізації таймера1 в 0
    TCCR1B = 0;

    // OCR1A = 15624; // встановлення регістра збігу десь на 1 секунду
    // OCR1A = 1562; // встановлення регістра збігу десь на 100 мілісекунд
    // OCR1A = 156; // встановлення регістра збігу десь на 10 мілісекунд
    OCR1A = 624; // встановлення регістра збігу десь на XXX мілісекунд
    TCCR1B |= (1 << WGM12); // включити CTC режим порівняння при збігу

    // встановлення бітів CS10 та CS12 для коефіцієнта дільника 1024
    TCCR1B |= (1 << CS10);
    TCCR1B |= (1 << CS12);

```

```

TIMSK1 |= (1 << OCIE1A); // включити переривання при збігу значення 624
sei(); // включити глобальні переривання контролера
//-----

#if defined(__AVR_ATmega32U4__) || defined(SERIAL_PORT_USBVIRTUAL) ||
defined(SERIAL_USB) /*stm32duino*/ || defined(USBCON) /*STM32_stm32*/ ||
defined(SERIALUSB_PID) || defined(ARDUINO_attiny3217)
    delay(4000); // To be able to connect Serial monitor after reset or power up and
before first print out. Do not wait for an attached Serial Monitor!
#endif
    // Just to know which program is running on my Arduino
#if defined(ESP8266) || defined(ESP32)
    Serial.println();
#endif
    Serial.println(F("START " __FILE__ " from " __DATE__ "\r\nUsing library version "
VERSION_TINYIR));

    // Enables the interrupt generation on change of IR input signal
    if (!initPCIInterruptForTinyReceiver()) {
        Serial.println(F("No interrupt available for pin " STR(IR_RECEIVE_PIN))); //
optimized out by the compiler, if not required :-))
    }
#if defined(USE_FAST_PROTOCOL)
    Serial.println(F("Ready to receive Fast IR signals at pin "
STR(IR_RECEIVE_PIN)));
#else
    Serial.println(F("Ready to receive NEC IR signals at pin " STR(IR_RECEIVE_PIN)));
#endif
}

// обробник переривання від таймера 1
ISR(TIMER1_COMPA_vect)
{
    statusA7();
}

// обробник аналогового сигналу давача фоторезистора ніч/день
void statusA7()
{
    if(seconds > 0)
    {
        seconds--;
    }
    else
    {
        if(analogRead(lightSensorPin) < 250) // рівень освітлення, менше значення менше
освітлення
        {
            seconds = 150;
            LIGHT_SENSOR_GLOBAL_COMMAND = 1;
            // for debug

```

```

        // Serial.print(" ON ");
        // Serial.println(seconds);
    }
    else
    {
        LIGHT_SENSOR_GLOBAL_COMMAND = 0;
        // for debug
        // Serial.print(" OFF ");
        // Serial.println(seconds);
    }
}
}

// ***** основний цикл програми
// *****
void loop() {
    if (TinyIRReceiverData.justWritten) { // отримуємо статус ресівера, якщо true -
        тоді обробляємо отриману команду із ІЧ-ресівера
        TinyIRReceiverData.justWritten = false;

        // debug mode
        //Serial.print(" Command: ");
        //Serial.print(TinyIRReceiverData.Command);

        IR_GLOBAL_COMMAND_OLD = IR_GLOBAL_COMMAND;
        IR_GLOBAL_COMMAND = TinyIRReceiverData.Command;

        // збільшуємо швидкість моргання
        if(IR_GLOBAL_COMMAND == KEY_PLUS || IR_GLOBAL_COMMAND == 17)
        {
            if(G_DELAY - G_DELTA > 0)
            {
                G_DELAY = G_DELAY - G_DELTA;
            }
            IR_GLOBAL_COMMAND = IR_GLOBAL_COMMAND_OLD;
        }

        // зменшуємо швидкість моргання
        if(IR_GLOBAL_COMMAND == KEY_MINUS || IR_GLOBAL_COMMAND == 64)
        {
            if(G_DELAY <= DELAY_MAX)
            {
                G_DELAY = G_DELAY + G_DELTA;
            }
            IR_GLOBAL_COMMAND = IR_GLOBAL_COMMAND_OLD;
        }

        // повторюваність команд та визначення розширених протоколів - зараз не
        використовуємо
        // if (TinyIRReceiverData.Flags == IRDATA_FLAGS_IS_REPEAT) {
        //     Serial.print(F(" Repeat"));

```

```

    // }
    // if (TinyIRReceiverData.Flags == IRDATA_FLAGS_PARITY_FAILED) {
    //     Serial.print(F(" Parity failed"));
    //     #if !defined(USE_EXTENDED_NEC_PROTOCOL) && !defined(USE_ONKYO_PROTOCOL)
    //         Serial.print(F(", try USE_EXTENDED_NEC_PROTOCOL or
USE_ONKYO_PROTOCOL"));
    //     #endif
    // }

    //Serial.println(); // for debug
}

// виклик обробників для світлових ефектів
// окремо обробляємо дві групи команд - від ІЧ-ресівера та фоторезистора ніч/день
if(LIGHT_SENSOR_GLOBAL_COMMAND == 1) // обробник сенсора день/ніч
{
    ALL_LED_ON();
    delay(G_DELTA);
}
else
{
    ALL_LED_OFF();
    LED_BLINK();
}

}

// обробник - програми світлові ефекти
// у розгалуженні опрацьовуємо основні коди команд (пульт Ардуїно) та додаткові для
пульту ІЧ-телевізора
void LED_BLINK()
{
    if(IR_GLOBAL_COMMAND == KEY_0 || IR_GLOBAL_COMMAND == 76) // -----
    -----
    {
        if(G_DELAY-G_DELTA <= 0)
        {
            digitalWrite(LED_WHT_D0, LOW); // led on
            digitalWrite(LED_WHT_D6, LOW); // led on
        }
        else
        {
            digitalWrite(LED_WHT_D0, LOW); // led on
            digitalWrite(LED_WHT_D6, LOW); // led on
            delay(G_DELAY); // wait for a second
            digitalWrite(LED_WHT_D0, HIGH); // led off
            digitalWrite(LED_WHT_D6, HIGH); // led off
            delay(G_DELAY); // wait for a second
        }
    }
}
else

```

```

{
    digitalWrite(LED_WHT_D0, HIGH); // високий рівень - гасимо світлодіод
}

if(IR_GLOBAL_COMMAND == KEY_1 || IR_GLOBAL_COMMAND == 73) // -----
-----
{
    if(G_DELAY-G_DELTA <= 0)
    {
        digitalWrite(LED_RED_D1, LOW); // led on
        digitalWrite(LED_RED_D5, LOW); // led on
    }
    else
    {
        digitalWrite(LED_RED_D1, LOW); // led on
        digitalWrite(LED_RED_D5, LOW); // led on
        delay(G_DELAY); // wait for a second
        digitalWrite(LED_RED_D1, HIGH); // led off
        digitalWrite(LED_RED_D5, HIGH); // led off
        delay(G_DELAY); // wait for a second
    }
}
else
{
    digitalWrite(LED_RED_D1, HIGH); // високий рівень - гасимо світлодіод
}

if(IR_GLOBAL_COMMAND == KEY_2 || IR_GLOBAL_COMMAND == 67) // -----
-----
{
    if(G_DELAY-G_DELTA <= 0)
    {
        digitalWrite(LED_YLW_D2, LOW); // led on
        digitalWrite(LED_YLW_D4, LOW); // led on
    }
    else
    {
        digitalWrite(LED_YLW_D2, LOW); // led on
        digitalWrite(LED_YLW_D4, LOW); // led on
        delay(G_DELAY); // wait for a second
        digitalWrite(LED_YLW_D2, HIGH); // led off
        digitalWrite(LED_YLW_D4, HIGH); // led on
        delay(G_DELAY); // wait for a second
    }
}
else
{
    digitalWrite(LED_YLW_D2, HIGH); // високий рівень - гасимо світлодіод
}

```

```

if(IR_GLOBAL_COMMAND == KEY_3 || IR_GLOBAL_COMMAND == 16) // -----
{
  if(G_DELAY-G_DELTA <= 0)
  {
    digitalWrite(LED_GRN_D3, LOW); // led on
  }
  else
  {
    digitalWrite(LED_GRN_D3, LOW); // led on
    delay(G_DELAY); // wait for a second
    digitalWrite(LED_GRN_D3, HIGH); // led off
    delay(G_DELAY); // wait for a second
  }
}
else
{
  digitalWrite(LED_GRN_D3, HIGH); // високий рівень - гасимо світлодіод
}

if(IR_GLOBAL_COMMAND == KEY_3) // -----
{
  if(G_DELAY-G_DELTA <= 0)
  {
    digitalWrite(LED_GRN_D3, LOW); // led on
  }
  else
  {
    digitalWrite(LED_GRN_D3, LOW); // led on
    delay(G_DELAY); // wait for a second
    digitalWrite(LED_GRN_D3, HIGH); // led off
    delay(G_DELAY); // wait for a second
  }
}
else
{
  digitalWrite(LED_GRN_D3, HIGH); // високий рівень - гасимо світлодіод
}

// -----
if(IR_GLOBAL_COMMAND == KEY_EQ || IR_GLOBAL_COMMAND == 85 || IR_GLOBAL_COMMAND ==
70 || IR_GLOBAL_COMMAND == 19 || IR_GLOBAL_COMMAND == 87)
{
  if(IR_GLOBAL_COMMAND == 85) {G_DELAY = 6 * G_DELTA;}
  if(IR_GLOBAL_COMMAND == 70) {G_DELAY = 4 * G_DELTA;}
  if(IR_GLOBAL_COMMAND == 19) {G_DELAY = 2 * G_DELTA;}
  if(IR_GLOBAL_COMMAND == 87) {G_DELAY = G_DELTA;}

  if(G_DELAY-G_DELTA <= 0)
  { // led on
    ALL_LED_ON();
  }
}

```

```

    }
    else
    {
        ALL_LED_ON();
        delay(G_DELAY);           // wait for a second
        ALL_LED_OFF();
        delay(G_DELAY);           // wait for a second
    }
}
else
{ // led off
    ALL_LED_OFF();
}
}

void ALL_LED_OFF()
{
    digitalWrite(LED_WHT_D0, HIGH);
    digitalWrite(LED_RED_D1, HIGH);
    digitalWrite(LED_YLW_D2, HIGH);
    digitalWrite(LED_GRN_D3, HIGH);
    digitalWrite(LED_YLW_D4, HIGH);
    digitalWrite(LED_RED_D5, HIGH);
    digitalWrite(LED_WHT_D6, HIGH);
}

void ALL_LED_ON()
{
    digitalWrite(LED_WHT_D0, LOW);
    digitalWrite(LED_RED_D1, LOW);
    digitalWrite(LED_YLW_D2, LOW);
    digitalWrite(LED_GRN_D3, LOW);
    digitalWrite(LED_YLW_D4, LOW);
    digitalWrite(LED_RED_D5, LOW);
    digitalWrite(LED_WHT_D6, LOW);
}

/*
 * Optional code, if you require a callback
 */
#ifdef USE_CALLBACK_FOR_TINY_RECEIVER
/*
 * This is the function, which is called if a complete frame was received
 * It runs in an ISR context with interrupts enabled, so functions like delay() etc.
should work here
 */
# if defined(ESP8266) || defined(ESP32)
IRAM_ATTR
# endif

void handleReceivedTinyIRData() {

```

```
# if defined(ARDUINO_ARCH_MBED) || defined(ESP32)
/*
 * Printing is not allowed in ISR context for any kind of RTOS, so we use the
slightly more complex,
 * but recommended way for handling a callback :-). Copy data for main loop.
 * For Mbed we get a kernel panic and "Error Message: Semaphore: 0x0, Not allowed
in ISR context" for Serial.print()
 * for ESP32 we get a "Guru Meditation Error: Core 1 panic'ed" (we also have an
RTOS running!)
 */
// Do something useful here...
# else
// As an example, print very short output, since we are in an interrupt context
and do not want to miss the next interrupts of the repeats coming soon
    printTinyReceiverResultMinimal(&Serial);
# endif
}
#endif
```