Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Факультет Комп'ютерної інженерії та управління
(повна назва)
Кафедра Автоматизації проєктування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Система підсвічування моніторів ПК на основі Arduino UNO R3

(тема)

Виконав: здобувач IV року навчання,
групи КІУКІ-21-7
Бибич Р.О.
(прізвище, ініціали)

спеціальності 123 – Комп'ютерна інженерія
(шифр і назва спеціальності)

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

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

Керівник ас. Корнієнко В.Р.
(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 проєктування обчислювальної техніки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Тип програми Освітньо-професійна

Освітня програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Бибичу Ростиславу Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Система підсвічування моніторів ПК на основі Arduino UNO R3

затверджена наказом університету від 21 05 2025 р. № 403Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10 06 2025 р.

3. Вихідні дані до роботи
Arduino
Адресна RGB світлодіодна стрічка WS2811
Інфрачервоний приймач VS1838B

4. Перелік питань, що потрібно опрацювати в роботі
Аналіз існуючих типів систем підсвічування
Розробка структурної схеми пристрою
Розробка функціональної схеми
Розробка алгоритмів керування підсвічуванням
Тестування та оптимізація роботи системи

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Видача теми проєкту, узгодження і затвердження теми	08.05.2025 – 09.05.2025	
2	Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів	09.05.2025 – 14.05.2025	
3	Розробка структурної схеми пристрою, вибір апаратної платформи	14.05.2025 – 16.05.2025	
4	Розробка функціональної схеми програми	16.05.2025 – 17.05.2025	
5	Розробка програмних модулів. Проведення тестування	17.05.2025 – 23.05.2025	
6	Оформлення пояснювальної записки	23.05.2025 – 25.05.2025	
7	Перевірка виконаного проєкту керівником, допуск до захисту	25.05.2025 – 10.06.2025	

Дата видачі завдання 08.05.2025 р.

Студент _____
(підпис)

Керівник роботи _____ ас. Корнієнко В.Р.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи містить 54 сторінки, 14 рисунків, 19 джерел посилання.

ARDUINO, ІЛЮМІНАЦІЯ, RGB, WS2811, LED, VS1838B, СВІТЛОВІ ЕФЕКТИ.

Метою кваліфікаційної роботи є розробка мікроконтролерної системи динамічного підсвічування моніторів персональних комп'ютерів на базі Arduino з використанням адресної RGB світлодіодної стрічки. У роботі проведено аналіз існуючих підходів до реалізації систем підсвічування, розглянуто їхні переваги та недоліки, а також виявлено можливі обмеження, які можуть виникати під час впровадження подібних рішень.

У ході виконання роботи було реалізовано програмно-апаратну частину системи керування світлодіодною стрічкою, яка дозволяє користувачеві створювати комфортні умови для роботи за комп'ютером, знижувати навантаження на зір і покращувати візуальне сприйняття простору. Програмна частина реалізована мовою програмування C++ у середовищі Arduino IDE. Для управління використовувалися апаратні засоби платформи Arduino та зовнішні компоненти.

У результаті розроблено функціональний прототип пристрою, який пройшов експериментальне тестування з метою перевірки його ефективності та зручності у використанні.

ABSTRACT

The report on pre-certification practice contains 54 pages, 14 figures, 19 references.

ARDUINO, ILLUMINATION, RGB, WS2811, LED, VS1838B, LIGHT EFFECTS.

The purpose of the qualification work is to develop a microcontroller system for dynamic backlighting of personal computer monitors based on Arduino using an addressable RGB LED strip. The work analyzes existing approaches to the implementation of backlighting systems, considers their advantages and disadvantages, and identifies possible limitations that may arise during the implementation of such solutions.

In the course of the work, the software and hardware part of the LED strip control system was implemented, which allows the user to create comfortable conditions for working at a computer, reduce the load on eyesight and improve visual perception of space. The software part is implemented in the C++ programming language in the Arduino IDE environment. The Arduino platform hardware and external components were used for control.

As a result, a functional prototype of the device was developed and experimentally tested to verify its effectiveness and ease of use.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

DIY – підхід, за якого людина самостійно виконує ремонтні, технічні, творчі або інші проєкти без залучення фахівців.

EEPROM – тип енергонезалежної пам’яті, що може багаторазово стиратися і записуватися за допомогою електричних імпульсів.

HSV – модель кольору, що описує колір за допомогою трьох параметрів: відтінку (Hue), насиченості (Saturation) та яскравості (Value).

LED – напівпровідниковий прилад, що випромінює світло при проходженні через нього електричного струму.

RGB – колірна модель, яка використовується для відтворення кольорів шляхом поєднання трьох базових кольорів.

RGBW – розширена колірна модель RGB з додаванням білого світлодіода для покращення передачі кольорів.

SRAM – тип енергозалежної пам’яті, яка зберігає дані доти, доки подається живлення, без потреби в періодичному оновленні, як у DRAM.

USB – стандартний інтерфейс для з’єднання комп’ютерів з периферійними пристроями.

ІЧ пульт – інфрачервоний пульт; пристрій дистанційного керування, що використовує інфрачервоне випромінювання для передачі сигналів.

КЗС (комп’ютерний зоровий синдром) – комплекс симптомів, що виникають у користувачів комп’ютерної техніки внаслідок тривалого зорового навантаження на близькій відстані.

ШІМ – широтно-імпульсна модуляція, метод управління потужністю в електронних пристроях.

ВСТУП

Сучасні технології активно впроваджуються у сферу персоналізації робочого простору, створюючи комфортні умови для роботи, відпочинку чи розваг. Однією з таких технологій є динамічне підсвічування монітора, яке покращує естетичне сприйняття, знижує напруження очей і створює атмосферу, що гармонійно доповнює візуальний контент. Тривала робота за комп'ютером часто призводить до зорового навантаження, що викликає втому, сухість і подразнення очей. Використання фонової підсвітки допомагає знизити контраст між яскравістю екрану та темним оточенням, що підвищує комфорт під час роботи. Завдяки доступності мікроконтролерів, таких як Arduino, створення систем динамічного підсвічування стало можливим навіть для новачків у сфері електроніки. Ці пристрої забезпечують широкі можливості для управління світлодіодними стрічками, забезпечуючи інтерактивність та індивідуальність у підсвічуванні.

Метою проєкту є розроблення системи підсвічування монітора з різноманітними світловими ефектами, такими як хвилі, мерехтіння та поступове затухання. Особливістю системи є можливість керування режимами за допомогою інфрачервоного пульта або клавіатури.

Розробка передбачає використання мікроконтролера Arduino для керування RGB-світлодіодною стрічкою WS2811, мови C++ у середовищі Arduino IDE та створення простого інтерфейсу для користувача. Інтеграція апаратної та програмної частини забезпечує динамічне управління підсвіткою в реальному часі.

Проект актуальний у контексті персоналізації робочого простору, сприяє підвищенню зручності та естетичної привабливості монітора і демонструє можливості використання мікроконтролерів у практичних застосунках.

1 АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМ ЗОВНІШНЬОГО ПІДСВІЧУВАННЯ МОНІТОРІВ ПК

1.1 Призначення та вплив зовнішнього підсвічування моніторів

Зовнішнє підсвічування моніторів, часто реалізоване у вигляді світлодіодних стрічок або спеціальних ламп, розташованих позаду або навколо дисплея, набуває все більшої популярності не лише як елемент декору, але і як засіб підвищення комфорту та продуктивності користувача. Його вплив є багатогранним, охоплюючи фізіологічні аспекти зору, психологічне сприйняття та загальну ергономіку робочого простору.

Однією з ключових переваг зовнішнього підсвічування, зокрема так званого bias lighting, є зниження втоми очей. При тривалій роботі за монітором, особливо в умовах недостатнього освітлення приміщення, виникає значний контраст між яскравістю екрана та темним оточенням. Це змушує очі користувача постійно адаптуватися, що призводить до напруження очних м'язів, сухості, подразнення та, як наслідок, до розвитку комп'ютерного зорового синдрому. Bias lighting, створюючи м'яке, розсіяне світло за монітором, зменшує цей контраст, забезпечуючи більш плавний перехід яскравості для очей[1]. Це, в свою чергу, знижує амплітуду акомодаційних зусиль очей, оскільки зіниці не змушені так різко звужуватися та розширюватися при переведенні погляду з екрана на навколишні предмети і навпаки. Дослідження показують, що оптимальне освітлення робочого місця, включаючи фонове, може суттєво зменшити симптоми КЗС[2].

Другою важливою перевагою є покращення сприйняття контрастності зображення на екрані. Коли позаду монітора є джерело світла, чорні та темні ділянки зображення на екрані здаються візуально глибшими та насиченішими. Це відбувається тому, що фонове світло створює точку відліку для сприйняття яскравості, дозволяючи очам краще розрізняти

градації темних відтінків на самому екрані[3].

Естетичний аспект також має важливе значення. Сучасні системи підсвічування, зокрема з використанням RGB світлодіодів, надають змогу створювати унікальну атмосферу для роботи чи гри, підкреслювати стиль інтер'єру та відображати індивідуальність користувача. Можливість налаштування кольору, яскравості та динамічних світлових ефектів перетворює монітор із простого пристрою для виведення інформації на ключовий елемент візуального оформлення простору.

Існують наукові дослідження, які підтверджують вплив освітлення на працездатність та психоемоційний стан людини. Колірна температура і рівень яскравості освітлення можуть значною мірою впливати на концентрацію, відчуття бадьорості та загальний настрій. Наприклад, холодне біле світло в межах 5000-6500K часто асоціюється з підвищеною увагою та ефективністю, тоді як тепле світло в діапазоні 2700-3500K сприяє релаксації та створенню комфортної атмосфери[4]. Динамічне RGB-підсвічування також здатне позитивно впливати на емоційне занурення під час ігор або перегляду медіа. Водночас слід пам’ятати, що неправильно підібрані кольори або надмірна яскравість можуть відволікати, спричиняти втому або викликати дискомфорт. Зокрема, дослідження показують, що м’яке зелене світло сприймається сприятливо й асоціюється з толерантністю, тоді як інтенсивне червоне світло може провокувати напруження та викликати агресивні емоції[5].

Окремо варто згадати про вплив синього світла, що випромінюється екранами моніторів, на циркадні ритми та якість сну. Доведено, що синє світло, особливо у вечірній час, пригнічує вироблення мелатоніну – гормону, відповідального за регуляцію сну[6]. Хоча зовнішнє підсвічування безпосередньо не зменшує кількість синього світла від самого монітора, воно може опосередковано сприяти більш комфортним умовам для вечірньої роботи, зменшуючи загальну зорову напругу та, можливо, дозволяючи використовувати нижчу яскравість екрану. Деякі системи підсвічування

також пропонують режими з теплим світлом, які є більш сприятливими для вечірнього часу.

Таким чином, зовнішнє підсвічування моніторів є багатофункціональною технологією, що позитивно впливає на зоровий комфорт, сприйняття зображення, естетику робочого місця та потенційно на продуктивність і настрій користувача. Фізіологічною основою зменшення втоми очей є стабілізація рівня освітленості навколо екрану, що мінімізує навантаження на акомодаційний апарат ока. При цьому, психологічний вплив, особливо кольорового підсвічування, потребує зваженого підходу до налаштувань для досягнення позитивного ефекту.

1.2 Класифікація систем зовнішнього підсвічування

Системи зовнішнього підсвічування моніторів персональних комп’ютерів можна класифікувати за призначенням, типом джерел світла та функціональністю. Таке класифікування допомагає краще зрозуміти потреби користувача й обрати або розробити найбільш відповідне рішення.

Одним із найпростіших варіантів є статичне фонове освітлення (bias lighting), яке створює рівномірне та незмінне світло позаду монітора. Основна мета такого підсвічування – зменшення зорової втоми за рахунок зниження контрасту між яскравістю екрана та темним фоном, а також покращення візуального сприйняття зображення[3]. Зазвичай для цього використовуються однотонні світлодіодні стрічки білого кольору з температурою світла близько 6500K, що відповідає денному освітленню та забезпечує точну передачу кольорів. У деяких випадках також застосовуються спеціалізовані лампи.

Такий тип підсвічування вирізняється простотою встановлення, ефективністю в забезпеченні зорового комфорту та доступною вартістю. Однак він має обмежені естетичні можливості: колір та інтенсивність світла зазвичай залишаються постійними або регулюються вручну в незначному

діапазоні. Використання теплого або кольорового світла може викривляти сприйняття кольорів на екрані, що знижує точність візуального відображення.

Динамічне декоративне підсвічування спрямоване на створення візуальних ефектів, індивідуалізацію робочого простору та формування певної атмосфери. Таке підсвічування може синхронізуватися з музикою, подіями в іграх або просто відображати різноманітні світлові анімації. Для реалізації зазвичай використовуються багатоколірні RGB або RGBW світлодіодні стрічки, де останній варіант забезпечує кращу передачу білого кольору.

Керування здійснюється за допомогою спеціальних контролерів, що дають змогу змінювати кольори, режими світіння та регулювати яскравість. Такий тип підсвічування відзначається високим рівнем кастомізації й приваблює можливістю створення унікального візуального середовища. Однак за надмірної яскравості або занадто швидких ефектів воно може відволікати та навіть спричиняти зоровий дискомфорт. На відміну від bias lighting, основний акцент у таких системах робиться не на комфорт для очей, а на декоративність та емоційне враження.

Адаптивне підсвічування відрізняється здатністю динамічно змінювати колір і яскравість освітлення відповідно до вмісту на екрані в реальному часі. Такий підхід створює ефект візуального розширення меж дисплея, посилюючи відчуття занурення під час перегляду фільмів або гри. Реалізація може бути як комерційною – наприклад, за допомогою систем Philips Hue Sync чи Govee Immersion, – так і самостійною, з використанням мікроконтролерів Arduino або одноплатних комп’ютерів Raspberry Pi у поєднанні з програмним забезпеченням на зразок Prismatik або Hyperion [7].

Цей тип підсвічування забезпечує найвищий рівень занурення та візуально доповнює зображення за межами екрану. Водночас реалізація DIY-проектів потребує технічних навичок, зокрема налаштування захоплення та аналізу відеосигналу, що може викликати затримки між зміною зображення

та реакцією підсвітки. Комерційні рішення також мають високу вартість.

Спеціалізовані моніторні лампи – це пристрої, що встановлюються на верхню частину монітора для освітлення робочого простору без утворення відблисків на екрані. Завдяки асиметричній оптичній конструкції світло спрямовується вниз і вперед, що дозволяє рівномірно підсвічувати клавіатуру, документи чи стільницю. Прикладами є BenQ ScreenBar, Xiaomi Mi Computer Monitor Light Bar та подібні моделі[8].

Такі лампи покращують ергономіку, зменшують кількість тіней і допомагають ефективно організувати освітлення без зайвого навантаження на очі. Водночас вони мають обмежену можливість регулювання напряму світла, можуть бути несумісними з певними типами моніторів, особливо з вигнутими або з нестандартною товщиною рамок, а якісні моделі мають відносно високу ціну. Важливо зазначити, що ці пристрої не є заміною для фонової підсвітки за монітором, а виконують допоміжну функцію.

Вибір конкретного типу системи підсвічування значною мірою залежить від пріоритетів користувача. Якщо головна мета – зниження зорової втоми при мінімальних витратах, статичне bias lighting буде оптимальним. Для створення атмосфери та візуальних ефектів підійде динамічне декоративне підсвічування. Якщо ж потрібне максимальне занурення в контент, варто розглядати адаптивні системи, хоча вони є складнішими та дорожчими. Спеціалізовані моніторні лампи вирішують завдання локального освітлення робочої зони. Проект, що розробляється в рамках даної роботи, з використанням Arduino та RGB стрічки з керованими ефектами, найбільше відповідає категорії динамічного декоративного підсвічування, з можливістю реалізації статичного режиму, що може виконувати функції bias lighting та також має функцію адаптивної підсвітки.

2 ВИБІР АПАРАТНИХ КОМПОНЕНТІВ СИСТЕМИ

2.1 Мікроконтролерна платформа Arduino

Arduino Uno R3 (рисунок 2.1) є однією з найпопулярніших і найвідоміших плат у лінійці Arduino. Вона побудована на основі 8-бітного мікроконтролера ATmega328P, розробленого компанією Microchip (раніше Atmel)[9].

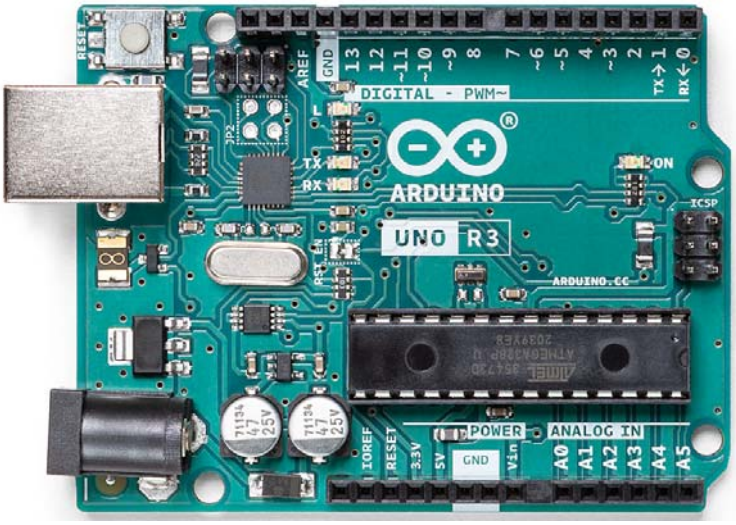

Рисунок 2.1 – Плата Arduino Uno R3

Плата працює при напрузі 5 В, а живлення може подаватися через роз’єм живлення або пін Vin у діапазоні 7–12 В. Гранично допустимі значення становлять від 6 до 20 В, проте при напрузі нижче 7 В стабілізатор може не забезпечити стабільні 5 В, а при перевищенні 12 В можливе його перегрівання. Arduino Uno R3 має 14 цифрових пінів вводу/виводу, шість із яких підтримують широтно-імпульсну модуляцію (ШІМ). Окрім того, є шість аналогових входів (A0–A5), які також можуть функціонувати як цифрові пини[9].

Максимально рекомендований струм на один пін становить 20 мА,

хоча короткочасно допускається до 40 мА. Вихід 3.3 В може забезпечувати струм до 50 мА. Для збереження програм передбачено 32 КБ flash-пам’яті, з яких 0.5 КБ використовується завантажувачем. Обсяг оперативної пам’яті (SRAM) становить 2 КБ, а енергонезалежної пам’яті (EEPROM) – 1 КБ. Плата працює на тактовій частоті 16 МГц, яка задається керамічним резонатором[9].

Плата має USB-з’єднання, що використовується для програмування та серійного зв’язку з комп’ютером. У версії Rev3 цей інтерфейс реалізований за допомогою додаткового мікроконтролера ATmega16U2. Також передбачено роз’єм ICSP для внутрішньосхемного програмування. Arduino Uno R3 підтримує апаратні послідовні інтерфейси UART (піни 0 – RX та 1 – TX), SPI (піни 10 – SS, 11 – MOSI, 12 – MISO, 13 – SCK) та I2C (піни A4 – SDA та A5 – SCL). На платі також є вбудований світлодіод, підключений до піна 13[9].

Arduino Uno вирізняється простотою у використанні, широкою підтримкою в спільноті, наявністю великої кількості навчальних матеріалів і прикладів. Завдяки цьому вона є чудовим вибором для освітніх цілей, прототипування та ознайомлення з мікроконтролерною технікою. Стандартні роз’єми дозволяють легко підключати різноманітні шилди та модулі. Однак у проектах, що потребують великої обчислювальної потужності або значних обсягів, ресурсів ATmega328P може бути недостатньо.

2.1.1 Порівняльний аналіз аналогів Arduino Uno для задач керування адресними світлодіодами

При виборі мікроконтролерної платформи для керування адресними світлодіодами, важливо враховувати не лише базові технічні характеристики плати, а й конкретні потреби самого проекту.

Arduino Nano (рисунок 2.2) є однією з найпопулярніших і доступних плат, яка базується на тому ж мікроконтролері ATmega328P, що й Arduino

Uno.

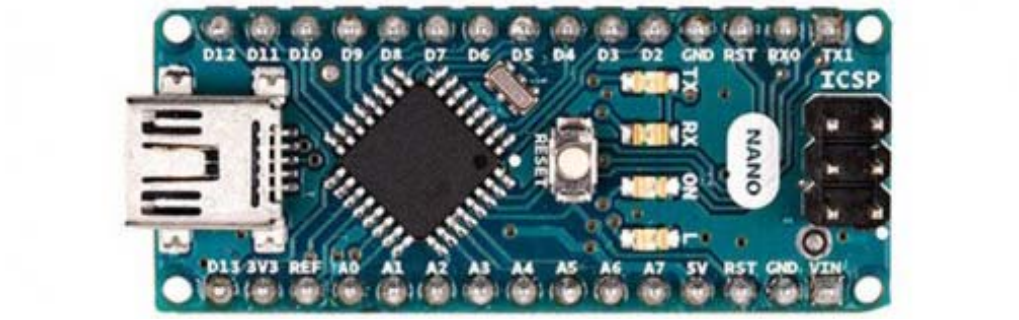

Рисунок 2.2 – Плата Arduino Nano

Вона має ідентичний обсяг пам’яті та кількість пінів, включаючи підтримку ШІМ, і працює на частоті 16 МГц. Основна відмінність полягає в розмірах: Nano значно компактніша і краще підходить для проєктів, де критичним є розмір пристрою або необхідний монтаж на макетній платі. Також Nano має більше аналогових входів, а живлення подається або через USB, або через пін Vin, оскільки окремого роз'єму живлення на платі не передбачено. Завдяки меншій вартості і компактності, Nano чудово підходить для невеликих пристроїв, де потрібне просте керування світлодіодною стрічкою та, наприклад, обробка сигналу з ІЧ-пульта[10].

У випадках, коли проект потребує більшої потужності, складніших ефектів або бездротового керування, доцільно звернути увагу на більш сучасні мікроконтролери – ESP8266 або ESP32 від компанії Espressif (рисунок 2.3). ESP8266 це 32-бітна платформа з вбудованим Wi-Fi, яка працює на частоті до 160 МГц і має значно більше пам’яті порівняно з Arduino. Хоча кількість доступних пінів тут менша, її вистачає для більшості простих задач. ESP32 є ще більш потужним рішенням – з двома ядрами, підтримкою Wi-Fi і Bluetooth, великим обсягом пам’яті та розширеними можливостями вводу-виводу, що дозволяє реалізовувати складні й масштабні проекти зі світлодіодами.

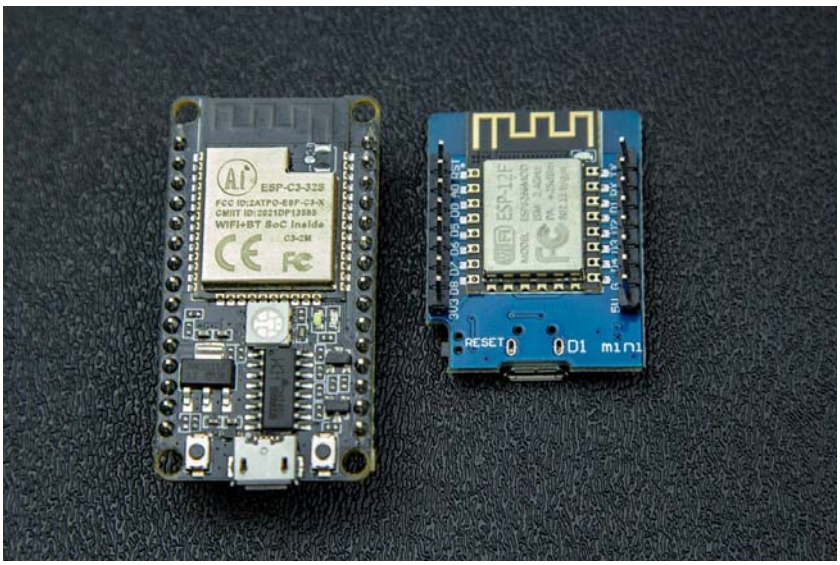

Рисунок 2.3 – Мікроконтролери ESP32 та ESP8266

Висока продуктивність ESP32 дає змогу створювати складніші світлові ефекти, керувати великою кількістю світлодіодів і швидко обробляти сигнали. Крім того, підтримка Wi-Fi та Bluetooth відкриває шлях до реалізації бездротового керування – наприклад, через мобільний додаток або веб-інтерфейс. Водночас варто враховувати, що ESP8266 і ESP32 працюють з логічними рівнями 3.3 В, тому при підключенні до 5-вольтових світлодіодів іноді потрібно використовувати перетворювач рівнів для стабільної роботи. Хоча на коротких відстанях система часто працює і без нього[11].

Програмування ESP8266/ESP32 дещо складніше, особливо через необхідність враховувати роботу Wi-Fi або FreeRTOS (у випадку ESP32), але Arduino IDE з відповідними бібліотеками робить цей процес більш доступним навіть для новачків[11].

Якщо проект не потребує бездротового зв’язку, а кількість світлодіодів і складність керування не перевищують можливості Arduino Uno чи Nano, то вибір більш продуктивної плати може бути зайвим. Проте якщо є плани розширити функціональність пристрою, додати віддалене керування або синхронізацію з онлайн-сервісами – доцільно обрати ESP32, який має значно більший ресурс для подальшого розвитку. У таблиці 2.1 наведено порівняння

основних технічних характеристик плат що розглядаются для розробки проекту.

Таблиця 2.1 – Порівняння мікроконтролерних платформ

Параметр	Arduino Uno R3	Arduino Nano	ESP32
Мікроконтролер	ATmega328P	ATmega328P	Dual-core Tensilica LX6
Архітектура	8-bit	8-bit	32-bit
Тактова частота	16 MHz	16 MHz	до 240 MHz
Flash-пам'ять	32 KB	32 KB	4 MB - 16 MB (залежить від модуля)
SRAM	2 KB	2 KB	520 KB
EEPROM	1 KB	1 KB	4KB (емулюється у Flash) або відсутня
Цифрові піни вводу/виводу	14 (6 ШІМ)	14 (6 ШІМ) (деякі версії до 22)	~34 (більшість з ШІМ)
Аналогові піни вводу	6	8	до 18 (12-bit АЦП)
Wi-Fi	Ні (потрібен шилд)	Ні (потрібен шилд)	Вбудований 802.11 b/g/n
Bluetooth	Ні (потрібен шилд)	Ні (потрібен шилд)	Вбудований Bluetooth v4.2 BR/EDR та BLE
Логічні рівні	5V	5V	3.3V
Робоча напруга	5V	5V	3.3V
Рекомендована вхідна напруга	7-12V	7-12V (Vin) / 5V (USB)	5V (USB) / 3.3V
Орієнтовна ціна (оригінал/клон)	~$25 / ~$5-10	~$20 / ~$3-7	~$7-15 (модуль)

2.2 Світлодіодні стрічки для реалізації підсвічування

Світлодіодні стрічки бувають трьох основних типів – одноколірні, неадресні RGB та адресні RGB. Кожен тип має свої особливості, переваги й недоліки. Одноколірні стрічки світять лише одним постійним кольором, найчастіше білим. Вони можуть мати різну колірну температуру – від теплого до холодного відтінку. Такі стрічки зазвичай використовують для м’якого фону, наприклад, для створення комфортного підсвічування за монітором, яке зменшує навантаження на очі під час роботи або перегляду контенту. Їхні головні переваги – це простота використання та невисока ціна. Для роботи часто достатньо просто подати живлення, а для регулювання яскравості – використовувати димер на основі ШІМ. Однак, через фіксований колір, такі стрічки мають обмежені декоративні можливості.

Неадресні RGB стрічки вже здатні змінювати колір світіння, проте всі світлодіоди на стрічці змінюють колір одночасно. Керування відбувається через три канали (червоний, зелений, синій) і зазвичай здійснюється за допомогою RGB-контролера або мікроконтролера з трьома ШІМ-виходами. До переваг таких стрічок належить можливість налаштовувати будь-який колір для всієї стрічки, а також відносна простота в керуванні. Водночас, їхній недолік – це неможливість керувати окремими ділянками, тому створити динамічні ефекти, наприклад, рух світла або анімацію, з ними не вдасться[12].

Адресні RGB стрічки, або піксельні, дають змогу керувати кожним світлодіодом окремо. Це стало можливим завдяки вбудованим мікросхемам, які обробляють сигнали з мікроконтролера. У результаті можна створювати надзвичайно гнучкі ефекти – рух хвиль, переливи, градієнти, анімації тощо. Такі стрічки ідеально підходять для творчих, інтерактивних проектів, хоча й вимагають складнішої реалізації. Потрібен мікроконтролер і спеціальні бібліотеки для передачі даних, а вартість вища порівняно з попередніми

варіантами. Проте для системи підсвічування з динамічними ефектами – саме вони є оптимальним вибором.

2.2.1 Детальний огляд адресних світлодіодних стрічок

Існує кілька популярних типів адресних світлодіодних стрічок, кожен із яких має свої особливості, пов’язані з принципом роботи, протоколом керування, напругою живлення та іншими технічними характеристиками.

Одним із найпоширеніших варіантів є стрічки на базі WS2811 (рисунок 2.4). Це мікросхема-драйвер, яка не є частиною самого світлодіода, а розташована окремо та керує зовнішніми RGB-світлодіодами. У варіантах на 12 вольт один такий драйвер зазвичай відповідає за три послідовно з'єднані світлодіоди. Це означає, що мінімальний керований блок складається з трьох світлодіодів, які світяться однаково[13].

Рисунок 2.4 – WS2811 Neopixel LED стрічка

Протокол керування в цьому випадку однопровідний та асинхронний, вимагає дуже точного дотримання часових параметрів сигналів, оскільки дані передаються по ланцюгу від одного модуля до наступного[14]. Перевагою є можливість використання 12V живлення, що зменшує падіння напруги на довгих стрічках і спрощує підключення, а також відносно низька вартість. Основним недоліком є те, що один контролер керує одразу кількома

світлодіодами, що знижує точність ефектів, і у випадку виходу з ладу одного елемента може перестати працювати вся ланка[13].

Вибір стрічки WS2811 для реалізації проекту обумовлений кількома вагомими перевагами. По-перше, керування здійснюється одночасно над трьома світлодіодами, об'єднаними в один блок під контролем мікросхеми WS2811. Це дозволяє значно знизити навантаження на пам'ять мікроконтролера, оскільки кількість керованих елементів зменшується втричі порівняно з адресними стрічками, де кожен світлодіод керується окремо. По-друге, така організація даних дає можливість більш ефективно використовувати ресурси мікроконтролера, звільняючи пам’ять для реалізації більшої кількості складних світлових ефектів і анімацій. По-третє, WS2811 працює від напруги 12 В, що зменшує падіння напруги на довгих ділянках стрічки, забезпечуючи більш стабільне і рівномірне світло без додаткових заходів по компенсації. Таким чином, стрічка WS2811 оптимально відповідає вимогам проєкту, поєднуючи зручність керування, ефективність використання ресурсів та стабільність роботи.

Більш просунутий варіант – це WS2812B (рисунок 2.5) та його аналоги, як-от SK6812. У цьому випадку мікросхема керування інтегрована безпосередньо в корпус кожного RGB-світлодіода. Завдяки цьому кожен елемент стрічки стає окремим пікселем з можливістю індивідуального керування кольором та яскравістю.

Рисунок 2.5 – Адресний світлодіод WS2812B

Як і у WS2811, передача даних здійснюється по одному каналу, але

система працює з кожним світлодіодом окремо, що відкриває значно більше можливостей для створення складних анімацій та світлових ефектів. Такі стрічки зазвичай працюють від 5 вольт. Однією з особливостей є використання ШІМ на частоті близько 400 Гц для WS2812B, що може викликати мерехтіння, помітне при відеозйомці[13]. У SK6812 (рисунок 2.6) частота ШІМ зазвичай вища – понад 1 кГц, що дозволяє уникнути цієї проблеми. Додатково SK6812 бувають у версіях RGBW, де окремо додано білий світлодіод, що забезпечує чистіший білий колір і ширший діапазон відтінків[15].

Рисунок 2.6 – Адресний світлодіод SK6812

Хоча такі стрічки більш гнучкі у використанні, вони потребують більш уважного підходу до організації живлення, оскільки при великій довжині може спостерігатись помітне падіння напруги. Ще одним недоліком є вразливість до виходу з ладу: якщо один світлодіод перестає працювати, можуть припинити роботу й усі наступні за ним у стрічці.

Іншим цікавим типом є стрічки на базі APA102 (рисунок 2.7) та їхні аналоги. Вони використовують інший принцип керування – синхронний двопровідний інтерфейс, подібний до SPI. Окремо передаються дані та сигнали синхронізації, що значно знижує вимоги до точності таймінгу на мікроконтролері[15].

Рисунок 2.7 – Адресний світлодіод APA102

Це дозволяє досягти вищої швидкості оновлення та стабільності керування, що особливо важливо для додатків із великою кількістю пікселів або візуалізацією відео. Напруга живлення для таких стрічок зазвичай становить 5 вольт. Головною перевагою є дуже висока частота ШІМ – до 20 кГц і більше, що усуває ефект мерехтіння навіть при зйомці на відео або швидкому русі очей. Такі стрічки ідеально підходять для POV-дисплеїв, відеопанелей та інших завдань, де потрібна максимальна якість зображення. Основними недоліками є вища ціна в порівнянні з WS281x, а також необхідність використання двох окремих пінів мікроконтролера для керування.

Таким чином, вибір між різними типами адресних стрічок залежить від вимог до ефектів, довжини стрічки, доступного живлення та чутливості до мерехтіння. WS2811 буде доречним для довгих 12-вольтових систем із простішими ефектами. WS2812B або SK6812 підійдуть, коли потрібна гнучкість і точність керування. Якщо ж проект передбачає запис на відео або високі швидкості оновлення, кращим вибором стануть APA102 чи її аналог SK9822.

2.3 Пристрій для керування системою

Інфрачервоне керування залишається одним із найзручніших і

найдоступніших способів передавання команд на невеликі відстані без використання дротів. Його часто застосовують у саморобних електронних проектах завдяки низькій вартості, простоті реалізації та широкій доступності відповідних компонентів. Принцип роботи полягає в тому, що при натисканні кнопки на пульті він починає випромінювати інфрачервоні імпульси, які модулюються на певній частоті – зазвичай це 36, 38 або 40 кГц. Кожна кнопка генерує унікальну комбінацію імпульсів, яка відповідає певному коду. На приймальному боці, наприклад, при використанні Arduino, встановлюється інфрачервоний приймач VS1838B (рисунок 2.8). Він вловлює ці сигнали, фільтрує частоту модуляції і перетворює отримані дані у цифрові сигнали, які легко обробити мікроконтролером[16].

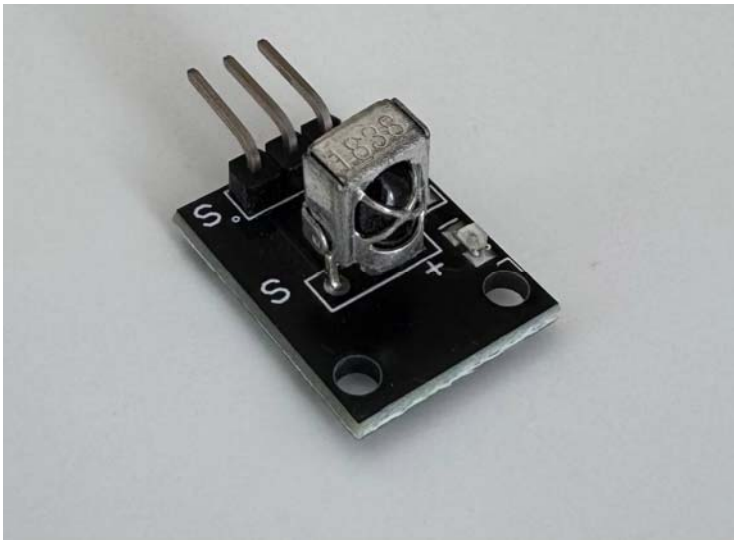

Рисунок 2.8 – Універсальний інфрачервоний приймач VS1838B

Інфрачервоний пульт дистанційного керування (рисунок 2.9) містить набір кнопок для виконання різних команд, таких як зміна режиму роботи, регулювання яскравості або увімкнення/вимкнення пристрою.

Рисунок 2.9 – ІЧ пульт дистанційного керування

Сигнали передаються за допомогою різних протоколів, що визначають структуру переданих даних. Найбільш поширеними є протоколи NEC, Sony SIRC, Philips RC-5 і RC-6, JVC, Samsung. Кожен з них має свою структуру, спосіб кодування сигналів та частоту модуляції. У більшості недорогих універсальних пультів використовується саме протокол NEC.

Кожна кнопка пульта має унікальний код, який дозволяє однозначно ідентифікувати команду, що була передана. Завдяки цьому мікроконтролер може точно розпізнати дію користувача і виконати відповідну функцію. Така унікальність кодів запобігає випадковим помилковим командам і забезпечує надійність керування системою.

Основною перевагою такого способу керування є його простота та низька вартість. Пульти й приймачі коштують зовсім недорого, легко підключаються до Arduino, а для роботи з ними існує багато готових бібліотек, наприклад IRremote, які значно спрощують розробку. Окрім того, ІЧ керування є енергоефективним і не потребує складних радіочастотних компонентів, що робить його ідеальним вибором для невеликих пристроїв з обмеженим енергоспоживанням. Завдяки простоті конструкції пультів, вони

мають тривалий термін служби без необхідності частого обслуговування або заміни батарей. Проте є й недоліки: пульт має бачити приймач, тобто між ними повинна бути пряма видимість. Крім того, дальність дії обмежена – зазвичай до 5–10 метрів, а інші джерела інфрачервоного випромінювання, наприклад сонячне світло або деякі типи ламп, можуть спричинити завади. Незважаючи на це, інфрачервоне керування залишається зручним та ефективним інструментом для створення простих систем дистанційного керування для будь-яких пристроїв.

2.4 Загальна електрична схема пристрою

Для коректної роботи системи підсвічування було розроблено просту, та ефективну схему з’єднань між основними компонентами: мікроконтролером Arduino, адресною світлодіодною стрічкою WS2811 та інфрачервоним приймачем VS1838B (рисунок 2.10).

Сигнал керування від Arduino подається на лінію даних світлодіодної стрічки через один із цифрових виходів плати. Для захисту входу стрічки від можливих стрибків струму використовується резистор з опором близько 220 Ом, що допомагає пом’якшити різкі імпульси і збільшити надійність роботи світлодіодів.

Живлення стрічки здійснюється від зовнішнього блоку живлення з вихідною напругою 12 В і силою струму до 3 А. Це забезпечує стабільне живлення навіть для довгих ділянок стрічки, мінімізуючи падіння напруги та уникнення мерехтіння світлодіодів. Важливо, що мінусовий провід блоку живлення з’єднаний із загальним GND Arduino, що створює спільний референс напруг і забезпечує правильну роботу лінії даних.

Інфрачервоний приймач VS1838B підключений до одного з цифрових входів Arduino, через який він передає розпізнані сигнали з ІЧ пульта. Приймач живиться від стабільного 5 В, що надходить безпосередньо з плати Arduino.

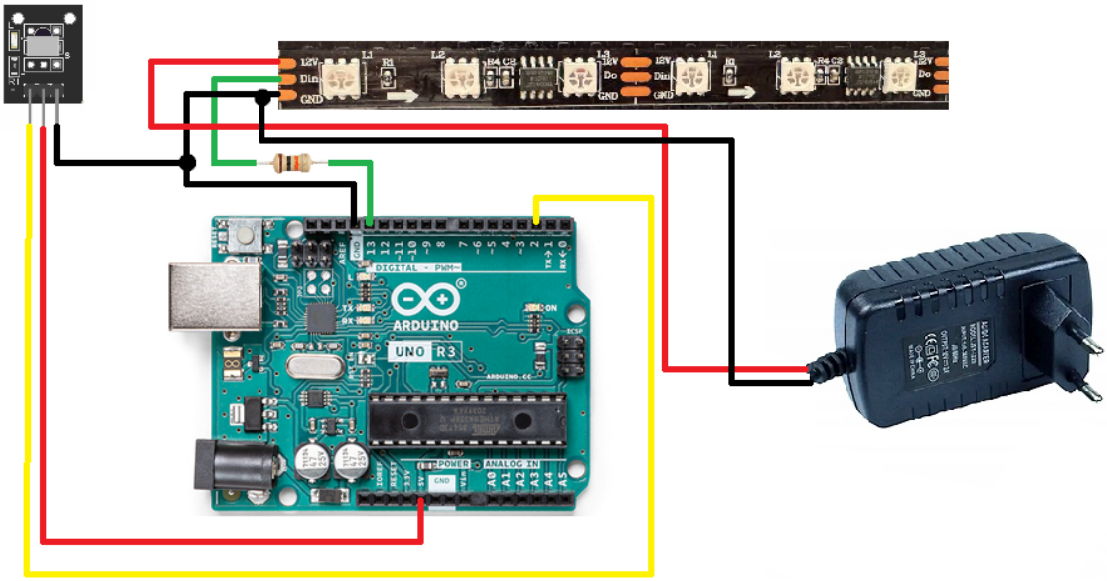

Рисунок 2.10 – Електрична функціональна схема пристрою

Заземлення світлодіодної стрічки підключене одночасно до блоку живлення та плати Arduino для створення спільного електричного потенціалу. Це забезпечує узгодження рівнів сигналів і напруг між пристроями, що мінімізує електричні шуми і перешкоди, а також гарантує стабільну передачу даних і надійну роботу системи.

Така схема підключення компонентів системи дозволяє забезпечити плавне і надійне керування підсвічуванням у реальному часі, а також зручне дистанційне управління за допомогою ІЧ пульта, без зайвих дротів і складних налаштувань.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРИСТРОЮ

3.1 Вибір мови програмування та середовища розробки

Програмна реалізація є ключовим етапом у створенні будь-якого мікроконтролерного пристрою, оскільки саме програмний код визначає логіку його роботи, функціональні можливості та взаємодію з користувачем. Для розробки програмного забезпечення системи динамічного підсвічування було обрано мову програмування C++ у рамках програмної оболонки Arduino та інтегроване середовище розробки Arduino IDE. Такий вибір зумовлений низкою факторів, що забезпечують ефективність розробки та оптимальну роботу пристрою на базі мікроконтролера.

Окрім цього, для реалізації більш складного адаптивного режиму підсвічування, що реагує на контент екрану в реальному часі, було використано мову програмування Python та середовище розробки Visual Studio Code (VS Code). Python надає широкі можливості для роботи із захопленням екрану, обробкою зображень та комунікацією з мікроконтролером через послідовний порт. Середовище VS Code було обране через його багатофункціональність, зручний інтерфейс, велику кількість розширень для роботи з Python та інтеграцію з системами контролю версій, що значно прискорило процес розробки та тестування адаптивного режиму.

3.1.1 Мова програмування C++

Мова C++, що використовується в середовищі Arduino (часто згадується як Arduino Sketch), є потужним інструментом для програмування мікроконтролерів. Вона надає можливості як для низькорівневого доступу до апаратних ресурсів мікроконтролера, що є критичним для керування периферійними пристроями, такими як адресні світлодіодні стрічки та ІЧ-

приймачі, так і для використання високорівневих абстракцій, що спрощують процес розробки[17].

Обрана мова програмування дає змогу ефективно використовувати пам’ять і оптимізувати швидкість виконання коду, а також підтримує об’єктно-орієнтовані підходи. Втім, у проєктах з обмеженими ресурсами, таких як мікроконтролери, цей підхід застосовується з певними застереженнями, щоб не перевантажувати систему. Однією з головних переваг є велика кількість готових бібліотек, які суттєво спрощують роботу з апаратурою і розширюють можливості стандартної платформи Arduino. У випадку даного проекту це, зокрема, бібліотеки для роботи зі світлодіодними стрічками та обробки сигналів інфрачервоного пульта.

Комбінація C++ з простотою фреймворку Arduino дозволяє швидко й гнучко створювати функціональні проекти, не заглиблюючись у складнощі низькорівневого програмування. Це дає змогу зосередитись на реалізації основних завдань, залишаючи технічні деталі керування апаратурою бібліотекам, які були вже ретельно протестовані і оптимізовані.

3.1.2 Мова програмування Python та середовище розробки Visual Studio Code

Для реалізації адаптивного режиму підсвічування, який автоматично аналізує зображення на екрані користувача і відправляє відповідні кольори на світлодіодну стрічку, була використана мова Python. З її допомогою здійснюється зняття скриншотів екрану, виконується обробка їх кольорів за допомогою бібліотек, таких як mss та Pillow, після чого дані передаються по послідовному інтерфейсу на Arduino.

Середовище Visual Studio Code було обране через його потужні можливості редагування коду, підтримку налагодження, інтеграцію з Git та зручність у роботі з Python-проектами. VS Code має широкий набір розширень, які полегшують роботу з Python, а також дозволяють швидко і

зручно тестувати та запускати код.

Поєднання Arduino IDE для вбудованої частини та Python і VS Code для зовнішнього керування дозволило створити комплексну систему адаптивної підсвітки, яка охоплює як апаратну, так і програмну складові, забезпечуючи гнучкість та розширюваність проекту.

3.1.3 Середовище розробки Arduino IDE

Інтегроване середовище розробки Arduino IDE було обрано завдяки його простоті, доступності та повноцінному набору інструментів, необхідних для реалізації проекту. Arduino IDE надає текстовий редактор для написання коду, вбудований компілятор, засоби для завантаження прошивки безпосередньо в пам'ять мікроконтролера, а також монітор послідовного порту, який є незамінним інструментом для налагодження та тестування[18, 19].

Однією з ключових переваг Arduino IDE є його кросплатформеність, що дозволяє працювати над проектом у різних операційних системах (Windows, macOS, Linux). Крім того, величезна спільнота користувачів Arduino забезпечує наявність великої кількості навчальних матеріалів, прикладів коду та форумів, де можна знайти відповіді на більшість питань, що виникають у процесі розробки. Хоча Arduino IDE може здатися дещо спрощеним порівняно з професійними IDE, такими як PlatformIO чи Atmel Studio, для проектів середньої складності, таких як розроблювана система підсвічування, його функціональності цілком достатньо.

PlatformIO, наприклад, пропонує розширені можливості керування проектами, підтримку багатьох платформ і систем контролю версій, але потребує складнішого налаштування і більших знань від розробника. Atmel Studio ж орієнтована на професіоналів і надає глибокий доступ до апаратних ресурсів, але вона більш громіздка і складна для початківців. Arduino IDE поєднує в собі простоту та достатній набір інструментів, що робить його

ідеальним вибором для швидкої розробки і тестування.

Для роботи з проектом у середовищі Arduino IDE необхідно встановити кілька додаткових бібліотек. Відкривши менеджер бібліотек, користувач може легко знайти та встановити необхідні пакети, зокрема бібліотеки FastLED для керування світлодіодними стрічками та IRremote для роботи з інфрачервоним приймачем (рисунок 3.1).

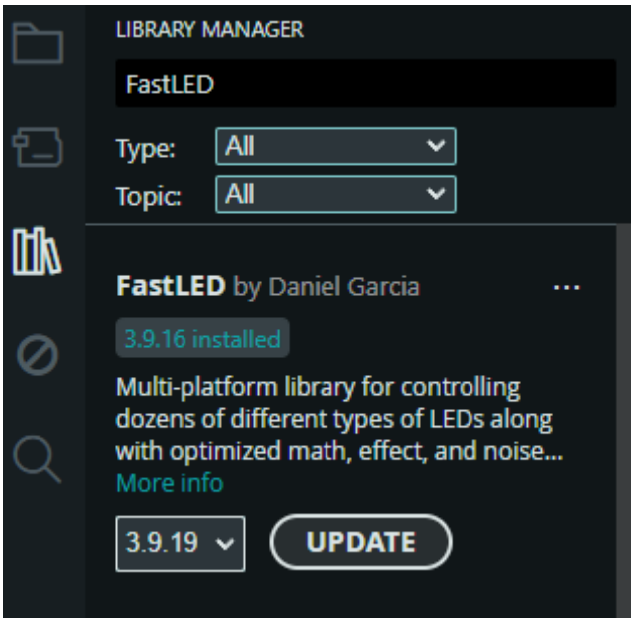

Рисунок 3.1 – Менеджер бібліотек Arduino IDE

Цей процес є інтуїтивно зрозумілим і не вимагає складних налаштувань, що значно пришвидшує підготовку до розробки.

3.2 Загальна структура та опис коду проекту

Програмне забезпечення системи динамічного підсвічування розроблене таким чином, щоб забезпечити гнучке управління світловими ефектами, їх параметрами та оперативну реакцію на команди користувача, які надходять із інфрачервоного пульта. Основу архітектури складає поєднання двох підходів: подієвої обробки сигналів та реалізації скінченного

автомата. Подієва частина відповідає за прийом і обробку команд, що надходять з інфрачервоного пульта. Після отримання команди система аналізує її та змінює свій внутрішній стан – наприклад, переключає світловий ефект, регулює яскравість чи швидкість анімації.

Реалізація скінченного автомата здійснюється через механізм режимів роботи Mode, де кожен режим відповідає конкретному світловому ефекту. Перемикання між режимами відбувається за допомогою команд з ІЧ-пульта. Основний цикл програми loop() безперервно перевіряє наявність нових команд та виконує функції, відповідні активному режиму, для генерації та відображення світлових ефектів. Такий підхід гарантує одночасну чутливість системи до дій користувача та безперервне відтворення візуальних ефектів, що забезпечує плавність анімації і швидку реакцію на зміну налаштувань, що важливо для комфортної роботи з пристроєм.

3.2.1 Підключення бібліотек

Для реалізації функціоналу системи використовуються дві основні зовнішні бібліотеки. Директива препроцесора #include <FastLED.h> включає до проекту бібліотеку FastLED. Це потужна та високоефективна бібліотека, призначена для керування різними типами адресних світлодіодних стрічок, включаючи WS2811, що використовується в даному проекті. Вона надає широкий набір функцій для роботи з кольорами, створення анімацій, керування яскравістю та багато іншого, значно спрощуючи процес програмування складних світлових ефектів.

Інша директива #include <IRremote.hpp> підключає бібліотеку IRremote, яка забезпечує функціонал для прийому та декодування сигналів, що надходять від інфрачервоних пультів дистанційного керування. Ця бібліотека підтримує велику кількість поширених ІЧ-протоколів, що дозволяє використовувати стандартні пульти для керування системою.

Використання цих бібліотек дозволяє абстрагуватися від

низькорівневих деталей взаємодії з апаратним забезпеченням та зосередитися на логіці роботи самої системи підсвічування.

3.2.2 Визначення констант та глобальних змінних

Для налаштування та керування роботою системи в програмному коді визначено низку констант і глобальних змінних. Константи використовуються для конфігурації світлодіодної стрічки. Зокрема, LED_PIN визначає цифровий пін мікроконтролера Arduino, до якого підключено лінію даних (Data In) стрічки. Значення NUM_LEDS вказує загальну кількість світлодіодів у стрічці, якою керує програма. Тип мікросхем, що керують світлодіодами, задається константою LED_TYPE зі значенням WS2811, що необхідно для коректної ініціалізації бібліотеки FastLED. Параметр COLOR_ORDER визначає порядок колірних каналів для конкретного типу стрічки; неправильне встановлення цього значення призводить до некоректного відображення кольорів що випливає у неправильне функціонування усієї системи.

Для ідентифікації команд, які надходять від інфрачервоного пульта, у коді визначено унікальні константи, що відповідають шістнадцятковим кодам кнопок пульта. У таблиці 3.1 наведено перелік цих констант та їх функціональне призначення, що допомагає зрозуміти інтерфейс керування системою. Користувач, натискаючи кнопки пульта, надсилає ці коди, які програма інтерпретує для виконання відповідних дій.

Таблиця 3.1 – Коди кнопок ІЧ-пульта та їх призначенн

Команда	Шістнадцятковий код	Призначення
IR_CODE_UP	0xE718FF00	Збільшити яскравість
IR_CODE_DOWN	0xAD52FF00	Зменшити яскравість

IR_CODE_RIGHT	0xA55AFF00	Збільшити швидкість ефекту
IR_CODE_LEFT	0xF708FF00	Зменшити швидкість ефекту
IR_CODE_OK	0xE31CFF00	Змінити колір
IR_CODE_1	0xBA45FF00	Режим RAINBOW
IR_CODE_2	0xB946FF00	Режим GRADIENT
IR_CODE_3	0xB847FF00	Режим STATIC_COLOR
IR_CODE_4	0xBB44FF00	Режим RUNNING_LIGHTS
IR_CODE_5	0xBF40FF00	Режим BLINKING
IR_CODE_6	0xBC43FF00	Режим COLOR_WAVES
IR_CODE_7	0xF807FF00	Режим COLOR_WIPE
IR_CODE_8	0xEA15FF00	Режим BREATHING
IR_CODE_9	0xF609FF00	Режим RANDOM_FLICKER
IR_CODE_0	0xE619FF00	Режим OFF
IR_CODE_STAR	0xE916FF00	Режим AMBIENT LIGHT
IR_CODE_HASH	0xF20DFF00	Не використовується

Глобальні змінні включають масив leds типу CRGB розміром NUM_LEDS, який зберігає інформацію про колір кожного світлодіода у форматі RGB. Поточний колір зберігається у змінній currentColor, яка ініціалізується синім кольором. Палітра доступних кольорів визначена масивом colorPalette, а індекс поточного кольору – змінною currentColorIndex. Значення яскравості керується змінною currentBrightness, що приймає значення від 0 до 255, і ініціалізується на рівні 127, що дорівнює середній яскравості яку може відобразити світлодіодна стрічка WS2811. Швидкість анімації контролюється змінною effectSpeed, яка задає затримку в

мілісекундах.

Перелік усіх доступних режимів роботи системи визначено перерахуванням Mode, що включає світлові ефекти та режим вимкнення. Поточний режим зберігається у змінній currentMode, яка за замовчуванням встановлена у стан OFF. Для роботи з інфрачервоним приймачем визначено пін IR_RECEIVE_PIN зі значенням 2, а також створено об’єкт irrecv класу IRrecv для прийому та обробки сигналів ІЧ-пульта.

3.2.3 Функція ініціалізації системи

Функція setup() що зображена у лістингу 3.1 виконується один раз при запуску або перезавантаженні мікроконтролера. Її основне призначення – виконати всі необхідні початкові налаштування апаратного та програмного забезпечення.

Лістинг 3.1 – Функція ініціалізації системи setup()

```
void setup() {
  Serial.begin(250000);
  FastLED.addLeds<LED_TYPE,    LED_PIN,    COLOR_ORDER>(leds,
NUM_LEDS);
  FastLED.setBrightness(127);
  irrecv.enableIRIn();

  for (int i = 0; i < NUM_LEDS; i++) {
    glitchDuration[i] = 0;
  }
}
```

У коді спочатку відкривається послідовний порт зі швидкістю 9600 біт/с за допомогою Serial.begin(9600);. Це дозволяє виводити повідомлення для налагодження в монітор послідовного порту Arduino IDE. Хоча в

готовому пристрої це не є обов’язковим, під час розробки та тестування проекту ця можливість виявилася дуже корисною.

Команда FastLED.addLeds<LED_TYPE, LED_PIN, COLOR_ORDER>(leds, NUM_LEDS) налаштовує бібліотеку FastLED для роботи зі світлодіодною стрічкою. Вона визначає тип стрічки, пін підключення, порядок кольорів, кількість світлодіодів та масив, де зберігатимуться дані про колір.

Далі виконується встановлення початкової яскравості стрічки за допомогою FastLED.setBrightness(currentBrightness), де currentBrightness – це значення від 0 до 255, задане раніше.

Наступним кроком команда irrecv.enableIRIn(); активує інфрачервоний приймач, щоб він міг приймати сигнали з пульта.

Таким чином, функція setup() готує систему до подальшої роботи і забезпечує готовність до прийому команд та керування світлодіодами.

3.2.4 Функція основного циклу програми

Функція loop() є основним циклом програми для мікроконтролера Arduino, який виконується безперервно після запуску функції setup(). У даному проекті loop() виконує дві ключові задачі: обробку команд, що надходять із інфрачервоного пульта, та виконання вибраного світлового ефекту.

На початку кожної ітерації циклу викликається метод irrecv.decode(), який перевіряє наявність нових ІЧ-сигналів. Якщо сигнал успішно прийнятий і декодований, функція повертає значення true, а інформація про команду зберігається у змінній decCode. Наступна перевірка виключає повторювані або некоректні сигнали – коди 0xFFFFFFFF та 0 не обробляються, оскільки перший сигнал зазвичай виникає при утриманні кнопки, а другий означає помилку.

При отриманні валідного коду відбувається аналіз за допомогою

послідовності умов else if. За допомогою цих перевірок реалізовано керування яскравістю (кнопки IR_CODE_UP і IR_CODE_DOWN), швидкістю ефектів (кнопки IR_CODE_RIGHT і IR_CODE_LEFT), зміною кольору (IR_CODE_OK), а також вибір режимів роботи світлодіодної стрічки (IR_CODE_1…IR_CODE_9 і IR_CODE_0 для вимкнення). Для обмеження значень яскравості та швидкості використовується функція constrain(), яка не дозволяє виходити за допустимі межі.

Після обробки поточного коду викликається метод irrecv.resume(), що скидає стан приймача і дозволяє отримувати нові сигнали.

Друга частина функції loop() відповідає за виконання активного світлового ефекту, який вибирається у залежності від значення змінної currentMode. Для цього використовується конструкція switch, що викликає відповідну функцію для кожного режиму.

Кожен режим відповідає одному зі значень перелічення Mode. Наприклад, у режимі RAINBOW викликається функція rainbowEffect(). У режимі OFF стрічка заповнюється чорним кольором, що фактично вимикає світлодіоди, стан стрічки оновлюється за допомогою FastLED.show(). Такий підхід забезпечує плавність анімацій і швидку реакцію на зміни, що важливо для комфортної взаємодії з системою.

3.3 Опис використаних бібліотек

Для реалізації програмної частини проекту були обрані дві ключові бібліотеки, які значно спростили розробку та забезпечили необхідний функціонал: FastLED для керування адресними світлодіодами та IRremote для роботи з інфрачервоним пультом.

Бібліотека FastLED – це потужний та гнучкий інструмент для роботи з різноманітними адресними світлодіодними стрічками та окремими світлодіодами, такими як WS2810, WS2811, WS2812 (Neopixel), LPD8806 і багатьма іншими. Вона стала дуже популярною серед розробників завдяки

високій продуктивності, широкому набору функцій і простоті використання.

У проекті FastLED забезпечує нативну підтримку стрічки WS2811, яка використовується для підсвічування. Це гарантує правильне управління кожним світлодіодом. Бібліотека оптимізована для роботи на мікроконтролерах з обмеженими ресурсами, що особливо важливо для створення плавних і складних анімацій. Особливістю FastLED є зручна робота з кольорами у моделі HSV, яка є більш інтуїтивною для людини, ніж традиційна RGB. Це дозволяє, наприклад, легко створювати ефект веселки, плавно змінюючи лише відтінок, при цьому зберігаючи максимальну насиченість і яскравість.

FastLED також пропонує безліч готових функцій для роботи з кольорами та заповнення стрічки різними ефектами, а також оптимізовані математичні функції, які допомагають створювати плавні хвилеподібні переходи, дихання світла та інші динамічні ефекти. Використання цієї бібліотеки дозволило зосередитися на творчій складовій проекту, не витрачаючи час на низькорівневу реалізацію протоколу передачі даних WS2811.

Друга важлива бібліотека – IRremote, яка є стандартним та надійним рішенням для прийому та обробки інфрачервоних сигналів за допомогою Arduino. Вона підтримує широкий спектр популярних ІЧ-протоколів, таких як NEC, Sony SIRC, Philips RC5 і RC6, Samsung, JVC та інші, що робить її універсальною і дозволяє використовувати різні пульти дистанційного керування.

IRremote суттєво спрощує роботу з інфрачервоними сигналами, абстрагуючи складну низькорівневу логіку декодування і надаючи простий інтерфейс для отримання готових команд. Для підключення приймача необхідно створити об’єкт класу IRrecv з вказівкою відповідного піна, ініціалізувати його у функції setup(), а у циклі loop() регулярно перевіряти наявність сигналів.

Хоча IRremote також підтримує роботу на основі переривань для

асинхронного прийому сигналів, у цьому проекті застосовується більш простий полінговий підхід, що цілком відповідає швидкості обробки та вимогам системи.

Завдяки цим бібліотекам, інтеграція функціоналу дистанційного керування і складних світлових ефектів у проект стала значно простішою і надійнішою. Основні зусилля були сконцентровані на реалізацію логіки взаємодії та аспектах дизайну підсвічування, уникаючи потенційних помилок при роботі з низькорівневими протоколами.

Також, у додаток до бібліотек Arduino було використано декілька Python бібліотек. Бібліотека mss була використана для для ефективного та швидкого захоплення скріншотів екрану в реальному часі. Ця бібліотека приваблива тим, що забезпечує кросплатформену роботу на Windows, macOS та Linux і дозволяє захоплювати не лише весь екран, а й вибрані користувачем області, що дуже важливо для роботи системи підсвічування, яка аналізує кольори конкретних зон монітора. mss використовує низькорівневі API операційної системи для максимальної швидкості захоплення, мінімізуючи навантаження на процесор, що дозволяє досягати високої частоти оновлення. Ця бібліотека має простий у використанні інтерфейс, який дозволяє легко вказувати координати зон для скріншотів. У проекті mss використовується для періодичного зняття зображень з визначених зон екрану, які відповідають розташуванню світлодіодів на стрічці, щоб визначити середній колір для кожного блоку світлодіодів стрічки.

Бібліотека Pillow, сучасний форк оригінальної PIL (Python Imaging Library), є потужним інструментом для обробки растрових зображень у Python. Вона надає широкий набір функцій для роботи з графікою і в проекті використовується для конвертації захоплених зображень у формат, зручний для аналізу кольорів, зокрема RGB. Pillow дозволяє обчислювати середній колір кожної вибраної області шляхом аналізу пікселів, що є важливим для точного визначення кольору, який має відтворювати відповідний світлодіод.

Крім того, бібліотека гнучка і швидка, що дозволяє реалізувати ефективний аналіз кольорів у реальному часі, що є основою адаптивного підсвічування.

Бібліотека pyserial є стандартним рішенням для роботи з послідовними портами в Python і дозволяє здійснювати комунікацію між комп'ютером та зовнішніми пристроями, зокрема мікроконтролерами Arduino. У контексті цього проекту pyserial використовується для встановлення стабільного та надійного з'єднання через USB-порт з Arduino, передачі сформованих пакетів кольорових даних у вигляді RGB значень згідно з протоколом, узгодженим із мікроконтролером. Вона забезпечує швидку передачу даних у режимі реального часу, що є необхідним для плавної та синхронної роботи підсвітки. Завдяки цій бібліотеці відбувається інтеграція зовнішньої логіки обробки екрану з апаратним управлінням світлодіодною стрічкою.

Взаємодія цих бібліотек забезпечує безперервне й ефективне сканування екрану, аналіз кольорів і передачу отриманих результатів на Arduino. Завдяки mss відбувається швидке захоплення зображень, Pillow відповідає за їх обробку і перетворення у потрібний формат, а pyserial передає керуючі сигнали на апаратний рівень. Така архітектура дозволяє реалізувати адаптивне підсвічування з високою частотою оновлення, плавністю переходів та мінімальною затримкою між зміною зображення на моніторі та реакцією світлодіодів.

3.4 Префікс Ada у протоколі передачі даних

Для забезпечення надійної та синхронізованої передачі даних між комп'ютером і мікроконтролером Arduino у проекті використовується спеціальний префікс – послідовність символів Ada (ASCII-коди 0x41 0x64 0x61). Цей префікс служить унікальним маркером початку пакету даних, який дозволяє Arduino чітко ідентифікувати, коли починається нова порція інформації.

Застосування такого маркера є необхідним, оскільки послідовний порт передає байти без явних меж між повідомленнями. Префікс Ada гарантує, що мікроконтролер не сплутує початок пакета з випадковими даними чи шумом і приймає лише коректні, повні пакети.

Обрана послідовність символів є простою для реалізації і в той же час достатньо унікальною, щоб зменшити ймовірність помилкової синхронізації. Вона складається з трьох байтів, що забезпечує надійне визначення початку пакета без зайвих накладних витрат на передачу. Крім того, вибір саме Ada має символічне значення, що сприяє зручності підтримки коду і налагодження.

Подібний підхід є стандартною практикою у побудові власних протоколів обміну через послідовний інтерфейс, що особливо важливо для забезпечення цілісності й надійності передачі даних у реальному часі. У поєднанні з додатковою перевіркою контрольної суми це дозволяє створити надійну систему, стійку до шумів і випадкових помилок.

Таким чином, пріоритет префіксу Ada було віддано як оптимальному компромісу між простотою, надійністю та зручністю, що є ключовими факторами для стабільної роботи режиму адаптивної підсвітки.

3.5 Основні функції та алгоритми

Програмне забезпечення системи побудоване на основі набору функцій та алгоритмів, які забезпечують гнучке керування світловими ефектами та оперативне реагування на команди користувача.

Основним механізмом взаємодії з користувачем є обробка сигналів із інфрачервоного пульта, реалізована у циклічній функції loop(). Програма регулярно перевіряє наявність нових команд і фільтрує повторювані чи некоректні сигнали, що дозволяє уникнути небажаних переходів і забезпечує точну реакцію системи лише на унікальні натискання кнопок. Кожен валідний код команди зіставляється з конкретною дією – змінюється

яскравість, швидкість анімації, колір або активний режим підсвічування.

Важливою частиною реалізації є скінченний автомат, що базується на переліку режимів роботи системи. Зміна стану автомата відбувається виключно за командами користувача, що забезпечує передбачувану і керовану поведінку пристрою. Кожен режим відображається відповідною функцією, яка генерує унікальний світловий ефект із врахуванням актуальних параметрів, таких як яскравість і швидкість.

Програма підтримує плавність переходів між ефектами, що досягається завдяки регулярному оновленню стану світлодіодів у циклі і використанню оптимізованих математичних функцій. Це гарантує не лише привабливу візуалізацію, а й високу продуктивність навіть на мікроконтролері з обмеженими ресурсами.

Обробка команд і виконання анімацій відбуваються незалежно, що забезпечує одночасну інтерактивність і безперервність роботи системи. Такий підхід робить користувацький досвід комфортним і дозволяє швидко реагувати на зміни без затримок.

3.5.1 Механізм перемикання режимів підсвічування

Центральним елементом керування світловими ефектами є перелік enum Mode та глобальна змінна currentMode цього типу. Перелік Mode визначає всі можливі режими роботи підсвічування (лістинг 3.2)

Лістинг 3.1 – Оголошення Mode та ініціалізація currentMode

```
enum Mode {
  RAINBOW, GRADIENT, STATIC_COLOR, RUNNING_LIGHTS, BLINKING,
  COLOR_WAVES,   COLOR_WIPE,   BREATHING,   RANDOM_FLICKER,
AMBIENT OFF
};
Mode currentMode = OFF;
```

Після того як користувач натискає кнопку на пульті, що відповідає за вибір режиму, у блоці обробки команд змінній currentMode присвоюється відповідне значення. Далі у конструкції switch (currentMode) викликається функція що реалізує логіку режиму, який обрав користувач. Такий підхід є реалізацією скінченного автомата, де кожен case відповідає одному стану, а переходи між станами ініціюються зовнішніми подіями (командами з ІЧ-пульта). Це робить код структурованим, легко читаним та розширюваним: для додавання нового ефекту достатньо додати нове значення в enum Mode, реалізувати відповідну функцію ефекту та додати новий case у конструкцію switch.

3.5.2 Реалізація світлових ефектів

Система підтримує декілька різноманітних світлових ефектів, кожен з яких реалізований у вигляді окремої функції. Кожен ефект має свій унікальний стиль та візуальне оформлення, що дозволяє користувачу обирати найбільш привабливий для конкретної ситуації режим роботи. У таблиці 3.2 наведено огляд цих ефектів з коротким описом їх функціональності.

Таблиця 3.2 – Огляд реалізованих світлових ефектів

Назва ефекту	Значення Mode	Короткий опис функціональності
Веселка	RAINBOW	Плавна зміна всіх кольорів веселки вздовж стрічки.
Градієнт	GRADIENT	Статичний градієнт між двома відтінками: від червоного до помаранчево-жовтого.
Статичний колір	STATIC_COLOR	Стрічка світиться одним обраним користувачем кольором.
Вогні, що біжать	RUNNING_LIGHTS	Групи світлодіодів обраного кольору послідовно пробігають вздовж стрічки.

Мерехтіння	BLINKING	Вся стрічка періодично вмикається обраним кольором та вимикається.
Кольорові хвилі	COLOR_WAVES	Плавні хвилі яскравості та кольору, що рухаються вздовж стрічки.
Заповнення кольором	COLOR_WIPE	Світлодіоди на стрічці послідовно запалюються обраним кольором, а потім так само послідовно гаснуть.
Дихання	BREATHING	Плавна зміна яскравості всієї стрічки, імітуючи дихання.
Випадкове мерехтіння	RANDOM_FLICKER	Короткі спалахи випадкових світлодіодів, що створюють ефект зламаної вивіски чи мерехтіння зірок.
Адаптивна підсвітка	AMBIENT	Динамічне підсвічування, яке аналізує кольори на екрані монітора в реальному часі та відтворює відповідні відтінки на світлодіодній стрічці.

Функція rainbowEffect() створює класичний ефект веселки, що рухається вздовж стрічки. Для цього застосовується статична змінна static uint8_t hue = 0;, яка на кожному виклику збільшується на одиницю, що призводить до плавного зсуву кольорової палітри. За допомогою функції fill_rainbow(leds, NUM_LEDS, hue, 7) відтінки веселки заповнюють весь масив світлодіодів із кроком 7 між сусідніми пікселями. Після заповнення виконується виклик FastLED.show(), що оновлює стан стрічки, забезпечуючи видиму анімацію руху кольорів.

Інший ефект – gradientEffect() – створює статичний градієнт між двома заданими відтінками. Для цього задаються початковий startHue і кінцевий endHue (наприклад, 0 і 32), між якими лінійно інтерполюється відтінок для кожного світлодіода в стрічці за формулою uint8_t hue = startHue + ((hueRange * i) / (NUM_LEDS - 1));. Кожен світлодіод отримує колір у моделі

HSV з максимальною насиченістю і яскравістю, а виклик FastLED.show() виводить оновлену палітру на стрічку.

У функції staticColorEffect() застосовується просте заповнення всієї стрічки одним кольором, який зберігається у глобальній змінній currentColor. Для цього використовується fill_solid(leds, NUM_LEDS, currentColor), що заповнює масив однаковим кольором, і виконується оновлення стрічки.

Ефект runningLightsEffect() створює рух світлових сегментів стрічки. Тут використовується статична змінна static int position = 0;, яка визначає зсув патерну. В циклі по всіх світлодіодах перевіряється умова ((i + position) % 4 == 0), і якщо вона істинна, світлодіод набуває кольору currentColor, інакше встановлюється чорний колір CRGB::Black. Значення position збільшується і скидається за модулем довжини стрічки, що забезпечує безперервний рух світлових "вогників".

Для blinkingLightsEffect() використовується булева змінна static bool state = false;, яка змінює свій стан на протилежний кожного виклику функції. Коли state рівна true, вся стрічка світиться кольором currentColor, інакше всі світлодіоди вимкнені.

Функція colorWavesEffect() генерує хвильові зміни кольору та яскравості. Вона оперує двома статичними змінними – waveOffset і hueBase, які відповідно відповідають за зсув фази хвилі та базовий колір. Для кожного світлодіода обчислюється яскравість через sin8(), а колір задається як CHSV(hueBase + (i * 5), 255, brightness), що дає ефект плавної хвилі, що рухається по стрічці з постійним оновленням цих змінних.

colorWipeEffect() реалізує послідовне заповнення і очищення стрічки кольором. Для цього використовуються змінні currentIndex та isFilling, які контролюють поточний індекс світлодіода та напрямок дії (заповнення або стирання). На кожному кроці функції світлодіод за індексом або підсвічується кольором, або гасне, після чого індекс збільшується, а при досягненні кінця стрічки напрямок змінюється.

У breathingEffect() плавно змінюється яскравість усієї стрічки, що

імітує ефект дихання. Змінна wavePosition використовується для обчислення синусоїдальної яскравості через sin16(), а колір встановлюється у відтінках сірого (CHSV(0, 0, brightness)). Плавність переходів забезпечує відтворення природного ефекту.

Функція randomFlickerEffect() створює випадкові короткі спалахи на окремих світлодіодах. Зберігається масив тривалості спалаху для кожного світлодіода, а також лічильник, що періодично ініціює нові випадкові спалахи. Світлодіоди, які не мерехтять, світяться базовим тьмяним кольором, що підтримує контраст і динаміку ефекту.

Функція ambientEffect() відповідає за прийом і відображення адаптивної підсвітки, яка динамічно відтворює кольори, що надходять від зовнішньої програми через послідовний порт. Вона очікує спеціальний префікс пакету даних ('A', 'd', 'a'), що сигналізує початок прийому, після чого послідовно зчитує службові байти, включно з контрольними сумами, для перевірки коректності передачі.

Основна логіка полягає у послідовному зчитуванні RGB-значень для кожного світлодіода в стрічці. Після отримання значень кожен світлодіод оновлюється з відповідним кольором у масиві leds[]. Оновлення фізичної стрічки здійснюється викликом FastLED.show(), що гарантує синхронне і плавне відображення отриманих кольорів.

Завдяки застосуванню блокування циклу до отримання повного пакету даних, функція забезпечує безперервний і коректний прийом інформації без пропусків, що є критично важливим для підтримки якості адаптивної підсвітки.

Таким чином, ambientEffect() є основою інтеграції зовнішнього аналізу екрану з апаратним керуванням світлодіодною стрічкою, забезпечуючи ефект живої і реактивної підсвітки.

Режим OFF заповнює стрічку чорним кольором, вимикаючи всі світлодіоди.

У функціях, що відповідають за анімовані ефекти (rainbowEffect,

runningLightsEffect, colorWavesEffect тощо), використовуються локальні статичні змінні з ключовим словом static. Це потрібно, бо функція loop() викликає ці ефекти кожної ітерації, інакше змінні, що зберігають стан анімації, наприклад, поточний відтінок чи фаза хвилі, ініціалізувалися б заново, унеможливлюючи плавність ефекту. Статичні змінні створюються лише один раз і зберігають значення між викликами, що забезпечує безперервність роботи. Різноманітні світлові ефекти реалізуються не лише через стандартні функції, а й різні алгоритмічні підходи – від інкрементування лічильників до тригонометричних функцій для плавних осциляцій та стохастичних ефектів, що керують кожним світлодіодом.

3.6 Інтеграція апаратної та програмної частин

Інтеграція апаратної та програмної складових є ключовою для забезпечення синхронної роботи системи адаптивної підсвітки. Взаємодія між Python-скриптом на персональному комп’ютері та мікроконтролером Arduino реалізована через послідовний інтерфейс з чітко визначеним протоколом передачі даних, що починається зі спеціального префікса Ada.

Python-скрипт виконує функцію аналізу екранного зображення, формує колірні дані, які потім передаються на Arduino. Зі свого боку, мікроконтролер очікує повні пакети інформації, перевіряє їх цілісність за допомогою контрольної суми і відповідно оновлює кольори світлодіодів.

Використання блокуючої логіки прийому даних гарантує коректність і синхронність роботи, уникаючи пропусків чи артефактів підсвітки. Швидкість передачі у 250000 біт/с та частота оновлення близько 48 кадрів на секунду забезпечують плавність та якість відтворення кольорів у режимі AMBIENT.

Такий розподіл функцій між зовнішнім ПЗ і мікроконтролером, а також надійний протокол обміну є запорукою стабільної та ефективної роботи системи.

4 ЗБІРКА ТА ТЕСТУВАННЯ СИСТЕМИ

Тестування системи проводилося з метою перевірки її працездатності, коректності реалізації світлових ефектів, реакції на команди користувача з інфрачервоного пульта, а також стабільності роботи приладу в цілому.

Пристрій було зібрано відповідно до розробленої у процесі планування електричної схеми. Живлення стрічки здійснювалося через AC/DC блок живлення на 12 В/3 А, Arduino живився через USB-порт. На рисунку 4.1 зображено зібраний пристрій керування світлодіодною стрічкою для зображення світлових ефектів. Після підключення пристрою виконувалося завантаження програмного коду та перевірка основних функцій.

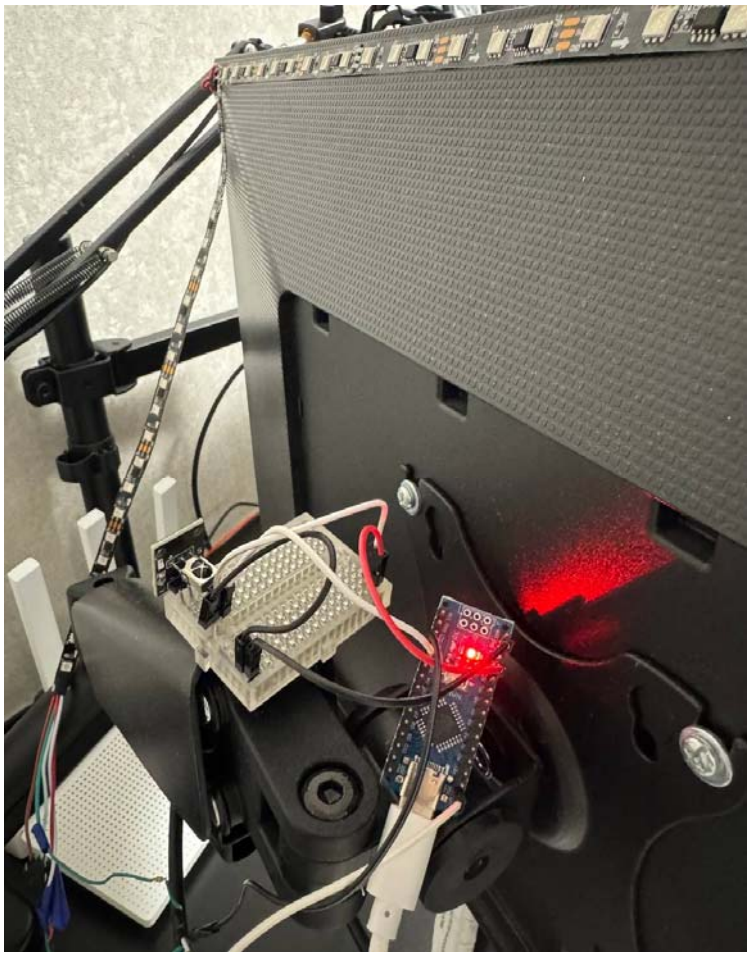

Рисунок 4.1 – Схема підключення системи динамічного підсвічування

Під час тестування системи перевірялися основні функціональні режими підсвічування, керування яскравістю, швидкістю ефектів та кольорами за допомогою інфрачервоного пульта. Система коректно реагувала на команди користувача, плавно перемикала режими та забезпечувала безперервну роботу світлових ефектів, як це було описано у програмній частині. Тестування підтвердило стабільну роботу протягом тривалого часу та відсутність перегріву або інших несправностей. Фотографії процесу тестування режимів підсвітки наведені на рисунках 4.2–4.3.

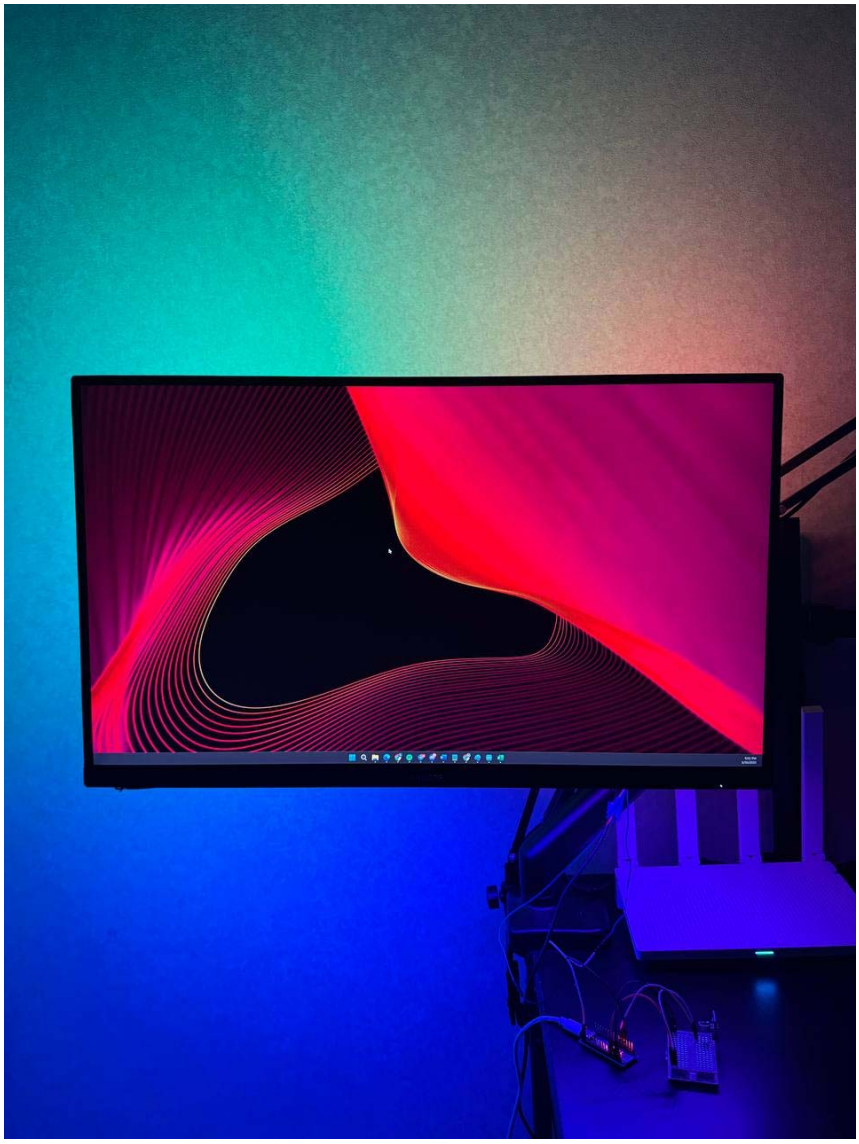

Рисунок 4.2 – Процес тестування режиму Rainbow

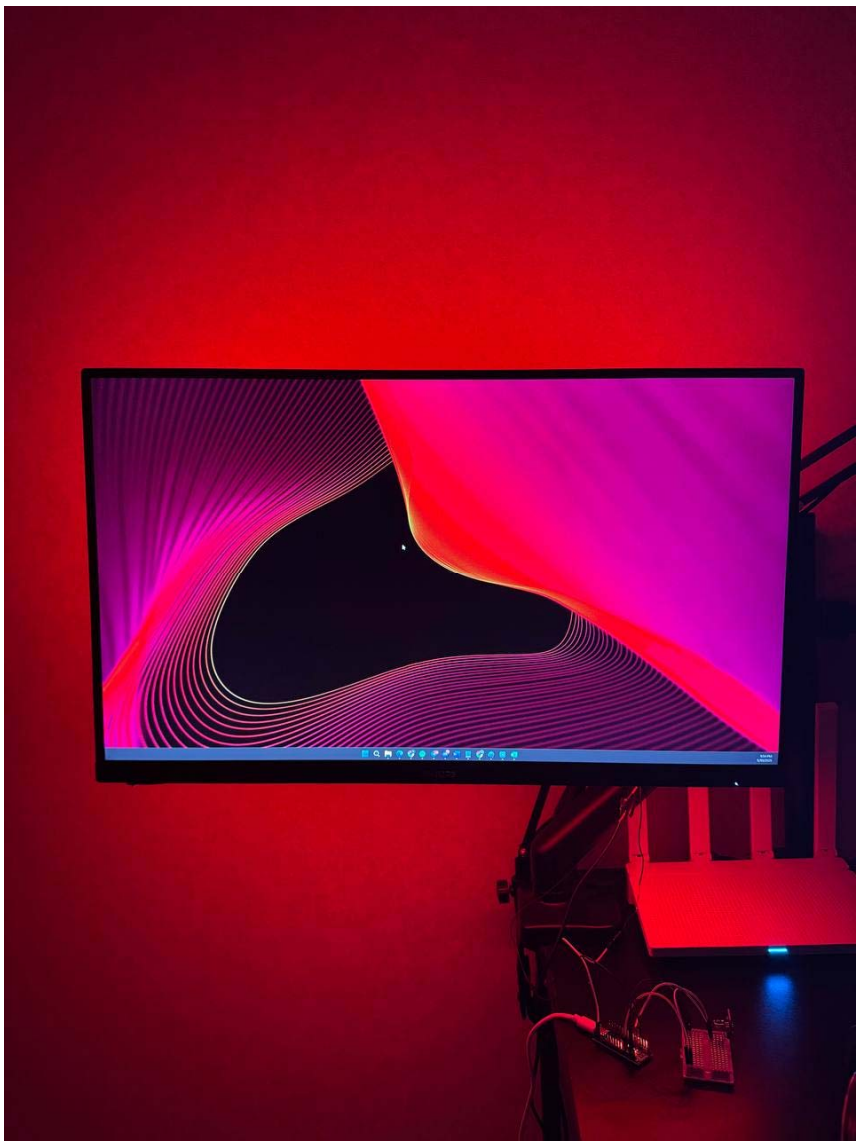

Рисунок 4.2 – Тестування режиму Static Color із налаштованим кольором

Аналогічно було протестовано всі інші режими роботи системи. Таким чином, проведене тестування підтвердило коректність роботи всіх режимів підсвічування, стабільність функціонування системи в цілому та відповідність заданим технічним вимогам. Результати випробувань свідчать про готовність пристрою до подальшого впровадження та експлуатації в реальних умовах. Всі світлові ефекти працюють згідно з очікуваннями, забезпечуючи плавні анімації та швидку реакцію на команди користувача.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи розроблено та реалізовано мікроконтролерну систему динамічного підсвічування моніторів на базі Arduino та адресної RGB світлодіодної стрічки WS2811. Створено функціональний прототип, що після всебічного тестування підтвердив свою працездатність, відповідність висунутим вимогам та стабільність роботи.

Процес розробки включав аналіз існуючих рішень, на основі якого було здійснено вибір апаратних компонентів: платформу Arduino через доступність та простоту, світлодіодну стрічку WS2811 завдяки її 12-вольтовому живленню для стабільності та ефективному використанню ресурсів мікроконтролера, а також інфрачервону систему для зручного дистанційного керування. Було спроектовано електричну функціональну схему та реалізовано програмне забезпечення мовою C++ в середовищі Arduino IDE з використанням спеціалізованих бібліотек FastLED для створення різноманітних світлових ефектів та IRremote для обробки команд з ІЧ-пульта. Система дозволяє користувачеві обирати світлові режими, керувати яскравістю, швидкістю анімацій та кольором.

Ключовою практичною значущістю розробленої системи є підвищення комфорту користувача під час тривалої роботи за комп’ютером, зокрема зниження зорової втоми та ризику розвитку комп’ютерного зорового синдрому за рахунок зменшення різкого контрасту між яскравістю екрана та темним оточенням. Система також покращує візуальне сприйняття контрастності зображення на екрані та надає можливості для естетичної персоналізації робочого простору. Робота демонструє успішне застосування мікроконтролерних технологій для створення доступних DIY-рішень у сфері ергономіки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. PubMed Central. The effect of ambient lighting on visual perception and performance. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC9434525/ (дата звернення: 20.05.2025).

2. Ophthalmology 24. How lighting affects your vision. URL: https://www.ophthalmology24.com/how-lighting-affects-your-vision (дата звернення: 20.05.2025).

3. BenQ. Behind Monitor Lighting: What You Need to Know. URL: https://www.benq.com/en-us/knowledge-center/knowledge/behind-monitor-lighting.html (дата звернення: 20.05.2025).

4. Apollo Technical. Optimizing Workspace Lighting for Productivity and Wellness. URL: https://www.apollotechnical.com/optimizing-workspace-lighting-for-productivity-and-wellness/ (дата звернення: 20.05.2025).

5. VONN Lighting. The Science of Color Temperature: How It Affects Mood and Ambiance. URL: https://www.vonn.com/blogs/articles/the-science-of-color-temperature-how-it-affects-mood-and-ambiance (дата звернення: 20.05.2025).

6. AILAS. Нічний режим у телефоні: чи врятує він ваші очі. URL: https://www.ailas.com.ua/novosti/publikacii_specialistov/savinec_tatyana_vladimirovna_publikacii/nichnij-rezhim-u-telefoni-chi-vrjatuje-vin-vashi-ochi.html (дата звернення: 20.05.2025).

7. Hyperion Project. Introduction to Hyperion. URL: https://docs.hyperion-project.org/user/Introduction.html (дата звернення: 20.05.2025).

8. BenQ. The Best Lamp for Your Monitor to Relieve Eye Strain. URL: https://www.benq.eu/uk-ua/knowledge-center/knowledge/the-best-lamp-for-your-monitor-to-relieve-eye-strain.html (дата звернення: 20.05.2025).

9. Arduino. Arduino UNO R3 – Documentation. URL:

https://docs.arduino.cc/hardware/uno-rev3/ (дата звернення: 20.05.2025).

10. Arduino. Arduino Nano – Documentation. URL: https://docs.arduino.cc/hardware/nano/ (дата звернення: 20.05.2025).

11. Arduino. ESP32 Guide – Arduino Cloud. URL: https://docs.arduino.cc/arduino-cloud/guides/esp32/ (дата звернення: 20.05.2025).

12. Svetcomplect. Світлодіодна стрічка: переваги. URL: https://svetcomplect.ua/statti/svitlodiodna-strichka-perevagy (дата звернення: 20.05.2025).

13. Lisle Apex. WS2811 vs WS2812B: What Are the Differences? URL: https://www.lisleapex.com/blog-ws2811-vs-ws2812b-what-are-differences (дата звернення: 20.05.2025).

14. SDIP Light. What is WS2811 and How to Use? URL: https://www.sdiplight.com/what-is-ws2811-and-how-to-use/ (дата звернення: 20.05.2025).

15. Strip Sled Light. Differences Between APA102, SK9822, HD107S, WS2812B, SK6812, WS2811, WS2815, WS2813. URL: https://www.stripsledlight.com/what-different-of-apa102sk9822hd107sws2812b-sk6812ws2811ws2815ws2813/ (дата звернення: 20.05.2025).

16. Arduino. IRremote Library Documentation. URL: https://docs.arduino.cc/libraries/irremote (дата звернення: 20.05.2025).

17. Arduino C++ Reference. URL: https://docs.arduino.cc/arduino-cloud/guides/arduino-c/ (дата звернення: 22.05.2025).

18. Arduino Software (IDE). URL: https://www.arduino.cc/en/software/ (дата звернення: 22.05.2025).

19. Arduino IDE v1 Basics Tutorial. URL: https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics/ (дата звернення: 23.05.2025).