

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ФАХОВИЙ КОЛЕДЖ
ДЕРЖАВНОГО НЕКОМЕРЦІЙНОГО ПІДПРИЄМСТВА
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»
Циклова комісія комп'ютерних систем та мереж
(повна назва циклової комісії)

Допустити до захисту
Голова випускової циклової комісії
комп'ютерних систем та мереж

(повна назва циклової комісії)


(підпис)

Ірина КРАВЧУК
(ім'я, ПРІЗВИЩЕ)

« 10 » 06 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬО-ПРОФЕСІЙНОГО СТУПЕНЯ
ФАХОВИЙ МОЛОДШИЙ БАКАЛАВР

Тема: «Дерево RGB на базі мікроконтролера Arduino»

Група: 3-011

Спеціальність: 123 «Комп'ютерна інженерія»

Здобувач освіти


(підпис)

Вікторія ТУРЧИН

(ім'я, ПРІЗВИЩЕ)

Керівник роботи


(підпис)

Оксана ОСАДЧА

(ім'я, ПРІЗВИЩЕ)

Консультант з оформлення
пояснювальної записки


(підпис)

Оксана ОСАДЧА

(ім'я, ПРІЗВИЩЕ)

Кривий Ріг 2025 р.

КРИВОРІЗЬКИЙ ФАХОВИЙ КОЛЕДЖ
ДЕРЖАВНОГО НЕКОМЕРЦІЙНОГО ПІДПРИЄМСТВА
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

Відділення комп'ютерної та програмної інженерії
Циклова комісія комп'ютерних систем та мереж
Освітньо-професійний ступінь фаховий молодший бакалавр
Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Голова випускової циклової комісії
комп'ютерних систем та мереж

(повна назва циклової комісії)

Ірина КРАВЧУК

(ім'я, ПРІЗВИЩЕ)


(підпис)

« 10 » 05 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ОСВІТИ

ТУРЧИН Вікторії Михайлівні

(прізвище, ім'я, по батькові)

1. Тема роботи Дерево RGB на базі мікроконтролера Arduino

Керівник роботи ОСАДЧА Оксана Георгіївна, викладач вищої категорії

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по коледжу від « 04 » 04 2025 року № 50-ст

2. Строк подання здобувачем освіти роботи з _____ по _____

3. Вихідні дані до роботи Інтерактивне світлодіодне "Дерево RGB" з керуванням

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити
аналіз принципу роботи та методів керування RGB-світлодіодами;
вибір та обґрунтування елементної бази; інженерна реалізація дерева RGB;
тестування та налагодження розробленої системи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Презентація Microsoft PowerPoint

6. Консультанти розділів роботи (проекту)

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

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|-------|----------------------------------------------------------------------------------------|-------------------------------|-----------------|
| | <i>Узгодження технічного завдання з керівником кваліфікаційної роботи</i> | <i>04.04.2025-07.04.2025</i> | <i>виконано</i> |
| | <i>Підбір та вивчення науково-технічної літератури за темою кваліфікаційної роботи</i> | <i>08.04.2025-14.04.2025</i> | <i>виконано</i> |
| | <i>Принцип роботи та методи керування RGB-світлодіодами</i> | <i>15.04.2025-21.04.2025</i> | <i>виконано</i> |
| | <i>Вибір та обґрунтування елементної бази</i> | <i>22.04.2025-28.04.2025</i> | <i>виконано</i> |
| | <i>Інженерна реалізація дерева RGB</i> | <i>29.04.2025-02.05.2025</i> | <i>виконано</i> |
| | <i>Написання та оформлення пояснювальної записки</i> | <i>12.05.2025-23.05.2025</i> | <i>виконано</i> |
| | <i>Перевірка на плагіат пояснювальної записки</i> | <i>26.05.2025-30.05.2025</i> | <i>виконано</i> |
| | <i>Попередній захист кваліфікаційної роботи</i> | <i>02.06.2025-06.06.2025</i> | <i>виконано</i> |
| | <i>Захист кваліфікаційної роботи</i> | | |

Здобувач освіти


(підпис)

Вікторія ТУРЧИН

(ім'я, ПРИЗВИЩЕ)

Керівник роботи


(підпис)

Оксана ОСАДЧА

(ім'я, ПРИЗВИЩЕ)

метадані

Звіт подібності

Назва організації
Ukrainian national aviation university
 Заголовок
Кваліфікаційна_робота_RGB_дерево
 Автор Науковий керівник / Експерт
ТурчинОсадча О.
 підрозділ
Криворізький Фаховий коледж

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

3.05%

3.05%

КП 1

0.39%

0.39%

КЦ

25

Довжина фрази для коефіцієнта подібності 2

8190

Кількість слів

62159

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про **МОЖЛИВІ** маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

| | | |
|------------------------|--|----|
| Заміна букв | | 5 |
| Інтервали | | 0 |
| Мікропробіли | | 0 |
| Білі знаки | | 0 |
| Парафрази (SmartMarks) | | 20 |

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

| ПОРЯДКОВИЙ НОМЕР | НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ) | Копію тексту |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| | | КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
| 1 | https://ekmair.ukma.edu.ua/bitstream/handle/123456789/22155/Zhukovskiy_Bakalavrska_robota.pdf?sequence=1 | 36 0.44 % |
| 2 | https://cases.media/article/rgb-kolirna-model | 35 0.43 % |
| 3 | https://cases.media/article/rgb-kolirna-model | 34 0.42 % |
| 4 | https://ekmair.ukma.edu.ua/bitstream/handle/123456789/22155/Zhukovskiy_Bakalavrska_robota.pdf?sequence=1 | 29 0.35 % |

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Дерево *RGB* на базі мікроконтролера *Arduino*» містить: 56 сторінок основного тексту, 27 рисунків, 3 таблиць, 16 використаних літературних джерел, 1 додаток.

ARDUINO, МІКРОКОНТРОЛЕР, *RGB*-СВІТЛОДІОДИ, *WS2812B*, *TPIC6B595N*, СВІТЛОДІОДНЕ ДЕРЕВО, ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ, *IoT*, ДЕКОРАТИВНЕ ОСВІТЛЕННЯ

Мета кваліфікаційної роботи – розробка та створення функціонального прототипу інтерактивного світлодіодного дерева *RGB* з програмним керуванням. У ході виконання кваліфікаційної роботи була успішно розроблена та реалізована апаратна частина світлодіодного дерева *RGB*. Ця система базується на мікроконтролері *Arduino Uno* та адресних світлодіодах *WS2812B*, що забезпечило її модульність та гнучку масштабованість. Паралельно з апаратною частиною було створено програмне забезпечення, яке надає можливість точного керування кожним індивідуальним світлодіодом. Це програмне рішення дозволяє генерувати широкий спектр динамічних кольорних ефектів, включаючи статичні кольори, плавні градієнти, складні анімації, а також забезпечує реакцію на різні зовнішні чинники. На завершальному етапі було проведено ретельне тестування та налагодження всієї розробленої системи, що підтвердило її стабільну роботу та повну відповідність заздалегідь визначеним функціональним вимогам.

ЗМІСТ

| | |
|----------------------------------------------------------------------------|-----------|
| <i>ВСТУП</i> | |
| <i>7 РОЗДІЛ 1 ПРИНЦИП РОБОТИ ТА МЕТОДИ КЕРУВАННЯ</i> | |
| <i>RGB-СВІТЛОДІОДАМИ</i> | |
| | <i>10</i> |
| 1.1 Конструктивні та функціональні особливості <i>RGB</i> -освітлення..... | |
| 10 | 1.1.1 |
| Принцип роботи <i>RGB</i> -світлодіодів..... | 11 |
| 1.1.2 Основні характеристики <i>RGB</i> -світлодіодів | 11 |
| 1.1.3 Особливості колірної моделі <i>RGB</i> для керування світлодіодними | |

| | |
|----------------------------------------------------------------------------------------------------------------|----|
| пристроями | 12 |
| 1.2 Методи керування яскравістю світлодіодів..... | |
| 18 1.2.1 Аналогове керування яскравістю | |
| 18 1.2.2 Широтно-імпульсна модуляція (ШІМ/PWM) | 21 |
| 1.3 Методи створення динамічних світлових ефектів за допомогою керування яскравістю дерева <i>RGB</i> | 25 |
| 1.3.1 Плавне змінення кольорів (фейдинг)..... | 25 |
| 1.3.2 Створення імпульсів та стробоскопічних ефектів..... | 26 |
| 1.3.3 Реалізація складних динамічних патернів освітлення | 27 |
| РОЗДІЛ 2 ВИБІР ТА ОБҐРУНТУВАННЯ ЕЛЕМЕНТНОЇ БАЗИ | |
| 30 2.1 Обґрунтування вибору компонентів..... | |
| 30 2.1.1 Вибір <i>RGB</i> -світлодіодів | |
| 31 2.1.2 Вибір мікроконтролера..... | 32 |
| 2.1.3 Вибір зсувних регістрів | 34 |
| 2.1.4 Вибір резисторів..... | 36 |
| 2.1.5 Вибір джерела живлення..... | 37 |
| 2.1.6 Вибір проводів та інших допоміжних компонентів | 37 |
| 2.1.7 Матеріали для виготовлення корпусу дерева..... | 38 |
| РОЗДІЛ 3 ІНЖЕНЕРНА РЕАЛІЗАЦІЯ ДЕРЕВА <i>RGB</i> | |
| 40 3.1 Опис з'єднань та апаратної реалізації..... | |
| 40 3.1.1 Підключення зсувних регістрів..... | 40 |
| 3.1.2 Підключення резисторів..... | |
| 40 3.1.3 Підключення <i>RGB</i> -світлодіодів із зсувними регістрами 41 | |
| 3.1.4 З'єднання з <i>Arduino Nano</i> 43 | |
| 3.2 Програмна реалізація..... | |
| 46 3.2.1 Вибір середовища розробки..... | |
| 46 3.2.2 Алгоритм роботи програми..... | |
| 48 3.3.3 Розробка програмного коду..... | |
| 48 3.3 Завершення та кінець розробки дерева <i>RGB</i> | 51 |

ВСТУП

Стрімкий розвиток технологій у XXI столітті фундаментально змінює підходи до освітлення, перетворюючи його з суто функціонального елемента на потужний інструмент дизайну, взаємодії та створення атмосфер. Зокрема, інтелектуальні системи на основі світлодіодів, і особливо *RGB*-освітлення, набувають дедалі більшої популярності. Їхня здатність відтворювати практично необмежений спектр кольорів та створювати динамічні, програмовані ефекти робить їх незамінними не лише у декоративному освітленні інтер'єрів та екстер'єрів, а й у архітектурному підсвічуванні, сценічному оформленні, рекламних конструкціях та різноманітних побутових пристроях.

Незважаючи на значну кількість готових комерційних рішень, ринок постійно потребує більш гнучких, економічно ефективних та персоналізованих систем керування освітленням. Такі системи дозволяють реалізовувати унікальні світлові сценарії, які неможливо відтворити за допомогою стандартних пристроїв. У цьому контексті розробка власних програмованих світлодіодних систем, керованих мікроконтролерами, стає не просто технічним завданням, а актуальним науково практичним викликом. Вона дає можливість не лише глибоко зануритися у принципи роботи апаратних та програмних складових таких систем, але й розробити рішення, що точно адаптовані до конкретних вимог користувача чи проєкту. Це особливо важливо в умовах зростаючої автоматизації та персоналізації технологій, де кастомізація та інтерактивність є ключовими перевагами.

Ця кваліфікаційна робота гармонійно вписується в актуальні науково технічні програми, що стосуються розвитку мікроконтролерних систем, вбудованих рішень та концепцій Інтернету речей (*IoT*). Вона відображає сучасні тенденції у створенні розумних пристроїв, систем автоматизації та інтерактивних об'єктів. Проєкт інтегрує комплексні знання та навички з декількох ключових інженерних дисциплін: електроніки (розуміння електричних схем та компонентів), програмування (написання коду для мікроконтролерів) та проєктування (розробка

системного рішення від ідеї до реалізації). Таке поєднання є фундаментом для інноваційних розробок у сучасному світі інженерії.

Метою кваліфікаційної роботи є розробка та практична реалізація інтерактивної *RGB*-гірлянди, що має форму декоративного дерева. Керування гірляндою здійснюватиметься за допомогою мікроконтролера *Arduino Nano*, з ключовою особливістю — можливістю вибору різноманітних світлових режимів через тактові кнопки, що забезпечить її інтерактивність та зручність використання.

Для досягнення цієї мети було визначено наступні завдання:

- Ґрунтовно дослідити теоретичні засади функціонування *RGB* світлодіодів, їхні ключові електричні та світлові характеристики, а також основні методи керування їхньою яскравістю та кольором, приділивши особливу увагу широтно-імпульсній модуляції (ШІМ) як ефективному інструменту для створення динамічних ефектів.
- Здійснити комплексний аналіз та обґрунтувати вибір елементної бази проекту, включаючи оптимальну модель мікроконтролера, тип *RGB*-світлодіодів, використання зсувних регістрів для розширення кількості керованих виходів, а також підбір необхідних резисторів, джерела живлення та інших допоміжних електронних компонентів.
- Розробити детальну принципову електричну схему підключення всіх електронних компонентів системи та на її основі здійснити фізичну, апаратну реалізацію *RGB*-гірлянди, забезпечивши правильне з'єднання та монтаж.
- Створити та імплементувати програмний код для мікроконтролера *Arduino Nano*, який забезпечить виконання різноманітних світлових режимів (таких як біжучі вогні, плавні переливи, статичні заливки тощо) та дозволить користувачеві інтерактивно керувати цими режимами за допомогою тактових кнопок.
- Провести повне тестування розробленої апаратно-програмної системи,

включаючи перевірку всіх функцій, режимів та стабільності роботи, а також виконати необхідні оптимізації коду для забезпечення максимальної ефективності та надійності.

9

Об'єктом дослідження в даній роботі є система керування світлодіодним освітленням на базі мікроконтролера, яка є автономною та інтерактивною. Предметом дослідження є принципи, методи та засоби реалізації складних динамічних світлових ефектів за допомогою програмованих *RGB*-світлодіодів та зсувних регістрів, керованих мікроконтролером.

У процесі виконання кваліфікаційної роботи був застосований комплексний підхід, що включав: методи системного аналізу для глибокого вивчення архітектури системи та взаємодії між її окремими компонентами; методи схемотехнічного проектування для розробки надійної та функціональної апаратної частини; а також методи структурного програмування та відладки для створення ефективного, модульного та надійного програмного коду. В результаті цієї роботи було успішно створено повністю функціональний прототип інтерактивної *RGB*-гірлянди, що яскраво демонструє широкий спектр програмованих світлових ефектів, які керуються через зручний інтерфейс з кнопками. Це переконливо підтверджує досягнення всіх поставлених мети та завдань кваліфікаційної роботи.

10

РОЗДІЛ 1

ПРИНЦИП РОБОТИ ТА МЕТОДИ КЕРУВАННЯ *RGB*-СВІТЛОДІОДАМИ

У цьому розділі розглядаються фундаментальні аспекти, що стосуються функціонування та керування *RGB*-світлодіодами. Спочатку буде детально описано конструктивні та функціональні особливості *RGB*-освітлення, включаючи принцип роботи самих світлодіодів та їх основні електричні характеристики. Особливу увагу буде приділено колірним моделям *RGB*, *HSV* (*HSL*), які є ключовими для розуміння того, як з трьох базових кольорів формується широкий

спектр відтінків.

Надалі, розділ зосередиться на методах керування яскравістю світлодіодів. Будуть проаналізовані два основні підходи: аналогове керування яскравістю та широтно-імпульсна модуляція (ШІМ/*PWM*). Детально буде розкрито переваги та недоліки кожного методу, а також обґрунтовано вибір ШІМ як основного методу керування яскравістю в даній кваліфікаційній роботі, що є критичним для ефективної реалізації *RGB*-дерева.

Завершальна частина розділу буде присвячена методам створення динамічних світлових ефектів за допомогою керування яскравістю *RGB*-дерева. Будуть описані такі ефекти, як плавне змінення кольорів (фейдинг), створення імпульсів та стробоскопічних ефектів, а також принципи реалізації складних динамічних патернів освітлення, що є основою для демонстрації функціональності розробленого пристрою.

1.1 Конструктивні та функціональні особливості *RGB*-освітлення

RGB-освітлення базується на адитивній моделі змішування світла (червоний, зелений, синій), що дозволяє відтворювати мільйони кольорів. Конструктивно воно використовує спеціальні *RGB*-світлодіоди (зі спільним анодом або катодом) та струмообмежувальні резистори, керовані електронікою (наприклад,

11

мікроконтролерами чи зсувними регістрами). Функціонально, це дозволяє не лише гнучко змішувати кольори за допомогою ШІМ, а й створювати різноманітні динамічні ефекти та незалежно регулювати яскравість і насиченість, що робить його універсальним у широкому спектрі застосувань.

1.1.1 Принцип роботи *RGB*-світлодіодів

RGB-світлодіоди являють собою спеціальний тип напівпровідникових компонентів з *PN*-переходом, який починає світитися при проходженні електричного струму в прямому напрямку. Це дозволяє візуально визначити активність електронного пристрою. У таких світлодіодах три основні кольори -

червоний, зелений і синій - інтегруються в один корпус, що дає змогу створювати широкий спектр кольорів завдяки змішанню світлових потоків кожного з каналів. Саме така конструкція і називається *RGB*- світлодіодом.

Ці елементи знайшли широке застосування в декоративному освітленні, сценічному оформленні, світлодіодних панелях, екранах та інших електронних системах візуалізації. Конструктивно вони мають чотири виводи: три з них відповідають за кольори (*R* - червоний, *G* - зелений, *B* - синій), а четвертий - це загальний контакт, що може бути анодом або катодом, залежно від типу світлодіода.

Формування конкретного кольору досягається шляхом зміни яскравості кожного з трьох внутрішніх світлодіодів. Комбінування цих інтенсивностей дозволяє генерувати будь-який колір на виході, що й забезпечує гнучкість у відображенні світлових ефектів.

1.1.2 Основні характеристики *RGB*-світлодіодів

RGB-світлодіоди - це світлодіодні пристрої, які генерують різні кольори шляхом поєднання трьох основних відтінків: червоного, зеленого та синього. Кожен *RGB*-світлодіод містить три окремі чіпи - для кожного базового кольору - розміщені в одному корпусі. Шляхом індивідуального регулювання інтенсивності кожного з цих кольорів за допомогою широтно-імпульсної модуляції (ШІМ) можна

12

досягти широкого спектру відтінків та точно контролювати яскравість випромінюваного світла.

Основні характеристики, що визначають можливості та застосування *RGB* світлодіодів, наведено в таблиці 1.1.

Таблиця 1.1 – Характеристика *RGB*-світлодіодів

| Параметр | Значення |
|----------------------------|----------|
| Робочий струм одного діода | 20 мА |

| | |
|------------------------------------------------------|--------------------|
| Робоча напруга червоний колір (625 Нм) | від 1.8 В до 2.2 В |
| Робоча напруга зелений колір (520 Нм) | від 2.8 В до 3.6 В |
| Робоча напруга синій колір (465 Нм) | від 2.8 В до 3.8 В |
| Максимальний струм одного діода при ШІМ 10% | 20 мА |
| Інтенсивність випромінювання червоний колір (625 Нм) | 1200 мкд |
| Інтенсивність випромінювання зелений колір (520 Нм) | 2000 мкд |
| Інтенсивність випромінювання синій колір (465 Нм) | 1200 мкд |
| Кут випромінювання | 30 ° |

1.1.3 Особливості колірної моделі *RGB* для керування світлодіодними пристроями

Завдяки фундаментальним дослідженням Ісаака Ньютона було встановлено, що біле світло не є однорідним, а складається з різних кольорів, які формують його спектр. Наочно це демонструє експеримент із розкладання білого світла за допомогою скляної призми (рисунок 1.1). Це відкриття лягло в основу розуміння адитивного змішування кольорів, що використовується, зокрема, в *RGB*-освітленні, де комбінування червоного, зеленого та синього світла дозволяє отримати широкий діапазон відтінків.

13



Рисунок

1.1 – Експеримент Ісаака Ньютона з розкладання світла за допомогою призми

Для точного керування інтенсивністю базових кольорів у світлодіодах, що є ключовим для формування бажаного відтінку, застосовуються електричні принципи. Це дозволяє контролювати яскравість кожного кольорового діода та реалізовувати складні колірні комбінації, закладені в моделі *RGB*.

Продовжуючи розвиток теорії кольору, у середині *XIX* століття видатний фізик Джеймс Максвелл, на той час молодий випускник Кембриджа, активно проводив дослідження у цій галузі. Для вивчення змішування кольорів у своїх експериментах він застосовував спеціальну кольорову дзигу, сектори якої були розфарбовані в різні кольори (рисунок 1.2).



Рисунок 1.2 – Джеймс Максвелл та його кольорова дзига для експериментів зі змішування кольорів

14

Тривалий час у мистецтві та деяких наукових поглядах основними кольорами вважалися синій, червоний та жовтий. На основі їх змішування такі діячі, як Майєр, Гьоте та Іттен, створювали свої колірні системи, припускаючи, що їхня комбінація мала б утворити білий колір.

Однак, експерименти Джеймса Максвелла у середині *XIX* століття, зокрема з використанням його кольорової дзиги (рисунок 1.2), довели, що таке змішування не призводить до білого кольору. Максвелл пояснив це, спираючись на роботи англійського фізика Томаса Юнга, який висунув гіпотезу про те, що трьома основними кольорами світла є саме червоний, зелений та синій.

Причина такого вибору основних кольорів криється в особливостях будови сітківки людського ока (рисунок 1.3). У ній містяться два основні види фоторецепторів: палички, що відповідають за зір в умовах низької освітленості, та колбочки, які забезпечують сприйняття кольорів. Саме колбочки поділяються на три види:

- перші чутливі до жовто-червоної частини спектра;
- другі – до зелено-жовтої;
- треті – до синьо-фіолетової.

Таким чином, завдяки цій трихроматичній будові нашого зорового апарату, люди сприймають білий колір саме як поєднання червоного, зеленого та синього світла, що є основою адитивної колірної моделі *RGB*.

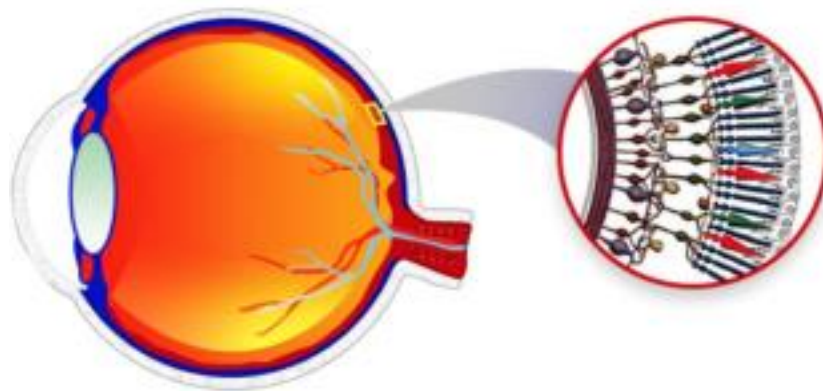


Рисунок 1.3 – Схема будови сітківки людського ока та її фоторецепторів

15

Принцип роботи *RGB*-освітлення та відображення кольору

Відображення та створення кольорів на сучасних цифрових екранах, таких як монітори, телевізори, планшети чи смартфони, ґрунтується на принципах адитивного змішування світла. Ми можемо провести простий експеримент: піднести збільшувальне скло до екрана пристрою з білим зображенням. При достатньо сильному збільшенні ми побачимо картину, що складається з безлічі окремих елементів - пікселів (рисунок 1.4). Саме в пікселях вимірюється роздільна здатність екрана (наприклад, 1920x1080).

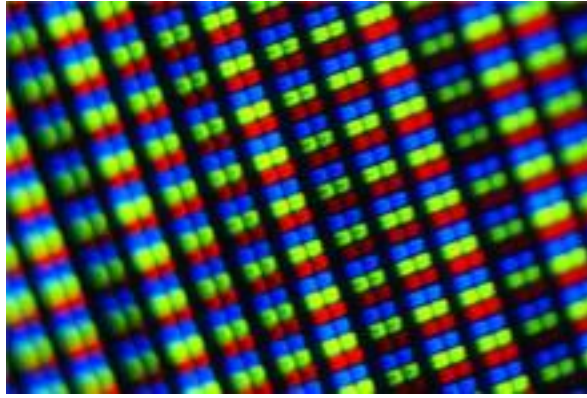


Рисунок 1.4 – Збільшене зображення пікселів цифрового дисплея з окремими субпікселями *RGB*

Те, що ми спостерігаємо на ілюстраціях, є сукупністю пікселів - термін, добре відомий у галузі дизайну. Власне, будь-яке зображення на моніторах складається саме з такої сукупності. Роздільна здатність екрана, наприклад, 1200x720 або 1920x1080, вимірюється в пікселях.

Кожен піксель містить у собі три крапки так званого люмінофора, також відомі як тріади. Ці тріади невидимі неозброєним оком, що є особливістю нашого зору - через їхній мініатюрний розмір він об'єднує ці дрібні крапки в одну цілісну одиницю. Саме завдяки змішуванню кольорів трьох субпікселів ми сприймаємо той чи інший колір пікселя.

Адитивний метод змішування кольорів RGB

У моделі *RGB* змішування кольорів відбувається за так званим адитивним методом (від англійського слова *addition* — «додавання»).

16

Завдяки дослідям Ньютона, ми знаємо, що білий колір містить у собі всі інші кольори, тоді як чорний, відповідно, є відсутністю кольорів як таких. Коли монітор вимкнено, ми бачимо чорний колір, що є стандартним станом. Якщо увімкнути на повну силу червоний субпіксель, ми додаємо червоний колір до чорного фону, і в результаті весь піксель відображається червоним. Аналогічно ми можемо отримати зелений або синій колір. Якщо ж додати до червоного зелений та синій, ми й отримаємо білий колір (рисунок 1.5).



Рисунок 1.5 – Принцип адитивного змішування кольорів *RGB*

Колірний простір *RGB*

Оскільки модель *RGB* базується на принципі додавання трьох основних кольорів світла, її можна візуалізувати як систему з трьома осями координат, кожна з яких відповідає одному з цих кольорів. Цей колірний простір є рівномірно обмеженим по всіх трьох координатах (рисунок 1.6).

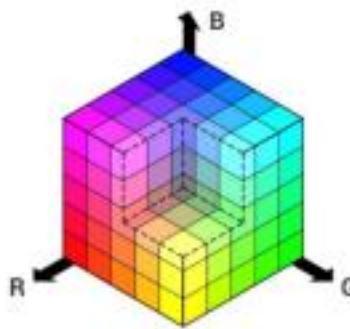


Рисунок 1.6 – Колірний простір *RGB* у вигляді куба

У комп'ютерних системах та графічних редакторах значення кожної з цих координат зазвичай задаються у діапазоні від 0 до 255 (рисунок 1.7). Таким чином, колір у цій моделі визначається як точка в межах цього куба, а його координати складаються з трьох значень (R, G, B).

17

Наприклад, у нульовій координаті $(0, 0, 0)$ розташований чорний колір, що відповідає повній відсутності світла. У максимальній координаті $(255, 255, 255)$ знаходиться білий колір, який є результатом максимальної інтенсивності всіх трьох базових кольорів. А чистий червоний колір буде представлений максимальною координатою за віссю R $(255, 0, 0)$, що означає повну інтенсивність червоного каналу та відсутність зеленого і синього.

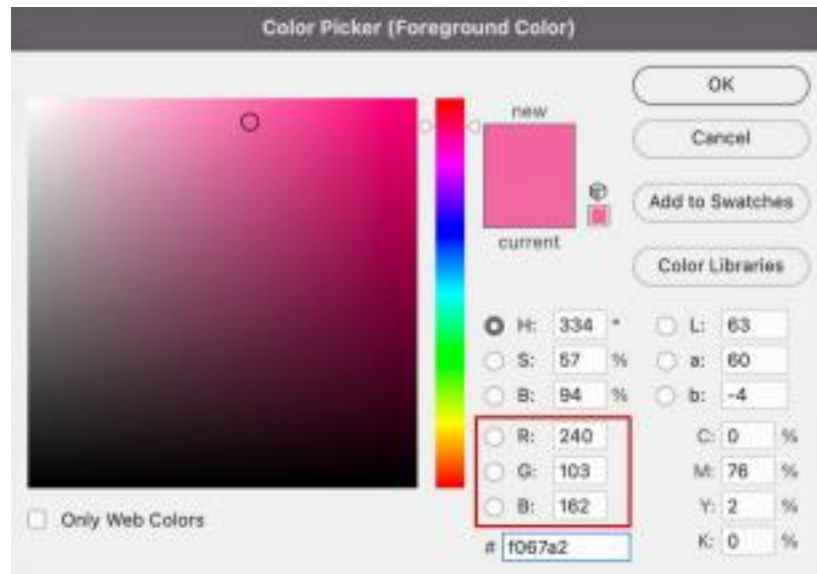


Рисунок 1.7 – Приклад представлення *RGB*-координат у колірному вибірнику (*Color Picker*)

Обмеження моделі *RGB*

Попри свою широку розповсюдженість, застосування моделі *RGB* не завжди дозволяє досягти бажаної точності відтворення кольору.

Зокрема, це зумовлено відмінностями у технологіях виробництва моніторів різних брендів. Як наслідок, навіть при ідентичних колірних координатах *RGB*, відображення одного й того ж кольору на двох різних пристроях може суттєво відрізнятись. Простіше кажучи, один і той самий фільм або розроблений макет дизайну, виглядатиме по-різному на різних моніторах.

Додаткове обмеження обумовлене безпосередньо методом змішування кольорів. Адитивний метод, на жаль, не дозволяє відтворити значну частину кольорів, присутніх у видимому спектрі та доступних для сприйняття людським оком (рисунок 1.8).



Рисунок 1.8 – Колірне охоплення пристрою в межах видимого спектра

1.2 Методи керування яскравістю світлодіодів

Керування яскравістю світлодіодів є ключовим аспектом при їх використанні, дозволяючи точно контролювати інтенсивність випромінюваного світла. Для досягнення цієї мети застосовуються різні підходи, які можна узагальнити у два основні методи: аналогове керування та широтно-імпульсна модуляція (ШИМ). Ці методи забезпечують гнучкість у налаштуванні світлового потоку відповідно до конкретних потреб застосування.

1.2.1 Аналогове керування яскравістю

Аналогове керування струмом є одним із методів регулювання яскравості світлодіодів, що передбачає активне та безперервне регулювання величини електричного струму, який протікає через світлодіод. Принцип дії ґрунтується на прямій залежності яскравості світлодіода від сили струму, що його живить: чим більший струм, тим інтенсивніше світить світлодіод.

Для реалізації аналогового керування застосовуються спеціалізовані електронні компоненти або схеми. До них належать регулятори струму, операційні підсилювачі в конфігурації джерела струму, а також спеціалізовані драйвери світлодіодів з аналоговим входом керування. Керування струмом здійснюється за

допомогою аналогового сигналу (керуючої напруги або опорного струму), який може бути сформований зовнішньою схемою або цифро-аналоговим

перетворювачем (ЦАП) мікроконтролера, що дозволяє програмно задавати бажаний рівень яскравості.

Застосування Закону Ома в аналоговому керуванні

Закон Ома є фундаментальним принципом електротехніки, який відіграє ключову роль у розумінні та практичній реалізації аналогового керування яскравістю світлодіодів. Він описує взаємозв'язок між напругою, силою струму та електричним опором у ділянці електричного кола.

Математично Закон Ома виражається формулою:

$$U=I \cdot R \text{ (1.1) де:}$$

U – напруга, В;

I – сила електричного струму, А;

R – електричний опір, Ом.

У контексті аналогового керування світлодіодами, Закон Ома є незамінним для вирішення двох основних завдань. По-перше, він дозволяє розраховувати струмообмежувальні резистори. Для захисту світлодіода від перевантаження та встановлення необхідного робочого струму ($I_{\text{робочий}}$) часто використовуються послідовні резистори. Оскільки світлодіод має певну пряму напругу падіння (U_f), а джерело живлення подає напругу $U_{\text{живлення}}$, резистор повинен "погасити" різницю напруг ($U_R = U_{\text{живлення}} - U_f$).

Опір такого резистора (R) визначається за формулою:

$$R = \frac{U_R}{I_{\text{робочий}}}$$

$$R = \frac{U_{\text{живлення}} - U_f}{I_{\text{робочий}}}$$

$$R = \frac{U_{\text{живлення}} - U_f}{I_{\text{робочий}}} \text{ (1.2)}$$

де:

R – опір струмообмежувального резистора, Ом;

$U_{\text{живлення}}$ – напруга джерела живлення, В;

U_f – пряма напруга падіння на світлодіоді, В;

$I_{\text{робочий}}$ – необхідний робочий струм світлодіода, А.

По-друге, Закон Ома лежить в основі регулювання струму через світлодіод. Яскравість світлодіода прямо пропорційна силі струму, що протікає через нього. При аналоговому керуванні, зміна яскравості досягається шляхом безперервної зміни цього струму. Це реалізується зміною керуючої напруги, яка подається на відповідну керуючу схему (наприклад, спеціалізований драйвер або операційний підсилювач). Згідно з Законом Ома, будь-яка зміна напруги, прикладеної до ділянки кола з певним опором, призводить до пропорційної зміни струму, що дозволяє плавно регулювати яскравість світлодіода.

Таким чином, Закон Ома є основоположним для всіх розрахунків та проектних рішень, що стосуються формування, стабілізації та регулювання струму, який живить світлодіод, забезпечуючи точне та передбачуване аналогове керування його яскравістю.

Переваги аналогового керування струмом:

- Плавне регулювання: Забезпечує надзвичайно плавні переходи між рівнями яскравості, оскільки струм змінюється безперервно, без дискретних кроків. - Відсутність мерехтіння: Світлодіод працює в постійному режимі, що повністю виключає ефект мерехтіння. Це критично важливо для освітлювальних приладів, призначених для відеозйомки, фотографії, або для середовищ, де чутливість до мерехтіння може викликати дискомфорт.

- Висока точність: З використанням прецизійних компонентів можна досягти дуже високої точності підтримки заданого струму, а отже, і яскравості.

Недоліки аналогового керування струмом:

- Енергоефективність: При низьких рівнях яскравості (малих струмах) ефективність світлодіодів може знижуватися, а частина енергії може розсіюватися у вигляді тепла в регулюючій схемі, що призводить до меншої енергоефективності порівняно з широтно-імпульсною модуляцією (ШІМ).

- Термостабільність: Характеристики світлодіодів (світловий потік, пряма

напруга) залежать від температури. Підтримка стабільного струму не завжди гарантує стабільну яскравість у широкому діапазоні температур без додаткових схем компенсації.

- Складність реалізації для багатоканальних систем: Для керування великою кількістю світлодіодів або *RGB*-світлодіодів аналогове керування потребує більшої кількості ЦАПів та складніших драйверів, що збільшує вартість і складність схеми.

- Обмеженість динамічних ефектів: Хоча аналогове керування є точним, воно рідко використовується для створення складних динамічних ефектів (наприклад, швидких змін кольору або мерехтіння) через необхідність швидкої та точної зміни аналогового сигналу, що є складнішим у реалізації порівняно з цифровими методами.

- Відсутність безпосереднього програмного керування: Без ЦАП та додаткових аналогових ланцюгів, пряме керування яскравістю з мікроконтролера стає неможливим, що обмежує гнучкість у програмному керуванні.

Таким чином, аналогове керування струмом є ефективним для стабільного та плавного регулювання яскравості, особливо там, де відсутність мерехтіння є пріоритетом. Однак для складних динамічних світлових ефектів та високоінтегрованих систем частіше застосовуються цифрові методи.

1.2.2 Широтно-імпульсна модуляція (ШІМ/PWM)

Широтно-імпульсна модуляція (ШІМ), також відома як *PWM* (*Pulse Width Modulation*), є одним з найпоширеніших і найефективніших методів керування яскравістю світлодіодів та інших навантажень. Її принцип ґрунтується не на зміні сили струму, що протікає через компонент, а на періодичному, дуже швидкому вмиканні та вимиканні світлодіода. Основна ідея ШІМ полягає у зміні тривалості увімкненого стану імпульсу, або ширини імпульсу, відносно загальної тривалості періоду сигналу. Ключовим параметром ШІМ є шпаруватість (*duty cycle*), яка

визначає відношення часу, протягом якого світлодіод увімкнений, до повного періоду сигналу і зазвичай виражається у відсотках (рисунок 1.9).

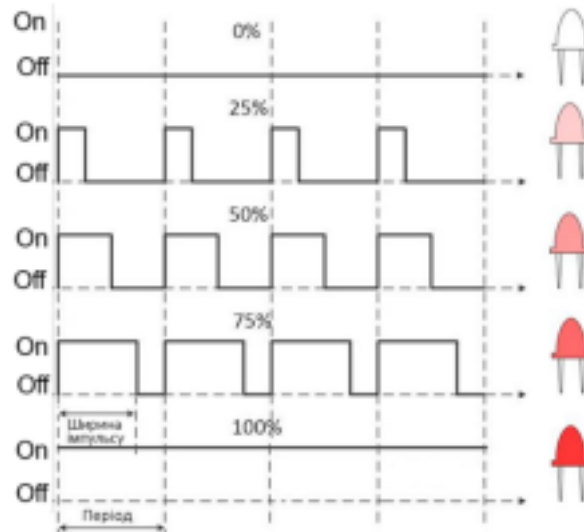


Рисунок 1.9 – Широтно-імпульсний модульований сигнал

Уявімо, що період ШІМ-сигналу є постійним. Якщо ширина імпульсу дорівнює половині періоду (тобто шпаруватість становить 50%), світлодіод буде увімкнений протягом половини часу і вимкнений протягом іншої половини. Оскільки частота ШІМ, зазвичай, є дуже великою (сотні Герц або кілогерци), людське око через свою інерційність не помічає швидкого мерехтіння світлодіода. Замість цього, мозок інтегрує імпульси світла, і візуально здається, що світлодіод світиться з половинною інтенсивністю. При збільшенні ширини імпульсу, наприклад, до 75% шпаруватості, світлодіод буде увімкнений 75% часу, і його яскравість візуально становитиме три чверті від повної. Шпаруватість у 100% означає, що світлодіод увімкнений постійно, забезпечуючи максимальну яскравість, тоді як 0% шпаруватості означає, що світлодіод завжди вимкнений.

Переваги ШІМ-керування:

- Висока енергоефективність: Світлодіоди працюють найефективніше при повній потужності. ШІМ дозволяє світлодіоду працювати в режимі "увімкнено/вимкнено" з мінімальними втратами енергії у керуючій схемі, оскільки ключові елементи перебувають або у повністю відкритому (низький опір), або у повністю закритому (високий опір) стані.

- Точне та стабільне керування: Яскравість контролюється цифровими імпульсами, що робить її менш залежною від коливань напруги живлення, температури або індивідуальних характеристик світлодіодів порівняно з аналоговим керуванням.

- Пряме програмне керування: ШІМ-сигнали можуть генеруватися безпосередньо вихідними пінами мікроконтролера, часто за допомогою вбудованих апаратних ШІМ-модулів. Це значно спрощує реалізацію програмного керування яскравістю та дозволяє створювати складні динамічні світлові ефекти, плавну зміну кольорів в *RGB*-системах, димінг тощо.

- Широкий діапазон регулювання: ШІМ дозволяє регулювати яскравість у дуже широкому діапазоні, від майже непомітного світіння до повної яскравості.
Недоліки ШІМ-керування:

- Можливість мерехтіння (стробоскопічний ефект): Хоча для більшості застосувань висока частота ШІМ усуває видиме мерехтіння, при певних обставинах (наприклад, при зйомці відео з високою частотою кадрів або в освітленні рухомих об'єктів) може виникати стробоскопічний ефект.

- Електромагнітні завади: Швидке перемикання великих струмів може генерувати електромагнітні завади (*EMI*), які можуть впливати на інші електронні компоненти. Це вимагає ретельного проектування схеми та фільтрації.

Завдяки своїй ефективності, точності та простоті інтеграції з цифровими системами, ШІМ є основним та найпоширенішим методом керування яскравістю світлодіодів у більшості сучасних застосувань, від побутової електроніки до промислового освітлення.

Обґрунтування вибору ШІМ для реалізації *RGB*-дерева

У даній кваліфікаційній роботі, при розробці системи керування *RGB* деревом з використанням зсувних регістрів *TPIC6B595N*, струмообмежувальних резисторів та *RGB*-світлодіодів, широтно-імпульсна модуляція (ШІМ) обрана як основний метод керування яскравістю. Таке рішення обґрунтовується низкою

ключових переваг, які ШІМ демонструє в порівнянні з аналоговими методами, особливо в контексті багатоканальних світлодіодних систем.

24

1. Енергоефективність: Використання ШІМ забезпечує високу енергоефективність системи. Оскільки світлодіоди працюють у режимі швидкого перемикачання "увімкнено/вимкнено", втрати потужності на керуючих елементах (зокрема на зсувних регістрах *TPIC6B595N*, які є джерелами струму) мінімізуються. Це дозволяє знизити енергоспоживання та тепловиділення, що є важливим для надійності та довговічності системи освітлення.

2. Точне та стабільне керування яскравістю: ШІМ-керування є цифровим за своєю природою, що забезпечує високу точність та стабільність яскравості світлодіодів. Яскравість визначається співвідношенням часу увімкненого стану до періоду (шпаруватістю), яке легко контролюється програмно. Ця перевага є критичною для *RGB*-систем, де точний контроль яскравості кожного кольору необхідний для формування широкого спектру відтінків. На відміну від аналогового керування, ШІМ менш чутлива до температурних дрейфів або індивідуальних відмінностей у характеристиках світлодіодів та напруги живлення.

3. Зручність програмної реалізації з мікроконтролером та зсувними регістрами: Зсувні регістри *TPIC6B595N* ідеально підходять для роботи з ШІМ сигналами. Вони дозволяють ефективно розширювати кількість вихідних каналів мікроконтролера, що є вкрай важливим для керування численними *RGB* світлодіодами. Мікроконтролери, як правило, мають вбудовані апаратні ШІМ модулі, що значно спрощує генерацію необхідних імпульсів без значного навантаження на центральний процесор. Це дозволяє легко реалізовувати складні динамічні світлові ефекти, анімації та плавні переходи кольорів через програмний контроль шпаруватості кожного *RGB*-каналу.

4. Спрощення схемотехніки для багатоканальних систем: На відміну від аналогового керування, яке потребувало б окремих цифро-аналогових перетворювачів (ЦАП) для кожного *RGB*-каналу (а отже, по одному ЦАП на кожен колір для кожного світлодіода у великій кількості), ШІМ дозволяє використовувати простіші цифрові виходи мікроконтролера та зсувних регістрів. Це значно зменшує складність, розміри та вартість апаратної частини, що

Враховуючи вищезазначені переваги, ШІМ є оптимальним вибором для керування RGB-деревом, забезпечуючи необхідну функціональність, гнучкість у програмуванні та ефективність системи.

1.3 Методи створення динамічних світлових ефектів за допомогою керування яскравістю дерева RGB

Динамічні світлові ефекти, що реалізуються за допомогою RGB-освітлення, надають широкі можливості для створення унікальної атмосфери та візуального акценту. Керування яскравістю кожного з трьох кольорових каналів (*Red*, *Green*, *Blue*) дозволяє не лише отримувати мільйони відтінків, але й контролювати їхню зміну у часі. Це відкриває шлях до створення різноманітних світлових сценаріїв, від плавних переходів до яскравих пульсацій та складних анімацій.

1.3.1 Плавне змінення кольорів (фейдинг)

Плавне змінення кольорів, або фейдинг (від англ. *fading* – згасання, плавний перехід), є одним із базових і найбільш естетично привабливих динамічних світлових ефектів, що досягається за допомогою RGB-освітлення. Суть цього ефекту полягає у безперервному та поступовому переході одного кольору в інший, створюючи м'які та гармонійні колірні переливи.

Технічна реалізація. Для плавного фейдингу зазвичай використовується широтно-імпульсна модуляція (ШІМ), оскільки вона дозволяє дуже точно й дискретно, але водночас візуально плавно, регулювати яскравість кожного світлодіода. Мікроконтролер або спеціалізований драйвер світлодіодів генерує ШІМ-сигнали для кожного RGB-каналу, постійно оновлюючи значення шпаруватості (*duty cycle*) з заданою швидкістю. Чим менші кроки зміни яскравості та чим вища частота оновлення, тим плавнішим буде перехід.

Застосування та ефект. Фейдинг широко застосовується для створення розслаблюючої або елегантної атмосфери. Його можна зустріти в архітектурному

у сценічному освітленні для створення певного настрою. Цей ефект не є агресивним для очей і дозволяє насолоджуватися грою кольорів без різких змін, що робить його універсальним рішенням для багатьох ситуацій. Різні швидкості фейдингу можуть створювати різні настрої – від повільних, медитативних переходів до більш швидких, динамічних змін.

1.3.2 Створення імпульсів та стробоскопічних ефектів

Створення імпульсів та стробоскопічних ефектів є ще одним різновидом динамічних світлових сценаріїв, що реалізуються за допомогою *RGB*-освітлення, і які відрізняються від плавного фейдингу своєю різкістю та динамікою. Ці ефекти спрямовані на привернення уваги, створення енергійної атмосфери або імітацію швидких змін.

Принцип роботи. Імпульси та стробоскопічні ефекти базуються на швидкому, часто миттєвому, вмиканні та вимиканні світлодіодів, або на різкій зміні їхньої яскравості та кольору. Замість плавних переходів, як у фейдингу, тут використовуються чіткі, дискретні стани: світло або є, або його немає, або його яскравість/колір різко змінюється.

Технічна реалізація. Для створення таких ефектів також широко застосовується широтно-імпульсна модуляція (ШІМ). Однак, замість плавних і послідовних змін шпаруватості (*duty cycle*), тут ми маніпулюємо нею дуже швидко, перемикаючи її між 0% (вимкнено) та 100% (повна яскравість) або іншими заданими значеннями.

Імпульси – це короткочасні спалахи світла, які можуть бути одного кольору або змінювати колір з кожним імпульсом. Вони створюються шляхом швидкого встановлення високої шпаруватості (наприклад, 100%) на дуже короткий період часу, а потім повернення до 0% або фонового рівня яскравості. Частота і тривалість імпульсів можуть бути налаштовані для створення різних ритмічних ефектів.

Стробоскопічні ефекти – це серія дуже швидких, коротких і повторюваних

спалахів світла. Ефект стробоскопа виникає, коли світлодіоди миттєво вмикаються і вимикаються з високою частотою, створюючи ілюзію сповільненого або

27

застиглого руху об'єктів. Для досягнення цього ефекту ШІМ-сигнал перемикається з 0% до 100% шпаруватості з високою, сталою частотою.

Застосування та ефект. Імпульсні та стробоскопічні ефекти активно використовуються у розважальній індустрії: на концертах, у нічних клубах, на дискотеках для створення енергійної, динамічної атмосфери. Вони також можуть застосовуватися в рекламних цілях для привернення уваги до вивісок або експонатів. У сценічному освітленні їх використовують для підкреслення певних моментів у виступах або для створення драматичних пауз. Важливо зазначити, що тривале перебування під стробоскопічним світлом може викликати дискомфорт або навіть проблеми зі здоров'ям у людей з фоточутливою епілепсією, тому їх використання вимагає відповідального підходу.

1.3.3 Реалізація складних динамічних патернів освітлення

Реалізація складних динамічних патернів освітлення є вершиною можливостей *RGB*-освітлення, що дозволяє створювати не просто окремі ефекти, а цілі світлові сценарії, що змінюються з часом і часто інтегровані з іншими елементами (музикою, відео, рухом). Це виходить за рамки простих переходів чи імпульсів, комбінуючи їх у складні, синхронізовані послідовності.

Принцип та можливості. Складні патерни освітлення досягаються шляхом програмованої та скоординованої зміни яскравості кожного *RGB*-каналу (або окремих *RGB*-світлодіодів/груп світлодіодів) у часі. Це дозволяє створювати:

- Багатоколірні анімації: Світло може "перетікати" через поверхні, створюючи ефект "хвилі", "водоспаду" або "пульсуючого дихання". -

Синхронізація з музикою: Світлові ефекти можуть реагувати на ритм, темп, гучність або навіть мелодію музики, створюючи захоплюючі аудіовізуальні шоу.

- Імітація природних явищ: Наприклад, імітація блискавки, мерехтіння

- Інтерактивні ефекти: Світло може реагувати на рух, дотик або інші зовнішні події, перетворюючи простір на динамічне та чутливе середовище. -

Сюжетні світлові шоу: У театрі, на концертах або в тематичних парках освітлення використовується для посилення сюжету, підкреслення емоцій або зміни місця дії.

Технічна реалізація. Для реалізації складних динамічних патернів необхідно:

- Мікроконтролер або спеціалізований контролер освітлення: Це "мозок" системи, який виконує заздалегідь запрограмовані алгоритми або реагує на вхідні дані. Сучасні контролери мають достатню обчислювальну потужність для керування тисячами *RGB*-пікселів.

- ШІМ-драйвери: Як і в попередніх ефектах, ШІМ є ключовим методом керування яскравістю окремих світлодіодів. Для великих систем використовуються багатоканальні ШІМ-драйвери, які отримують команди від контролера.

- Програмне забезпечення: Створення складних патернів часто вимагає спеціалізованого програмного забезпечення для розробки анімацій, налаштування часових ліній, синхронізації з аудіо/відео та програмування логіки реакцій. Це може бути як вбудоване ПЗ контролера, так і зовнішні програми на ПК.

- Адресні світлодіоди: Для дуже складних патернів, де кожен світлодіод або невелика група світлодіодів має індивідуально керуватися, використовуються адресні *RGB*-світлодіоди (наприклад, *WS2812*, *APA102*). Вони містять вбудований контролер, що значно спрощує розводку та керування.

Застосування. Складні динамічні патерни освітлення знаходять застосування у:

- Архітектурному освітленні: динамічне підсвічування будівель, мостів,

фонтанів.

- Сценічному та клубному освітленні: створення захоплюючих світлових шоу, що синхронізуються з музикою.

- Музеях та виставкових просторах: для привернення уваги до експонатів, створення імерсивних інсталяцій.

29

- Розважальних парках: динамічні елементи атракціонів.

- Рекламі та медіафасадах: великі екрани та вивіски, що відображають анімовані зображення.

Реалізація таких патернів вимагає комплексного підходу, поєднуючи знання електроніки, програмування та художнього дизайну світла.

30

РОЗДІЛ 2

ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТНОЇ БАЗИ

У цьому розділі буде представлено детальний вибір та обґрунтування компонентів, що складають апаратну частину проекту "RGB-дерево". Кожен елемент системи відіграє ключову роль у забезпеченні її функціональності, надійності та ефективності.

Спочатку буде розглянуто вибір RGB-світлодіодів, з урахуванням їхніх характеристик, необхідної яскравості та кута розсіювання для створення бажаного візуального ефекту. Далі буде обґрунтовано вибір мікроконтролера, який є "мозком" системи, з акцентом на його можливості генерації ШІМ-сигналів та взаємодії зі зсувними регістрами.

Особливу увагу буде приділено вибору зсувних регістрів, зокрема моделі *TPIC6B595N*, їхнім функціональним особливостям та перевагам для керування великою кількістю світлодіодів. Також буде розглянуто вибір резисторів для

струмообмеження та джерела живлення, що забезпечить стабільну та безпечну роботу всіх компонентів.

На завершення, буде обґрунтовано вибір проводів та інших допоміжних компонентів, а також матеріалів для виготовлення корпусу дерева, які забезпечать не лише електричне з'єднання, а й естетичний вигляд та механічну міцність усієї конструкції. Обґрунтування кожного вибору ґрунтуватиметься на технічних вимогах, доступності компонентів та економічній доцільності.

2.1 Обґрунтування вибору компонентів

Вибір правильних компонентів є критично важливим етапом у розробці будь якої електронної системи, особливо коли йдеться про створення динамічного *RGB* освітлення. Кожен елемент схеми — від джерела світла до керуючого мікроконтролера та елементів живлення — має відповідати поставленим

31

функціональним вимогам, забезпечувати надійність, ефективність та безпеку роботи. Ретельний підхід до вибору компонентів дозволяє досягти бажаних світлових ефектів, оптимізувати енергоспоживання та забезпечити довговічність пристрою.

2.1.1 Вибір *RGB*-світлодіодів

RGB-світлодіод є інтегрованим напівпровідниковим пристроєм, що містить три окремі емісійні кристали: червоного (*Red*), зеленого (*Green*) та синього (*Blue*) спектральних діапазонів. Ця архітектура дозволяє реалізувати адитивне змішування кольорів шляхом незалежного керування інтенсивністю світіння кожного кристала, що забезпечує формування широкого спектра відтінків видимого світла.

Конструктивно *RGB*-світлодіоди поділяються на два основні типи за схемою внутрішнього з'єднання електродів:

Зі спільним анодом (*Common Anode*): У цій конфігурації позитивні виводи (анооди) всіх трьох світлодіодних кристалів з'єднані між собою та виведені на один

спільний контакт. Для функціонування даного типу світлодіода спільний анод підключається до позитивного полюса джерела живлення, тоді як негативні виводи (католи) кожного окремого кристала (червоного, зеленого, синього) підключаються до відповідних керуючих елементів.

Зі спільним катодом (*Common Cathode*): На відміну від попереднього типу, у цій конфігурації негативні виводи (католи) всіх трьох кристалів об'єднані та виведені на спільний контакт. Спільний катод підключається до "землі" (мінусового полюса живлення), а позитивні виводи (аноли) кожного кристала підключаються до керуючих елементів, які подають на них позитивну напругу для вмикання.

Для даної кваліфікаційної роботи було обрано *RGB*-світлодіоди зі спільним анодом (рисунок 2.1), що зумовлено їхньою оптимальною сумісністю зі зсувним регістром *TPIC6B595N*, який використовується в керуючій схемі. Особливість регістра *TPIC6B595N* полягає в тому, що його виходи працюють за принципом "з'єднання з землею" (*sink current*), а не подачі напруги. Таким чином, для коректної

32

роботи *RGB*-світлодіода зі спільним анодом, його спільний анод підключається безпосередньо до джерела живлення, тоді як католи окремих кольорових кристалів підключаються до відповідних виходів зсувного регістра. Ця конфігурація забезпечує ефективне керування яскравістю кожного кольорового каналу, сприяє спрощенню схемотехнічного рішення та підвищує загальну стабільність і надійність системи. Саме ці аргументи стали визначальними при виборі світлодіодів зі спільним анодом замість світлодіодів зі спільним катодом.

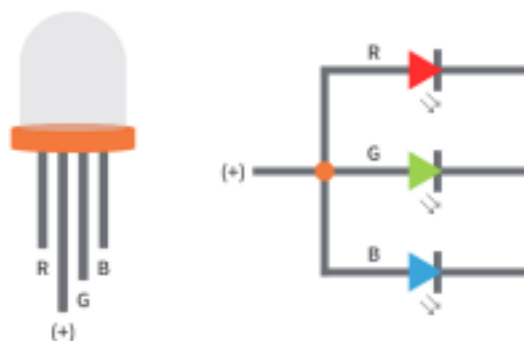


Рисунок 2.1 – Схема *RGB*-світлодіода зі спільним анодом

2.1.2 Вибір мікроконтролера

Arduino є відкритою платформою для розробки електронних пристроїв, що базується на мікроконтролерах і вирізняється високим рівнем доступності для широкого кола користувачів, від початківців до досвідчених інженерів. Програмування плат *Arduino* здійснюється за допомогою мови, яка є похідною від C/C++, в інтегрованому середовищі розробки (*IDE*) *Arduino*.

Екосистема *Arduino* охоплює різноманітні типи плат, кожна з яких характеризується унікальними технічними специфікаціями та цільовим призначенням. Для кращого розуміння їхніх відмінностей зазвичай надається порівняльна характеристика основних моделей (таблиця 2.1).

33

Таблиця 2.1 – Порівняння плат *Arduino* за апаратними параметрами

| Плата <i>Arduino</i> | Мікроконтролер | Робоча напруга (В) | Цифрові входи виходи | Виходи з ШІМ | Аналогові входи виходи | ОЗП (КБ) | Максимальний струм з піна вводу-виводу | Flash пам'ять (КБ) |
|----------------------|----------------------------------------|--------------------|----------------------|--------------|------------------------|----------|----------------------------------------|--------------------|
| Uno | <i>ATmega328</i> | 5 | 14 | 6 | 6 | 2 | 40 | 32 |
| Leonardo | <i>ATmega32u4</i> | 5 | 20 | 7 | 12 | 2,5 | 40 | 32 |
| Nano | <i>ATmega328</i> | 5 | 14 | 6 | 8 | 2 | 40 | 32 |
| Mega | <i>ATmega2560</i> | 5 | 54 | 14 | 16 | 8 | 40 | 256 |
| Due | <i>Atmel SAM3X8E ARM Cortex-M3 CPU</i> | 3.3 | 54 | 12 | 12 | 96 | 800 | 512 |
| Mini | <i>ATmega328</i> | 5 | 14 | 6 | 6 | 2 | 40 | 32 |
| Micro | <i>ATmega32u4</i> | 5 | 20 | 7 | 12 | 2,5 | 40 | 32 |
| M0 | <i>Atmel SAMD21</i> | 3.3 | 20 | 12 | 6 | 32 | 7 | 256 |
| LilyPad | <i>ATmega328p</i> | 2.7-5.5 | 20 | 6 | 6 | 2 | 40 | 32 |
| Fio | <i>ATmega328P</i> | 3.3 | 14 | 6 | 8 | 2 | 40 | 32 |
| Pro | <i>ATmega168 ATmega328P</i> | 3 або 5 | 14 | 6 | 6 | 1 | 40 | 16 |
| Robot | <i>ATmega32U4</i> | 5 | 5 | 6 | 4 | 2,5 | 40 | 32 |
| Esplora | <i>Atmel ATmega32u4</i> | 5 | - | - | - | 2,5 | - | 32 |

| | | | | | | | | |
|-----------------------------------|----------------------|-----------|----|----|---|-----|-------|-----|
| TRE | <i>Sitara AM335x</i> | - | 14 | 7 | 6 | 512 | - | 32 |
| LilyPad USB | <i>ATmega32U4</i> | 3,3 | 9 | 4 | 4 | 2,5 | 40 | 32 |
| LilyPad Arduino SimpleSnap | <i>ATmega328</i> | 2,7 - 2,5 | 9 | 5 | 4 | 2 | 40 | 32 |
| LilyPad Arduino Simple | <i>ATmega328</i> | 2,7 - 2,5 | 9 | 5 | 4 | 2 | 40 | 32 |
| Ethernet | <i>ATmega328</i> | 5 | 14 | 4 | 6 | 2 | 40 | 32 |
| YUN | <i>ATmega32U4</i> | 5 | 14 | 7 | 6 | 2,5 | 50-40 | 32 |
| ADK | <i>ATmega2560</i> | 5 | 54 | 15 | 4 | 8 | 40 | 256 |

У межах даної роботи було обрано мікроконтролерну плату *Arduino Nano* (рисунок 2.2). Такий вибір обґрунтований її компактними габаритними розмірами, що робить її ідеальною для інтеграції в обмежені простори та зручного монтажу на макетних платах. Незважаючи на свої мініатюрні розміри, *Arduino Nano* забезпечує повний функціонал, необхідний для реалізації поставлених завдань у роботі "Дерево *RGB*". Зокрема, вона має достатню кількість цифрових виходів для ефективного керування зсувними регістрами, а також підтримує зв'язок з персональним комп'ютером через інтерфейс *USB*, що значно спрощує процеси програмування та налагодження.

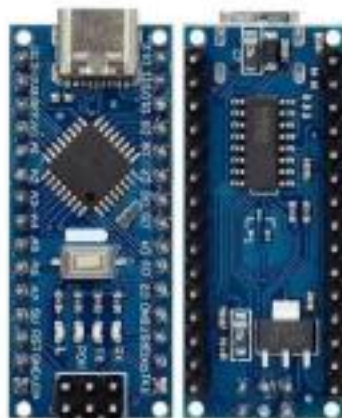


Рисунок 2.2 – Зовнішній вигляд мікроконтролера *Arduino Nano*

2.1.3 Вибір зсувних регістрів

Зсувний регістр є цифровою інтегральною мікросхемою, функціональне призначення якої полягає у перетворенні послідовних даних у паралельні, або, навпаки, паралельних у послідовні. Цей компонент є надзвичайно корисним у ситуаціях, коли виникає необхідність керування значною кількістю периферійних пристроїв, таких як світлодіоди, при обмеженій кількості доступних виводів мікроконтролера. Принцип його роботи полягає в послідовному прийомі бітів даних, їхньому збереженні у внутрішніх регістрах та подальшому одночасному виведенні на кілька паралельних виходів.

35

Для вибору найбільш підходящого зсувного регістру важливо порівняти їх основні характеристики (таблиця 2.2).

Таблиця 2.2 – Порівняльна таблиця характеристик зсувних регістрів

| Назва регістра | Тип виходів | Кількість виходів | Напруга живлення (V) | Максимальний струм на канал | Особливості та застосування |
|----------------|---------------------------------------------------------|-------------------|----------------------|-----------------------------|-----------------------------------------------------------------------------------|
| 74HC595 | Логічний (CMOS) | 8 | 2–6 | ~20 мА | Стандартний регістр, не призначений для великих навантажень |
| SN74HCT595N | Логічний (TTL сумісний) | 8 | 4.5–5.5 | ~35 мА | Сумісний із TTL рівнями, використовують для світлодіодів низької потужності |
| TPIC6B595N | Відкритий колектор (open-drain, N-канальні транзистори) | 8 | 4.5–5.5 | до 150 мА | Ідеальний для керування яскравими світлодіодами та навантаженнями високого струму |
| CD4094B | CMOS | 8 | 3–15 | ~10 мА | Підходить для низькошвидкісних застосувань |

У даній кваліфікаційній роботі було обрано інтегральну мікросхему

TPIC6B595N (рисунок 2.3). Вибір цієї моделі зумовлений її специфічною архітектурою з відкритим колектором, що робить її ідеальною для керування навантаженнями з високим струмом, зокрема *RGB*-світлодіодами. *TPIC6B595N* інтегрує функціональність стандартного зсувного регістра (аналогічного *74HC595*) з потужними вихідними транзисторами, які здатні працювати зі значно більшими навантаженнями без необхідності застосування додаткових зовнішніх підсилювачів. Ця комбінація забезпечує ефективне та спрощене керування яскравістю кожного каналу *RGB*-світлодіодів у проєкті. Додатково, використання *TPIC6B595N* сприяє підвищенню надійності системи та дозволяє легко масштабувати кількість керованих каналів без суттєвого ускладнення схемотехніки.

36



Рисунок 2.3 – Зсувний регістр *TPIC6B595N*

2.1.4 Вибір резисторів

Резистор (рисунок 2.4) є фундаментальним пасивним електронним компонентом, основною функцією якого є обмеження електричного струму в колі. Його застосування критично важливе для захисту інших елементів схеми від надмірних струмів, стабілізації рівнів напруги та формування необхідних електричних режимів роботи.

У цій кваліфікаційній роботі було передбачено використання резисторів номіналом 220 Ом. Цей вибір є обґрунтованим, оскільки такий опір забезпечує оптимальне обмеження струму для *RGB*-світлодіодів, підключених до зсувного регістра *TPIC6B595N*. Без належного обмеження струму, світлодіод міг би піддаватися надмірному протіканню струму, що потенційно призвело б до його перегрівання та, як наслідок, виходу з ладу. Номінал 220 Ом є типовим і широко рекомендованим для живлення стандартних світлодіодів від джерела напруги 5 В,

забезпечуючи при цьому помірну яскравість та безпечний рівень струму (близько 20 мА). Це значення струму відповідає як експлуатаційним характеристикам самих світлодіодів, так і технічним вимогам зсувного регістра, гарантуючи стабільну та надійну роботу всієї системи.



Рисунок 2.4 – Резистор 220 Ом

2.1.5 Вибір джерела живлення

З метою забезпечення стабільного енергопостачання системи, ми обрали стаціонарний блок живлення, що підключається до електричної мережі змінного струму з номінальною напругою 220 В. Такий вибір обґрунтований його зручністю, високою надійністю та здатністю забезпечувати стабільне функціонування всієї схеми. Зазначений блок живлення виконує функцію перетворення змінної напруги з електромережі на стабілізовану постійну напругу 5 В, яка є необхідною для коректної роботи мікроконтролерної плати *Arduino Nano*, зсувних регістрів *TPIC6B595N* та *RGB*-світлодіодів.

Перевага даного типу живлення полягає в його здатності забезпечити достатню потужність та струм, які вимагає схема, особливо в умовах одночасного функціонування значної кількості світлодіодів. Крім того, використання стаціонарного блоку живлення дозволяє інтегрувати пристрій у стаціонарні інсталяції без залежності від зовнішніх джерел живлення, таких як персональний комп'ютер, батареї або портативні акумулятори (*Power Bank*).

2.1.6 Вибір проводів та інших допоміжних компонентів

На основі проведеного аналізу було здійснено ретельний підбір та обґрунтування допоміжних компонентів, критично важливих для коректного з'єднання та монтажу електронних елементів схеми. Для комутації *RGB* світлодіодів обрано тонкі багатожилльні проводи. Такий вибір обґрунтований

їхньою гнучкістю та стійкістю до багаторазових згинань, що є важливою перевагою під час монтажу та подальшої експлуатації. Багатожилінні проводи мінімізують ризик пошкодження жили при механічних навантаженнях, які можуть виникнути під час складання та експлуатації пристрою. Хоча вони можуть вимагати трохи більше уваги при зачистці ізоляції перед паянням, їхні переваги в надійності та простоті укладання в компактній конструкції *RGB*-дерева є визначальними.

Для забезпечення надійного електричного контакту між компонентами застосовувалися стандартні інструменти та матеріали для пайки, включаючи паяльник, припій та каніфоль (флюс). З метою захисту місць пайки від механічних

38

пошкоджень та запобігання коротких замикань використовувалася термоусадкова трубка, яка забезпечує щільне облягання з'єднання після термічного впливу. На етапі розробки та налагодження схеми застосовувалася макетна плата. Її використання дозволяє оперативно організовувати тимчасові електричні з'єднання між компонентами без необхідності постійного монтажу, що значно прискорює процеси тестування та модифікації схематичних рішень.

Додатково, для інтерактивного керування режимами світіння *RGB*-дерева, було інтегровано тактові кнопки. Ці кнопки слугуватимуть засобом взаємодії з користувачем, дозволяючи йому перемикати заздалегідь запрограмовані світлові ефекти та патерни освітлення. Підключення тактових кнопок до мікроконтролера дає можливість реалізувати просту та інтуїтивно зрозумілу навігацію між різними режимами, розширюючи функціональність пристрою та підвищуючи зручність його використання.

2.1.7 Матеріали для виготовлення корпусу дерева

Для конструювання корпусу системи "Дерево *RGB*" було обрано матеріали, що характеризуються доступністю, легкістю в обробці та функціональною відповідністю поставленим завданням. Основу конструкції, що імітує стовбур та гілки дерева, формуватиме алюмінієвий дріт. Перевага алюмінію полягає у його високій пластичності, що забезпечує легкість деформації та утримання наданої форми, що є критично важливим для створення складної розгалуженої структури.

Крім того, алюмінієвий дріт легко обробляється за допомогою ручних інструментів, не вимагаючи спеціалізованого обладнання для формування.

В якості основи (підставки) конструкції буде використано горщик для квітів, який виконуватиме подвійну функцію: естетичну та функціональну. Всередині горщика буде розміщено макетну плату з електронними компонентами. Над макетною платою буде встановлено фанерну перегородку. Ця перегородка слугуватиме захисним бар'єром, запобігаючи потраплянню декоративних камінців (які будуть розміщені у горщику для створення естетичного вигляду та додаткової стійкості) на електронні компоненти та саму плату. Фанера обрана завдяки її

39

легкості, механічній міцності та простоті обробки (свердління для кріплення), що забезпечує надійну фіксацію та захист внутрішніх елементів системи. Для надання декоративного вигляду конструкції та маскуванню електронних проводів буде застосована фольга для обмотки каркасу. Окрім своєї маскувальної функції, фольга додає конструкції унікальний естетичний вигляд, створюючи імітацію металевого або футуристичного "фантастичного дерева".

40

РОЗДІЛ 3

ІНЖЕНЕРНА РЕАЛІЗАЦІЯ ДЕРЕВА *RGB*

3.1 Опис з'єднань та апаратної реалізації

RGB-дерево реалізується на основі мікроконтролера *Arduino Nano*, зсувних регістрів *TPIC6B595N*, резисторів та *RGB*-світлодіодів. Всі елементи змонтовані на макетній платі методом пайки для забезпечення надійності й компактності конструкції.

3.1.1 Підключення зсувних регістрів

При монтажі системи керування *RGB*-освітленням, зсувні регістри *TPIC6B595N* ретельно припаюються до макетної плати. Цей етап є критично

важливим для забезпечення надійного електричного контакту та механічної стабільності компонентів у схемі. Правильне позиціонування та якісна пайка гарантують безперебійну передачу даних та живлення, що є основою функціонування всієї системи керування світлодіодами.

3.1.2 Підключення резисторів

Для забезпечення безпечної роботи *RGB*-світлодіодів, до кожного з вихідних каналів *DRAIN0* – *DRAIN7* зсувного регістра *TPIC6B595N* послідовно підключається струмообмежувальний резистор номіналом 220 Ом (рисунок 3.1). Ці резистори припаюються між выводами регістра та відповідними катодами світлодіодів. Такий номінал (220 Ом) забезпечує оптимальний струм близько 20 мА, що захищає світлодіоди від перевантаження та відповідає можливостям регістра *TPIC6B595N*, гарантуючи стабільну роботу системи.

41

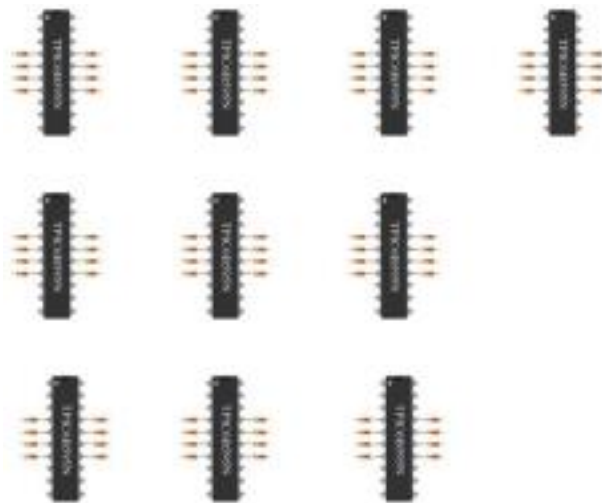


Рисунок 3.1 – Схема підключення струмообмежувальних резисторів до зсувних регістрів *TPIC6B595N*

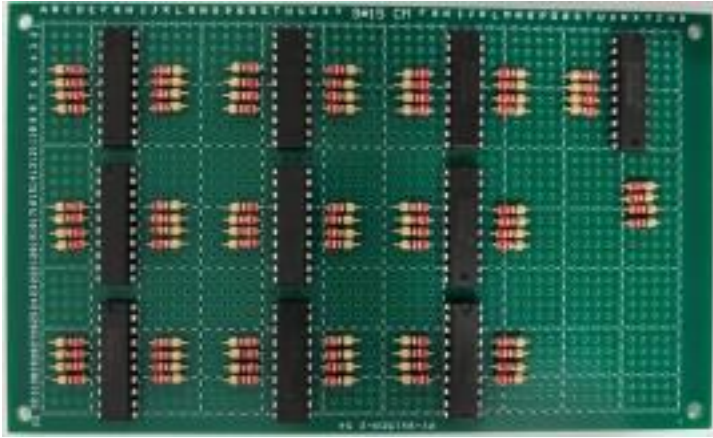


Рисунок 3.2 – Апаратна реалізація підключення резисторів до зсувних регістрів

3.1.3 Підключення RGB-світлодіодів із зсувними регістрами RGB- світлодіоди зі спільним анодом інтегруються в схему шляхом послідовного з'єднання з обмежувальними резисторами та зсувними регістрами *TPIC6B595N*. Кожен кольоровий кристал (червоний, зелений, синій) світлодіода підключається своїм катодом до індивідуального резистора номіналом 220 Ом. Кожен із цих резисторів, у свою чергу, з'єднаний з відповідним вихідним каналом (від *DRAIN0* до *DRAIN7*) зсувного регістра *TPIC6B595N* (як показано на рисунку 3.4). Таким чином, до кожної ніжки *DRAIN0-DRAIN7* припаяно по одному резистору 220 Ом.

42

Оскільки *TPIC6B595N* функціонує як "струмовий стік", а спільний анод *RGB* світлодіода підключений до джерела живлення +5В, струм проходить через світлодіод і резистор до регістра. Це забезпечує точне керування яскравістю кожного кольору.

Для коректної роботи системи важливо правильно організувати живлення та передачу даних між зсувними регістрами. У нашій схемі плюси живлення регістрів об'єднані та підключені до виводів 2 та 8 *TPIC6B595N*. Мінуси або загальна земля підключені до виводів 9, 10, 11 та 19. Передача даних між кількома зсувними регістрами здійснюється послідовно: вихід *SER OUT* попереднього регістра з'єднаний з входом *SER IN* наступного регістра, що дозволяє каскадувати регістри для керування великою кількістю світлодіодів.

Рисунок 3.3 – Електронна схема з *RGB* світлодіодами та каскадним з'єднанням зсувних регістрів

43

Рисунок 3.4 – Реалізація електронної схеми з *RGB*-світлодіодами та каскадним з'єднанням зсувних регістрів

3.1.4 З'єднання з *Arduino Nano*

З'єднання компонентів (рисунок 3.6) – зсувного регістра *TPIC6B595N* та

тактових кнопок – з платою *Arduino Nano* реалізується наступним чином для забезпечення керування *RGB*-світлодіодами та взаємодії з користувачем:

Підключення зсувного регістра *TPIC6B595N*:

- Лінія тактового сигналу (*Clock*): Цифровий вивід *D2* плати *Arduino Nano* підключається до виводу *13 (SRCK)* регістра *TPIC6B595N*. Цей сигнал використовується для синхронізації зсуву даних у регістрі.

- Лінія фіксації (*Latch*): Цифровий вивід *D3* плати *Arduino Nano* з'єднується з виводом *12 (RCK)* регістра *TPIC6B595N*. Сигнал *Latch* використовується для фіксації даних із зсувного регістра у вихідних буферах, після чого вони відображаються на виходах *DRAINx*.

44

- Лінія даних (*Data Input*): Цифровий вивід *D4* плати *Arduino Nano* підключається до виводу *3 (SER IN)* регістра *TPIC6B595N*. Це канал, по якому послідовно передаються дані для керування станом світлодіодів.

- Живлення логіки (*Vcc*): Вивід *2 (Vcc)* регістра *TPIC6B595N* підключається до виводу *5V* на платі *Arduino Nano*. Це забезпечує живлення логічної частини мікросхеми.

- Скидання/Вмикання виходів (*SRCLR*): Вивід *8 (SRCLR)* регістра *TPIC6B595N* також підключається до виводу *5V* на платі *Arduino Nano* (тобто повинен бути у стані *HIGH*). Це запобігає постійному скиданню регістра і дозволяє йому працювати нормально.

- Загальна земля (*GND*): Вивід *9 (G)* регістра *TPIC6B595N* підключається до виводу *GND* на платі *Arduino Nano*. Крім того, виводи *10, 11 та 19* регістра також підключаються до *GND* плати *Arduino Nano* для забезпечення стабільної роботи.

Підключення тактових кнопок:

Для взаємодії з користувачем та керування функціоналом схеми використовуються тактові кнопки. Кожна кнопка підключена до аналогових

выводів *Arduino Nano* з використанням підтягуючих (*pull-down*) резисторів номіналом 10 кОм:

- Одна сторона кожної тактової кнопки підключається до виводу *5V Arduino Nano* (або до лінії живлення *+5V* на макетній платі).

- Інша сторона кожної кнопки підключається до відповідного аналогового входу *Arduino Nano* (*A0, A1, A2, A3*).

- Паралельно до кожного аналогового входу та кнопки (тобто від аналогового входу до *GND*) підключається резистор 10 кОм. Цей резистор забезпечує визначений стан входу (*LOW*), коли кнопка не натиснута, запобігаючи "плаваючому" стану та хибним спрацьовуванням.

Таким чином, ці з'єднання дозволяють *Arduino Nano* надсилати послідовні дані до *TPIC6B595N* для керування світлодіодами, а також отримувати вхідні сигнали від користувача через тактові кнопки.

Рисунок

3.6 – Прототип схеми керування *RGB*-світлодіодами з інтерфейсом на тактових кнопках

46

3.2 Програмна реалізація

Цей розділ охоплює програмну складову розробки, що забезпечує функціонування *RGB*-системи. Тут буде детально описано процес створення програмного забезпечення, починаючи від вибору відповідного середовища розробки та закінчуючи етапами тестування та налагодження. Основна увага приділяється алгоритмам, що керують динамічними світловими ефектами, та методам їхньої імплементації на мікроконтролері.

3.2.1 Вибір середовища розробки

Інтегроване Середовище Розробки (*IDE*) є невід'ємною складовою платформи *Arduino*. Це середовище забезпечує повний функціонал для розробки програмного забезпечення, включаючи текстовий редактор, засоби відображення повідомлень, консоль, а також панель інструментів та меню для керування функціями.

IDE реалізує взаємодію з апаратними компонентами *Arduino*, що дозволяє

завантажувати та налагоджувати програмний код. Отже, для імплементації функціоналу плати та розробки необхідно написати та завантажити скетч – термін, що позначає програму для керування мікроконтролером *Arduino*.

Скетчі (програмний код для *Arduino*) (рисунок 3.8) створюються у вбудованому текстовому редакторі та зберігаються з розширенням файлу *.ino*. Функціонал редактора включає стандартні операції редагування тексту, такі як вирізання/вставка, а також можливості пошуку та заміни.

Область повідомлень *IDE* надає користувачеві оперативний зворотний зв'язок під час процесу компіляції коду, інформуючи про його успішне завершення або виявляючи синтаксичні та логічні помилки. Повноцінна інформація щодо процесу компіляції та детальний опис виявлених помилок відображаються у текстовій консолі. Додатково, у правому нижньому куті вікна *IDE* відображаються поточні налаштування обраної плати *Arduino* та підключеного послідовного порту, що забезпечує контроль за конфігурацією апаратного зв'язку.

скетчу

Рисунок 3.8 – Інтерфейс середовища програмування *Sketch*

Детальний аналіз функціонального призначення кожної з кнопок інтерфейсу програми представлений на рисунку 3.8.

- 1 - компіляція коду;
- 2 - завантаження програми;
- 3 - створення нового файлу;
- 4 - завантаження готового коду із бібліотеки;
- 5 - збереження коду на ПК.

48

3.2.2 Алгоритм роботи програми

Програма на *Arduino Nano* розроблена для керування *RGB*-світлодіодами через зсувні регістри *TPIC6B595N*, а також для взаємодії з користувачем через тактові кнопки. Основний алгоритм передбачає ініціалізацію всіх підключених компонентів, циклічне опитування стану кнопок та відповідне оновлення послідовних даних, що надсилаються до зсувних регістрів для керування світлодіодами. Тактові кнопки використовуються для зміни режиму освітлення *RGB*-світлодіодів.

3.3.3 Розробка програмного коду

Для реалізації керування світлодіодами через зсувні регістри було розроблено програмний код мовою *Arduino (C++)*, який виконує основні функції: ініціалізація, зчитування кнопок, перемикання між режимами, а також керування *LED*-ефектами.

Програмний код можна умовно поділити на кілька ключових частин: **1. Ініціалізація пінів і змінних**

У блоці оголошення змінних встановлюються номери пінів *Arduino*, які підключені до зсувних регістрів та кнопок, а також масиви для зберігання стану виходів і керування режимами (рисунок 3.9).

Рисунок 3.9 – Початковий фрагмент коду *Arduino IDE*: оголошення пінів та глобальних змінних для роботи з апаратною частиною

49

2. Робота з кнопками

Кожна кнопка перемикає між двома режимами освітлення. Реалізована система *debounce* для уникнення помилкових спрацьовувань. При натисканні викликається відповідна функція з масиву режимів.

Логіка керування режимами освітлення реалізована через кнопки у функції *loop()* (рисунок 3.10).

Рисунок 3.10 – Фрагмент програмного коду *Arduino IDE*, що демонструє логіку керування режимами освітлення через кнопки у функції *loop()*

50

3. Реалізація режимів освітлення

Кожен режим є окремою функцією, яка визначає певний ефект роботи світлодіодів: рух, заповнення, кольорова хвиля тощо.

Реалізацію режиму ***Knight Rider*** (вогні, що біжать) продемонстровано на рисунку 3.11.

Рисунок 3.11 – Фрагмент програмного коду *Arduino IDE*: реалізація світлового режиму "Вогні, що біжать" (*Knight Rider*)

Функція *showColor()* – відображення одного кольору, реалізується програмним кодом, зображеним на рисунку 3.12.

Рисунок 3.12 – Фрагмент програмного коду *Arduino IDE*: функція *showColor()* для встановлення одного кольору світлодіодів

51

4. Допоміжні функції для роботи з регістрами

Ці функції відповідають за керування окремими виходами та оновлення стану всіх зсувних регістрів (рисунок 3.13).

Рисунок 3.13 – Фрагмент програмного коду *Arduino IDE*: допоміжні функції для керування світлодіодами та зсувними регістрами

3.3 Функціональність та тестування розробленої системи

Завершення розробки дерева *RGB* (рисунок 3.10) стало кульмінацією кваліфікаційної роботи. На цьому етапі ми прийшли до висновку щодо фінальної архітектури та готовності пристрою.

52

Ми обрали режими, які забезпечують різноманітні світлові ефекти, та інтегрували кнопки для їхнього зручного перемикання. На цьому етапі ми провели ретельне тестування та оптимізацію, щоб переконатися, що всі функції працюють стабільно та бездоганно. Також ми приділили увагу документуванню коду, щоб він залишався зрозумілим і керованим для подальших змін.

Зовнішній вигляд та ключові елементи дерева *RGB* представлені на рисунку 3.14.

Рисунок 3.14 – Дерево *RGB*

53

ВИСНОВКИ

У цій кваліфікаційній роботі нами була успішно реалізована інтерактивна *RGB*-гірлянда у формі дерева, що стало наочною демонстрацією та підтвердженням можливостей програмованого керування освітленням. Мета роботи — створення функціонального та гнучкого пристрою освітлення — була

повністю досягнута.

Для виконання поставлених завдань, ми перш за все глибоко дослідили теоретичні основи функціонування *RGB*-світлодіодів та ключові методи керування їхньою яскравістю та кольором. Особливу увагу було приділено технології широтно-імпульсної модуляції (ШІМ), яка є фундаментальною для досягнення плавних переходів та динамічних світлових ефектів. Це дозволило нам не лише сформуванню міцну теоретичну базу, але й обґрунтувати вибір конкретних алгоритмів для створення різноманітних візуалізацій, від простих заливок до складних рухомих патернів.

Вибір компонентної бази кваліфікаційної роботи був ретельно обґрунтований, враховуючи вимоги до функціональності, надійності та економічної доцільності. Ми визначили оптимальні моделі *RGB*-світлодіодів, мікроконтролера *Arduino Nano* як центрального елемента керування, зсувних регістрів *74HC595* для розширення кількості виходів, а також необхідних резисторів, джерела живлення та допоміжних матеріалів. Кожен обраний елемент відповідав визначеним критеріям, забезпечуючи стабільну та ефективну роботу всієї системи.

Інженерна реалізація проєкту охопила як апаратну, так і програмну складові. Було детально сплановано та здійснено фізичне підключення всіх електронних компонентів, забезпечуючи їх коректну взаємодію. Паралельно велася розробка програмного коду, який є "мозком" системи. У рамках цієї розробки ми інтегрували низку світлових режимів, від класичних рухомих вогнів до повних кольорових заливок, та реалізували інтуїтивне керування ними за допомогою чотирьох тактових кнопок. Кожна кнопка дозволяє перемикатися між двома призначеними їй

54

режимами, що значно розширює інтерактивність та зручність використання пристрою.

На етапі завершення розробки ми прийшли до висновку, що всі цілі, поставлені перед кваліфікаційною роботою, були успішно досягнуті. Проведено всебічне тестування функціоналу, включаючи перевірку всіх світлових режимів та коректності роботи кнопок, а також здійснено необхідну оптимізацію коду для

підвищення його стабільності та ефективності. Результатом є повністю функціональна *RGB*-гірлянда, яка не тільки демонструє отримані знання та навички в галузі електроніки та програмування мікроконтролерів, але й має потенціал для подальшого розширення та вдосконалення.

55

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бондаренко В. В., Діденко В. І., Куликов С. В. Основи світлотехніки : підручник. Вінниця : ВНТУ, 2020. 415 с.
2. Гончаров М. С., Дем'яненко І. І. Мікроконтролери. Теорія та практика застосування : підручник. Дніпро : Східний видавничий дім, 2018. 420 с. 3. Григор'єв В. О. Цифрова схемотехніка та мікропроцесорна техніка : підручник. Київ : Університет, 2019. 400 с.
4. Казакевич В. С. Цифрова електроніка : навч. посіб. Львів : Новий Світ 2000, 2017. 350 с.
5. Коваленко І. А., Петренко Р. О. Електроніка та мікросхемотехніка : підручник. Харків : Технології, 2021. 510 с.
6. Мартін Ван. Ефективне керування світлодіодами: дослідження ШІМ затемнення. URL: <https://www.ledylighting.com/uk/led-pwm-dimming/> (дата звернення: 24.04.2025)
7. Матвієнко Сергій. ШІМ в ESP8266. URL: <https://itmaster.biz.ua/electronics/esp8266/esp8266-pwm.html> (дата звернення: 28.04.2025)
8. Мельник Л. В., Савчук В. М. Основи програмування мікроконтролерів *Arduino* : навч. посіб. Львів : Промінь, 2019. 320 с.
9. Павлишин В. В., Демчук М. І. Світлодіодні технології в освітленні: монографія. Івано-Франківськ : Факел, 2018. 280 с.
10. Петров О. В., Сидоренко І. М. Теорія автоматичного керування: підручник. Київ : НТУУ "КПІ", 2020. 480 с.
11. Проектування вбудованих систем / за ред. П. П. Петрова. Одеса: Астропринт, 2022. 580 с.
12. Савченко А. В. Оптоелектронні прилади та пристрої: підручник. Київ: Ліра-К, 2019. 390 с.

13. Турчин В. М., Осадча О. Г. Методи керування яскравістю RGB-діодів для створення динамічного освітлення. *Авіація і космонавтика*: тези допов. IV Міжнародна науково-практична конференція. Кривий Ріг, 2025 р. С. 209-211

56

14. Що таке RGB? *LEDYi Lighting*. URL:

<https://www.ledyilighting.com/uk/what-is-rgb-> (дата звернення: 08.04.2025).

15. Що таке світлодіод RGB: схема та його робота. *FMUSER.NET*. URL:

<https://uk.fmuser.net/content/?21078.html> (дата звернення: 01.04.2025).

16. Що таке ШІМ (Широтно-імпульсна модуляція)? *Bitkit*. URL: <https://bitkit.com.ua/chto-takoe-shim-shirotno-impulsnaya-> (дата звернення: 24.04.2025).

57

ДОДАТОК А

Лістинг програми

```
// Піни Arduino
const int dataPin = 4; // SER IN
const int latchPin = 3; // RCK
const int clockPin = 2; // SRCK
// Піни для кнопок (14, 15, 16, 17)
const int buttonPin1 = 14;
const int buttonPin2 = 15;
const int buttonPin3 = 16;
const int buttonPin4 = 17;
const int NUM_REGISTERS = 10;
const int NUM_OUTPUTS = NUM_REGISTERS * 8; // 80
виходів byte registerState[NUM_REGISTERS] = {0};
// --- Керування режимами ---
// Загальна кількість всіх унікальних режимів, які у вас є
// (4 динамічні + 1 повне заповнення + 3 базові кольори = 8 режимів) const int
TOTAL_MODES = 8; // Загальна кількість режимів, що можуть бути викликані
// Визначення тривалості режиму в циклі loop, якщо він працює в
автоматичному режимі.
```

```
// При перемиканні кнопками ця затримка не використовується в
loop() //, але може бути використана всередині функцій режимів.
const int MODE_DISPLAY_DURATION = 1000; // Це буде мінімальна
тривалість відображення, якщо режим коротший
// Змінні для керування кнопками та їх станом
unsigned long lastDebounceTime[4] = {0, 0, 0, 0}; // Час останнього
зчитування для кожної кнопки
const long debounceDelay = 50; // Затримка для уникнення "дребезгу" кнопки
(мс)
```

58

```
int buttonState[4] = {HIGH, HIGH, HIGH, HIGH}; // Поточний стан кнопки
(HIGH, якщо не натиснута)
int lastButtonState[4] = {HIGH, HIGH, HIGH, HIGH}; // Попередній стан
кнопки
// Масив пінів кнопок для зручності
const int buttonPins[] = {buttonPin1, buttonPin2, buttonPin3,
buttonPin4}; const int NUM_BUTTONS = sizeof(buttonPins) /
sizeof(buttonPins[0]); // Індекси поточного активного режиму для
кожної кнопки
// Ми будемо зберігати поточний режим для кожної пари
int currentModeIndex[NUM_BUTTONS] = {0, 0, 0, 0};
// Масиви, що визначають, які режими призначені для кожної
кнопки // Кожна кнопка буде перемикати між двома режимами
const int modePairs[NUM_BUTTONS][2] = {
  {0, 1}, // Кнопка 1 перемикає між knightRider (індекс 0) і colorChase (індекс
1)
  {2, 3}, // Кнопка 2 перемикає між snakeMode (індекс 2) і sequentialWave
(індекс 3)
  {4, 5}, // Кнопка 3 перемикає між fullFill (індекс 4) і showColor(0) (індекс 5)
{6, 7} // Кнопка 4 перемикає між showColor(1) (індекс 6) і showColor(2) (індекс
7)
```

```

};
// Тип для функцій режимів
typedef void (*LightModeFunction)();
// Функції для окремих кольорів (щоб їх можна було додати в масив
функцій) void runShowRed() { showColor(0); }
void runShowGreen() { showColor(1); }
void runShowBlue() { showColor(2); }
// Масив усіх функцій режимів у послідовності (важливо, щоб індекси
відповідали modePairs)
LightModeFunction allModes[] = {

knightRider, // Індекс 0
colorChase, // Індекс 1
snakeMode, // Індекс 2
sequentialWave, // Індекс 3
fullFill, // Індекс 4
runShowRed, // Індекс 5
runShowGreen, // Індекс 6
runShowBlue // Індекс 7
};
void setup() {
pinMode(dataPin, OUTPUT);
pinMode(latchPin, OUTPUT);
pinMode(clockPin, OUTPUT);
// Ініціалізація пінів кнопок як INPUT_PULLUP
for (int i = 0; i < NUM_BUTTONS; i++) {
pinMode(buttonPins[i], INPUT_PULLUP);
}
clearAllOutputs();
updateShiftRegisters();
randomSeed(analogRead(A0)); // Ініціалізація генератора випадкових чисел

```

```

}
void loop() {
    // Зчитуємо стан кнопок
    for (int i = 0; i < NUM_BUTTONS; i++) {
        int reading = digitalRead(buttonPins[i]);
        // Якщо стан кнопки змінився
        if (reading != lastButtonState[i]) {
            lastDebounceTime[i] = millis(); // Скидаємо таймер "дребезгу" }
            // Якщо пройшов час "дребезгу" і кнопка стабільна

            if ((millis() - lastDebounceTime[i]) > debounceDelay) {
                if (reading != buttonState[i]) {
                    buttonState[i] = reading;
                    // Якщо кнопка була натиснута (перехід з HIGH на LOW) if
                    (buttonState[i] == LOW) {
                        // Перемикаємо індекс режиму для цієї кнопки
                        currentModeIndex[i] = (currentModeIndex[i] == 0) ? 1 : 0; // Перемикаємо між
0 і 1
                        // Очищаємо всі виходи перед запуском нового режиму
                        clearAllOutputs();
                        updateShiftRegisters();
                        // Важливо: ми запускаємо лише один режим за раз,
                        // і він буде відображатися до наступного натискання кнопки. //
                        Викликаємо відповідний режим з масиву allModes
                        // Вибираємо режим з пари modePairs[i][0] або modePairs[i][1]
                        allModes[modePairs[i][currentModeIndex[i]]]();
                    }
                }
            }
            lastButtonState[i] = reading; // Зберігаємо поточний стан для наступної
ітерації

```

```
}  
delay(10);  
}
```

```
//
```

```
=====
```

```
//
```

61

ФУНКЦІЇ РЕЖИМІВ

```
//
```

```
=====
```

```
// Включити всі світлодіоди заданого кольору
```

```
// colorOffset: 0 = Red, 1 = Green, 2 = Blue
```

```
void showColor(int colorOffset) {
```

```
clearAllOutputs();
```

```
for (int i = colorOffset; i < NUM_OUTPUTS; i += 3) {
```

```
setOutput(i, true);
```

```
}
```

```
updateShiftRegisters();
```

```
}
```

```
// Послідовна хвиля: світлодіоди вмикаються і вимикаються по  
одному void sequentialWave() {
```

```
// Цей режим виконує свою анімацію самостійно
```

```
for (int j = 0; j < 2; j++) { // Виконати хвилю двічі, щоб вона була  
помітнішою for (int i = 0; i < NUM_OUTPUTS; i++) {
```

```
clearAllOutputs(); // Очищаємо перед кожним кроком, якщо потрібно
```

```
setOutput(i, true);
```

```
updateShiftRegisters();
```

```
delay(30);
```

```

}
for (int i = 0; i < NUM_OUTPUTS; i++) {
    setOutput(i, false);
    updateShiftRegisters();
    delay(30);
}
}

```

62

clearAllOutputs(); // Переконаємось, що все вимкнене після завершення анімації

```

    updateShiftRegisters();
}

```

// Режим "Змія"

```

void snakeMode() {

```

```

    const int trailLength = 10;

```

```

    int head = 0;

```

```

    for (int k = 0; k < NUM_OUTPUTS + trailLength + 20; k++) { // Збільшили

```

цикл для дольшої анімації

```

        clearAllOutputs();

```

```

        head = k;

```

```

        if (head < NUM_OUTPUTS) {

```

```

            setOutput(head, true);

```

```

        }

```

```

        for (int i = 1; i <= trailLength; i++) {

```

```

            int tailSegment = head - i;

```

```

            if (tailSegment >= 0 && tailSegment < NUM_OUTPUTS) {

```

```

                setOutput(tailSegment, true);

```

```

            }

```

```

        }

```

```

        updateShiftRegisters();

```

```

        delay(75);

```

```

}
clearAllOutputs(); // Переконаємось, що все вимкнене після завершення
анімації

updateShiftRegisters();
}
// Режим "Повне заповнення" - Всі світлодіоди світяться
одночасно void fullFill() {

for (int i = 0; i < NUM_OUTPUTS; i++) {
setOutput(i, true);
}
updateShiftRegisters();
// Цей режим просто вмикає всі світлодіоди. Вони залишаються світитися,
// доки не буде викликаний інший режим або clearAllOutputs() }
// Режим "Вогні, що біжать" (Knight Rider)
void knightRider() {
const int moveSpeed = 50;
for (int j = 0; j < 2; j++) { // Запускаємо анімацію двічі для кращого ефекту
for (int i = 0; i < NUM_OUTPUTS; i++) {
clearAllOutputs();
setOutput(i, true);
updateShiftRegisters();
delay(moveSpeed);
}
for (int i = NUM_OUTPUTS - 1; i >= 0; i--) {
clearAllOutputs();
setOutput(i, true);
updateShiftRegisters();
delay(moveSpeed);
}
}
}
}

```

```

    clearAllOutputs(); // Переконаємось, що все вимкнене після завершення
анімації

    updateShiftRegisters();
}

// Режим "Кольорова погоня"
void colorChase() {

    const int chaseSpeed = 70;
    for (int j = 0; j < 2; j++) { // Запускаємо анімацію двічі для кращого ефекту
for (int k = 0; k < NUM_OUTPUTS + 6; k++) { // Додано +6, щоб всі "кольори"
пройшли

```

64

```

    clearAllOutputs();
    if (k < NUM_OUTPUTS) {
        setOutput(k, true); // Перший "колір"
    }
    if (k - 3 >= 0 && k - 3 < NUM_OUTPUTS) {
        setOutput(k - 3, true); // Другий "колір"
    }
    if (k - 6 >= 0 && k - 6 < NUM_OUTPUTS) {
        setOutput(k - 6, true); // Третій "колір"
    }
    updateShiftRegisters();
    delay(chaseSpeed);
}
}

clearAllOutputs(); // Переконаємось, що все вимкнене після завершення
анімації

```

65

```

// ДОПОМІЖНІ ФУНКЦІЇ
//

```

```
=====
=====

// Встановити або скинути конкретний пін (0–79)
void setOutput(int pinIndex, bool state) {
    if (pinIndex < 0 || pinIndex >= NUM_OUTPUTS) return;
    int reg = pinIndex / 8;
    int bit = pinIndex % 8;
    if (state) {
        registerState[reg] |= (1 << bit);
    } else {
        registerState[reg] &= ~(1 << bit);
    }
}

// Очистити всі виходи
void clearAllOutputs() {
    for (int i = 0; i < NUM_REGISTERS; i++) {
        registerState[i] = 0;
    }
}

// Вивести дані в усі 10 регістрів
void updateShiftRegisters() {
    digitalWrite(latchPin, LOW);
    for (int i = NUM_REGISTERS - 1; i >= 0; i--) {
        shiftOut(dataPin, clockPin, MSBFIRST, registerState[i]);
    }
    digitalWrite(latchPin, HIGH);
}
```

КРИВОРІЗЬКИЙ ФАХОВИЙ КОЛЕДЖ
ДЕРЖАВНОГО НЕКОМЕРЦІЙНОГО ПІДПРИЄМСТВА
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

РЕЦЕНЗІЯ

на кваліфікаційну роботу

випускника спеціальності: 123 «Комп'ютерна інженерія»

відділення: комп'ютерної та програмної інженерії

циклова комісія: комп'ютерних систем та мереж

Вікторія ТУРЧИН

(ім'я, прізвище)

1. Актуальність теми: тема кваліфікаційної роботи «Дерево RGB на базі мікроконтролера Arduino» є актуальною та сучасною.
2. Кваліфікаційна робота відповідає темі, затвердженій наказом.
3. Завдання на виконання кваліфікаційної роботи виконано у повному обсязі.
4. У роботі послідовно розкриваються як теоретичні основи щодо принципів керування світлодіодними системами на базі мікроконтролерів Arduino, так і практичні питання проектування апаратної частини, розробки програмного забезпечення та тестування функціонального прототипу "Дерева RGB".
5. Пояснювальна записка оформлена відповідно до вимог чинних стандартів. Матеріал викладено чітко, логічно та послідовно. Текст роботи добре структурований, графічний матеріал (рисунки та таблиці) виконаний якісно, є наочним і доповнює текст.
6. Суттєвих недоліків, що знижують цінність роботи, не виявлено. У якості перспективних напрямків подальшого розвитку проекту "Дерево RGB" можна виділити інтеграцію можливості керування режимами роботи та кольорними ефектами за допомогою мобільного додатку.
7. Кваліфікаційна робота заслуговує оцінку «відмінно».

Рецензент _____ викладач вищої категорії

(науковий ступінь, посада)

« _____ » _____ 2025 р.

(підпис)

Світлана ДАЦЕНКО

(ім'я, прізвище)

З рецензією ознайомлений _____

(підпис)

Вікторія ТУРЧИН

(ім'я, прізвище)

}