

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)

Пристрій для керування лампою зі світлодіодною стрічкою на основі Arduino
(тема)

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

Кацімон О.Д.
(прізвище, ініціали)

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

Тип програми освітньо-професійна
(код і повна назва спеціальності)

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

Керівник ас. Кулак Г.К.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Чумаченко С.В
(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)

« _____ » _____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувача _____ Кацімону Олегу Дмитровичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Пристрій для керування лампою зі світлодіодною
стрічкою на основі Arduino

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

2. Термін подання здобувачем роботи до екзаменаційної комісії _____ 17.06.2025 _____

3. Вихідні дані до роботи _____
Arduino Uno R3

Світлодіодна стрічка WS2812B, кнопка, резистор

Середовище моделювання Tinkercad

Мова програмування C++

4. Перелік питань, що потрібно опрацювати в роботі _____
Аналіз предметної області.

Розробка алгоритму роботи.

Вибір компонентів.

Тестування роботи пристрою. Програмна реалізація пристрою.

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Видача теми проекту, узгодження і затвердження теми	06.05.2025-07.05.2025	
2	Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів	07.05.2025 - 09.05.2025	
3	Розробка структурної схеми пристрою, вибір апаратної платформи	09.05.2025 -13.05.2025	
4	Розробка функціональної схеми пристрою	13.05.2025 -14.05.2025	
5	Розробка програми для пристрою	14.05.2025 -20.05.2025	
6	Тестування системи	20.05.2025 -22.05.2025	
7	Оформлення пояснювальної записки	22.05.2025 -27.05.2025	
8	Перевірка виконаного проекту керівником, допуск до захисту	27.05.2025 -17.06.2025	
9	Захист проекту	17.06.2025 -20.06.2025	

Здобувач _____
(підпис)

Керівник роботи _____ ас. Кулак Г.К.
(підпис)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи бакалавра: 54 с., 19 рис.,
12 джерел.

ARDUINO UNO, ПРИСТРІЙ ДЛЯ КЕРУВАННЯ, МІКРОКОНТРОЛЕР, РЕЗИСТОР, СВІТЛОДІОДНА СТРІЧКА, WS2812B

Метою кваліфікаційної роботи є розробка мікроконтролерного пристрою для керування світлодіодною стрічкою. В ході роботи було досліджено сучасні методи автоматизації освітлення, проаналізовано їх функціональні можливості, переваги та недоліки, вивчено типові архітектури подібних систем.

Було вирішено використати класичну схему з адресною світлодіодною стрічкою WS2812B, яка дозволяє незалежно керувати кожним світлодіодом NeoPixel. Створений пристрій складається з плати Arduino Uno, світлодіодної стрічки, кнопки для перемикання режимів та резистора для захисту сигнальної лінії. Розроблений пристрій дозволяє вмикати та змінювати світлові ефекти за допомогою кнопки.

ABSTRACT

The explanatory note of the qualification work contains: 54 pages, 19 pictures, 12 sources of links.

MODULE, PROJECT, MICROCONTROLLER, PIN, ARDUINO, AUTO WATERING SYSTEM

The goal of the qualification work is to develop a microcontroller device for controlling an LED strip. In the course of the work, modern methods of lighting automation were investigated, their functionality, advantages and disadvantages were analyzed, and typical architectures of such systems were studied.

It was decided to use the classic WS2812B addressable LED strip scheme, which allows for independent control of each NeoPixel LED. The created device consists of an Arduino Uno board, an LED strip, a button for switching modes, and a resistor to protect the signal line. The developed device allows you to turn on and change light effects using the button.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ СИСТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Мікроконтролерні системи	9
1.2 Вибір мікроконтролера.....	11
1.2.1 Arduino Nano.....	11
1.2.2 Arduino Uno.....	15
1.2.3 ESP8266.....	18
1.2.4 STM32103	22
1.2.5 Порівняння та остаточний вибір мікроконтролера	25
2 АПАРАТНА РЕАЛІЗАЦІЯ ПРОЕКТУ	27
2.1 Вибір складових частин для пристрою керування освітленням.....	27
2.1.1 Світлодіодна стрічка	28
2.1.2 Резистор	30
2.1.3 Кнопка.....	32
3 ВИБІР ТА ПОРІВНЯННЯ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ	35
3.1 Вибір середовища моделювання	35
3.1.1 Wokwo Arduino Simulator	35
3.1.2 Fritzing.....	36
3.1.3 Tinkercad	37
3.1.4 Порівняння та остаточний вибір середовища моделювання	37
3.2 Вибір мови програмування.....	38
3.2.1 C++.....	39
3.2.2 BlocklyDuino	39
3.2.3 Остаточний вибір мови програмування.....	40
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА АНАЛІЗ ПРОЕКТУ.....	41
4.1 Бібліотека Adafruit_NeoPixel	41
4.2 Опис програмної реалізації.....	42
4.3 Аналіз проекту.....	49
ВИСНОВКИ	52
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	53
ДОДАТОК А	55
ДОДАТОК Б.....	61

ВСТУП

Розвиток сучасних технологій і мікроконтролерів дозволяє реалізовувати нові ідеї взаємодії з побутовими приладами, які працюють на електриці, це стосується і освітлення. Можливості, які відкриваються при використанні світлодіодних стрічок зробили їх неймовірно привабливими для освітлення інтер'єрів по типу декоративних, комерційних та домашніх.

Цю популярність можна обґрунтувати їх довговічністю, можливостями створювати різноманітні світові ефекти, енергоефективністю та низьким рівнем тепловиділення. Для максимально ефективного використання систем подібного типу потрібно задіяти інтелектуальні рішення для управління освітленням, які будуть задовольняти потреби окремих користувачів щодо роботи світлодіодних стрічок. Використання саме мікроконтролера Arduino обумовлено тим, що він є одним з найбільш ефективних та доступних технологій серед аналогів подібного типу та також дає змогу реалізовувати змінні та адаптивні системи управління освітленням.

Не дивлячись на різноманітні можливості застосування Arduino для керування освітленням, досі до кінця не вивчені можливості інтегрування таких систем у різні побутові умови. Наявні можливості по більшій частини зводяться до звичайних функцій по типу вмикання або вимикання світла, зміна кольору або яскравості світла. Саме тому саме процес інтегрування таких систем до інших сенсорів або голосових помічників є дуже важливим та необхідним питанням подальших досліджень.

На сьогоднішній день зростає попит та інтерес до систем розумного освітлення через те, що за їх допомогою можна значно зменшити свої витрати на освітлення та зробити життя більш комфортним. Це проявляється у тому, що вже зараз існують системи з адаптивним освітленням, які споживають меншу кількість енергії та забезпечують його в залежності від активності

користувача, часу доби або ж навіть від настрою. Подальше впровадження таких рішень з іншими елементами розумного дому є затребуваним та актуальним напрямком на цей момент.

Цей науковий проект є актуальним, оскільки він полягає у розробці доступного та легкого в установці та налаштуванні пристрою для керування лампою зі світлодіодними стрічками, що надасть можливість впровадити його в будь-яке середовище. Система подібного типу може бути використана в будь-яких цілях, як і у комерційних так і побутових. Це надасть можливість користувачам придумувати та реалізовувати будь-які сюжети освітлення в залежності від індивідуальних потреб.

Отже, ця робота вплине на подальший розвиток розумних систем освітлення, які окрім підвищення легкості використання, також сприяють економії енергії та підвищують рівень комфорту в приміщеннях.

1 АНАЛІЗ СИСТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Мікроконтролерні системи

Мікроконтролери – це мініатюрні комп'ютери, спеціально розроблені для керування різноманітними пристроями та операціями. Вони дають змогу автоматизувати численні процеси, включаючи регулювання освітлення, контроль температури, вимірювання об'єму води або управління робототехнікою. Зазвичай мікроконтролери містять в собі низку важливих складових, які дозволяють їм виконувати свої функції. Також, важливо відзначити те, що мікроконтролери є фундаментом для створення вбудованих систем, наприклад, у ноутбуках, телефонах тощо.

Основним компонентом є центральний процесор, який представляє з себе основний елемент мікроконтролера, який виконує обчислення та відповідає за роботу всіх інших компонентів. Він виконує програму, яка зберігається в пам'яті і відповідає за роботу усіх інших компонентів мікроконтролера. Важлива роль у мікроконтролері займає пам'ять, яка поділяється на окремі типи:

- оперативна пам'ять (RAM) відповідає за тимчасове збереження даних, які використовуються під час роботи мікроконтролера;
- постійна пам'ять (ROM), де зберігається основна програма, яка не піддається змінам під час роботи пристрою;
- flash-пам'ять, яка зберігає програми, що піддаються змінам.

Важливим елементом мікроконтролера є вхідно-вихідні порти(GPIO), які відповідають за взаємодію з іншими пристроями та дозволяють під'єднувати різні компоненти по прикладу датчиків, світлодіодів чи кнопок. Працюють вони як на вхід, тобто отримання даних, так і на вихід для управління пристроями, або ж передачі даних. Що стосується роботи з аналоговими сигналами, то для цього мікроконтролер використовує

аналогово-цифрові та цифро-аналогові перетворювачі (ADC та DAC відповідно). АЦП перетворює аналоговий сигнал у цифровий, а ЦАП виконує зворотнє перетворення, тобто з цифрового в аналоговий.

Для відслідковування процесів або ж створення інтервалів часу мікроконтролери мають таймери та лічильники, які окрім цього дозволяють рахувати події та вимірювати час. У свою чергу, щоб обмінюватися інформацією з іншими пристроями, мікроконтролери мають спеціальні шинні комунікації, прикладом яких є I2C, SPI, UART. Їх функцією є передача даних між різними елементами системи. І, звичайно, для забезпечення живлення та його підключення використовуються спеціальні порти живлення.

Що стосується потужності та призначення мікроконтролерів, то тут можна виділити декілька типів. По-перше, це 8-бітні мікроконтролери, які використовують для простих завдань, таких як управління базовими пристроями, наприклад, в Arduino Uno на базі ATMEGA328. Вони обробляють дані в 8 біт, чого буде достатньо для виконання простих функцій. По-друге, це 16-бітні мікроконтролери, які мають більшу потужність і через це можуть виконувати складніші функції. Зазвичай вони мають популярність використання у пристроях середнього рівня складності. Якщо брати до уваги більш складні обчислення та великих обсягів даних, то частіше за все використовують 32-бітні мікроконтролери, які здатні обробляти більшу кількість інформації, що робить їх дуже практичним для мобільних телефонів, робототехніки та інших складних пристроїв. Існують також і 64-бітні мікроконтролери, які використовуються в обчислювальних системах, що потребують високої продуктивності.

Мікроконтролери роблять пристрої більш ефективними, зручними у використанні та дають можливість до впровадження нових технологій у повсякденне життя. Саме тому вони і застосовуються у різноманітних областях, таких як побутова техніка, робототехніка, автомобільна промисловість та інше.

1.2 Вибір мікроконтролера

Найважливішим компонентом для пристрою, який дозволяє керувати світлодіодною стрічкою на основі Arduino є мікроконтролер, тому потрібно дуже ретельно перейти до його вибору та звернути увагу на основні параметри. Серед них можна виділити ціну, доступність підтримки та документації, зручність у програмуванні та використанні, а також кількість входів та виходів, які будуть необхідні при підключенні інших компонентів.

1.2.1 Arduino Nano

Arduino - це система електронного конструювання, яка включає мікроконтролер (крихітний комп'ютер) і програмне забезпечення, що дозволяє розробникам створювати програми для мікроконтролера. Вона надзвичайно проста у використанні, що робить її ідеальною для початківців, але досить потужна для розробки складних проектів.

За допомогою Arduino можна легко створювати пристрої, які можуть взаємодіяти з навколишнім середовищем, наприклад, вимірювати температуру, керувати освітленням, приводити в дію двигуни, виявляти рух або навіть створювати роботів. Програми для Arduino розробляються спеціальною мовою програмування C/C++, а середовище розробки Arduino IDE робить цей процес простим і доступним.

Arduino Nano (рис. 1.1) - одна з найкомпактніших і найулюбленіших плат в сімействі Arduino, яка використовує мікроконтролер на базі ATmega328P. При розробці цієї плати метою було вмістити якомога більше функцій на найменшій площі. Вона є частиною відкритої апаратної платформи, яка має величезну підтримку і спільноту, що дозволяє використовувати Arduino Nano для різних проектів, від найпростіших до найскладніших.

Arduino Nano є чудовим рішенням для проектів, де економія місця або мобільність пристрою є критично важливими, а також для створення

прототипів, оскільки вона є одночасно малою і продуктивною. Завдяки своїм компактним розмірам Arduino Nano ідеально підходить для розробки мобільних або інтегрованих пристроїв, де потрібна мініатюризація.



Рисунок 1.1 – Arduino Nano

Arduino Nano оснащений мікроконтролером ATmega328P, який є одним з найпопулярніших у світі. Плата працює на частоті 16 МГц, що забезпечує достатню швидкість для виконання численних завдань, таких як керування світлодіодами, серводвигунами або обробка даних з датчиків. Arduino Nano підтримує стандартне живлення 5 В, що забезпечує сумісність з більшістю компонентів, які підключаються до плати. Крім того, плата працює на 3,3 В, що робить її універсальним рішенням для підключення різноманітних датчиків та інших компонентів, які можуть потребувати більш низької напруги.

Незважаючи на скромні розміри (18x45 мм), Arduino Nano оснащена 14 цифровими виводами вводу/виводу (рис. 1.2), 6 з яких - ШІМ (широтно-імпульсна модуляція). Це відкриває можливість керувати різними компонентами, такими як світлодіоди або серводвигуни, дозволяючи регулювати їх швидкість або яскравість. Крім того, Arduino Nano має 8 аналогових входів, завдяки яким можна отримувати дані з різних датчиків, таких як датчики температури, вологості або освітленості. Плата підтримує

ряд комунікаційних інтерфейсів, таких як I2C, SPI і UART, що дозволяє інтегрувати її з зовнішніми пристроями і передавати дані по різних каналах зв'язку.

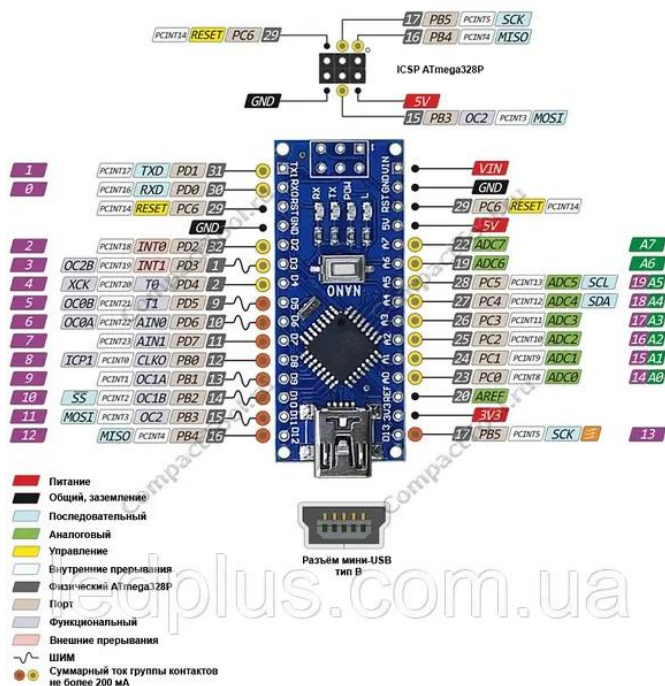


Рисунок 1.2 – Контакти Arduino Nano

Arduino Nano можна підключити до комп'ютера за допомогою USB-кабелю, що дозволяє не тільки програмувати плату, але й живити її від стандартного USB-порту. Це робить плату зручною і практичною для широкого кола користувачів. Якщо брати до уваги пам'ять, то її можна поділити на 3 типи:

- Flash пам'ять: 32 КБ, з яких 2 КБ зайняті завантажувачем для програмування через USB;
- SRAM: 2 КБ, використовується для зберігання змінних даних під час роботи програми;
- EEPROM: 1 КБ для зберігання постійних даних, які не втрачаються після вимкнення живлення.

Однією з ключових переваг Arduino Nano є її компактний розмір, що робить її відмінним рішенням для проектів, де важлива економія місця. Це дозволяє інтегрувати Arduino Nano в мобільні пристрої, для яких мініатюризація є критичною вимогою, наприклад, при розробці роботів або портативних сенсорних систем. Незважаючи на невеликі розміри, функціональність плати залишається на високому рівні, що дозволяє використовувати її навіть у складних проектах, які передбачають підключення великої кількості зовнішніх компонентів. Ще один важливий аспект - доступна вартість. Ціна є помірною, що робить її привабливою для широкого кола користувачів, від студентів до досвідчених професіоналів. Ціна дає можливість почати знайомство з мікроконтролерами, не витрачаючи багато грошей на обладнання. Крім того, простота програмування в середовищі Arduino IDE сприяє швидкому налаштуванню плати і дозволяє почати розробку програмного забезпечення навіть користувачам без попереднього досвіду програмування.

Величезна підтримка спільноти - ще одна велика перевага Arduino Nano. Завдяки численним підручникам, бібліотекам і форумам, ви можете швидко знайти рішення для будь-яких труднощів, які можуть виникнути під час розробки. Це значно полегшує як навчання, так і розробку, оскільки ви можете отримати допомогу від досвідчених користувачів і знайти готові рішення для своїх проектів. Крім того, Arduino Nano є гнучкою завдяки великій кількості виводів вводу/виводу, що дозволяє підключати різноманітні датчики, світлодіоди, двигуни та інші пристрої. Підтримка таких інтерфейсів, як I2C, SPI і UART, дозволяє підключати плату до інших пристроїв для більш складних проектів.

З іншого боку, обмежений обсяг пам'яті слід розглядати як один із недоліків цієї плати. Arduino Nano має лише 32 КБ флеш-пам'яті, з яких 2 КБ відведено під завантажувач, що може бути критично мало для масштабних або складних програм. Для проектів, які вимагають обробки великих обсягів інформації, кращим рішенням буде більш потужна плата, така як Arduino

Mega. Крім того, Arduino Nano має обмежену кількість контактів вводу/виводу. Хоча цього достатньо для більшості простих проектів, для складних систем, де потрібно підключити велику кількість компонентів, кількість виводів може стати вузьким місцем. У таких випадках ми рекомендуємо розглянути плату з більшою кількістю виводів, наприклад, Arduino Mega.

Поширеним недоліком є обмежена функціональність при взаємодії з більш складними датчиками або компонентами, які вимагають значного обсягу пам'яті або обчислювальної потужності. Arduino Nano може бути недостатньо ефективним для обробки великих обсягів даних або швидкої обробки сигналів, що обмежує його використання у високопродуктивних або промислових проектах.

Нарешті, обмеження по напрузі також можуть стати проблемою в деяких проектах. Arduino Nano працює на 5 В, але для взаємодії з компонентами, які потребують вищої напруги, може знадобитися додаткове обладнання для підвищення напруги живлення.

1.2.2 Arduino Uno

Arduino Uno (рис. 1.3) - одна з найвідоміших і найпоширеніших плат сімейства Arduino. В її основі лежить мікроконтролер ATmega328P, і вона є основною платою для численних проектів з електроніки та робототехніки. Arduino Uno - це проста в освоєнні плата, яка ідеально підходить як для початківців, так і для експертів, завдяки простоті програмування, підтримці спільноти та низькій ціні.

Arduino Uno використовується в прототипуванні, освітніх проектах та робототехнічних системах. Вона базується на відкритій платформі розробки, яка дозволяє легко реалізовувати проекти з наявними датчиками, актуаторами та іншими компонентами. Завдяки цьому Arduino Uno стала стандартом для багатьох освітніх програм та аматорів по всьому світу.

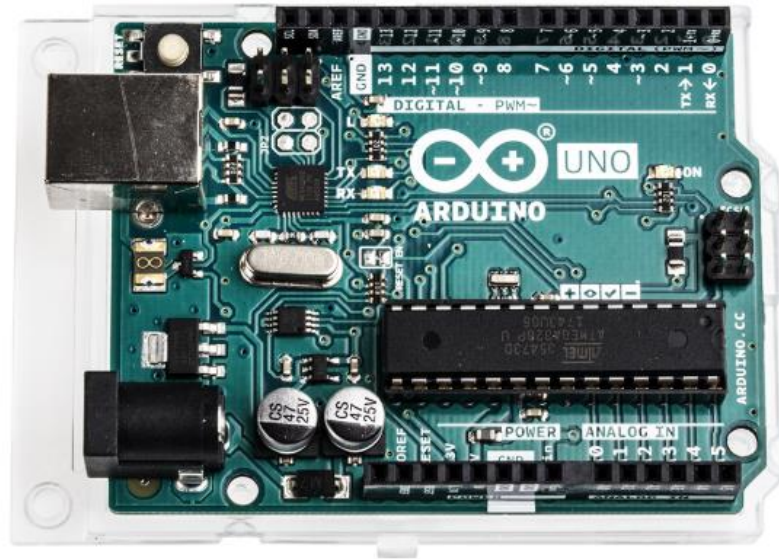


Рисунок 1.3 – Arduino Uno

Arduino Uno має певні особливості, які роблять його відмінним вибором мікроконтролера для створення проектів різного рівня складності. Серцем плати є мікроконтролер ATmega328P, ключовий елемент, що відповідає за обробку даних і виконання програмних інструкцій. Робоча частота цього мікроконтролера становить 16 МГц, що забезпечує прийнятну швидкість обчислень для більшості простих проектів. Що стосується пам'яті, то маємо 3 типи:

- Flash пам'ять: 32 КБ, з яких 0.5 КБ зарезервовано для завантажувача, що дозволяє легко програмувати плату через USB;
- SRAM: 2 КБ, використовується для тимчасового зберігання даних, що обробляються в процесі виконання програм;
- EEPROM: 1 КБ для зберігання постійних даних, таких як налаштування, або конфігурації.

Плата оснащена 14 цифровими входами/виходами (рис. 1.4), 6 з яких можна використовувати для широтно-імпульсної модуляції (ШІМ), що дозволяє керувати різними пристроями, такими як світлодіоди, двигуни та

інші компоненти, які потребують контролю інтенсивності світла або швидкості. Плата також має 6 аналогових входів, які можна використовувати для підключення різних датчиків, таких як датчики температури і вологості.

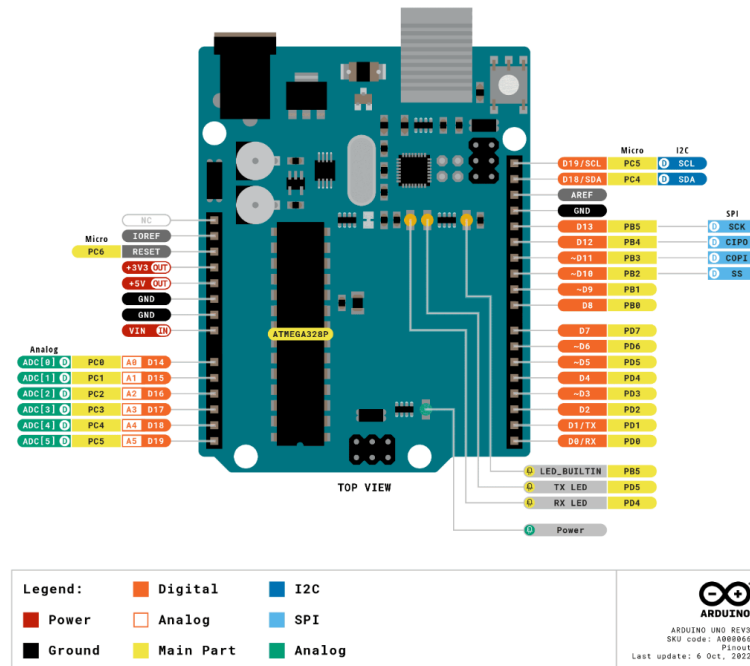


Рисунок 1.4 – Контакти Arduino Uno

Arduino Uno здатна видавати максимальний струм 40 мА на вивід, що дозволяє легко підключати до неї компактні пристрої без необхідності використання додаткових драйверів або транзисторів. З точки зору інтерфейсів, плата оснащена USB-роз'ємом для програмування та живлення. Також є UART (послідовний порт) для зв'язку із зовнішніми пристроями, I2C для підключення різних датчиків або дисплеїв і SPI для швидкого обміну даними з іншими пристроями.

Розміри плати складають 68,6 x 53,4 мм, що робить її зручною для використання в середніх і великих проектах. Вага ArduinoUno складає всього 25 грамів, що робить її досить легкою для використання в портативних пристроях.

Arduino Uno має багато переваг, які роблять його таким популярним серед програмістів та студентів. Перше - це простота використання. Середовище розробки Arduino IDE, яке включає велику бібліотеку зразків коду, дозволяє початківцям швидко навчитися програмувати мікроконтролери і без особливих труднощів реалізовувати власні ідеї. Плата також пропонує велику кількість контактів для підключення різноманітних датчиків, реле, двигунів і світлодіодів.

Потужна підтримка спільноти - ще одна важлива перевага Arduino Uno. Будучи однією з найпоширеніших плат серед користувачів Arduino, ви маєте доступ до величезної кількості ресурсів, бібліотек, форумів та навчальних посібників. Ви можете легко знайти рішення будь-якої проблеми або отримати підтримку на форумах Arduino або інших онлайн-ресурсах.

Крім того, Arduino Uno має доступну ціну, що робить його ідеальним інструментом для навчання та створення прототипів без значних фінансових вкладень. Він ідеально підходить для розробки простих і помірно складних проектів, таких як системи управління освітленням, робототехніка, датчики температури або навіть для автоматизації повсякденних завдань.

Водночас Arduino Uno має деякі недоліки: перш за все, обмежений обсяг пам'яті. На відміну від більш потужних платформ, таких як Arduino Mega, Arduino Uno має лише 32 КБ флеш-пам'яті, чого може бути недостатньо для великих проектів або складних алгоритмів обробки даних. Якщо ваш проект вимагає значного обсягу пам'яті або великих обсягів даних, Arduino Mega або інші більш потужні мікроконтролери можуть бути кращим рішенням.

1.2.3 ESP8266

ESP8266 (рис. 1.5) - це широко використовуваний мікроконтролер з вбудованим Wi-Fi, що робить його відмінним вибором для проектів Інтернету

речей (IoT). Його основною перевагою є можливість зв'язуватися з бездротовими мережами і передавати дані по Wi-Fi без необхідності використання додаткових модулів або плат. ESP8266 завоював популярність завдяки низькому енергоспоживанню, доступній ціні та простоті використання, що робить його ідеальним для проектів, які потребують бездротового підключення, таких як автоматизоване освітлення та системи моніторингу.

ESP8266 базується на 32-розрядній архітектурі процесора і має вбудовану підтримку Wi-Fi, що дозволяє йому підключатися до Інтернету, передавати дані і працювати в якості сервера або клієнта в мережах IoT. Цей мікроконтролер завоював велику популярність як серед споживачів, так і серед професіоналів, знайшовши широке застосування в таких проектах, як розумне освітлення, системи моніторингу, домашня автоматизація, датчики і багато інших додатків IoT.

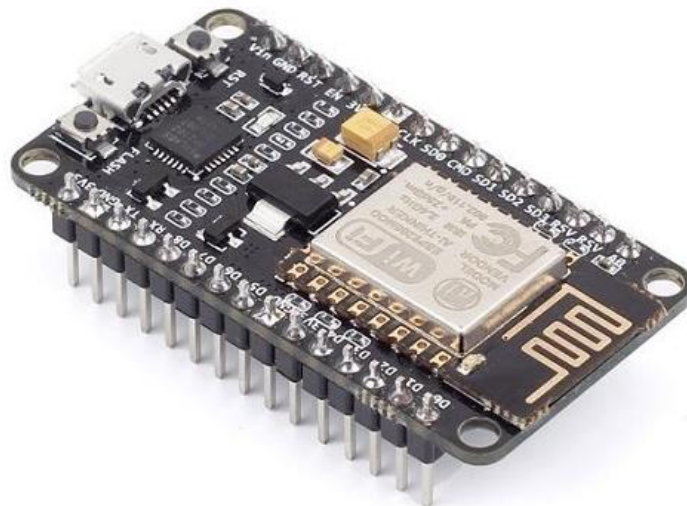


Рисунок 1.5 – ESP8266

ESP8266 – це мікроконтролер із вбудованим модулем Wi-Fi, завдяки якому він бездротово підключається до мереж для обміну інформацією через

Інтернет. Він побудований на базі Tensilica L106, 32-бітного процесора, який може працювати на частотах до 160 МГц, хоча стандартна частота для роботи переважно становить 80 МГц. Цей процесор має достатню продуктивність для більшості завдань, що зустрічаються в IoT-проектах, таких як передача даних через Інтернет або взаємодія з простими датчиками та виконавчими механізмами. Мікроконтролер оснащений 512 КБ флеш-пам'яті, яка використовується для зберігання програм і даних, та 160 КБ SRAM для тимчасового зберігання даних під час виконання програми.

ESP8266 має 17 цифрових контактів вводу/виводу (рис. 1.6), що дозволяє підключати різноманітні компоненти, такі як світлодіоди, реле, датчики та двигуни. Мікроконтролер має 1 аналоговий вхід з діапазоном від 0 до 1 В, що робить його придатним для використання з датчиками, які генерують аналогові сигнали. Підтримка інтерфейсів UART, SPI та I2C дозволяє підключати до ESP8266 широкий спектр пристроїв та взаємодіяти з ними за допомогою стандартних протоколів обміну даними. Вбудований модуль Wi-Fi підтримує стандарти 802.11 b/g/n, що дозволяє підключатися до більшості бездротових мереж та створювати мережі Wi-Fi-клієнтів або точок доступу Wi-Fi.

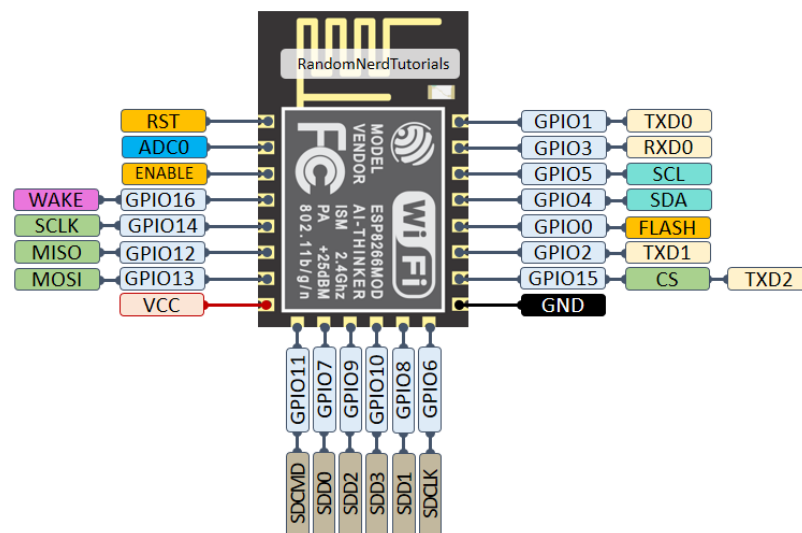


Рисунок 1.6 – Контакти ESP8266

Щодо живлення, ESP8266 розрахований на роботу від 3,3 В. Це важлива вимога, оскільки багато стандартних плат Arduino працюють від 5 В, а підключення компонентів, що використовують 5 В до ESP8266, може вимагати додаткових елементів для зниження напруги до 3,3 В. ESP8266 споживає до 200 мА струму в активному режимі роботи, тому його потрібно живити через стабільне джерело 3.3 В.

Мікроконтролер оснащений потужним вбудованим модулем Wi-Fi, що робить його оптимальним для реалізації ідей в Інтернеті речей (IoT). Тут необхідно забезпечити підключення пристроїв до мережі Інтернет без додаткових бездротових модулів. Завдяки можливості підключення до мережі Wi-Fi, можливо дистанційно керувати пристроями або збирати дані. Оскільки ESP8266 вирізняється привабливою вартістю, він здобув популярність серед любителів та розробників, які хочуть створювати бюджетні бездротові рішення. Таким чином, ESP8266 є вдалим поєднанням продуктивності, доступності та простоти інтеграції з іншими пристроями. Це робить його чудовим варіантом для проектів, які потребують підключення до Wi-Fi та компактних розмірів.

Однією з ключових переваг ESP8266 є його вбудована підтримка Wi-Fi, що дозволяє легко підключати пристрої до Інтернету та реалізовувати IoT-проекти. Завдяки цьому мікроконтролер здобув широку популярність у розробці різних пристроїв, які можуть передавати інформацію через Інтернет або отримувати команди дистанційно. Це робить ESP8266 ідеальним рішенням для «розумних» будинків, систем спостереження та інших проектів, що потребують бездротового з'єднання.

Ще однією суттєвою перевагою є низьке енергоспоживання, що дозволяє використовувати ESP8266 у портативних або автономних проектах, де важлива тривала автономна робота. Наприклад, цей мікроконтролер можна використовувати в бездротових датчиках або пристроях, що працюють від батарейок з обмеженим джерелом живлення.

Доступність є значною перевагою ESP8266, оскільки цей

мікроконтролер значно дешевший за багато інших подібних пристроїв з функціоналом Wi-Fi, що сприяє його популярності серед ентузіастів та любителів. Ще одним важливим плюсом є підтримка Arduino IDE, яка значно полегшує програмування мікроконтролера та дозволяє швидко розпочати розробку. Ви можете використовувати те саме середовище програмування, яке використовуєте для інших плат Arduino, що забезпечує зручність та швидкість налаштування.

Незважаючи на численні позитивні сторони, ESP8266 має деякі недоліки. Найсуттєвішим з них є обмежена пам'ять. Пристрій має лише 512 КБ флеш-пам'яті та 160 КБ SRAM, чого може бути недостатньо для масштабних та складних проектів, що потребують обробки великих обсягів інформації або зберігання громіздких програм.

Варто також враховувати, що ESP8266 працює на напрузі 3,3 В, на відміну від 5 В, що використовуються багатьма платами Arduino, тому потрібно бути обережним під час підключення компонентів. Це означає, що деякі елементи, розраховані на 5 В, можуть бути несумісні з ESP8266 без використання додаткових перетворювачів напруги.

Обмежена кількість контактів на ESP8266 робить його невідповідним вибором для проектів, що потребують великої кількості компонентів або датчиків. Однак для простих завдань, таких як керування світлодіодами через Wi-Fi, цього більш ніж достатньо.

1.2.4 STM32103

STM32F103 (рис. 1.7) – це мікроконтролер із серії STM32 від STMicroelectronics. Він належить до потужного сімейства мікроконтролерів, заснованих на архітектурі ARM Cortex-M3. Мікроконтролери STM32F103 пропонують вражаючу продуктивність та адаптивність, що робить їх добре підходящими для різноманітних завдань у багатьох галузях, включаючи

промислові системи, автоматизацію, робототехніку та побутову електроніку. Мікроконтролер STM32F103 характеризується потужними обчислювальними можливостями, які досягаються завдяки 32-бітній архітектурі ARM Cortex-M3. Це дозволяє йому виконувати операції набагато швидше, ніж багатьом мікроконтролерам з 8-бітною або 16-бітною архітектурою. Цей пристрій також відрізняється енергоефективністю, що робить його придатним для проектів, де потрібна висока обчислювальна потужність з обмеженим джерелом живлення.

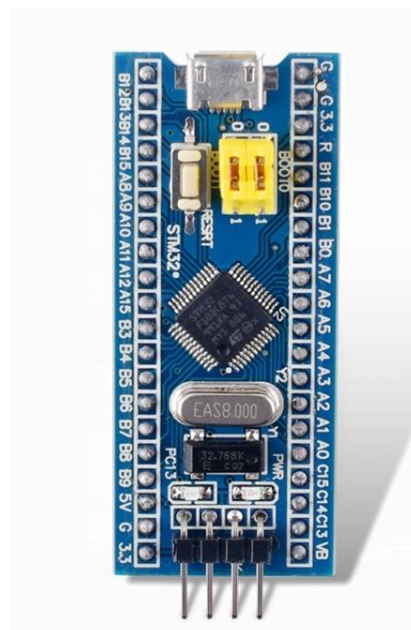


Рисунок 1.7 – STM32103

STM32F103 – це мікроконтролер на базі ARM Cortex-M3, який забезпечує обчислювальну потужність завдяки своїй 32-бітній архітектурі. Цей процесор може працювати на частотах до 72 МГц, що гарантує швидке та ефективне виконання завдань середньої складності. Пам'ять мікроконтролера включає 32 КБ флеш-пам'яті для заданого коду, 20 КБ SRAM для тимчасового зберігання інформації під час обробки та 1 КБ EEPROM для зберігання постійних даних. Такий обсяг пам'яті дозволяє використовувати його в

проектах з високими вимогами до ресурсів, наприклад, в системах автоматизації або робототехніці.

Мікроконтролер оснащений до 37 цифрових виводів, які можна використовувати для введення та виведення. Деякі з цих виводів реалізують функцію ШІМ (широтно-імпульсної модуляції), що відкриває можливості для керування такими пристроями, як світлодіоди або двигуни. Крім того, плата має до 16 аналогових входів, що дозволяють підключати різні датчики з аналоговими сигналами, наприклад, датчики температури або освітлення. Мікроконтролер працює на напрузі від 2,0 В до 3,6 В, що забезпечує його сумісність з багатьма компонентами та датчиками, розрахованими на такий діапазон напруги.

STM32F103 має кілька варіантів підключення до зовнішніх пристроїв за допомогою UART (послідовний порт), I2C, SPI та CAN. Це забезпечує легкий зв'язок між мікроконтролером та різними периферійними пристроями, такими як клапани, датчики, реле та навіть інші мікроконтролери. Крім того, він підтримує JTAG для налагодження та програмування, що значно полегшує розробку великих та складних систем. Мікроконтролер характеризується високою швидкістю обробки даних завдяки тактовій частоті до 72 МГц, що робить його оптимальним вибором для проектів, які потребують швидкої обробки інформації. Крім того, він оснащений широким набором функцій переривання з можливістю підтримки до 16 пріоритетних каналів, що дозволяє ефективно обробляти синхронні події в режимі реального часу.

Таким чином, STM32F103 – це потужний та універсальний мікроконтролер, який знаходить застосування в розробці складних електронних систем, що потребують значної обробки даних та взаємодії з різними периферійними пристроями.

1.2.5 Порівняння та остаточний вибір мікроконтролера

Arduino Nano, Arduino Uno, STM32F103 та ESP8266 – це мікроконтролери, кожен зі своїми перевагами. Arduino Nano – це зменшена версія Arduino Uno з подібними характеристиками, але в меншому форм-факторі. Arduino Uno – один з найпопулярніших мікроконтролерів, має достатньо контактів для простих проектів, а також підтримується в популярному середовищі програмування Arduino IDE. STM32F103 оснащений потужним 32-бітним процесором, пропонує вищу продуктивність та розширену кількість контактів, але складніший у програмуванні та вимагає додаткових налаштувань. ESP8266 має вбудовану підтримку Wi-Fi, що робить його чудовим вибором для проектів Інтернету речей, хоча він працює від 3,3 В та має менше контактів.

Було обрано Arduino Uno, враховуючи його простоту, доступність, достатню кількість входів/виходів для керування світлодіодами, а також чудову підтримку спільноти. Він також чудово підходить для початківців та проектів середнього рівня, таких як цей.

1.3 Постановка задачі

Завданням даної кваліфікаційної роботи є створення функціональної системи керування світлодіодною стрічкою за допомогою мікроконтролера Arduino. Метою є реалізація системи, яка дозволить ефективно та зручно керувати освітленням, адаптуючи його до різних потреб та оточення. Система може бути корисною як для домашнього, так і для офісного освітлення, а також для досягнення різноманітних світлових ефектів.

Основні етапи роботи включають в себе:

- розробка концепції та проектування структури мікроконтролерної системи, що відповідає освітленню, сюди входить вибір компонентів, необхідних для роботи, таких як світлодіодна стрічка, кнопка та резистор;

- реалізація апаратної частини системи: підключення світлодіодної стрічки, кнопки регулювання ефектів та інших компонентів до мікроконтролера Arduino;
- розробка програмного забезпечення для мікроконтролера Arduino, воно повинно забезпечувати автоматизоване управління освітленням відповідно до режимів роботи;
- аналіз готової системи, виявлення та усунення будь-яких недоліків та помилок.

2 АПАРАТНА РЕАЛІЗАЦІЯ ПРОЕКТУ

2.1 Вибір складових частин для пристрою керування освітленням

Важливим кроком є визначення апаратних компонентів, які забезпечать безперебійну, продуктивну та надійну роботу пристрою. Архітектура системи включає основний мікроконтролер, виконавчі та сигнальні компоненти, елементи керування, а також додаткові модулі, необхідні для живлення та зв'язку. В якості блоку управління був обраний мікроконтролер Arduino Uno, оскільки він поєднує в собі достатню продуктивність, простоту програмування і широкий спектр цифрових і аналогових портів. Ця плата забезпечує легке підключення до інших компонентів системи, підтримує велику кількість бібліотек для роботи з різними модулями та має достатній обсяг пам'яті для реалізації необхідної логіки управління.

Для створення візуального світлового ефекту ми обрали адресну світлодіодну стрічку WS2812B. Цей тип світлодіодів дозволяє керувати кожним світловим елементом окремо, задаючи для кожного певний колір і яскравість. Це дає гнучкість у створенні динамічних світлових сцен та ефектів. Підключення здійснюється через цифровий вивід мікроконтролера з використанням спеціалізованих бібліотек. Для забезпечення взаємодії користувача з пристроєм використовується кнопка перемикання режимів освітлення. Її підключення здійснюється через цифровий вхід, а обробка натискань реалізована програмно з використанням захисту від дребезгу контактів. Вся система живиться від стабілізованого джерела 5 В, яке підключено до відповідного входу мікроконтролера. Світлодіодна стрічка, враховуючи її довжину, може потребувати зовнішнього джерела живлення з достатнім струмом, тому можна розділити живлення мікроконтролера і елементів освітлення загальним проводом заземлення.

На рис. 2.1 наведено основну схему пристрою.

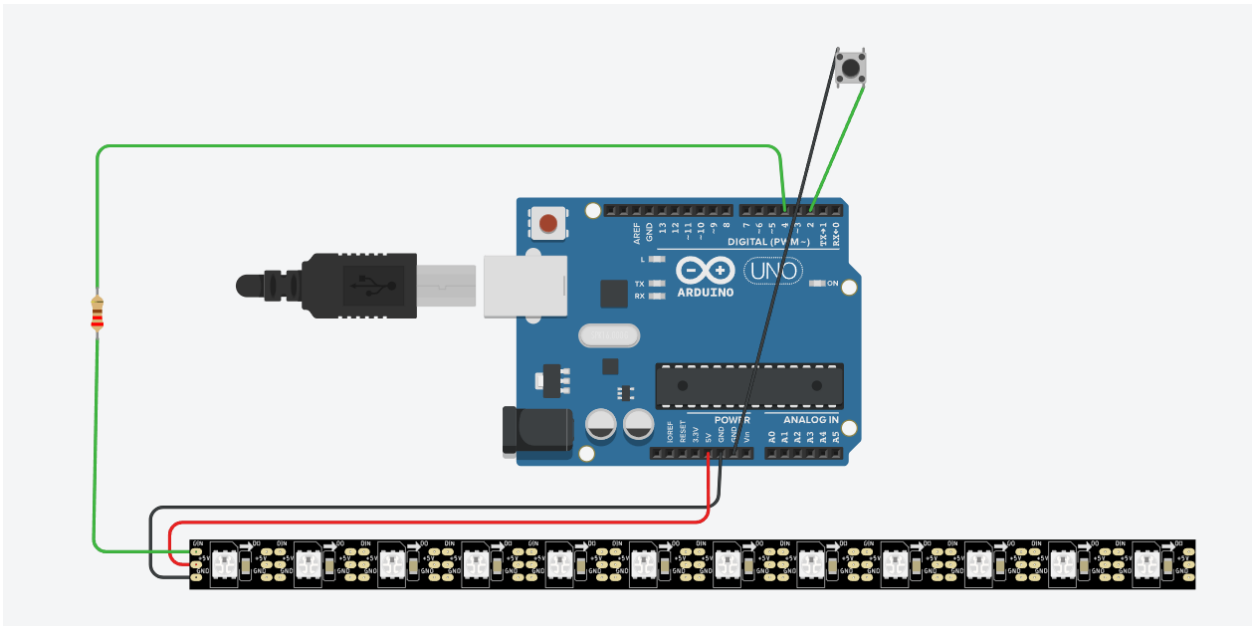


Рисунок 2.1 – Схема пристрою для керування лампою зі світлодіодною стрічкою на основі Arduino UNO

2.1.1 Світлодіодна стрічка

Світлодіодна стрічка WS2812B (рис. 2.2) складається з ланцюжка RGB-світлодіодів, кожен з яких має вбудований драйвер WS2811. Така архітектура дозволяє незалежно регулювати колір і яскравість кожного світлодіода, що відкриває безліч можливостей для створення рухомих світлових ефектів - від рівномірного підсвічування до складної анімації.

Кожен світлодіод на стрічці отримує цифровий сигнал, обробляючи лише призначену йому частину, а решту надсилає далі по ланцюжку. Таким чином реалізується послідовний метод передачі даних, який дозволяє керувати великою кількістю світлодіодів, використовуючи лише один цифровий вивід мікроконтролера. Це значно спрощує монтаж і зменшує кількість з'єднань.



Рисунок 2.2 – Світлодіодна стрічка WS2812B

Світлодіодна стрічка має три ключові контакти:

- VDD (5V) - живлення;
- GND - загальний провід (заземлення);
- DIN - вхід для цифрового сигналу управління.

Схема підключення контактів до мікроконтролера наведено на рис. 2.3.

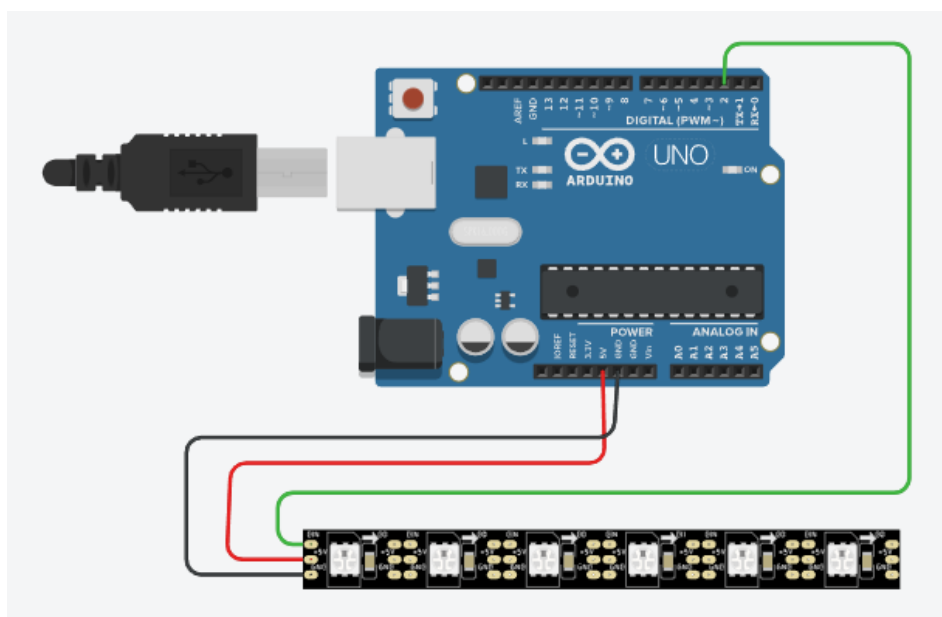


Рисунок 2.3 – Схема підключення світлодіодної стрічки до мікроконтролера

Для керування стрічкою використовується будь-який цифровий вихід мікроконтролера, наприклад, вивід D4. Лінія DIN підключається до резистора, VDD - до виходу 5В, а GND - до загальної землі (GND). При необхідності можна послідовно з'єднати кілька світлодіодних секцій, підключивши вихід DOUT попередньої секції до входу DIN наступної.

Світлодіодна стрічка є зручним, ефективним і універсальним рішенням для створення декоративного або функціонального освітлення, гарантуючи при цьому високий рівень адаптивності та інтерактивності з мінімальними вимогами до проводки.

2.1.2 Резистор

Розроблена система керування освітленням включає в себе резистор (рис. 2.4), інтегрований в сигнальну лінію. Він розташований між мікроконтролером Arduino Uno та DIN-входом першого сегмента світлодіодної стрічки. Його ключова функція - електротехнічне пом'якшення фронтів сигналів і зниження рівня високочастотних перешкод, які можуть виникати під час передачі цифрових даних на стрічку.



Рисунок 2.4 - Резистор

Оскільки в проекті використовується адресна світлодіодна стрічка типу WS2812B, яка керується цифровим виводом, критично важливо забезпечити надійну і чітку передачу даних, особливо для першого світлодіода. З цією метою встановлюється резистор 220 Ом між виводом плати Arduino і входом DIN-стрічки. Це значення є оптимальним для мінімізації електромагнітних завад без суттєвого впливу на швидкість та якість сигналу. Фізичне підключення резистора реалізовано послідовно в сигнальній лінії: один його вивід підключено до цифрового виходу на Arduino, а інший - до DIN-входу стрічки. Приклад наведено на рис. 2.5.

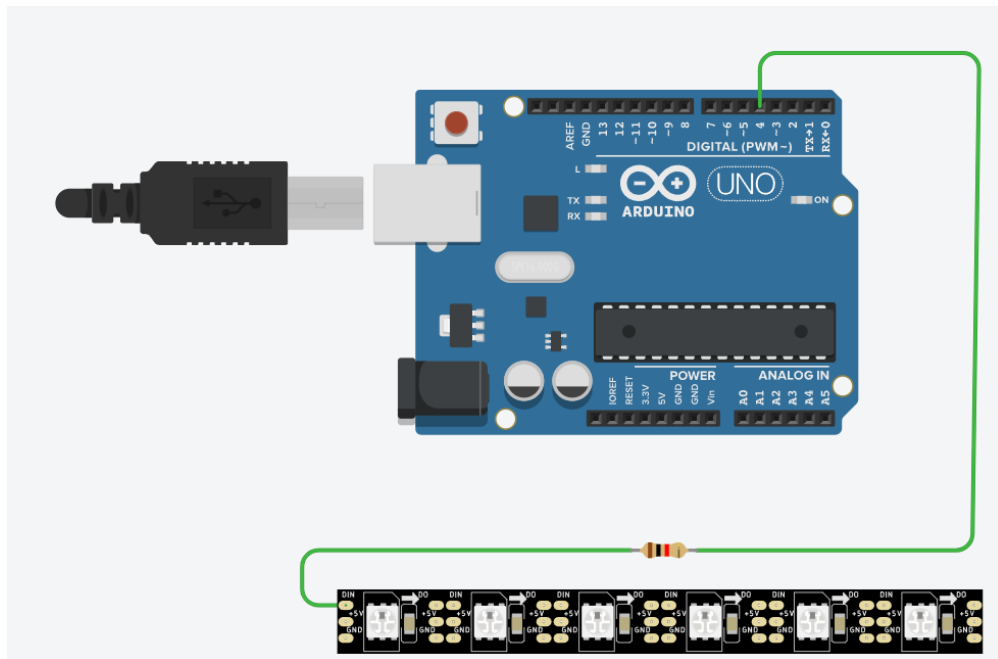


Рисунок 2.5 – Підключення резистора в сигнальній лінії

Такий спосіб реалізації не потребує внесення змін до програмного коду і простий у реалізації, при цьому підвищує надійність системи, особливо при довжині дроту більше 10 см або при наявності потенційних джерел завад у навколишньому середовищі. Тому інтеграція резистора в схему є ключовим

компонентом конструкторського рішення, що підвищує стабільність роботи пристрою, гарантує захист першого світлодіода від можливих перенапруг сигнальної лінії, а також сприяє зниженню ймовірності збоїв в роботі стрічки.

2.1.3 Кнопка

Кнопка (рис. 2.6) - це простий електромеханічний перемикач, призначений для короткочасного замикання або розмикання електричного кола при натисканні. У системі керування вона виконує роль вхідного елемента, який дозволяє користувачеві викликати певну дію або змінити режим роботи пристрою. Факт натискання кнопки визначається за зміною логічного рівня на вході мікроконтролера, який зчитує її стан у цифровому вигляді.



Рисунок 2.6 – Кнопка

Типова кнопка має чотири контакти (рис. 2.7), які утворюють дві пари (1a-1b і 2a-2b). Ці пари електрично з'єднані по діагоналі посередині і замикаються при натисканні. Тільки одна пара контактів використовується для цифрового з'єднання: один з них підключений до цифрового входу мікроконтролера (наприклад, D2), а інший - до загального проводу (GND). Для забезпечення стабільного логічного рівня в Arduino використовується внутрішній підтягуючий резистор, який активується режимом INPUT_PULLUP, що дозволяє обійтися без зовнішніх деталей.

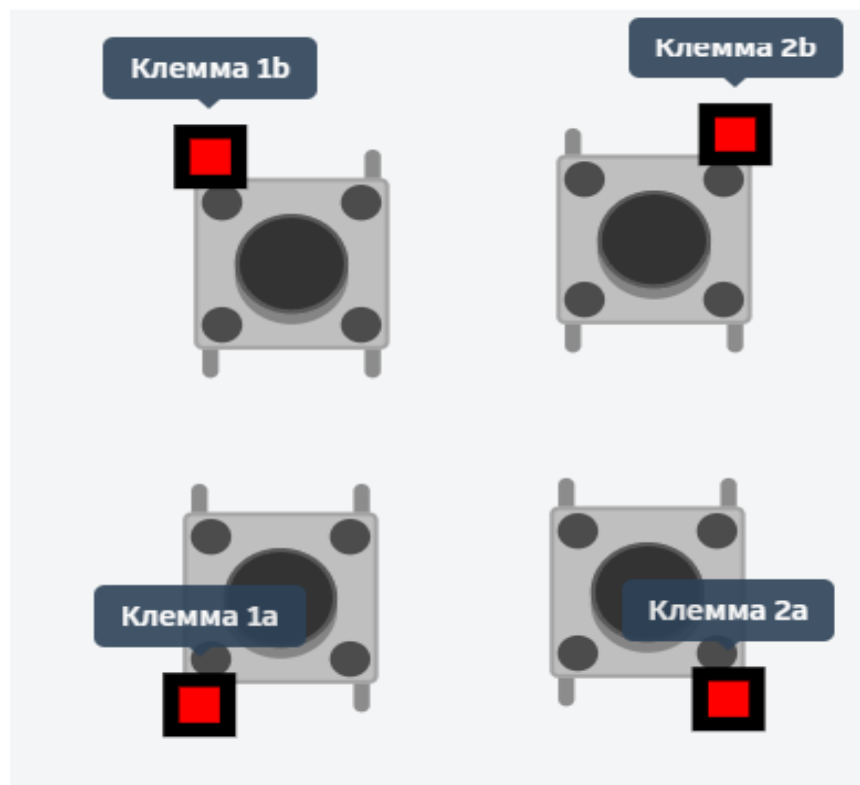


Рисунок 2.7 – Кнопка з парами контактів (1a-1b, 1b-2b)

Кнопка, що використовується в конструкції, є тактильним елементом з нормально відкритим контактом, який замикається тільки при натисканні. Вона має чотири контакти, робочу напругу до 12 В і витримує струм до 50 мА, що повністю відповідає вимогам цифрових входів Arduino. Розміри корпусу зазвичай становлять 6×6 мм або 12×12 мм, залежно від типу, і монтується

вона за допомогою наскрізних отворів (ТНТ), що дозволяє легко використовувати кнопку на макетних або друкованих платах. Її робота заснована на принципі миттєвого замикання: контакт замикається при натисканні і автоматично розмикається при відпусканні. Це робить кнопку зручною для перемикання режимів, активації функцій та налаштування параметрів у мікроконтролерних системах.

3 ВИБІР ТА ПОРІВНЯННЯ ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

3.1 Вибір середовища моделювання

Вибір середовища моделювання є критично важливим кроком у розробці мікроконтролерних систем, оскільки дозволяє перевірити функціональність схеми та програмного коду без фізичної збірки пристрою. Це допомагає виявити та усунути помилки на початкових етапах проекту, що в кінцевому підсумку економить час і ресурси. Використання симуляторів дозволяє вивчити поведінку мікроконтролера, різних модулів і датчиків в різних умовах, тим самим підвищуючи надійність кінцевого продукту. Відповідний вибір середовища моделювання забезпечує ефективну реалізацію ідеї проекту, гарантуючи точність і комфорт розробки.

3.1.1 Wokwi Arduino Simulator

Wokwi - це сучасне онлайн-середовище, яке дозволяє моделювати проекти Arduino, дозволяючи створювати електронні схеми, кодувати та негайно тестувати свою роботу. Для розробки систем, подібних до тієї, що використовується в нашому дипломному проекті, Wokwi має ряд ключових переваг: доступність без необхідності встановлення додаткового програмного забезпечення, підтримка базових мікроконтролерів Arduino (в тому числі Arduino Uno, що використовується в нашій схемі), можливість імітації кнопок, світлодіодів, резисторів та підтримка бібліотек, необхідних для світлодіодних стрічок, таких як Adafruit NeoPixel. Інтеграція з GitHub та можливість створювати онлайн-посилання для швидкого доступу до проектів також є значною перевагою.

З іншого боку, слід зазначити і обмеження цього середовища: Wokwi - це лише симулятор цифрової логіки і не підходить для моделювання

аналогових схем або складних електронних систем. Деякі специфічні компоненти, які можуть знадобитися для просунутих проєктів, відсутні в бібліотеці платформи. Крім того, Wokwi не передбачає проєктування друкованих плат, що є необхідним етапом у створенні повноцінного апаратного рішення. В контексті нашої роботи Wokwi можна розглядати як корисний інструмент для перевірки базової логіки коду та тестування алгоритмів роботи мікроконтролера, але його функціоналу недостатньо для повноцінної розробки проєкту, адже в нашому випадку необхідно також враховувати особливості підключення апаратних компонентів та тестування їх фізичної взаємодії.

3.1.2 Fritzing

Fritzing - популярне середовище проєктування електричних схем, яке дозволяє демонструвати з'єднання компонентів, малювати електричні схеми і навіть планувати розміщення на друкованих платах. Для проєктів, подібних до нашого, Fritzing є зручним інструментом для створення візуального представлення взаємодії таких компонентів, як Arduino Uno, світлодіодна стрічка WS2812B, кнопка перемикачів режимів та резистор у сигнальному ланцюзі. Програма дозволяє розташовувати компоненти на макетній платі, створювати зрозумілі схеми підключення та експортувати креслення, які можна використовувати для пояснення роботи пристрою або його монтажу.

Серед переваг Fritzing - зручний інтерфейс, можливість проєктувати схеми з реалістичним виглядом компонентів, а також функції для підготовки до виробництва друкованих плат. Варто підкреслити, що Fritzing дозволяє візуально перевірити правильність з'єднань між мікроконтролером, світлодіодною стрічкою та іншими елементами, важливими для фізичної збірки пристрою. Водночас, слід зазначити, що Fritzing не є симулятором - він не дозволяє запускати програмний код або імітувати динаміку роботи системи. Також програма має деякі недоліки з бібліотекою компонентів, а деякі модулі

можна показати лише у вигляді спрощених символів. Що стосується нашої дипломної роботи, то Fritzing можна використовувати для розробки зрозумілих схем з'єднань, що дозволяє наочно пояснити, як функціонує пристрій і допомагає планувати розташування компонентів. Однак для перевірки програмного коду та логіки роботи системи необхідно використовувати інші програми моделювання, оскільки Fritzing не передбачає програмної симуляції.

3.1.3 Tinkercad

TinkerCAD — це безкоштовна онлайн-платформа від Autodesk, яка пропонує моделювання схем і периферійних пристроїв Arduino у вкладці «Схеми» без необхідності купувати фізичне обладнання. У цьому середовищі користувач перетягує компоненти (Arduino Uno, кнопки, резистори, світлодіоди, стрічки NeoPixel тощо) на віртуальну макетну плату, з'єднує їх дротами і відразу активує симуляцію, спостерігаючи за рухом електричних сигналів між вузлами. Водночас код (C/C++ або візуальна мова блоків) визначає роботу пристрою. Основними перевагами TinkerCAD є миттєвий запуск без необхідності встановлення додаткового програмного забезпечення, інтуїтивний інтерфейс перетягування, підтримка емуляції стандартних бібліотек Arduino (зокрема, Adafruit_NeoPixel), вбудований послідовний монітор для налагодження та можливість ділитися макетами за допомогою посилань. Це дозволяє швидко перевірити логіку пристрою, уникнути помилок підключення та отримати знімок екрана світлодіодної стрічки, не витрачаючи гроші на фізичні компоненти.

3.1.4 Порівняння та остаточний вибір середовища моделювання

Після огляду трьох популярних інструментів для моделювання проектів Arduino — Tinkercad, Fritzing і Wokwi — відразу стало зрозуміло, що Fritzing

зручний для малювання схем і макетів друкованих плат, але не дозволяє тестувати код і служить лише для створення візуального представлення для документації. Wokwi дозволяє налагоджувати код за допомогою Serial Monitor і підтримує різні плати (Uno, Nano, ESP32), але підключення компонентів там здійснюється за допомогою налаштувань JSON, що менш інтуїтивно зрозуміло, а візуальне представлення макетної плати помітно поступається Tinkercad. Tinkercad Circuits, з іншого боку, поєднує в собі веб-доступність, тобто працює без встановлення додаткового програмного забезпечення, інтерфейс макетної плати з функцією перетягування, повну емуляцію коду Arduino з підтримкою бібліотеки Adafruit_NeoPixel і вбудований послідовний монітор. Це дозволяє миттєво бачити реакцію світлодіодної стрічки на програмний код, перевіряти правильність підключень і логіку кнопки, а також легко ділитися проектом за допомогою посилання. Саме завдяки вдалому поєднанню простоти, зрозумілості та повної функціональності для тестування роботи світлодіодної стрічки Tinkercad був обраний основним середовищем моделювання для нашого пристрою.

3.2 Вибір мови програмування

Вибір мови програмування для Tinkercad Circuits є ключовим моментом при роботі над нашим проектом Arduino, оскільки він визначає, наскільки зручно буде створювати алгоритми і як швидко ми зможемо налаштувати управління RGB-стрічкою. У цьому середовищі у нас є два варіанти: класичний текстовий C++ і візуальна блокова мова BlocklyDuino. Деякі розробники швидше освоюють базову логіку за допомогою блокового інтерфейсу, а інші віддають перевагу точному контролю над синтаксисом C++. З огляду на це, важливо розглянути обидва підходи: оцінити, наскільки вони відповідають вимогам нашого проекту, і вирішити, який з них забезпечить оптимальне поєднання простоти кодування та можливості точного налаштування світлових ефектів.

3.2.1 C++

У Tinkercad Circuits вкладка «код» відкриває звичне текстове середовище Arduino, де програмний код пишеться на C++. Це дозволяє працювати з тими ж інструментами та бібліотеками, що і на реальній платі: ви пишете код у звичному синтаксисі, додаєте `#include` для необхідних бібліотек (наприклад, `Adafruit_NeoPixel`) і відразу бачите результат виконання в симуляторі. Перевага використання C++ полягає в повному контролі над усіма аспектами: ви можете вручну налаштовувати таймери, обробляти переривання, оптимізувати використання пам'яті та безпосередньо керувати кожним контактом Arduino. Крім того, при виникненні помилок компілятор виводить докладні повідомлення, що полегшує налагодження. Що стосується керування RGB-стрічками, C++ дозволяє гнучко створювати складні світлові ефекти, точно регулювати затримки між оновленнями та детально взаємодіяти з кожним пікселем. Тому, якщо вам потрібен найнадійніший і найрозширюваніший підхід для реалізації всіх потреб вашого проекту, ми рекомендуємо розглянути текстове програмування на C++ в Tinkercad Circuits.

3.2.2 BlocklyDuino

Tinkercad Circuits також має візуальну мову блоків під назвою BlocklyDuino. Програма створюється шляхом перетягування готових блоків (наприклад, «цикл», «умова», «функція») замість написання коду на C++. Такий підхід полегшує освоєння основ алгоритмізації, оскільки користувач може візуально бачити структуру коду і швидко створювати базові програми, не вивчаючи синтаксис. Використовуючи BlocklyDuino для управління RGB-стрічкою, можна, наприклад, використовувати блоки «встановити колір світлодіода» або «зачекати N мілісекунд» для реалізації простих ефектів без ручного підключення бібліотек.

Водночас цей метод може мати обмеження для більш складних налаштувань або оптимізації. Не всі функції бібліотеки та детальні параметри можуть бути представлені блоками, тому для складних анімацій або нестандартних схем доведеться перейти на C++. Однак для швидкого створення прототипів базової логіки управління освітленням та початкового тестування алгоритмів використання BlocklyDuino є практичним і продуктивним рішенням.

3.2.3 Остаточний вибір мови програмування

Для реалізації нашої ідеї управління RGB-стрічкою на Arduino в Tinkercad Circuits ми вибрали мову програмування C++. Хоча BlocklyDuino пропонує швидкий спосіб створення прототипів базових логічних структур, текстове програмування на C++ відкриває широкий спектр можливостей, включаючи повний доступ до всіх бібліотек і функціональних можливостей Arduino. Це є ключовим фактором у створенні складних і точно налаштованих світлових ефектів, реалізації обробки натискання кнопок з дебаунсом і забезпеченні оптимальної продуктивності. C++ забезпечує необхідний рівень контролю над усіма аспектами функціональності світлодіодної стрічки, гарантуючи, що код працює в реальному апаратному забезпеченні так само, як і в симуляції.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА АНАЛІЗ ПРОЕКТУ

4.1 Бібліотека Adafruit_NeoPixel

Для реалізації адресованого керування світлодіодною стрічкою WS2812 ми вибрали бібліотеку Adafruit_NeoPixel, оскільки вона забезпечує зручний високорівневий інтерфейс до протоколу WS2812, автоматично розраховує необхідні часові параметри та дозволяє нам зосередитися на логіці світлових ефектів без зайвих низькорівневих налаштувань. Після підключення бібліотеки до ескізу створюється об'єкт класу Adafruit_NeoPixel pixels, в якому вказується кількість діодів, номер виводу для передачі даних і параметри формату.

Функція setup() викликає pixels.begin(), яка ініціалізує внутрішні буфери бібліотеки та встановлює апаратний вивід у режим передачі даних. Відразу після цього смуга очищується — всі світлодіоди встановлюються в чорний колір — і вперше викликається pixels.show(), щоб відобразити початковий стан (вимкнений). Генератор випадкових чисел, який необхідний для випадкових кольірних режимів, також ініціалізується в setup():

Лістинг 4.1 – Функція setup()

```
void setup() {  
  pixels.begin();  
  clearPixels();  
  pixels.show();  
  randomSeed(analogRead(0));  
}
```

Щоб оновити колір окремого діода, було використано метод pixels.setPixelColor(index, pixels.Color(r, g, b)), який приймає індекс і три 8-бітні компоненти кольору. Щоб зміни набули чинності, потрібно викликати

`pixels.show()`. Якщо потрібно вимкнути всю смугу одночасно, можна використовувати вбудовану функцію `pixels.clear()` перед `pixels.show()` або визначити власну утилітарну функцію.

Для створення плавного спектрального переходу («Веселка») використовується допоміжна функція `Wheel()`, яка повертає інтерпольований клас RGB на основі вхідного значення від 0 до 255:

Лістинг 4.2 – Функція `Wheel()`

```
uint32_t Wheel(byte pos) {
  if (pos < 85) {
    return pixels.Color(pos * 3, 255 - pos * 3, 0);
  } else if (pos < 170) {
    pos -= 85;
    return pixels.Color(255 - pos * 3, 0, pos * 3);
  } else {
    pos -= 170;
    return pixels.Color(0, pos * 3, 255 - pos * 3);
  }
}
```

Бібліотека `Adafruit_NeoPixel` усуває необхідність ручного керування синхронізацією сигналів WS2812, надає зручні методи адресації кожного світлодіода та масового оновлення, а також підтримує різні порядки каналів (RGB, GRB тощо). Це дозволяє зосередитися на розробці самих візуальних ефектів, не турбуючись про низькорівневі деталі протоколу.

4.2 Опис програмної реалізації

Програмна частина проекту організована у вигляді послідовності семи світлових ефектів, між якими користувач може перемикатися одним натисканням кнопки. Після останнього ефекту лічильник режимів повертається до першого, забезпечуючи безперервний цикл. Реалізовано наступні режими – «Випадкові кольори», «Єдиний колір», «Заповнення», «Театральна гонка», «Іскри», «Рухома точка» та «Веселка».

Режим «Випадкові кольори» полягає в тому, що кожен світлодіод на стрічці отримує свій унікальний набір випадкових значень для червоного, зеленого та синього каналів, заповнюючи всю стрічку хаотичною мозаїкою відтінків. У режимі «Один колір» всі світлодіоди одночасно світяться одним і тим же випадковим кольором, створюючи єдину суцільну пляму світла. Режим «Заповнення» імітує хвилю, коли стрічка послідовно заповнюється вибраним кольором — спочатку світиться перший світлодіод, потім другий і так далі, поки не заповниться вся стрічка. У режимі «Театральний пробіг» кожен третій світлодіод вмикається і вимикається у фіксованому порядку вздовж стрічки, створюючи враження рухомої смуги світла. Режим «Іскри» реалізує короткі «помиговіння»: випадково вибрані світлодіоди короткочасно світяться яскравим відтінком, а потім гаснуть, як мерехтливі іскри. У режимі «Рухома точка» один яскравий світлодіод рухається вперед і назад між кінцями стрічки, змінюючи колір при кожному відскоку, імітуючи скануючий промінь. Нарешті, режим «Rainbow» створює плавний, безперервно змінюваний градієнт, в якому кольори спектру поступово проходять по всій довжині смуги, забезпечуючи безперервне райдужне мерехтіння.

Для запуску обладнання в `setup()` відбувається:

- ініціалізація бібліотеки `Adafruit_NeoPixel` (`pixels.begin()`);
- встановлення піна кнопки в режимі;
- вимкнення всіх діодів (`clearPixels()`; `pixels.show()`);
- підготовка генератора випадкових чисел.

У головному циклі `loop()` виконується опитування за допомогою функції `buttonPressed()`, яка виявляє перехід від HIGH до LOW і затримує 50 мс, щоб усунути відскок. Якщо виявляється натискання, змінна `mode` збільшується на одиницю (або скидається до нуля, якщо вона була на максимумі), після чого за допомогою `switch(mode)` викликається відповідна функція ефекту і відбувається пауза 200 мс для комфортної швидкості анімації.

Лістинг 4.3 демонструє режим «Іскри», який реалізовано у вигляді серії швидких «підморгувань»: десять разів випадково обирається індекс

світлодіода, йому призначається яскравий випадковий колір, через 75 мс піксель вимикається.

Лістинг 4.3 - Функція для режиму «Іскри»

```
void sparkle() {
  for (int i = 0; i < 10; i++) {
    int idx = random(NUMPIXELS);
    int r = random(0,255), g = random(0,255), b = random(0,255);
    pixels.setPixelColor(idx, pixels.Color(r, g, b));
    pixels.show();
    delay(75);
    pixels.setPixelColor(idx, pixels.Color(0, 0, 0));
  }
}
```

Лістинг 4.4 показує код для ефекту «Рухома точка», у якому створюється один яскравий маркер, який рухається між краями стрічки вперед і назад; при кожному «відштовхуванні» від кінця напрямок інвертується, а точка набуває випадкового кольору. Вигляд рухомої точки представлено на рисунку 4.1.

Лістинг 4.4 - Функція для режиму «Рухома точка»

```
void bounce() {
  static int j = 0;
  static int dir = 1;
  static int r = random(0, 255), g = random(0, 255), b =
random(0, 255);
  clearPixels();
  pixels.setPixelColor(j, pixels.Color(r, g, b));
  pixels.show();
  delay(100);
  j += dir;
  if (j <= 0 || j >= NUMPIXELS - 1) {
    dir = -dir;
    r = random(0, 255);
    g = random(0, 255);
    b = random(0, 255);
  }
}
```

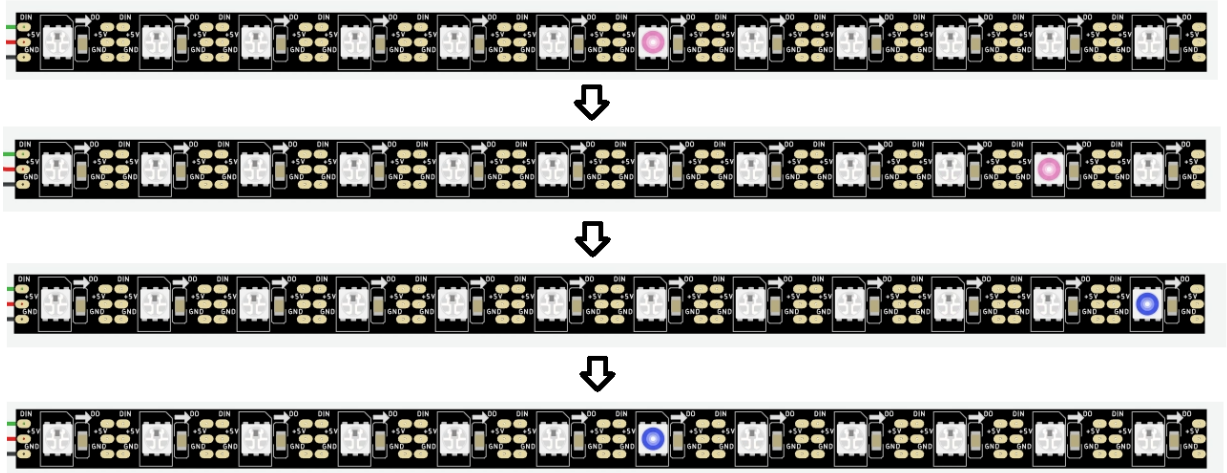


Рисунок 4.1 – Демонстрація ефекту «Рухома точка»

Для режиму «Веселка» використовується допоміжна функція `Wheel()`, що перетворює значення $0 \dots 255$ у колірний градієнт. Основна логіка наведена у лістингу 4.5.

Лістинг 4.5 - Реалізація ефекту «Веселка»

```
void rainbowEffect() {
    static uint16_t j = 0;
    for (int i = 0; i < NUMPIXELS; i++) {
        pixels.setPixelColor(i, Wheel((i + j) & 255));
    }
    pixels.show();
    j++;
}
//Допоміжна функція Wheel:
uint32_t Wheel(byte WheelPos) {
    if (WheelPos < 85) {
        return pixels.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
    } else if (WheelPos < 170) {
        WheelPos -= 85;
        return pixels.Color(255 - WheelPos * 3, 0, WheelPos * 3);
    } else {
        WheelPos -= 170;
        return pixels.Color(0, WheelPos * 3, 255 - WheelPos * 3);
    }
}
```

Лістинг 4.6 демонструє реалізацію режиму «Випадкові кольори», в якому кожен світлодіод отримує свій власний набір випадкових значень

каналів R, G, B. Через це вся стрічка заповнюється красивим різноманіттям кольорів. Рисунок 4.2 ілюструє результат виконання цього режиму.

Лістинг 4.6 – Реалізація ефекту «Випадкові кольори»

```
void randomEachPixel() {
  for (int i = 0; i < NUMPIXELS; i++) {
    pixels.setPixelColor(i, pixels.Color(
      random(0,255), random(0,255), random(0,255)
    ));
  }
  pixels.show();
}
```

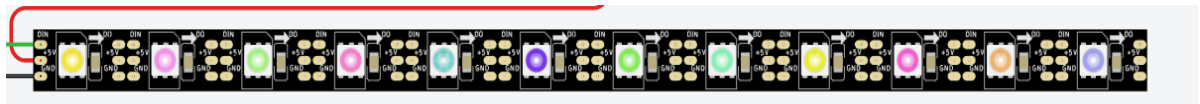


Рисунок 4.2 – Демонстрація ефекту «Випадкові кольори»

Лістинг 4.7 показує режим «Єдиний колір», коли одне випадкове значення RGB, яке було згенеровано на початку застосовується до всіх світлодіодів одночасно, тим самим створюється однокольорова стрічка світла. На рисунку 4.3 видно як виглядає суцільна заливка.

Лістинг 4.7 – Реалізація ефекту «Єдиний колір»

```
void randomOneColor() {
  redColor = random(0, 255);
  greenColor = random(0, 255);
  blueColor = random(0, 255);
  for (int i = 0; i < NUMPIXELS; i++) {
    pixels.setPixelColor(i, pixels.Color(redColor, greenColor,
    blueColor));
  }
  pixels.show();
}
```

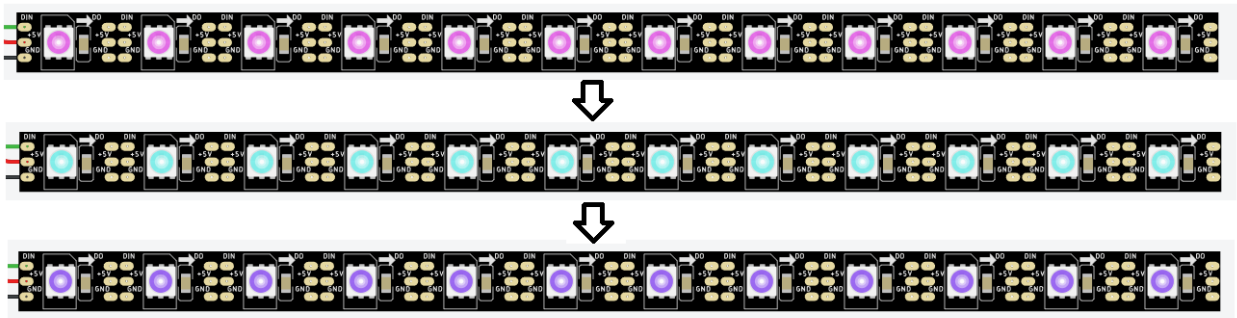


Рисунок 4.3 – Демонстрація ефекту «Єдиний колір»

Лістинг 4.8 демонструє ефект «Заповнення»: стрічка плавно заповнюється випадковим кольором зліва направо ніби кольорова хвиля. Ілюстрація цього процесу на рисунку 4.4.

Лістинг 4.8 – Реалізація ефекту «Заповнення»

```
void colorWipe() {
  redColor  = random(0, 255);
  greenColor = random(0, 255);
  blueColor  = random(0, 255);
  for (int i = 0; i < NUMPIXELS; i++) {
    pixels.setPixelColor(i, pixels.Color(redColor, greenColor,
    blueColor));
    pixels.show();
    delay(50);
  }
}
```

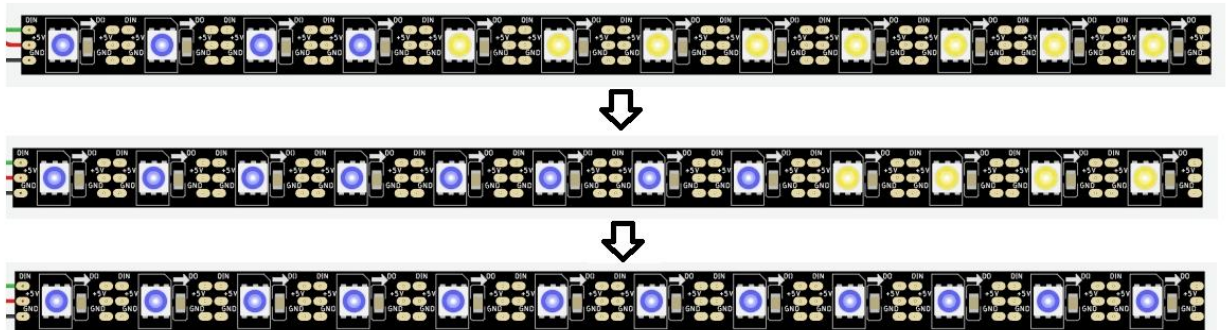


Рисунок 4.4 – Демонстрація ефекту «Заповнення»

Лістинг 4.9 показує режим «Театральна гонка», у якому по черзі мигає кожен третій світлодіод, тим самими створюючи рухому лінію світла уздовж стрічки. Демонстрацію цього режиму можна побачити на рисунку 4.5.

Лістинг 4.9 – Реалізація ефекту «Театральна гонка»

```
void theaterChase() {
  redColor  = random(0, 255);
  greenColor = random(0, 255);
  blueColor  = random(0, 255);
  for (int a = 0; a < 10; a++) {
    for (int b = 0; b < 3; b++) {
      clearPixels();
      for (int i = b; i < NUMPIXELS; i += 3) {
        pixels.setPixelColor(i, pixels.Color(redColor,
greenColor, blueColor));
      }
      pixels.show();
      delay(100);
    }
  }
}
```

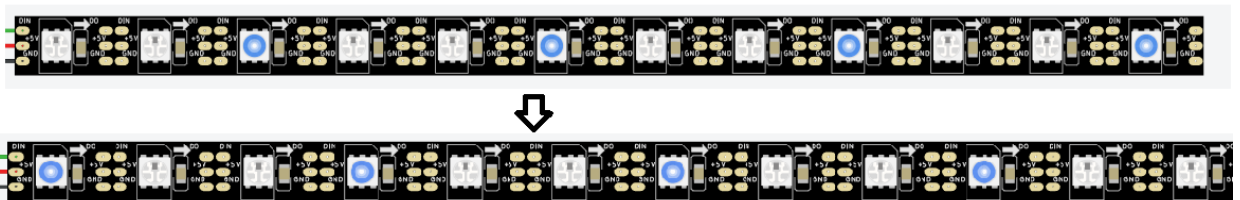


Рисунок 4.5 – Демонстрація ефекту «Театральна гонка»

Ключовий фрагмент перемикання режимів у `loop()` наведений у лістингу 4.10.

Лістинг 4.10 - Логіка циклічного перемикання ефектів

```
void loop() {
  if (buttonPressed()) {
    if (mode == -1) mode = 0;
    else mode = (mode + 1) % NUM_MODES;
  }
}
```

```

}
if (mode == -1) return;

switch (mode) {
  case 0: randomEachPixel(); break;
  case 1: randomOneColor(); break;
  case 2: colorWipe(); break;
  case 3: theaterChase(); break;
  case 4: sparkle(); break;
  case 5: bounce(); break;
  case 6: rainbowEffect(); break;
}
delay(200);
}

```

Конфігураційні параметри, які визначають кількість пікселів, пін кнопки, кількість режимів та час антидребезгу задані за допомогою констант:

Лістинг 4.7 - Константи налаштувань

```

#define PIN 4 // Пін керування світлодіодами
#define NUMPIXELS 12 // Кількість світлодіодів у
стрічці
#define BUTTON_PIN 2
const int NUM_MODES = 7;
unsigned long debounceDelay = 50;

```

Повний текст програми наведено у додатку Б.

4.3 Аналіз проекту

При аналізі створеного пристрою ми розглянули апаратне та програмне забезпечення, розроблене для управління світлодіодним освітленням на базі мікроконтролера Arduino. Результатом стала повністю функціональна прототипна модель, здатна відтворювати сім різних візуальних ефектів за допомогою адресованих світлодіодних стрічок WS2812. Перемикання між режимами здійснюється натисканням однієї кнопки, що забезпечує простоту використання та мінімалістичний дизайн. Такий підхід дозволяє створити ефективну, функціональну і водночас економічну систему декоративного або

атмосферного освітлення.

До переваг розробленої системи належать її проста конструкція та використання невеликої кількості компонентів, що знижує виробничі витрати та полегшує збірку. Програмне забезпечення має модульну структуру, що дозволяє легко змінювати ефекти або додавати нові. Ще однією важливою перевагою є низьке енергоспоживання системи при використанні стрічки з помірною кількістю світлодіодів. Всі ефекти візуально привабливі, і рішення може бути адаптоване для різних застосувань: настільні лампи, освітлення меблів, освітлення дитячої кімнати, арт-інсталяції або акцентне освітлення в інтер'єрах.

Однак конструкція має певні недоліки. Наприклад, відсутність збереження останнього вибраного режиму після вимкнення живлення означає, що кожного разу, коли лампа вмикається, вона запускається в початковому стані. Крім того, використання лише однієї кнопки обмежує зручність управління, особливо при розширенні функцій або ефектів. Система також не передбачає регулювання яскравості, що може бути важливим вночі або для створення м'якого освітлення. Ще одним обмеженням є масштабованість – із збільшенням довжини світлодіодної стрічки зростає потреба в більш потужному джерелі живлення і, можливо, в оптимізації коду для стабільної роботи. Незважаючи на це, проект має значний потенціал для вдосконалення. У подальшому розвитку можна реалізувати збереження останнього режиму в пам'яті контролера, наприклад, за допомогою EEPROM. Для більшої зручності управління можна реалізувати сенсорну панель або пульт дистанційного керування, наприклад, на основі Bluetooth або інфрачервоного передавача. Також доцільно додати регулювання яскравості або адаптивне освітлення, яке змінюється залежно від рівня навколишнього освітлення. У майбутньому можна буде мініатюризувати пристрій, перейшовши на менші мікроконтролери (наприклад, ATtiny або ESP), що дозволить вбудувати систему безпосередньо в корпуси ламп.

Таким чином, запропонований пристрій є повністю функціональним,

простим у реалізації та адаптованим до нових функціональних можливостей. Він може використовуватися не тільки як навчальний або демонстраційний проект, але й масштабуватися для домашнього або комерційного використання, зокрема в галузі внутрішнього освітлення, розумних будинків та декоративного дизайну.

ВИСНОВКИ

В результаті кваліфікаційної роботи було розроблено та реалізовано мікроконтролерну систему динамічного керування світлодіодним освітленням на базі Arduino. Актуальність проекту зумовлена зростаючим попитом на адаптивні освітлювальні рішення, що поєднують енергоефективність, декоративність та простоту керування. Створена система здатна візуалізувати сім різних світлових ефектів, які можуть бути використані в світильниках, декоративних елементах інтер'єру, інтерактивних інсталяціях або в складі «розумного будинку».

Під час роботи було проведено аналіз технічних можливостей доступних контролерів та засобів управління адресованими світлодіодами. За результатами аналізу в якості основної платформи управління було обрано Arduino Uno, оскільки вона відповідає критеріям доступності, функціональності, простоти інтеграції та підтримує велику кількість бібліотек, включаючи Adafruit_NeoPixel, що значно спрощує реалізацію ефектів. Була розроблена та протестована апаратна частина системи, що включає мікроконтролер, кнопку управління та адресовану світлодіодну стрічку WS2812. Програмне забезпечення реалізовано у вигляді модульного коду з окремими функціями для кожного ефекту та логікою перемикання режимів.

Успішно реалізований прототип продемонстрував життєздатність концепції: всі режими візуально відображаються відповідно до очікуваних результатів, управління інтуїтивно зрозуміле, а сама система має потенціал для подальшого вдосконалення. Зокрема, можна додати нові режими, регулювання яскравості, сенсорне або бездротове управління, а також адаптувати проект до більш компактних платформ. Запропоноване рішення є прикладом ефективної реалізації керованого декоративного освітлення з низьким споживанням ресурсів, яке може бути використане як у повсякденному житті, так і в дизайнерських або комерційних об'єктах.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Теслюк, В. М. Проектування систем Інтернету речей на основі AVR- та ESP-контролерів [Текст] : навч. посіб. / В. М. Теслюк, С. І. Ковалюк. – Львів : Вид-во Львівської політехніки, 2024. – 152 с. – ISBN 978-966-941-968-2.
2. Кемерон, Н. Проекти на ESP8266 та ESP32. Практичний підхід до створення IoT-систем [Текст] / Н. Кемерон. – К. : Діалектика, 2023. – 312 с. – ISBN 978-617-7315-57-3.
3. Шварц, М. Інтернет речей за допомогою ESP8266. Основи, приклади та підключення до хмарних сервісів [Текст] / М. Шварц. – К. : Наш Формат, 2022. – 256 с.
4. Елементи інформатики [Текст] : довідник / В. С. Височанський, А. І. Кардаш, В. С. Костев, В. В. Черняхівський. – К. : Наук. думка, 2003. – 192 с.
5. Adafruit. NeoPixel Überguide [Електронний ресурс] : технічна документація / Adafruit Industries. – Режим доступу: <https://learn.adafruit.com/adafruit-neopixel-uberguide/overview>.
6. SparkFun. WS2812 Breakout Hookup Guide [Електронний ресурс] : технічний посібник / SparkFun Electronics. – Режим доступу: <https://learn.sparkfun.com/tutorials/ws2812-breakout-hookup-guide>.
7. Arduino. Arduino Uno Technical Specifications [Електронний ресурс] / Arduino.cc. – Режим доступу: <https://docs.arduino.cc/hardware/uno-rev3>.
8. MakerGuides. ESP8266 Pinout Reference and How to Use GPIO Pins [Електронний ресурс] / MakerGuides. – Режим доступу: <https://www.makerguides.com/esp8266-pinout/>.
9. STM32F103 Reference Manual [Електронний ресурс] : technical reference / STMicroelectronics. – Режим доступу: https://www.st.com/resource/en/reference_manual/CD00171190.pdf.
10. Instructables. NeoPixel Effects with Arduino [Електронний ресурс] /

Instructables.com. – Режим доступа: <https://www.instructables.com/id/Arduino-WS2812B-LED-Strip/>.

11. DIYIOT. ESP8266 NodeMCU vs Arduino UNO: which is better? [Электронный ресурс] / DIYIOT.com. – Режим доступа: <https://diyi0t.com/esp8266-vs-arduino-uno/>

12. Wokwi. Arduino Simulator – Wokwi Projects [Электронный ресурс] / Wokwi.com. – Режим доступа: <https://wokwi.com>.