

Міністерство освіти і науки України  
Черкаський національний університет імені Богдана Хмельницького  
Факультет обчислювальної техніки, інтелектуальних та управляючих систем  
Кафедра програмного забезпечення автоматизованих систем

## КУРСОВА РОБОТА

з дисципліни «Комп'ютерна схемотехніка»  
на тему: «Фітолампа з управлінням за допомогою android додатку»

студента другого курсу групи КТ-22  
спеціальності 123 Комп'ютерна інженерія  
Мачульський Андрій Олегович

(прізвище та ініціали)

Керівник: Авраменко. А. С.

(прізвище та ініціали)

Оцінка за шкалою:

\_\_\_\_\_  
(національною, кількість балів, ECTS)

Члени комісії:

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(прізвище та ініціали)

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(прізвище та ініціали)

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(прізвище та ініціали)

Черкаси – 2023

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**  
**ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ім. Б. ХМЕЛЬНИЦЬКОГО**  
**Кафедра інформаційних технологій**

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
Веретельник В.В.

(підпис)  
„\_\_\_\_” „\_\_\_\_” 2023 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ**  
**на курсову роботу**

студента(ки) \_\_\_\_\_ Мачульського Андрія Олеговича групи КТ-22 другого курсу \_\_\_\_\_

**ТЕМА** \_\_\_\_\_ Фітолампа з управлінням за допомогою android додатку \_\_\_\_\_

**Вихідні дані до курсової роботи:** \_\_\_\_\_

Зміст **Пояснювальної записки** до курсової роботи:

Індивідуальне завдання

Вступ

1 Аналіз

2 Реалізація апаратно/програмної частини

3 Налагодження і тестування. Інструкція користувача.

Висновки

Список літератури

Додатки (за необхідності)

Перелік наочного матеріалу

Апаратно/програмний прототип ..., пояснювальна записка, презентація результатів роботи.

**Календарний план виконання роботи:**

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу: до ...	Примітка
1.	Отримання завдання на курсову роботу.		
2.	Огляд джерел технічної інформації за темою роботи.		
3.	Виконання аналізу методів реалізації завдання		
4.	Реалізація структурної схеми, визначення вимог до апаратного забезпечення, опис будови та роботи обраної електронної схеми		
5.	Програмування алгоритму, створення програмного продукту		
6.	Тестування розробленого програмного продукту		
7.	Написання пояснювальної записки		
8.	Подача роботи керівнику для написання відгуку.		
9.	Корегування роботи за результатами розгляду керівника. Остаточне оформлення пояснювальної роботи.		
10.	Написання доповіді, створення слайдів.		
11.	Захист курсової роботи		

студента(ки) \_\_\_\_\_ /Мачульський.А.О./

(підпис)

Науковий керівник \_\_\_\_\_ /Авраменко.А. С /

(прізвище, ініціали)

(підпис)

(прізвище,  
ініціали)

Завдання видане « \_\_\_\_ » \_\_\_\_\_ 2023 року

# Зміст

<b>Вступ .....</b>	<b>4</b>
<b>Розділ 1 Огляд і аналіз існуючих технічних рішень.....</b>	<b>5</b>
<b>1.1 Означення фітолампи .....</b>	<b>5</b>
<b>1.2 Визначення оптимального технічного рішення .....</b>	<b>6</b>
<b>1.3 Визначення методів організації програми .....</b>	<b>7</b>
<b>Висновок до розділу .....</b>	<b>8</b>
<b>Розділ 2 Реалізація структурної схеми, визначення вимог до апаратного забезпечення .....</b>	<b>9</b>
<b>2.1 Алгоритм роботи фітолампи.....</b>	<b>9</b>
<b>2.2 Визначення апаратного забезпечення .....</b>	<b>10</b>
<b>2.3 Побудова та розбір принципової електричної схеми .....</b>	<b>12</b>
<b>2.4 Визначення вимог до програмного забезпечення .....</b>	<b>14</b>
<b>2.5 Реалізація схеми .....</b>	<b>14</b>
<b>Висновок до розділу .....</b>	<b>18</b>
<b>Розділ 3 Реалізація програмного коду та інструкцій для користувача.....</b>	<b>20</b>
<b>3.1 Реалізація програмного коду для приладу .....</b>	<b>20</b>
<b>3.2 Реалізація програмного коду для Android додатку.....</b>	<b>25</b>
<b>3.3 Тестування схеми і Android- додатка та інструкція для користувача</b>	<b>28</b>
<b>Висновки .....</b>	<b>31</b>
<b>Список використаних джерел.....</b>	<b>32</b>

					ЧНУ.232032.105 ПЗ		
Зм..	Лист	№ документа	Підпис	Дата			
Розробив		Мачульський.А.О			Фітолампа з управлінням за допомогою android додатку	Літера	Аркуш
Перевірив		Авраменко.А. С					Аркушів
Н. Контроль						Кафедра ІТ	
Затвердив						Група КТ-22	

# Вступ

Сучасний розвиток технологій та зростання інтересу до дизайну "розумних" систем спричиняють появу нових інноваційних рішень для оптимізації нашого повсякденного життя. У цьому контексті виникає необхідність в розробці ефективних та екологічно чистих рішень для підтримки життєвого циклу рослин. Відомо, що світло грає важливу роль у рості та розвитку рослин, і сучасні технології використовуються для створення оптимальних умов для їхнього здоров'я.

Актуальність даної розробки обумовлена потребою в створенні інноваційного рішення - фітолампи з інтегрованим андроїд-додатком. Це сполучення технічних можливостей спеціалізованого освітлення та функціональності мобільного додатка дозволить користувачам надавати рослинам оптимальні умови вирощування, враховуючи їхні індивідуальні потреби та фази розвитку. Фітолампа може працювати як самостійний прилад, але для досягнення найкращих результатів вона може бути використана у сполученні з іншими технічними засобами для створення оптимального середовища для росту рослин.

Метою даної курсової роботи є розробка та технічне обґрунтування фітолампи з андроїд-додатком як інтегрованого рішення для управління ростом та розвитком рослин у домашніх умовах. Робота буде фокусуватися на детальному аналізі технічних вимог до пристрою, враховуючи його застосування в області рослинництва. У контексті даного дослідження також розглядаються умови експлуатації та переваги використання даного пристрою для сприяння зростанню та розвитку рослин в домашніх умовах.

# Розділ 1 Огляд і аналіз існуючих технічних рішень

## 1.1 Означення фітолампи

Один із важливих аспектів сучасного рослинництва та вирощування рослин в умовах штучного освітлення — використання фітоламп. Фітолампи являють собою спеціально розроблені світлодіодні або газорозрядні лампи, призначені для стимулювання росту та фотосинтезу рослин.

Однією з важливих характеристик фітоламп є спектр світла, який вони випромінюють. Рослини активно використовують світло в певних діапазонах, зокрема в червоному і синьому спектрах. Фітолампи спеціально створені таким чином, щоб максимально задовольняти потреби рослин у цих ключових спектральних областях.

Інтенсивність світла, яку випромінює фітолампа, також є важливим параметром. Вона вимірюється в мікромолях фотонів на квадратний метр в секунду ( $\mu\text{mol}/\text{m}^2/\text{s}$ ). Інтенсивність повинна бути оптимальною для конкретного типу рослин та їх фази росту.

Однією з основних функцій фітоламп є стимуляція фотосинтезу та зростання рослин. Штучне світло, що імітує природний спектр, дозволяє рослинам здійснювати фотосинтез більш ефективно, що призводить до покращеного зростання і розвитку.

Фітолампи можуть бути налаштовані для випромінювання специфічних спектрів світла в різних фазах росту рослин. Наприклад, використання більше синього світла може стимулювати вегетативний ріст, тоді як червоне світло може бути активоване під час цвітіння та плодоношення.

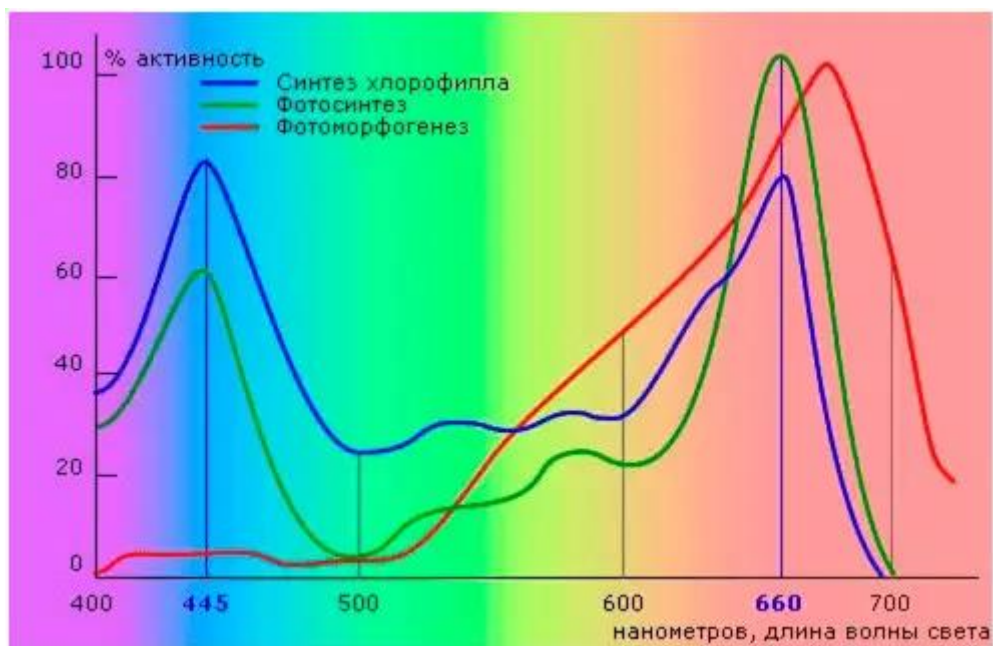


Рисунок 1.1 – Графік світлового потоку для рослин.

## 1.2 Визначення оптимального технічного рішення

Основним елементом фітолампи є світлодіоди які повинні бути керовані. Спочатку ми повинні вибрати мікроконтролер, для функціонування фітолампи, для зміни її кольору роботи, і зміни яскравості, ми можемо використати багато різних мікроконтролерів. Є різні мікроконтролери такі як Raspberry Pi, ESP тощо. Для нашого проекту візьмемо найпоширеніший мікроконтролер який має назву Arduino.

Для освітлення використаємо світлодіоди синього і червоного кольору для емуляції найкращого спектру кольорів для рослин. Для контролю або обмеження напруги, регулювання сили струму та розподілу напруги використаємо резистори які будуть підключені до світлодіодів.

В ході проектування було розглянуто декілька варіантів керування лампою. Так як в самому завданні вказано, що керування повинно відбуватися за допомогою Android додатка. Варіанти були такі:

1. За допомогою підключення до WIFI мережі
2. За допомогою підключення до Bluetooth модуля

І в цих варіантах були свої недоліки, наприклад при підключенні до WIFI мережі у користувача може і не бути самої мережі в наслідок цього ,керування лампою є неможливим, а при підключенні до Bluetooth у користувача можуть виникнути проблеми з самим смартфоном наприклад: несправний сам смартфон, або батарея смартфона розрядилась. Слідуючи з цього було прийнято рішення до використання інфрачервоного- датчика на самому смартфоні, так як для керування не потребується доступ до мережі, і якщо смартфон в не робочому стані або зі розрядженою батареєю користувач завжди може скористатися ІЧ – пультом який буде в комплекті.

### 1.3 Визначення методів організації програми

В нашому проекті фітолампа повинна мати керування за допомогою Android додатка. Спочатку ми розберемо принцип запрограмування самого мікроконтролер Arduino. Наша фітолампа повинна мати функції вмикання і вимикання, повинна мати декілька режимів яскравості , і мати певні режими освітленості ,а саме червоний, синій та фіолетовий (утворений при освітленні червоного і синього ) для певних етапів росту рослин. Середовище в якому ми запрограмуємо наш мікроконтролер (Arduino IDE) вимагає мову C++ яка є одна із найпоширеніших мов в світі. C++ використовується для програмування на платформі Arduino, яка є відкритою апаратно-програмною платформою для створення пристроїв з мікроконтролерами. Основна мова програмування для Arduino - це мова C++, з деякими додатковими бібліотеками та фреймворками, які спрощують взаємодію з апаратним забезпеченням.

Для Android додатка ми повинні організувати керування нашою лампою за допомогою ІЧ- передатчика який знаходиться в багатьох смартфонах. Сам Android додаток буде написаний в середовищі створення Android додатків Android Studio. Мова на якій буде реалізовано Android додаток буде Kotlin. Ця мова розроблена компанією JetBrains, вона призначена для вдосконалення сучасного програмування та інтеграції з існуючим кодом на Java. Kotlin широко використовується для мобільної розробки (зокрема, на платформі Android),

серверного програмування та веб-розробки. Однією з переваг Kotlin є те, що вона дозволяє розробникам писати більш безпечний та ефективний код, прискорюючи процес розробки.

## Висновок до розділу

В данному розділі ми розкрили ключові аспекти використання фітоламп в контексті штучного освітлення для рослин. Описано основні характеристики фітоламп, зокрема їхній спектр світла та інтенсивність, які визначають їхню ефективність у стимулюванні росту та фотосинтезу рослин.

Окрема увага приділяється визначенню оптимального технічного рішення для фітолампи. Обговорено вибір мікроконтролера, як основного елемента для керування функціями лампи. Зазначено, що для проекту використовується Arduino, який надійно відповідає вимогам контролю за кольором та яскравістю фітоламп.

У розділі також висвітлено важливі аспекти вибору методу керування фітолампю з допомогою Android додатка. Розглянуті варіанти через WIFI та Bluetooth мережі, але визначено, що оптимальним рішенням є використання інфрачервоного-датчика на смартфоні для надійного керування, уникнення проблем з мережею або смартфоном.

У розділі детально описано методи організації програми для мікроконтролера та Android додатка. Зазначено, що мова програмування для мікроконтролера - C++, а середовище - Arduino IDE. Для Android додатка використовується Kotlin у середовищі Android Studio. Розглянуті функціональні можливості, такі як вмикання/вимикання, регулювання яскравості та режими освітленості.

Цей розділ надає повну картину базових концепцій та технічних вирішень, необхідних для реалізації фітолампи з керуванням через Android додаток.



## Розділ 2 Реалізація структурної схеми, визначення вимог до апаратного забезпечення

### 2.1 Алгоритм роботи фітолампи

Для основи алгоритму було взято функціонал фітолампи а саме спектр кольорів яку вона повинна генерувати. Цей спектр у нас будуть генерувати світлодіоди червоного і синього кольорів. Користувач може вибрати яскравість лампи та вибрати режим освітлення. Якщо лампа уже не потрібна користувач її просто виключає.

Для кращого розуміння побудуємо блок-схему нашого алгоритму роботи (рис 2.1)

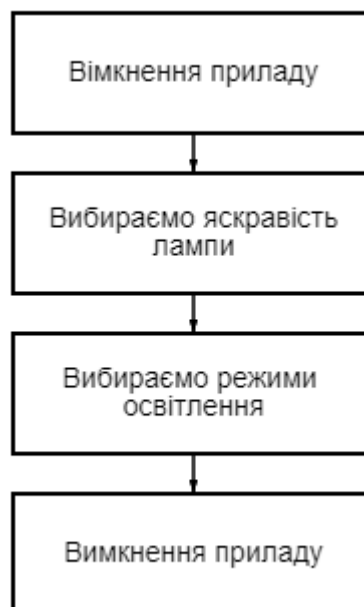


Рисунок 2.1 – Блок-схема алгоритму роботи

## 2.2 Визначення апаратного забезпечення

Для обробки команд користувача і передачі їх до світлодіодів ми візьмемо платформу Arduino а саме найпоширеніший мікроконтролер Arduino UNO R3.

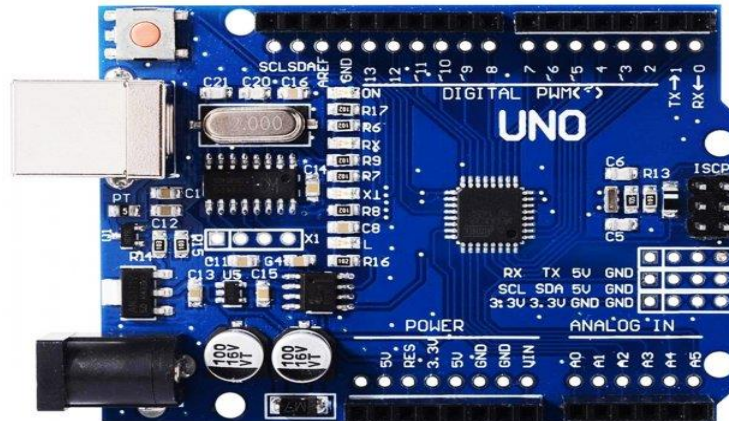


Рисунок 2.2 – мікроконтролер Arduino UNO

Розглянемо його характеристики:

Мікроконтролер: ATmega328

Робоча напруга: 5В

Напруга живлення 7-12В

Цифрові входи: 14

Аналогові входи: 6

Flash-пам'ять: 32кБ

SRAM: 2 КБ

Тактова частота 16МГц

Слідуючи з його характеристик цей мікроконтролер підійде нам найкраще всього для подачі команд.

Далі ми повинні додати самі світлодіоди вони повинні бути червоні і сині для створення потрібного нам спектру кольорів

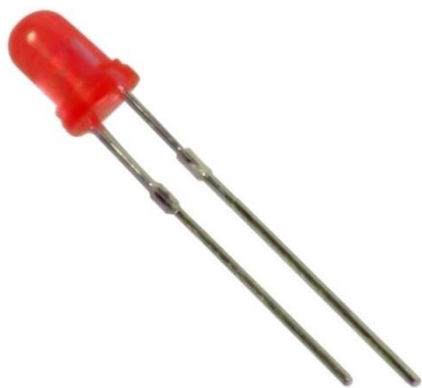


Рисунок 2.3 – червоний світлодіод



Рисунок 2.4 – синій світлодіод

Далі наступним компонентом у нас стане ІЧ приймач, для прийняття команд з пульта або зі смартфона. Для нашої схеми використаємо ІЧ-приймач VS1838B



Рисунок 2.5 – ІЧ-приймач VS1838B

## 2.3 Побудова та розбір принципової електричної схеми

В цьому підрозділі опишемо і розглянемо принципову електричну схему. Розробимо нашу принципову схему за допомогою середовища Tinkercad, в якому є всі потрібні елементи для нашого приладу. Розглянемо нашу схему на (рис 2.6)

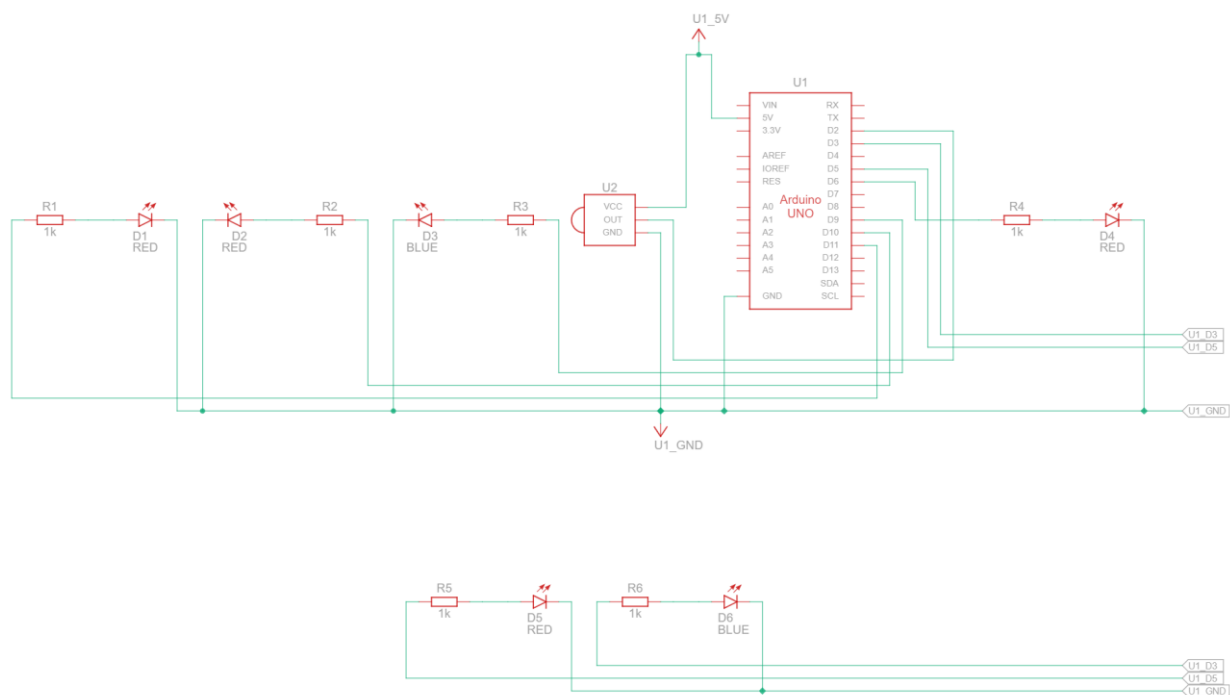


Рисунок 2.6 – Принципова електрична схема

Розглянувши принципову електричну схему проаналізуємо кожну частину цієї схеми. На схемі ми можемо замітити наш ІЧ-приймач який підключається через 3 входи (VCC, OUT, GRD), що являється живленням сигналом і землею відповідно.

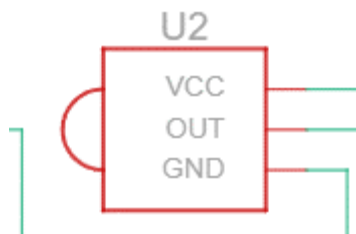


Рисунок 2.7 – ІЧ-приймач

В нас є також сам мікроконтролер до якого під'єднано всі інші електричні елементи

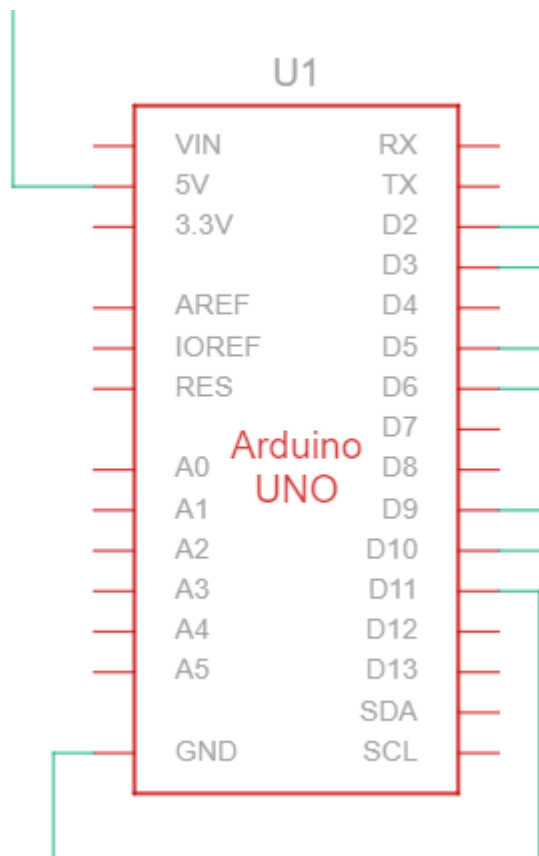


Рисунок 2.8 – Arduino UNO

А також головне в нашій схемі це світлодіоди які під'єднані до землі і до резисторів які саме і регулюють струм які тим часом під'єднанні послідовно до самого мікроконтролера

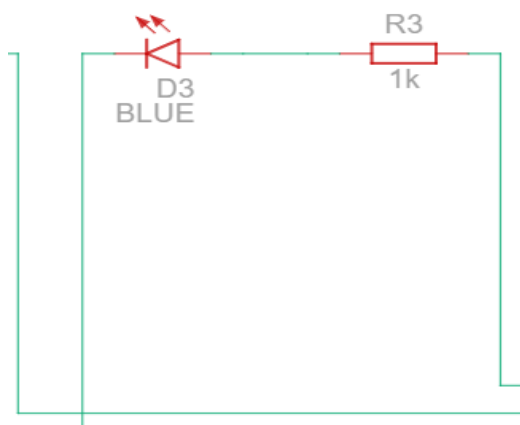


Рисунок 2.8 – підключення світлодіода до резистора

## 2.4 Визначення вимог до програмного забезпечення

Основною метою програмного забезпечення являється керування та подача команд на наші світлодіоди. Наша програма повинна мати функцію включення і виключення ,для встановлення яскравості пристрою має бути функція зменшення або збільшення рівня яскравості, повинна бути реалізована функція встановлення різних режимів освітленості а саме червоний , синій та фіолетовий.

Для Android- додатка ми повинні реалізувати свій аналог фізичного пульта, щоб завдяки смартфону і ІЧ передатчика в ньому ми змогли виконувати керування над нашим приладом. Повинні бути реалізовані всі подачі команд, які перераховані вгорі.

## 2.5 Реалізація схеми

Перед тим як ми нашу схему змайструємо в фізичному виді, побудуємо її в цифровому форматі для аналізу, та вдосконалення. Цей етап проектування дозволяє інженерам визначити ефективність схеми, виявити можливі помилки та оптимізувати її параметри перед фактичною реалізацією в матеріальній формі. Будувати будемо в середовищі Tinkercad, який має всі елементи які потрібні в нашій схемі.

Спочатку покладемо на наш “віртуальний” стіл всі наші потрібні компоненти такі як:

## Мікроконтролер Arduino UNO R3(рис 2.9)

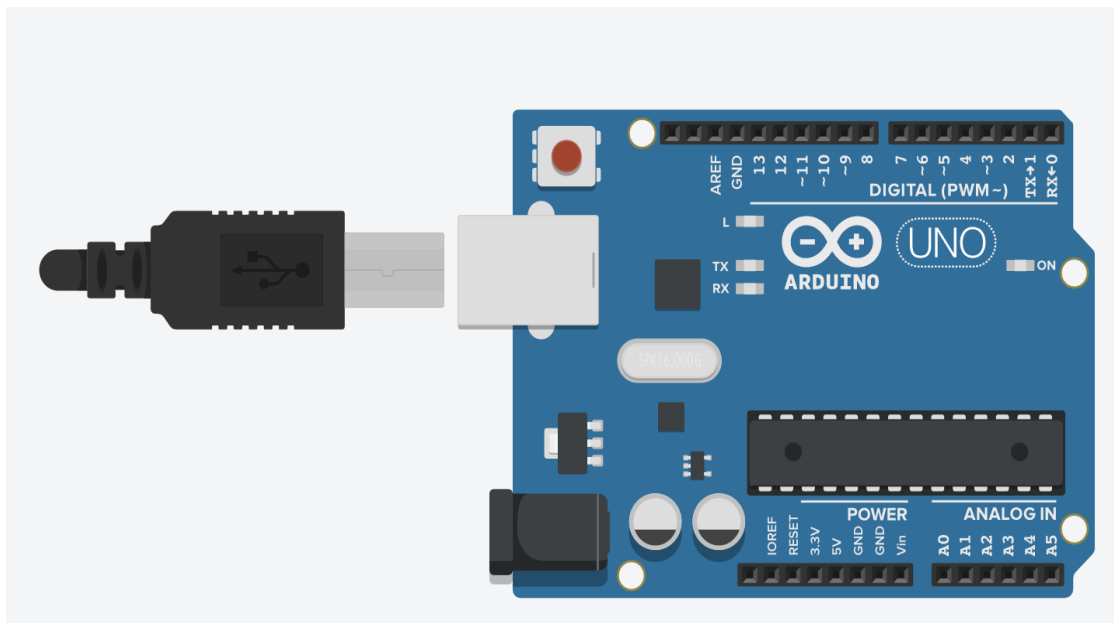


Рисунок 2.9 – мікроконтролер Arduino UNO R3 в середовищі Tinkercad

## 4 світлодіоди червоного кольору і 2 світлодіоди синього кольору(рис 2.10)

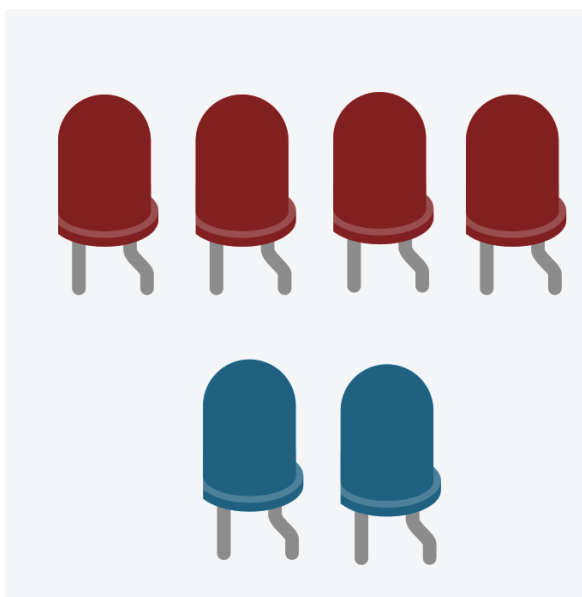


Рисунок 2.10 – Світлодіоди синього та червоного кольору в середовищі Tinkercad.

Сам нашій ІЧ приймач (рис 2.11)

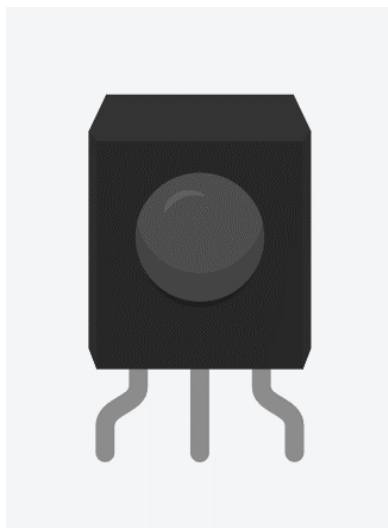


Рисунок 2.11 – ІЧ-приймач в середовищі Tinkercad.

Після чого ми повинні наші елементи паралельно з'єднати між собою, але так як у нас буде велика кількість проводів для зручності з'єднань ми використаємо схему макетування.

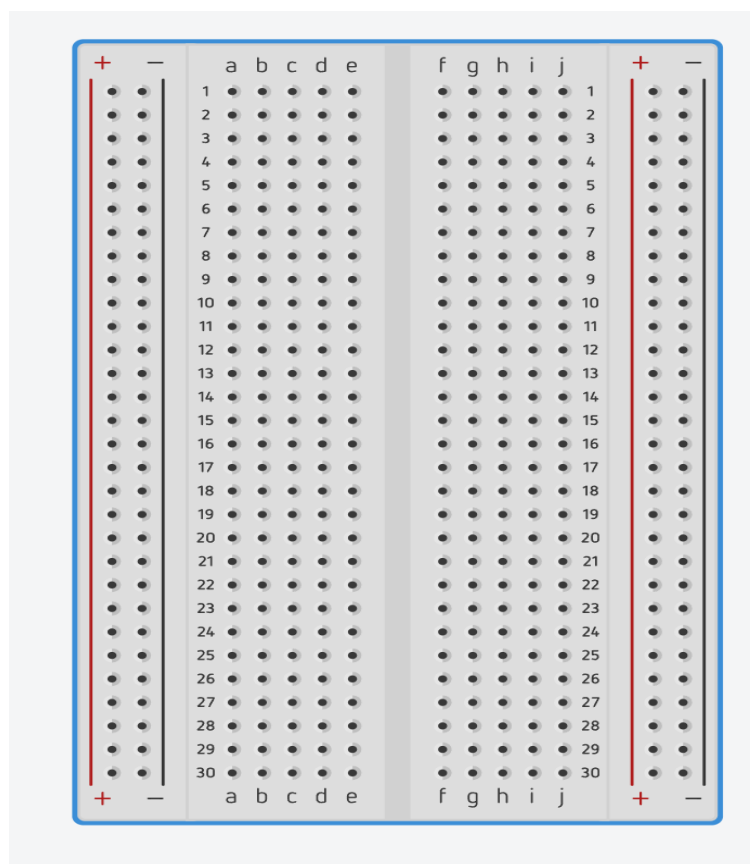


Рисунок 2.12 – схему макетування.



Так як ми маємо всі основні елементи на нашому робочому столі. Залишилось їх тільки з'єднати дивлячись на принципову електричну схему.

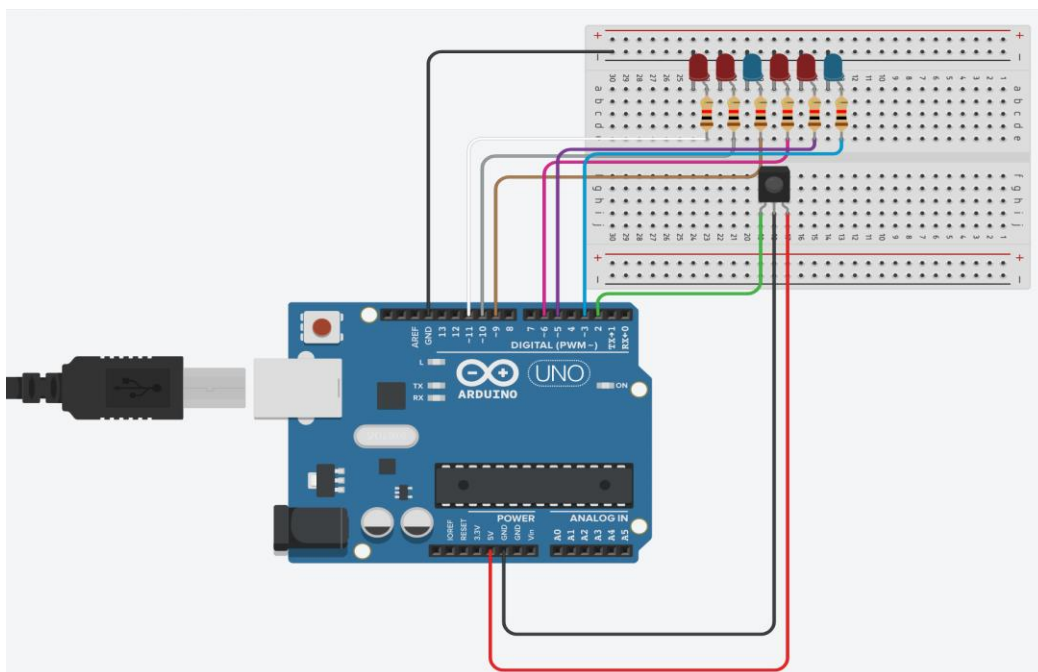


Рисунок 2.12 – схема приладу в середовищі Tinkercad

Ця схема була зібрана з підключеними світлодіодами до відповідних пінів а саме (11, 10, 9, 6, 5, 3) з метою подальшого створення зміни рівня освітленості, так як ці піни являються PWM тобто призначені для генерації аналогового сигналу. Яскравість світлодіода при підключенні до пінів PWM змінюється шляхом регулювання відсотка часу, протягом якого пін виводить високий рівень напруги. При використанні PWM пін генерує швидкість вимикачів (цикли) з фіксованою частотою.

Тепер зібравши нашу схему в цифровому форматі зберемо цю схему в фізичному виді зіставивши всі елементи так само як і на схемі.

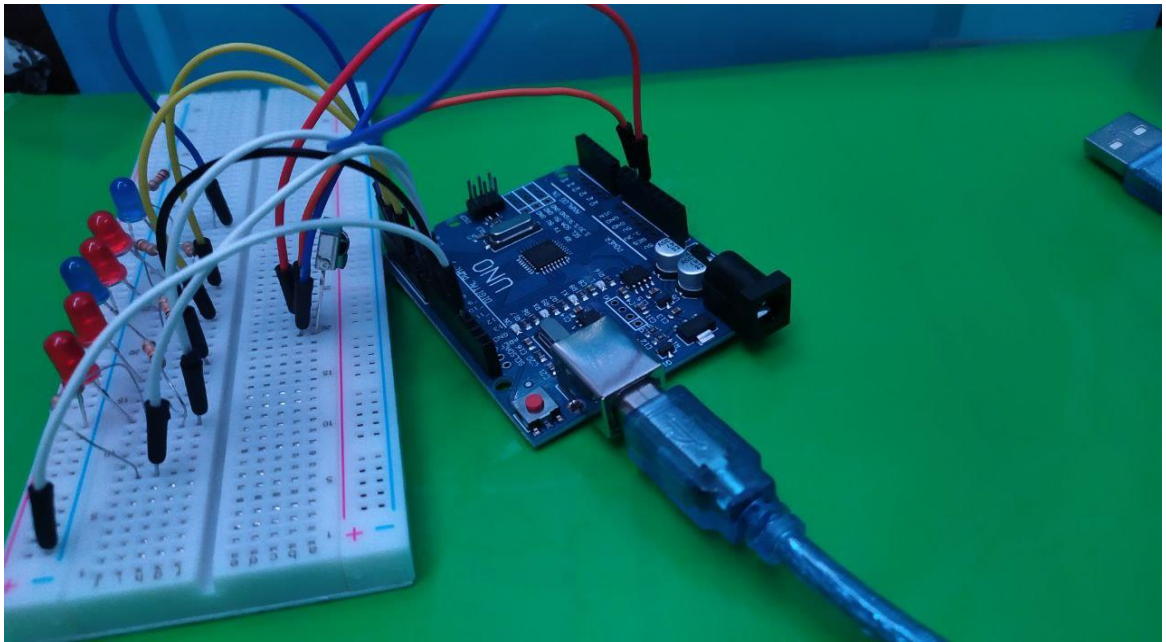


Рисунок 2.13 – прилад в фізичному виді

## Висновок до розділу

У даному розділі курсової роботи було розглянуто та визначено основні етапи реалізації фітолампи з керуванням за допомогою Android-додатка. Розділ розпочався з опису алгоритму роботи фітолампи, в якому було враховано функціонал пристрою, зокрема спектр кольорів, який генерують світлодіоди червоного і синього кольорів. Користувач має можливість вибору яскравості лампи та режиму освітлення.

Було наведено блок-схему алгоритму роботи, яка вказує на послідовність операцій та рішень, які приймаються в процесі роботи фітолампи.

У розділі також проведено визначення апаратного забезпечення для реалізації функцій пристрою. Обрано мікроконтролер Arduino UNO R3, який є поширеним та досить потужним для керування світлодіодами та обробки команд користувача. Також використано світлодіоди червоного і синього кольорів, а також ІЧ-приймач VS1838B для прийому команд з пульта або смартфона.

Було проведено побудову та розбір принципової електричної схеми. Зображена схема у середовищі Tinkercad, яка має всі необхідні компоненти для

роботи фітолампи. Проаналізовано кожен елемент схеми, зокрема ІЧ-приймач, мікроконтролер Arduino UNO R3 та світлодіоди.

Було також визначено вимоги до програмного забезпечення. Програма повинна реалізовувати функції включення, виключення, регулювання яскравості та вибір режимів освітлення. Також розглянуто необхідність розробки Android-додатка для зручного керування фітолампюю.

Була проведена реалізація схеми в середовищі Tinkercad перед фізичною реалізацією. Зображено віртуальний прототип фітолампи з усіма компонентами. Та була реалізація цієї схеми в фізичному форматі

Загальною метою цього розділу було познайомлення із структурною схемою та апаратним забезпеченням розробленої фітолампи, а також визначення вимог до програмного забезпечення. Даний розділ є основою для подальшого розгортання розробки, будівництва та програмування пристрою основуючись на вимогах програмного забезпечення.

# Розділ 3 Реалізація програмного коду та інструкцій для користувача

## 3.1 Реалізація програмного коду для приладу

В цьому розділі наша мета розробити програмний код для Android додатка та для мікроконтролера Arduino та подати інструкції по використанню для користувача. Для запрограмування мікроконтролера ми використаємо середовище Arduino IDE який приймає мову програмування C++.

Спочатку підключимо бібліотеку для роботи з нашим ІЧ-приймачем яка має назву IRremote.hpp за допомогою команди #include. (рис 3.1)

```
#include <IRremote.hpp>
```

Рисунок 3.1 – Підключення бібліотеки

Ця бібліотека функціональність нашому ІЧ-приймачу, а саме зчитування та передачу команд до мікроконтролера.

Далі налаштуємо піни для наших світлодіодів та ІЧ-приймача (рис 3.2)

```
const int RECV_PIN = 2;  
  
const int LEDS[] = {9, 10, 11, 6, 5, 3};
```

Рисунок 3.2 – Піни елементів

Ініціалізуємо наші елементи для схеми а саме світлодіоди і ІЧ-приймач в методі void setup() (рис 3.3)

```
void setup()  
{  
  for(int i =0; i<6;i++)  
  {  
    pinMode(LEDS[i], OUTPUT);  
  }  
  Serial.begin(9600);  
  IrReceiver.begin(RECV_PIN, ENABLE_LED_FEEDBACK);  
}
```

Рисунок 3.3 – Ініціалізація елементів

Цикл `for` використовується для ініціалізації світлодіодів, а `IrReceiver.begin(RECV_PIN, ENABLE_LED_FEEDBACK)` для ІЧ приймача. Так як ми ініціалізували всі наші елементи напишемо для них обробник команд за допомогою команд які є в нашій бібліотеці (рис 3.4)

```
if (IrReceiver.decode()) {
    switch (IrReceiver.decodedIRData.decodedRawData) {
```

Рисунок 3.4 – Оброблення сигналів ІЧ-приймача

За допомогою команди `IrReceiver.decode()` ми перевіряємо надійшов якийсь сигнал до ІЧ- приймача чи ні. `IrReceiver.decodeIRData.decodedRawData` це команда яка зчитує сигнали з приймача і переключає функції вслід з її значенням. Додавимо тепер функцію включення і виключення для лампи

```
case 0x25AE7EE0:
if (count % 2 == 0)
{
    update();
    count += 1;
    break;
}
else
{
    off();
    count += 1;
    break;
}
```

Рисунок 3.5 – Функція включення і виключення

Case `0x20BB7EE6` – це одна з команд на нашому пульті коли ми натиснемо кнопку на пульті з таким же HEX кодом то відбудеться виключення і включення лампи.

`Count % 2 == 0` це умова при якій відбувається включення і виключення якщо значення `Count` парне то лампа виключається якщо не парне то виключається і якщо ми один раз натиснули кнопку то до змінної `count` додамо 1 для подальшого включення і виключення

`Update()` – це метод оновлення даних про наші світлодіоди.

Off() – це метод для виключення світлодіодів, а саме зробивши їх яскравість рівною 0

```
void off()
{
    for(int i =0; i<6;i++)
    {
        analogWrite(LEDs[i], 0);
    }
    return;
}
```

Рисунок 3.5 – Метод off()

Напишемо код для обробки світлодіодів а саме метод Update()

```
if(blueoff==false&&redoff==false)
{
    for(int i =0; i<6;i++)
    {
        analogWrite(LEDs[i], brightness);
    }
}
```

Рисунок 3.6 – Метод update()

Далі ми повинні реалізувати зміну яскравості нашої лампи

```

case 0x20BB7EE6:
    if (IrReceiver.decodedIRData.decodedRawData != 0xE9CA1E0C && brightness < 255 && !blueoff && !redoff) {
        brightness += 51;
        update();
        break;
    } else if (IrReceiver.decodedIRData.decodedRawData != 0xE9CA1E0C && redbrightness < 255 && blueoff && !redoff) {
        redbrightness += 51;
        update();
        break;
    } else if (IrReceiver.decodedIRData.decodedRawData != 0xE9CA1E0C && bluebrightness < 255 && redoff && !blueoff) {
        bluebrightness += 51;
        update();
        break;
    }
    break;
case 0xE9CA1E0C:
    if (IrReceiver.decodedIRData.decodedRawData != 0x20BB7EE6 && brightness > 0 && !blueoff && !redoff) {
        brightness -= 51;
        update();
        break;
    } else if (IrReceiver.decodedIRData.decodedRawData != 0x20BB7EE6 && redbrightness > 0 && blueoff && !redoff) {
        redbrightness -= 51;
        update();
        break;
    } else if (IrReceiver.decodedIRData.decodedRawData != 0x20BB7EE6 && bluebrightness > 0 && redoff && !blueoff) {
        bluebrightness -= 51;
        update();
        break;
    }
    break;

```

Рисунок 3.7 – Функція зміни яскравості

При натисненні кнопки з HEX кодом 0x20BB7EE6 ми збільшимо яскравість нашої лампи якщо натиснемо кнопку з HEX кодом 0xE9CA1E0C то зменшимо яскравість шляхом збільшення або зменшення змінної brightness яка була проініціалізована пізніше і оновленні дані про світлодіоди методом Update().

```
int brightness = 255;
```

Рисунок 3.8 – змінна brightness

Чим більше значення змінної brightness тим більше яскравості має наша лампа. Максимальне значення яскравості це 255, мінімальне це 0  
Тепер реалізуємо режими освітлення нашого прилада.

```

case 0xCFB1259D:
    if (IrReceiver.decodedIRData.decodedRawData != 0x34CF3F65 && IrReceiver.decodedIRData.decodedRawData != 0xA9AEB37) {
        redoff = true;
        blueoff = false;
        update();
        break;
    }
    break;

case 0x34CF3F65:
    if (IrReceiver.decodedIRData.decodedRawData != 0xCFB1259D && IrReceiver.decodedIRData.decodedRawData != 0xA9AEB37) {
        redoff = false;
        blueoff = true;
        update();
        break;
    }
    break;

case 0xA9AEB37:
    if (IrReceiver.decodedIRData.decodedRawData != 0xCFB1259D && IrReceiver.decodedIRData.decodedRawData != 0x34CF3F65) {
        redoff = false;
        blueoff = false;
        update();
    }
    break;
}

```

Рисунок 3.9 – Функція зміни режиму освітленості

Ми змінюємо режим шляхом перемиканню кнопок з певним значеннями HEX кодів. Перемикання відбувається шляхом перемиканням флажка redoff – це якщо червоні світлодіоди вимкнуті і blueoff –це якщо сині світлодіоди вимкнуті. В слід цього змінимо логіку в наших попередніх методах для змінення яскравості при певному режимі і включення і виключення при певних режимах



```

void update()
{
    if(blueoff==false&&redoff==false)
    {
        for(int i =0; i<6;i++)
        {
            analogWrite(LED5[i], brightness);
        }
    }
    else if (blueoff==true &&redoff==false)
    {
        for(int i =0; i<6;i++)
        {
            if (i !=2&&i!=5) {
                analogWrite(LED5[i], redbrightness);
            } else {
                analogWrite(LED5[i], 0);
            }
        }
    }

    else if (blueoff==false &&redoff==true)
    {
        for(int i =0; i<6;i++)
        {
            if (i !=2&&i!=5)
            {
                analogWrite(LED5[i], 0);
            }
            else
            {
                analogWrite(LED5[i], bluebrightness);
            }
        }
    }
}

```

Рисунок 3.10 – Оновлений метод Update()

## 3.2 Реалізація програмного коду для Android додатку

Для написання коду для Android додатку використаємо середовище Android Studio і мову програмування Kotlin.

По перше підключимо певні бібліотеки для роботи з ІЧ-передатчиком на нашому смартфоні за допомогою команди import

```
import androidx.appcompat.app.AppCompatActivity
```

Рисунок 3.11 – Бібліотека роботи з ІЧ-передатчиком

Далі ініціалізуємо сам ІЧ-приймач на нашому смартфоні для нашої програми

```
private lateinit var consumerIrManager: ConsumerIrManager
```

Рисунок 3.12 – Ініціалізація ІЧ-приймача

Далі для передачі сигналів використаємо бібліотечний метод `consumerIrManager.transmit`

```
consumerIrManager.transmit( carrierFrequency: 32000, irPattern)
```

Рисунок 3.13 – Передача сигналів за допомогою `consumerIrManager.transmit`

32000- це значення частоти на якій ми подаємо наші сигнали і вона дорівнює 32кГц. `IrPattern` це масив чисел які ми подаємо як паттерн нашого сигналу. Сам паттерн представляє собою послідовність імпульсів які із себе випуска інфрачервоний пульт. Напишемо код для відсилання певних імпульсів для пульта.

```
val btnSendIRCode = findViewById<Button>(R.id.btnSendIRCode)
btnSendIRCode.setOnClickListener { it: View!
    val irPattern = intArrayOf(1000, 2000, 1000, 2000, 1000, 2000)
    cop(irPattern)
}
val btnSendIRCode1 = findViewById<Button>(R.id.btnSendIRCode1)
btnSendIRCode1.setOnClickListener { it: View!
    val irPattern = intArrayOf(1000, 1000, 1000, 2000, 2000, 2000, 3000, 3000, 3000)
    cop(irPattern)
}
val btnSendIRCode2 = findViewById<Button>(R.id.btnSendIRCode2)
btnSendIRCode2.setOnClickListener { it: View!
    val irPattern = intArrayOf(1000, 2000, 3000, 2000, 1000, 2000, 1000, 2000, 1000, 2000)
    cop(irPattern)
}
val btnSendIRCode3 = findViewById<Button>(R.id.btnSendIRCode3)
btnSendIRCode3.setOnClickListener { it: View!
    val irPattern = intArrayOf(1000, 2000, 3000, 1000, 2000, 3000, 1000, 2000, 3000)
    cop(irPattern)
}
val btnSendIRCode4 = findViewById<Button>(R.id.btnSendIRCode4)
btnSendIRCode4.setOnClickListener { it: View!
    val irPattern = intArrayOf(1000, 2000, 3000, 1000, 2000, 3000, 2000, 2000, 2000)
    cop(irPattern)
}
val btnSendIRCode5 = findViewById<Button>(R.id.btnSendIRCode5)
btnSendIRCode5.setOnClickListener { it: View!
    val irPattern = intArrayOf(1000, 2000, 3000, 1000, 2000, 3000, 1000, 2000, 3000, 1000, 2000, 3000)
    cop(irPattern)
}
```

Рисунок 3.14 – Реалізація подачі паттернів

Розглянемо на прикладі першого паттерна. При натисканні кнопки в нашому додатку ІЧ-передатчик виконує послідовну передачу імпульсів в першому

випадку маємо 6 імпульсів як мають 3 імпульса по 1000 мілісекунд та 3 імпульса по 2000 мілісекунд. Ці імпульси конвертуються в HEX код і в відповідності до цього виконуються команди на самому мікроконтролері.

Так як у нас є вже функціонал самого додатку залишилось написати інтерфейс для нашого додатка.

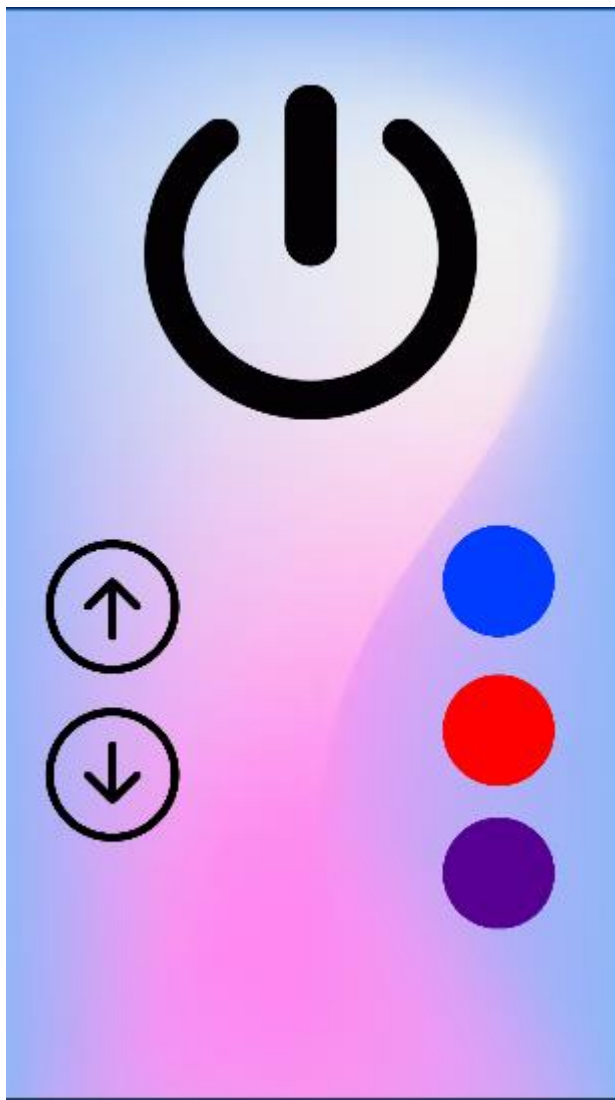


Рисунок 3.14 – Інтерфейс додатка

В нашому додатку ми маємо 6 кнопок за допомогою яких ми будемо виконувати керування нашою фітолампю. Кнопка включення, дві кнопки стрілки які відповідають за яскравість лампи. Три різнокольорових круга відповідають за режими свічення лампи Синій- вмикає синій режим освітлення

,Червоний - вмикає червоний режим освітлення, Фіолетовий вмикає фіолетовий режим освітлення.

### 3.3 Тестування схеми і Android- додатка та інструкція для користувача

Для перевірки схеми підключимо її до комп'ютера і за допомогою Android додатка будемо позмінювати їй режими роботи. Спочатку провіримо чи світяться наші світлодіоди підключенні до схеми чи ні.

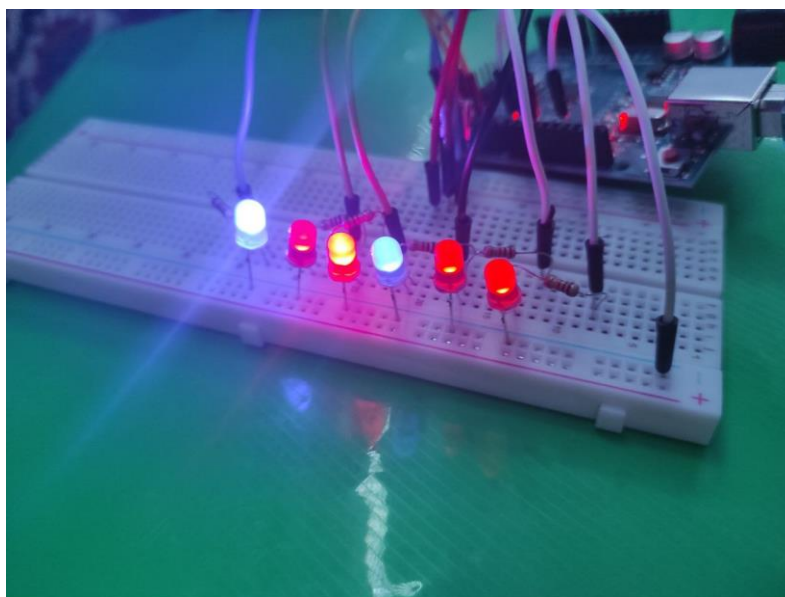


Рисунок 3.15 – Включення лампи.

Як бачимо все працює всі світлодіоди горять. Тепер з допомогою додатка спробуємо виключити лампу

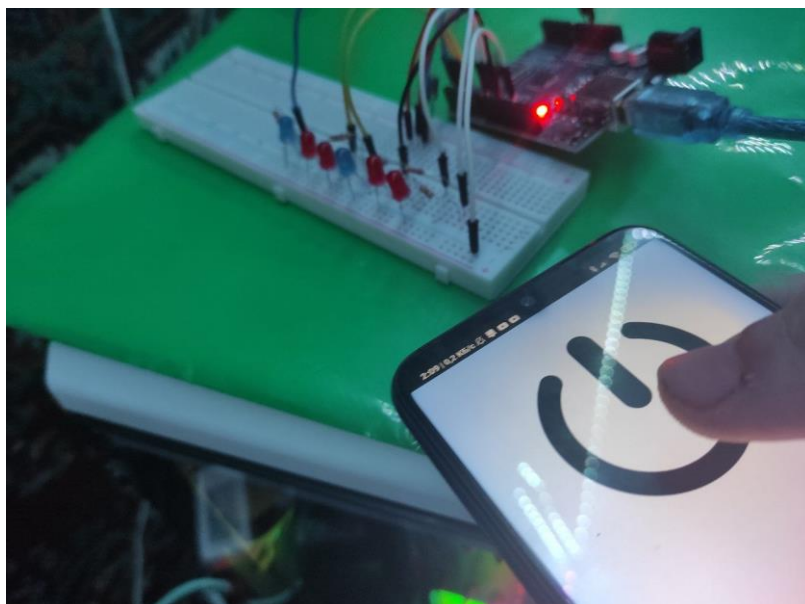


Рисунок 3.16 – Виключення лампи.

Як бачимо лампа потухла. Тепер включимо її знову і включимо режим освітлення синій.



Рисунок 3.17 –включення синього режиму.

Синій режим працює. Тепер спочатку включимо червоний режим потім спробуємо змінити яскравість на світлодіодів спочатку до мінімуму потім до максимуму.

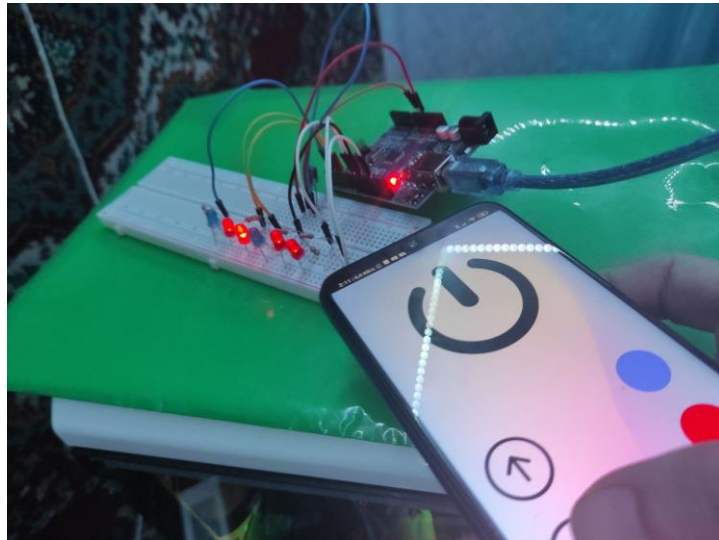


Рисунок 3.18 – включений червоний режим при мінімальній яскравості

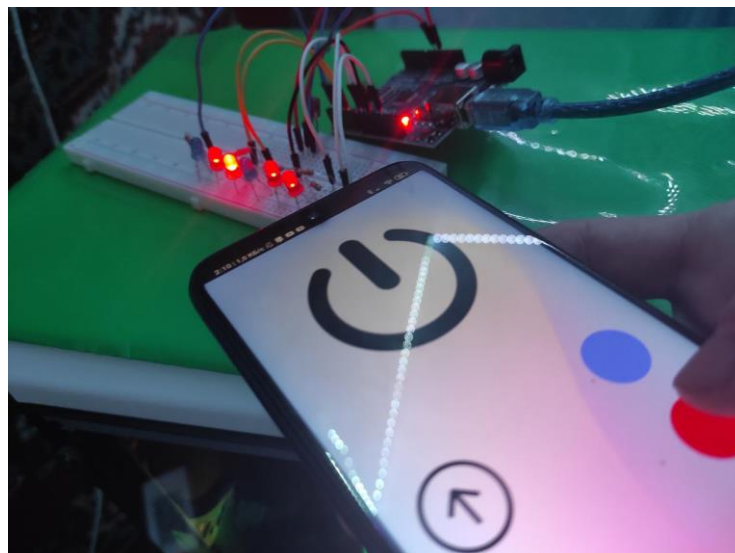


Рисунок 3.19 – включений червоний режим при максимальній яскравості

Всі ці тести були проведені з Android додатком який також працює так само як і режими на фітолампі.

## Висновки

У даній курсовій роботі було розглянуто та реалізовано проект фітолампи з системою керування за допомогою мобільного додатка на платформі Android. Основною метою проекту було створення зручного та ефективного інструмента для керування освітленням рослин у домашніх умовах, що дозволить оптимізувати умови їх росту та розвитку.

У процесі виконання курсової роботи було реалізовано мобільний додаток для керування фітолампюю яка дозволила користувачеві зручно встановлювати режими роботи, регулювати інтенсивність світла та використовувати розклади для автоматичного включення та виключення освітлення.

У підсумку, розроблена фітолампа з мобільним додатком для керування представляє сучасний та ефективний засіб для підтримки росту рослин у домашніх умовах. Цей проект може знайти застосування серед гідропонічних систем, індивідуальних садів чи просто серед любителів рослин, які бажають оптимізувати умови їх вирощування.



## Список використаних джерел

1. Плодоношення за допомогою фіто LED світильників у квартирі. [Електронний ресурс]. Режим перегляду: <https://ledukraine.com/svitlo-innovatsii-fitolampy-led-dlia-vashoho-rostu/>
2. Arduino Uno. [Електронний ресурс]. Режим перегляду: <https://doc.arduino.ua/ru/hardware/Uno>
3. Потреба рослин в світловому потоці. [Електронний ресурс]. Режим перегляду: <https://ledbox.ua/ua/a172278-potrebnost-rastenij-svetovom.html>
4. Що таке ІК-Приймач? [Електронний ресурс]. Режим перегляду: <http://ua.allight-zd.com/news/what-is-an-ir-receiver-module-19259656.html>
5. А. В. Ярмілко, В.В. Веретельник - Методичні вказівки до виконання та оформлення курсової роботи з дисципліни «Електроніка та комп'ютерна схемотехніка»



Додаток А  
**Лістинг програми для мікроконтролера**

```
#include <IRremote.hpp>

const int RECV_PIN = 2;

const int LEDS[] = {9, 10, 11, 6, 5, 3};
int count = 0;
int brightness = 255;
int bluebrightness = 255;
int redbrightness = 255;
bool blueoff = false;
bool redoff = false;

void setup()
{
    for(int i =0; i<6;i++)
    {
        pinMode(LEDS[i], OUTPUT);
    }
    Serial.begin(9600);
    IrReceiver.begin(RECV_PIN, ENABLE_LED_FEEDBACK);
}

void update()
{
    if(blueoff==false&&redoff==false)
    {
        for(int i =0; i<6;i++)
        {
            analogWrite(LEDS[i], brightness);
        }
    }
    else if (blueoff==true &&redoff==false)
    {
        for(int i =0; i<6;i++)
        {
            if (i !=2&&i!=5) {
                analogWrite(LEDS[i], redbrightness);
            } else {
                analogWrite(LEDS[i], 0);
            }
        }
    }

    else if (blueoff==false &&redoff==true)
    {

```

```

for(int i =0; i<6;i++)
{
    if (i !=2&& i!=5)
    {
        analogWrite(LED5[i], 0);
    }
    else
    {
        analogWrite(LED5[i], bluebrightness);
    }
}
}

void off()
{
    for(int i =0; i<6;i++)
    {
        analogWrite(LED5[i], 0);
    }
    return;
}

void loop() {
    if (IrReceiver.decode()) {

        switch (IrReceiver.decodedIRData.decodedRawData) {
            case 0x25AE7EE0:
if (count % 2 == 0)
{
    update();
    count += 1;
    break;
}
else
{
    off();
    count += 1;
    break;
}

            case 0x20BB7EE6:
                if (IrReceiver.decodedIRData.decodedRawData != 0xE9CA1E0C && brightness <
255 && !blueoff && !redoff) {
                    brightness += 51;
                    update();
                    break;
                } else if (IrReceiver.decodedIRData.decodedRawData != 0xE9CA1E0C &&
redbrightness < 255 && blueoff && !redoff ) {

```

```

        redbrightness += 51;
        update();
        break;
    } else if (IrReceiver.decodedIRData.decodedRawData != 0xE9CA1E0C &&
bluebrightness < 255 && redoff && !blueoff ) {
        bluebrightness += 51;
        update();
        break;
    }
    break;

    case 0xE9CA1E0C:
        if (IrReceiver.decodedIRData.decodedRawData != 0x20BB7EE6 && brightness > 0
&& !blueoff && !redoff) {
            brightness -= 51;
            update();
            break;
        } else if (IrReceiver.decodedIRData.decodedRawData != 0x20BB7EE6 &&
redbrightness > 0 && blueoff && !redoff ) {
            redbrightness -= 51;
            update();
            break;
        } else if (IrReceiver.decodedIRData.decodedRawData != 0x20BB7EE6 &&
bluebrightness > 0 && redoff && !blueoff) {
            bluebrightness -= 51;
            update();
            break;
        }
        break;

    case 0xCFB1259D:
        if (IrReceiver.decodedIRData.decodedRawData != 0x34CF3F65 &&
IrReceiver.decodedIRData.decodedRawData != 0xA9AEAB37) {
            redoff = true;
            blueoff = false;
            update();
            break;
        }
        break;

    case 0x34CF3F65:
        if (IrReceiver.decodedIRData.decodedRawData != 0xCFB1259D &&
IrReceiver.decodedIRData.decodedRawData != 0xA9AEAB37) {
            redoff = false;
            blueoff = true;
            update();
            break;
        }
        break;

```

```

        case 0xA9AEAB37:
            if (IrReceiver.decodedIRData.decodedRawData != 0xCFB1259D &&
                IrReceiver.decodedIRData.decodedRawData != 0x34CF3F65) {
                redoff = false;
                blueoff = false;
                update();
            }
            break;
        }
        IrReceiver.resume();
    }
}

```

## Додаток В

### Лістинг програми для Android додатка

```

package com.example.myapplication

import android.content.Context
import android.hardware.ConsumerIrManager
import android.os.Bundle
import android.widget.Button
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var consumerIrManager: ConsumerIrManager
    fun cop (irPattern: IntArray) {
        if (consumerIrManager.hasIrEmitter()) {
            consumerIrManager.transmit(32000, irPattern)
        } else {
            Toast.makeText(this, "Данна програма не підтримується  
вашим смартфоном!", Toast.LENGTH_SHORT).show()
        }
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        val irPattern = intArrayOf();
        super.onCreate(savedInstanceState)
        setContentView(R.layout.active)

        consumerIrManager =
            getSystemService(Context.CONSUMER_IR_SERVICE) as ConsumerIrManager

        val btnSendIRCode =
            findViewById<Button>(R.id.btnSendIRCode)
        btnSendIRCode.setOnClickListener {

```

```

        val irPattern = intArrayOf(1000, 2000, 1000, 2000,
1000, 2000)
        cop(irPattern)
    }
    val btnSendIRCode1 =
findViewById<Button>(R.id.btnSendIRCode1)
    btnSendIRCode1.setOnClickListener {
        val irPattern = intArrayOf(1000, 1000, 1000, 2000,
2000, 2000, 3000, 3000, 3000)
        cop(irPattern)
    }
    val btnSendIRCode2 =
findViewById<Button>(R.id.btnSendIRCode2)
    btnSendIRCode2.setOnClickListener {
        val irPattern = intArrayOf(1000, 2000, 3000, 2000,
1000, 2000, 1000, 2000, 1000, 2000)
        cop(irPattern)
    }
    val btnSendIRCode3 =
findViewById<Button>(R.id.btnSendIRCode3)
    btnSendIRCode3.setOnClickListener {
        val irPattern = intArrayOf(1000, 2000, 3000, 1000,
2000, 3000, 1000, 2000, 3000)
        cop(irPattern)
    }
    val btnSendIRCode4 =
findViewById<Button>(R.id.btnSendIRCode4)
    btnSendIRCode4.setOnClickListener {
        val irPattern = intArrayOf(1000, 2000, 3000, 1000,
2000, 3000, 2000, 2000, 2000)
        cop(irPattern)
    }
    val btnSendIRCode5 =
findViewById<Button>(R.id.btnSendIRCode5)
    btnSendIRCode5.setOnClickListener {
        val irPattern = intArrayOf(1000, 2000, 3000, 1000,
2000, 3000, 1000, 2000, 3000, 1000, 2000, 3000)
        cop(irPattern)
    }
}
}
}

```