

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ
ANDROID ДЛЯ УПРАВЛІННЯ РОЗУМНИМ СВІТИЛЬНИКОМ
RESEARCH AND DEVELOPMENT OF ANDROID SOFTWARE
APPLICATION FOR A SMART LAMP CONTROL

Виконав: студент 2 курсу, групи КНТ-110м
Спеціальності 121 Інженерія
програмного забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Петренський В.О.

(прізвище та ініціали)

Керівник Каплієнко Т.І.

(прізвище та ініціали)

Рецензент Шитікова О.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ПРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 20 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

Петренського Віталія Олександровича
 (прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та розробка програмного застосунку
Android для управління розумним світильником
Research and Development of Android Software Application for a Smart Lamp
Control

керівник проєкту (роботи) Каплієнко Тетяна Ігорівна, доцент,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “5” листопада 2021 року № 435

2. Строк подання студентом проєкту (роботи) 13 грудня 2021 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне
завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Аналіз програмних та апаратних засобів. 3. Опис практичної реалізації розумного світильника 4. Керівництво програміста. 5. Керівництво користувача.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Каплієнко Т.І., доцент		
Нормоконтроль	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання “1” вересня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постанова завдання	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2-3 тижні	Розділ 1
3	Аналіз програмних та апаратних засобів	4-5 тижні	Розділ 2
4	Опис практичної реалізації розумного світильника	6-8 тиждень	Розділ 3
5	Керівництво програміста	9-11 тиждень	Розділ 4
6	Керівництво користувача	12-13 тижні	Розділ 5
7	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	14-15 тиждень	Додатки
8	Захист роботи	16 тиждень	

Студент

(підпис) Петренський В.О.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Каплієнко Т.І.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
96 с., 56 рис., 12 табл., 2 дод., 24 джерел.

RGB, ARDUINO, WI-FI, ESP32, GPIO, WITTY CLOUD, B4A, LED,
Android, IDE, IoT, ЛЮМЕН

Об'єкт дослідження – світловий потік та вплив світла на організм людини, а також процес дистанційного керування розумним світильником.

Предмет дослідження – методи та засоби дистанційного керування розумного світильника на основі вимірювання світлового потоку.

Мета роботи – створення розумного світильника та програмна реалізація android застосунку для дистанційного керування на основі використання бездротових та мобільних технологій, а також коригувати яскравість світильника на основі даних світлового потоку, які залежать від часу доби.

Матеріали, методи та технічні засоби: світодіодна матриця 16x16, набір дротів, пластмасовий плафон LLDPE, сенсорна кнопка TTP223, програмна частина реалізована в середовищі розробки Arduino, апаратним засобом було обрано модуль ESP8266 та реалізація дистанційного керування розумного світильника виконується за допомогою застосунку на android, який з'єднується по мережі Wi-Fi.

Виконана розробка призначена для покращення комфорту та атмосфери у приміщенні, дозволяє скоротити витрати на час керування розумним світильником та електроенергії, а також на основі даних світлового потоку змінювати яскравість розумного світильника.

ABSTRACT

Explanatory note to the diploma qualifying work of the master's degree: 96 pages, 11 tables, 56 figures, 2 appendixes, 24 sources.

RGB, ARDUINO, WI-FI, ESP32, GPIO, WITTY CLOUD, B4A, LED, Android, IDE, IoT, Jm

The object of study – the light flux and the impact of light on the human body, as well as the process of remote control of a smart lamp.

The subject of research – methods and means of remote control of a smart lamp based on the measurement of luminous flux.

The purpose of the work – is to create a smart lamp that works on the basis of luminous flux data and software implementation on android applications for remote control based on the use of wireless and mobile technologies.

Materials, methods and hardware: 16x16 LED matrix, wire set, LLDPE plastic cover, TTP223 touch button, software implemented in Arduino development environment, ESP8266 module was selected as hardware and remote lighting control is implemented using android application. connects via Wi-Fi.

The developed design is designed to improve the comfort and atmosphere in the room, reduces the cost of control of lighting and electricity, as well as on the basis of luminous flux data to change the brightness of a smart lamp.

ЗМІСТ

	С.
ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК	9
ВСТУП	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Вплив світла на організм людини	11
1.2 Поняття кольору та його характеристика.....	13
1.3 Світлова чутливість людського ока	14
1.3 Психологія сприйняття кольору	16
1.4 Якість освітлення та зоровий комфорт	17
1.5 Технології Led	18
1.6 Висновки	22
2 АНАЛІЗ ПРОГРАМНИХ ТА АПАРАТНИХ ЗАСОБІВ	23
2.1 Огляд особливостей протоколів зв'язку для розумного світильника	23
2.1.1 1-Wire	23
2.1.2 X10.....	23
2.1.3 KNX	24
2.1.4 Wi-Fi	24
2.1.5 Insteon	25
2.1.6 ZigBee	25
2.1.7 Z-Wave.....	26
2.1.8 Bluetooth	27
2.2 Огляд особливостей середовищ програмування	29
2.2.1 Microsoft Visual Studio	29
2.2.2 PlatformIO.....	30
2.2.3 Atmel Studio	30
2.2.4 Arduino IDE	30
2.3 Огляд особливостей середовищ програмування на Android	32
2.3.1 Eclipse	33
2.3.2 IntelliJ IDEA	33

2.3.3 Android Studio	34
2.3.4 B4A	35
2.4 Різновиди мікроконтролерів серії ESP	37
2.4.1 ESP-01.....	37
2.4.2 ESP-03.....	38
2.4.3 ESP-07.....	39
2.4.4 ESP-12E NodeMCU	40
2.4.5 ESP8266 Witty Cloud.....	47
2.5 Висновки	50
3 ОПИС ПРАКТИЧНОЇ РЕАЛІЗАЦІЇ РОЗУМНОГО СВІТИЛЬНИКА	51
3.1 Світловий потік	51
3.1.1 Освітленість	51
3.1.2 Вимірювання світлового потоку	53
3.2 Аналіз вимог	54
3.3 Архітектура системи.....	55
3.4 Алгоритм роботи розумного світильника	57
3.5 Опис особливостей програмної реалізації.....	58
3.6 Висновки	61
4 КЕРІВНИЦТВО ПРОГРАМІСТА	62
4.1 Призначення і умови використання Arduino IDE	62
4.2 Налаштування скетчу.....	64
4.3 Призначення і умови використання B4A	67
4.4 Налаштування проєкту на Android	68
4.5 Повідомлення програмісту.....	69
5 КЕРІВНИЦТВО КОРИСТУВАЧА	71
5.1 Призначення застосунка LedLamp	71
5.2 Умови виконання програми	71
5.3 Як запустити програму	72
5.4 Виконання програми.....	72
ВИСНОВКИ.....	77

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	78
Додаток А Текст програми.....	81
Додаток Б Слайди презентації	87

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

B4A	– Basic for Android;
ESP	– Electronic Stability Program;
FHSS	– Frequency-hopping spread spectrum;
IDE	– Integrated Development Environment;
IoT	– Internet of Jhings;
LED	– Light-emitting diode;
RGB	– Red, Green, Blue;
Wi-Fi	– Wireless Fidelity;
АЦП	– аналого-цифровий перетворювач
БНіП	– будівельні норми і правила;
ДЕСТ	– державний стандарт;
КПК	– кишеньковий персональний комп'ютер;
Лк	– люкс;
Лм	– люмен;
МК	– мікроконтролер;
Нм	– нанометр;
ПК	– персональний комп'ютер
ОС	– операційна система.

ВСТУП

Забезпечення повноцінного та правильного освітлення у будинку – це не лише декоративна необхідність. Крім естетичних нюансів, рівень і тип підсвічування прямо впливає на багато аспектів життя людини, включаючи емоційний стан, продуктивність, а також фізичне та психічне здоров'я.

В першу чергу освітлення впливає на зорові нерви людини, через які ми отримуємо 90% усієї інформації про навколишній світ. Недостатній рівень світла змушує напружувати зір, що призводить до швидкої втоми очних м'язів, загальної сонливості, загальних болів та мігрені.

Вирішення цієї проблеми є метою дипломної роботи, а саме створення розумного світильника, який працює на основі даних світлового потоку та програмна реалізація android застосунку для дистанційного керування на основі використання бездротових та мобільних технологій.

Розумний світильник на даний момент пропонує людям гнучкість у застосуванні, безпеку, а також економію витрат на електроенергію. Такі світильники, наприклад, можна запрограмувати на увімкнення або вимкнення у певний час. До того ж більшість з таких світильників керується дистанційно за допомогою мобільного застосунку або планшета.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Вплив світла на організм людини

Для створення безпечних умов праці потрібна [1] не тільки достатня освітленість робочих місць, [1] а й раціональний напрямок світла, відсутність різких тіней і відблисків, що викликають сліпучу дію [1].

Правильне освітлення та фарбування обладнання, небезпечних місць дає можливість стежити за ними уважніше [1], а попереджувальне фарбування небезпечних місць дозволить зменшити травматизм [1]. Крім того, підбір правильного поєднання [1] кольорів та їх інтенсивності зведе до мінімуму час адаптації очей [1]. Правильно підібране фарбування може впливати на настрій робітників, а отже, і на [1] продуктивність праці [1]. Таким чином, недооцінка впливу освітлення, вибору кольору та [1] світла призводять до передчасної втоми організму, накопичення помилок, зниження продуктивності праці, та як наслідок, до травматизму [1]. Певна зневага до питань [1] освітленості викликана тим, що око людини має дуже широкий діапазон пристосування: від 20 лк (повний місяць) до 100000 лк [1].

Колір та світло [1] взаємопов'язані між собою [1]. Кольори, що спостерігаються людиною, поділяються на [1] хроматичні та ахроматичні. Ахроматичні кольори (білий, сірий, чорний) мають різні коефіцієнти відображення і тому основною [1] їх характеристикою є яскравість [1]. Хроматичні кольори (червоний, помаранчевий, жовтий, зелений, блакитний, синій та фіолетовий) характеризуються, [1] в основному, тоном, що визначається довжиною хвилі та чистотою чи насиченістю (ступінь "розбавленості" основного кольору білим) [1]. Забарвлення обладнання, матеріалів та ін. у [1] чорний колір пригнічує людину [1]. При перенесенні стандартних ящиків білого та чорного кольору всі робітники заявили, що чорні ящики важчі [1]. Чорну нитку на білому тлі видно у 2100 разів краще [1], ніж на чорному, але при цьому спостерігається різкий контраст (відношення яскравостей) [1]. Зі збільшенням яскравості та висвітлення до певних меж

посилюється гострота зору і яскравість, з якою око розрізняє окремі предмети, тобто швидкість розрізнення [1]. Занадто велика яскравість світла негативно впливає на органи зору, викликаючи [1] засліплення та різь в очах [1]. Пристосування очей до зміни [1] яскравості називається темною та світлою адаптацією [1]. Працюючи на верстаті темно-сірого [1] кольору (відбиває 5% світла) і з блискучою деталлю (відбиває 95% кольору) робочий переводить погляд зі верстата деталь 1 разів у хвилину, у своїй адаптацію очі витрачається приблизно 5 секунд [1]. За семигодинний робочий [1] день буде втрачено 35 хвилин [1]. Якщо за тих же умов роботи змінити [1] час адаптації до 1 секунди за рахунок правильного підбору контрасту, втрата робочого часу дорівнюватиме 7 хвилин [1].

Неправильний підбір освітлення впливає не лише на втрату робочого часу та втому робітників, але й збільшує травматизм [1] у період адаптації, коли робітник не бачить або погано бачить деталь, та виконує робочі операції автоматично [1]. Подібні умови спостерігаються [1] при монтажних роботах, роботі крана та інших видах робіт у вечірній [1] час при штучному освітленні [1]. Тому ставлення яскравостей (сутність розмаїття) повинно [1] бути великим [1].

Нормування освітлення виробничих [1] приміщень та робочих місць здійснюється на науковій основі з [1] урахуванням наступних основних вимог [1]:

- достатня та рівномірна [1] освітленість робочих місць та оброблюваних деталей [1];
- відсутність яскравості, бляклості [1] та сліпучої дії в полі зору робітників [1];
- відсутність різких тіней та контрастів [1];
- оптимальна економічність та [1]безпека освітлювальних систем [1].

Отже, правильного світлового режиму необхідно враховувати весь комплекс гігієнічних умов, тобто. кількісну та якісну сторони освітлення [1].

Для вимірювання освітлених робочих місць та загальної освітленості приміщень використовують люксметр типу Ю-116, Ю-117, універсальний люксметр – яскравомір ТЕС 0693, фотометр типу 1105 фірми Брюль і Кер. Принцип роботи приладів ґрунтується на використанні фотоелектричного ефекту – емісії електронів під дією світла [1].

При виконанні різних видів робіт застосовують природне, штучне та змішане освітлення, параметри яких регламентуються державний стандарт (ДЕСТ) 12.1.013-78 [1], будівельні норми і правила (БНіП) II-4-79 "Природне та штучне освітлення" [1], інструкцією з проектування електричного освітлення будівельних майданчиків. Усі приміщення з постійним перебуванням людей повинні мати природне освітлення [1].

Там, де неможливо здійснити природне висвітлення або якщо воно не регламентується БНіП II-4-79, застосовується штучне або змішане висвітлення [1].

1.2 Поняття кольору та його характеристика

Все що нас оточує бачимо має свій колір. Колір присутній майже у всьому, тому справедливо, що він впливає [2] на людину так само, як і все інше. Для того, щоб розібратися [2] в тому, як діє колір [2], необхідно визначити, звідки він [2] береться і які фактори впливають на його формування [2]. Колір можливий лише при [2] освітленні предмета джерелом світла – настільною лампою, ліхтариком, сонцем. Основні характеристики кольору це світло і темрява, чорне та біле. Це так звані нейтральні кольори. У 1676 році сер Ісаак Ньютон за допомогою [2] тригранної призми розклав біле сонячне світло на колірний спектр. Подібний спектр містив усі кольори, за винятком пурпурового [2]. Ньютон ставив свій досвід у такий [2] спосіб: сонячне світло пропускалося через вузьку щілину і падало на призму. У призмі промінь світла

розшаровувався [2] на чисті спектральні кольори. Потім промінь заломлювався і прямував на [2] екран, де утворювалося райдужне зображення спектра. Спектр поширювався від червоного до фіолетового [2].

1.3 Світлова чутливість людського ока

Здатність ока сприймати світло і розпізнавати різного ступеня його яскравості прийнято [3] називати світлопочуттям, а здатність пристосовуватися [3] до різної яскравості освітлення адаптацією ока; світлова чутливість [3] оцінюється величиною порога світлового подразника [3]. Людина з гарним [3] зором здатна розглянути вночі світло від свічки на відстані кількох кілометрів. Максимальна [3] світлова чутливість досягається після тривалої [3] темнотної адаптації. Її визначають під дією [3] світлового потоку в тілесному вугіллі 50° при [3] довжині хвилі 500 нм (максимум чутливості ока) [3]. У цих умовах порогова енергія світла близько 10^{-9} ерг/с, що еквівалентно потоку кількох [3] квантів оптичного діапазону за секунду через зіницю [3]. Вклад зіниці в регулювання чутливості [3] ока вкрай незначний [3]. Весь діапазон яскравостей, які наш зоровий механізм [3] здатний сприйняти, величезний: від 10^{-6} кдм² для ока, повністю адаптованого до темряви, до 10^6 кдм² для ока [3], повністю адаптованого до світла. Механізм такого широкого діапазону чутливості криється в розкладанні [3] та відновленні фоточутливих пігментів у фоторецепторах сітківки – колбочках та паличках [3]. В оці людини містяться два типи [3] світлочутливих клітин (рецепторів): високочутливі палички, що відповідають за сутінковий (нічний) зір, та менш чутливі колбочки, що відповідають за кольоровий зір [3].

У сітківці очі людини є три [3] види колб, максимумами чутливості яких припадають на червоний, зелений і синій ділянки спектра [3]. Розподіл типів колб у сітківці нерівномірно: «сині» колбочки знаходяться ближче до периферії, у той час як «червоні» і «зелені» розподілені випадковим чином [3]. Відповідність типів колб трьом [3] «основним» кольорам забезпечує

розпізнавання тисяч кольорів та відтінків [3]. Криві спектральної чутливості трьох видів колб частково перекриваються [3], що сприяє явищу метамерії це показано на рисунку 1.1. Пунктиром показано сутінкову, «чорно-білу» сприйнятливість паличок [3].

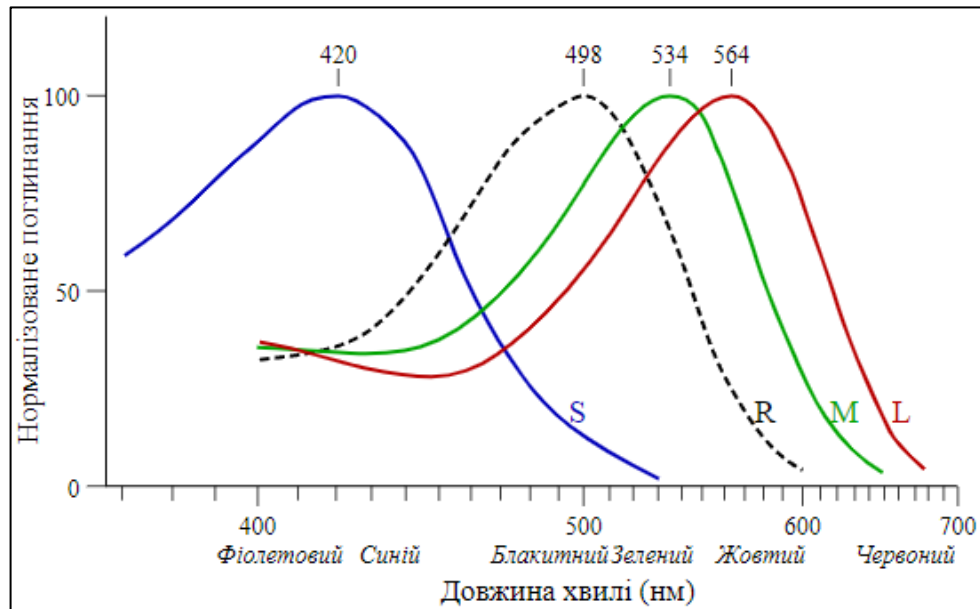


Рисунок 1.1 – Графік світлочутливості колб людського ока S, M, L [3]

Природне освітлення – це видимий діапазон (рисунок 1.2) випромінювання електромагнітних хвиль сонячної енергії довжиною 380 – 780 нм (1 Нанометр(нм) = 10^{-9} м) [3]. Видиме світло (білий) складається з спектру кольорів: фіолетовий (390 - 450 нм) [3], синій (450 - 510 нм) [3], зелений (510 - 575 нм) [3], жовтий (575 - 620 нм) [3], червоний (620 - 750 нм) [3]. Випромінювання з довжиною хвилі більше 780 нм [3] називається інфрачервоним, а з довжиною хвилі менше 390 нм – ультрафіолетовим [3].

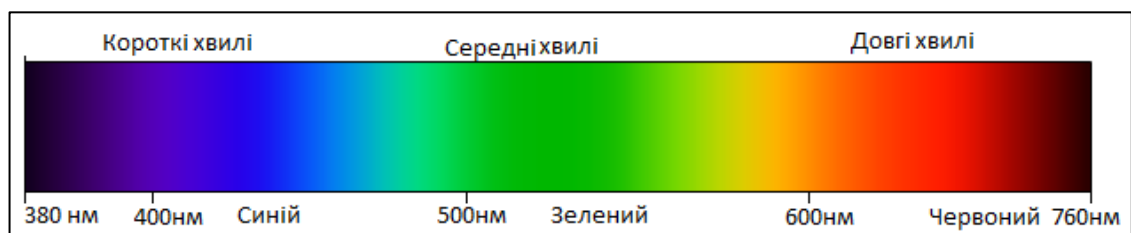


Рисунок 1.2 – Видимий діапазон частот [3]

Рівномірне роздратування всіх трьох елементів, що відповідає середньозваженому денному світлу [3], також викликає почуття білого кольору. За колірний зір людини відповідають гени [3], що кодують світлочутливі білки (описані в таблиці 1.1) [3].

Таблиця 1.1 – Сприйняття кольору [3]

Тип колб [3]	Позначення [3]	Сприймання довжини хвиль [3]	Максимум чутливості [3]
S [3]	β [3]	400 – 500 нм [3]	420 – 440 нм [3]
M [3]	α [3]	450 – 630 нм [3]	534 – 555 нм [3]
L [3]	ρ [3]	500 – 700 нм [3]	564 – 580 нм [3]

1.4 Психологія сприйняття кольору

Колір впливає на фізіологічні процеси людини та її психологічний стан. Знаючи особливості кожного кольору, можна сформувати певний образ, викликати певні емоції, асоціації [4].

Вплив кольору [5] на фізіологію людини [5] підтверджено експериментальним шляхом [5] і залежить від кількості та якості кольору [5], періоду дії, особливостей нервової системи, віку, статі та інших факторів [5].

Далі в таблиці 1.2 наведено приклад як саме кольори впливають на фізіологію людини.

Таблиця 1.2 – Порівняння кольорів

Колір	Збуджувальні	Пригнічуючі	Заспокійливі
Червоний	+		
Помаранчевий	+		
Жовтий	+		

Продовження таблиці 1.2

Колір	Збуджувальні	Пригнічуючи	Заспокійливі
Зелений			+
Голубий			+
Фіолетовий		+	
Темно-сірий (тон)		+	
Чорний (тон)		+	

1.5 Якість освітлення та зоровий комфорт

Світильники забезпечують рівномірне та комфортне освітлення приміщення, хорошу насиченість світлом, не створюють сліпучого ефекту та дискомфорту. Щоб підвищити економічність освітлення, обробка стелі має бути виконана з максимальним коефіцієнтом відбиття.

Комфорт сучасного житла знаходиться у прямій залежності від якісного освітлення. Воно дозволяє розрізняти колір, форму, яскравість предметів інтер'єру. Досягнення гарного зорового комфорту визначається рівнем світлотехнічних параметрів освітлювальних приладів, їхньою освітленістю, перенесенням кольорів, гармонією світла і інше. Грамотний підхід до вибору світильників та вміле їх поєднання з різними предметами інтер'єру за стилем, формою, кольором та конфігурацією, створює комфортну світлову атмосферу, надають інтер'єру особливої чіткості, виразності та індивідуальності. Будь-який світильник складається з освітлювальної арматури та джерела світла (лампи). Арматура світильника відіграє важливу роль у світловому просторі, вона здатна перерозподіляти світло лампи, перетворювати його властивості та створювати певні світлові ефекти. Особливість конструкції освітлювальної арматури впливають на якість освітлення [6].

Залежно від функціонального призначення світильники поділяють на окремі групи: загального, місцевого (локального), комбінованого, декоративного, комбінованого та експозиційного освітлення. За способом установки розрізняють стельові світильники, підвісні, настінні, настільні, підлогові і вбудовані. Світильники прямого світла зазвичай направляють світловий потік вниз, світильники розсіяного світла забезпечують співвідношення світлового потоку направлено вниз і вгору рівною мірою, світильники відбитого світла направляють повністю або частково світловий потік вгору [6].

Встановлюючи той чи інший світильник, потрібно керуватися принципами дизайнерських рішень. Важливо, щоб світильник був встановлений на своєму місці, і тільки тоді можна домогтися гарного зорового комфорту, гармонії світла і тіні. Безумовно, сприйняття інтер'єру залежить і від правильного розподілу яскравості та кольору між елементами внутрішнього оздоблення та предметами, розташованими в приміщенні. Якщо приміщення має витягнуту форму, досить вузьке і світильники в ньому розташовані вздовж середньої лінії стелі, воно буде здаватися ще вужчим. Розташувавши світильники по лінії, зміщеній до однієї зі стін приміщення, можна його візуально розширити [6].

Встановлюючи той чи інший світильник, потрібно керуватися принципами дизайнерських рішень. Важливо, щоб світильник був встановлений на своєму місці, і тільки тоді можна домогтися гарного зорового комфорту, гармонії світла та інше. Сприйняття інтер'єру залежить і від правильного розподілу яскравості та кольору між елементами внутрішнього оздоблення та предметами, розташованими в приміщенні [6].

1.6 Технології Led

Світловипромінюючий діод (LED) вважається одним із джерел освітлення нового покоління з низкою переваг, таких як висока ефективність,

висока надійність, тривалий термін служби, висока швидкість відгуку та енергозбереження. Оптичні характеристики світлодіодного світильника, такі як освітленість, яскравість, корельована колірна температура та розподіл спектральної потужності визначають умови освітлення, які можуть впливати на фізіологічні характеристики ока людини [7].

Завдяки постійно зростаючим знанням у галузі світлового потоку, узгодженості кольору та довговічності, світлодіоди отримали широкий спектр застосування, включаючи внутрішнє освітлення, комерційне освітлення, автомобільне освітлення, медичне освітлення, інтелектуальне освітлення тощо. Щодо внутрішнього освітлення, попередні дослідження зазвичай були зосереджені на енергоефективності та захисті навколишнього середовища, проте цих двох аспектів здається недостатньо без урахування людського фактора. Дедалі більше вчених почали висловлювати занепокоєння щодо візуального комфорту за різних умов висвітлення, що також відомо як зорова втома, викликана висвітленням. Для певного середовища освітлення є три виміри, які можуть описати її характеристики: якість світла, коефіцієнт денного світла, автономність денного світла та розподілу світла. Для світильника застосовні оптичні характеристики якості світла та яскравості, такі як освітленість, яскравість, корельована колірна температура, та розподіл спектральної потужності. Індекс світлорозподілу незвичайний для світильників, але часто використовується для внутрішнього освітлення. Ці оптичні індекси можуть охоплювати загальні характеристики доквілля освітлення та існують відповідні стандарти для оцінок, більшість з яких засновані на суб'єктивному сприйнятті дискомфорту [7].

На сьогоднішній день існують такі джерела світла: люмінесцентні, галогенні, газорозрядні, світлодіодні. Світлодіодні, незважаючи на недавню появу, активно набирають популярності та завойовують дедалі більшу частку ринку, витісняючи з неї аналоги [8]. Це відбувається тому, що для освітлення однієї і тієї ж площі приміщення Led вимагає потужності значно меншої, ніж для лампочки розжарювання [8].

Далі наведено порівняльну таблицю потужностей різних типів лампочок (таблиця 1.3) [8].

Таблиця 1.3 – Відповідності потужностей лампочок [8]

Потужність, Вт			Світловий потік у Люменах
Накалювання	Люмінесцентна	Світлодіодна	
20	5-7	2-3	200
40	10-13	4-5	400
60	15-16	8-10	700
75	18-20	10-12	900
100	25-35	12-15	1200
150	40-50	18-20	1800
200	60-80	25-30	2500

Як видно з таблиці, для освітлення одного і того ж простору потужність світлодіодних джерел світла при рівному значенні світлового потоку значно нижча, ніж у аналогів [8]. Для виробництва світлодіодних лампочок використовують найефективніші та найменш ресурсомісткі процеси [8]. Можна помітити, що потужність led лампочок більше працює на створення світлових потоків та мінімально – на виробництво тепла [8]. У той же час у лампах розжарювання до 90% електроенергії витрачається на вироблення тепла і лише 10% - безпосередньо на освітлення [8].

Потужність, це лише один з головних параметрів лампочки. Далі наведено порівняльну таблицю 1.4 технічних характеристик освітлювальних приладів [8].

Головним недоліком у світлодіодного освітлення це ціна, серед аналогів їхня вартість вища [8]. Але вони мають вищу потужність та більш тривалий термін роботи, таким чином вони компенсують високу ціну [8].

Рівень нагріву світлодіодних джерел набагато нижчий, ніж у ламп розжарювання, але все ж таки присутній та може вплинути на систему освітлення [8].

Технологія led впевнено охоплює більшу частину ринку освітлення. І хоча досі існує думка, що світло лампочки розжарювання більш приємне для очей, але і тут світлодіодний аналог може переграти за рахунок великого вибору кольору свічення – від теплого до холодного [8].

Таблиця 1.4 – Таблиця порівняння різних тип лампочок [8]

Технічні характеристики	Світлодіодна	Розжарювання	Люмінесцентна
Строк служби	50 000	1 000	25 000
Ефективність світла	80-100 Лм/Вт	10 Лм/Вт	140 Лм/Вт
Виділення тепла під час роботи	70°C	180°C	60°C
Світловий потік	450 Лм	420 Лм	450 Лм
Стійкість до перепадів напруги	низька	висока	висока
Чутливість до частого увімкнення/вимкнення	не впливає на термін служби	скорочує термін служби	скорочує термін служби
Температура кольорів, Кельвіна (K)	2000-6500	2700	3300
Допустима температура навколишнього середовища	-40°C +40°C	-60 °C +100°C	-60 °C +120°C
Екологічність	так	так	містить ртуть
Індекс передачі кольорів	80	100	80

Продовження таблиці 1.4

Технічні характеристики	Світлодіодна	Розжарювання	Люмінесцентна
Вібростійкість	висока	середня	низька
Спеціальна утилізація	не потрібно	не потрібно	Потребує спеціальних заходів утилізації
ККД	70-100%	50-80%	45-75%

1.7 Висновки

Отже, все що нас оточує має свій колір, тому справедливо, що він впливає на людину так само, як і все інше. Оскільки колір впливає на фізіологічні процеси людини та її психологічний стан, таким чином при сформуванні певного образу можна викликати різні стани емоцій або асоціацій.

Тому для розумного світильника основним освітленням буде світлодіодна матриця 16 на 16. Світильник також матиме різні кольорові ефекти для комфорту та атмосфери у приміщенні. Сам світильник буде зібраний в пластмасовому плафоні LLDPE для більшого розсіювання світла.

2 АНАЛІЗ ПРОГРАМНИХ ТА АПАРАТНИХ ЗАСОБІВ

2.1 Огляд особливостей протоколів зв'язку для розумного світильника

Протоколи використовуються для зв'язку з компонентами системи один з одним. Можливих варіантів дуже багато, так що розглянемо які користуються найбільшою популярністю в Європі.

2.1.1 1-Wire

1-Wire для передачі даних використовує двонаправлену шину, яка в найпростішому випадку виглядає як двожильний провід. Іншими словами, при створенні мережі можна обійтися навіть дешевим телефонним кабелем. Один провід в даному випадку використовується для живлення і передачі даних, інший – для заземлення. Топологія мережі – загальна шина. Ведені пристрої буквально «нанизуються» на єдиний кабель, як намистини на нитку. Звичайно, мережа 1-Wire не зобов'язана виглядати незграбно: можна взяти за основу якісну «виту пару» або FireWire – кабель, а компоненти підключити через RJ – розетки. Власне, чим краще організована мережа, тим більшу протяжність вона може мати: в ідеальних умовах це значення досягає 300 метрів. Проте, головна «фішка» даного стандарту – дешевизна і невибагливість. Є у нього і негативна характеристика – низька відмовостійкість [9].

2.1.2 X10

Протокол зв'язку X10 – це стандарт, який був створений в 1975 році. Пік його популярності [9] вже позаду, але, тим не менш [9], він як і раніше активно використовується при проектуванні розумних будинків [9]. Секрет подібної [9] живучості – надзвичайна універсальність [9] при відносно низькій вартості [9]. На відміну від 1 – Wire, X10 не вимагає прокладки спеціального

кабелю: для передачі [9] сигналу використовується електропроводка будівлі [9]. Також передбачена можливість [9] використання трансиверів, які ловлять радіосигнал від бездротових [9] пристроїв, перетворюють його в [9] потрібний формат і передають [9] в електричну мережу [9]. Призначений для взаємодії з датчиками і пультами дистанційного керування [9].

2.1.3 KNX

Протокол зв'язку KNX – один з самих дорогих [9] протоколів зв'язку [9], який дуже популярний у Європі [9]. Стандарт характеризується [9] великою кількістю закладених в [9] нього функцій, а також складністю проектування і монтажу [9]. Як середовище для [9] передачі даних протокол KNX може використовувати [9] шину (кручену пару), електричну мережу або радіоканал [9]. Найчастіше застосовується [9] перший варіант, нерідко необхідні дроти [9] прокладаються [9] сумісно з силовими [9] кабелями ще на етапі будівництва. Стандарт передбачає [9] різні варіанти топології мережі [9]. Підсумкова система [9] повинна мати власне джерело живлення, але при цьому в ній може [9] бути відсутнім центральний контролер [9]. KNX дозволяє створювати децентралізовані [9] рішення [9], в яких сенсори і виконавчі модулі взаємодіють безпосередньо. Протокол підходить [9] для автоматизації великих будівель, в одну [9] мережу можна об'єднати до 58000 пристроїв [9]. Актуатори в даному випадку відрізняються [9] різноманітністю і дозволяють під зав'язку «упакувати» [9] будинок нетривіальним функціоналом [9].

2.1.4 Wi-Fi

Протокол зв'язку Wi-Fi використовується [9] майже в кожному будинку. Зазвичай цей стандарт застосовується [9] для зв'язку смартфона або планшета з уже готовою автоматизованою системою [9]. Мобільний пристрій завжди під рукою [9], а тому управляти будинком з його допомогою [9], часом

[9], зручніше [9], ніж використовувати для цієї мети [9] комп'ютер, настінну сенсорну панель, пульт дистанційного [9] керування або інтерпретатор мови [9]. Особливо це актуально [9] в тих випадках, коли для взаємодії [9] з системою передбачено спеціальний додаток, а не тільки веб-інтерфейс [9].

Іноді Wi-Fi застосовується для зв'язку з пристроями [9], які можуть функціонувати автономно [9], без допомоги «розумної» мережі [9]. Приклад такого гаджета – лампочка LIFX. Останнім часом [9] подібні пристрої набирають популярність [9], адже вони дозволяють долучитися [9] до технологій розумного будинку [9], не витрачаючи час і [9] сили на установку допоміжного обладнання [9]. Для складних систем автоматизації [9] Wi-Fi не підходить: модулі зв'язку цього стандарту дороги [9] і дуже ненажерливі [9], а демонстровані ними [9] швидкості в рамках розумного будинку просто не затребувані [9].

2.1.5 Insteon

Протокол зв'язку Insteon має [9] займає велику популярність в США [9], проте в Європу [9] він прийшов зовсім недавно [9]. Фактором [9], що стримує поширення, стала несумісність [9] початкової версії протоколу з [9] нашими електромережами: як і X10, Insteon використовує [9] проводку будівлі для передачі сигналів [9]. Однак, на відміну від застарілого конкурента [9], новий стандарт також підтримує [9] зв'язок по радіоканалу [9], причому провідна і бездротова [9] мережа функціонують одночасно [9], доповнюючи один [9] одного і істотно [9] підвищуючи надійність [9] автоматизованої системи [9]. Крім того, у Insteon немає проблем з чуйністю, властивих X10 [9].

2.1.6 ZigBee

Протокол зв'язку ZigBee [9] – використовується по радіоканалу або чудово вписується [9] в концепцію розумного будинку [9]. По-перше, стандарт

дозволяє [9] створювати датчики з низьким енергоспоживанням [9] і чудовою чуйністю: більшу частину [9] часу їх бездротові модулі знаходяться в сплячому режимі, але на пробудження останніх [9] витрачається всього 15 мілісекунд [9]. По-друге, ZigBee [9] підтримує комірчасту [9] топологію мережі, при якій [9] окремі компоненти [9] можуть виступати [9] в якості посередника, що передає сигнал від одного пристрою до іншого [9]. Подібна структура здатна до самоорганізації і самовідновлення [9], вихід з ладу одного – двох елементів, як правило [9], не призводить до серйозних наслідків [9]. Mesh – мережі також дозволяє істотно [9] збільшити зону покриття бездротової мережі [9], так що при грамотному підході ZigBee можна використовувати для автоматизації не тільки житлових будинків, але і великих робочих приміщень [9].

2.1.7 Z-Wave

Z-Wave – це запатентований [9] бездротовий протоколом зв'язку [9], розробленим для [9] домашньої автоматизації [9], зокрема контролю та управління в житлових і [9] комерційних об'єктах [9]. Технологія використовує малопотужні [9] та мініатюрні радіочастотні модулі [9], які вбудовуються в побутову електроніку [9] та різні пристрої [9], такі як освітлювальні прилади, опалювальні прилади [9], пристрої контролю доступу [9], розважальні системи та побутову техніку [9].

Z-Wave розроблена [9] спеціально для дистанційного керування [9]. На відміну від Wi-Fi та [9] інших IEEE 802.11 стандартів передачі даних, призначених в основному для великих потоків інформації [9], Z-Wave працює в діапазоні частот до 1 ГГц [9] та оптимізована для передачі простих керуючих команд з малими [9] затримками наприклад: увімкнути-вимкнути [9], змінити гучність [9], яскравість тощо [9]. Вибір низького радіочастотного діапазону для Z-Wave обумовлений малою кількістю [9] потенційних джерел перешкод на відміну від завантаженого діапазону 2,4 ГГц [9], в якому доводиться

вдаватися до заходів, що зменшують можливі перешкоди від різних побутових бездротових пристроїв [9], що працюють — Wi-Fi, ZigBee, Bluetooth [9].

Z-Wave призначений для створення недорогої та енергоефективної споживчої електроніки [9], у тому числі пристроїв на батарейках [9], таких як пульти дистанційного керування [9], датчики диму [9], температури [9], вологості [9], руху та інших датчиків безпеки [9].

2.1.8 Bluetooth

Bluetooth бере початок ще 1994 року. Ця технологія має солідний вік навіть від таких протоколів [10], як Z-Wave або ZigBee [10]. Спочатку Bluetooth використовувався для синхронізації даних [10] між мобільними телефонами, але стандарт [10] швидко став основною технологією обміну даними в [10] персональних обчислювальних мережах: смартфони, персональних комп'ютерів(ПК) [10], кишеньковий персональний комп'ютер(КПК) [10] та підключення різних периферійних пристроїв (гарнітури, бездротові клавіатури, миші, принтери і т.д.) [10].

Основним напрямком використання [10] Bluetooth має стати створення так званих персональних мереж (PAN, або private area networks) [10], які включають такі різнопланові пристрої [10], як мобільні телефони, КПК, MP3-плеєри, комп'ютери [10] і навіть мікрохвильові печі з холодильниками [10]. Можливість передачі голосу [10] дає змогу вбудовувати [10] інтерфейс Bluetooth [10] в бездротові телефони або [10], наприклад [10], бездротові гарнітури для телефонів [10]. Можливості застосування Bluetooth на практиці безмежні [10]: крім синхронізації КПК з настільним комп'ютером або під'єднування до низькошвидкісної периферії на [10] зразок клавіатур або мишей [10], інтерфейс дає змогу дуже просто і з невеликими витратами організувати домашню мережу [10]. Причому вузлами цієї мережі можуть [10] бути будь-які пристрої, що мають потребу в інформації або володіють необхідною інформацією [10].

Технологія Bluetooth працює [10] за принципом FHSS (Frequency-hopping spread spectrum) [10]. Передавач розбиває дані на пакети і передає їх за псевдовипадковим [10] алгоритмом стрибкоподібної перебудови частоти(, або шаблоном (pattern), складеному з 79 підчастот [10]. «Зрозуміти» один одного можуть тільки ті пристрої [10], які налаштовані на один [10] і той самий шаблон передачі — для сторонніх приладів передана [10] інформація буде звичайним шумом [10].

Далі наведено порівняльну таблицю 2.1 переваг та недоліків протоколів зв'язку.

Таблиця 2.1 – Порівняння протоколів зв'язку

Технологія	Wi-Fi	Bluetooth	ZigBee	Thread
Стандарт зв'язку	IEEE 802.11	IEEE 802.15.4	IEEE 802.15.4	IEEE 802.15.4
Дальність	До 100м	80м	100м	До 30м
Швидкість передачі	300+ Мбіт/с	До 3 Мбіт/с	250 Кбіт/с	250 Кбіт/с
Споживання енергії	Високе	Низьке	Низьке	Низьке
Частотний діапазон	2,4 ГГц	2,4 ГГц	2,4 ГГц	2,4 ГГц
Підтримка IP-технології	+	-	-	+
Топологія	«зірка»	«зірка»	«mesh»	«mesh»

2.2 Огляд особливостей середовищ програмування

На сьогодні існує величезна кількість середовищ програмування (Integrated Development Environment, IDE) для кодування модулів та плат. Розглянемо найпопулярніші з них.

2.2.1 Microsoft Visual Studio

Інтегроване середовище розробки Visual Studio – це багатофункціональна програма, яка підтримує багато аспектів розробки програмного забезпечення. Інтегроване середовище розробки Visual Studio – це стартовий майданчик для написання, налагодження та складання коду, а також подальшої публікації програм. Крім стандартного редактора і відладчика, які є в більшості середовищ IDE, Visual Studio включає компілятори, засоби автозавершення коду, графічні конструктори і багато інших функцій для поліпшення процесу розробки. [11].

Далі наведено найпопулярніші можливості Visual Studio:

- хвилясті лінії і швидкі дії;
- очищення коду;
- рефакторинг;
- IntelliSense;
- пошук в Visual Studio;
- Live Share;
- ієрархія викликів;
- CodeLens.

2.2.2 PlatformIO

PlatformIO IDE – це інтегроване середовище яка заснована мовою Python з відкритим вихідним кодом, зручна для розробки проектів інтернету речей (IoT – Internet of Things), кросплатформної розробки з єдиним (уніфікованим) відладчиком, здатним працювати у таких операційних системах як Windows, Mac та Linux. Підтримує практично всі популярні сьогодні мікроконтролери та готові платформи, такі як Arduino, ESP32, ESP8266. Містить велику кількість різних прикладів та бібліотек. Для своєї роботи вимагає встановлений на комп'ютері Python [12].

PlatformIO дозволяє проводити швидку розробку застосунків подібно до C / C ++, підтримує швидке завершення коду та інтелектуальний код Літера, недоступні в Arduino IDE [12].

2.2.3 Atmel Studio

Це інтегрована платформа розробки (IDP), призначена для проектування та налагодження програм для мікроконтролерів Atmel на базі процесорів ARM® Cortex™-M, а також мікроконтролерів AVR®. Вона є єдиною, простою у використанні середовище розробки, компонування та налагодження додатків, написаних мовою C/C++ чи асемблері [13].

2.2.4 Arduino IDE

Arduino IDE – це програмне середовище розробки за допомогою якого можна створювати свої скетчі для платформи Arduino. Скетч - це ім'я, яке Arduino використовує для програми. Це одиниця коду, яка завантажується та запускається на платі Arduino.

В Arduino мовою програмування для є C ++, а для полегшення написання програм використовується компілятор AVR-GCC [14].

Далі наведено порівняльну таблицю 2.2 різних середовищ програмування.

Таблиця 2.2 – Порівняння середовищ програмування на Arduino

IDE	Переваги	Недоліки
Arduino	Доступність та зручний для використання [15] і розуміння інтерфейс [15]. Програма сумісна з усіма версіями [15] операційних систем Windows [15]. Наявність необхідних для роботи інструментів Кілька [15] варіантів мов програмування [15]. Вбудований набір прикладів програм [15]. Функції збереження, експорту, перевірки [15], пошуку, заміни скетчів [15].	Відсутність кастомізації (теми, підсвічування синтаксису). Немає автодоповнення коду (як в сучасних IDE). не надто зручна організація Проєктів. Немає підтримки будь-яких плагінів. Незручний менеджер бібліотек. Не дуже зрозумілий builder, з яким досить важко шукати помилки в програмі.
PlatformIO	Підсвічування синтаксису та підказки. Вбудована підтримка GitHub дозволяє легко працювати з проектом з кількох різних комп'ютерів, використовуючи GitHub як «хмару».	Вищий поріг входження Більш складний менеджер бібліотек. У PІO у деяких випадках просто немає графічного інтерфейсу для інтерфейсу для налаштувань, лише командний рядок.

Продовження таблиці 2.2

IDE	Переваги	Недоліки
Atmel Studio	Підсвічування синтаксису CodeGuard (при початковому написанні змінної, функції, регістру пропонує вибір, що полегшує написання коду). Легко підключати зовнішні додаткові бібліотеки. Швидка заливка скомпільованого коду в мікроконтролер (режим налагодження). Генерація вихідних hex, sof, eep файлів, для можливості заливання через зовнішній програматор. Підтримка середовища розробки для arduino Підтримує усі контролери компанії Atmel.	Немає великої різноманітності вбудованих функцій. Напрямок портів налаштовується безпосередньо регістрами мікроконтролерів. Жахливий симулятор виконання коду.

2.3 Огляд особливостей середовищ програмування на Android

Зв'язка смартфона Android та Arduino надає величезні можливості для створення сучасних проєктів, спрощуючи віддалене програмування, управління та синхронізацію даних. Мобільні застосунки на смартфоні дають взаємодіяти з реальним фізичним світом з будь-якої точки землі за допомогою

звичних всім гаджетів. До уваги було обрано найпопулярніші середовища програмування для роботи з Android.

2.3.1 Eclipse

Eclipse – безкоштовне середовище розробки від некомерційної організації Eclipse Foundation. Сама програма – це основа, до якої підключаються різні модулі. Наприклад, Java Development Tools (для створення програм Java), C/C++ Development Tools (для розробки програм мовою C чи C++) тощо [16].

Завдяки активному розвитку, а також підтримці з боку компанії та сторонніх розробників, на даний момент в цій IDE має такі переваги:

- офіційна русифікація інтерфейсу та документації;
- чудова продуктивність на слабких машинах;
- велика кількість доповнень (наприклад, для роботи із сервером, базою даних тощо);
- можливість підключення модулів;
- можливість групової розробки.

Eclipse була дуже популярною кілька років тому і вважалася монополістом на ринку IDE для Android. Однак у зв'язку з виходом Android Studio, у 2014 році Google перестала підтримувати Eclipse як основне середовище для розробки додатків під Android [16].

2.3.2 IntelliJ IDEA

Розробкою цього середовища програмування займається компанія JetBrains. Як і Eclipse, це середовище розробки дає можливість створювати програми кількома мовами програмування. Крім цього, середовище має потужний двигун і великі можливості. Якщо розглядати програмування під

Android між IntelliJ IDEA та Eclipse, то перший варіант має певні переваги щодо свого конкурента [16]:

Якщо розглядати програмування під Android між IntelliJ IDEA та Eclipse, то перший варіант кращий, тому що це середовище має певні переваги:

- швидше налагодження значень;
- автозаповнення методів (також реалізовано в Eclipse, але поки що у тестовому варіанті);
- наявність рефакторингу (автоматичного підбору значень);
- зручний інтерфейс;
- відмінно підходить для програмування Java.

Головний недолік – наявність платної версії.

2.3.3 Android Studio

Android Studio - офіційне інтегроване середовище розробки на платформі Android, яке підтримує компанія Google. Вона заснована на інтегрованому середовищі розробки IntelliJ Idea, яке існує вже давно, але мала відносно невеликий рівень використання в спільноті Java, поки компанія Google не включила в неї свій модуль, що підключається, та не перейменувала цю версію в Android Studio [16].

Основні функції:

- застосувати зміни – що дозволяє вносити зміни в програму без необхідності перезавантажити програму;
- інтелектуальний редактор коду – отримуйте пропозиції щодо кращого коду під час введення;
- швидкий і багатофункціональний емулятор - дозволяє дуже швидко встановлювати та запускати тестові програми для різних пристроїв Android;

– шаблони коду та приклади програм – знайдіть шаблони з проектів, подібних до ваших.

Серед інших функцій Android Studio дозволяє підключатися до інших файлів проекту, таких як C/C++. Також є інтеграція з хмарою Google.

2.3.4 B4A

B4A (або Basic for Android) — B4A (або Basic for Android) - найменш відомий інструмент для розробки Android додатків від компанії Anywhere Software, що спеціалізується на концепції швидкої розробки додатків RAD (Rapid Application Development). Як випливає з назви, B4A - це IDE та інтерпретатор, який дозволяє розробникам створювати програми з використанням мови програмування BASIC. Незважаючи на те, що B4A - це проект, розроблений всього однією людиною, йому вдалося з'єднати в собі безліч корисних розширених функцій, таких як бездротове налагодження через Bluetooth та візуальний редактор [17].

В середовище розробки B4A можна робити практично все те саме, що за допомогою Java, але набагато швидше та з меншою кількістю шаблонів.

Далі наведено порівняльну таблицю 2.3 найпопулярніших IDE на Android.

Таблиця 2.3 – Порівняння IDE на Android

IDE для розробки Android	Мова програмування	Цільова ОС	Операційна система	Ліцензія
Android Studio	Java C C++ Kotlin	Android	Windows MacOS Linux	Безкоштовне програмне забезпечення

Продовження таблиці 2.3

IDE для розробки Android	Мова програмування	Цільова ОС	Операційна система	Ліцензія
Android Debug Bridge	Java C C++	Кросплатформена	Windows MacOS Linux	Безкоштовне програмне забезпечення
Eclipse	Java C C++ C# JavaScript Python	Android iOS Linux MacOS Windows	Будь-яка ОС, яка підтримує Java	Публічна ліцензія Eclipse
Visual Studio	C++ C C# Visual Basic PHP JavaScript	Кросплатформена Windows Android iOS	Windows MacOS Linux	Власний код Visual Studio є відкритим вихідним кодом MIT
IntelliJ IDEA	Java Scala Groovy Kotlin JavaScript TypeScript SQL	Будь-яка ОС, яка підтримує Java	Windows MacOS Linux	Запатентована, Community Edition є Ліцензія Apache 2.0
NetBeans	Java C C++ HTML PHP	Кросплатформена	Windows MacOS Linux Solaris	CDDL 1.0 and GPL2

Продовження таблиці 2.3

IDE для розробки Android	Мова програмування	Цільова ОС	Операційна система	Ліцензія
Komodo	Java JavaScript Python PHP HTML Ruby	Кросплатформена	Windows MacOS Linux	Запатентована, Komodo Edit є Mozilla Публічна ліцензія
B4A	Visual Basic and Visual Basic .Net	Кросплатформена	Windows MacOS Linux	Власне ПО (Безкоштовне)

2.4 Різновиди мікроконтролерів серії ESP

За допомогою модулів буде виконуватись реалізація розумного світильника. Існує близько півтора десятка модулів версій мікроконтролерів (МК) серії ESP і величезна кількість плат з ними. Розглянемо найпопулярніші з них.

2.4.1 Модуль ESP-01

Модуль ESP-01 є невеликою платою [18], чорного кольору, на якій розташовані два основних чіпа, це мікроконтролер ESP8266 та flash пам'ять на 1 Мб [18]. Поруч розташований кварцитовий резонатор [18] та надрукована антена. На платі встановлено два світлодіоди [18], червоний та блакитний [18]. Червоний світлодіод [18], світиться коли на модулі є живлення, а синій блимає при виконанні команд [18].

Має 8 розлучених контактів (VCC, GND, UTXD, URXD, CH_PD, GPIO0, GPIO2, GPIO6) і PCB-антену (друкований провідник на самій платі). Даний модуль наведено на рисунку 2.1 [18].

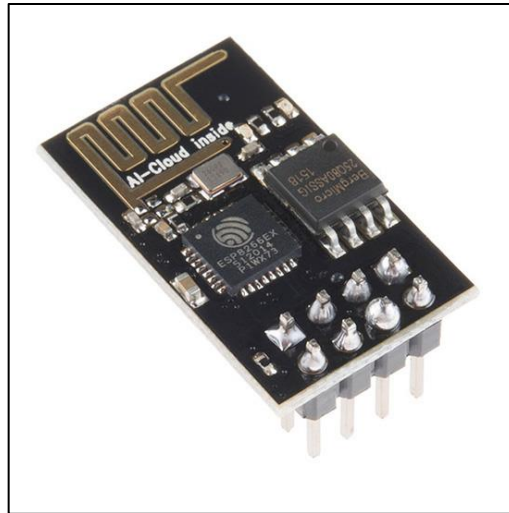


Рисунок 2.1 – Мікроконтролер ESP-01 [18]

2.4.2 Модуль ESP-03

ESP-03 – це недорогий [18] модуль для підключення різних [18] пристроїв до мережі через Wi-Fi. Плата побудована на базі чіпа ESP8266 компанії Espressif [18], що відрізняється енергоефективністю [18] та високим ступенем інтеграції, що [18] дозволяє використовувати мінімум [18] елементів обв'язування чіпа. ESP-03 [18] відрізняється від плати ESP-01 компактним розміром, використанням SMD-антени [18], а також наявністю розведення 7 пінів GPIO [18].

Чіп ESP8266 може [18] використовуватися двома способами [18]:

- як міст UART Wi-Fi [18], при цьому він керується зовнішнім пристроєм за допомогою AT-команд [18];
- як самостійний пристрій [18], управління здійснюється вбудованим у кристал [18] мікроконтролером Tensilica's L106 Diamond series [18].

Також модуль пропонує [18] кілька варіантів роботи з [18] Wi-Fi мережами, у тому числі може виступати як клієнт мережі [18], і сам створювати [18] точку доступу. Також на платі розведені [18] всі доступні роз'єми GPIO [18]. Даний модуль наведено на рисунку 2.2 [18].

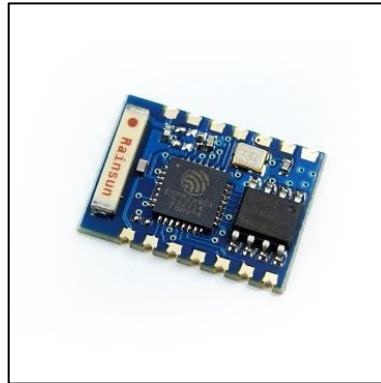


Рисунок 2.2 – Мікроконтролер ESP-03 [18]

2.4.3 Модуль ESP-07

Модифікація плати з урахуванням Wi-Fi [18] модуля ESP8266EX [18]. На модулі ESP-07 є мікросхема пам'яті 25Q04 ємністю 512Кб [18]. Оперативна пам'ять вбудована [18] в мікросхему та надає 80Кб статичної пам'яті програм та 32Кб [18] оперативної пам'яті інструкцій [18]. Для взаємодії модуля із зовнішніми [18] пристроями використовується 16 виходів із кроком 2 мм [18].

Даний модуль наведено на рисунку 2.3.



Рисунок 2.3 – Мікроконтролер ESP-07 [18]

2.4.4 Модуль ESP-12E NodeMCU

Плата має модуль ESP-12E [19], що містить мікросхему [19] ESP8266 з мікропроцесором [19] RISC [19] Tensilica Xtensa® 32-bit LX106 [19], який працює з регульованою тактовою [19] частотою від 80 до 160 МГц і підтримує RTOS [19]. На рисунку 2.4 зображено мікросхема та антена на платі ESP8266 [19].

Характеристики [19] NodeMCU ESP8266 [19]:

- 32-розрядний [19] LX106 від [19] Tensilica Xtensa [19];
- тактова частота [19] від 80 до 160 [19] МГц [19];
- 128 КБ вбудованої [19] оперативної пам'яті [19];
- 4 МБ зовнішньої [19] флеш-пам'яті [19];
- приймач [19] Wi-Fi 802.11b/g/n [19].

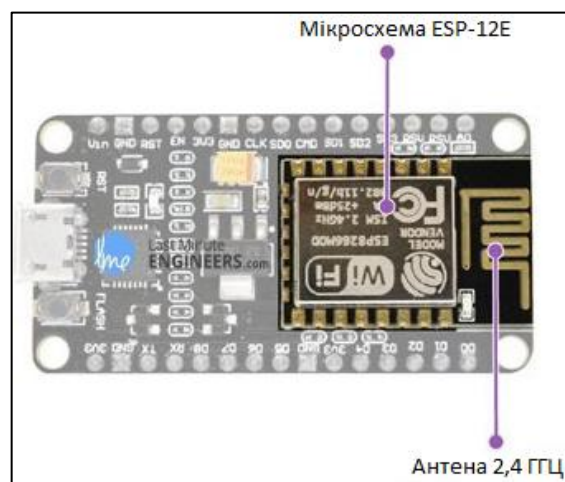


Рисунок 2.4 – ESP8266 [19] NodeMCU [19]

Також даний модуль має 128 [19] КБ ОЗУ та 4 МБ флеш-пам'яті (для зберігання програм та даних), [19] достатніх, щоб впоратися з [19] великими рядками, які становлять вебсторінки [19], даними в JSON/XML та на пристрої IoT [19].

ESP8266 містить вбудований приймач [19] Wi-Fi 802.11b/g/n HT40, тому він може не тільки [19] підключатися до мережі Wi-Fi та [19] взаємодіяти

з інтернетом [19], але і встановлювати власну мережу, дозволяючи іншим пристроям підключатися [19] безпосередньо до нього. Це робить ESP8266 NodeMCU [19] ще більш універсальним [19].

Оскільки діапазон робочої [19] напруги ESP8266 становить від 3 до 3,6 В, дана плата для [19] підтримки [19] постійної напруги на рівні 3,3 В поставляється з LDO стабілізатором напруги [19]. Він може надійно забезпечувати струм до 600 мА, чого має бути більш ніж достатньо, оскільки ESP8266 під час радіочастотних [19] передач споживає до 80 мА [19]. Вихід стабілізатора [19] також виводиться [19] на висновки на сторонах плати та позначений як 3V3 [19]. За допомогою цього можна [19] використовувати для подачі живлення [19] на зовнішні компоненти [19]. На рисунку 2.5 зображено елементи живлення ESP8266 [19].

Вимоги до живлення:

- робоча напруга: від 2,5 до 3,6 В [19];
- вбудований стабілізатор: 3,3 В, 600 мА [19];
- робочий струм: 80 мА [19];
- споживання в режимі сну: 20 мкА [19].

Живлення до NodeMCU ESP8266 подається через вбудований USB-роз'єм MicroB [19]. Як альтернатива, якщо у вас є стабілізоване джерело напруги 5 В, можна використовувати висновок VIN для безпосереднього живлення ESP8266 та його периферії [19].

NodeMCU ESP8266 має загалом 17 виходів GPIO [19], виведених на роз'єми з [19] обох сторін налагоджувальної [19] плати [19]. Ці висновки можуть використовуватись [19] для виконання різних периферійних завдань, у тому числі [19]:

- вхід АЦП – канал 10-розрядного АЦП [19];
- інтерфейс UART – інтерфейс UART використовується для завантаження коду послідовного зв'язку [19];

- виходи ШІМ – висновки ШІМ можуть використовуватися для регулювання яскравості [19] світлодіодів або [19] керування двигунами [19];
- інтерфейси SPI, I2C – інтерфейси [19] використовуються SPI та I2C для підключення [19] всіляких датчиків та периферійних пристроїв [19];
- інтерфейс I2S – інтерфейс I2S використовується [19] для цифрової передачі звуку [19].

Мультиплексовані виходи введення/виводу [19]

- 1 канал АЦП [19];
- 2 інтерфейси [19] UART [19];
- 4 [19] виходи [19] ШІМ [19];
- інтерфейси SPI [19], I2C [19] та I2S [19].

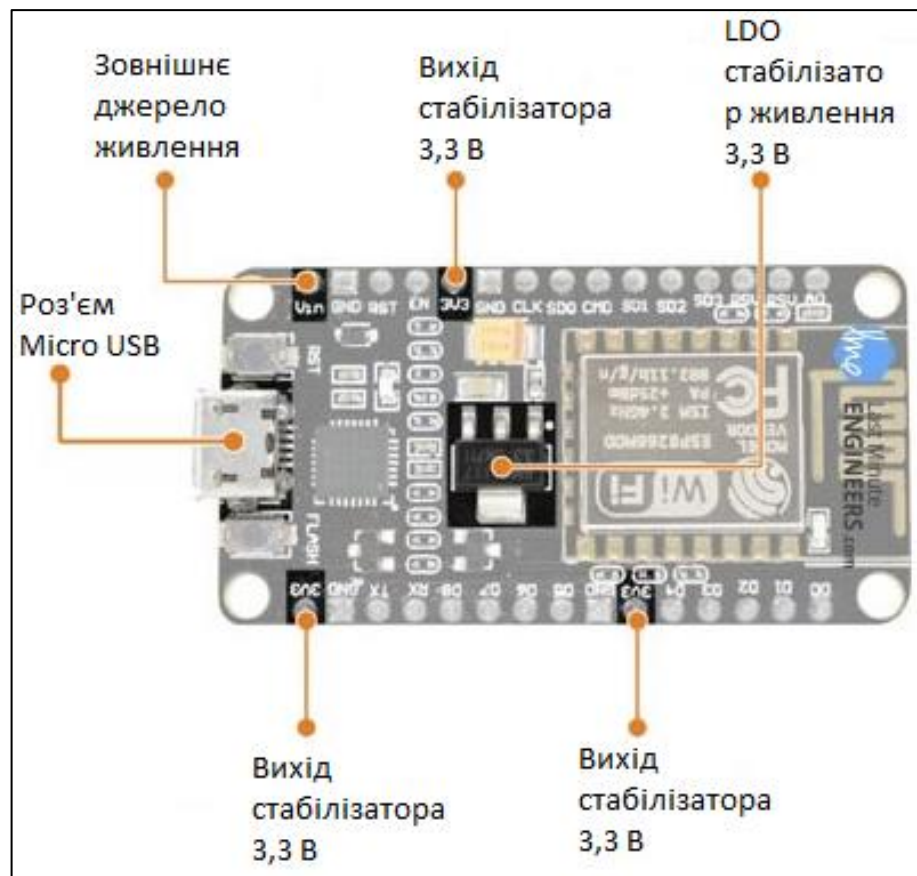


Рисунок 2.5 – Елементи живлення NodeMCU ESP8266 [19]

ESP8266 використовується [19] функція мультиплексування виходів [19] рисунок 2.6. Це означає, що один [19] вихід GPIO може [19] діяти як PWM/UART/SPI [19].



Рисунок 2.6 – Мультиплексовані виходи GPIO плати NodeMCU ESP8266 [19]

На платі ESP8266 NodeMCU є 2 кнопки рисунок 2.7. Кнопка RST, розташована у верхньому [19] лівому кутку, є кнопкою скидання [19], яка, звичайно ж [19], використовується для [19] скидання мікросхеми ESP8266 [19]. Інша кнопка, FLASH, у лівому нижньому кутку – це кнопка завантаження [19], яка використовується під час оновлення прошивки [19].

Кнопки та індикатори [19]

- RST - скидання чіпа [19] ESP8266 [19];
- FLASH – завантаження [19] нової програми [19];
- синій світлодіод - програмується [19] користувачем [19].

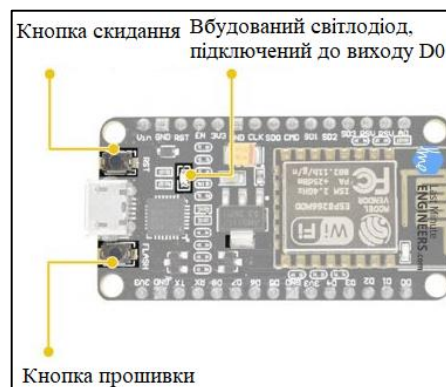


Рисунок 2.7 – Кнопки та світлодіод на платі ESP8266 NodeMCU [19]

На платі встановлено [19] контролер USB-UART [19] CP2102 від Silicon Labs [19] (рисунок 2.8), який перетворює USB сигнал на сигнал [19] послідовного [19] порту і дозволяє комп'ютеру [19] програмувати та взаємодіяти з [19] мікросхемою ESP8266 [19].

Послідовний зв'язок [19]

- USB-UART перетворювач CP2102 [19];
- швидкість зв'язку 4,5 Мбіт/с [19];
- підтримка керування потоком [19].



Рисунок 2.8 – Перетворювач USB↔TTL CP2102 [19]

Плата NodeMCU ESP8266 має [19] тільки 30 виходів [19].

Вихід живлення – на платі розташовано чотири виходи живлення, а саме: один вихід VIN та три виходи 3.3V [19]. Якщо у вас є стабілізоване джерело напруги 5 В, вихід VIN можна використовувати для безпосереднього живлення ESP8266 та його периферії [19]. Вихід 3.3V – це виходи вбудованого стабілізатора напруги [19]. Ці виходи можуть бути використані для подачі живлення на зовнішні компоненти [19].

GND – це виведення землі плати ESP8266 NodeMCU.

Вихід I2C використовуються для підключення всіх видів датчиків та периферійних пристроїв на шині I2C [19]. Підтримуються I2C Master та I2C

Slave [19]. Робота інтерфейсу I2C можна реалізувати програмно, а тактова частота становить максимум 100 кГц [19]. Слід зазначити, що тактова частота I2C повинна бути вищою за найнижчу тактову частоту з ведених пристроїв [19].

Виходи на GPIO ESP8266 NodeMCU має 17 виходів GPIO [19], які можна призначати програмно на різні функції, такі як: I2C, I2S, UART, PWM, дистанційне інфрачервоне управління, світлодіодний індикатор та кнопка [19]. Кожен увімкнений вихід GPIO може бути налаштований або на внутрішню підтяжку до землі або до шини живлення, або встановлений на високоімпедансний стан [19]. При конфігуруванні на вхід для генерування переривань процесора може бути налаштований на спрацьовування або по фронту, або за спадом [19].

Вихід ADC подає сигнал на наявний NodeMCU, вбудований 10-розрядний прецизійний аналого-цифровий перетворювач послідовного наближення (SAR ADC) [19]. За допомогою цього АЦП можуть бути реалізовані дві функції [19]: перевірка напруги живлення на виході VDD3P3 та перевірка вхідної напруги на виході TOUT (але не одночасно) [19].

Виходи UART ESP8266 NodeMCU має 2 інтерфейси UART, тобто UART0 та UART1, які забезпечують асинхронний зв'язок (RS232 та RS485) та можуть обмінюватися даними зі швидкістю до 4,5 Мбіт/с [19]. Для зв'язку можна використовувати UART0 (виходи TXD0, RXD0, RST0 та CTS0), який підтримує керування потоком [19]. UART1 (вихід TXD1) підтримує лише сигнал передачі, тому він зазвичай використовується для друку журналу подій [19].

Виходи SPI ESP8266 має два інтерфейси SPI (SPI та HSPI), що підтримують і ведений (slave) та ведучий (master) режими. Ці інтерфейси SPI також підтримують такі функції SP [19]:

- 4 режими синхронізації передачі SPI [19];
- до 80 МГц та тактові частоти, отримані розподілом 80 МГц [19];

- до 64 байт FIFO [19].

Виходи SDIO ESP8266 має захищений цифровий інтерфейс вводу/виводу (SDIO, Secure Digital Input/Output Interface), який використовується для прямого підключення карток SD. Підтримуються 4-бітний 25 МГц SDIO v1.1 та 4-бітний 50 МГц SDIO v2.0 [19].

Виходи PWM На платі є 4 канали широтно-імпульсної модуляції (PWM) [19]. Вихід ШІМ може бути реалізований програмно та використаний для управління двигунами та світлодіодами [19]. Частотний діапазон ШІМ регулюється від 1000 до 10000 мкс, тобто від 100 Гц до 1 кГц [19].

Виходи керування використовуються для керування ESP8266. Ці виходи включають вихід включення мікросхеми EN, вихід скидання RST та вихід пробудження WAKE [19].

- вихід EN – мікросхема ESP8266 включена, коли виведення EN подається високий логічний рівень. За низького логічного рівня мікросхема працює на мінімальній потужності [19];
- вихід RST використовується для скидання мікросхем ESP8266 [19];
- вихід WAKE використовується для виведення чіпа із глибокого сну [19].

Нижче показано розпінування налагоджувальної плати на рисунку 2.9.

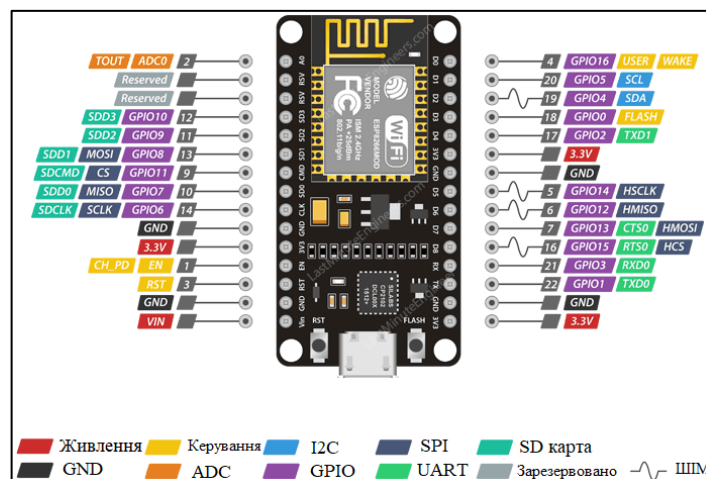


Рисунок 2.9 – Розпінування ESP8266 NodeMCU [19]

2.4.5 ESP8266 Witty Cloud

ESP8266 Witty Cloud – китайський мікроконтролер (МК) від виробника Espressif з підтримкою WiFi-інтерфейсу. Управляти всім цим можна не тільки з браузера, а й з застосунків на Android / iOS / Desktop [20]. Даний модуль наведено на рисунку 2.10.

Witty Cloud складається з двох окремих плат це зображено на рисунку 2.11. На верхній розташований сам Wi-Fi модуль ESP8266, стабілізатор на 3.3В, RGB світлодіод, фоторезистор і програмована кнопка. На нижній – USB–UART конвертер на чіпі CH340 з обов'язкою. За допомогою вибору програм (нижньої плати) можна прошивати і інші плати з серії ESP – 12, встановивши їх на сумісний по терморегулятори адаптер, і припаявши на нього стабілізатор [20]. Піни модуля зображено на рисунку 2.12.

Характеристики:

- живлення модуля – 5 В / 220 мА;
- режим Wi-Fi: 802.11 b / g / n, 2.4 ГГц;
- шифрування: WPA / WPA2;
- доступні мережеві протоколи: IPv4, TCP, UDP, HTTP, FTP;
- GPIO: 11;
- інтерфейс: UART, HSPI, I2C, I2S;
- внутрішня пам'ять: 4 Мб;
- USB–UART міст: CH340
- розміри: 33×30×18 мм.

Режим роботи:

- клієнт на STA;
- точка доступу на AP;
- клієнт + точка доступу (STA + AP).



Рисунок 2.10 – Модуль ESP8266 Witty Cloud [20]

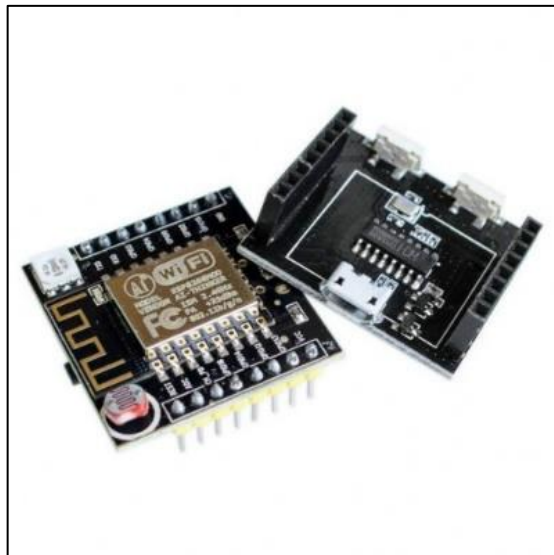


Рисунок 2.11 – Верхня та нижня частина ESP8266 [20]

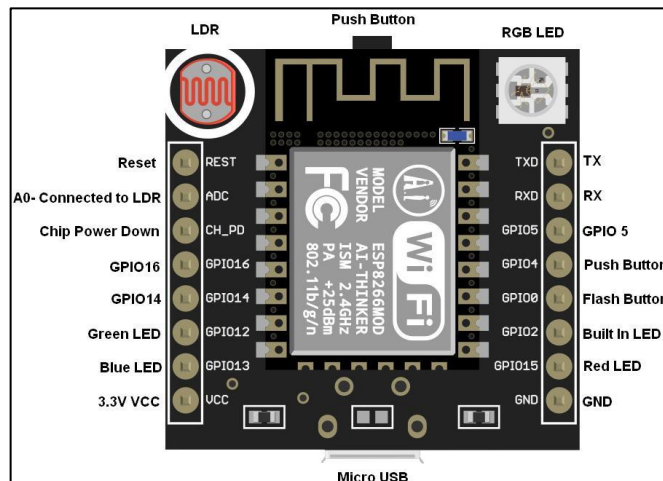


Рисунок 2.12 – Піни модуля ESP8266 Witty Cloud [20]

Далі наведено порівняльну таблицю 2.4 с ESP8266 та ESP32.

Таблиця 2.4 – Порівняння ESP модулів

Технічні характеристики	ESP8266	ESP32
Мікроконтролер	Xtensa Single-Core 32-bit L106	Xtensa Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Так, HT20	Так, HT40
Bluetooth	Немає	Bluetooth 4.2 та нижче
Частота	80 Гц	160 Гц
Пам'ять	160 Кб	512 Кб
Прошивка	SPI до 16 Мб	SPI
GPIO	17	36
Hardware / Software PWM	Немає / 8 Канали	1 / 16 Канали
SPI / I2C / UART	2 / 1 / 2 / 2	4 / 2 / 2 / 2
ADC	10-біт	12-біт
CAN	Не застосовується	1
Ethernet MAC Interface	Не застосовується	1
Модуль сенсорного датчика	Не застосовується	Так
Температурний сенсор	Не застосовується	Так
Робоча температура	-40° C – 125° C	-40° C – 125° C

2.5 Висновки

В даному розділі було розглянуто різні середовища розробки їх переваги та недоліки, Android застосунки, які працюють з Arduino, а також різновиди мікроконтролерів серії ESP.

Засобом зв'язку було обрано протокол Wi-Fi, тому що це найдоступніший та малозатратний вид зв'язку. В наш час вже в кожному будинку є Інтернет, тому за допомогою Wi-Fi можна легко робити різні маніпулювання в розумному світильнику через смартфон або планшет.

Для реалізації проєкту було обрано Arduino IDE. Хоча воно і має ряд недоліків, але є доступним та легким у використанні. Для людей, які тільки почали займатися кодуванням та прошиванням модулів, Arduino IDE буде найкращим вибором.

Для створення мобільного застосунку було обрано інструмент розробки B4A. B4A включає візуальний конструктор, який спрощує процес створення інтерфейсів користувача для телефонів та планшетів з різними розмірами екрана. Скомпільовані програми можна протестувати в емуляторах AVD Manager або на реальних пристроях Android за допомогою Android Debug Bridge та B4A Bridge.

Під обрану задачу було обрано мікроконтролер ESP32, а саме плата на його основі – NodeMcu ESP8266. Це забезпечить наявність можливості для керування розумним світильником за допомогою Wi-Fi, оскільки даний модуль вбудований до цієї плати. Вона є досить розповсюдженою, що забезпечує актуальність пристрою.

3 ОПИС ПРАКТИЧНОЇ РЕАЛІЗАЦІЇ РОЗУМНОГО СВІТИЛЬНИКА

3.1 Світловий потік

Кожен світильник має [21] величину споживаної потужності, однак показник споживання електроенергії [21] не настільки важливий, адже інтенсивність освітлення визначається світловим випромінюванням [21].

Світловий потік ламп вимірюється в люменах (лм). Люмен - це одиниця виміру всього сприйманого світлового потоку, що випускається джерелом світла [21].

Знаючи необхідну величину світлового потоку [21] можна з легкістю визначити кількість необхідних [21] світлодіодних ламп для освітлення приміщення [21]. Параметр світлового потоку [21], наприклад, для виробничих приміщень та учнівських класів буде відрізнятися [21].

Іншими словами, чим вищий у світильника світловий потік [21], тим яскравіше він світить [21].

Світловий потік – це кількісне значення «світлової» потужності у всьому потоці випромінювання. Він залежить від спектральної чутливості ока середньостатистичної людини [22]. Людське око на довжині хвилі монохроматичного випромінювання, що дорівнює 555,016 нм, є найбільш чутливим. Таким чином, одна кандела визначається на найбільш чутливій для людського ока частоті монохроматичним випромінюванням джерела світла, що має потужність випромінювання $1/683$ Вт на кожний стерadian, тобто 1,46 мВт/ст [22].

3.1.1 Освітленість

Освітленість – це світлова величина, що дорівнює відношенню світлового потоку, що падає на малу ділянку поверхні, до його площі.

Погана освітленість приміщень, робочого місця або кімнати у квартирі негативно впливає на здоров'я людини, знижує концентрацію уваги,

працездатність, з'являється дратівливість. Дуже яскраве світло також є подразником та не дає нічого позитивного для людини.

Як наслідок, світло впливає на психологічний стан людини: якщо світла недостатньо – він відчуває пригніченість, знижену працездатність, сонливість; якщо світло надто яскраве, воно сприяє збудженню, підключенню додаткових ресурсів організму, викликаючи їх підвищений знос.

Якщо ж світло підібрано правильно, то завдяки покращенню освітленості продуктивність на робочому місці може бути підвищена.

Тому необхідно забезпечити нормальну освітленість приміщень, що регламентується певним стандартом БНіП. Для цього потрібна проста установка відповідних ламп освітлення для кожного приміщення.

Далі наведено таблицю 3.1 з даними освітленості у різних випадках.

Таблиця 3.1 – Приклади освітленості[23]

Опис	Освітлення лк
Поза атмосферою на середній відстані Землі від Сонця [23]	135 000
Найбільша сонячна освітленість при чистому небі [23]	100 000
Звичайна освітленість влітку в середніх широтах опівдні [23]	17 000
У хмарну погоду влітку опівдні [23]	12 000
При кінозйомці у студії [23]	10 000
Звичайна освітленість узимку в середніх широтах [23]	5000
На футбольному стадіоні (штучне освітлення) [23]	12000
На відкритому місці у похмурий день [23]	1000-2000

Продовження таблиці 3.1

Опис	Освітлення лк
Схід та захід Сонця у ясну погоду [23]	1000
У світлій кімнаті поблизу вікна [23]	400-500
На екрані кінотеатру [23]	85-120
Необхідне для читання [23]	30-50
У морі на глибині 50-60 м [23]	до 20
Вночі у повний місяць [23]	0,2
У безмісячну ніч [23]	0,001 – 0,002
У безмісячну ніч у хмарність [23]	до 0,0002

3.1.2 Вимірювання світлового потоку

Для вимірювання світлового потоку виконується за допомогою спеціального приладу. Важкість виміру полягає в тому, що потрібно вимірювання виконується у всіх напрямках. Тому для спрощення виміру будемо визначати світловий потік за допомогою фоторезистора та визначати його опір.

Фоторезистор – це резистор, опір якого залежить від кількості світла, що падає на нього. Фоторезистори виготовляються з напівпровідникових матеріалів, завдяки цьому вони чутливі до властивостей світла [24]. Найбільш популярний матеріал для виготовлення фоторезисторів – це сульфід кадмію. Фоторезистори працюють за принципом «фотопровідності» [24], тобто коли світло починає падати на фоторезистор, його провідність збільшується [24], відповідно, його опір зменшується. А коли кількість світла, що падає на фоторезистор, зменшується, його опір збільшується [24].

Далі наведено формулу (3.1) для визначення опору фоторезистора:

$$R2 = \frac{R1}{\frac{Vdd}{ADCR2} - 1}, \quad (3.1)$$

де $R1$ – опір верхнього плеча, $R1 = 10000$;

$R2$ – опір нижнього плеча;

Vdd – найбільше можливе значення с аналого-цифрового перетворювача(АЦП), $Vdd = 1023$.

$ADCR2$ розраховується за формулою (3.2):

$$ADCR2 = \frac{Vdd}{\frac{R1}{R2} + 1}, \quad (3.2)$$

де $ADCR2$ – перетворення аналогового значення в цифрове.

За допомогою формули можна дізнатися опір фоторезистора, при відомому опорі верхнього плеча $R1$, напрузі живлення Vdd та падінні напруги на датчику $ADCR2$.

3.2 Аналіз вимог

Головна ідея розумного світильника – це покращення комфорту та атмосфери у приміщенні, скорочення витрат на час керування розумним світильником та електроенергії. За допомогою цього проєкту користувачі зможуть керувати світильником, за допомогою сенсорної кнопки зі смартфона або планшету по мережі Wi-Fi.

Далі наведено наступні функціональні вимоги:

- включення/виключення світильника;
- зміна ефектів світильника;
- зміна швидкості, яскравості та масштаб ефектів світильника;
- зміна яскравості від отриманих даних світлового потоку.

Нефункціональними вимогами виступають не висока ціна на саму систему. Також щоб керувати світильником за допомогою смартфона або планшета потрібно мати постійне підключення в одній локальній мережі з світильником.

3.3 Архітектура системи

В загальному вигляді розумний світильник складається з: світодіодної матриці 16x16, пластмасового плафона LLDPE, сенсорної кнопки TTP223, блока живлення 5В-3А, комплекту дротів, плати NodeMCU ESP8266, для дистанційного керування використовується мобільний застосунок, який з'єднується по протоколу зв'язку Wi-Fi.

Використання мобільного застосунку – це найкращий спосіб для дистанційного керування світильником, що дозволяє реалізовувати максимум його можливостей.

На рисунку 3.1 зображена схема мережі розумного світильника. Для керування світильником із застосунка, девайс повинен перебувати в одній локальній мережі з світильником. Для цього необхідно задати IP-адресу та порт управління світильником.

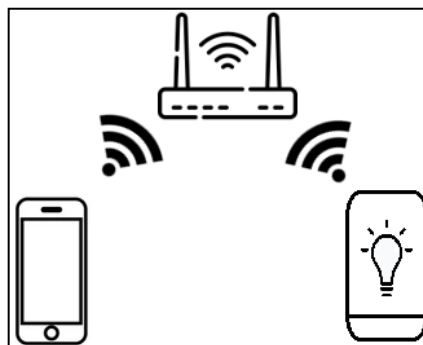


Рисунок 3.1 – Схема роботи мережі проекту

На рисунку 3.2 зображено діаграму активності взаємодії користувача з розумним світильником.

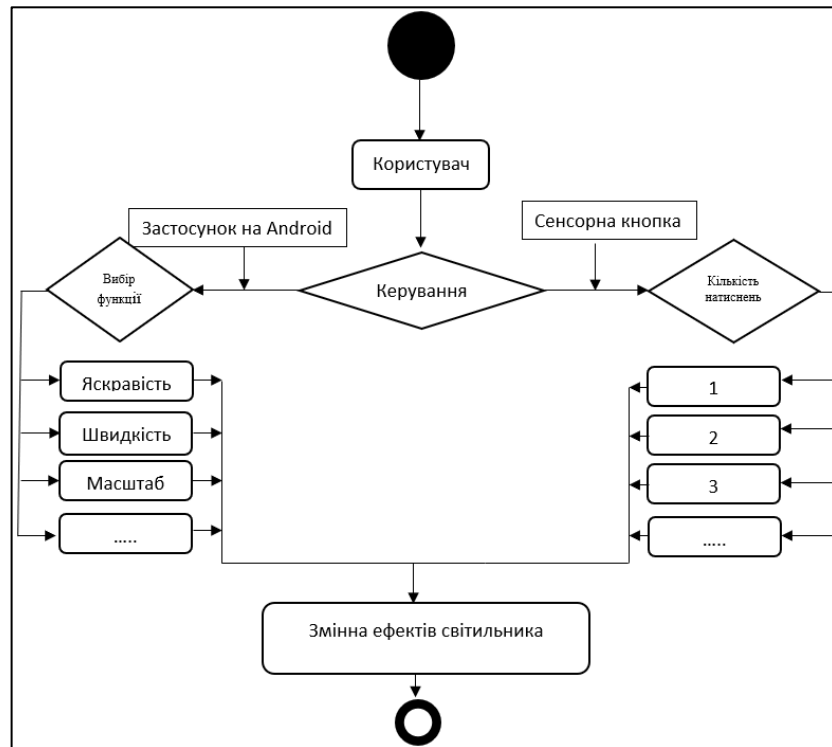


Рисунок 3.2 – Діаграма активності роботи розумного світильника

На рисунку 3.3 зображена схема збору розумного світильника, де джерело живлення VCC – позитивний(червоний), GND – негативний контакт(синій), а DIN – шина для з'єднання двох контролерів (зелений та фіолетовий).

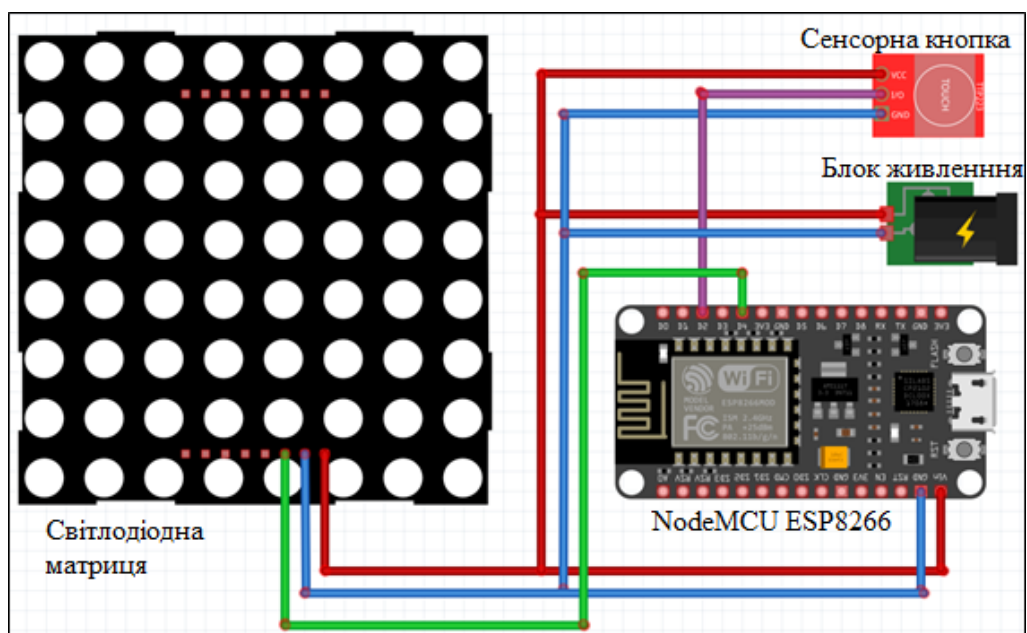


Рисунок 3.3 – Схема збору розумного світильника

3.4 Алгоритм роботи розумного світильника

Розумний світильник надає користувачеві можливість обрати певний ефект, це здійснюється за допомогою сенсорної кнопки або мобільного застосунку. Користувач може змінювати яскравість, швидкість та масштаб поточного ефекту, здійснювати перехід на наступний або попередній ефект, а також включення або виключення світильника (рисунки 3.4 – 3.5), а також коригування яскравості світильника на основі даних світлового потоку, які залежать від часу доби.

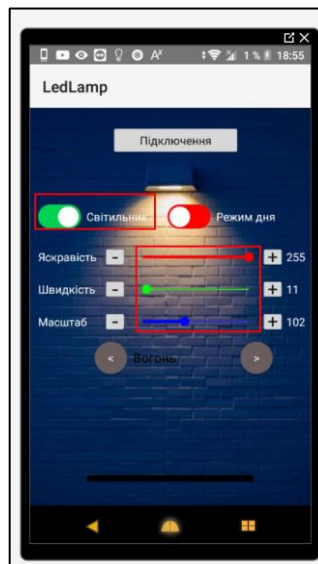


Рисунок 3.4 – Приклад роботи розумного світильника



Рисунок 3.5 – Приклад роботи розумного світильника

3.5 Опис особливостей програмної реалізації

Структуру файлів проєкту та їх опис наведено в таблиці 3.2. В таблиці 3.3 наведено список використовуваних бібліотек в Arduino, які потрібні для роботи з платою.

Таблиця 3.2 – Структура файлів проєкту на Arduino

Файл	Призначення
LedLamp.ino	Головний файл проєкту
efTic.ino	Файл містить список ефектів
effects.ino	Файл містить код ефектів
button.ino	Файл містить налаштування сенсорної кнопки
parsing.ino	Файл містить логіку отримання команд з мобільного застосунку та їх обробку

Таблиця 3.3 – Бібліотеки проєкту на Arduino

Бібліотека	Призначення
FastLED.h	Використовується для створення різних ефектів на адресній світлодіодній стрічці WS2812B
ESP8266WiFi.h	Використовується для з'єднання з модулем ESP8266 по мережі Wi-Fi
DNSServer.h	Використовується для реалізації простого DNS-сервера, який можна використовувати як у стаціонарному режимі (STA), так і в режимі точки доступу (AP)

Продовження таблиці 3.3

Бібліотека	Призначення
ESP8266WebServer.h	Використовується для отримання доступу методів, які потрібні для налаштувати сервера та обробляти вхідні HTTP-запити
WiFiManager.h	Використовується для керування Wi-Fi з'єднанням за допомогою порталу авторизації. При невдалому підключенні до Wi-Fi ця бібліотека запускає режим точки доступу , а потім портал авторизації, де можна ввести дані своєї Wi-Fi мережі
WiFiUdp.h	Використовується для програмування UDP-функціоналу.
EEPROM.h	Використовується для читання або запису даних на flash пам'ять
NTPClient.h	Використовується для отримання дати та часу з NTP-сервера
GyverButton.h	Використовується для багатофункціонального відпрацювання натискання кнопки

У таблиці 3.4 наведено створені методи для роботи розумного світильника.

Таблиця 3.4 – Методи проєкту на Arduino

Метод	Опис
buttonTick()	Управління розумним світильником за допомогою сенсорної кнопки
ef_Tick()	Вибір ефекту світіння
changePower()	Змінення стану світильника (включення/виключення)
parseUpdate()	Отримання команд з мобільного застосунку та їх обробка
sendCurrent()	Відправка поточного стану та параметрів світильника (яскравість, масштаб, швидкість)
sendIP()	Відправка IP адреси на Android застосунок
saveEEPROM()	Запис до енергонезалежної пам'яті поточного ефекту світильника

На рисунках 3.6 та 3.7 зображена блок-схема отримання команд з мобільного застосунку та їх обробка.

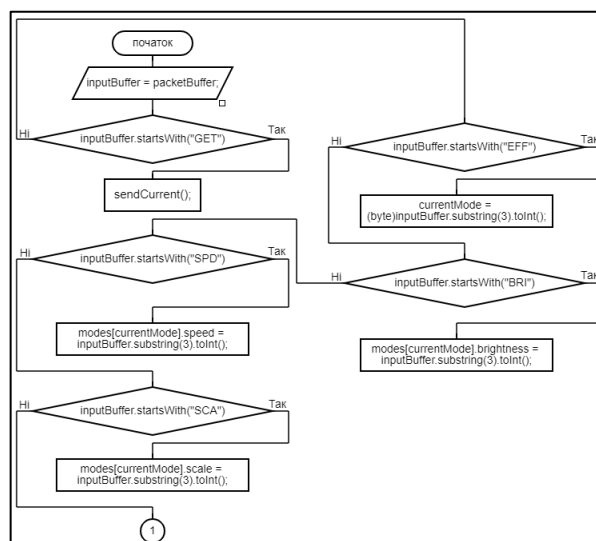


Рисунок 3.6 – Блок-схема отримання команд

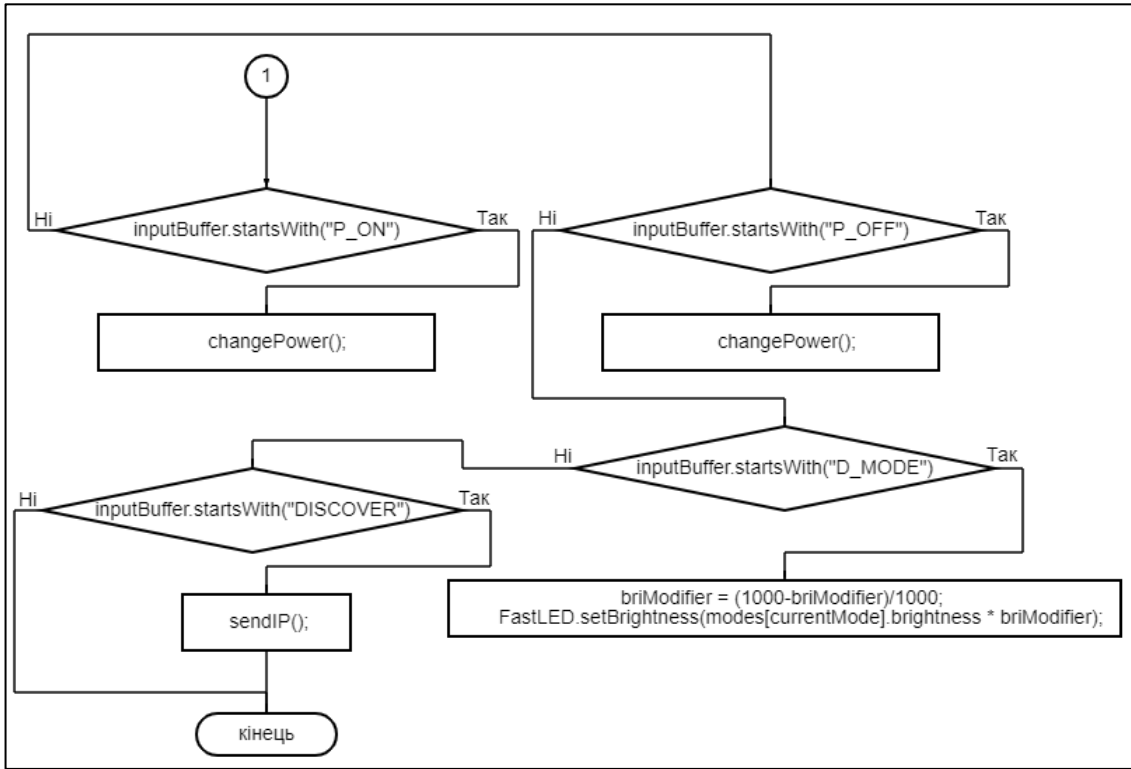


Рисунок 3.7 – Продовження блок-схеми

3.6 Висновки

В даному розділі було розглянуто архітектура роботи розумного світильника з користувачем та умови її використання. Було зроблено огляд файлової структури та бібліотек в Arduino, які використовуються для роботи з модулем ESP8266.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення і умови використання Arduino IDE

Програма призначена для керування розумним світильником. За допомогою Arduino будемо кодувати та прошивати модулі ESP8266, а за допомогою розробленого застосунка на android вже буде виконуватись дистанційне керування.

Системні вимоги до Arduino:

- операційна система: Windows XP / Vista / 7/8 / 8.1 / 10;
- обсяг пам'яті: потрібно 512 МБ ОЗУ;
- місце на жорсткому диску 1 ГБ вільного місця;
- процесор: Intel Dual Core або новіше.

Arduino це безкоштовний застосунок, тому його можна завантажити на офіційному сайті. Після встановлення Arduino на комп'ютер потрібно завантажити плату ESP. Для цього потрібно перейти на вкладку: «Файл → Настройки → Дополнительные ссылки» для Менеджера плат та ввести «https://arduino.esp8266.com/stable/package_esp8266com_index.json» (зображено на рисунку 4.1). Потім потрібно перейти в «Инструменты» → «Плата» → «Менеджер Плат» в поле пошуку вписати ESP та встановити плату (рисунок 4.2).

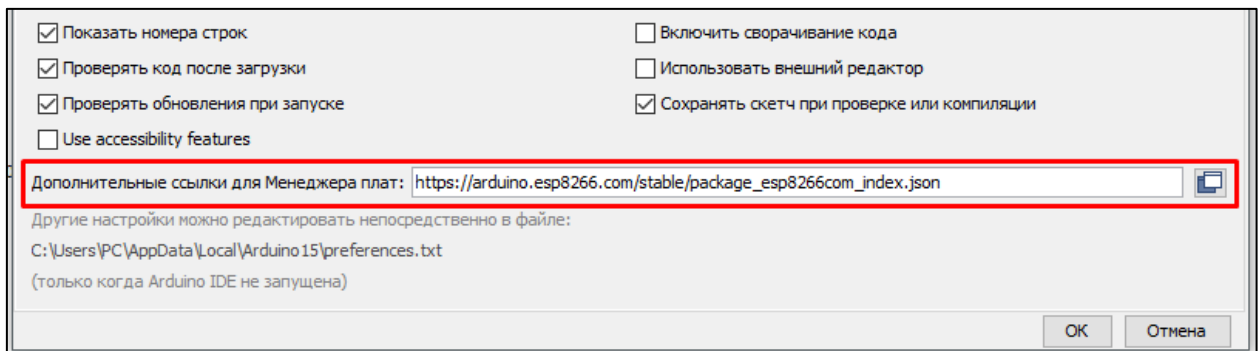


Рисунок 4.1 – Установка бібліотеки ESP

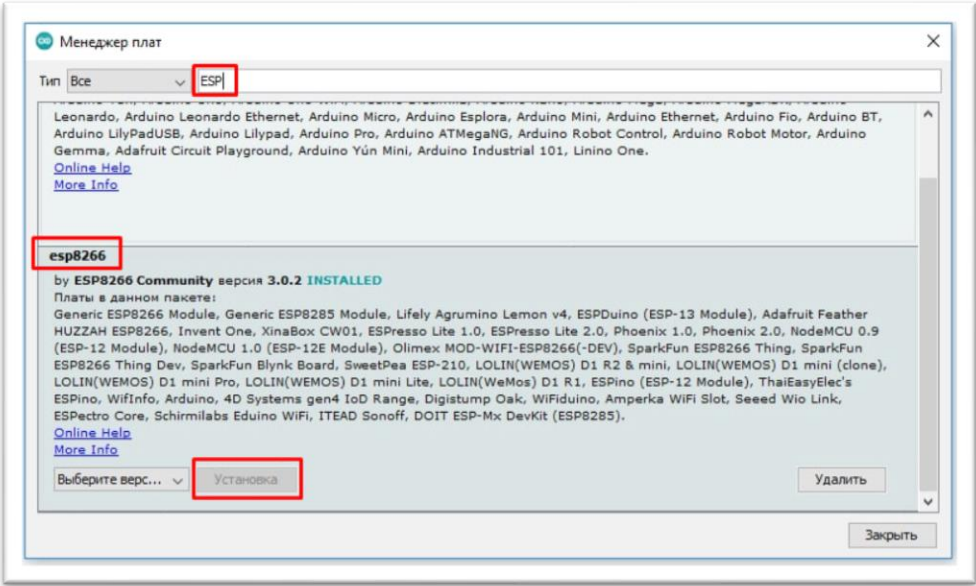


Рисунок 4.2 – Установка Менеджер Плат

Далі необхідно обрати налаштування в Arduino IDE для модуля ESP8266 (зображено на рисунку 4.3), для цього потрібно вибрати вкладку «Інструменти» та встановити наступні налаштування:

- плата: NodeMCU 1.0 (ESP-12E Module);
- upload Speed: 115200;
- erase Flash: Only Sketch.

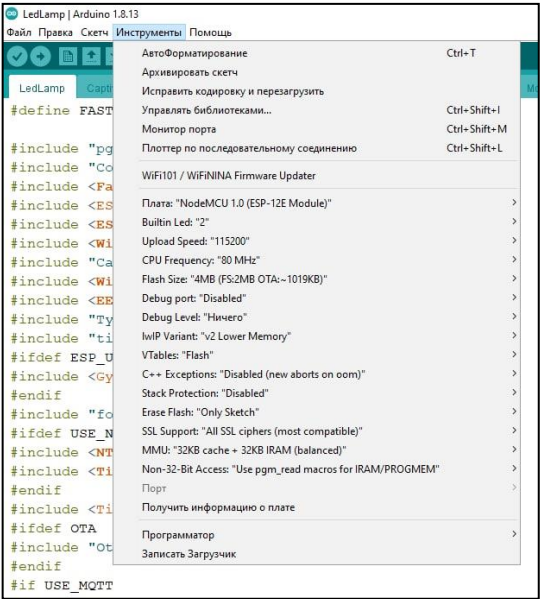


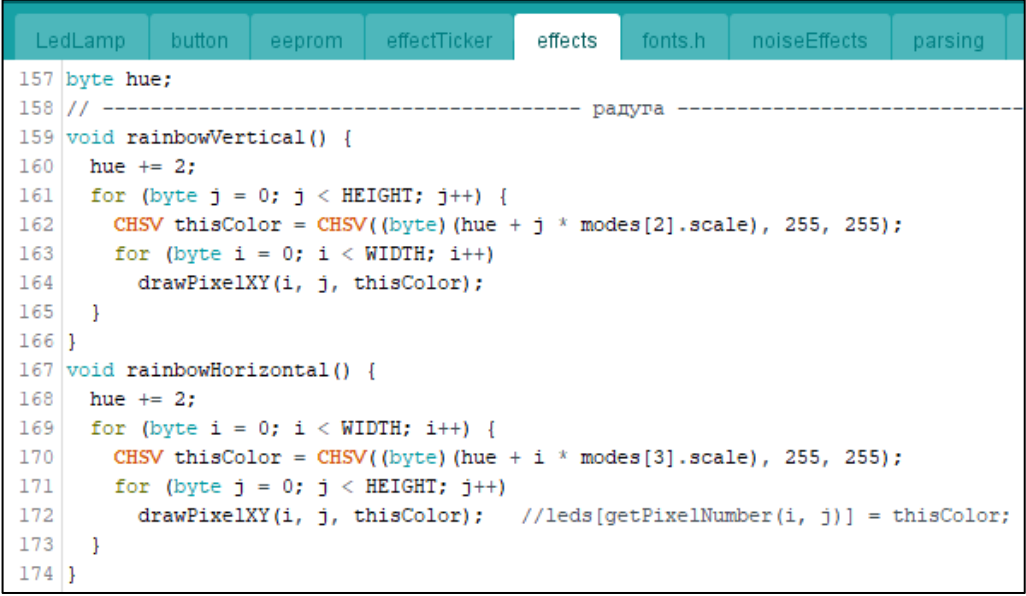
Рисунок 4.3 – Налаштування завантаження скетчу

4.2 Налаштування скетчу

Після встановлення Arduino IDE на комп'ютер далі можна відкривати головний файл проєкту LedLamp.ino для редагування або додавання змін скетчу. Скетч – це назва, яку Arduino використовує для програми. Це одиниця коду, яка завантажується та запускається на платі Arduino.

Для додавання або редагування ефекту потрібно робити це в 3 файлах:

- у файлі effects.ino – додається програмний код самого ефекту (рисунк 4.4);
- у файлі effectTicker.ino – вказуються назви ефектів "EFF_.....", задаються порядкові номери та встановлюється процедура виклику ефекту на відповідний їй "EFF_....." (рисунк 4.5);
- у файлі noiseEffects.ino – задається колір, швидкість та масштабування ефекту (рисунк 4.6).



```

157 byte hue;
158 // ----- радуга -----
159 void rainbowVertical() {
160   hue += 2;
161   for (byte j = 0; j < HEIGHT; j++) {
162     CHSV thisColor = CHSV((byte)(hue + j * modes[2].scale), 255, 255);
163     for (byte i = 0; i < WIDTH; i++)
164       drawPixelXY(i, j, thisColor);
165   }
166 }
167 void rainbowHorizontal() {
168   hue += 2;
169   for (byte i = 0; i < WIDTH; i++) {
170     CHSV thisColor = CHSV((byte)(hue + i * modes[3].scale), 255, 255);
171     for (byte j = 0; j < HEIGHT; j++)
172       drawPixelXY(i, j, thisColor); //leds[getPixelNumber(i, j)] = thisColor;
173   }
174 }
  
```

Рисунок 4.4 – Код ефекту

```

1 uint32_t effTimer;
2
3 void effectsTick() {
4   if (!dawnFlag) {
5     if (ONflag && millis() - effTimer >= ((currentMode < 5 || currentMode > 13) ? modes[currentMode].speed : 50) ) {
6       effTimer = millis();
7       switch (currentMode) {
8         case 0: sparklesRoutine();
9           break;
10        case 1: fireRoutine();
11          break;
12        case 2: rainbowVertical();
13          break;
14        case 3: rainbowHorizontal();
15          break;
16        case 4: colorsRoutine();
17          break;
18        case 5: madnessNoise();
19          break;
20        case 6: cloudNoise();
21          break;

```

Рисунок 4.5 – Назва та виклик ефектів

```

47 }
48 ihue += 1;
49 }
50 void rainbowNoise() {
51   if (loadingFlag) {
52     loadingFlag = false;
53     currentPalette = RainbowColors_p;
54     scale = modes[9].scale;
55     speed = modes[9].speed;
56     colorLoop = 1;
57   }
58   fillNoiseLED();
59 }

```

Рисунок 4.6 – Параметри ефектів

Для створення точки доступу світильника та підключення до Wi-Fi мережі вказуються у файлі LedLamp.ino (рисунок 4.7).

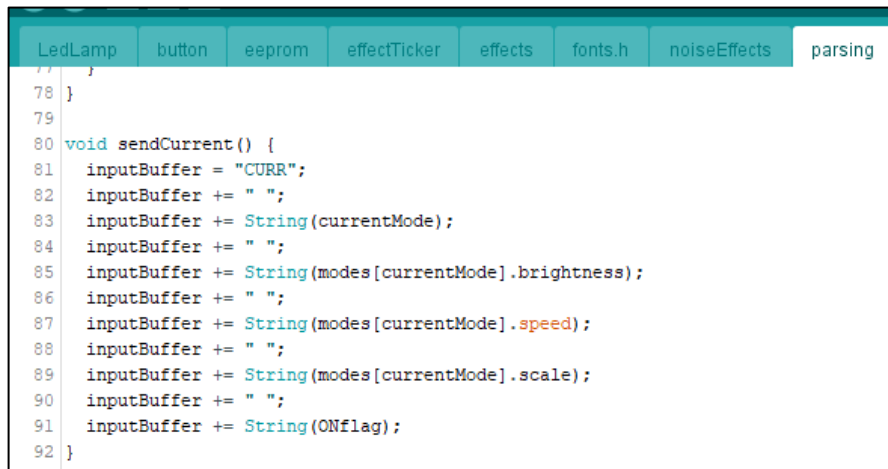
```

74 // ----- AP (точка доступу) -----
75 #define AP_SSID "LedLamp"
76 #define AP_PASS "12345678"
77 #define AP_PORT 8888
78
79 // ----- Менеджер WiFi -----
80 #define AC_SSID "TP-LINK_2DE6"
81 #define AC_PASS "76396717"

```

Рисунок 4.7 – Встановлення параметрів мережі

У файлі `parsing.ino` описується логіка отримання команд з мобільного застосунку, їх обробка та виведення результату на світильник (рисунок 4.8).



```

78 }
79
80 void sendCurrent() {
81   inputBuffer = "CURR";
82   inputBuffer += " ";
83   inputBuffer += String(currentMode);
84   inputBuffer += " ";
85   inputBuffer += String(modes[currentMode].brightness);
86   inputBuffer += " ";
87   inputBuffer += String(modes[currentMode].speed);
88   inputBuffer += " ";
89   inputBuffer += String(modes[currentMode].scale);
90   inputBuffer += " ";
91   inputBuffer += String(ONflag);
92 }

```

Рисунок 4.8 – Поточний стан світильника

У файлі `button.ino` описується налаштування сенсорної кнопки (рисунок 4.9).



```

1  boolean brightDirection;
2
3  void buttonTick() {
4    touch.tick();
5    if (touch.isSingle()) {
6      if (dawnFlag) {
7        manualOff = true;
8        dawnFlag = false;
9        loadingFlag = true;
10       FastLED.setBrightness(modes[currentMode].brightness);
11       changePower();
12     } else {
13       if (ONflag) {
14         ONflag = false;
15         changePower();
16       } else {
17         ONflag = true;
18         changePower();
19       }
20       sendSettings_flag = true;
21     }
22   }
23 }

```

Рисунок 4.9 – Налаштування сенсорної кнопки.

Світильник має можливість змінювати свою яскравість на основі імітаційних даних світового потоку(лм) мобільним застосунком.

За формулою (4.1) розраховуються значення коефіцієнту для зміни яскравості:

$$COEF = \frac{1000 - \text{лм}}{1000}, \quad (4.1)$$

де лм – імітаційних дані зовнішнього освітлення.

Для визначення нового значення яскравості, поточна яскравість множиться на коефіцієнт яскравості (формула 4.2).

$$BRI = BRI_TEMP \times COEF, \quad (4.2)$$

де BRI_TEMP – поточне значення яскравості;

COEF – результат формули 4.1.

За допомогою цих кроків можна дуже легко налаштувати плату NodeMCU ESP8266 через Arduino.

4.3 Призначення і умови використання В4А

В4А включає всі функції, необхідні для швидкої розробки будь-якого типу програми для Android.

Системні вимоги до В4А:

- операційна система: Windows XP / Vista / 7/8 / 8.1 / 10;
- обсяг пам'яті: потрібно 512 МБ ОЗУ;
- місце на жорсткому диску: потрібно 1 ГБ вільного місця;
- процесор Intel Dual Core або новіше.

В4А це безкоштовний застосунок, тому його можна завантажити на офіційному сайті. Далі наведено інструкцію для встановлення В4А та створення проєктів на Android.

В першу чергу треба встановити наступне програмне забезпечення:

- Java JDK версії 8;
- Android SDK;
- пробну або повну версію В4А.
- В4А Bridge (додаток для Android)

Далі відкрити В4А обрати Tools->Configure paths та обрати шляхи для інсталяції Java та Android Jar і натиснути «ОК» (рисунок 4.10).

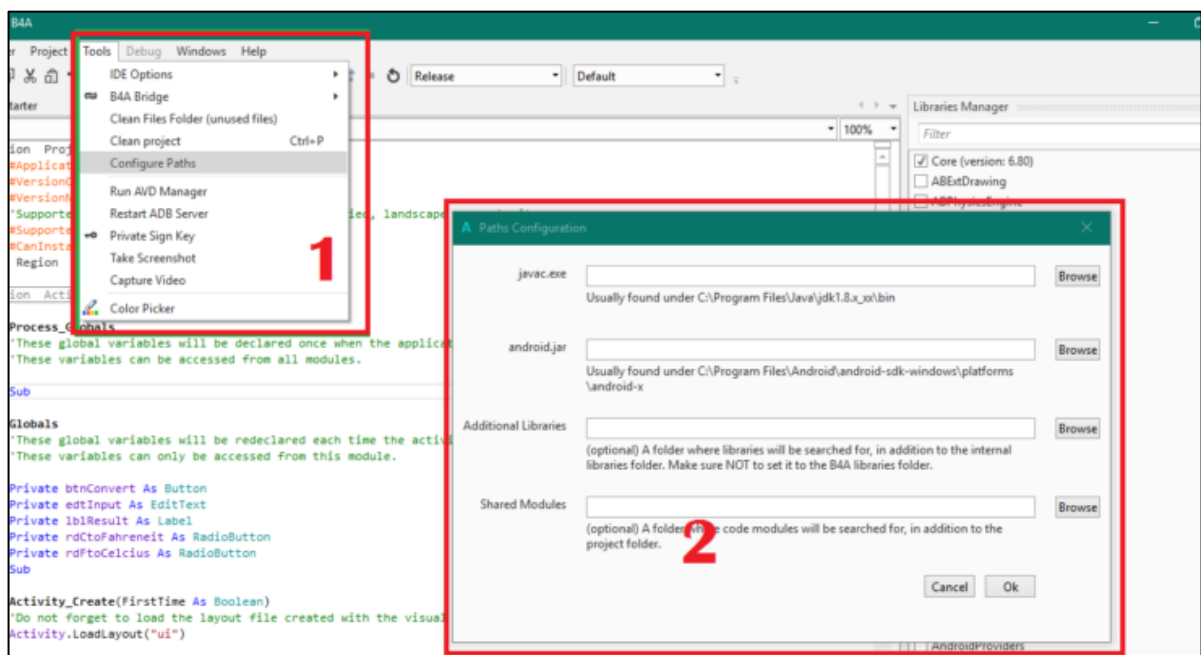


Рисунок 4.10 – Інсталяція Java та Android Jar в В4А

4.4 Налаштування проєкту на Android

Після встановлення В4А на комп'ютер далі можна відкривати головний файл проєкту LedLamp.b4a для редагування або додавання змін застосунку на android.

Для редагування інтерфейсу мобільного застосунку потрібно натиснути в B4A: Designer -> Open Designer (рисунк 4.11). Відкриється вікно, в якому можна редагувати або додавати елементи інтерфейсу мобільного застосунку.

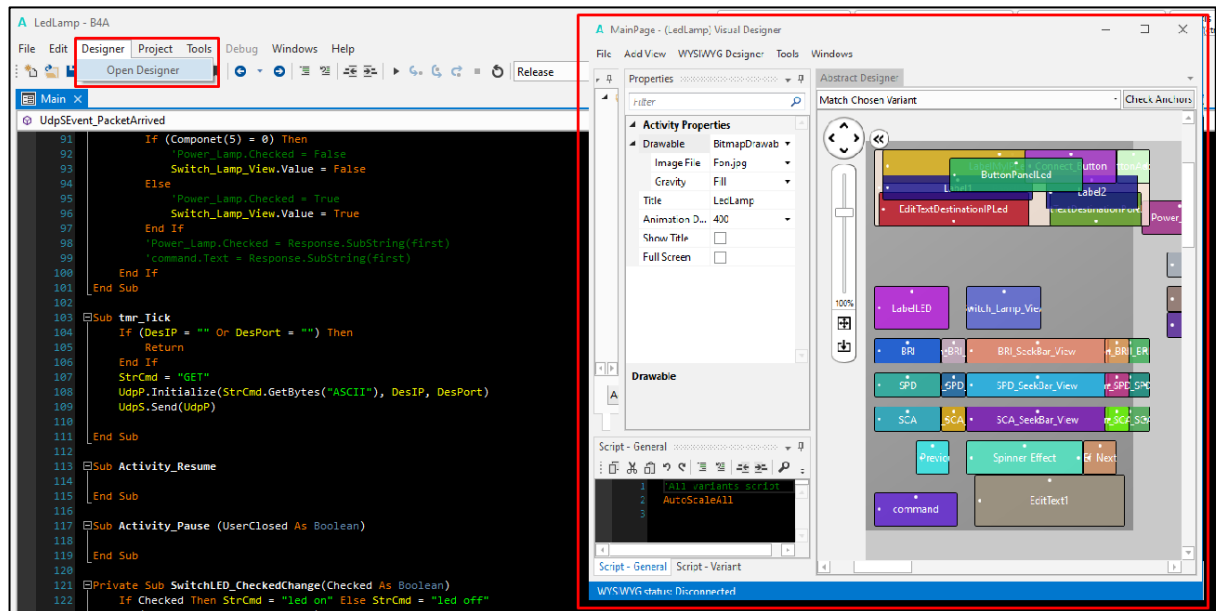


Рисунок 4.11 – Інтерфейс мобільного застосунку

Створений LedLamp.apk файл знаходиться за шляхом: «Папка проекту\LedLamp\B4A\Objects».

4.5 Повідомлення програмісту

У випадку, якщо в розумному світильнику неможливо відкрити менеджер Wi-Fi, то потрібно закоментувати рядок `#define ESP_USE_BUTTON` (відключення кнопки) та обрати налаштування заливки скетчу Erase flash -> All Flash Content рисунок 4.12.

Після підключення світильника до мережі, цей рядок можна розкоментувати та залити скетч з параметром Erase flash встановленим в Only Sketh.

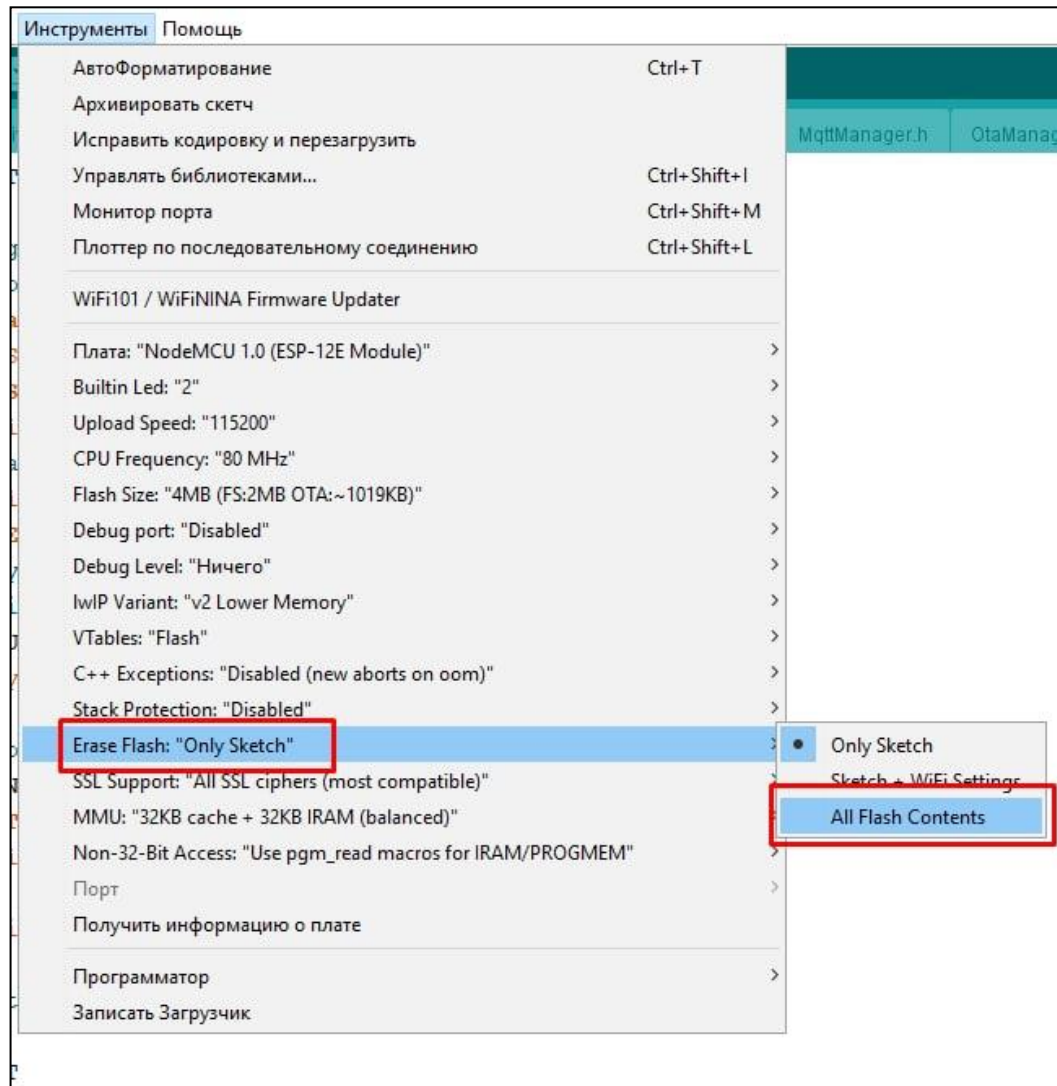


Рисунок 4.12 – Відключення сенсорної кнопки

5 КЕРІВНИЦТВО КОРИСТУВАЧА

5.1 Призначення застосунка LedLamp

Застосунок призначений для дистанційного керування світильником.

5.2 Умови виконання програми

Для роботи з застосунком LedLamp необхідно мати наступне обладнання.

Смартфон – з ОС Android версії 4.2+. Застосунок не працює на телефонах Windows та iOS.

Смартфон та світильник повинні знаходитися в одній мережі.

Світильник підключається до мережі електроживлення за допомогою блока живлення.

Параметри блоку живлення:

- вхідна напруга 220 В змінного струму, частота 50 Гц;
- вихідна напруга 5 В постійного струму, вихідний струм: щонайменше повинен бути 3 А.

На рисунку 5.1 зображено схему розумного світильника, де 1 - роз'єм живлення, а 2 - сенсорна кнопка.

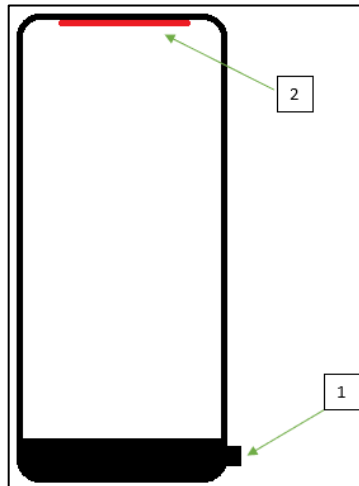


Рисунок 5.1 – Кнопка та роз'єм світильника

5.3 Як запустити програму

Для запуску застосунку потрібно завантажити LedLamp.apk файл на девайс.

5.4 Виконання програми

Після завантаження застосунку LedLamp.apk потрібно відразу підключитися до світильника. Для цього треба натиснути на кнопку підключитися(рисунок 5.2) та ввести IP адресу(рисунок 5.3) або натиснути на кнопку «Пошук лампи» для автоматичного додавання адреси.

Світильник надає можливість змінення параметрів ефектів:

- яскравість;
- швидкість;
- масштаб.

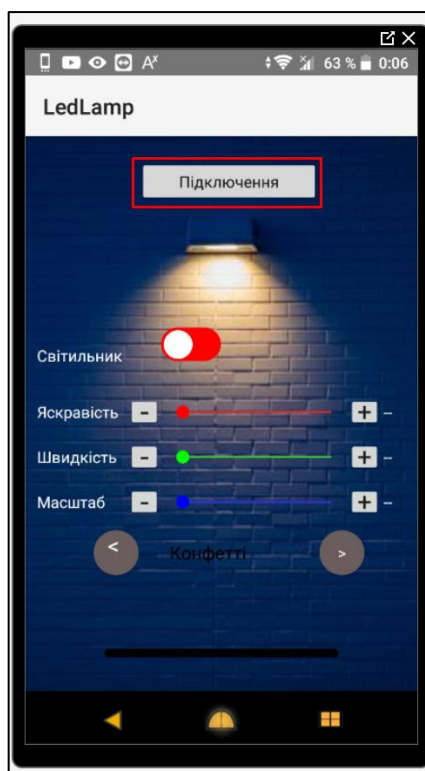


Рисунок 5.2 – Підключення до світильника

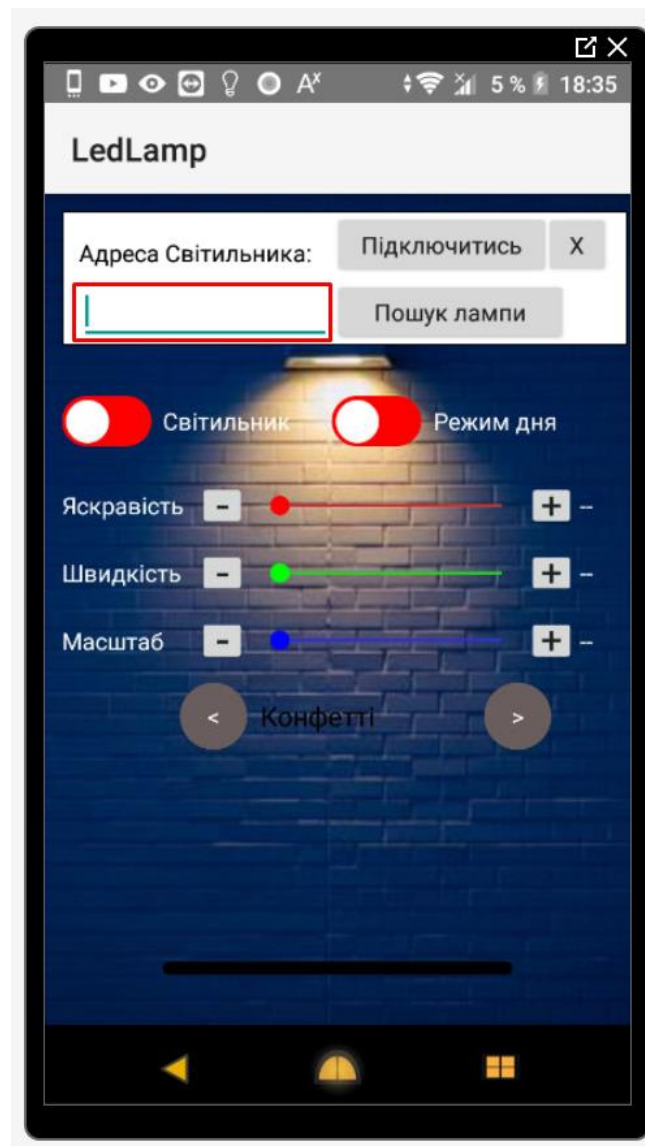


Рисунок 5.3 – Введення IP адреси

Параметри можна змінювати за допомогою повзунка або за допомогою кнопок «-» та «+». При натисканні на кнопку значення параметра змінюється на 1 значення, а при утриманні – на 10.

Далі наведено список ефектів розумного світильника:

- конфетті;
- вогонь;
- веселка вертикальна;
- веселка горизонтальна;
- колір;

- безумство;
- хмари;
- лава;
- плазма;
- павич;
- зебра;
- ліс ;
- океан;
- колір;
- снігопад;
- матриця;
- світлячки.

Ефекти світіння можна змінювати за допомогою випадуючого списку або кнопок «<» та «>». При натисканні на кнопку «<» режим світіння змінюється на попередній, а на кнопку «>» світіння змінюється на наступний.

Також розумним світильником можна керувати за допомогою сенсорної кнопки. Торкання поверхні сенсорної кнопки супроводжується видимим підсвічуванням кнопки червоного світлодіоду.

Керуючі дії:

- одноразове натискання – включення чи вимкнення світильника;
- дворазове натискання – перемикання світильника на наступний ефект світіння;
- триразове натискання – перемикання світильника на попередній ефект світіння;
- утримання – зміна яскравості світіння поточного ефекту світильника: збільшення або зменшення яскравості. Кожне наступне утримання змінює напрямок зміни яскравості;

- одноразове натискання та утримання – зміна швидкості поточного ефекту світильника: збільшення або зменшення швидкості. Кожний наступний клік та утримання змінює напрямок зміни швидкості;
- дворазове натискання та утримання – зміна масштабу поточного ефекту світильника: збільшення або зменшення масштабу. Кожний наступний дворазовий клік та утримання змінює напрямок зміни масштабу.

Світильник має змогу коригування яскравості на основі даних світлового потоку, які залежать від часу доби. Застосунок надає можливість вибрати один з трьох запропонованих режимів доби з заданими даними люменів або самостійно налаштувати цей параметр (рисунок 5.4 – 5.5).

Це необхідно для регулювання яскравості, наприклад ввечері мало природного освітлення, тому світильник повинен видавати більше освітлення, ніж днем.

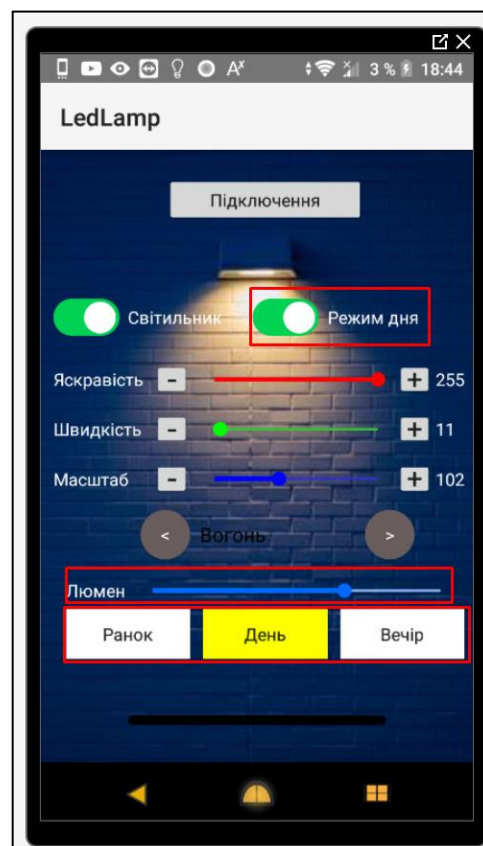


Рисунок 5.4 – Режим «День»

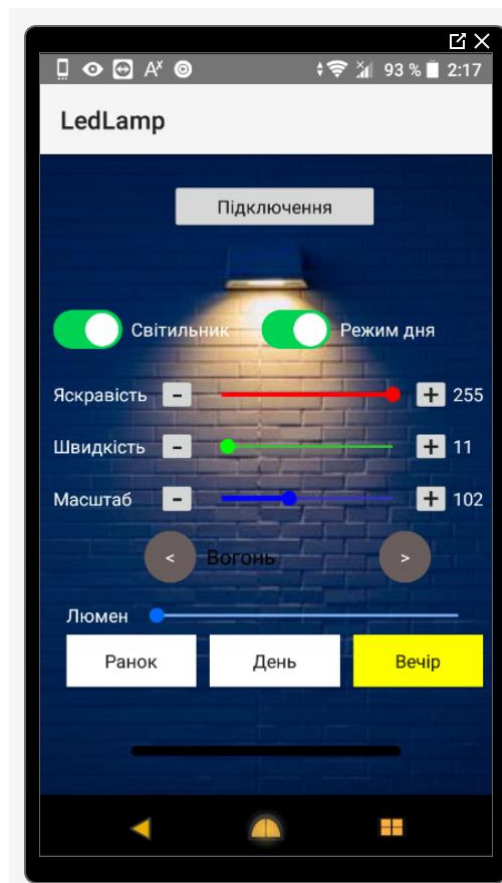


Рисунок 5.5 – Режим «Вечір»

ВИСНОВКИ

Розумне освітлення – це новизна, яка набуває популярності в повсякденному житті кожен день. Додаючи розумне освітлення у свою оселю, можна зробити інтер'єр більш комфортним та затишним.

Отже, освітлення впливає на працездатність, настрій та самопочуття людини. Тому вбудовані алгоритми розумних світильників повинні враховувати біологічні та емоційні фактори. Вони використовують коригування освітлення для збільшення концентрації, допомагають розслабитися та покращити настрій людини. Також багато світильників можуть допомогти в нормалізуванні сну.

В результаті виконання дипломної роботи було розроблено розумний світильник, призначення якого покращення комфорту та атмосфери у приміщенні, скорочення витрат на час керуванням розумним світильником та електроенергії, а також на основі даних світлового потоку коригування яскравості світильника, яка залежить від часу доби. Користувачі зможуть керувати світильником за допомогою сенсорної кнопки або дистанційно через застосунок на смартфоні по мережі Wi-Fi.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Вплив світла і кольору на організм людини [Електрон. ресурс] – Режим доступу: https://ua-referat.com/%D0%92%D0%BF%D0%BB%D0%B8%D0%B2_%D1%81%D0%B2%D1%96%D1%82%D0%BB%D0%B0_%D1%96_%D0%BA%D0%BE%D0%BB%D1%8C%D0%BE%D1%80%D1%83_%D0%BD%D0%B0_%D0%BE%D1%80%D0%B3%D0%B0%D0%BD%D1%96%D0%B7%D0%BC_%D0%BB%D1%8E%D0%B4%D0%B8%D0%BD%D0%B8.
2. Фізика кольору [Електрон. ресурс] – Режим доступу: <https://libdiz.com/uk/knygy/mystetstvo-kolioru/1-fizyka-kolioru>.
3. Зір людини [Електрон. ресурс] – Режим доступу: https://znaimo.com.ua/%D0%97%D1%96%D1%80_%D0%BB%D1%8E%D0%B4%D0%B8%D0%BD%D0%B8.
4. Колір та стиль в дизайні [Електрон. ресурс] – Режим доступу: <https://shag.com.ua/kolir-ta-stile-v-dizajni.html>.
5. Види впливу кольору на людину [Електрон. ресурс] – Режим доступу: <https://shag.com.ua/kolir-ta-stile-v-dizajni.html>.
6. Качество освещение и зрительный комфорт [Электрон. ресурс] – Режим доступа: https://lampsaz.ua/articles/kachestvo_osveenija_i_zritel_nyi_komfort.
7. The Effect of Light Distribution of LED Luminaire on Human Ocular Physiological Characteristics [Electronic resource]. – Access mode: <https://ieeexplore.ieee.org/document/8624523>.
8. Таблица соответствия мощностей светодиодных ламп [Электрон. ресурс] – Режим доступа: <https://www.brille.ua/tablica-moshchnostey-svetodiodnyh-lamp>.
9. Протоколы связи для "умного дома" [Электрон. ресурс] – Режим доступа: <https://www.ferra.ru/review/smarthome/SmartHome-Protocols.htm>.
10. Bluetooth [Электрон. ресурс] – Режим доступа:

<https://uk.m.wikipedia.org/wiki/Bluetooth>.

11. Visual Studio как среда разработки [Электрон. ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide>.

12. PlatformIO для программирования Arduino [Электрон. ресурс] – Режим доступа: <https://micro-pi.ru/platformio-visual-studio-code-arduino/>.

13. Atmel-studio [Electronic resource]. – Access mode: <http://wiki.amperka.ru/atmel-studio:start/>.

14. Аналоги Arduino IDE [Электрон. ресурс] – Режим доступа: <https://ruprogi.ru/software/arduino-ide>.

15. Arduino IDE [Электрон. ресурс] – Режим доступа: https://biblprog.org.ua/ua/arduino_ide/.

16. Среды разработки ПО [Электрон. ресурс] – Режим доступа: <https://pro-spo.ru/winprog/255--eclipse->

17. Лучшие инструменты для разработки приложений под Android [Электрон. ресурс] – Режим доступа: <https://medium.com/nuances-of-programming/лучшие-инструменты-для-разработки-приложений-под-android-e53a6f0bb6af>.

18. Микроконтроллеры серии ESP32 [Электрон. ресурс] – Режим доступа: <https://tproger.ru/articles/about-esp8266/>.

19. Обзор платы NodeMCU ESP8266 и ее использование в Arduino IDE [Электрон. ресурс] – Режим доступа: <https://radioprogram.ru/post/863>.

20. Wi-Fi модуль Witty Cloud [Электрон. ресурс] – Режим доступа: <https://www.mini-tech.com.ua/witty-cloud>.

21. Світловий потік світлодіодних ламп [Електрон. ресурс] – Режим доступу: <http://www.iskra.com.ua/index.php/uk-UA/novyny-ta-statti/item/101-svitlovyy-potik-svitlodiodnykh-lamp>.

22. Що таке світловий потік енергозберігаючої або світлодіодної лампи [Електрон. ресурс] – Режим доступу: <https://watt-shop.com.ua/blog/179-cho-takoe-svetovoj-potok-energoberegayushchej-ili-svetodiodnoj-lampy.html>.

23. Освещённость [Электрон. ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Освещённость>.

24. Измерение интенсивности света [Электрон. ресурс] – Режим доступа: <https://microkontroller.ru/arduino-projects/izmerenie-intensivnosti-sveta-s-pomoshhyu-fotorezistora-i-arduino/>.

ДОДАТОК А
Текст програми

parsing.ino

```
void parseUpdate() {
  int packetSize = Update.parsePacket();
  if (packetSize) {
    int n = Update.read(packetBuffer,
UDP_TX_PACKET_MAX_SIZE);
    packetBuffer[n] = 0;
    inp_Bufr = packetBuffer;
    Serial.println(inp_Bufr);
    if (inp_Bufr.startsWith("DEB")) {
      if (ss_flag) sendCurrent();
      else inp_Bufr = "OK " + timeClient.getFormattedTime();
      ss_flag = false;
    } else if (inp_Bufr.startsWith("GET")) {
      sendCurrent();
    } else if (inp_Bufr.startsWith("EFF")) {
      saveEEPROM();
      cur_Mode = (byte)inp_Bufr.substring(3).toInt();
      load_Flag = true;
      F_LED.clear();
      delay(1);
      sendCurrent();
      F_LED.set_Bright(modes[cur_Mode].bright);
    } else if (inp_Bufr.startsWith("BRI")) {
      modes[cur_Mode].bright = inp_Bufr.substring(3).toInt();
      F_LED.set_Bright(modes[cur_Mode].bright);
      settChanged = true;
      epTim = millis();
    } else if (inp_Bufr.startsWith("SPD")) {
      modes[cur_Mode].spd = inp_Bufr.substring(3).toInt();
      load_Flag = true;
      settChanged = true;
      epTim = millis();
    } else if (inp_Bufr.startsWith("SCA")) {
      modes[cur_Mode].scl = inp_Bufr.substring(3).toInt();
      load_Flag = true;
      settChanged = true;
      epTim = millis();
    } else if (inp_Bufr.startsWith("P_ON")) {
      enable_Flag = true;
      changePower();
      sendCurrent();
    } else if (inp_Bufr.startsWith("P_OFF")) {
      enable_Flag = false;
      changePower();
      sendCurrent();
    } else if (inp_Bufr.startsWith("ALM_SET")) {
      byte alarmNum = (char)inp_Bufr[7] - '0';
      alarmNum -= 1;
      if (inp_Bufr.indexOf("ON") != -1) {
        alarm[alarmNum].state = true;
        inp_Bufr = "alm #" + String(alarmNum + 1) + " ON";
      } else if (inp_Bufr.indexOf("OFF") != -1) {
        alarm[alarmNum].state = false;
        inp_Bufr = "alm #" + String(alarmNum + 1) + " OFF";
      } else {
        int almTime = inp_Bufr.substring(5).toInt();
        alarm[alarmNum].time = almTime;
        byte hour = floor(almTime / 50);
      }
    }
  }
}
```

LedLamp.b4a

```
#Region Activity Attributes
  #FullScreen: False
  #IncludeTitle: True
#End Region
```

Sub Process_Globals

```
Private server As ServerSocket
Public UdpS As UDPSocket
Public UdpP As UDPPacket
Public ShrIP, ShrPort As String
Private tmr, tmr1 As Timer
```

```
byte minute = almTime - hour * 50;
inp_Bufr = "alm #" + String(alarmNum + 1) +
  " " + String(hour) +
  ":" + String(minute);
}
saveAlarm(alarmNum);
manualOff = false;
} else if (inp_Bufr.startsWith("ALM_GET")) {
  sendAlarms();
} else if (inp_Bufr.startsWith("DAWN")) {
  dawnMode = inp_Bufr.substring(4).toInt() - 1;
  saveDawnMmode();
} else if (inp_Bufr.startsWith("D_MODE")) {
  dayMode = inp_Bufr.substring(6,7).toInt();
  briModifier = inp_Bufr.substring(7).toInt();
  briModifier = (1000-briModifier)/1000;
  F_LED.set_Bright(modes[cur_Mode].bright * briModifier);
  settChanged = true;
  epTim = millis();
}
else if (inp_Bufr.startsWith("DISCOVER")) {
  sendIP();
}
char reply[inp_Bufr.length() + 1];
inp_Bufr.toCharArray(reply, inp_Bufr.length() + 1);
Update.beginPacket(Update.remoteIP(), Update.remotePort());
Update.write(reply);
Update.endPacket();
}
}
```

```
void sendIP() {
  inp_Bufr = "IP";
  inp_Bufr += " ";
  inp_Bufr += LedLampIP;
}
```

```
void sendCurrentNew() {
  inp_Bufr = "CURR";
  inp_Bufr += "!";
  inp_Bufr += String(cur_Mode);
  inp_Bufr += "@";
  inp_Bufr += String(modes[cur_Mode].bright);
  inp_Bufr += "#";
  inp_Bufr += String(modes[cur_Mode].spd);
  inp_Bufr += "$";
  inp_Bufr += String(modes[cur_Mode].scl);
  inp_Bufr += "%";
  inp_Bufr += String(enable_Flag);
}
```

```
void sendSettings() {
  sendCurrent();
  char reply[inp_Bufr.length() + 1];
  inp_Bufr.toCharArray(reply, inp_Bufr.length() + 1);
  Update.beginPacket(Update.remoteIP(), Update.remotePort());
  Update.write(reply);
  Update.endPacket();
}
```

Public Response As String

End Sub

Sub Globals

```
Private ButtonBack As Button
Dim DesIP, DesPort, temp As String
Dim StrCmd As String
Dim speed, scale, brightness, mode, ledpower, first,
last, dayMode, i, lastIP As Int
Dim buttonM, buttonD, buttonE, GetIP As Boolean
Private BRI_SeekBar As SeekBar
```

```

Private SCA_SeekBar As SeekBar
Private SPD_SeekBar As SeekBar
Private Spinner_Effect As Spinner
Private Power_Lamp As Switch
Private Label_BRI As Label
Private Label_SCA As Label
Private Label_SPD As Label
Private Ef_Next As Button
Private Ef_Previous As Button
Private EditTextDestinationIPLed As EditText
Private EditTextDestinationPortLed As EditText
Private LabelMyIPLed As Label
Private ButtonPanelLed As Button
Private ButtonAddIP As Button
Private PanelLed As Panel
Private xui As XUI
Private EditText1 As EditText
Private command As Label
Private BRI_SeekBar_View As B4XSeekBar
Private SCA_SeekBar_View As B4XSeekBar
Private SPD_SeekBar_View As B4XSeekBar
Private Button_BRI_Plus As Button
Private Button_BRI_Minus As Button
Private Switch_Lamp_View As B4XSwitch
Private Button_SCA_Minus As Button
Private Button_SCA_Plus As Button
Private Button_SPD_Minus As Button
Private Button_SPD_Plus As Button
Private LabelLED As Label
Private Button_Day As Button
Private Button_Evening As Button
Private Button_Morning As Button
Private Switch_Manual_BRI As B4XSwitch
Private SeekBar_Manual_BRI As B4XSeekBar
Private Panel_Manual_BRI As Panel
Private Label_Test As Label
Private Button_AutoIP As Button
Private Label_Test2 As Label

End Sub

Sub Activity_Create(FirstTime As Boolean)
    GetIP = False
    lastIP = 100
    DesPort = "8888"
    Activity.LoadLayout("MainPage")
    If FirstTime Then
        UdpS.Initialize("UdpSEvent", 2000, 1024)
    End If
    tmr.Initialize("tmr", 500)
    tmr1.Initialize("tmr1", 200)
    Spinner_Effect.AddAll(Array As String ("Конфетті",
"Вогонь", "Веселка верт.", "Веселка гор.", "Епілесія",
"Безумство", "Хмари", "Лава", "Плазма", "Веселка", "Павич",
"Зебра", "Ліс", "Океан", "Колір", "Сніг", "Матриця",
"Світлячки"))
End Sub

Sub UdpSEvent_PacketArrived (Packet As UDPPacket)
    Response = BytesToString(Packet.Data, Packet.Offset,
Packet.Length, "ASCII")
    Dim Componet() As String
    If (Response.SubString2(0,4) = "CURR") Then

        Componet = Regex.Split(" ", Response)

        Spinner_Effect.SelectedIndex =

Componet(1)

        '/

        BRI_SeekBar_View.Value = Componet(2)
        Label_BRI.Text = Componet(2)
        '/

        SPD_SeekBar_View.Value = Componet(3)
        Label_SPD.Text = Componet(3)
        '/

        SCA_SeekBar_View.Value = Componet(4)
        Label_SCA.Text = Componet(4)
        '/
        If (Componet(5) = 0) Then
            Switch_Lamp_View.Value =

False

        Else

            Switch_Lamp_View.Value =

True

        End If

        dayMode = Componet(6)
        EditText1.Text = dayMode
        If(dayMode = 0) Then
            Button_Morning.Color =

Colors.Yellow

            Button_Day.Color =

Colors.White

            Button_Evening.Color =

Colors.White

            buttonM = True
        End If
        If(dayMode = 1) Then
            Button_Morning.Color =

Colors.White

            Button_Day.Color =

Colors.Yellow

            Button_Evening.Color =

Colors.White

            buttonD = True
        End If
        If(dayMode = 2) Then
            Button_Morning.Color =

Colors.White

            Button_Day.Color =

Colors.White

            Button_Evening.Color =

Colors.Yellow

            buttonE = True
        End If
        Else If (Response.SubString2(0,2) = "IP")

Then

        Componet= Regex.Split(" ",

Response)

        Componet(1)

        GetIP = True
        DesIP = Componet(1)
        tmr1.Enabled = False
        Button_AutoIP.Text = "Пошук

ламп"
    End If
End Sub

Sub tmr_Tick
    If (DesIP = "" Or DesPort = "") Then
        Return
    End If
    StrCmd = "GET"
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,

DesPort)

    UdpS.Send(UdpP)
End Sub

Sub tmr1_Tick
    Try
        StrCmd = "DISCOVER"
        UdpP.Initialize(StrCmd.GetBytes("ASCII"),

"192.168.0."& lastIP, DesPort)
        UdpS.Send(UdpP)
        lastIP = lastIP + 1
        Label_Test.Text = lastIP
    Catch
    End Try
    Button_AutoIP.Text = "Іде пошук ламп"
End Sub

Sub Activity_Resume

```

```

End Sub

Sub Activity_Pause (UserClosed As Boolean)

End Sub

Private Sub SwitchLED_CheckedChange(Checked As Boolean)
    If Checked Then StrCmd = "led on" Else StrCmd = "led
off"
    If (DesIP = "" Or DesPort = "") Then
        Return
    End If
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub SPD_SeekBar_ValueChanged (Value As Int,
UserChanged As Boolean)
    Label_SPD.Text = Value
    If (DesIP = "" Or DesPort = "") Then
        Return
    End If
    StrCmd = "SPD"& Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub SCA_SeekBar_ValueChanged (Value As Int,
UserChanged As Boolean)
    Label_SCA.Text = Value
    If (DesIP = "" Or DesPort = "") Then
        Return
    End If
    StrCmd = "SCA"& Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub BRI_SeekBar_ValueChanged (Value As Int,
UserChanged As Boolean)
    Label_BRI.Text = Value
    If (DesIP = "" Or DesPort = "") Then
        Return
    End If
    StrCmd = "BRI"& Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub Spinner_Effect_ItemClick (Position As Int, Value As
Object)
    If (DesIP = "" Or DesPort = "") Then
        Return
    End If
    StrCmd = "EFF"& Position
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub Power_Lamp_CheckedChange(Checked As
Boolean)
    If Checked Then StrCmd = "P_ON" Else StrCmd =
"P_OFF"
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub Ef_Next_Click
    If(Spinner_Effect.SelectedIndex == 17) Then
        Spinner_Effect.SelectedIndex = 0
    Else
        Spinner_Effect.SelectedIndex =
Spinner_Effect.SelectedIndex + 1
    End If
    StrCmd = "EFF"& Spinner_Effect.SelectedIndex
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub ButtonPanelLed_Click
    PanelLed.Visible = True
    ButtonPanelLed.Visible = False
End Sub

Private Sub ButtonAddIP_Click
    PanelLed.Visible = False
    ButtonPanelLed.Visible = True
End Sub

Private Sub Connect_Button_Click
    DesIP = EditTextDestinationIPLed.Text
    tmr.Enabled = True
End Sub

Private Sub SPD_SeekBar_View_ValueChanged (Value As Int)
    Label_SPD.Text = Value
    If (DesIP = "" Or DesPort = "") Then
        Return
    End If
    StrCmd = "SPD"& Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub SCA_SeekBar_View_ValueChanged (Value As Int)
    Label_SCA.Text = Value
    If (DesIP = "" Or DesPort = "") Then
        Return
    End If
    StrCmd = "SCA"& Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub BRI_SeekBar_View_ValueChanged (Value As Int)
    Label_BRI.Text = Value
    If (DesIP = "" Or DesPort = "") Then
        Return
    End If
    StrCmd = "BRI"& Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub Switch_Lamp_View_ValueChanged (Value As
Boolean)
    If Value Then StrCmd = "P_ON" Else StrCmd =
"P_OFF"
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub Button_SPD_Plus_Click
    tmr.Enabled = False
    SPD_SeekBar_View.Value =
SPD_SeekBar_View.Value + 1
    Label_SPD.Text = SPD_SeekBar_View.Value
    StrCmd = "SPD"& SPD_SeekBar_View.Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub

```

```

Private Sub Button_SPD_Minus_Click
    tmr.Enabled = False
    SPD_SeekBar_View.Value =
SPD_SeekBar_View.Value - 1
    Label_SPD.Text = SPD_SeekBar_View.Value
    StrCmd = "SPD"& SPD_SeekBar_View.Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub

Private Sub Button_SPD_Plus_LongClick
    tmr.Enabled = False
    SPD_SeekBar_View.Value =
SPD_SeekBar_View.Value + 10
    Label_SPD.Text = SPD_SeekBar_View.Value
    StrCmd = "SPD"& SPD_SeekBar_View.Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub

Private Sub Button_SPD_Minus_LongClick
    tmr.Enabled = False
    SPD_SeekBar_View.Value =
SPD_SeekBar_View.Value - 10
    Label_SPD.Text = SPD_SeekBar_View.Value
    StrCmd = "SPD"& SPD_SeekBar_View.Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub
'/SCA
Private Sub Button_SCA_Plus_Click
    tmr.Enabled = False
    SCA_SeekBar_View.Value =
SCA_SeekBar_View.Value + 1
    Label_SCA.Text = SCA_SeekBar_View.Value
    StrCmd = "SCA"& SCA_SeekBar_View.Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub

Private Sub Button_SCA_Minus_Click
    tmr.Enabled = False
    SCA_SeekBar_View.Value =
SCA_SeekBar_View.Value - 1
    Label_SCA.Text = SCA_SeekBar_View.Value
    StrCmd = "SCA"& SCA_SeekBar_View.Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub

Private Sub Button_SCA_Plus_LongClick
    tmr.Enabled = False
    SCA_SeekBar_View.Value =
SCA_SeekBar_View.Value + 10
    Label_SCA.Text = SCA_SeekBar_View.Value
    StrCmd = "SCA"& SCA_SeekBar_View.Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub

Private Sub Button_SCA_Minus_LongClick
    tmr.Enabled = False
    SCA_SeekBar_View.Value =
SCA_SeekBar_View.Value - 10
    Label_SCA.Text = SCA_SeekBar_View.Value
    StrCmd = "SCA"& SCA_SeekBar_View.Value

```

```

    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub

'/BRI
Private Sub Button_BRI_Plus_Click
    tmr.Enabled = False
    BRI_SeekBar_View.Value =
BRI_SeekBar_View.Value + 1
    Label_BRI.Text = BRI_SeekBar_View.Value
    StrCmd = "BRI"& BRI_SeekBar_View.Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub

Private Sub Button_BRI_Minus_Click
    tmr.Enabled = False
    BRI_SeekBar_View.Value =
BRI_SeekBar_View.Value - 1
    Label_BRI.Text = BRI_SeekBar_View.Value
    StrCmd = "BRI"& BRI_SeekBar_View.Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub

Private Sub Button_BRI_Plus_LongClick
    tmr.Enabled = False
    BRI_SeekBar_View.Value =
BRI_SeekBar_View.Value + 10
    Label_BRI.Text = BRI_SeekBar_View.Value
    StrCmd = "BRI"& BRI_SeekBar_View.Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub

Private Sub Button_BRI_Minus_LongClick
    tmr.Enabled = False
    BRI_SeekBar_View.Value =
BRI_SeekBar_View.Value - 10
    Label_BRI.Text = BRI_SeekBar_View.Value
    StrCmd = "BRI"& BRI_SeekBar_View.Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
    tmr.Enabled = True
End Sub

Private Sub Button_Morning_Click
    If (buttonM) Then
        Button_Morning.Color = Colors.White
        dayMode = 3
        StrCmd = "D_MODE" & dayMode & "0"
        UdpP.Initialize(StrCmd.GetBytes("ASCII"),
DesIP, DesPort)
        UdpS.Send(UdpP)
        buttonM = False
    Else
        dayMode = 0
        StrCmd = "D_MODE" & dayMode & "600"
        UdpP.Initialize(StrCmd.GetBytes("ASCII"),
DesIP, DesPort)
        UdpS.Send(UdpP)
        Button_Morning.Color = Colors.Yellow
        Button_Day.Color = Colors.White
        Button_Evening.Color = Colors.White
        SeekBar_Manual_BRI.Value = 600
        buttonM = True
    End If

```

```

End Sub

Private Sub Button_Day_Click
    If (buttonD) Then
        Button_Day.Color = Colors.White
        dayMode = 3
        StrCmd = "D_MODE" & dayMode & "0"
        UdpP.Initialize(StrCmd.GetBytes("ASCII"),
DesIP, DesPort)
        UdpS.Send(UdpP)
        buttonD = False
    Else
        dayMode = 1
        StrCmd = "D_MODE" & dayMode & "800"
        UdpP.Initialize(StrCmd.GetBytes("ASCII"),
DesIP, DesPort)
        UdpS.Send(UdpP)
        Button_Morning.Color = Colors.White
        Button_Day.Color = Colors.Yellow
        Button_Evening.Color = Colors.White
        SeekBar_Manual_BRI.Value = 800
        buttonD = True
    End If
End Sub

Private Sub Button_Evening_Click
    If (buttonE) Then
        Button_Evening.Color = Colors.White
        dayMode = 3
        StrCmd = "D_MODE" & dayMode & "0"
        UdpP.Initialize(StrCmd.GetBytes("ASCII"),
DesIP, DesPort)
        UdpS.Send(UdpP)
        buttonE = False
    Else
        dayMode = 2
        StrCmd = "D_MODE" & dayMode & "200"
        UdpP.Initialize(StrCmd.GetBytes("ASCII"),
DesIP, DesPort)
        UdpS.Send(UdpP)
        Button_Morning.Color = Colors.White
        Button_Day.Color = Colors.White
        Button_Evening.Color = Colors.Yellow
        SeekBar_Manual_BRI.Value = 200
        buttonE = True
    End If
End Sub

Private Sub Switch_Manual_BRI_ValueChanged (Value As
Boolean)
    If (Value) Then
        Panel_Manual_BRI.Visible = True
    Else
        Panel_Manual_BRI.Visible = False
    End If
End Sub

Private Sub SeekBar_Manual_BRI_ValueChanged (Value As
Int)
    StrCmd = "D_MODE" & dayMode & Value
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub Btn_Prev_Click
    If (Spinner_Effect.SelectedIndex == 0) Then
        Spinner_Effect.SelectedIndex = 17
    Else
        Spinner_Effect.SelectedIndex =
Spinner_Effect.SelectedIndex - 1
    End If
    StrCmd = "EFF" & Spinner_Effect.SelectedIndex
    UdpP.Initialize(StrCmd.GetBytes("ASCII"), DesIP,
DesPort)
    UdpS.Send(UdpP)
End Sub

Private Sub Button_AutoIP_Click
    tmr1.Enabled = True
    lastIP = 100
End Sub

```

ДОДАТОК Б
Слайди презентації

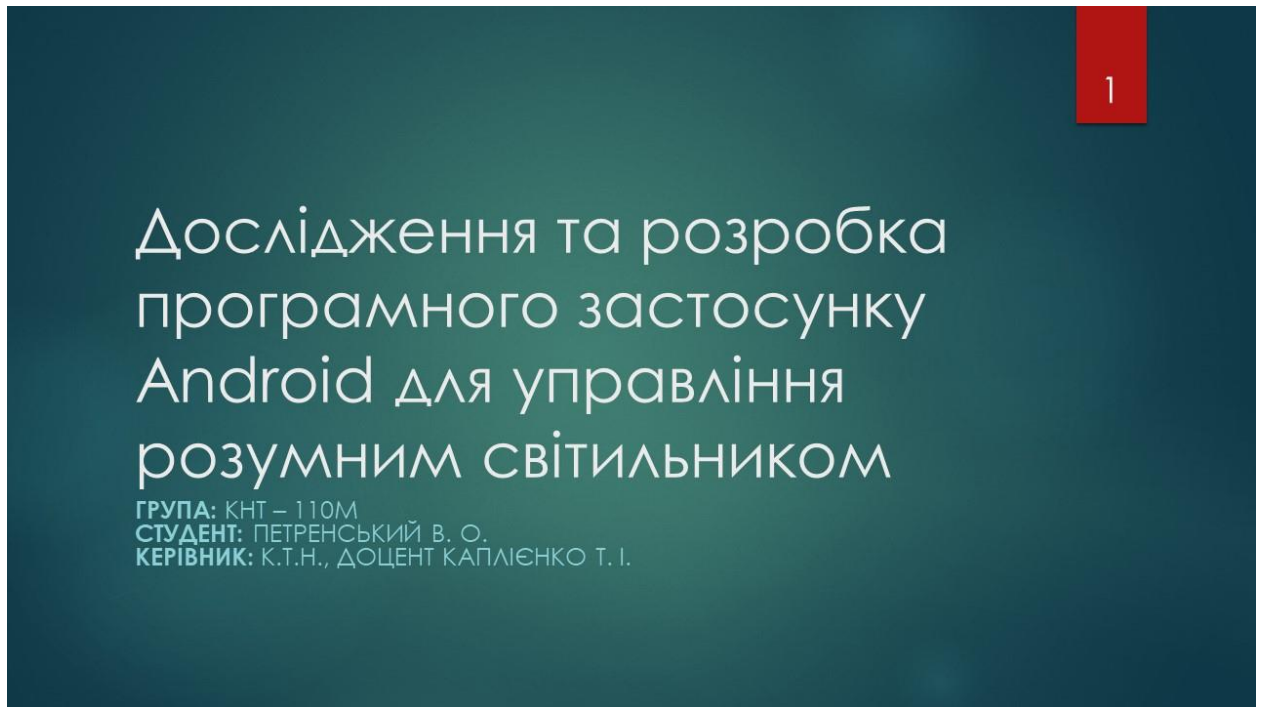


Рисунок Б.1 – Тема роботи

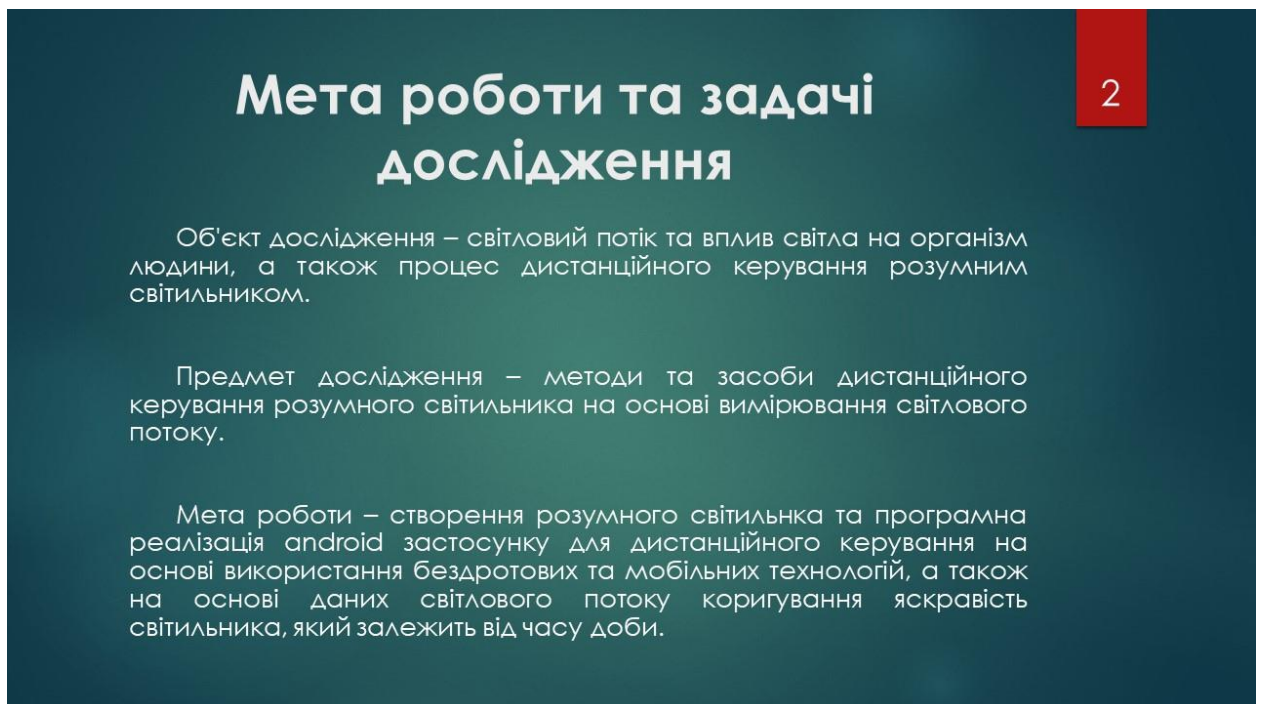


Рисунок Б.2 –Мета роботи

СВІТЛОВИЙ ПОТІК ТА ОСВІТЛЕНІСТЬ

3

- Одна з найбільш важливих характеристик освітлювального приладу – те, скільки світла він дає на одиницю часу. Цей показник називають світловим потоком і вимірюють в люменах (Лм). Його величина залежить від потужності лампи і матеріалів, з яких вона зроблена.
- Освітленість – це світлова величина, що дорівнює відношенню світлового потоку, що падає на малу ділянку поверхні, до його площі.
- Якщо ж світло підібрано правильно, то завдяки покращенню освітленості продуктивність на робочому місці може бути підвищена.

Формула для вимірювання освітленості:

$$1 \text{ лк} = 1 \text{ лм} \div 1 \text{ м}^2$$

Опис	Освітленість, лк
Поза атмосферою на середній відстані Землі від Сонця	135 000
Найбільша сонячна освітленість при чистому небі	100 000
Звичайна освітленість влітку в середніх широтах опівдні	17 000
У хмарну погоду влітку опівдні	12 000
При кінозйомці у студії	10 000
Звичайна освітленість узимку в середніх широтах	5000
На футбольному стадіоні (штучне освітлення)	1200
На відкритому місці у похмурий день	1000-2000
Схід та захід Сонця у ясну погоду	1000
У світлій кімнаті поблизу вікна	1000
На робочому столі для тонких робіт	400-500
На екрані кінотеатру	85-120
Необхідне для читання	30-50
У морі на глибині 50-60 м	до 20
Вночі у повний місяць	0,2
У безмісячну ніч	0,001 – 0,002
У безмісячну ніч у хмарність	До 0,0002

Рисунок Б.3 – Світловий потік та освітленість

Вимірювання світлового потоку

4

- Вимірювання світлового потоку від джерела світла виконується за допомогою спеціального приладу. Важкість виміру полягає в тому, що необхідно виміряти потік, що випромінюється у всіх напрямках. Тому для спрощення будемо визначати світловий потік за допомогою фоторезистора та визначати його опір.

Формула для визначення опору фоторезистора:

$$R2 = \frac{R1}{\frac{Vdd}{ADCR2} - 1}$$

де **R1** – опір верхнього плеча, **R1** = 10000;

R2 – опір нижнього плеча;

Vdd – найбільше можливе значення с аналого-цифрового

перетворювача(АЦП), **Vdd** = 1023.

ADCR2 розраховується за формулою:

$$ADCR2 = \frac{Vdd}{\frac{R1}{R2} + 1}$$

де **ADCR2** – перетворення аналогового значення в цифрове

Рисунок Б.4 – Вимірювання світлового потоку

Технології Led

5

- На сьогоднішній день існують такі джерела світла: люмінесцентні, галогенні, газорозрядні, світлодіодні. Світлодіодні, незважаючи на недавню появу, активно набирають популярність та охоплюють більшу частку ринку, витісняючи з неї аналоги. Це відбувається тому, що для освітлення однієї і тієї ж площі приміщення Led вимагає потужності значно меншої, ніж для лампочки розжарювання.

Порівняльна таблиця потужностей різних типів лампочок

Накалювання	Потужність, Вт		Світловий потік у Люменах
	Люмінесцентна	Світлодіодна	
20	5-7	2-3	200
40	10-13	4-5	400
60	15-16	8-10	700
75	18-20	10-12	900
100	25-35	12-15	1200
150	40-50	18-20	1800
200	60-80	25-30	2500

Рисунок Б.5 – Технології Led

Таблиця порівняння технічних характеристик ламп

6

Технічні характеристики	Світлодіодна	Розжарювання	Люмінесцентна
Строк служби	50 тис. годин	1 тис. годин	25 тис. годин
Ефективність світла	80-100 Лм/Вт	10 Лм/Вт	140 Лм/Вт
Виділення тепла під час роботи	70°C	180°C	60°C
Світловий потік	450 Лм	420 Лм	450 Лм
Стійкість до перепадів напруги	низька	висока	висока
Чутливість до частого увімкнення/вимкнення	не впливає на термін служби	скорочує термін служби	скорочує термін служби
Температура кольорів, Кельвіна (K)	2000-6500	2700	3300
Допустима температура навколишнього середовища	-40°C +40°C	-60 °C +100°C	-60 °C +120°C
Екологічність	так	так	містить ртуть
Індекс передачі кольорів	80	100	80
Вібростійкість	висока	середня	низька

Рисунок Б.6 – Таблиця порівняння

Протоколи зв'язку

7

- Засобом зв'язку було обрано протокол Wi-Fi, тому що це найдоступніший та малозатратний вид зв'язку. В наш час вже в кожному будинку є Інтернет, тому за допомогою Wi-Fi можна легко робити різні маніпулювання в розумному світильнику через смартфон або планшет.

Технологія	Wi-Fi	Bluetooth	ZigBee	Thread
Стандарт зв'язку	IEEE 802.11	IEEE 802.15.4	IEEE 802.15.4	IEEE 802.15.4
Дальність	До 100м	80м	100м	До 30м
Швидкість передачі	300+ Мбіт/с	До 3 Мбіт/с	250 Кбіт/с	250 Кбіт/с
Споживання енергії	Високе	Низьке	Низьке	Низьке
Частотний діапазон	2,4 ГГц	2,4 ГГц	2,4 ГГц	2,4 ГГц
Підтримка IP-технології	+	-	-	+
Топологія	«зірка»	«зірка»	«mesh»	«mesh»

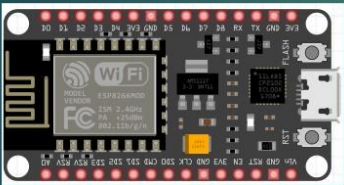
Рисунок Б.7 – Протоколи зв'язку

Вибір мікроконтролера

8

Під обрану задачу було обрано мікроконтролер ESP-12E, а саме плата на його основі – NodeMcu ESP8266. Хоча він має гірші характеристики, але цього достатньо для виконання заданих задач.

Для роботи з мікроконтролером було обрано Arduino IDE.



NodeMcu ESP8266

Технічні характеристики	ESP8266	ESP32
Мікроконтролер	Xtensa Single-Core 32-bit L106	Xtensa Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Так, HT20	Так, HT40
Bluetooth	Немає	Bluetooth 4.2 та нижче
Частота	80 Гц	160 Гц
Пам'ять	160 КБ	512 КБ
Прошивка	SPI до 16 МБ	SPI
GPIO	17	36
Hardware / Software PWM	Немає / 8 Канали	1 / 16 Канали
SPI / I2C / UART	2 / 1 / 2 / 2	4 / 2 / 2 / 2
ADC	10-біт	12-біт
CAN	Не застосовується	1
Ethernet MAC	Не застосовується	1
Interface	Не застосовується	Так
Модуль сенсорного датчика	Не застосовується	Так
Температурний сенсор	Не застосовується	Так
Робоча температура	-40° C – 125° C	-40° C – 125° C

Рисунок Б.8 – Вибір мікроконтролера

Схема збору розумного світильника 9

На рисунку зображена схема збору розумного світильника, де:

VCC – позитивний(червоний);

GND – негативний контакт(синій);

DIN – шина для з'єднання двох контролерів(зелений та фіолетовий).

Розумний світильник складається з: світлодіодної матриці 16x16, пластмасового плафона LLDPE, сенсорної кнопки TP223, блока живлення 5В-3А, комплект дротів, плати NodeMCU ESP8266, для дистанційного керування використовується мобільний застосунок, який з'єднується по протоколу зв'язку Wi-Fi.

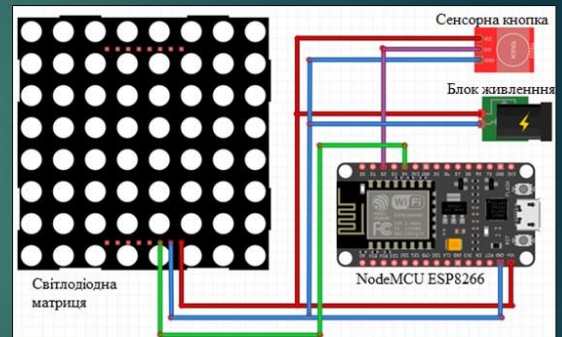


Рисунок Б.9 – Схема збору розумного світильника

Діаграма активності роботи розумного світильника 10

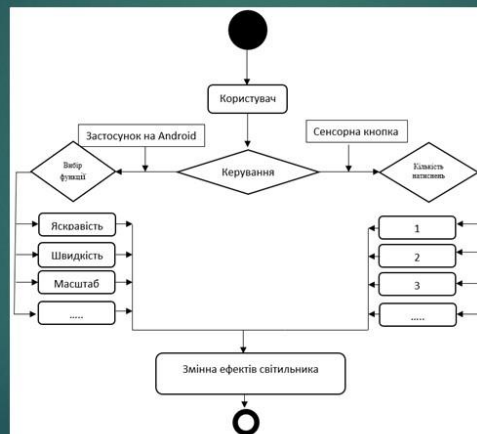


Рисунок Б.10 – Діаграма активності

Файлова структура проєкту та їх методи

11

Структура файлів проєкту на Arduino

Файл	Призначення
ledLamp.ino	Головний файл проєкту
effic.ino	Файл містить список ефектів
effects.ino	Файл містить код ефектів
button.ino	Файл містить налаштування сенсорної кнопки
parsing.ino	Файл містить логіку отримання команд з мобільного застосунку та їх обробку

Методи проєкту на Arduino

Метод	Опис
buttonTick()	Управління розумним світильником за допомогою сенсорної кнопки
ef_Tick()	Вибір ефекту світіння
changePower()	Змінення стану світильника (включення/виключення)
parseUpdate()	Отримання команд з мобільного застосунку та їх обробка
sendCurrent()	Відправка поточного стану та параметрів світильника (яскравість, масштаб, швидкість)
sendIP()	Відправка IP адреси на Android застосунок
saveEEPROM()	Запис до енергонезалежної пам'яті поточного ефекту світильника

Рисунок Б.11 – Файлова структура проєкту та їх методи

Блок-схема отримання команд з мобільного застосунку та їх обробка

12

Даний алгоритм реалізовано у методі parseUpdate()

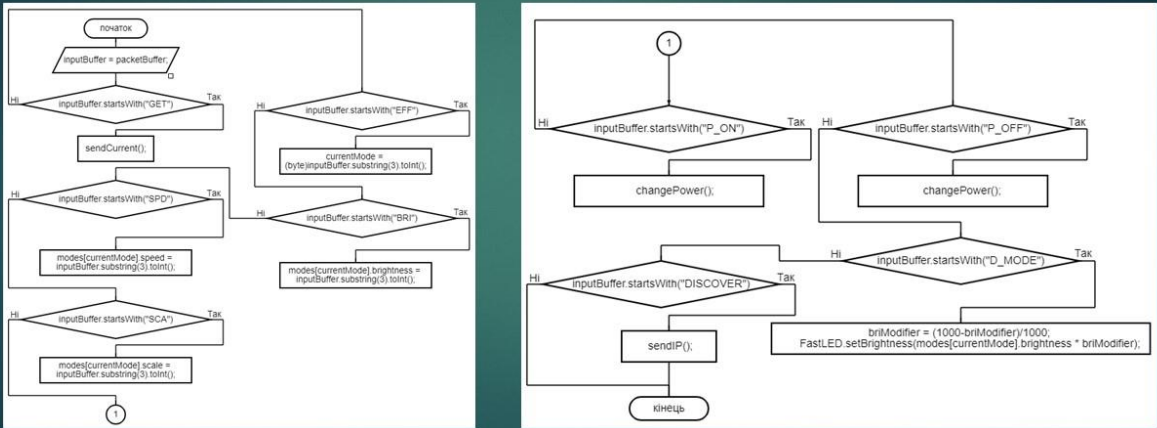


Рисунок Б.12 – Блок-схема

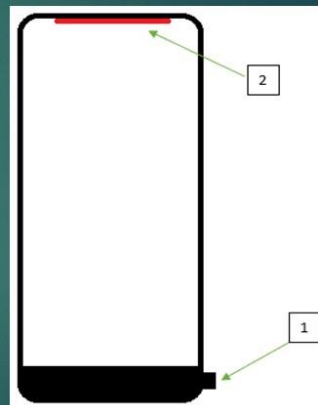
Підключення електроживлення розумного світильника

13

Світильник підключається до мережі електроживлення за допомогою блока живлення.

Параметри блоку живлення:

- ▶ вхідна напруга 220 В змінного струму, частота 50 Гц;
- ▶ вихідна напруга 5 В постійного струму, вихідний струм: щонайменше повинен бути 3 А.



1 - роз'єм живлення, а 2 - сенсорна кнопка

Рисунок Б.13 – Підключення живлення до розумного світильника

Підключення до розумного світильника

14

Після завантаження LedLamp.apk на мобільний застосунок потрібно відразу підключитися до світильника.

Для цього треба натиснути на кнопку підключитися та ввести IP адресу або натиснути на кнопку «Пошук лампи» для пошуку світильника у локальній мережі.

Світильник та мобільний застосунок повинні бути в одній мережі.



Рисунок Б.14 – Підключення до розумного світильника

Приклад роботи розумного світильника

15

Застосунок отримує поточні дані стану світильника (включений/вимкнутий, ефект світіння та його параметри: яскравість, швидкість, масштаб).

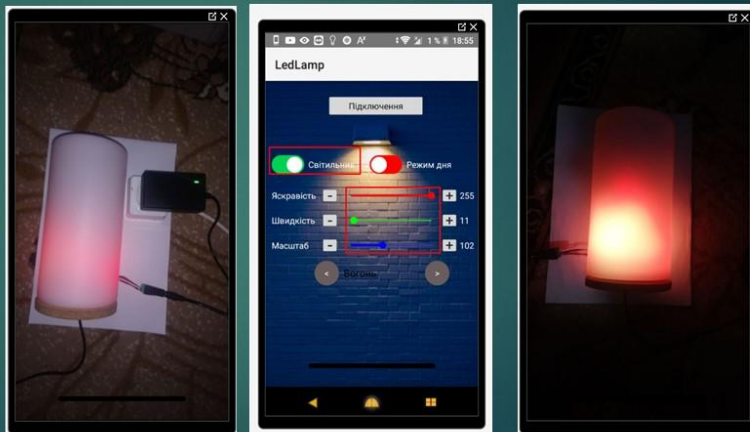


Рисунок Б.15 – Приклад роботи розумного світильника

Імітація даних світлового потоку

15

- Світильник має змогу коригування яскравості на основі даних світлового потоку, які залежать від часу доби.
- Застосунок надає можливість вибрати один з трьох запропонованих режимів доби з заданими даними люменів або самостійно налаштувати цей параметр.
- Це необхідно для регулювання яскравості, наприклад ввечері мало природного освітлення, тому світильник повинен видавати більше освітлення, ніж днем.

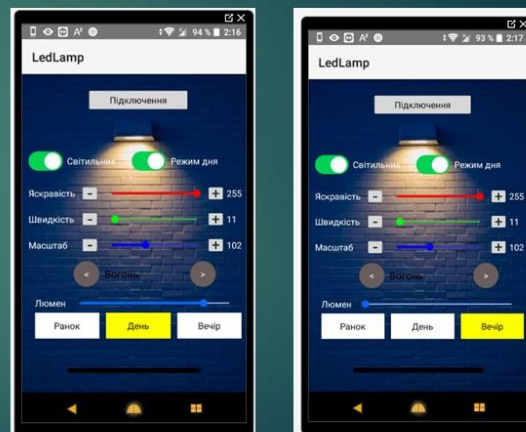


Рисунок Б.16 – Імітація даних світлового потоку

Демонстрація проєкту

17

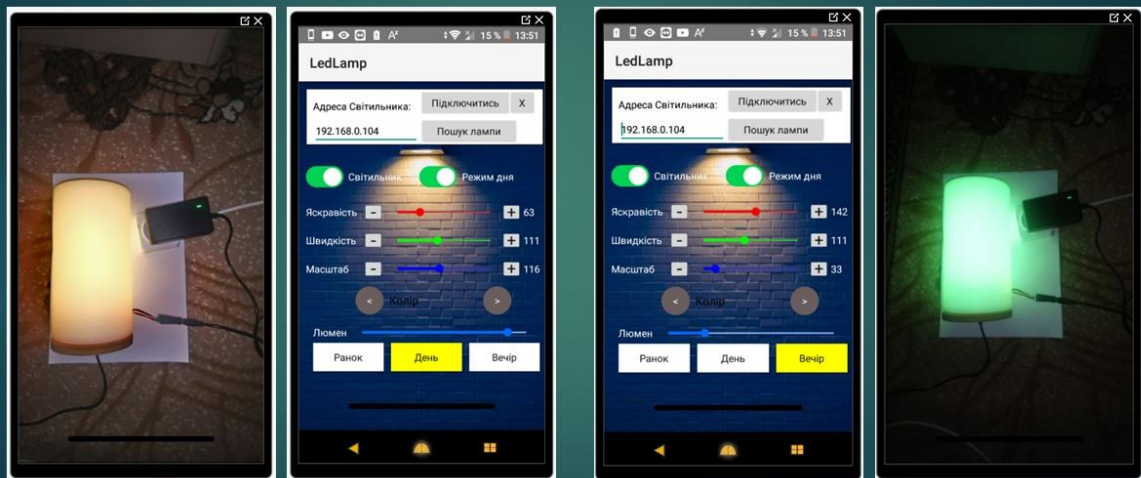


Рисунок Б.17 – Демонстрація проєкту

Висновок

17

В результаті виконання дипломної роботи було розроблено розумний світильник, призначення якого покращення комфорту та атмосфери у приміщенні, скорочення витрат на час керуванням розумним світильником та електроенергії, а також на основі даних світлового потоку коригування яскравості світильника, яка залежить від часу доби. Користувачі зможуть керувати світильником за допомогою сенсорної кнопки або дистанційно через застосунок на смартфоні по мережі Wi-Fi.

Користувачі зможуть керувати світильником за допомогою сенсорної кнопки або дистанційно через застосунок на смартфоні по мережі Wi-Fi.

Рисунок Б.18 – Висновок