

Міністерство освіти і науки України
Відокремлений структурний підрозділ
«ПЕРВОМАЙСЬКИЙ ФАХОВИЙ КОЛЕДЖ»
Національного університету кораблебудування імені адмірала Макарова
відділення «Інформаційних технологій»
Обслуговування комп'ютерних систем
(циклова комісія)

Пояснювальна записка

до дипломного проєкту
фаховий молодший бакалавр
(освітньо-кваліфікаційний рівень)

на тему: «Створення пристрою «Вогняна лампа» на апаратно-програмній
платформі ESP8266»

Виконав: студент 4 курсу, групи 411-KI-20
спеціальності
123 Комп'ютерна інженерія
(шифр і назва спеціальності)

	<u>Гевель Є.Р.</u> (прізвище та ініціали)
Керівник	<u>Чумаченко Н.М.</u> (прізвище та ініціали)
Рецензент	<u>Бесараб І.С.</u> (прізвище та ініціали)

Первомайськ - 2024 року



Ім'я користувача: Наталія Камінська

Дата перевірки:
10.06.2024 08:12:19 EEST

Дата звіту:
10.06.2024 08:27:24 EEST

ID перевірки:
1016340849

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100013810

Назва документа: Гевель Є_P_411-KI-20_2024_плагіат

Кількість сторінок: 44 Кількість слів: 5209 Кількість символів: 37145 Розмір файлу: 3.15 MB ID файлу: 1016142042

Виявлено модифікації тексту (можуть впливати на відсоток

14.8%
Схожість

Найбільша схожість: 9.45% з джерелом з Бібліотеки (ID файлу: 1016126307)

4.47% Джерела з

Сторінка 46

11% Джерела з

Сторінка 47

0% Цитат

Вилучення цитат

Вилучення списку бібліографічних посилань

0%
Вилучень

Немає вилучених

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені

Підозріле

Міністерство освіти і науки України
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ПЕРВОМАЙСЬКИЙ ФАХОВИЙ КОЛЕДЖ»
Національного університету кораблебудування імені адмірала Макарова

Відділення Інформаційних технологій
Циклова комісія Комп'ютерних технологій
Освітньо-професійний ступінь фаховий молодший бакалавр
Галузь знань 12 «Інформаційні технології»
(шифр і назва)
Спеціальність 123 «Комп'ютерна інженерія»
(шифр і назва)
Освітня програма: «Комп'ютерна графіка і web-дизайн»

ЗАТВЕРДЖУЮ
Голова циклової комісії

_____ Островська І.О.
«16» _____ травня _____ 2024 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ ЗДОБУВАЧУ ОСВІТИ

Гевелю Єгору Романовичу
(прізвище, ім'я, по батькові)

1. Тема проєкту «Створення пристрою «Вогняна лампа» на апаратно-програмній платформі ESP8266»

керівник проєкту _____ Чумаченко Наталія Михайлівна,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “12” квітня 2024 року № 46-у

2. Строк подання студентом проєкту _____ 21.06.2024

3. Вихідні дані до проєкту текстові та графічні матеріали контенту:

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1 Дослідження та аналіз вимог до створення пристрою «Вогняна лампа» на апаратно-програмній платформі ESP8266

1.1 Огляд існуючих рішень

1.2 Дослідження апаратно-програмної платформи ESP8266

1.3 Формування вимог до апаратного та програмного забезпечення пристрою «Вогняна лампа»

2 Апаратна реалізація функціональності лампи

2.1 Проєктування структурної схеми пристрою

2.2 Збірка та складання пристрою

2.3 Налагодження пристрою

3 Програмна реалізація функціональності лампи

4 Охорона праці

5. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Чумаченко Н.М.	20.05.24	
2	Чумаченко Н.М.	20.05.24	
3	Чумаченко Н.М.	20.05.24	
4	Чумаченко Н.М.	20.05.24	

6. Дата видачі завдання _____ 20.05.2024 р. _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту	Строк виконання етапів проєкту	Примітка
1	Видача завдання. Узгодження плану роботи	22.05.24	
2	Опрацювання літератури. Структурування дослідження	23.05.24	
3	Робота над загальним розділом	27.05.24	
4	Аналіз існуючих рішень в напрямку розробки	28.05.24	
5	Обґрунтування вибору інструментарію	30.05.24	
6	Робота над спеціальним розділом	06.06.24	
7	Інструкція користувача	13.06.24	
8	Оформлення розділу «Охорона правці»	14.06.24	
9	Оформлення пояснювальної записки	17.06.24	
10	Підготовка презентаційних матеріалів	18.06.24	
11	Оформлення доповіді	19.06.24	
12	Перевірка на плагіат	20.06.24	

Здобувач освіти

(підпис)

Єгор ГЕВЕЛЬ

(ім'я та прізвище)

Керівник проєкту

(підпис)

Наталія ЧУМАЧЕНКО

(ім'я та прізвище)

ЗМІСТ

Вступ	3
1 Дослідження та аналіз вимог до створення пристрою «Вогняна лампа» на апаратно-програмній платформі ESP8266	5
1.1 Огляд існуючих рішень	5
1.2 Дослідження апаратно-програмної платформи ESP8266	8
1.3 Формування вимог до апаратного та програмного забезпечення пристрою «Вогняна лампа»	12
2 Апаратна реалізація функціональності лампи	14
2.1 Проектування структурної схеми пристрою	14
2.2 Збірка та складання пристрою	17
2.3 Налагодження пристрою	20
3 Програмна реалізація функціональності лампи	27
4 Охорона праці	34
4.1 Аналіз умов праці	34
4.2 Гігієна праці та виробнича санітарія	39
4.3 Електробезпека	40
4.4 Пожежна безпека виробничого приміщення	41
4.5 Виконання розрахункової частини	42
Висновок	44
Список літератури	45
Додаток А Інструкція користувача	
Додаток Б Опис програмного коду	

					ВСП ПФК НУК 123.411.24.03 ПЗ									
Змн.	Арк.	№ докум.	Підпис	Дат	Створення пристрою «Вогняна лампа» на апаратно-програмній платформі ESP8266					Літ.	Арк.	Аркушів		
Розроб.	Гевель Є.Р.											2	45	
Перевір.	Островська І.О.									ВСП ПФК НУК 411-КІ				
Рецензен.	Бесараб І.С.													
Н. Контр.	Ткаченко Т.І.													
Затверд.	Островська І.О.													

ВСТУП

Світильники, лампи, ліхтарики, плафони є невід’ємною частиною нашого повсякденного життя і використовуються для освітлення приміщень, вулиць, декоративних цілей та інших потреб. Проте, з розвитком технологій, лампи та світильники перетворилися на більше, ніж просто джерела світла. Сучасні лампи часто поєднують у собі функції освітлення з додатковими можливостями, такими як керування через додатки на смартфоні, зміна кольору світла, імітація різних ефектів та інші цікаві функції.

Одним із таких пристроїв є «Вогняна лампа» – інноваційний світильник, головною особливістю якого є здатність імітувати реалістичний ефект вогняного полум’я за допомогою світлодіодів та програмного забезпечення.

Об’єктом дослідження є технології, інструменти, підходи та варіанти до створення декоративних освітлювальних пристроїв з можливістю імітації вогняного полум’я та реалізації різноманітних світлових ефектів за допомогою програмування та керування світлодіодами.

Предметом даного дипломного проєкту є процес розробки апаратного та програмного забезпечення для створення пристрою «Вогняна лампа» на основі апаратно-програмної платформи ESP8266.

Метою є вивчення та аналіз існуючих рішень для створення ефекту вогняного полум’я, дослідження апаратно-програмної платформи ESP8266, формування вимог до апаратного та програмного забезпечення пристрою «Вогняна лампа» та розробка рекомендацій щодо його реалізації.

У першому розділі розглядаються існуючі рішення для створення ефекту вогняного полум’я, включаючи комерційні продукти та проєкти ентузіастів. Проводиться порівняльний аналіз різних апаратно-програмних платформ, для визначення найбільш оптимального рішення. Формуються детальні вимоги до апаратного та програмного забезпечення пристрою, включаючи необхідні

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дат		

компоненти, функціональність програмного забезпечення та вимоги до корпусу.

Другий розділ присвячений процесу проектування структурної схеми пристрою, вибору компонентів, підключення компонентів, збірки та остаточному складанню готового пристрою в корпус.

В третьому розділі викладено програмну реалізацію функціоналу лампи, а саме створення програмного коду для одного із ефектів лампи.

Останній, четвертий розділ присвячений питанням охорони праці при розробці програмного забезпечення в домашніх умовах. Розглядаються фактори виробничого середовища, вимоги до мікроклімату, освітлення, санітарії та пожежної безпеки. Виконується розрахункова частина щодо відповідності параметрів робочого місця нормативним показникам.

В додатках А та Б надано інструкцію користувача та детальний опис програмного коду одного із ефектів відповідно.

Результатом дипломного проекту буде створення повністю функціонуючого пристрою «Вогняна лампа» на апаратно-програмній платформі ESP8266.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дат		

1 ДОСЛІДЖЕННЯ ТА АНАЛІЗ ВИМОГ ДО СТВОРЕННЯ ПРИСТРОЮ «ВОГНЯНА ЛАМПА» НА АПАРАТНО-ПРОГРАМНІЙ ПЛАТФОРМІ ESP8266

1.1 Огляд існуючих рішень

Одним із представників розумних світильників є GyverLamp («вогняна лампа») – це єдина в своєму роді лампа (рис. 1.1) , що об’єднує в собі світильник, будильник та адресну світлодіодну матрицю. Основною ідеєю є режим імітації полум’я вогню за допомогою матриці адресних світлодіодів. Управління здійснюється за допомогою додатка на смартфоні або сенсорної кнопки на самій лампі.

Так як цей проєкт містить відкритий програмний код, існує велика спільнота людей, які модернізують проєкт, створюють власні варіанти лампи та пишуть власні прошивки. Так, на даний момент, для лампи написано понад десяток прошивок та створено десятки варіантів її реалізації. Наприклад, при прошивці від іншого автора, додається можливість інтеграції з розумним будинком у вигляді додатків Home Assistant та Apple Home Kit.

Популярність лампи зростає і її автор (AlexGyver) написав прошивку GyverLamp2.

GyverLamp2 – прошивка, яку написано з нуля з урахуванням усіх недоліків першої версії. Повністю переосмислено і спрощено підключення, абсолютно всі налаштування робляться зі смартфона, велика кількість нових режимів і можливостей, ОТА-оновлення, стабільна робота і багато іншого. Вартість першої версії лампи – 1000 грн (1650 грн.), другої –1300 грн. [12]

MiJia Bedside Lamp 2 – простий приліжковий нічник (рис. 1.2), який обладнаний різними налаштуваннями яскравості, можливістю вибору кольору світіння, а також декількох режимів. [13]

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дат		



Рисунок 1.1 GyverLamp

Управління здійснюється за допомогою панелі керування, за допомогою смартфона або голосового помічника. Лампа інтегрується в систему розумного будинку та має компактні габарити. Вартість лампи 1899 грн.



Рисунок 1.2 MiJia Bedside Lamp 2

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дат		

WiFi світильник Live Light Magic (FireFly) – український комерційний проєкт світлодіодної лампи повністю оснований на прошивці від GyverLamp2 (рис. 1.3). Управління також здійснюється за допомогою додатка на смартфоні або сенсорної кнопки на самій лампі. Лампа також може інтегруватись в систему розумного будинку. Вартість лампи 1900 грн. [14]



Рисунок 1.3 Live Light Magic (FireFly)

Всі вище описані лампи-світильники об'єднують наступні властивості:

- підтримка Wi-Fi підключення;
- управління зі смартфона;
- управління з кнопки на самій лампі;
- різні режими роботи;
- інтеграція з розумним будинком.

На підставі вищесказаного, можна зробити висновок, що GyverLamp виділяється своєю відкритістю та функціональністю за невелику вартість. Також дану лампу можна зібрати власноруч або за невелику доплату купити готовий до використання пристрій.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дат		

MiJia Bedside Lamp 2 має середню вартість та найменшу функціональність.

Live Light Magic (FireFly) – майже повна копія GyverLamp2 з додаванням деяких незначних особливостей за значно більшу ціну.

На ринку вже існує чимало рішень розумних світильників та ламп, але більшість з них є комерційними продуктами із фіксованими функціями та обмеженими можливостями для налаштування та розширення (такими як MiJia Bedside Lamp 2) або пристроями, заснованими на інших, раніше створених проєктах (такими як Live Light Magic (FireFly)).

Проте, завдяки розвитку відкритих апаратно-програмних платформ, таких як Arduino та ESP8266, з'явилася можливість створювати власні пристрої з різними ефектами, як приклад ефект вогняного полум'я, які можуть бути повністю налаштовані та адаптовані до індивідуальних потреб користувача.

1.2 Дослідження апаратно-програмної платформи ESP8266

Таблиця 1.1 Специфікація ESP8266

Характеристика	Значення
Напруга живлення	3,3 В
Енергоспоживання	10 мка...170 мА
Флеш-пам'ять	До 16 мб максимум (зазвичай 512 кб)
Процесор	Tensilica L106, 32 біта
Швидкість процесора	80...160 МГц
ОЗП	32 кб + 80 кб
Порти введення-виведення загального призначення	17 (мультиплексовані з іншими функціями)
АЦП	1 введення з роздільною здатністю 1024
Підтримка 802.11	b/g/n/d/e/i/k/r
Максимальне число підключень TCP	5

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дат		

ESP8266 – це мікроконтролер, розроблений у 2014 році, який випускає компанія Espressif Systems – китайська компанія з Шанхая. Він являє собою мережеве рішення з Wi-Fi-трансивером, плюс можливість виконання записуваних у його пам'ять додатків.

Мікроконтролер привернув увагу в 2014 році у зв'язку з виходом перших продуктів на його базі за надзвичайно низькою ціною. Специфікацію мікроконтролера наведено в таблиці 1.1 та на рисунку 1.4.

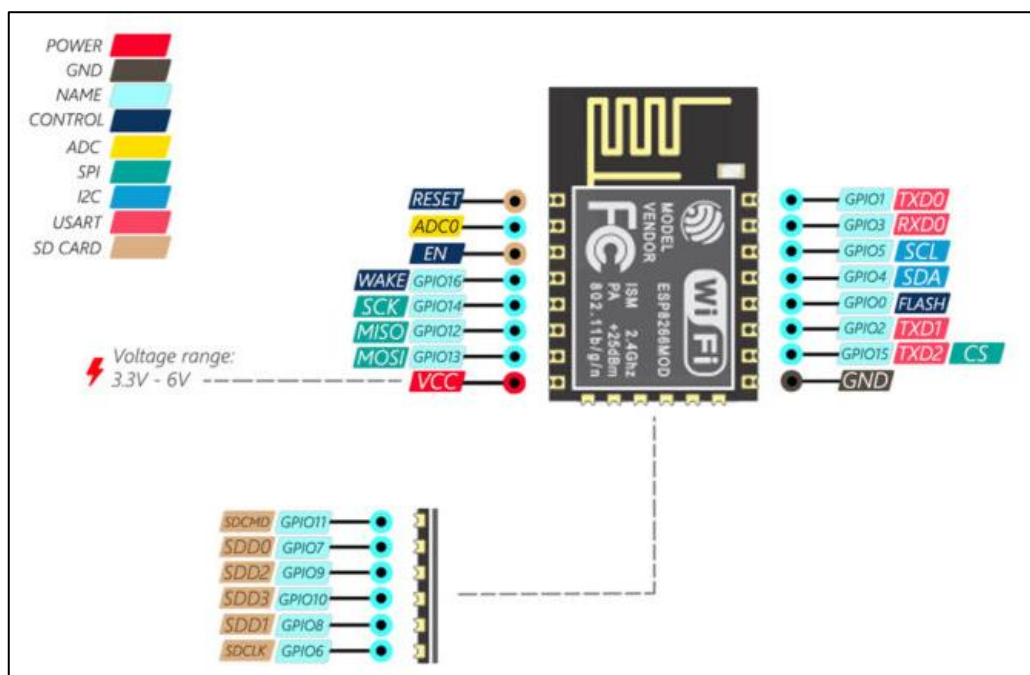


Рисунок 1.4 Схема розташування виводів ESP8266

Завдяки низькому енергоспоживанню та можливості виконувати власні додатки, записані у флеш-пам'ять, ESP8266 швидко здобув популярність серед ентузіастів, винахідників та початківців в області електроніки.

Однак, з плином часу та стрімким розвитком технологій, на ринку з'являлися нові мікроконтролери, які пропонували подібні або навіть кращі характеристики порівняно з ESP8266.

З цією метою доцільно провести порівняльний аналіз параметрів ESP8266 та інших популярних мікроконтролерів (головним критерієм відбору стала

підтримка Wi-Fi), щоб визначити, який з них є найбільш підходящим для реалізації «Вогняної лампи». Порівняльна таблиця (табл. 1.2) дозволить наочно представити відмінності між мікроконтролерами за такими критеріями, як тип та потужність процесора, обсяг оперативної та флеш-пам'яті, підтримка бездротових інтерфейсів, рівень енергоспоживання та вартість.

Для порівняння були обрані: ESP32, Raspberry Pi Pico W, ESP32-C3.

Таблиця 1.2 Порівняння мікроконтролерів

Параметр	ESP32	ESP8266	Raspberry Pi Pico W	ESP32-C3
Процесор	Xtensa LX6Dual-core 32-bit	Tensilica L106 32-bit	Dual-core ARM Cortex-M0+	RISC-V (RV32IMC) 32-bit single-core
Частота процесора	240 МГц	80-160 МГц	133 МГц	160 МГц
Оперативна пам'ять	520 КБ	80 КБ	264 КБ	400 КБ
Флеш-пам'ять	До 16 МБ	До 4 МБ	2 МБ (зовнішня)	До 16 МБ
Підтримка Wi-Fi	802.11 b/g/n	802.11 b/g/n	802.11 n	802.11 b/g/n
Bluetooth	Bluetooth 4.2	Немає	Немає	Bluetooth 5.0 LE
Споживання енергії	Середнє	Низьке	Низьке	Низьке
Ціна	\$4-\$10	\$2-\$5	\$6	\$3-\$5

Поява цих нових мікроконтролерів на ринку створила конкуренцію для ESP8266, змушуючи розробників ретельно оцінювати переваги та недоліки кожного рішення перед тим, як обрати оптимальну платформу для своїх проєктів.

Хоча кожен з цих мікроконтролерів має свої переваги та недоліки, для конкретного проєкту, такого як створення пристрою «Вогняна лампа», важливо обрати найбільш оптимальне рішення, яке відповідатиме вимогам до продуктивності, енергоспоживання, функціональності та вартості.

ESP32 пропонує вищу продуктивність та більше пам'яті, але ці переваги не є критичними для створення «Вогняної лампи». Крім того, ESP32 коштує дорожче та споживає більше енергії.

Raspberry Pi Pico W також є дорожчим варіантом з обмеженою продуктивністю та підтримкою лише однієї смуги Wi-Fi 802.11n.

ESP32-C3 має подібні характеристики до ESP8266, але є менш популярним та ще не має великої екосистеми як у інших мікроконтролерів.

На підставі вищесказаного та поставлених умов можна зробити висновок, що ESP8266 є гарним варіантом для створення пристрою «Вогняна лампа» з наступних причин:

- оптимальна продуктивність: хоча ESP8266 не є найпотужнішим мікроконтролером, його 80-160 МГц процесора та 80 КБ ОЗП цілком достатньо для керування матрицею світлодіодів та створення різних ефектів;
- низьке енергоспоживання: ESP8266 споживає менше енергії, ніж потужніші мікроконтролери, що робить його ідеальним для живлення від батарей або невеликих блоків живлення;
- низька ціна: ESP8266 є одним з найдешевших мікроконтролерів з підтримкою Wi-Fi, що робить його економічно вигідним вибором для створення бюджетних проєктів;
- розвинена екосистема: ESP8266 має величезну спільноту користувачів та розробників, що забезпечує доступність численних бібліотек, прикладів коду та ресурсів для швидкої розробки проєктів.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дат		

Таким чином, завдяки своїй доступності, низькому енергоспоживанню, достатній продуктивності та розвиненій екосистемі, ESP8266 є оптимальним вибором для реалізації пристрою «Вогняна лампа» з точки зору співвідношення ціни та функціональності.

1.3 Формування вимог до апаратного та програмного забезпечення пристрою «Вогняна лампа»

Успіх будь-якого проєкту значною мірою залежить від ретельного планування та чіткого визначення вимог на початковому етапі розробки. Пристрій «Вогняна лампа» не є винятком, і для забезпечення його належного функціонування необхідно сформулювати комплексний набір вимог до апаратного та програмного забезпечення.

Формування вимог є критично важливим кроком, оскільки це визначає основні параметри та характеристики, яким повинен відповідати кінцевий продукт. Тільки після ретельного аналізу потреб користувачів, технічних можливостей та обмежень можна розробити ефективне та якісне рішення.

Апаратні вимоги визначатимуть такі аспекти, як використання мікроконтролера ESP8266 з вбудованим Wi-Fi модулем для забезпечення можливості віддаленого керування. Адресна світлодіодна матриця або стрічка, наприклад WS2812B, для створення реалістичних ефектів. Стабільне джерело живлення 5В. Сенсорна кнопка (типу TTP223) для локального керування режимами роботи лампи.

Програмні вимоги охоплюватимуть функціональність програмного забезпечення, такі як:

- декілька режимів роботи для імітації різних ефектів;
- налаштування швидкості, яскравості та «масштабу» для кожного ефекту;
- налаштування ефектів із додатка на смартфоні;

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дат		

- робота системи як у локальній мережі, так і в режимі «точки доступу»;
- вбудований Wi-Fi менеджер для зручного налаштування мережі;
- керування кнопкою: зміна режиму, налаштування яскравості, увімкнення/вимкнення;

- режим будильник: менеджер будильників на тиждень у застосунку.

При формуванні вимог до корпусу необхідно приділити особливу увагу його функціональним та естетичним характеристикам. Корпус має бути не лише привабливим зовні, а й забезпечувати надійний захист внутрішніх компонентів та ефективно розсіювання світла та тепла.

Отож корпус повинен мати:

- естетичний та сучасний дизайн, що поєднується з різними інтер'єрними стилями;
- захист електронних компонентів від пилу та вологи;
- ефективно розсіювання тепла від світлодіодів для запобігання перегріву;
- ефективно розсіювання світла від світлодіодів для насиченості ефектів;
- можливість зручного доступу до компонентів та сенсорної кнопки.

Формулювання чітких та всебічних вимог дозволить забезпечити, що розроблений пристрій буде зручним у використанні та матиме привабливі візуальні ефекти. Крім того, це полегшить процес розробки та тестування, оскільки всі компоненти та функції будуть визначені заздалегідь.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дат		

2 АПАРАТНА РЕАЛІЗАЦІЯ ФУНКЦІОНАЛЬНОСТІ ЛАМПИ

2.1 Проектування структурної схеми пристрою

Головним компонентом пристрою є адресна світлодіодна матриця на чіпах WS2812B (рис. 2.1), завдяки якій і будуть відображатись всі ефекти.

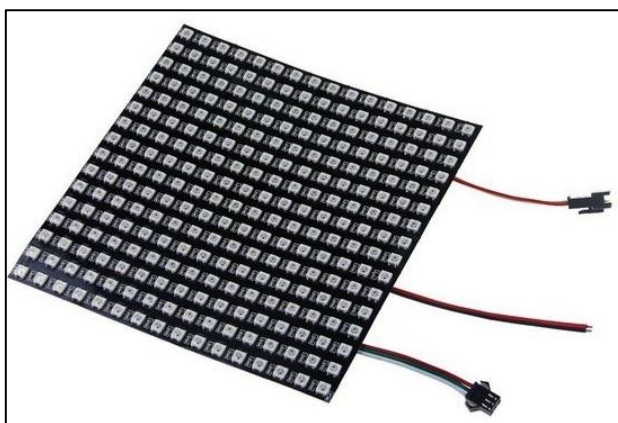


Рисунок 2.1 Матриця WS2812B

«Мозком» пристрою є мікроконтролер ESP8266, реалізований як плата Wemos D1 mini (рис. 2.2). Wemos D1 Mini є мініатюрним Wi-Fi модулем на базі мікроконтролера ESP8266, має вбудований MicroUSB/TypeC-порт із чипом CH340 для зручного підключення до комп'ютера для програмування та зарядки. Також має вбудовану пам'ять Flash для зберігання програмного коду. На платі передбачена кнопка перезавантаження.

Цей модуль підтримує Lua-скриптову мову програмування, а також може бути запрограмований за допомогою Arduino IDE. Він має широкі можливості для під'єднання до різних периферійних пристроїв і сенсорів завдяки роз'ємам GPIO, ADC, I2C, SPI та іншим інтерфейсам.

Wemos D1 Mini є популярним модулем серед спільноти розробників і має багато доступних бібліотек і ресурсів для розроблення різних проектів, як-от IoT-пристрої, домашні автоматизації, контролери освітлення та багато інших.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дат		

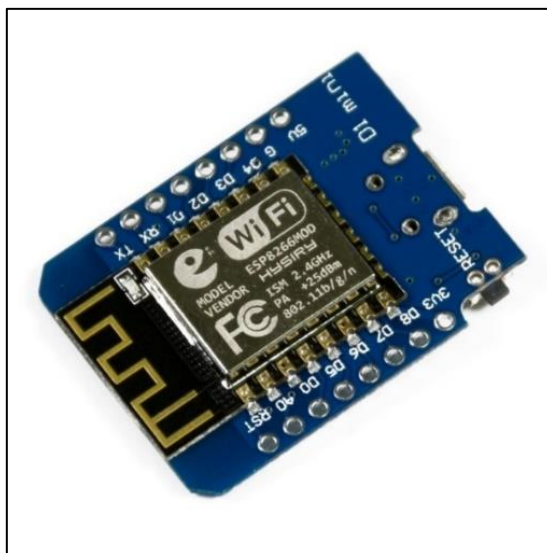


Рисунок 2.2 Плата Wemos D1 mini

Для управління лампою використовується модуль цифрового сенсорного датчика на базі мікросхеми TTP223 (рис. 2.3), який призначений для активації пристроїв за допомогою дотику. За допомогою цього модуля можна замінити звичайні тактові кнопки. Також модуль можна використовувати для керування світильниками, лампами, контролю рівня води в будь-якій ємності тощо. Сенсорний датчик може реагувати як на простий дотик, так і на дотик через скло або пластик (товщиною до 3 мм).

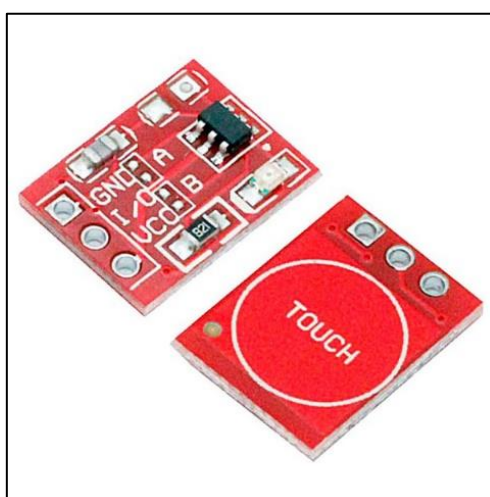


Рисунок 2.3 Плата TTP223

Після того, як було підібрано всі необхідні електронні компоненти для майбутнього пристрою, було зібрано схему (рис. 2.4) підключення компонентів до плати мікроконтролера.

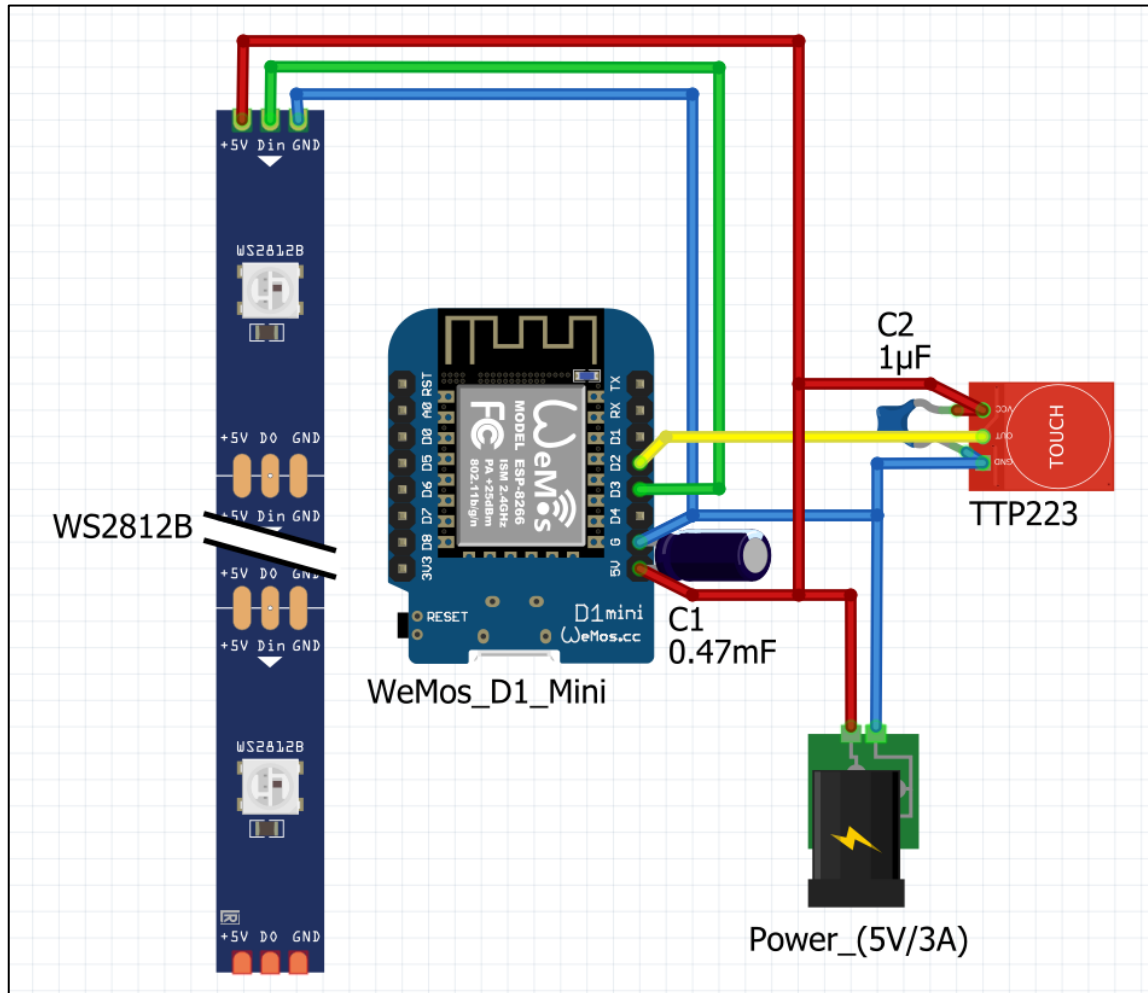


Рисунок 2.4 Схема пристрою

Як видно з рисунку 2.4, схема містить два конденсатори та джерело живлення. Джерелом живлення забезпечує роботу лампи, конденсатори мають своє специфічне призначення.

Конденсатор C1, ємністю 0.47 mF, покращує стабільність живлення схеми та використовується для гасіння перепон та перешкод по 5-ти вольтовій лінії. Конденсатор C2, ємністю 1 uF використовується для гасіння шумів від сенсорної кнопки та має кращу якість передачі сигналу від кнопки до плати мікроконтролера.

2.2 Збірка та складання пристрою

За схемою (рис. 2.5) були з'єднані всі окремі деталі пристрою за допомогою паяльника. Довжина кабелів була оптимально підібрана під корпус лампи.

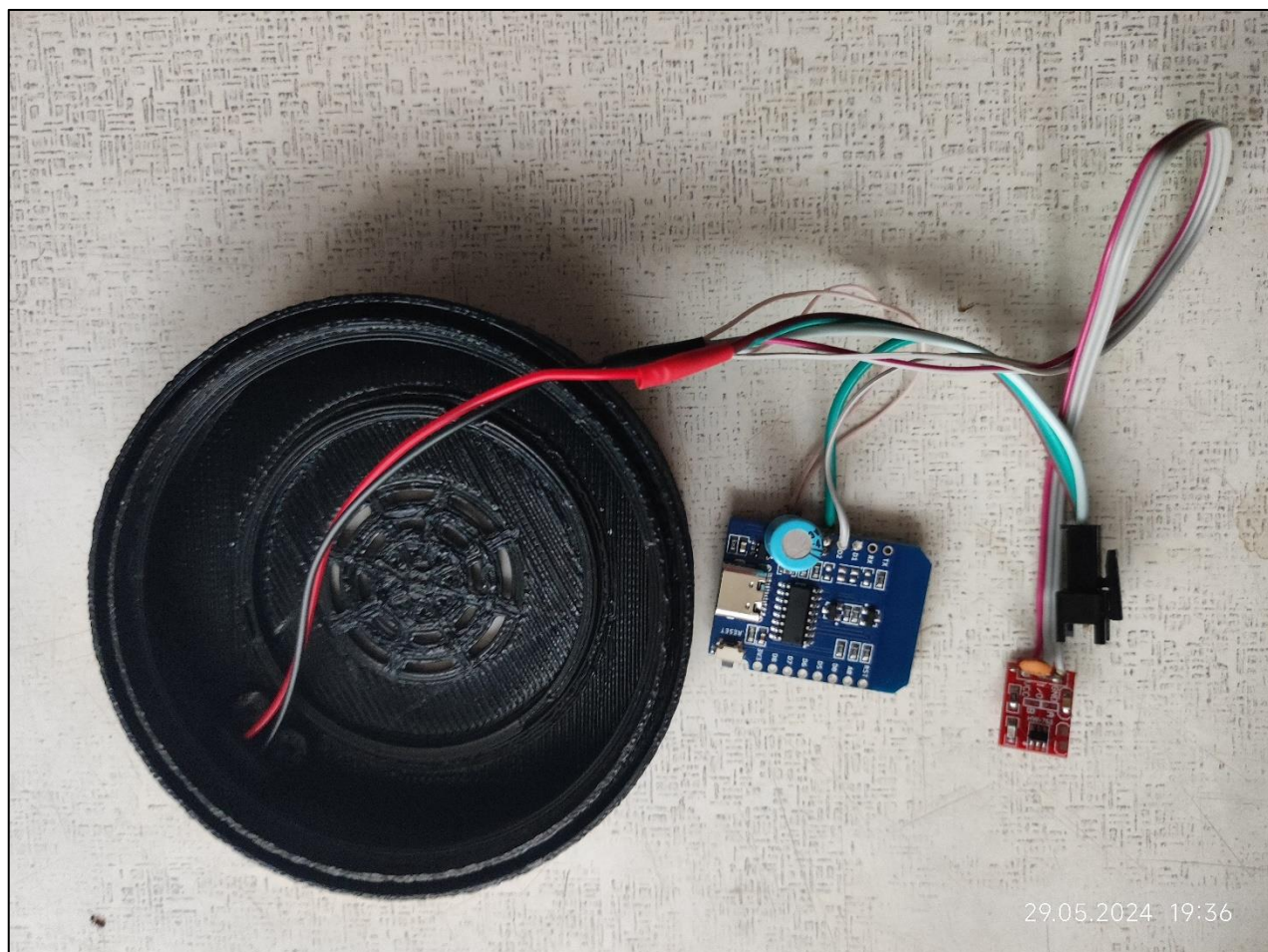


Рисунок 2.5 Зібрана схема

Варіантів створення корпусу більше десятка, це і скляні плафони для світильників, і різні пластикові прозорі циліндри, і друківані на 3D принтері корпуси. Кожен із варіантів має свої плюси та мінуси, але проведення порівняльного аналізу переконало у виборі друківаного корпусу. Оптимальним варіантом став корпус з інтернет магазину, який спеціалізується на 3D друкуванні [7]. Вартість даного корпусу становила 470 грн.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дат		

Корпус складається з наступних частин: плафон, розсіювач, перфорована труба для кріплення матриці, верхня кришка, нижня кришка з віконцем (для доступу до електроніки, зміни схеми або прошивки мікроконтролера) та розділяюча пластина (рис. 2.6).

Корпус має наступні розміри:

- діаметр плафона 90 мм.;
- висота плафона 180 мм.;
- загальна висота лампи 202 мм.;
- діаметр перфорованої труби 50 мм.

Зібрана схема розташовується в нижній частині корпусу (рис. 2.7), кабелі матриці та кнопки протягуються через розділяючу пластину. Після чого лампа остаточно збирається (рис. 2.8).



Рисунок 2.6 Комплектація корпусу

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дат		

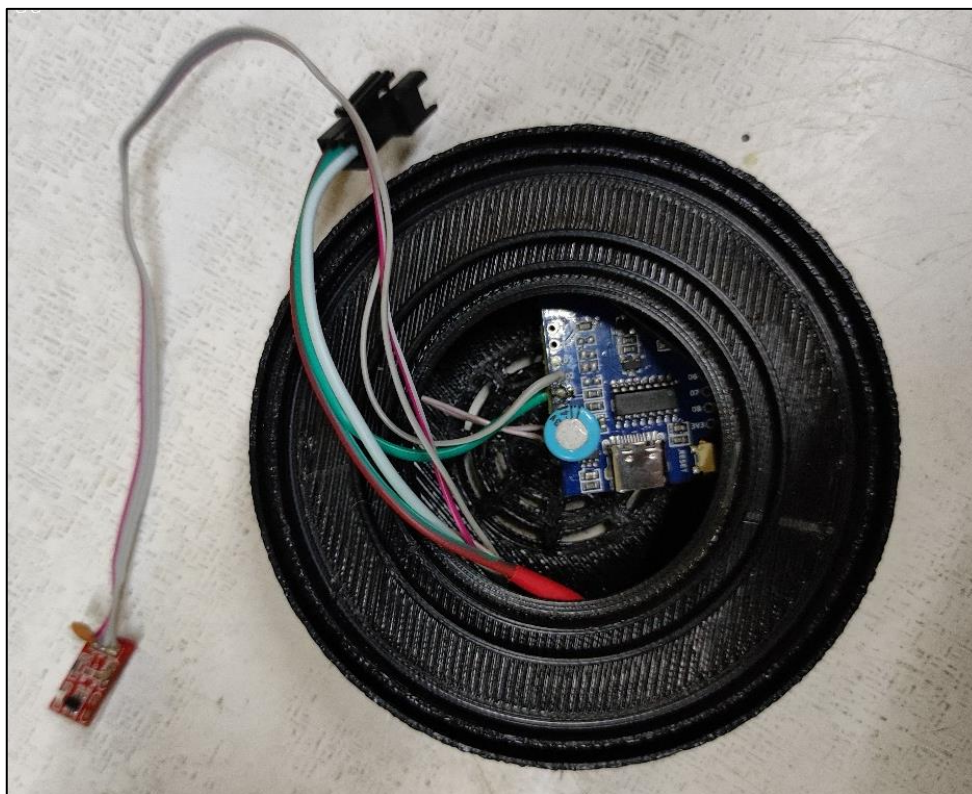


Рисунок 2.7 Розміщення схеми в корпусі



Рисунок 2.8 Зібрана лампа

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дат		

2.3 Налагодження пристрою

Для завантаження, налаштування і редагування прошивки потрібне середовище розробки Arduino IDE.

Arduino IDE – це програмне середовище розробки, що використовує мову програмування C++ і призначене для програмування всіх плат роду Ардуіно-подібних плат (рис. 2.9).

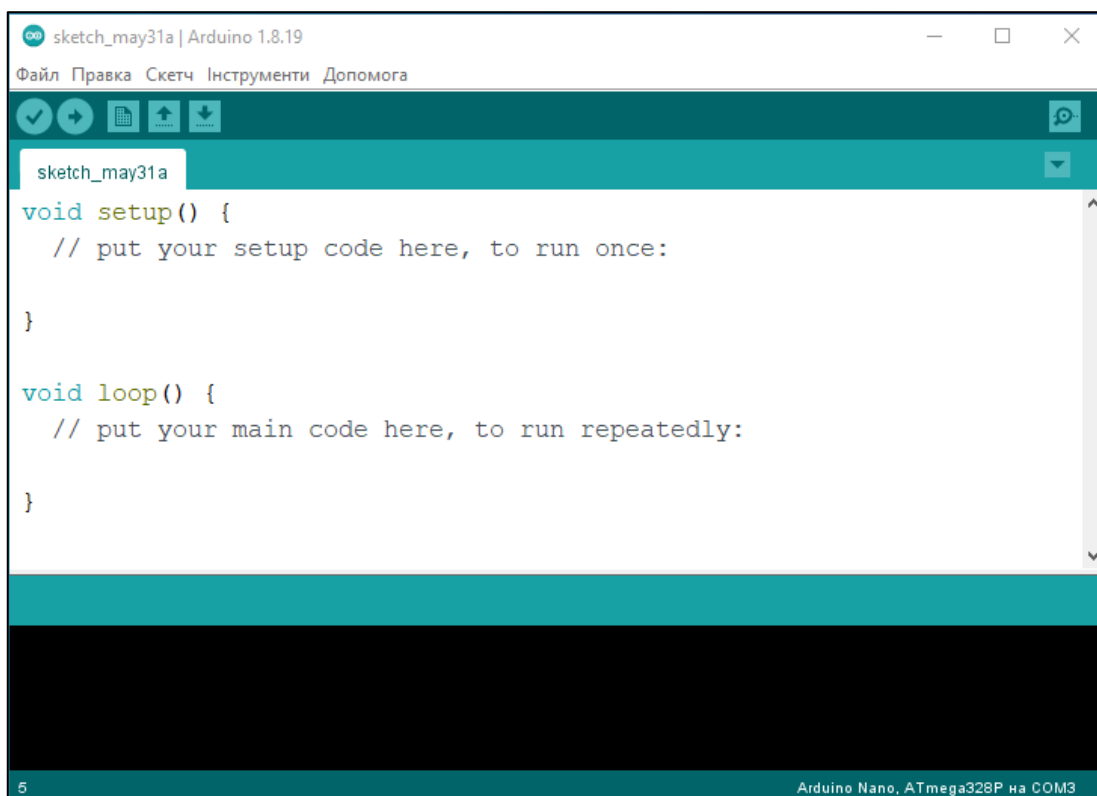


Рисунок 2.9 Arduino IDE

Оригінальні плати від Arduino використовують USB-контролер виробництва FTDI, або окремий мікроконтролер як USB-інтерфейс. Якщо використовувати китайські плати, то знадобиться додатковий драйвер на китайський USB-TTL контролер. Називається він CH340 (CH341), є виробленим у Китаї дешевим аналогом дорогого шведського FTDI чіпа.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дат		

Завантажується драйвер з офіційного сайту [15]. Необхідно запустити файл і у віконці що з'явиться, натиснути INSTALL (рис. 2.10).

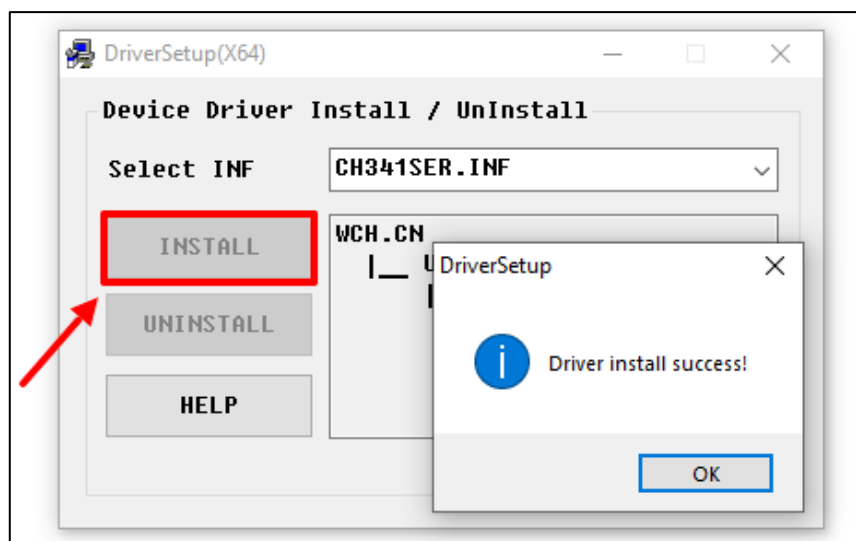


Рисунок 2.10 Встановлення драйвера

З архіву проєкту необхідно додати бібліотеки в папку з програмою для роботи та взаємодії з компонентами схеми. Для цього необхідно з папки бібліотек проєкту, скопіювати папки в папку бібліотек Arduino IDE (рис. 2.11).

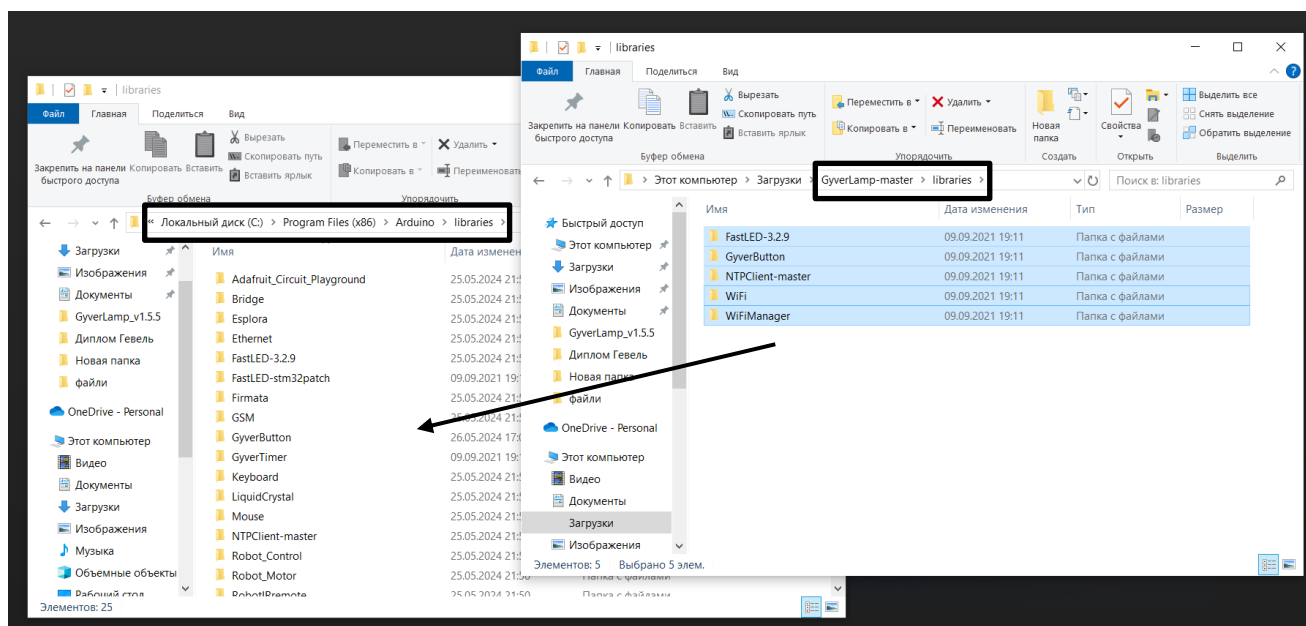


Рисунок 2.11 Копіювання бібліотек

Далі необхідно додати підтримку сімейства плат ESP в менеджер плат. Для цього потрібно зайти в налаштування Arduino IDE, відкрити поле посилань для менеджера плат та вставити посилання на плати (рис. 2.12). Посилання можна взяти в інтернеті або скопіювати з додатків до прошивки.

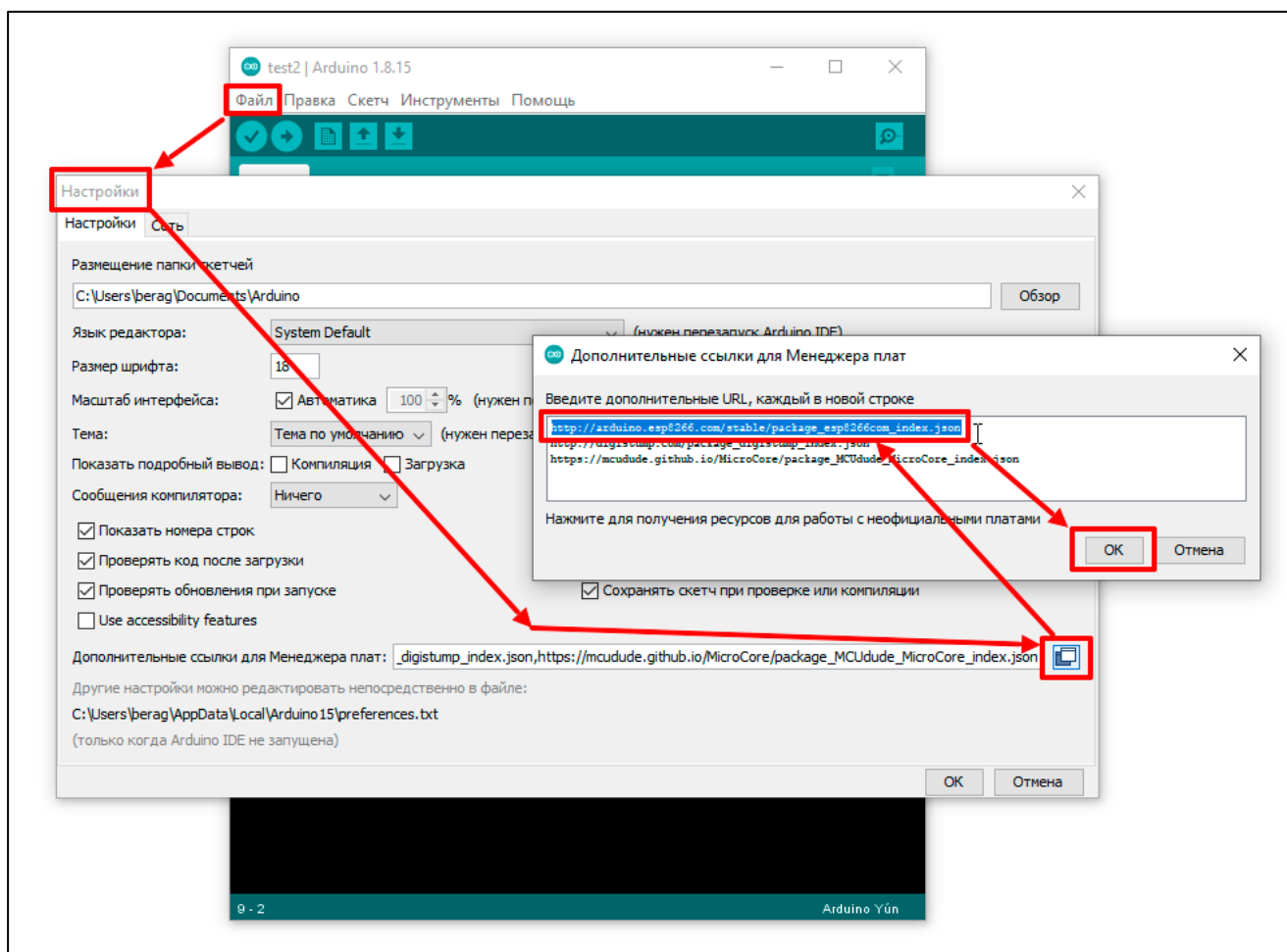


Рисунок 2.12 Додавання плат сімейства ESP до Arduino IDE

Після чого потрібно перейти за шляхом Arduino IDE/Інструменти/Плата/Менеджер плат (рис. 2.13). В полі пошуку набрати «ESP8266». Вибрати та встановити ESP8266 Community (рис. 2.14).

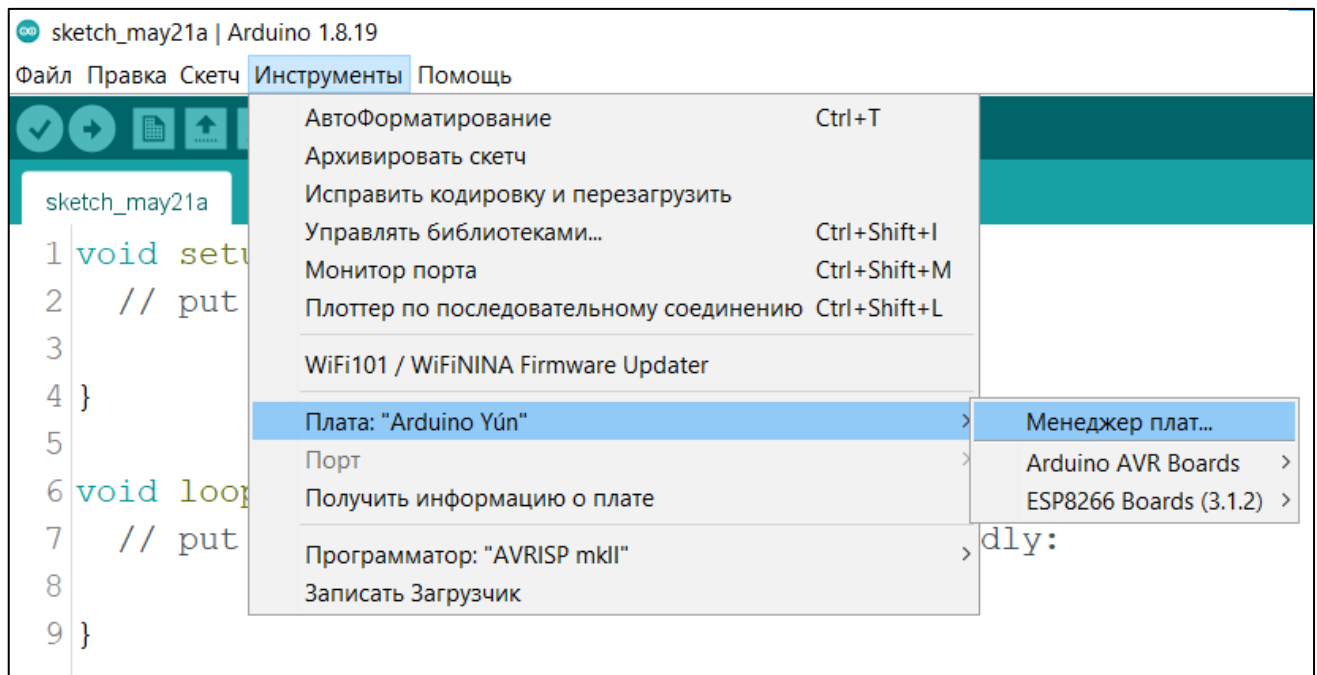


Рисунок 2.13 Перехід в менеджер плат

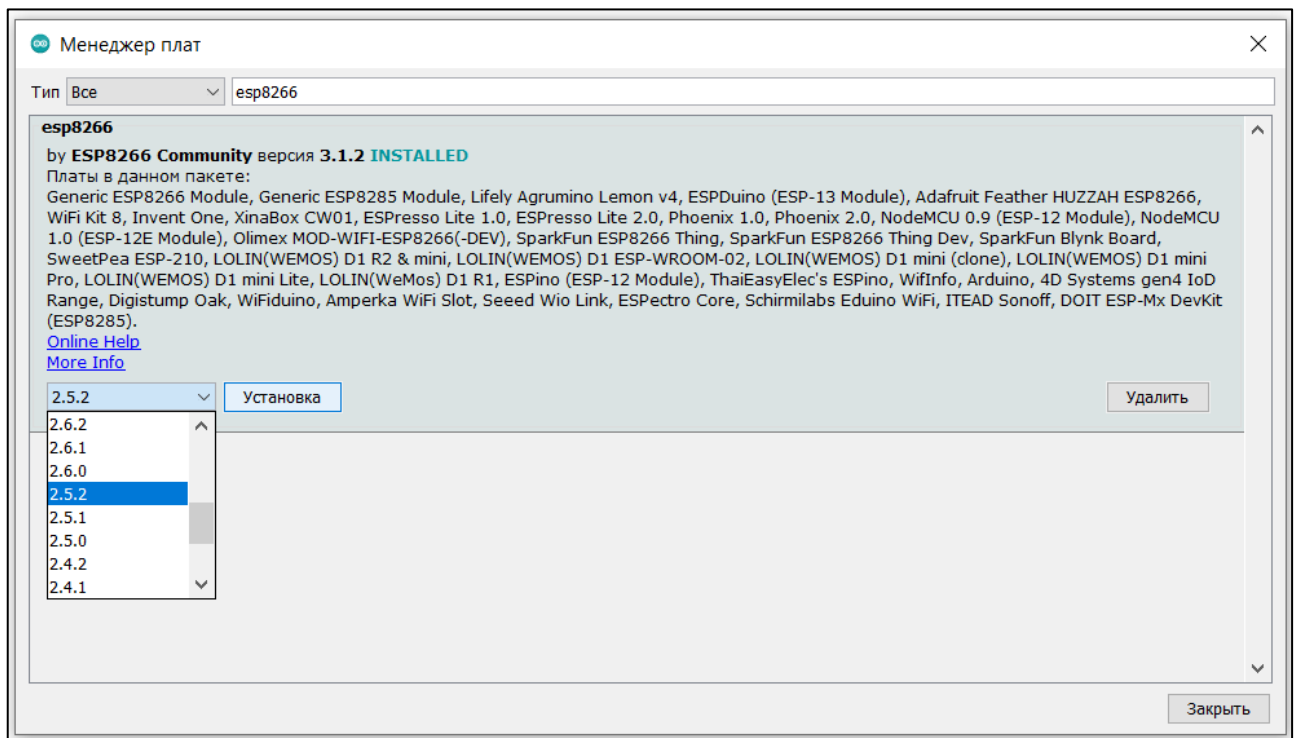


Рисунок 2.14 Встановлення плат в менеджері плат

Далі необхідно вибрати плату. У списку плат Інструменти/Плата/ з'явиться сімейство плат ESP. Вибираємо плату LOLIN Wemos D1 R2 & mini (рис. 2.15).

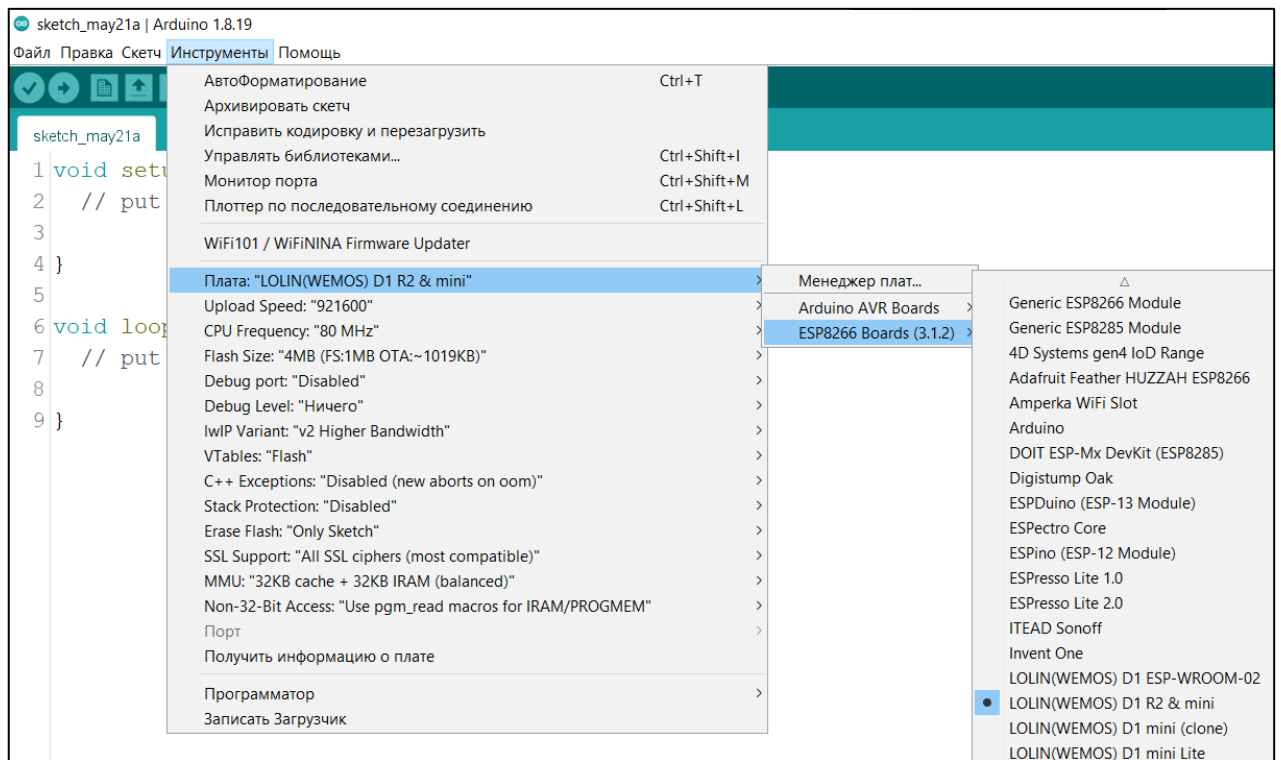


Рисунок 2.15 Вибір плати

У вкладці Інструменти необхідно вказати параметри зображені на рисунках 2.16, 2.17.

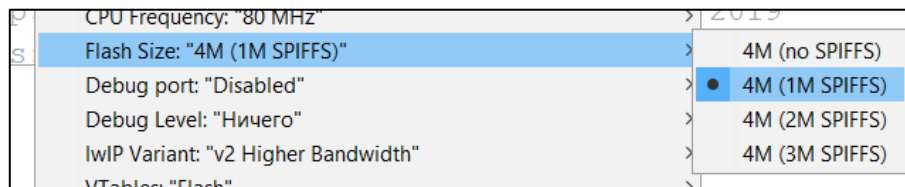


Рисунок 2.16 Перший параметр

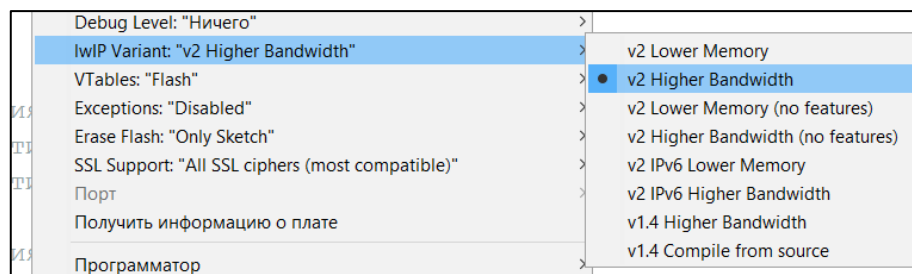


Рисунок 2.17 Другий параметр

Перед підключенням до комп'ютера рекомендується провести візуальний огляд плати на предмет дефектів пайки компонентів. Які дефекти пайки можна зустріти:

- замкнуті піни;
- не припаяна нога компонента;
- компоненти типу резисторів і конденсаторів, що «стирчать» вгору або під кутом, припаяні тільки з одного боку;
- компоненти зі зміщенням;
- капля припою між ногами компонента.

Плату з виявленим дефектом не рекомендується підключати до комп'ютера! Всі дефекти можна виправити паяльником.

Після підключення плати до комп'ютера необхідно вказати порт до якого підключена плата (рис. 2.17).

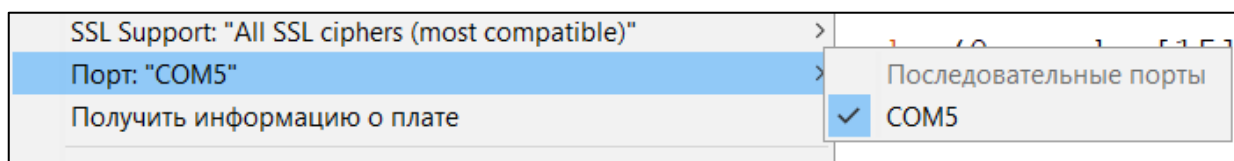


Рисунок 2.17 Вибір порту

В головному скетчі є налаштування, які можна змінювати під власні потреби, також більшість рядків прокоментовано для орієнтування в прошивці (рис. 2.18).

Тепер все готово для завантаження прошивки, можна натискати кнопку Завантаження (рис. 2.19).

Починається процес завантаження прошивки в плату, у нижній частині екрана будуть з'являтися помаранчевий текст з інформацією про процес прошивки та відсотки завантаження (рис. 2.20). У результаті завантаження дійде до 100% і трохи вище буде напис «Завантаження завершено» (рис. 2.21). В додатку А наведено інструкцію користувача для подальшого користування лампою.

```

GyverLamp_v1.5.5  button  eeprom  effectTicker  effects  fonts.h  noiseEffects  parsing  runningText  time  timer2Minim.h  utility
40
41 // ===== НАЛАШТУВАННЯ =====
42 // ----- КНОПКА -----
43 #define USE_BUTTON 1    // 1 - використовувати кнопку, 0 - ні
44
45 // ----- ЧАС -----
46 #define GMT 3
47 #define NTP_ADDRESS "europe.pool.ntp.org"    // сервер часу
48
49 // ----- СВІТАНОК -----
50 #define DAWN_BRIGHT 200    // макс. яскравість світанку
51 #define DAWN_TIMEOUT 1    // скільки світанок сяє після часу будильника, хвилин
52

```

Рисунок 2.18 Частина налаштувань

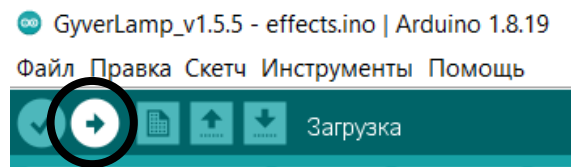


Рисунок 2.19 Кнопка завантаження

```

Завантаження...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 333760 bytes to 237452...
Writing at 0x0000c000... (26 %)

```

Рисунок 2.20 Процес завантаження

```

Завантаження завершено.
Auto-detected Flash size: 4MB
Compressed 333760 bytes to 237452...
Wrote 333760 bytes (237452 compressed) at 0x0
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

```

Рисунок 2.21 Завантаження успішно завершено

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дат		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ФУНКЦІОНАЛЬНОСТІ ЛАМПИ

Прошивка складається з більше ніж десятка файлів. При відкритті головного файлу (скетчу) всі інші підв'язуються до нього і стають вкладками (рис. 3.1). Кожна вкладка відповідає за свою функцію, в свою чергу головний скетч поєднує їх всіх і може звертатись до цих вкладок (скетчів) за необхідними функціями.



Рис. 3.1 Вкладки прошивки

Вкладка button відповідає за опрацювання сигналів (дотиків) від сенсорної кнопки (рис. 3.2). Скетч реалізує обробку взаємодії з користувачем через кнопку для керування.

```
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  }
```

Рисунок 3.2 Частина коду вкладки button

Вкладка eeprom відповідає за збереження налаштувань і даних в енергонезалежній пам'яті EEPROM (рис. 3.3).

```

GyverLamp_v1.5.5  button  eeprom  effectTicker  effects  fonts.h  noiseEffects  parsing  runningText  time  tm
1 void saveEEPROM() {
2   EEPROM.put(3 * currentMode + 40, modes[currentMode]);
3   EEPROM.commit();
4 }
5
6 void eepromTick() {
7   if (settChanged && millis() - eepromTimer > 30000) {
8     settChanged = false;
9     eepromTimer = millis();
10    saveEEPROM();
11    if (EEPROM.read(200) != currentMode) EEPROM.write(200, currentMode);
12    EEPROM.commit();
13  }
14 }
15
16 void saveAlarm(byte almNumber) {
17   EEPROM.write(5 * almNumber, alarm[almNumber].state); // расцвет
18   eeWriteInt(5 * almNumber + 1, alarm[almNumber].time);
19   EEPROM.commit();
20 }
21
22 void saveDawnMmode() {

```

Рисунок 3.3 Частина коду вкладки eeprom

Вкладки effects, effectTicker та noiseEffects відповідають за реалізацію різноманітних візуальних ефектів для матриці світлодіодів, керування візуальними ефектами на матриці світлодіодів, налаштування ефектів та зміну яскравості під час увімкнення або вимкнення світильника (рис. 3.4).

```

GyverLamp_v1.5.5  button  eeprom  effectTicker  effects  fonts.h  noiseEffects  parsing  runningText  time  timer2Minim.h  utility
1 // ===== ЕФЕКТИ =====
2
3 // ----- конфеті -----
4 void sparklesRoutine() {
5   for (byte i = 0; i < modes[0].scale; i++) {
6     byte x = random(0, WIDTH);
7     byte y = random(0, HEIGHT);
8     if (getPixColorXY(x, y) == 0)
9       leds[getPixelNumber(x, y)] = CHSV(random(0, 255), 255, 255);
10  }
11  fader(70);
12 }
13
14 // функція плавного згасання кольору для всіх пікселів
15 void fader(byte step) {
16   for (byte i = 0; i < WIDTH; i++) {
17     for (byte j = 0; j < HEIGHT; j++) {
18       fadePixel(i, j, step);
19     }
20   }
21 }
22 void fadePixel(byte i, byte j, byte step) { // новий фейдер

```

Рисунок 3.4 Частина коду вкладки effects

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дат		

Вкладки `fonts.h` та `runningText` реалізують функціональність для відображення біжучого тексту на матриці світлодіодів (рис. 3.5).

```

GyverLamp_v1.5.5 | button | eeprom | effectTicker | effects | fonts.h | noiseEffects | parsing | runningText | time | timer2Mini
1 // робота з біжучим текстом
2
3 // ***** НАЛАШТУВАННЯ*****
4 #define TEXT_DIRECTION 1 // 1 - по горизонталі, 0 - по вертикалі
5 #define MIRR_V 0 // відобразити текст по вертикалі (0 / 1)
6 #define MIRR_H 0 // відобразити текст по горизонталі (0 / 1)
7
8 #define TEXT_HEIGHT 3 // висота, на якій біжить текст (від низу матриці)
9 #define LET_WIDTH 5 // ширина букви шрифту
10 #define LET_HEIGHT 8 // висота букви шрифту
11 #define SPACE 1 // пропуск
12
13 // ----- ДЛЯ РОЗРОВНИКІВ -----
14
15 int offset = WIDTH;
16 uint32_t scrollTimer;
17
18 void resetString() {
19     offset = WIDTH;
20 }
21
22 boolean fillString(String text, CRGB textColor, boolean clear) {

```

Рисунок 3.5 Частина коду вкладки `runningText`

Вкладка `parsing` забезпечує можливість керування світлодіодним світильником через UDP-пакети, дозволяючи змінювати налаштування ефектів, керувати живленням, налаштовувати будильники (рис. 3.6).

```

GyverLamp_v1.5.5 | button | eeprom | effectTicker | effects | fonts.h | noiseEffects | parsing | runningText |
1 void parseUDP() {
2     int packetSize = Udp.parsePacket();
3     if (packetSize) {
4         int n = Udp.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE);
5         packetBuffer[n] = 0;
6         inputBuffer = packetBuffer;
7
8         if (inputBuffer.startsWith("DEB")) {
9             if (sendSettings_flag) sendCurrent();
10            else inputBuffer = "OK " + timeClient.getFormattedTime();
11            sendSettings_flag = false;
12        } else if (inputBuffer.startsWith("GET")) {
13            sendCurrent();
14        } else if (inputBuffer.startsWith("EFF")) {
15            saveEEPROM();
16            currentMode = (byte)inputBuffer.substring(3).toInt();
17            loadingFlag = true;
18            FastLED.clear();
19            delay(1);
20            sendCurrent();
21            FastLED.setBrightness(modes[currentMode].brightness);
22        } else if (inputBuffer.startsWith("BRI")) {

```

Рисунок 3.6 Частина коду вкладки `parsing`

Вкладки time та timer2Minim відповідають за роботу з часом та клас таймера (рис. 3.7).

```
GyverLamp_v1.5.5  button  eeprom  effectTicker  effects  fonts.h  noiseEffects  parsing  runningText  time  timer2Minim.h  utility
1 byte minuteCounter = 0;
2 CHSV dawnColor;
3
4 void timeTick() {
5     if (ESP_MODE == 1) {
6         if (timeTimer.isReady()) { // кожен секунду
7             secs++;
8             if (secs == 60) {
9                 secs = 0;
10                mins++;
11                minuteCounter++;
12                updTime();
13            }
14            if (mins == 60) {
15                mins = 0;
16                hrs++;
17                if (hrs == 24) {
18                    hrs = 0;
19                    days++;
20                    if (days > 6) days = 0;
21                }
22                updTime();
23            }
24        }
25    }
26}
```

Рисунок 3.7 Частина коду вкладки time

Вкладка utility відповідає за службові функції та деякі налаштування, зокрема налаштування матриці (рис. 3.8).

```
GyverLamp_v1.5.5  button  eeprom  effectTicker  effects  fonts.h  noiseEffects  parsing  runningText  time  timer2Minim.h  utility
1 // службові функції
2
3 void fillAll(CRGB color) {
4     for (int i = 0; i < NUM_LEDS; i++) {
5         leds[i] = color;
6     }
7 }
8
9 // функція відтворення точки за координатами X Y
10 void drawPixelXY(int8_t x, int8_t y, CRGB color) {
11     if (x < 0 || x > WIDTH - 1 || y < 0 || y > HEIGHT - 1) return;
12     int thisPixel = getPixelNumber(x, y) * SEGMENTS;
13     for (byte i = 0; i < SEGMENTS; i++) {
14         leds[thisPixel + i] = color;
15     }
16 }
17
18 // функція отримання кольору пікселя за його номером
19 uint32_t getPixColor(int thisSegm) {
20     int thisPixel = thisSegm * SEGMENTS;
21     if (thisPixel < 0 || thisPixel > NUM_LEDS - 1) return 0;
22     return (((uint32_t)leds[thisPixel].r << 16) | ((long)leds[thisPixel].g << 8) | (long)leds[thisPixel].b);
23 }
```

Рисунок 3.8 Частина коду вкладки utility

Для реалізації можливостей в поставлених вимогах було використано бібліотеки, створено змінні та типи (рис. 3.9, 3.10, 3.11, 3.12).

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дат		

```

89 // ----- БІБЛІОТЕКИ -----
90 #define FASTLED_INTERRUPT_RETRY_COUNT 0
91 #define FASTLED_ALLOW_INTERRUPTS 0
92 #define FASTLED_ESP8266_RAW_PIN_ORDER
93 #define NTP_INTERVAL 60 * 1000 // оновлення (1 хвилина)
94
95 #include "timer2Minim.h"
96 #include <FastLED.h>
97 #include <ESP8266WiFi.h>
98 #include <DNSServer.h>
99 #include <ESP8266WebServer.h>
100 #include <WiFiManager.h>
101 #include <WiFiUdp.h>
102 #include <EEPROM.h>
103 #include <NTPClient.h>
104 #include <GyverButton.h>
105 #include "fonts.h"

```

Рисунок 3.9 Використані бібліотеки

```

117 // ----- ЗМІННІ -----
118 const char* autoConnectSSID = AC_SSID;
119 const char* autoConnectPass = AC_PASS;
120 const char AP_NameChar[] = AP_SSID;
121 const char WiFiPassword[] = AP_PASS;
122 unsigned int localPort = AP_PORT;
123
124 char packetBuffer[UDP_TX_PACKET_MAX_SIZE + 1];
125 String inputBuffer;
126 static const byte maxDim = max(WIDTH, HEIGHT);
127 struct {
128     byte brightness = 50;
129     byte speed = 30;
130     byte scale = 40;
131 } modes[MODE_AMOUNT];
132
133 struct {
134     boolean state = false;
135     int time = 0;
136 } alarm[7];

```

Рисунок 3.10 Змінні

```

107 // ----- ТИПИ -----
108 CRGB leds[NUM_LEDS];
109 WiFiServer server(80);
110 WiFiUDP Udp;
111 WiFiUDP ntpUDP;
112 NTPClient timeClient(ntpUDP, NTP_ADDRESS, GMT * 3600, NTP_INTERVAL);
113 timerMinim timeTimer(1000);
114 timerMinim timeStrTimer(120);
115 GButton touch(BTN_PIN, LOW_PULL, NORM_OPEN);

```

Рисунок 3.11 Типи

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дат		

```

138 const byte dawnOffsets[] = {5, 10, 15, 20, 25, 30, 40, 50, 60};
139 byte dawnMode;
140 boolean dawnFlag = false;
141 float thisTime;
142 boolean manualOff = false;
143 boolean sendSettings_flag = false;
144
145 int8_t currentMode = 0;
146 boolean loadingFlag = true;
147 boolean ONflag = true;
148 uint32_t eepromTimer;
149 boolean settChanged = false;
150
151 unsigned char matrixValue[8][16];
152 String lampIP = "";
153 byte hrs, mins, secs;
154 byte days;
155 String timeStr = "00:00";

```

Рисунок 3.12 Додаткові змінні

Також було реалізовано створення програмного коду для одного із режимів роботи «Вогняної лампи».

Ефект «Снігопад» імітує падіння сніжинок, створюючи ілюзію снігопаду (рис. 3.13).



Рисунок 3.13 Ефект «Снігопад»

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дат		

Код складається з однієї функції `snowRoutine()`. Ця функція відповідає за обчислення та відображення положення сніжинок на віртуальному полі, яке представлено двовимірним масивом пікселів. Функція використовує вкладені цикли та умовні конструкції для управління рухом існуючих сніжинок та генерації нових.

Перший вкладений цикл `for` проходить по всіх пікселях віртуального поля, починаючи з верхнього лівого кута, і переміщує значення кольору кожного пікселя на один рядок вгору. Це створює ефект руху сніжинок згори вниз, імітуючи падіння.

Другий вкладений цикл `for` проходить по нижньому рядку віртуального поля. Якщо в передостанньому рядку немає сніжинки (значення кольору дорівнює 0) і випадкове число відповідає певному критерію, то в нижньому рядку генерується нова сніжинка зі світло-блакитним кольором з різними відтінками. Це досягається за допомогою функції `random()`, яка забезпечує варіацію кольорів сніжинок. В іншому випадку, в нижньому рядку встановлюється чорний колір (0x000000), що імітує відсутність сніжинки.

Таким чином, код `snowRoutine()` забезпечує динамічне відображення падіння сніжинок на віртуальному полі, створюючи реалістичний ефект снігопаду з різноманітними відтінками сніжинок.

Детальний опис програмного коду наведений в додатку Б.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дат		

4 ОХОРОНА ПРАЦІ

4.1 Аналіз умов праці

Під час проходження переддипломної практики було проведено аналіз умов праці в робочому середовищі, де розроблявся пристрій «Вогняна лампа».

У просторій житловій кімнаті площею 20 м² та стелею висотою 2,6 метра облаштоване робоче місце, яке обладнане комп'ютерним столом висотою 74 см (рис. 4.1).

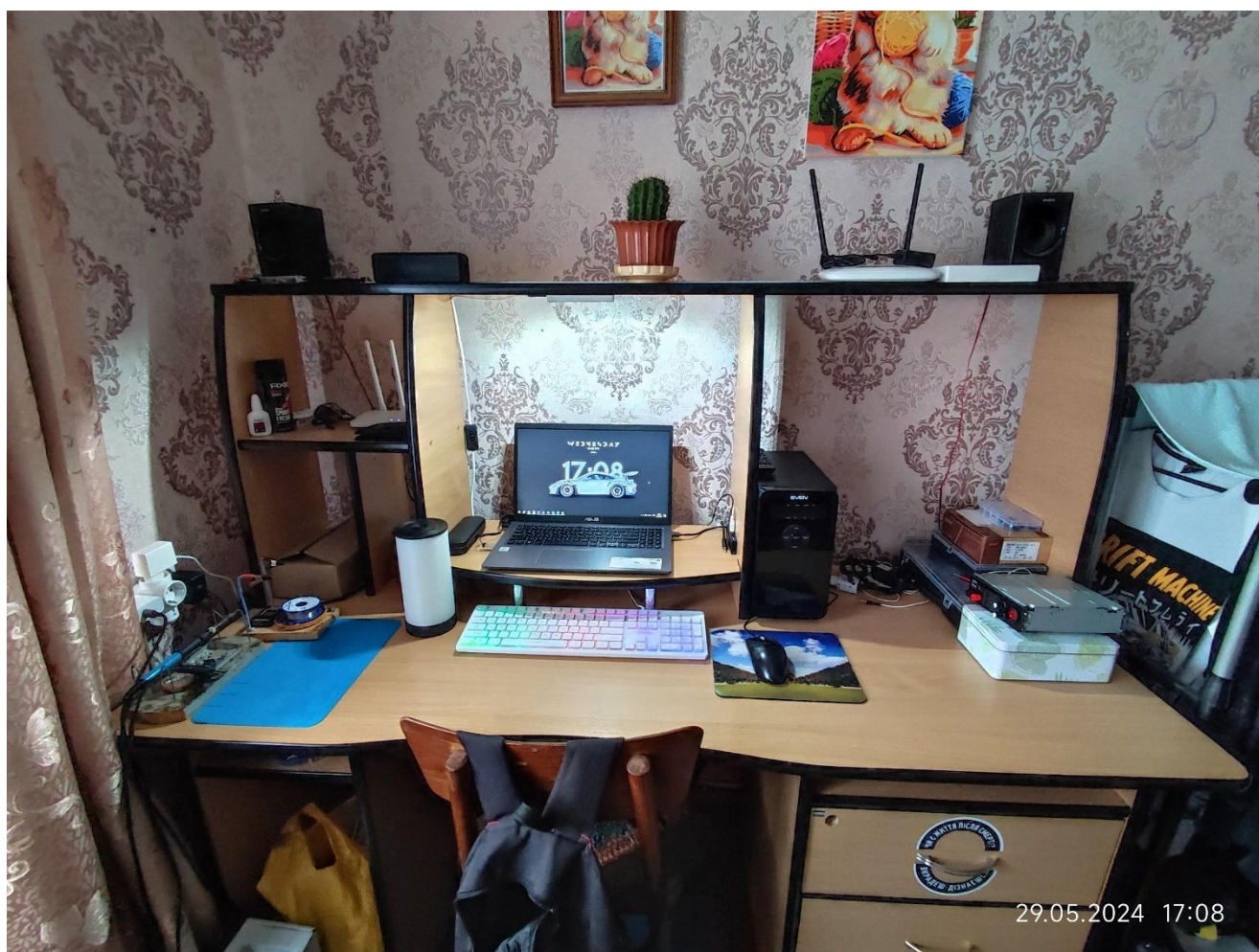


Рисунок 4.1 Робоче місце

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дат		

Робочий стіл має розміри 62,5 см в глибину та 1,6 метра в довжину, який дозволяє розмістити все необхідне обладнання для створення пристрою «Вогняна лампа» та ноутбук.

Враховуючи санітарно-гігієнічні норми монітор ноутбука (рис. 4.2) знаходиться на відстані 65 см від очей.

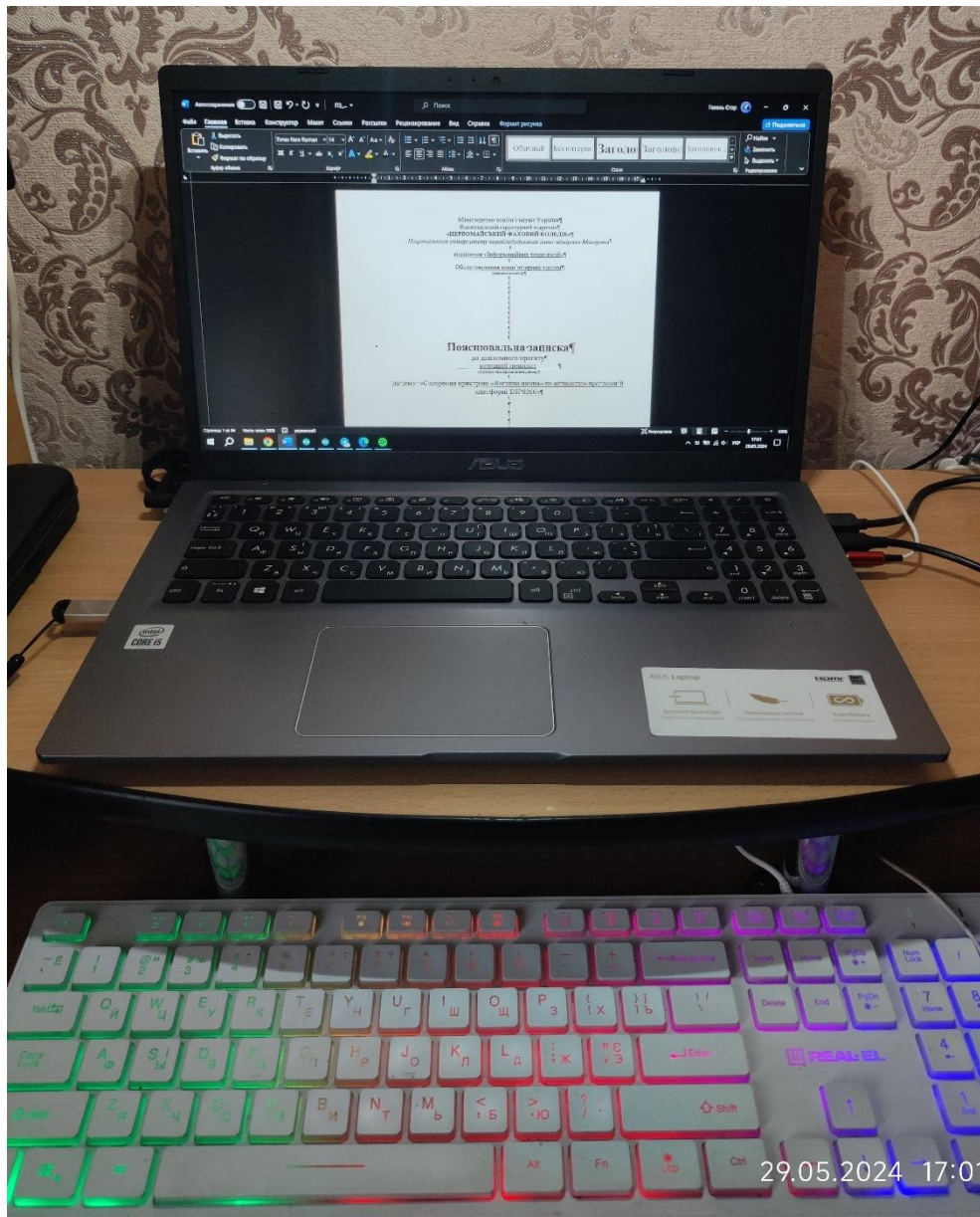


Рисунок 4.2 Ноутбук

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		37

На робочому столі знаходяться термостійкий коврик (рис. 4.3), паяльник з підставкою (рис. 4.4), підставка для паяння з матеріалом для збирання лампи, а саме – припій та флюс, що зображені на рисунку 4.5.



Рисунок 4.3 Термостійкий коврик



Рисунок 4.4 Паяльник з підставкою

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дат		



Рисунок 4.5 Матеріали для паяння

У кімнаті є три вікна, що створюють належний рівень освітлення протягом денного часу, як показано на рисунку 4.6. Для роботи у вечірні години передбачено додаткове підсвічування світлодіодною стрічкою робочої зони, зображеною на рисунку 4.7, а також загальне освітлення самої кімнати (рис. 4.8).

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дат		



Рисунок 4.6 Вікно

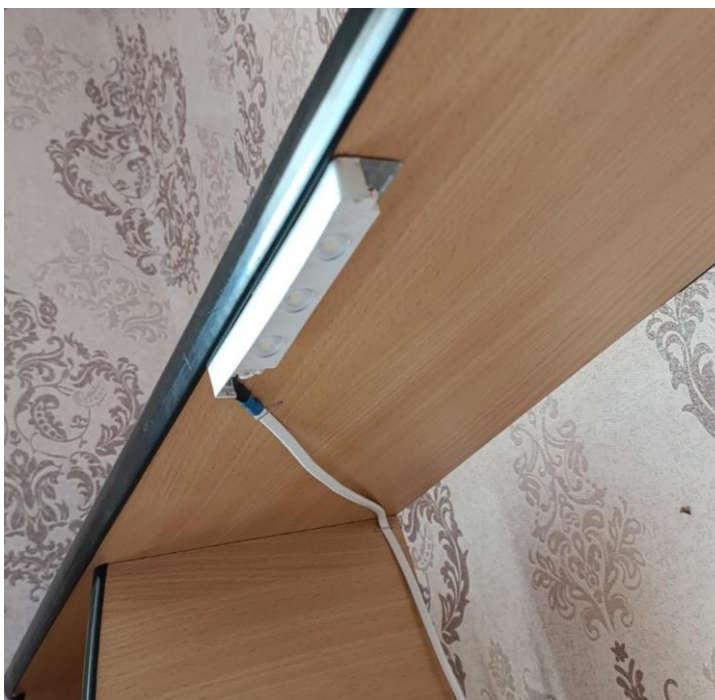


Рисунок 4.7 Світлодіодна стрічка

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дат		



Рисунок 4.8 Люстра

Стілець відповідає санітарно-гігієнічним нормам. Висота стільця до сидіння – 45 см, спинка стільця висотою – 40 см та площа сидіння стільця розміром 40 см², які дозволяють мінімізувати навантаження на тіло під час виконання робіт.

4.2 Гігієна праці та виробнича санітарія

Під час роботи в домашніх умовах важливо створити безпечне робоче місце, для дотримання принципів гігієни та чистоти.

Потрібно ретельно піклуватись про якість повітря у робочій кімнаті та забезпечити належну вентиляцію. Для цього потрібно регулярно провітрювати приміщення, відчиняти вікна та двері для постійного оновлення повітряних мас та усунення забруднення від накопичення парів паяння, пилу чи інших джерел.

Приділити увагу для підтримання оптимального температурного режиму в комфортних межах 20-24°C. Це забезпечить сприятливі умови праці.

Необхідно дбати про якісне освітлення робочої зони, поєднувати природне світло з штучними джерелами освітлення. Це допомагає запобігти передчасній втомі очей, головним болям та іншим негативним наслідкам.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дат		

Обов'язковою процедурою є регулярне вологе прибирання у кімнаті та на робочому місці.

Дбати про особисту гігієну, підтримувати чистоту не лише робочого місця, а й тіла та власного одягу. Робити регулярні перерви для виконання гігієнічних процедур.

Дотримання всіх правил гігієни праці та створення належних санітарних умов, допоможуть підтримувати здорове та продуктивне робоче середовище під час роботи, без завдання шкоди власному самопочуттю та здоров'ю.

4.3 Електробезпека

Забезпечення електробезпеки при паянні в домашніх умовах є надзвичайно важливим питанням, адже недотримання основних правил безпеки може призвести до нещасних випадків або навіть пожеж.

Потрібно забезпечити належну вентиляцію в будинку, відкрити вікна або використати витяжну систему для належного повітрообміну.

Підготувати негорючу, стійку робочу поверхню, наприклад, стіл з термостійким ковриком.

Необхідно використовувати якісне паяльне обладнання від перевірених виробників. Потрібно переконатися, що паяльна станція має захист від перегріву та перенапруги. Регулярно перевіряти стан кабелю живлення та паяльного жала на предмет пошкоджень, тріщин або зносу ізоляції.

Не торкатися гарячим паяльним жалом до легкозаймистих матеріалів, таких як папір, тканина або пластик. Це може спричинити займання або пожежу. Не залишати ввімкнену паяльну станцію без нагляду, оскільки жало може перегрітися та стати причиною загоряння. Після закінчення роботи потрібно вимкнути паяльну станцію та дати їй охолонути перед прибиранням.

Не торкатися оголеними руками до свинцевого припою або його парів, оскільки свинець є токсичною речовиною.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дат		

Використовувати захисні рукавички, окуляри та щипці для роботи з припоєм. Після паяння ретельно вимити руки з милом, щоб видалити залишки припою.

Дотримуючись цих простих правил електробезпеки, можна забезпечити безпечні умови праці для уникнення нещасних випадків під час паяння в домашніх умовах.

4.4 Пожежна безпека виробничого приміщення

Забезпечення пожежної безпеки в приватному будинку, де проводиться робота з паяння є важливим завданням для запобігання виникнення пожежі.

Необхідно дотримуватись правил пожежної безпеки під час використання електричного обладнання, а саме: справність розеток, електропроводки паяльника та відсутність оголених дротів, що можуть спричинити іскріння. обережно поводитися з відкритим джерелом вогню, гарячими поверхнями та потенційним джерелом запалювання, слідкувати щоб біля робочого столу де проводяться роботи з паяння не знаходились легкозаймисті речі.

Якщо під час паяння в будинку виникла пожежа, необхідно негайно відключити паяльник від електромережі, спробувати загасити невелику пожежу накривши осередок загоряння щільною ковдрою, евакуювати з будинку людей. Якщо не вдається загасити пожежу, викликаємо пожежну службу за номером 101. Після виходу з будинку не повертатися всередину через небезпеку задимлення і розповсюдження вогню. Після прибуття пожежників надати їм максимально інформації про місце виникнення і можливі причини загоряння.

Повне виконання правил пожежної безпеки та наявність відповідних засобів для гасіння вогню, дозволять значно знизити ймовірність виникнення пожежі та гарантувати належний рівень захисту під час роботи вдома.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дат		

4.5 Виконання розрахункової частини

Забезпечення безпечних та комфортних умов праці було ключовим пріоритетом під час розробки пристрою «Вогняна лампа» у домашньому середовищі. Для досягнення цієї мети було проаналізовано параметри робочого місця та виробничого середовища і проведено порівняння їх з чинними санітарно-гігієнічними нормами.

Таблиця Показники робочого місця та параметри середовища

Фактор виробничого середовища	Санітарно-гігієнічні норми	Існуючі показники	Висновок
Температура	19-25°C	21 °C	В межах норми
Відносна вологість	40-60%	47%	В межах норми
Освітленість робочої поверхні	Не менше 300 Лк	450 Лк	Норма
Відстань від очей до монітора	600-700 мм	650 мм	В межах норми
Висота столу	680-720 мм	740 мм	Вище норми
Ширина столу	900 мм	625 мм	Нижче норми
Довжина столу	1600 мм	1615 мм	В межах норми
Висота стільця	400-500 мм	400 мм	В межах норми
Розміри сидіння стільця	380-400 мм	400 мм	В межах норми
Площа на одного працівника	6м ² (мінімально 4,5м ²)	10 м ²	Вище норми

Дотримання нормативних вимог дозволило створити сприятливі умови для тривалої роботи розробника, захистивши його від шкідливих виробничих факторів та мінімізувавши ризики виникнення професійних захворювань чи травм. Це було досягнуто шляхом аналізу та коригування ключових показників робочого простору та виробничого середовища.

Детальні розрахунки основних параметрів, таких як освітлення, мікроклімат, ергономіка робочого місця та інші, були представлені в таблиці. Отримані результати були зіставлені з припустимими нормативними значеннями, що дозволило виявити потенційні відхилення та вжити необхідних заходів для їх усунення.

Завдяки дотриманню санітарно-гігієнічних стандартів, вдалося створити оптимальні умови праці для розробника, забезпечивши йому комфортне та безпечне робоче середовище під час створення пристрою «Вогняна лампа».

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дат		

ВИСНОВОК

У ході дипломного проектування було проведено дослідження та аналіз розробки пристрою «Вогняна лампа» на апаратно-програмній платформі ESP8266.

В ході дипломного проектування було проведено огляд існуючих рішень декоративного освітлення та обґрунтовано вибір апаратної платформи ESP8266, сформовані технічні вимоги та розроблена структурна схема пристрою. Було описано процес збірки, складання та налагодження апаратної та програмної частини пристрою з налаштуваннями та тестуванням. Розглянуто питання організації безпечних умов праці при розробці прикладного програмного забезпечення. Також було надано інструкцію користувача по експлуатації готового пристрою та описано програмний код до одного із ефектів лампи.

На основі результатів досліджень було розроблено пристрій «Вогняна лампа» на апаратно-програмній платформі ESP8266. Пристрій успішно імітує реалістичні ефекти, зокрема ефект вогняного полум'я, за допомогою світлодіодів та програмного забезпечення.

Результатом дипломного проекту є створення повністю функціонуючого пристрою «Вогняна лампа», який може бути використаний для декоративного освітлення приміщень, створення атмосфери затишку та комфорту, а також для інших цілей.

Мету дипломного проектування досягнуто.

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дат		

СПИСОК ЛІТЕРАТУРИ

1. Бельська М.В. Дипломне проектування. Методичні вказівки до виконання розділу дипломного проекту «Організація охорони праці та протипожежний захист». – Первомайськ, 2023 – 16 с.
2. <https://github.com/AlexGyver/GyverLamp/>
3. <https://community.alexgyver.ru/threads/wifi-lampa-budilnik.1406/>
4. <https://uk.wikipedia.org/wiki/ESP8266>
5. <https://arduino.ua/ua/cat122-esp8266>
6. <https://uk.wikipedia.org/wiki/ESP32>
7. <https://electrochip.mk.ua/ru/product/korpus-dlya-led-lampy-lampy-gajvera-gyver-lamp-v3-0/>
8. <https://habr.com/ru/articles/541874/>
9. <https://electropeak.com/learn/esp8266-pinout-reference-how-to-use-esp8266-gpio-pins/>
10. <https://www.raspberrypi.com/products/raspberry-pi-pico/>
11. <https://www.youtube.com/watch?v=771-Okf0dYs>
12. <https://electrochip.mk.ua/ru/product/nochnyk-svetylnyk-led-lamp-wifi-ognennaya-lampa-lampa-gajvera/>
13. <https://mi-store.pl/product-ukr-1993-Xiaomi-Mi-Bedside-Lamp-2-Colorful-RGBW.html>
14. <https://cck.com.ua/ua/wifi-svetilnik-firefly/?sku=1647>

					ВСП ПФК НУК 123.411.24.03 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дат		

Додаток А Інструкція користувача

Щоб заживити лампу потрібен блок живлення. Блок живлення повинен мати наступні характеристики: 5 вольт і 3 амperi (рис. 4.1).



Рисунок 1 Блок живлення

Також більшість сучасних блоків живлення від смартфонів та планшетів мають схожі характеристики (рис. 2).



Рисунок 2 Блок живлення для смартфона

Штекер блока живлення вмикається в роз'єм лампи (рис. 3).



Рисунок 3 Вмикання лампи

Якщо це перший запуск лампи після прошивки або після невдалого підключення до роутера (тільки для режиму в локальній мережі), то відбувається процес ініціалізації, тобто мікроконтролер створює Wi-Fi мережу в залежності від обраного режиму роботи (рис. 4, 5).

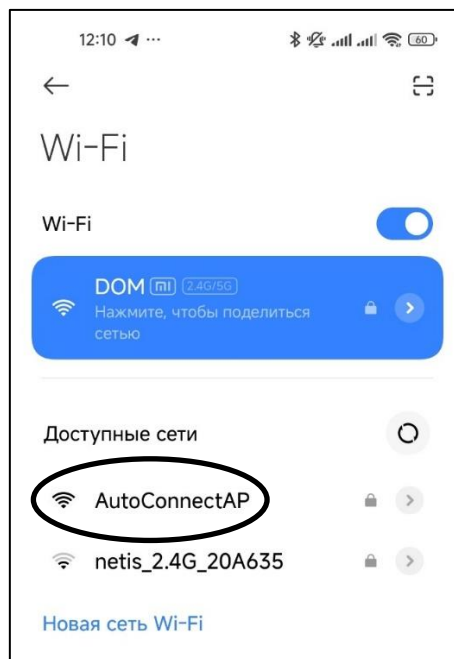


Рисунок 4 Wi-Fi мережа в локальному режимі роботи

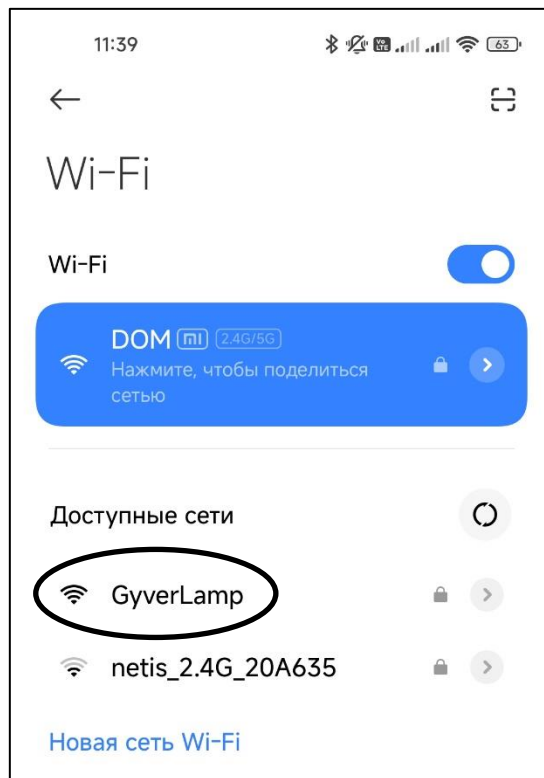


Рисунок 5 Wi-Fi мережа в режимі точки доступу

Режим роботи, назви і паролі для Wi-Fi мереж можна задати в прошивці (рис. 6). Режим роботи можна обрати вказуючи в рядку `#define ESP_MODE` значення 0 або 1, де 0 – режим роботи в точці доступу, а 1 – режим роботи в локальній мережі.

```
68 // ----- ESP -----
69 #define ESP_MODE 1
70 // 0 - точка доступу
71 // 1 - локальний
72 byte IP_AP[] = {192, 168, 4, 1};
73
74 // ----- AP (точка доступу) -----
75 #define AP_SSID "GyverLamp"
76 #define AP_PASS "12345678"
77 #define AP_PORT 8888
78
79 // ----- Менеджер WiFi -----
80 #define AC_SSID "AutoConnectAP"
81 #define AC_PASS "12345678"
```

Рисунок 6 Вибір режиму роботи та налаштування Wi-Fi мереж

В режимі локальної мережі при підключенні до Wi-Fi мережі з'являється наступне вікно конфігурації (рис. 7).

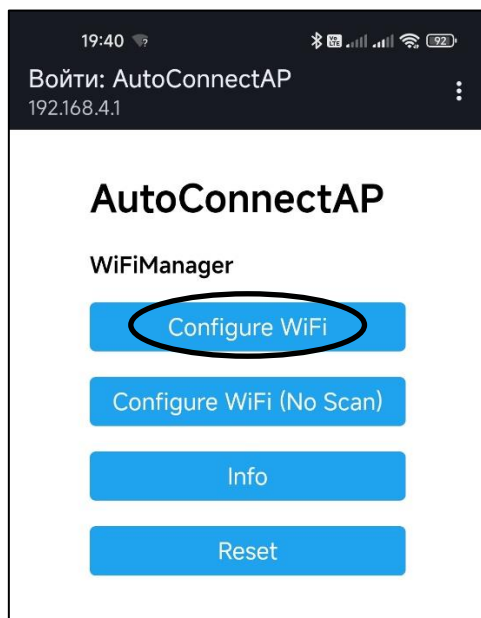


Рисунок 7 Вікно Wi-Fi менеджера

Для подальшого підключення потрібно натиснути першу кнопку «Configure WiFi» для пошуку WiFi мереж. З'являється вікно з доступними WiFi мережами поблизу. Потрібно обрати власний WiFi, ввести пароль та натиснути кнопку «Save» (рис. 8).

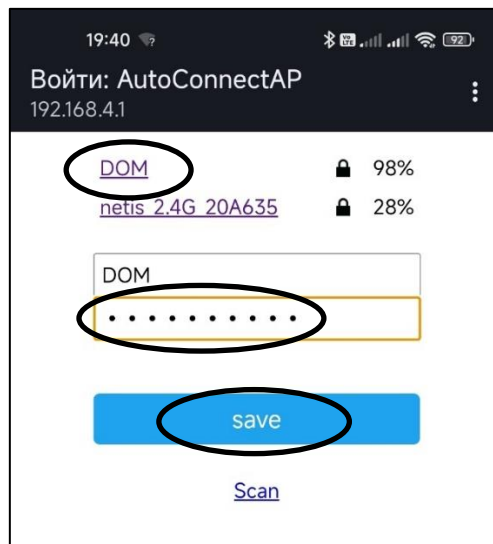


Рисунок 9 Вибір WiFi мережі з списку доступних

Далі в автоматичному режимі лампа підключається до WiFi мережі (рис 10) і запускається.

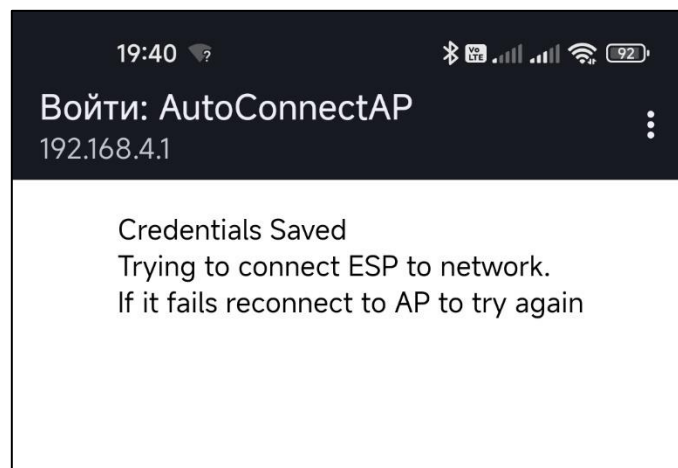


Рисунок 10 Ініціалізація підключення

Після підключення лампи до WiFi мережі, необхідно підключитись до лампи зі смартфона. Для цього потрібно завантажити та встановити додаток GyverLamp (рис. 11).

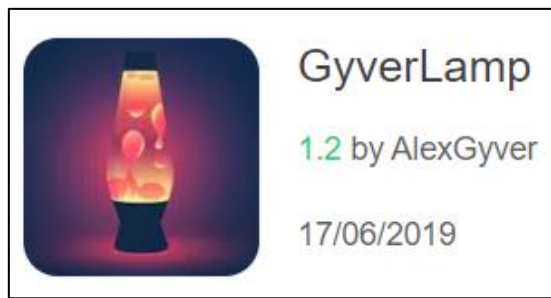


Рисунок 11 Логотип та інформація про додаток

В магазині додатків від Google (Google Play) є додаток тільки для другої версії лампи. Додаток для першої версії Google видалив, через занадто стару версію SDK.

Тому файл додатку можна знайти та завантажити з вільних джерел, наприклад з сайту <https://apkpure.com/ru/gyverlamp/com.beragumbo.GyverLamp>. Потрібно перейти на сайт, прогорнути нижче сторінку та натиснути кнопку «Завантажити APK» (рис. 12).

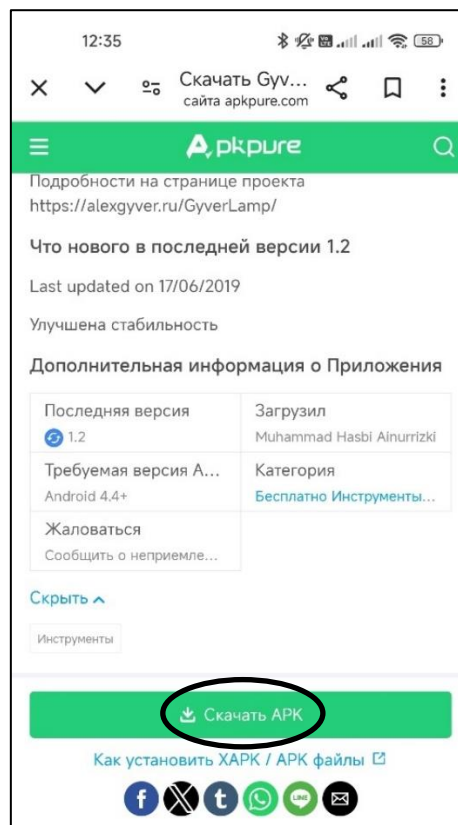


Рисунок 12 Кнопка завантаження

Далі додаток встановлюється як звичайний арк-файл (рис.13).

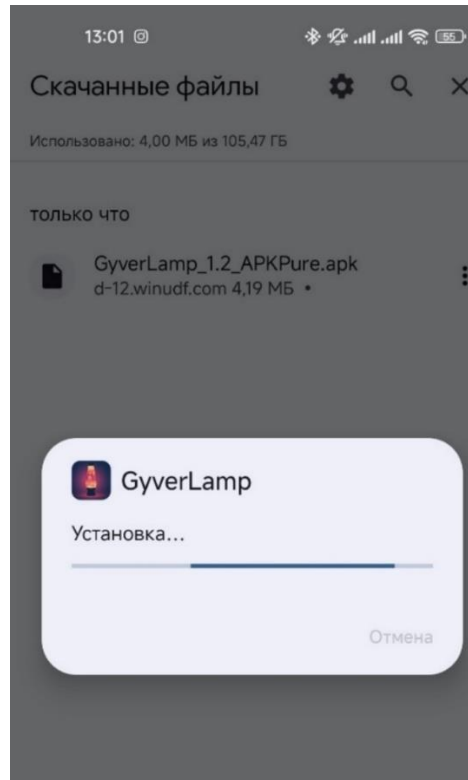


Рисунок 13 Встановлення GyverLamp

Після запуску, додаток одразу зустрічає вкладкою з налаштуванням підключення до лампи (рис. 14). Якщо цього не відбулось, потрібно натиснути на значок WiFi в правому верхньому куті.

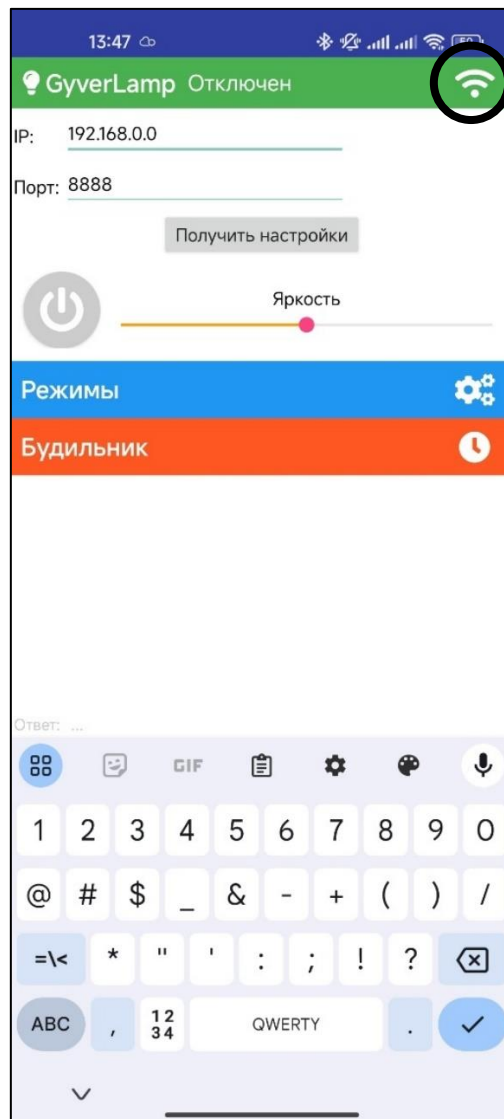


Рисунок 14 Вкладка підключення

Для підключення до лампи потрібно дізнатися IP адресу яку їй надав маршрутизатор. Це можна зробити двома способами:

- Перший спосіб: через монітор порту;
- Другий спосіб: безпосередньо на лампі.

Для того щоб дізнатись IP-адресу за першим способом потрібно: після підключення лампи до мережі WiFi підключити плату до комп'ютера, відкрити Arduino IDE та натиснути кнопку «Монітор порту» в правому верхньому куті програми (рис. 15).

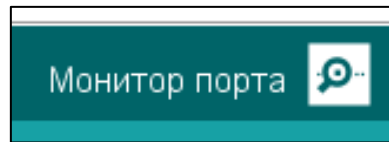


Рисунок 15 Монітор порту

Далі треба натиснути на платі кнопку RST (Reset) зображену на рисунку 16.

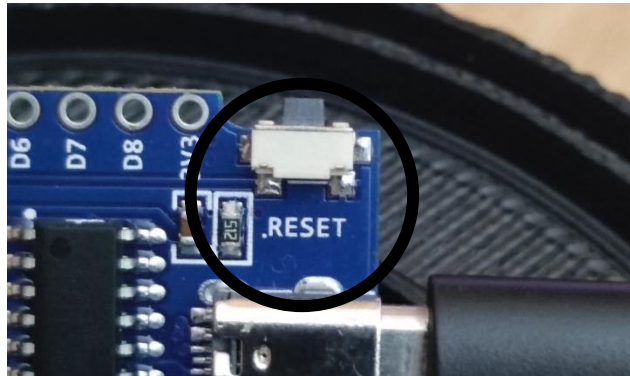


Рисунок 16 Кнопка Reset

Після чого в моніторі порту з'явиться налагоджувана інформація в якій буде вказано IP-адресу лампи в локальній мережі та порт (рис. 17).

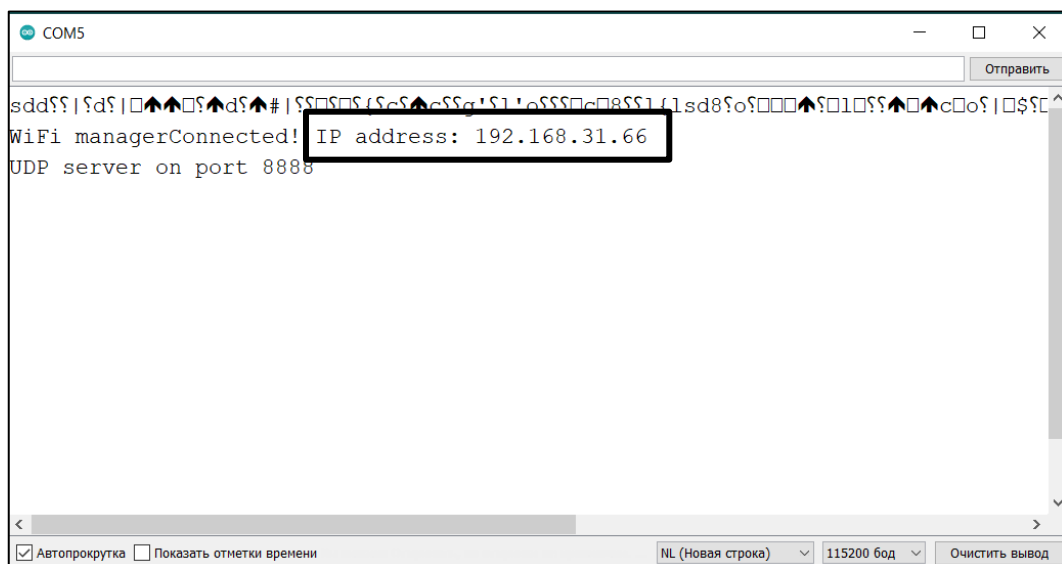


Рисунок 17 Налагоджувана інформація

Для того щоб дізнатись IP-адресу за другим способом потрібно: на вже підключеній лампі натиснути 5 разів по сенсорній кнопці, IP-адреса буде виведена

на матрицю в вигляді рухомого рядку. Приклад виведення адреси в макеті зображено на рисунку 18.

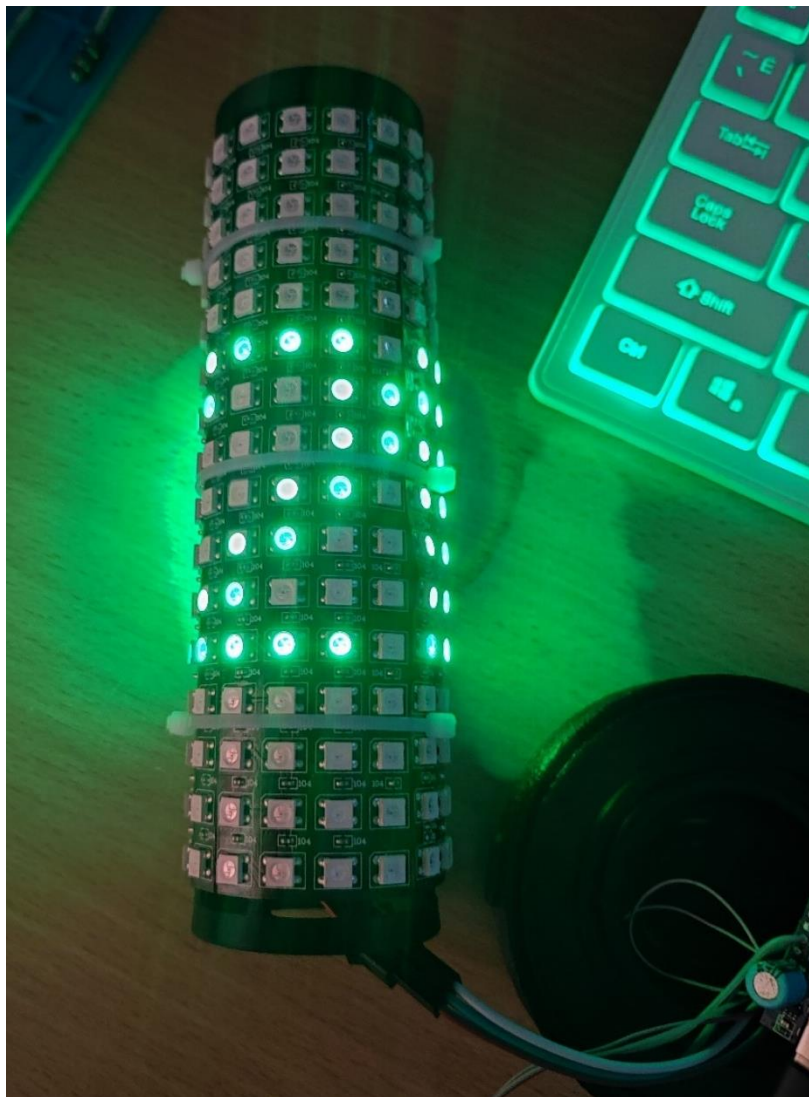


Рисунок 18 Виведення IP-адреси

Отриману IP-адресу та порт потрібно записати в додатку в відповідні поля та натиснути кнопку «Отримати налаштування». Якщо всі дії раніше були виконані правильно та коректно введено IP-адресу, то в додатку, в верхньому полі, зміниться напис з «Відключено» на «Підключено» (рис. 19). Лампа посилає відповіді на запити з додатку, тому в лівому нижньому куті можна спостерігати відповіді від лампи в вигляді напису «ОК» та часу з інтернету.

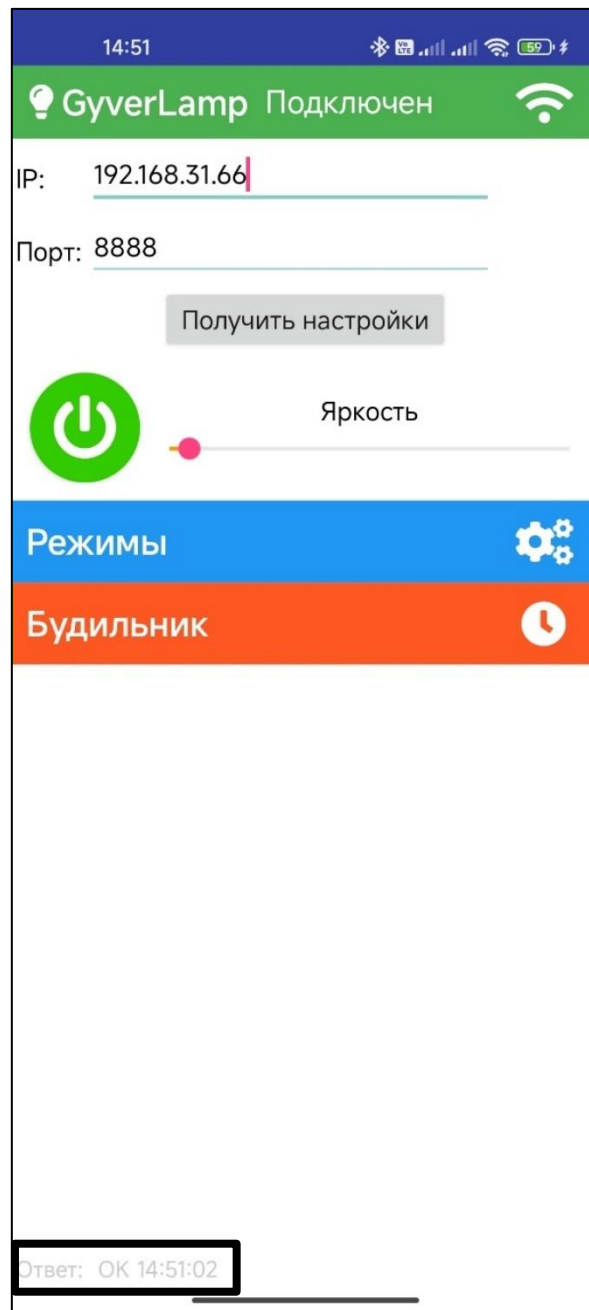


Рисунок 19 Вдале підключення до лампи з додатку

В режимі точки доступу необхідно здійснити підключення до WiFi точки доступу яку створює мікроконтролер (рис. 20).

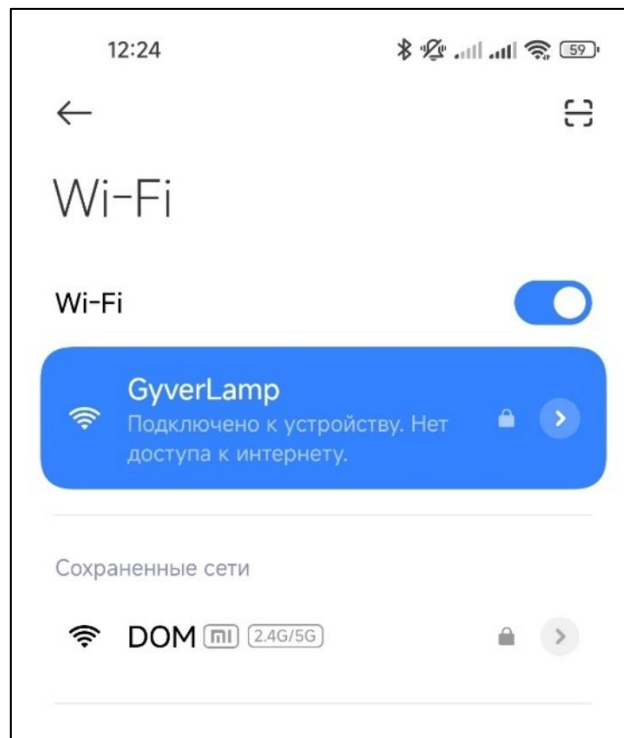


Рисунок 20 Підключення до точки доступу

Далі потрібно дізнатись IP-адресу лампи, для цього можна подивитись встановлену адресу в прошивці (рис. 21) або зайти в властивості підключеної мережі на смартфоні (рис. 22).

```

68 // ----- ESP -----
69 #define ESP_MODE 0
70 // 0 - точка доступа
71 // 1 - локальний
72 byte IP_AP[] = {192, 168, 4, 1};
73

```

Рисунок 4.21 Встановлена статична IP-адреса в прошивці

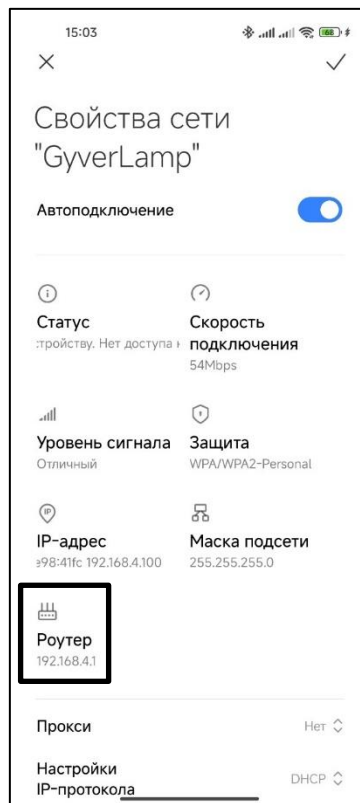


Рисунок 22 Властивості мережі

Після чого потрібно зайти в додаток та відкрити вкладку з налаштуванням підключення до лампи, ввести отриману IP-адресу та натиснути кнопку «Отримати налаштування», повинен змінитися напис з «Відключено» на «Підключено» (рис. 23).

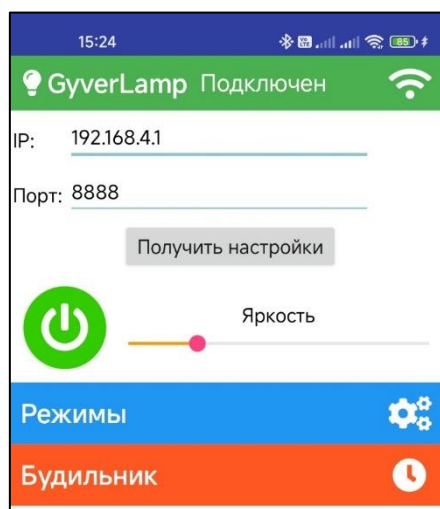


Рисунок 23 Вдале підключення до лампи з додатку

Управління лампою здійснюється локально, тобто за допомогою кнопки, так і з додатка.

Управління з кнопки:

- Одиночний клік: вимкнути/вимкнути світло, також вимкнення режиму будильника;
- Подвійний клік: зміна режиму вперед;
- Потрійний клік: зміна режиму назад;
- Клік 5 разів: виведення IP адреси лампи;
- Утримання: зміна яскравості;
- Утримання при ввімкненні (приблизно 7 секунд): скидання налаштувань WiFi.

З додатка можна налаштовувати більш поглиблено індивідуально кожен ефект, а саме: регулювання швидкості ефекту, «масштабу» ефекту (рис. 24, 25), встановлення та налаштування будильників (рис. 26, 27, 28).

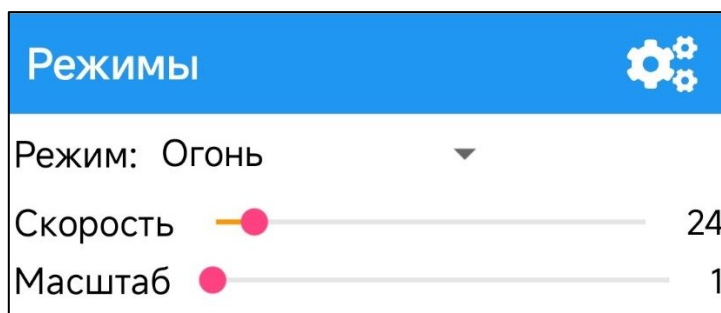


Рисунок 24 Індивідуальне налаштування режиму «Вогонь»

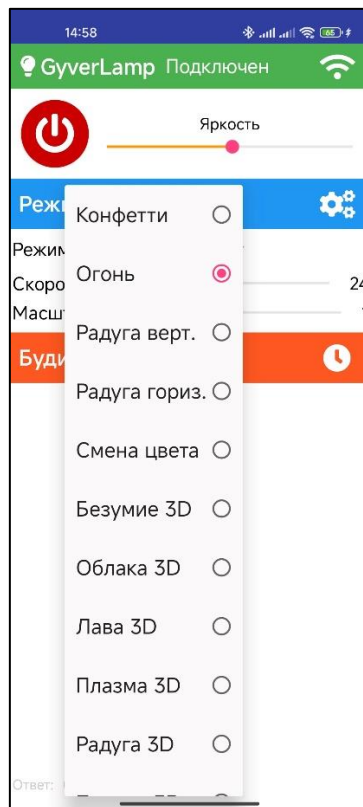


Рисунок 25 Список эффектов

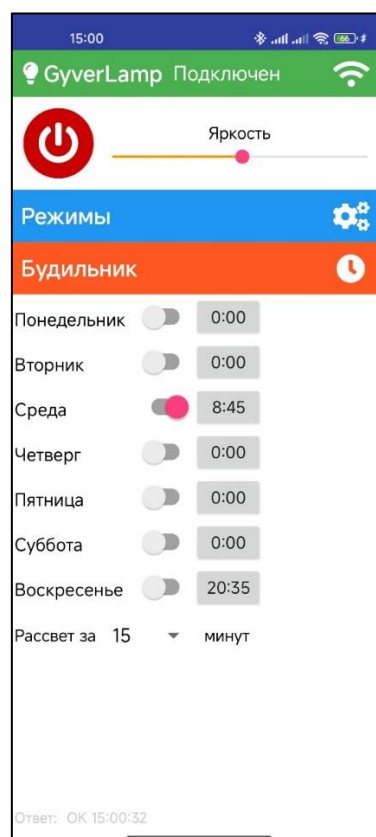


Рисунок 26 Налаштування розкладу будильників

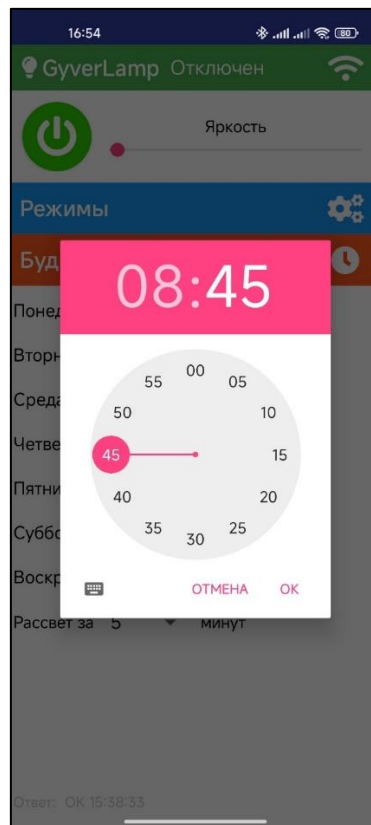


Рисунок 27 Встановлення часу будильника

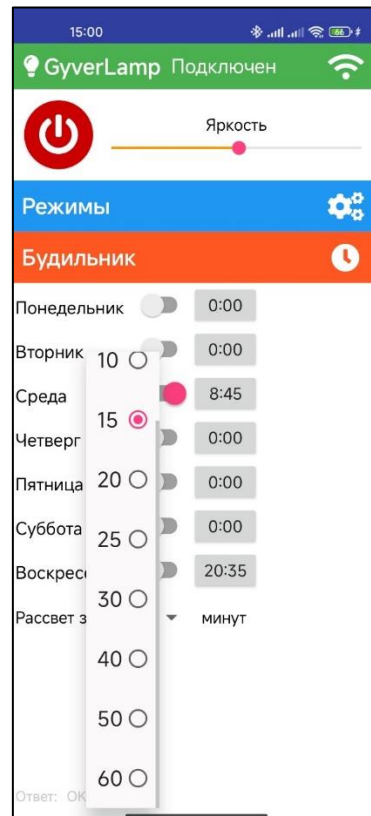


Рисунок 28 Встановлення параметру світанку

Додаток Б Опис програмного коду

Функція «sparklesRoutine()» (рис. 1) генерує випадкові координати «x» і «y» на матриці світлодіодів та встановлює випадковий колір для відповідного пікселя за допомогою функції «CHSV()». Перед встановленням кольору перевіряється, чи піксель вимкнений (getPixColorXY(x, y) == 0). Після генерації нових пікселів викликається функція «fader(70)» для поступового згасання всіх пікселів на матриці.

```
3 // ----- Конфеті -----
4 //функція створення ефекту
5 void sparklesRoutine() {
6     // цикл для генерації заданої кількості "конфеті"
7     for (byte i = 0; i < modes[0].scale; i++) {
8         byte x = random(0, WIDTH); // генерація випадкової координати x
9         byte y = random(0, HEIGHT); // генерація випадкової координати y
10        if (getPixColorXY(x, y) == 0) // перевірка, чи піксель з координатами (x, y) вимкнений
11            leds[getPixelNumber(x, y)] = CHSV(random(0, 255), 255, 255); // встановлення випадкового кольору для пікселя (x, y)
12    }
13    fader(70); // виклик функції згасання з кроком 70
14 }
```

Рисунок 1 Функція «sparklesRoutine()»

Функція «fader(byte step)» (рис. 2) проходить через всі пікселі на матриці та викликає функцію «fadePixel(i, j, step)» для кожного пікселя, передаючи його координати «i» та «j», а також крок згасання «step».

```
16 // функція плавного згасання кольору для всіх пікселів
17 void fader(byte step) {
18     // цикли для проходження по всіх пікселях матриці
19     for (byte i = 0; i < WIDTH; i++) {
20         for (byte j = 0; j < HEIGHT; j++) {
21             fadePixel(i, j, step); // виклик функції згасання для кожного пікселя
22         }
23     }
24 }
```

Рисунок 2 Функція «fader(byte step)»

Функція «fadePixel(byte i, byte j, byte step)» (рис. 3) отримує координати пікселя «i» та «j», а також крок згасання «step». Спочатку координати перетворюються у лінійний індекс у масиві «leds» за допомогою функції «getPixelNumber(i, j)». Далі перевіряється, чи піксель вимкнений (getPixColor(pixelNum) == 0). Якщо піксель активний, перевіряється, чи значення будь-якої з компонент кольору (червоної, зеленої або синьої) більше або дорівнює 30. Якщо так, то викликається функція «fadeToBlackBy(step)», яка зменшує інтенсивність кольору пікселя на заданий крок. Якщо всі компоненти кольору менші за 30, то піксель повністю вимикається (leds[pixelNum] = 0).

```
25 //функція згасання
26 void fadePixel(byte i, byte j, byte step) {
27     int pixelNum = getPixelNumber(i, j); // перетворення координат у лінійний індекс
28     if (getPixColor(pixelNum) == 0) return; // якщо піксель вимкнений, вихід з функції
29     //Якщо інтенсивність будь-якого з каналів кольору достатньо висока
30     if (leds[pixelNum].r >= 30 ||
31         leds[pixelNum].g >= 30 ||
32         leds[pixelNum].b >= 30) {
33         leds[pixelNum].fadeToBlackBy(step); // поступове згасання кольору
34     } else {
35         leds[pixelNum] = 0; // повне вимкнення пікселя, якщо інтенсивність кольору занадто низька
36     }
37 }
```

Рисунок 3 Функція «fadePixel(byte i, byte j, byte step)»