

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРОГРАМНИХ СИСТЕМ ТА ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

(підпис) Віталій РОМАНКЕВИЧ

“ ____ ” червня 2026 р.

Дипломний проєкт

**на здобуття ступеня бакалавра
за освітньо-професійною програмою**

**«Системне програмування та спеціалізовані комп'ютерні системи»
спеціальності 123 «Комп'ютерна інженерія»**

**на тему: «Офлайн-система голосового керування адресною LED-стрічкою на базі
ESP32»**

Виконала: студентка IV курсу, групи KB-23

Приймак Юліанна Павлівна

(підпис)

Керівник: асистент каф. СПіСКС, д.ф.

Молчанов Олексій Андрійович

(підпис)

Консультант з нормоконтролю: доц. каф. СПіСКС, к.т.н.,

доц. Клятченко Ярослав Михайлович

(підпис)

Рецензент: проф. каф. ОТ, д.т.н., проф.

Сергієнко Анатолій Михайлович

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студентка

(підпис)

Київ – 2026 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРОГРАМНИХ СИСТЕМ ТА ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування та спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Віталій РОМАНКЕВИЧ
(підпис)
“ ____ ” червня 2026 р.

**ЗАВДАННЯ
на дипломний проєкт студента
Приймак Юліанни Павлівни**

1. Тема проєкту «Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32», керівник проєкту асистент каф. СПіСКС, д.ф. Молчанов Олексій Андрійович, затверджені наказом університету від «28» травня 2026 р. № 1984-с
2. Термін подання студентом проєкту 8 червня 2026 р.
3. Вихідні дані до проєкту див. Технічне завдання
4. Зміст пояснювальної записки:
 - Порівняльний аналіз існуючих рішень
 - Вибір апаратних компонентів системи голосового керування
 - Розробка системи голосового керування
 - Тестування розробленої системи
5. Перелік графічного матеріалу:
 - Схема роботи програми
 - Обробка голосових команд. Схема алгоритму
 - Взаємодія модулів. Схема структурна

– Схема електрична принципова

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я. М., доцент кафедри СПіСКС, к.т.н., доцент		

7. Дата видачі завдання 21.01.2026 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Терміни виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	15.01.2026	
2.	Розробка та узгодження технічного завдання	21.01.2026	
3.	Аналіз існуючих рішень	26.04.2026	
4.	Підготовка матеріалів першого розділу дипломного проєкту	03.05.2026	
5.	Підготовка матеріалів другого розділу дипломного проєкту	10.05.2026	
6.	Підготовка матеріалів третього розділу дипломного проєкту	17.05.2026	
7.	Підготовка матеріалів четвертого розділу дипломного проєкту	23.05.2026	
8.	Підготовка графічної частини дипломного проєкту	27.05.2026	
9.	Оформлення документації дипломного проєкту	31.05.2026	
10.	Попередній огляд матеріалів диплому на кафедрі	05.06.2026	

Студентка

(підпис)

Юліанна ПРИЙМАК

Керівник проєкту

(підпис)

Олексій МОЛЧАНОВ

АНОТАЦІЯ

Бакалаврський дипломний проєкт включає пояснювальну записку (55 стор., 25 рис., 11 табл., список використаної літератури з 18 найменувань, 6 додатків).

Мета проєкту — розробка офлайн-системи голосового керування адресною LED стрічкою на базі мікроконтролера ESP32 що забезпечує повністю автономну роботу без підключення до інтернету чи зовнішніх сервісів.

Розроблена система дає змогу керувати адресною LED стрічкою WS2812B голосовими командами в офлайн режимі — змінювати колір, яскравість, колірну температуру та активувати динамічні світлові ефекти без необхідності підключення до мережі інтернет чи використання смартфона.

В ході виконання дипломного проєкту:

- проведено аналіз існуючих рішень для керування LED освітленням з голосовим керуванням;
- обґрунтовано вибір апаратних компонентів системи;
- розроблено офлайн-систему голосового керування адресною LED стрічкою;
- реалізовано 30 голосових команд та 6 динамічних світлових ефектів;
- проведено тестування та аналіз роботи системи за критеріями точності розпізнавання, дистанції, часу відгуку та надійності при тривалій роботі.

В процесі розробки були використані: мікроконтролер ESP32 на базі ядра Xtensa LX6, модуль розпізнавання голосу DFRobot Gravity SEN0539 з нейромережевим чипом DF2301Q, адресна LED стрічка WS2812B, середовище розробки Arduino IDE, бібліотеки Adafruit NeoPixel та DFRobot_DF2301Q.

Область подальшого використання — системи розумного дому, декоративне та функціональне освітлення приміщень.

Ключові слова: голосове керування, офлайн розпізнавання голосу, ESP32, WS2812B, адресна LED стрічка, нейромережевий класифікатор, I2C.

ABSTRACT

Bachelor's diploma project includes an explanatory note (55 pages, 25 figures, 11 tables, list of references with 18 sources, 6 appendices).

The purpose of the project is to create an offline-system for voice control of an addressable LED strip based on the ESP32 microcontroller that ensures fully autonomous operation without connecting to the internet or external services.

The developed system allows controlling the WS2812B addressable LED strip using voice commands in offline mode — changing color, brightness, color temperature and activating dynamic lighting effects without the need for internet connection or smartphone usage.

In the course of the diploma project:

- an analysis of existing solutions for LED lighting control with voice control was conducted;
- the selection of hardware components of the system was justified;
- offline-system for voice control of an addressable LED strip was developed;
- 30 voice commands and 6 dynamic lighting effects were implemented;
- testing and analysis of the system performance was conducted according to the criteria of recognition accuracy, distance, response time and stability.

In the development process the following were used: ESP32 microcontroller based on the Xtensa LX6 core, DFRobot Gravity SEN0539 voice recognition module with the DF2301Q neural network chip, WS2812B addressable LED strip, Arduino IDE development environment, Adafruit NeoPixel and DFRobot_DF2301Q libraries.

Scope of further application — smart home systems, decorative and functional indoor lighting.

Keywords: voice control, offline speech recognition, ESP32, WS2812B, addressable LED strip, neural network classifier, I2C.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.468300.002 ТЗ	Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32 Технічне завдання	4		
	A4	ІАЛЦ.468300.003 ТП	Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32 Відомість технічного проекту	2		
	A4	ІАЛЦ.468300.004 ПЗ	Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32 Пояснювальна записка	55		
	A4	ІАЛЦ.468300.005 Д1	Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32 Схема роботи системи	1		
ІАЛЦ.468300.001 ОА						
Змін.	Арк.	№ докум.	Підпис	Дата		
Розробив	Приймак Ю.П.				Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32 Літ. Аркуш 1 Аркушів 2 КПП ім. Ігоря Сікорського, ФПСМ КВ-23	
Перевірив	Молчанов О.А.					
Консульт.						
Н. контроль	Клятченко Я.М.					
Зав. каф.	Романкевич В. О.					
Опис альбому						

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.468300.006 Д2	Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32	1		
			Обробка голосових команд.			
			Схема алгоритму			
	A4	ІАЛЦ.468300.007 Д3	Схема структурна керування адресною LED-стрічкою на базі ESP32	1		
			Взаємодія модулів.			
			Схема структурна			
	A4	ІАЛЦ.468300.008 Е3	Схема структурна керування адресною LED-стрічкою на базі ESP32			
			Схема електрична принципова	2		
		Диск CD-ROM	Текст пояснювальної записки			
			Графічний матеріал			

Змін	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.001 ОА

Арк.
2

ЗМІСТ

1.	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ	2
2.	ПІДСТАВА ДЛЯ РОЗРОБКИ	2
3.	МЕТА І ПРИЗНАЧЕННЯ РОБОТИ.....	2
4.	ДЖЕРЕЛА РОЗРОБКИ.....	2
5.	ТЕХНІЧНІ ВИМОГИ.....	2
5.1	Вимоги до програмного продукту, що розробляється	2
5.2	Вимоги до апаратного забезпечення.....	3
5.3	Вимоги до програмного та апаратного забезпечення користувача.....	3
6.	ЕТАПИ РОЗРОБКИ	4

					ІАЛЦ.468300.002 ТЗ			
Змін	Арк.	№ докум.	Підпис	Дата				
Розробив		Приймак Ю.П.			<i>Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32</i> Технічне завдання		Літ.	Аркуш
Перевірив		Молчанов О.А.						1
Н. контроль		Клятченко Я.М.					КПІ ім. Ігоря Сікорського, ФПСМ КВ-23	
Затвердив		Романкевич В.О.						

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32».

Галузь застосування: системи розумного освітлення, декоративне підсвічування, автономні вбудовані системи керування освітленням.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проектування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є розробка програмно-апаратної системи керування адресною LED стрічкою WS2812B з використанням офлайн розпізнавання голосових команд на базі мікроконтролера ESP32 та модуля DFRobot Gravity SEN0539, що забезпечує повністю автономну роботу без підключення до інтернету чи зовнішніх сервісів.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється

- реалізація офлайн розпізнавання голосових команд без підключення до інтернету
- підтримка не менше 15 голосових команд для керування освітленням
- керування кольором стрічки (червоний, помаранчевий, жовтий, зелений, блакитний, синій, фіолетовий, білий)

					ІАЛЦ.468300.002 ТЗ	Арк.
Зм	Лист	№ докум.	Підп.	Дата		2

- керування яскравістю (збільшення, зменшення, максимум, мінімум)
- керування колірною температурою білого освітлення
- реалізація не менше 6 динамічних світлових ефектів (пульсація, біжучий вогонь, мерехтіння, вогонь, тривога, відпочинок)
- реалізація спеціальних режимів освітлення (денний, місячний, вечірковий, холодний, теплий)
- вмикання та вимикання системи голосовою командою
- візуальна індикація стану ініціалізації системи
- неблокуюча реалізація анімацій для забезпечення безперервного отримання команд

5.2 Вимоги до апаратного забезпечення

- мікроконтролер ESP32 з тактовою частотою не менше 240 МГц та обсягом RAM не менше 520 КБ
- модуль офлайн розпізнавання голосу з підтримкою speaker independent режиму та інтерфейсом I2C
- адресна LED стрічка WS2812B з напругою живлення 5V
- резистор номіналом 1 кОм на лінії даних
- електролітичний конденсатор ємністю 1000 мкФ для захисту стрічки

5.3 Вимоги до програмного та апаратного забезпечення користувача

- розпізнавання голосу будь-якого користувача
- відсутність необхідності у смартфоні, інтернеті чи додаткових пристроях для керування системою
- дистанція подачі голосової команди — до 3 метрів

					ІАЛЦ.468300.002 ТЗ	Арк.
						3
Зм	Лист	№ докум.	Підп.	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Терміни виконання етапів проєкту
1.	Вивчення літератури за тематикою проєкту	15.01.2026
2.	Аналіз існуючих рішень	21.01.2026
3.	Розробка технічного завдання	26.04.2026
4.	Підготовка матеріалів першого розділу дипломного проєкту	03.05.2026
5.	Підготовка матеріалів другого розділу дипломного проєкту	10.05.2026
6.	Підготовка матеріалів третього розділу дипломного проєкту	17.05.2026
7.	Підготовка матеріалів четвертого розділу дипломного проєкту	23.05.2026
8.	Підготовка графічної частини дипломного проєкту	27.05.2026
9.	Оформлення документації дипломного проєкту	31.05.2026
10.	Попередній огляд матеріалів диплому на кафедрі	05.06.2026

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.468300.004 ПЗ	Офлайн-система голосового Керування адресною LED-стрічкою на базі ESP32	55		
			Пояснювальна записка			
	A4	ІАЛЦ.468300.005 Д1	Офлайн-система голосового Керування адресною LED-стрічкою на базі ESP32	1		
			Схема роботи програми			
	A4	ІАЛЦ.468300.006 Д2	Офлайн-система голосового Керування адресною LED-стрічкою на базі ESP32	1		
			Обробка голосових команд			
			Схема алгоритму			
	A4	ІАЛЦ.468300.007 Д3	Офлайн-система голосового Керування адресною LED-стрічкою на базі ESP32	1		
			Взаємодія модулів			
			Схема структурна			

ІАЛЦ.468300.003 ТП					Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32 Відомість технічного проєкту		
Змін	Арк.	№ докум.	Підпис	Дата			
Розробив		Приймак Ю.П.			Літ.		Аркуш
Перевірив		Молчанов О.А.					Аркушів
						1	2
Н. контроль		Клятченко Я.М.			КПІ ім. Ігоря Сікорського, ФПСІМ КВ-23		
Затвердив		Романкевич В.О.					

[illegible]

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему: «Офлайн-система голосового керування адресною LED стрічкою
на базі ESP32»

Київ – 2026

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	3
ВСТУП.....	4
РОЗДІЛ 1 ПОРІВНЯЛЬНИЙ АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	5
1.1 Огляд наявних рішень	5
1.1.1 Контролер GL-C-310WL	6
1.1.2 Контролер GL-C-211WL	7
1.1.3 Контролер SUPERNIGHT Bluetooth Controller	8
1.1.4 Контролер VOICEEMITY Dynamic White SD Controller	9
1.1.5 Контролер VOICEEMITY Dynamic White SI controller	10
1.2 Порівняння наявних рішень.....	11
1.3 Обґрунтування вибору теми дипломного проєкту	12
1.4 Висновки до першого розділу.....	13
РОЗДІЛ 2 ВИБІР АПАРАТНИХ КОМПОНЕНТІВ СИСТЕМИ ГОЛОСОВОГО КЕРУВАННЯ	15
2.1 Мікроконтролер.....	15
2.1.1 Мікроконтролер Arduino UNO	15
2.1.2 Мікроконтролер STM32	16
2.1.3 Мікроконтролер ESP32	17
2.2 Модуль розпізнавання голосу.....	18
2.2.1 Модуль розпізнавання голосу Elechouse V3	19
2.2.2 Модуль розпізнавання голосу LD3320	20
2.2.3 Модуль розпізнавання голосу DFRobot Gravity SEN0539	21
2.2.4 Обґрунтування вибору модулю розпізнавання голосу.....	23
2.3 Адресна LED стрічка.....	23
2.3.1 Адресна LED стрічка APA102 (SK9822).....	24
2.3.2 Адресна LED стрічка SK6812	24
2.3.3 Адресна LED стрічка WS2812B.....	25

					ІАЛЦ.468300.004 ПЗ			
Змін	Арк.	№ докум.	Підпис	Дата				
Розробив	Приймак Ю.П.				Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32 Пояснювальна записка	Літ.	Аркуш	Аркушів
Перевірив	Молчанов О.А.						1	55
						КПІ ім. Ігоря Сікорського, ФПСІМ КВ-23		
Н. контроль	Клятченко Я.М.							
Затвердив	Романкевич В.О.							

2.3.4 Обґрунтування вибору адресної LED стрічки	26
2.4 Поєднання обраних засобів.....	26
2.5 Висновки до другого розділу	27
РОЗДІЛ 3 РОЗРОБКА СИСТЕМИ ГОЛОСОВОГО КЕРУВАННЯ	28
3.1 Загальна архітектура системи	28
3.2 Схема підключення	29
3.3 Програмна реалізація	31
3.3.1 Ініціалізація системи	31
3.3.2 Отримання голосової команди.....	33
3.3.3 Обробка команди.....	34
3.3.4 Формування керуючого сигналу	36
3.3.5 Реалізація світлових ефектів	36
3.4 Реалізовані функції системи	37
3.5 Порівняння з аналогами	39
3.6 Висновки до третього розділу	41
РОЗДІЛ 4 ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ.....	42
4.1 Точність розпізнавання голосових команд	42
4.2 Дистанція розпізнавання.....	44
4.3 Час відгуку системи	45
4.5 Стійкість до фонового шуму.....	46
4.6 Стабільність роботи	47
4.7 Порівняння розробленої системи з аналогами	48
4.8 Висновки до четвертого розділу	50
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
ДОДАТОК А. Схема роботи системи	56
ДОДАТОК Б. Обробка голосових команд. Схема алгоритму	58
ДОДАТОК В. Взаємодія модулів. Схема структурна.....	60
ДОДАТОК Г. Схема електрична принципова	62
ДОДАТОК Д. Презентація	65
ДОДАТОК Е. Лістинг коду	75

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

2

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

LED (Light Emitting Diode) — світловипромінювальний діод, напівпровідниковий пристрій, що перетворює електричний струм на світло.

ПЗ — програмне забезпечення.

DTW (Dynamic Time Warping) — алгоритм порівняння часових послідовностей, що вимірює схожість між двома сигналами що можуть відрізнятися за швидкістю або тривалістю.

HMM (Hidden Markov Model) — статична модель що описує систему з прихованими станами, яка використовується для розпізнавання мовлення шляхом моделювання імовірнісних закономірностей звукових послідовностей..

BLE (Bluetooth Low Energy) — енергоефективна версія технології Bluetooth.

RGB (Red Green Blue) — колірна модель що описує колір за допомогою комбінації трьох кольорів: червоного, зеленого, синього.

RGBW (Red Green Blue White) — розширена модель RGB з додатковим білим каналом.

CCT (Correlated Color Temperature) — корельована колірна температура, параметр що описує відтінок білого світла в кельвінах від теплого до холодного.

DW (Dynamic White) — динамічний білий, тип освітлення з можливістю регулювання колірної температури білого світла.

SD (Speaker Dependent) — залежний від голосу користувача.

SI (Speaker Independent) — незалежний від голосу користувача.

I2C (Inter-Integrated Circuit) — послідовний інтерфейс передачі даних що використовує дві лінії зв'язку.

SDA (Serial Data Line) — лінія зв'язку для передачі даних.

SCL (Serial Clock Line) — лінія зв'язку для тактового сигналу.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

3

ВСТУП

Розвиток технологій розпізнавання мовлення та мікроелектроніки відкрив нові можливості для створення інтуїтивних систем керування побутовими пристроями. Голосове керування поступово стає стандартом взаємодії людини з технікою — воно дає змогу керувати пристроями природним способом без необхідності фізичного контакту з органами керування. Паралельно адресні LED стрічки набувають широкого застосування в декоративному та функціональному освітленні завдяки можливості індивідуального керування кожним світлодіодом та створення динамічних світлових ефектів.

Поєднання цих двох технологій — голосового керування та адресного LED освітлення — є перспективним напрямком розробки систем розумного дому. Проте більшість існуючих рішень реалізують голосове керування через хмарні сервіси що робить їх повністю залежними від стабільного інтернет з'єднання. Відключення мережі або недоступність серверів третіх сторін миттєво позбавляє систему основної функціональності. Комерційні рішення з вбудованим офлайн розпізнаванням у свою чергу є вузькоспеціалізованими — підтримують лише білі стрічки та обмежену кількість команд що суттєво знижує їх практичну цінність.

Враховуючи специфіку поставленої задачі та недоліки наявних рішень, актуальним є розроблення системи що поєднує переваги офлайн розпізнавання голосу з повноцінним керуванням адресним RGB освітленням. Реалізація такої системи на базі мікроконтролера ESP32 та спеціалізованого модуля розпізнавання голосу DFRobot Gravity SEN0539 з нейромережевим процесором дає змогу досягти повної автономності роботи, speaker independent розпізнавання голосу будь-якого користувача та широких можливостей керування освітленням без залежності від інтернету чи зовнішніх сервісів.

РОЗДІЛ 1

ПОРІВНЯЛЬНИЙ АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Огляд наявних рішень

Системи керування LED стрічками з голосовим вводом мають різні програмні та апаратні реалізації. Залежно від реалізації, такі системи можна класифікувати за такими параметрами, як: інтерфейс керування (напр. Wi-Fi, Bluetooth, Offline тощо), тип підтримуваних LED стрічок (аналогові, адресні або білі), методом обробки голосових команд (через хмарних помічників Amazon Alexa/Google Home, спеціалізованого ПЗ або безпосередньо на пристрої) та ступенем відкритості системи.

Онлайн-системи, що є дуже поширеними на ринку, передають сигнал на зовнішні сервери для обробки, що забезпечує високу точність розпізнавання голосових команд, однак є залежними від стабільного інтернет-з'єднання та хмарних помічників. Офлайн-системи виконують розпізнавання безпосередньо на вбудованому чипі, використовуючи такі алгоритми як DTW, НММ та нейромережеві класифікатори, що забезпечує повну автономність пристрою. На відміну від онлайн-систем, вони не мають вільно доступних прикладів на ринку.

Для виявлення технічних переваг та обмежень наявних рішень та обґрунтування розробки власної системи голосового управління LED стрічкою було проведено порівняльний аналіз представників різних категорій LED контролерів з голосовим керуванням.

					ІАЛЦ.468300.004 ПЗ	Арк.
						5
Змін.	Арк.	№ докум.	Підпис	Дата		

1.1.1 Контролер GL-C-310WL

GL-C-310WL [1] — це компактний контролер на базі ESP32, призначений для керування адресними LED стрічками. Його зовнішній вигляд наведено на рис. 1.1.



Рисунок 1.1 — Зовнішній вигляд контролера GL-C-310WL

Керування здійснюється за допомогою безкоштовного додатку WLED, що є популярним відкритим програмним забезпеченням, що надає широкий набір інструментів для керування стрічкою, та має можливість інтеграції з розумним домом через голосовий помічник Alexa від Amazon [2]. Пристрій працює лише при підключенні через Wi-Fi та не має вбудованого керування голосом, а вбудований мікрофон використовується тільки для синхронізації зміни світла в ритм музики.

Для підключення та використання контролера потрібні: адресна стрічка на кшталт WS2812B, блок живлення 5-24V залежно від стрічки що використовується, смартфон з Wi-Fi та вище згаданий додаток WLED. При першому підключенні пристрій створює власну точку доступу для початкового налаштування.

Пристрій має доволі широкий набір функцій, що охоплює понад 100 динамічних режимів, можливість налаштування власних ефектів, сегментне керування та збереження до 250 користувацьких пресетів.

1.1.2 Контролер GL-C-211WL

GL-C-211WL [2] — це контролер на базі мікроконтролера ESP32 що використовує метод широтно-імпульсної модуляції для керування аналоговими LED стрічками. Зовнішній вигляд контролера представлено на рис. 1.2.



Рисунок 1.2 — Зовнішній вигляд контролера GL-C-211WL

Даний контролер не підтримує адресні стрічки, що не надає можливість індивідуального керування кожним світлодіодом і керує стрічкою як єдиним цілим, змінюючи інтенсивність кожного кольорового каналу через швидке перемикавання напруги — технологія відома як широтно-імпульсна модуляція.

Пристрій є універсальним рішенням для аналогового освітлення, так як підтримує RGB, RGBW, CCT та монохромні стрічки. Як і попередній контролер, GL-C-211WL використовує програмне забезпечення WLED, та підтримує інтеграцію з розумними голосовими помічниками через хмарні сервіси. Для використання пристрою потрібні блок живлення 12-24V, залежно від використаної стрічки, смартфон підключений до Wi-Fi та додаток WLED.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

7

1.1.3 Контролер SUPERNIGHT Bluetooth Controller

SUPERNIGHT Bluetooth Controller [4] — як і попередній, цей контролер підтримує лише аналогові LED стрічки, без підтримки адресних. На рис. 1.3 наведено зовнішній вигляд цього контролера.

На відміну від обох попередніх контролерів, SUPERNIGHT використовує керування через Bluetooth та не потребує підключення до Wi-Fi для користування. Керування стрічкою реалізується за допомогою безкоштовного додатку без необхідності підключення до Інтернету.



Рисунок 1.3 — Зовнішній вигляд контролера SUPERNIGHT Bluetooth Controller

Пристрій є закритою системою без можливості власного налаштування, а виробник не надає публічної інформації про використаний мікроконтролер та не має вільнодоступної технічної документації. Для використання потрібні блок живлення 12-24V, аналогова стрічка (накшталт SMD 2835, 3528, 5050), смартфон з Bluetooth 4.0 та завантаженим вище згаданим додатком.

Перевагою SUPERNIGHT Bluetooth Controller є повна незалежність від Інтернету завдяки Bluetooth, але це обмежує його радіус дії до 9 метрів. Додатково технологія BLE забезпечує низьке енергоспоживання порівняно з Wi-Fi рішеннями, що є великою перевагою при використанні живлення від батареї.

1.1.4 Контролер VOICEEMITY Dynamic White SD Controller

Voiceemity DW SD [5] — це вузькоспеціалізований контролер, що працює виключно з білими стрічками, такими як CCT та Single White. Його зовнішній вигляд можна побачити на рис. 1.4.

На жаль, технічна документація цього виробника дуже обмежена, тому невідомо, який мікроконтролер використовується всередині, а сам пристрій не є представленим у відкритому продажі.



Рисунок 1.4 — Зовнішній вигляд контролера VOICEEMITY Dynamic White Speaker Dependent Controller

На відміну від попередніх контролерів, цей має вбудоване розпізнавання голосу. На основі технічних характеристик, можна припустити використання DTW, що порівнює шаблони записаного голосу. Таке рішення є одним з найпростіших та найменш ресурсомістких та усуває мовний бар'єр, так як пристрій розпізнає лише мову конкретної людини незалежно від мови та вимови, тому технічно командою може бути будь-який звук. Система підтримує до 8 команд та потребує окремого тренування для кожної перед використанням.

Найбільшими перевагами такого рішення є низьке енергоспоживання та незалежність від зовнішніх пристроїв, проте дуже обмежена кількість команд

та дистанція розпізнавання до 1 метра, робота виключно з білими стрічками та необхідність тренування є суттєвими недоліками.

1.1.5 Контролер VOICEEMITY Dynamic White SI controller

Як і попередній контролер, Voiceemity DW SI [6] є вузькоспеціалізованим та працює з білими стрічками. Зовнішній вигляд контролера показано на рис. 1.5.

Документація на стільки ж обмежена, а пристрій не є представленим у відкритому продажу. У порівнянні з Voiceemity DW SD, дистанція розпізнавання голосу суттєво більша — до 5 метрів.



Рисунок 1.5 — Зовнішній вигляд контролера VOICEEMITY Dynamic White Speaker Independent controller

Контролер має вбудоване офлайн розпізнавання голосу, реалізоване безпосередньо на чипі без використання Wi-Fi чи Bluetooth. Ця система не потребує попереднього тренування та використовує набір вбудованих голосових команд, розпізнаючи голос будь-якого користувача, це потребує використання більш складного алгоритму розпізнавання, що також пояснює збільшення дистанції розпізнавання до 5 метрів.

Ці технічні вдосконалення роблять пристрій більш практичним за Speaker Dependent версію, але попри кращий підхід до розпізнавання голосу, контролер залишається вузькоспеціалізованим через ті ж обмежену кількість

команд, підтримку виключно білих стрічок, закритої архітектури та недоступність на ринку.

1.2 Порівняння наявних рішень

В таблиці 1.1 можна побачити, що контролери на базі Wi-Fi, такі як GL-C-310WL та GL-C-211WL мають найбільший набір функцій, однак їм бракує незалежності від зовнішніх пристроїв, а голосове керування реалізоване лише через хмарні сервіси.

Таблиця 1.1 — Порівняльна таблиця контролерів

Характеристика	GL-C-310WL	GL-C-211WL	SUPERNIGHT	Voiceemity DW SD	Voiceemity DW SI
Тип керування	Додаток Wi-Fi	Додаток Wi-Fi	Додаток Bluetooth	Голос	Голос
Інтерфейс зв'язку	Wi-Fi	Wi-Fi	Bluetooth	Офлайн	Офлайн
Тип стрічки	Адресна	Аналогова	Аналогова	Біла	Біла
Голосове керування	Через розумних помічників	Через розумних помічників	Через ПЗ	Вбудоване	Вбудоване
Speaker Independent	Так	Так	Так	Ні	Так
Дистанція розпізнавання	Залежно від зовн. пристрою	Залежно від зовн. пристрою	Залежно від зовн. пристрою	До 1м	До 5м
ПЗ	WLED	WLED	HappyLighting	-	-
Живлення	5-24В	12-24В	12-24В	11-27В	11-27В
Ціна	\$15-20	\$15-20	\$15-20	-	-

SUPERNIGHT є гарним прикладом незалежності від Wi-Fi завдяки Bluetooth, але все ще має проблему залежності від смартфона та відсутності

підтримки адресних LED стрічок що значно звужує потенційний функціональність пристрою.

Контролери Voiceemity є єдиними автономними серед розглянутих та що мають вбудоване голосове керування. Проте, обидва дуже обмежені через підтримку виключно білих стрічок та мінімальною кількістю підтримуваних команд.

1.3 Обґрунтування вибору теми дипломного проєкту

На сучасному ринку систем керування LED стрічками можна знайти широкий вибір готових рішень, однак під час їх аналізу можна виявити ряд обмежень, що обумовлюють актуальність розробки власної системи. Першим обмеженням є повна залежність від Інтернету та хмарних сервісів. Більшість контролерів, що мають опцію голосового керування, реалізують його через голосові помічники, такі як Alexa від Amazon та Google Home. Це веде до повної залежності від стабільного підключення до Інтернету, що робить повністю недоступною функцію голосового керування при його відсутності або недоступності серверів хмарних помічників.

Також, проведений аналіз продемонстрував залежність контролерів від наявності смартфона для будь-якого керування, що робить їх функціональність повністю залежною від додаткового пристрою та заряду його батареї.

Оглянуті автономні системи виявились дуже вузькоспеціалізованими, бо підтримують тільки білі стрічки, мають до 8 команд та є закритими системами без можливості модифікації. Також, обидва Voiceemity DW SI та Voiceemity DW SD не є вільно доступними для купівлі та не мають оприлюдненої технічної документації.

Саме усунення вищевказаних недоліків є основною метою дипломного проєкту — розробка системи керування адресною LED стрічкою, з офлайн

голосовим керуванням та відкритою архітектурою з можливістю розширення, що поєднує переваги розглянутих рішень та позбавляється ключових обмежень.

1.4 Висновки до першого розділу

Під час проведення порівняльного аналізу розглянуто LED контролери з різними підходами до керування освітленням — Wi-Fi/Bluetooth залежні, з інтеграцією розумних помічників, таких як Amazon Alexa та Google Home, а також незалежні офлайн-системи з вбудованим голосовим керуванням. За результатами можна виділити наступне:

1. Виконано огляд існуючих рішень для керування LED стрічками — Wi-Fi контролерів Gledopto, Bluetooth контролера SUPERNIGHT та офлайн голосових контролерів Voiceemity. Визначено їх основні технічні характеристики, функціональні можливості, переваги та недоліки.
2. Проведено порівняльний аналіз розглянутих рішень за типом керування, ступенем автономності, наявністю та видом голосового керування, інтерфейсом зв'язку, різноманітністю функціональності та підтримуваними типами стрічок. Встановлено, що попри різноманітну функціональність Wi-Fi-контролерів, їм бракує незалежності від зовнішніх пристроїв та стабільного Інтернет підключення. Bluetooth-контролер SUPERNIGHT усуває залежність від Інтернету завдяки технології BLE, але є залежним від смартфона та має закриту систему. Контролери Voiceemity є по справжньому автономними, однак мають великі недоліки, бо є вузькоспеціалізованими з обмеженим функціоналом.
3. Обґрунтовано актуальність теми дипломного проекту, оскільки вона спрямована на поєднання автономності, що можна побачити у

офлайн-пристроїв, таких як Voicemity, з більш різноманітною функціональністю Wi-Fi- та Bluetooth-залежних контролерів, можливістю керування саме кольоровою LED стрічкою та відкритого програмного забезпечення.

					ІАЛЦ.468300.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		14

РОЗДІЛ 2

ВИБІР АПАРАТНИХ КОМПОНЕНТІВ СИСТЕМИ ГОЛОСОВОГО КЕРУВАННЯ

Розроблювана система керування адресною LED стрічкою з голосовим керуванням складається з трьох основних компонентів: мікроконтролер, модуль розпізнавання голосу та адресна LED стрічка. Для кожного з перелічених компонентів у наступних підрозділах надано опис, розглянуто кілька рішень та обґрунтовано вибір оптимального варіанту.

2.1 Мікроконтролер

Мікроконтролер є центральним елементом розроблюваної системи, що забезпечує розробку голосових команд від модуля розпізнавання голосу та формування керуючих сигналів для LED стрічки. Для вибору оптимального рішення розглянуто три варіанти що є дуже розповсюдженими у навчальних проектах.

2.1.1 Мікроконтролер Arduino UNO

Arduino UNO [7], зображений на рис. 2.1, є найбільш поширеним мікроконтролером серії Arduino.

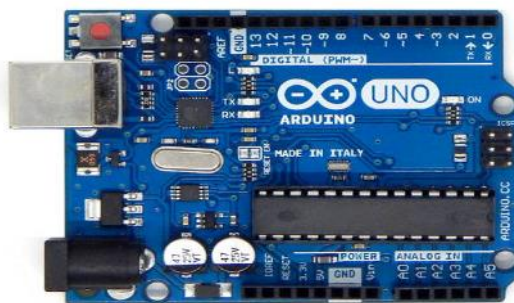


Рисунок 2.1 — Зовнішній вигляд Arduino UNO

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

15

Він виконаний у стандартному форм-факторі, що є еталоном для розробки прототипів пристроїв. Побудований на базі чипа ATmega328P з тактовою частотою 16 МГц.

Нативна логіка 5В забезпечує повну сумісність з більшістю LED стрічок без перетворювачів рівнів. Проте, більшість сучасних модулів розпізнавання голосу, зокрема обраний DFRobot Gravity SEN0539 працюють з логікою 3.3В. Підключення таких модулів потребує використання перетворювачів рівнів I2C для запобігання пошкодження модуля.

2.1.2 Мікроконтролер STM32

STM32 [8] — 32-бітний мікроконтролер промислового стандарту, що пропонує високу обчислювальну потужність на базі ядра ARM Cortex-M. Серія охоплює різноманітні моделі, але для навчальних проектів зазвичай використовуються серії STM32F1 та STM32F4 з тактовою частотою 72-168 МГц. Зовнішній вигляд мікроконтролера зображено на рис. 2.2.



Рисунок 2.2 — Зовнішній вигляд STM32

Змін.	Арк.	№ докум.	Підпис	Дата

Перевагами є висока продуктивність, низьке енергоспоживання та підтримка промислових протоколів. Проте для задач простішого навчального рівня має значні недоліки, такі як суттєво складніше середовище порівняно з Arduino IDE та необхідність програматора ST-Link. Також, значними недоліком є вища вартість порівняно з ESP32 та Arduino UNO.

2.1.3 Мікроконтролер ESP32

ESP32 [9] — 32-бітний мікроконтролер компанії Espressif Systems на базі ядра Xtensa LX6 з тактовою частотою до 240 МГц та 520 КБ оперативної пам'яті. Зовнішній вигляд мікроконтролера зображено на рис. 2.3.



Рисунок 2.3 — Зовнішній вигляд ESP32

Ключовою особливістю є вбудовані Wi-Fi та Bluetooth модулі, що є дуже корисним для IoT проектів і робить його найпопулярнішим рішенням. Підтримує I2C, SPI, UART інтерфейси.

Великою перевагою є сумісність з Arduino IDE через офіційний пакет підтримки, що значно спрощує розробку навчальних проектів. 3.3В логіка

					ІАЛЦ.468300.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		17

забезпечує комфортну роботу з модулем DFRobot Gravity SEN0539, без необхідності використання перетворювачів рівнів. Також варто зазначити, що обсяг оперативної пам'яті 520 КБ є більш ніж достатнім для одночасного зберігання буферів даних та світлових ефектів.

2.1.4 Обґрунтування вибору мікроконтролера

Для реалізації роботи обрано мікроконтролер ESP32 з наступних причин:

1. Arduino UNO має ряд обмежень, включаючи недостатній обсяг оперативної пам'яті, всього 2 КБ. Цього недостатньо для одночасного зберігання світлових ефектів та буферів I2C комунікацій. Також, тактова частота 16МГц є занадто низькою для одночасної обробки голосових команд та керування LED стрічкою.
2. STM32 має кращу продуктивність за Arduino UNO, але складніше середовище розробки, необхідність окремого програматора та значно вища ціна роблять його надлишковим рішенням для даної роботи.
3. ESP32 забезпечує баланс між продуктивністю, функціоналом та простотою розробки. Тактова частота та оперативна пам'ять є більш ніж достатніми для одночасної обробки голосових команд та керування LED стрічкою. Також, пряма сумісність DFRobot за допомогою 3.3В логіки суттєво спрощує підключення, прибираючи необхідність використання перетворювачів рівнів.

2.2 Модуль розпізнавання голосу

Модуль розпізнавання голосу є ключовим компонентом системи. Він забезпечує перетворення голосових команд у цифровий сигнал для подальшої обробки. Основною вимогою є можливість офлайн розпізнавання голосу без

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

18

потрібності у Wi-Fi та Bluetooth підключення та незалежність від голосу користувача. Для вибору оптимального рішення було розглянуто 3 рішення з різними підходами до реалізації.

2.2.1 Модуль розпізнавання голосу Elechouse V3

Elechouse Voice Recognition Module V3 [10], зображений на рис. 2.4, — один з найпоширеніших модулів розпізнавання голосу для любительських та навчальних проектів.



Рисунок 2.4 — Зовнішній вигляд Elechouse Voice Recognition Module V3

Працює на базі алгоритму DTW що порівнює акустичні шаблони записаного голосу. DTW [11] є методом вимірювання схожості між двома часовими послідовностями що можуть відрізнятись за швидкістю або тривалістю. В контексті розпізнавання мовлення алгоритм знаходить оптимальне нелінійне вирівнювання між еталонним акустичним шаблоном записаної команди та вхідним мовним сигналом — це дає змогу коректно розпізнавати команди навіть якщо користувач вимовляє їх з різною швидкістю при кожному зверненні. Алгоритм є одним з найпростіших та найменш ресурсомістких підходів до розпізнавання мовлення що робить його придатним для реалізації на простих вбудованих мікроконтролерах з обмеженими обчислювальними ресурсами.

Перед використанням, кожен команду треба записати тричі для формування шаблону, що робить цей модуль повністю залежним від спікера — він розпізнає голос виключно користувача, що записував команди. Він підтримує до 80 команд, але тільки 7 можуть бути одночасно активними.

Найбільшою перевагою є низька вартість та широка доступна документація. Головними недоліками є залежність від голосу конкретного користувача, що не дає змогу використовувати кількома людьми одночасно та обмежена кількість одночасно активних команд. Також варто зазначити незручність від необхідності записувати кожен команду окремо, що займає багато часу.

2.2.2 Модуль розпізнавання голосу LD3320

LD3320 [12] — спеціалізований чип розпізнавання мовлення, призначений для вбудованих офлайн-систем. На відміну від попереднього, не є залежним від спікера, тобто не потребує попереднього тренування та розпізнає будь-який голос. Його зовнішній вигляд представлено на рис. 2.5.



Рисунок 2.5 — Зовнішній вигляд LD3320

Для розпізнавання голосових команд цей чип використовує НММ, що є статистичною моделлю що описує систему через послідовність прихованих

Змін.	Арк.	№ докум.	Підпис	Дата

станів [13]. У контексті розпізнавання мовлення кожен стан відповідає певному акустичному сегменту звуку. Алгоритм моделює мовлення як послідовність імовірнісних переходів між станами де кожен стан генерує спостережувані акустичні ознаки з певною ймовірністю. Це дає змогу системі розпізнавати команди незалежно від індивідуальних особливостей голосу різних користувачів, оскільки модель оперує статистичними характеристиками мовлення а не конкретними акустичними шаблонами як у випадку DTW. Саме ця властивість забезпечує SI розпізнавання голосу будь-якого користувача.

Проте чип має значні недоліки — документація переважно доступна лише китайською мовою, що ускладнює налаштування, розробку та підключення. Також точність розпізнавання є значно нижчою порівняно з іншими сучасними рішеннями, особливо в умовах фонового шуму.

2.2.3 Модуль розпізнавання голосу DFRobot Gravity SEN0539

DFRobot Gravity SEN0539 [14], представлений на рис. 2.6, — це офлайн модуль розпізнавання голосу, побудований на базі спеціалізованого чипа, з вбудованим нейромережевим процесором. Може підключитися до мікроконтролера як за допомогою інтерфейсу UART, так і I2C та живиться від напруги 3.3В.



Рисунок 2.6 — Зовнішній вигляд DFRobot Gravity SEN0539

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

21

Модуль є незалежним від голосу користувача, бо використовує нейромережевий класифікатор для розпізнавання будь-якого голосу. Нейромережеві класифікатори [15] є універсальними апроксиматорами функцій що здатні навчитись розпізнавати складні патерни у вхідних даних на основі великої навчальної вибірки. В контексті розпізнавання мовлення нейромережа навчається виділяти характерні акустичні ознаки конкретних слів та команд незалежно від індивідуальних особливостей голосу — тембру, висоти тону та акценту різних користувачів. На відміну від DTW що порівнює вхідний сигнал з конкретним записаним шаблоном, нейромережевий класифікатор узагальнює знання отримані під час навчання на великій кількості різних голосів що і забезпечує SI розпізнавання. У випадку модуля DFRobot Gravity SEN0539 нейромережа попередньо навчена виробником на великій вибірці голосових даних та вбудована безпосередньо в чип DF2301Q що дає змогу виконувати розпізнавання локально без підключення до інтернету.

Модуль підтримує 121 вбудовану голосову команду англійською мовою та 17 слотів для користувацьких команд, що записуються через програмний інтерфейс. Користувацькі команди потребують самостійного навчання, але залишаються SI та не є обов'язковими, так як вбудовані команди модулю охоплюють достатньо команд керування світлом для комфортного використання. Згідно документації, голосовий модуль забезпечує розпізнавання голосових команд на відстані до 3 метрів, але на практиці, залежно від фонового шуму, оточення та гучності вимови команд, може розпізнавати їх і на більших відстанях.

					ІАЛЦ.468300.004 ПЗ	Арк.
						22
Змін.	Арк.	№ докум.	Підпис	Дата		

2.2.4 Обґрунтування вибору модулю розпізнавання голосу

Для реалізації обрано модуль DFRobot Gravity SEN0539 з наступних міркувань:

1. Elechouse Voice Recognition Module V3 попри низьку вартість та детальну документацію, є залежним від голосу користувача та потребує тренування для кожної команди окремо. Також, він лише підтримує 7 одночасно активних команд, що може бути не достатнім для керування системою освітлення.
2. LD3320 є незалежним від голоса користувача, але низька точність в умовах шуму, відсутність якісної документації та необхідність додаткових компонентів для підключення роблять його не найкращим рішенням для проекту.
3. DFRobot Gravity SEN0539 є оптимальним рішенням, бо поєднує в собі незалежність від голосу користувача, детальну документацію для підключення, програмування і використання, офлайн розпізнавання на базі неймережевого процесора та офіційна бібліотека для Arduino IDE. Також, пристрій має 121 вбудовану команду та 17 слотів для користувацького налаштування.

2.3 Адресна LED стрічка

LED стрічка є вихідним пристроєм системи. Вона забезпечує візуальне відображення результату обробки голосових команд. Такі стрічки дозволяють керувати кожним діодом окремо, що надає можливість створювати різноманітні світлові ефекти. Для вибору оптимального рішення розглянемо три типи адресних стрічок з різними характеристиками та інтерфейсами передачі даних.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

23

2.3.1 Адресна LED стрічка APA102 (SK9822)

APA102 [16] — адресна стрічка, що використовує інтерфейс SPI. Зовнішній вигляд її світлодіода зображено на рис. 2.7.



Рисунок 2.7 — Зовнішній вигляд APA102 (SK9822)

Особливістю стрічки є те, що вона використовує лише два дроти даних — тактовий сигнал та лінію даних. Це забезпечує вищу частоту оновлення, проте збільшує кількість необхідних GPIO пінів до двох, що ускладнює розводку та збільшує ризик виникнення конфліктів між периферійними пристроями при проектуванні системи де також використовуються інші інтерфейси, зокрема голосовий модуль. Також недоліками є вища вартість та менша поширеність.

2.3.2 Адресна LED стрічка SK6812

SK6812 [17] — адресна LED стрічка на інтерфейсі 1-Wire, де дані передаються послідовно через один дріт. Зовнішній вигляд світлодіода LED стрічки представлено на рис. 2.8.



Рисунок 2.8 — Зовнішній вигляд SK6812

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

24

K6812 має додатковий білий канал — кожен піксель містить чотири діоди замість трьох, що забезпечує більш точніше відтворення білого кольору в порівнянні з іншими стрічками.

Для задач з керуванням кольоровим RGB освітленням додатковий канал не є необхідним і тільки ускладнює керування стрічкою. Додатковими недоліками, як і у випадку з APA102 є вища вартість та менша доступність, що є важливими у навчальних проектах.

2.3.3 Адресна LED стрічка WS2812B

Адресна стрічка WS2812B [18], світлодіод якої зображено на рис. 2.9, — це адресна стрічка, що як і попередня SK6812 використовує 1-Wire інтерфейс та є найпопулярнішим вибором у навчальних проектах через свою доступність та невисоку вартість. Кожен піксель містить 3 світлодіоди (червоний, зелений, синій) з вбудованим драйвером, що забезпечує керування яскравістю кожного каналу та надає можливість відтворення до 16 мільйонів кольорів.



Рисунок 2.9 — Зовнішній вигляд WS2812B

Також варто зазначити, що бібліотека Adafruit Neopixel забезпечує повну підтримку WS2812B для обраного мікроконтролера ESP32.

Змін.	Арк.	№ докум.	Підпис	Дата

2.3.4 Обґрунтування вибору адресної LED стрічки

Для реалізації проекту обрано LED стрічку WS2812B з наступних причин:

1. APA102 забезпечує вищу частоту порівняно з іншими стрічками завдяки інтерфейсу SPI, але наявність двох ліній даних збільшує кількість потрібних GPIO пінів, що ускладнює підключення при наявності декількох периферійних пристроїв. Також вища вартість та нижча доступність є великими недоліками для навчальних проектів.
2. SK6812 є близьким до WS2812B рішенням, але великою відмінністю є наявність додаткового білого каналу ускладнює керування і не є необхідним для навчального проекту.
3. WS2812B є оптимальним рішенням що має баланс між функціональністю, простотою у підключенні та використанні, ціною та доступністю в порівнянні з іншими розглянутими стрічками.

2.4 Поєднання обраних засобів

Обрані компоненти ESP32, DFRobot Gravity SEN0539 та WS2812B утворюють взаємосумісну апаратну платформу.

ESP32 та SEN0539 взаємодіють через інтерфейс I2C що забезпечується сумісністю їх 3.3В логіки без необхідності додаткових перетворювачів рівнів. Живлення модуля здійснюється від виводу 3В вбудованого стабілізатора мікроконтролера.

ESP32 та WS2812B взаємодіють через однодротовий інтерфейс 1-Wire з використанням бібліотеки Adafruit Neopixel що має офіційну підтримку даної платформи. Стрічка живиться від виводу 5В, що забезпечує достатню напругу безпосередньо від USB з'єднання.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

26

Така архітектура забезпечує мінімальну кількість з'єднань та компонентів при максимальній функціональності системи — для підключення всіх трьох компонентів використовується лише одне джерело живлення.

2.5 Висновки до другого розділу

У розділі розглянуто апаратні засоби для розробки системи голосового керування LED стрічкою. За результатами можна виділити наступне:

1. Виконано огляд мікроконтролерів, а саме — Arduino UNO, STM32, ESP32. Описано їх основні технічні характеристики, особливості, переваги та недоліки. За результатами огляду для реалізації обрано мікроконтролер ESP32 як оптимальне рішення, що забезпечує необхідну продуктивність, сумісність з усіма компонентами системи та простоту розробки в середовищі Arduino IDE.
2. Виконано огляд модулів розпізнавання голосу, таких як Elechouse Voice Recognition Module V3, LD3320, DFRobot Gravity SEN0539. За результатами огляду для реалізації системи обрано модуль DFRobot Gravity SEN0539, він забезпечує SI офлайн розпізнавання на базі нейромережевого процесора з достатньою кількістю вбудованих команд, високою точністю розпізнавання голосу та сумісністю з ESP32.
3. Виконано огляд адресних LED стрічок, зокрема APA102, SK6812, WS2812B. За результатами огляду обрано WS2812B як адресну LED стрічку для реалізації системи. Вона забезпечує баланс між простотою підключення через однодротовий інтерфейс, підтримкою бібліотеки Adafruit Neopixel та доступністю
4. Обґрунтовано вибір компонентів для реалізації системи. Сукупність обраних компонентів забезпечує повністю автономну офлайн роботу системи, незалежне від голосу користувача розпізнавання голосу та широкі можливості керування кольоровим адресним освітленням.

					ІАЛЦ.468300.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		27

РОЗДІЛ 3

РОЗРОБКА СИСТЕМИ ГОЛОСОВОГО КЕРУВАННЯ

3.1 Загальна архітектура системи

Розроблена система є програмно-апаратним комплексом, що реалізує голосове керування адресною LED стрічкою без підключення до мережі Інтернет, Bluetooth та інших зовнішніх сервісів. Загальну архітектуру системи наведено у додатку А.

Система складається з трьох компонентів та програмного забезпечення що виконується на мікроконтролері:

1. DFRobot Gravity SEN0539 — модуль розпізнавання голосу, є вхідним пристроєм системи. Забезпечує розпізнавання голосових команд безпосередньо на модулі та передачу їх мікроконтролеру через інтерфейс I2C.

2. ESP32 — центральний елемент системи що виконує програмну логіку керування. Отримує ідентифікатори керування, передані голосовим модулем, визначає відповідний ефект, формує та відправляє керуючий сигнал для LED стрічки.

3. WS2812B — адресна LED стрічка, є вихідним пристроєм системи. Отримує керуючий сигнал від мікроконтролера через 1-Wire інтерфейс та відображає відповідний світловий ефект.

Взаємодія між компонентами здійснюється за допомогою двох інтерфейсів передачі даних:

1. Зв'язок між мікроконтролером ESP32 та модулем SEN0539 реалізовано за допомогою модуля I2C за використання двох ліній — SDA та SCL. Цей інтерфейс було обрано замість альтернативного UART. Обидва інтерфейси використовують дві лінії передачі даних, але I2C має суттєву перевагу — шина I2C надає можливість підключення інших пристроїв на ті самі піни SDA та SCL без збільшення кількості з'єднань тоді як UART є точковим з'єднанням між мікроконтролером та пристроєм і потребує окремих

					ІАЛЦ.468300.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		28

пінів для кожного пристрою. Цей вибір забезпечує можливість розширення системи в майбутньому без збільшення кількості використаних пінів мікроконтролера.

2. Зв'язок між адресною LED стрічкою та мікроконтролером ESP32 реалізовано через однодротовий інтерфейс 1-Wire. Протокол передачі даних стрічки базується на кодуванні бітів тривалістю імпульсів — кожен піксель отримує 24 біти (по 8 біт на кожен канал), зчитує свій пакет та передає решту наступному в ланцюжку. Це дає змогу керувати всіма світлодіодами через один пін мікроконтролера.

Програмне забезпечення складається з двох файлів — головного файлу керування VoiceLED.ino та файлу, що містить світлові ефекти effects.h. Для керування стрічкою використовується бібліотека Adafruit NeoPixel, а для роботи з голосовим модулем — DFRobot_DF2301Q.

Принцип роботи системи наступний: після ініціалізації всіх компонентів система переходить у режим очікування. Голосовий модуль SEN0539 постійно аналізує вхідний сигнал та чекає на команду пробудження — “Hello robot”, яку користувач може змінити за бажанням. Після розпізнавання команди пробудження, модуль аналізує вхідний сигнал на наявність запрограмованих команд. Далі, при розпізнаванні команди передає її числовий ідентифікатор до ESP32. Мікроконтролер зчитує отриманий ідентифікатор у головному циклі програми, знаходить відповідний світловий ефект та передає керуючий сигнал WS2812B через 1-Wire.

Система працює повністю автономно, після підключення живлення не потребує додаткових пристроїв або підключення до мережі. Єдиним засобом взаємодії користувача з системою є голос.

3.2 Схема підключення

Схему підключення компонентів системи наведено у додатку Г. Система має два кола живлення — 3.3В для DFRobot та 5В для LED стрічки.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

29

Модуль розпізнавання голосу підключається до мікроконтролера через інтерфейс I2C відповідно до таблиці 3.1.

Таблиця 3.1 — Підключення DFRobot SEN0539 до ESP32

DFRobot SEN0539	ESP32	Призначення
VCC	3.3V	Живлення
GND	GND	Спільна земля
SDA	GPIO 16	Лінія даних I2C
SCL	GPIO 17	Тактова лінія I2C

Живлення модуля здійснюється від виводу 3.3V ESP32. Це забезпечує пряму сумісність рівнів логічних сигналів.

Адресна стрічка підключається до ESP32 відповідно до таблиці 3.2.

Таблиця 3.2 — Підключення WS2812B до ESP32

WS2812B	ESP32	Призначення
VCC	5V	Живлення
GND	GND	Спільна земля
DIN	GPIO 4	Лінія даних 1-Wire

Живлення стрічки забезпечується виводом 5V ESP32, що забезпечує напругу 5В безпосередньо від USB з'єднання. При необхідності у використанні стрічок більшої довжини або вищої яскравості рекомендується підключення зовнішнього блоку живлення 5В безпосередньо до стрічки із забезпеченням спільного потенціалу землі з ESP32.

Для забезпечення стабільної роботи системи використовується два захисних компоненти.

Резистор номіналом 1кОм, що розміщений на лінії даних між GPIO 4 та входом DIN стрічки. Він обмежує швидкість наростання фронтів імпульсів що запобігає виникненню відбиттів сигналу на лінії передачі та захищає вивід мікроконтролера від можливих перешкод з боку стрічки.

Електролітичний конденсатор ємністю 1000 мкФ підключається паралельно до ліній VCC та GND безпосередньо біля входу стрічки. При підключенні живлення виникає короточасний струмовий сплеск що може пошкодити перші світлодіоди стрічки — конденсатор згладжує цей сплеск та стабілізує напругу живлення під час різких змін споживаного струму при перемиканні світлодіодів.

3.3 Програмна реалізація

Програмне забезпечення реалізовано мовою C++ в середовищі Arduino IDE. Воно складається з двох файлів — головного файлу керування VoiceLED.ino та файлу, що містить світлові ефекти effects.h. Така структура забезпечує модульність проекту та спрощує можливість його розширення у майбутньому. Додавання нового ефекту потребує змін лише у файлі effects.h без необхідності модифікації файлу з основною логікою програми.

3.3.1 Ініціалізація системи

Ініціалізація системи виконується у функції setup() що викликається лише один раз на початку роботи при підключенні живлення. Процес ініціалізації складається з трьох етапів — ініціалізація LED стрічки, ініціалізація голосового модуля та запуск стартової анімації.

Змін.	Арк.	№ докум.	Підпис	Дата

Виконується ініціалізація бібліотеки Adafruit Neopixel та налаштування початкового стану стрічки. Для візуальної індикації стану ініціалізації стрічка підсвічується синім кольором. Реалізацію показано на рис. 3.1.

```

123 strip.begin();
124 strip.setBrightness(currentBrightness);
125 strip.clear();
126 strip.show();
127
128 strip.fill(strip.Color(0, 0, 255));
129 strip.show();
130 Serial.println("LEDS INITIALIZED\n");

```

Рисунок 3.1 — Ініціалізація LED стрічки

Далі ініціалізується шина I2C з зазначенням пінів GPIO 16 та 17 як SDA та SCL. Виконується підключення голосового модуля SEN0539. У разі виникнення помилки під час підключення система виводить повідомлення про помилку в Serial Monitor, а LED стрічка підсвічується червоним кольором. Повторна спроба підключення повторюється кожні 3 секунди. Реалізацію показано на рис. 3.2.

```

132 delay(2000);
133 Wire.begin(21, 22);
134 while (!(asr.begin())) {
135     Serial.println("DFROBOT SETUP FAILED\n");
136     strip.fill(strip.Color(255, 0, 0));
137     strip.show();
138     delay(3000);
139 }
140 Serial.println("DFROBOT INITIALIZED");

```

Рисунок 3.2 — Ініціалізація шини I2C та голосового модуля

Після успішної ініціалізації голосового модуля встановлюються його параметри — гучність, вимкнення режиму тиші та час активного очікування команди після розпізнавання команди пробудження.

```

142   asr.setVolume(7);
143   asr.setMuteMode(0);
144   asr.setWakeTime(20);

```

Рисунок 3.3 — Ініціалізація параметрів голосового модуля

Завершальним етапом є запуск стартової анімації, реалізація якої показана на рис. 3.4, що послідовно підсвічує пікселі блакитним кольором. Вона візуально підтверджує готовність системи до роботи.

```

106 void startupAnimation() {
107     for (int i = 0; i < strip.numPixels(); i++) {
108         strip.setPixelColor(i, strip.Color(0, 100, 255));
109         strip.setPixelColor((i + 1) % strip.numPixels(), strip.Color(0, 50, 128));
110         strip.show();
111         delay(40);
112     }
113     strip.clear();
114     strip.show();
115 }

```

Рисунок 3.4 — Стартова анімація

3.3.2 Отримання голосової команди

Отримання голосової команди реалізовано у головному циклі програми loop() через виклик методу getCMDID(), який зображено на рис. 3.5. Метод повертає числовий ідентифікатор розпізнаної команди або 0 якщо команда не була розпізнана.

```

150 void loop() {
151     uint8_t CMDID = asr.getCMDID();

```

Рисунок 3.5 — Метод отримання голосової команди

Модуль SEN0539 працює в двох режимах — режимі очікування та активному режимі. В режимі очікування модуль постійно аналізує вхідний аудіосигнал на наявність команди пробудження “Hello Robot”. Після розпізнавання команди пробудження модуль переходить в активний режим та

протягом встановленого часу очікування аналізує вхідний сигнал на наявність запрограмованих команд.

3.3.3 Обробка команди

Обробка отриманого ідентифікатора команди реалізована у два етапи. На першому етапі виконується пошук ідентифікатора у таблиці кольорів — масиві структур Colors, зображеному на рис. 3.6, що зберігає відповідності між ідентифікаторами команд та значеннями RGB каналів.

```
46 struct Color{
47     uint8_t cmdID;
48     uint8_t r, g, b;
49 };
50
51 const Color colorCommands[] = {
52     {116, 255, 0, 0}, // red - червоний
53     {117, 255, 128, 0}, // orange - помаранчевий
54     {118, 255, 255, 0}, // yellow - жовтий
55     {119, 0, 255, 0}, // green - зелений
56     {120, 0, 255, 255}, // cyan - блакитний
57     {121, 0, 128, 255}, // blue - синій
58     {122, 153, 51, 255}, // purple - фіолетовий
59     {123, 255, 255, 255}, // white - білий
60 };
```

Рисунок 3.6 — Таблиця кольорів

Використання таблиці кольорів замість окремих case операторів забезпечує масштабованість, тобто додавання нового кольору потребує лише одного нового рядка в таблиці без зміни логіки обробки. При знаходженні відповідності ідентифікатор скидається до 0 для запобігання подальшій обробці в конструкції switch/case.

Якщо команда не була розпізнана на першому етапі, вона переходить на другий етап для обробки в конструкції switch/case що розподіляє їх за функціональними групами — керування живленням, яскравістю, колірною температурою, режимами та кастомними ефектами.

Для керування поточним режимом роботи використовується перелічуваний тип `enum Mode`, показаний на рис. 3.7, що визначає дев'ять можливих станів.

```
22  enum Mode {
23      MODE_STATIC, // static color – статичний колір
24      MODE_RAINBOW, // rainbow – райдуга
25      MODE_BREATHING, // pulsation – пульсація
26      MODE_CHASE, // running light – біжучий вогонь
27      MODE_TWINKLE, // twinkle – мерехтіння
28      MODE_FIRE, // fire – імітація вогню
29      MODE_ALARM, // alarm – тривога
30      MODE_RELAX, // relaxation – режим відпочинку
31      MODE_PARTY // party – вечірковий режим
32  };
```

Рисунок 3.7 — Динамічні режими

Активація будь-якого динамічного режиму виконується через функцію `activateMode()`, зображену на рис. 3.8, що централізовано керує переходом між режимами — встановлює новий режим, скидає всі змінні стану ефектів до початкових значень та налаштовує яскравість.

```
104 void activateMode(Mode mode, int brightness = -1) {
105     currentMode = mode;
106     isOn = true;
107     breathBrightness = 0;
108     breathDirection = true;
109     chasePosition = 0;
110     alarmState = false;
111     relaxBrightness = 5;
112     relaxDirection = true;
113     rainbowHue = 0;
114     partyHue = 0;
115     if (brightness >= 0) strip.setBrightness(brightness);
116     else strip.setBrightness(currentBrightness);
117 }
```

Рисунок 3.8 — Функція активації динамічних режимів

Скидання змінних стану при кожному переході між режимами забезпечує коректний старт кожного ефекту незалежно від попереднього стану системи.

3.3.4 Формування керуючого сигналу

Формування керуючого сигналу для LED стрічки реалізовано за допомогою бібліотеки Adafruit Neopixel що забезпечує повну підтримку WS2812B.

Основними методами бібліотеки що використовуються є:

1. `strip.fill()` — заповнення всіх пікселів одним кольором;
2. `strip.setPixelColor()` — встановлення кольору окремого пікселя;
3. `strip.setBrightness()` — встановлення яскравості;
4. `strip.show()` — передача буферу даних до стрічки;
5. `strip.clear()` — очищення буферу пікселів.

Виклик методу `strip.show()` є обов'язковим після зміни стану пікселів. Лише після його виклику дані з буферу передаються до стрічки.

3.3.5 Реалізація світлових ефектів

Світлові ефекти реалізовано в окремому файлі `effects.h` що забезпечує модульність коду. Кожен ефект реалізований як окрема функція з префіксом `tick` що викликається циклічно з головного циклу програми.

Ключовою особливістю реалізації є використання функції `millis()` що повертає кількість мілісекунд з моменту запуску. На відміну від функції `delay()` що повністю зупиняє виконання програми на заданий час, підхід з `millis()` дає змогу системі отримувати нові голосові команди під час відтворення анімацій. Принцип роботи коду, зображеного на рис. 3.11, полягає у порівнянні

Змін.	Арк.	№ докум.	Підпис	Дата

поточного часу з часом попереднього оновлення ефекту — якщо різниця перевищує встановлений інтервал, ефект оновлюється.

```
247 unsigned long now = millis();
248
249 switch (currentMode) {
250     case MODE_RAINBOW:
251         if (now - lastUpdate > 20) { tickRainbow(); lastUpdate = now; } break;
252     case MODE_BREATHING:
```

Рисунок 3.11 — Оновлення ефектів

3.4 Реалізовані функції системи

Розроблена система реалізує керування адресною LED стрічкою через голосові команди що розподілені за функціональними групами. Повний перелік команд наведено в таблиці 3.3.

Таблиця 3.3 — Перелік команд

ID	Команда	Опис
5	Користувацька — Twinkle	Мерехтіння
6	Користувацька — Fire	Вогонь
7	Користувацька — Breathe	Пульсація
8	Користувацька — Chase	Біжучий вогонь
9	Користувацька — Alarm	Сигнал тривоги
10	Користувацька — Relax	Режим відпочинку
103	Turn on the light	Увімкнути світло
104	Turn off the light	Вимкнути світло
105	Brighten the light	Збільшити яскравість
106	Dim the light	Зменшити яскравість

Продовження таблиці 3.3

ID	Команда	Опис
107	Adjust brightness to maximum	Максимальна яскравість
108	Adjust brightness to minimum	Мінімальна яскравість
109	Increase color temperature	Збільшити температуру світла
110	Decrease color temperature	Зменшити температуру світла
113	Daylight mode	Денний режим
114	Moonlight mode	Нічний режим
115	Color mode	Райдуга
116	Set to red	Червоний
117	Set to orange	Помаранчевий
118	Set to yellow	Жовтий
119	Set to green	Зелений
120	Set to cyan	Блакитний
121	Set to blue	Синій
122	Set to purple	Фіолетовий
123	Set to white	Білий
128	Cool mode	Холодний режим
129	Heat mode	Теплий режим

Реалізовані функції розподілено за такими групами:

1. Керування живленням — команди вмикання та вимикання стрічки (ID 103, 104).

2. Керування яскравістю — чотири команди що забезпечують як плавне регулювання яскравості, так і повну зміну яскравості до мінімального або максимального рівня (ID 105-108). Мінімальна яскравість обмежена на рівні 5 замість 0, що забезпечує видиму підсвітку навіть при мінімальній яскравості.

3. Керування колірною температурою — дві команди що забезпечують плавне регулювання відтінку білого освітлення (ID 109, 110).

4. Керування кольором — вісім команд що встановлюють статичний колір стрічки (ID 116-123).

5. Режими освітлення — чотири попередньо налаштованих режими: денний, місячний, холодний та теплий (ID 113-115, 128, 129).

6. Динамічні ефекти — сім користувацьких команд що активують анімаційні режими (ID 5-10).

3.5 Порівняння з аналогами

Таблиця 3.4 — Порівняння з існуючими системами

Характеристика	GL-C-310WL	GL-C-211WL	SUPERNIG T	Voiceemit y DW SD	Voiceemit y DW SI	Розроблена система
Тип керування	Додаток Wi-Fi	Додаток Wi-Fi	Додаток Bluetooth	Голос	Голос	Голос
Інтерфейс зв'язку	Wi-Fi	Wi-Fi	Bluetooth	Офлайн	Офлайн	Офлайн
Тип стрічки	Адресна	Аналогова	Аналогова	Біла	Біла	Адресна
Голосове керування	Через розумних помічників	Через розумних помічників	Через ПЗ	Вбудоване	Вбудоване	Вбудоване
Speaker Independent	Так	Так	Так	Ні	Так	Так

Продовження таблиці 3.4

Характеристика	GL-C-310WL	GL-C-211WL	SUPERNIGHT	Voiceemity DW SD	Voiceemity DW SI	Розроблена система
Дистанція розпізнавання	Залежно від зовн. пристрою	Залежно від зовн. пристрою	Залежно від зовн. пристрою	До 1м	До 5м	До 3м
ПЗ	WLED	WLED	HappyLighting	-	-	Власне ПЗ
Живлення	5-24В	12-24В	12-24В	11-27В	11-27В	3-5В
Ціна	\$15-20	\$15-20	\$15-20	-	-	\$35

Виходячи з порівняння, можна виділити наступні переваги розробленої системи. Розроблена система є єдиним серед розглянутих рішень що одночасно забезпечує повністю автономне офлайн голосове керування без залежності від інтернету чи смартфона, підтримку адресних RGB стрічок типу WS2812B з можливістю індивідуального керування кожним світлодіодом, та відкриту архітектуру з можливістю вільного розширення функціоналу. Кількість реалізованих голосових команд значно перевищує можливості офлайн контролерів Voiceemity що підтримують лише 8 команд. Підтримка колірної температури в поєднанні з керуванням RGB кольорами забезпечує широкі можливості налаштування освітлення під різні сценарії використання.

З недоліків варто зазначити що розроблена система все ще має менший функціонал порівняно з Wi-Fi залежними системами керування, що мають необмежену кількість кольорів та ефектів за допомогою програмного забезпечення WLED. Також, ціна є вищою за існуючі комерційні рішення, що пояснюється використанням спеціалізованого модуля офлайн розпізнавання голосу DFRobot Gravity SEN0539 з вбудованим нейромережевим процесором — саме цей компонент є основною складовою вартості та водночас ключовою перевагою системи що забезпечує автономне SI розпізнавання голосу без підключення до інтернету.

					ІАЛЦ.468300.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		40

3.6 Висновки до третього розділу

У даному розділі представлено результати розробки офлайн-системи голосового керування адресною LED стрічкою. За результатами можна виділити наступне:

1. Розроблено загальну архітектуру системи що складається з трьох апаратних компонентів: мікроконтролера ESP32, модуля розпізнавання голосу DFRobot Gravity SEN0539 та адресної LED стрічки WS2812B через інтерфейси 1-Wire та I2C. Система працює повністю автономно без підключення до інтернету чи зовнішніх сервісів.

2. Розроблено схему підключення компонентів системи з підключенням резистора 1 кОм на лінії даних стрічки та конденсатора ємністю 1000 мкФ для захисту від струмових сплесків. Живлення здійснюється від виводу 3.3В мікроконтролера для голосового модуля та від виводу 5В для адресної стрічки.

3. Розроблено програмне забезпечення системи мовою C++ в середовищі Arduino IDE. Воно складається з двох файлів — основного файлу керування VoiceLED.ino та файлу з реалізованими динамічними ефектами effects.h. Реалізовано модульну архітектуру коду з використанням таблиць команд у вигляді масивів структур та не блокуючого підходу до реалізації анімацій через millis() що забезпечує безперервне отримання голосових команд під час відтворення ефектів.

4. Реалізовано 27 голосових команд розподілених між шістьма функціональними групами: керування живленням, керування яскравістю, керування колірною температурою, керування кольором, режими освітлення, динамічні ефекти.

5. За результатами порівняння розробленої системи з існуючими аналогами встановлено, що розроблена система є єдиним рішенням, що поєднує повністю автономне офлайн керування, підтримку адресних RGB LED-стрічок та відкриту архітектуру з можливістю розширення функціоналу.

					ІАЛЦ.468300.004 ПЗ	Арк.
						41
Змін.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 4

ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

Тестування розробленої системи проводилось за п'ятьма критеріями — точність розпізнавання, дистанція розпізнавання, час відгуку системи, стійкість до фонового шуму та стабільності тривалої роботи. Тестування проводиться в приміщенні одним користувачем. Зовнішній вигляд розробленої системи керування зображено на рис. 4.1.

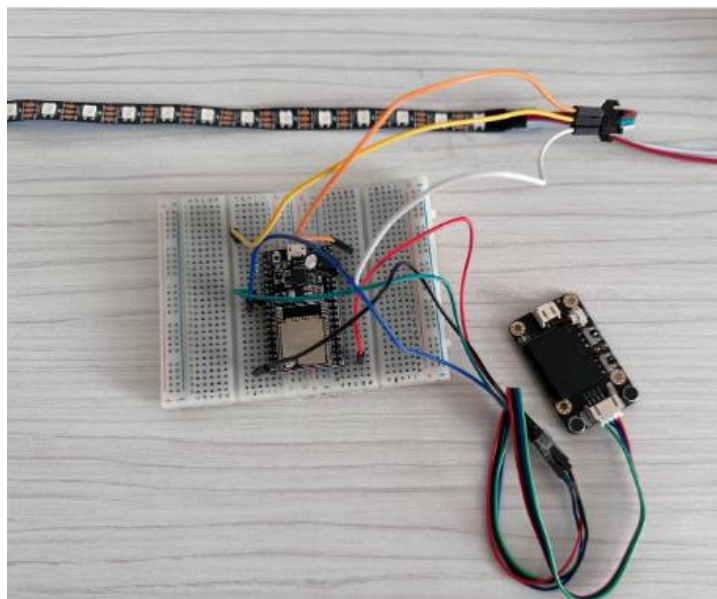


Рисунок 4.1 — Зовнішній вигляд системи

4.1 Точність розпізнавання голосових команд

Для оцінки точності розпізнавання було обрано 10 команд що охоплюють різні функціональні групи системи. Кожна команда повторювалась 10 разів з відстані 1 метр в тихому приміщенні. Результати наведено в таблиці [4.1].

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

42

Таблиця 4.1 — Результати тестування точності розпізнавання команд

Команда	ID	Спроб	Успішних
Turn on the light	103	10	9
Turn off the light	104	10	9
Set to red	116	10	10
Set to blue	121	10	10
Brighten the light	105	10	9
Dim the light	106	10	9
Daylight mode	113	10	9
Moonlight mode	114	10	10
Fire	6	10	10
Alarm	9	10	9
Загальне		100	94

Загалом було успішно розпізнано 94 команди зі 100, що є високим показником для офлайн-системи без підключення до хмарних помічників. Найвищу точність показали команди встановлення конкретних кольорів, місячного режиму та ефекту вогню. Це пояснюється чіткою фонетичною відмінністю цих команд. Решта показала точність 90% що свідчить про стабільну роботу модуля.

Змін.	Арк.	№ докум.	Підпис	Дата

4.2 Дистанція розпізнавання

Тестування дистанції розпізнавання, результати якого можна побачити у таблиці 4.2, було проведено з використанням команди “Turn on the light”. Відстань збільшувалась з кроком 0.5 метра, починаючи від 0.5 метра до 3 метрів, після чого тестувались відстані 4 та 5 метрів.

Таблиця 4.2 — Результати тестування дистанції розпізнавання

Відстань	Спроб	Успішних
0.5 м	5	5
1.0 м	5	5
1.5 м	5	5
2.0 м	5	5
2.5 м	5	5
3.0 м	5	5
4.0 м	5	4
5.0 м	5	4

Система забезпечує стабільне розпізнавання з точністю 100% на відстані до 3 метрів що відповідає заявленим характеристикам модуля SEN0539. На відстані від 4 до 5 метрів, точність знижується до 80% що є гарним результатом враховуючи що офіційно заявлена дистанція розпізнавання 3 метри. Зниження точності на великих відстанях пояснюється затуханням звукового сигналу та збільшенням відношення рівня фонового шуму до рівня корисного сигналу.

4.3 Час відгуку системи

Час відгуку системи було виміряно двома складовими — часом обробки команди програмним забезпеченням та загальним часом відгуку системи від моменту закінчення вимовляння до реакції стрічки. Для вимірювання часу обробки команди було додано вимірювання через функцію `millis()`, показане на рис. 4.1.

```
260 Serial.print("Response time: ");  
261 Serial.print(millis() - responseStart);
```

Рисунок 4.2 — Реалізація вимірювання часу обробки команди

Результати вимірювань наведено в таблиці 4.3.

Таблиця 4.3 — Результати вимірювання часу обробки команди

Команда	ID	Час обробки (мс)
Set to red	116	1
Set to blue	121	1
Set to purple	122	1
Twinkle	5	0
Fire	6	0
Turn on the light	103	1
Turn off the light	104	0
Brighten the light	105	1
Daylight mode	113	1
Moonlight mode	114	0
Середнє		0.6 мс

Загальний час відгуку системи складається з чотирьох послідовних етапів. Першим є етап розпізнавання голосу безпосередньо на чипі DF2301Q модуля SEN0539. Згідно технічної документації виробника цей процес займає від 500 до 800 мс.

Другим етапом є передача числового ідентифікатора розпізнаної команди від голосового модуля до мікроконтролера ESP32 через інтерфейс I2C. Швидкість передачі даних по шині I2C в стандартному режимі складає 100 кбіт/с. Це забезпечує передачу одного байту ідентифікатора за час порядку від 1 до 2 мс.

Третім етапом є обробка отриманого ідентифікатора програмним забезпеченням ESP32. Як показали результати наведені в таблиці 4.3, цей етап в середньому займає 0.6 мс що пояснюється використанням оптимізованих структур даних у вигляді масивів структур та високою тактовою частотою ESP32 240 МГц.

Останнім етапом є передача керуючого сигналу від ESP32 до стрічки WS2812B через однодротовий інтерфейс 1-Wire. Протокол WS2812B передає 24 біти даних на кожен піксель зі швидкістю 800 кбіт/с. Для стрічки з 60 пікселями повне оновлення займає близько 1.8 мс.

Таким чином загальний час відгуку системи від моменту закінчення вимовляння команди до видимої реакції стрічки становить приблизно 500-800 мс що є прийнятним показником для систем голосового керування побутовими пристроями де затримка до 1 секунди є стандартною для подібних рішень.

4.5 Стійкість до фонового шуму

Тестування було проведено з використанням команди “Turn on the light” в чотирьох різних акустичних умовах — в повній тиші, при фоновій розмові, інструментальній музиці та при музиці зі словами. На кожен умову

виконувалось 5 спроб на відстані 1 метр. Результати тестування наведено в таблиці 4.4.

Таблиця 4.4 — Результати тестування стійкості до фонового шуму

Умова	Спроб	Успішних
Тиша	5	5
Фонова розмова	5	4
Музика (мелодія)	5	5
Музика (зі словами)	5	4

Система показала стабільну роботу в тиші та при інструментальній музиці з точністю 100%. Цей результат пояснюється тим що інструментальна музика не містить мовних компонентів і нейромережевий алгоритм модуля не сприймає її як перешкоду для розпізнавання голосових команд.

При наявності фонові розмови та музики зі словами точність знизилась до 80% — одна з п'яти спроб не була розпізнана. Це пояснюється тим що вокал та людська мова на фоні містять мовні компоненти на тих самих частотах що й голосові команди, що створює акустичні перешкоди для алгоритму розпізнавання. Загалом система продемонструвала достатню стійкість до фонового шуму для практичного використання в побутових умовах.

4.6 Стабільність роботи

Тестування стабільності було проведено протягом 60 хвилин безперервної роботи системи. Кожні 10 хвилин була подана голосова команда та було перевірено коректність роботи системи. Було використано різні режими та ефекти для перевірки стабільності всіх реалізованих у системі

функцій. Використані команди та коректність їх виконання подано у таблиці 4.5.

Таблиця 4.5 — Результати тестування стабільності роботи

Команда	ID	Час	Результат
Turn on the light	103	10 хв	Кореткно, без перебоїв та помилок
Set to red	116	20 хв	Кореткно, без перебоїв та помилок
Dim the light	106	30 хв	Кореткно, без перебоїв та помилок
Twinkle	5	40 хв	Кореткно, без перебоїв та помилок
Moonlight mode	114	50 хв	Кореткно, без перебоїв та помилок
Fire	6	60 хв	Кореткно, без перебоїв та помилок

Протягом години безперервної роботи система не показала жодних збоїв, всі голосові команди були розпізнані коректно та відповіді світлові ефекти відтворено без видимих артефактів чи зависань. Перегрів компонентів не спостерігався — мікроконтролер ESP32 та модуль розпізнавання голосу DFRobot Gravity SEN0539 залишились при кімнатній температурі.

4.7 Порівняння розробленої системи з аналогами

Для оцінки розробленої системи відносно комерційних аналогів проведено порівняння за фактичними показниками отриманими в ході тестування та технічними характеристиками з офіційної документації виробників. Результати порівняння наведено в таблці 4.6.

Таблиця 4.6 — Порівняння системи з розглянутими аналогами

Контролер	GL-C-310WL	SUPERNIGHT	Voiceemity DW SD	Voiceemity DW SI	Розроблена система
Тип голосового керування	Через розумного помічника	Через смартфон	Вбудоване офлайн	Вбудоване офлайн	Вбудоване офлайн
Точність розпізнавання	Залежно від зовнішнього пристрою	Залежно від зовнішнього пристрою	Н/Д	Н/Д	94%
Дистанція (100%)	Залежно від зовнішнього пристрою	Залежно від зовнішнього пристрою	1 м	5 м	3 м
Дистанція (80%)	Залежно від зовнішнього пристрою	Залежно від зовнішнього пристрою	1 м	5 м	5 м
Кількість команд	Необмежено	Необмежено	8	8	27
Час відгуку (обробка)	Залежно від зовнішнього пристрою	Залежно від зовнішнього пристрою	Н/Д	Н/Д	0.6 мс
Тип стрічки	Адресна	Аналогова	Біла CCT	Біла CCT	Адресна RGB
Speaker Independent	Так	Так (через смартфон)	Ні	Так	Так
Офлайн робота	Ні	Так	Так	Так	Так
Відкрита архітектура	Так	Ні	Ні	Ні	Так
Вартість	\$15-20	\$10-15	Н/Д	Н/Д	\$35

GL-C-310WL є технічно розвиненим рішенням з підтримкою адресних стрічок та відкритою архітектурою WLED, однак голосове керування реалізоване через хмарні сервіси Amazon Alexa що створює повну залежність

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

49

від інтернету. SUPERNIGHT забезпечує незалежність від інтернету завдяки Bluetooth, проте залишається залежним від смартфона та підтримує лише аналогові стрічки.

Контролери Voiceemity є найближчими аналогами за підходом до голосового керування — обидва використовують вбудоване офлайн розпізнавання. Проте документація виробника не містить даних про точність розпізнавання та час відгуку. За дистанцією розроблена система перевищує DW SD але поступається заявленим 5 метрам DW SI. Кількість команд значно перевищує можливості обох Voiceemity що обмежені лише 8 командами.

Вартість розробленої системи є вищою порівняно з комерційними аналогами що пояснюється використанням спеціалізованого модуля DFRobot Gravity SEN0539 з вбудованим нейромережевим процесором — саме цей компонент є основною складовою вартості та ключовою перевагою системи.

4.8 Висновки до четвертого розділу

У даному розділі проведено комплексне тестування розробленої системи за п'ятьма критеріями. За результатами тестування можна виділити наступне:

1. Проведено тестування точності розпізнавання голосових команд на вибірці з 10 команд різних функціональних груп. Середня точність склала 94% що є високим показником для офлайн-системи без підключення до інтернету.
2. Проведено тестування дистанції розпізнавання. Встановлено що система забезпечує точність 100% до 3 метрів та зберігає працездатність на відстані до 5 метрів з точністю 80% що перевищує офіційно заявлену максимальну дистанцію модуля.
3. Проведено вимірювання часу відгуку. Час обробки команди програмним забезпеченням ESP32 складає в середньому 0.6 мс —

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

50

загальний час відгуку системи визначається переважно часом розпізнавання модулем SEN0539 та становить приблизно 500-800 мс.

4. Проведено тестування стійкості до фонового шуму. Система зберігає точність 100% при інструментальній музиці та знижує її до 80% лише при наявності мовних компонентів у фоні.
5. Проведено тестування стабільності протягом 60 хвилин. Система працювала без жодних збоїв — всі команди розпізнавались коректно та всі ефекти відтворювались без помилок.
6. Проведено порівняння з комерційними аналогами. Розроблена система перевищує Voiceemity DW SD за дистанцією розпізнавання та значно перевершує обидва контролери Voiceemity за кількістю команд та підтримкою адресних RGB стрічок.

					ІАЛЦ.468300.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		51

ВИСНОВКИ

У дипломному проєкті розроблено офлайн-систему голосового керування адресною LED-стрічкою на базі мікроконтролера ESP32 та отримано наступні результати:

1. Проведено аналіз існуючих рішень для керування LED освітленням з голосовим керуванням — WiFi контролерів Gledopto GL-C-310WL та GL-C-211WL, Bluetooth контролера SUPERNIGHT та офлайн голосових контролерів Voiceemity DW SD та DW SI. Встановлено що жодне з розглянутих комерційних рішень не поєднує одночасно повністю автономне офлайн голосове керування, підтримку адресних RGB стрічок та відкриту архітектуру з можливістю розширення що обумовило актуальність даного проєкту.
2. Обґрунтовано вибір апаратних компонентів системи — мікроконтролера ESP32, модуля розпізнавання голосу DFRobot Gravity SEN0539 та адресної LED стрічки WS2812B. Обрані компоненти утворюють взаємосумісну апаратну платформу що забезпечує повністю автономну офлайн роботу без залежності від інтернету чи зовнішніх сервісів.
3. Розроблено програмно-апаратну систему керування адресною LED стрічкою з офлайн розпізнаванням голосових команд. Програмне забезпечення реалізоване мовою C++ в середовищі Arduino IDE та складається з двох модулів — основного файлу керування VoiceLED.ino та файлу світлових ефектів effects.h. Реалізовано 27 голосових команд розподілених за шістьма функціональними групами — керування живленням, яскравістю, колірною температурою, статичними кольорами, режимами освітлення та 6 динамічними світловими ефектами.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

52

4. Проведено комплексне тестування розробленої системи за п'ятьма критеріями. Середня точність розпізнавання голосових команд склала 94%, система забезпечує стабільну роботу на відстані до 3 метрів з точністю 100% та зберігає працездатність до 5 метрів з точністю 80%. Час обробки команди програмним забезпеченням ESP32 складає в середньому 0.6 мс, загальний час відгуку системи становить приблизно 600 мс. Тестування стабільності підтвердило надійність системи — протягом 60 хвилин безперервної роботи не виявлено жодних збоїв.

Розроблена система є єдиним серед розглянутих рішень що поєднує повністю автономне офлайн голосове керування, SI розпізнавання голосу будь-якого користувача без попереднього тренування, підтримку адресних RGB стрічок та відкриту архітектуру з можливістю розширення функціоналу шляхом додавання нових ефектів та команд.

Система має широкі перспективи подальшого розвитку. З програмної точки зору, можливим є збільшення кількості світлових ефектів, додавання підтримки музичної синхронізації, реалізація таймерів та розкладів для автоматичного керування освітленням. З апаратної точки зору система може бути розширена датчиками навколишнього середовища для автоматично зміни кольору залежно від їх параметрів, а також підключенням додаткових LED стрічок через шину I2C.

Змін.	Арк.	№ докум.	Підпис	Дата

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gledopto: (GL-C-310WL) ESP32 WLED Digital LED Controller. URL: <https://gledopto.com/h-pd-124.html> (дата звернення 28.04.2026)
2. Kno.wled.ge: WLED Documentation. URL: <https://kno.wled.ge> (дата звернення 28.04.2026)
3. Gledopto: (GL-C-211WL) ESP32 WLED PWM LED Controller. URL: <https://gledopto.com/h-pd-100.html> (дата звернення 29.04.2026)
4. Amazon: SUPERNIGHT Wireless Bluetooth Controller. URL: <https://www.amazon.com/SUPERNIGHT-Wireless-Bluetooth-Controller-Control/dp/B08SJ513KR> (дата звернення 29.04.2026)
5. Led Professional: Offline Lighting Voice Controller — An Alternative to Voice Assistants. URL: <https://www.led-professional.com/resources-1/articles/offline-lighting-voice-controller-2014-an-alternative-to-voice-assistants> (дата звернення 29.04.2026)
6. Led Professional: Offline Lighting Voice Controller 2014 Review. URL: https://www.led-professional.com/downloads/LpR77_Full_441931.pdf (дата звернення 01.05.2026)
7. Arduino Docs: UNO R3. URL: <https://docs.arduino.cc/hardware/uno-rev3/> (дата звернення 02.05.2026)
8. STM: STM32 Microcontrollers (MCUs). URL: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html> (дата звернення 02.05.2026)
9. Espressif Systems: ESP32. URL: <https://www.espressif.com/en/products/socs/esp32> (дата звернення 03.05.2026)

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.468300.004 ПЗ

Арк.

54

10. Elechouse: Speak Recognition. Voice Recognition Module V3, URL:
<https://www.elechouse.com/product/speak-recognition-voice-recognition-module-v3/> (дата звернення 05.05.2026)
11. Meinard Müller “Dynamic Time Wrapping” (January 2007), 83 p.
12. Xpart Electronic: A Comprehensive Guide to Using the LD3320 Speech Recognition Module in Embedded Systems. URL:
<https://xpart.org/voice-to-control-a-comprehensive-guide-to-using-the-ld3320-speech-recognition-module-in-embedded-systems/> (дата звернення 06.05.2026)
13. Lawrence R. Rabiner “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition” (February 1989), 30 p.
14. DFRobot: Gravity DF2301Q Offline Voice Recognition Sensor. URL:
<https://wiki.dfrobot.com/sen0539-en/> (дата звернення 08.05.2026)
15. Christopher Bishop “Neural Networks for Pattern Recognition” (November 23, 1995), 498 p.
16. Pololu Robotics and Electronics: Addressable RGB 60-LED Strip (SK9822). URL:
<https://www.pololu.com/product/3089> (дата звернення 08.05.2026)
17. Adafruit: Technical Data Sheet SK6812. URL:
<https://cdn-shop.adafruit.com/product-files/1138/SK6812+LED+datasheet+.pdf> (дата звернення 08.05.2026)
18. Adafruit: WS2812B — Intelligent control LED integrated light source. URL:
<https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf> (дата звернення 09.05.2026)

ДОДАТОК А

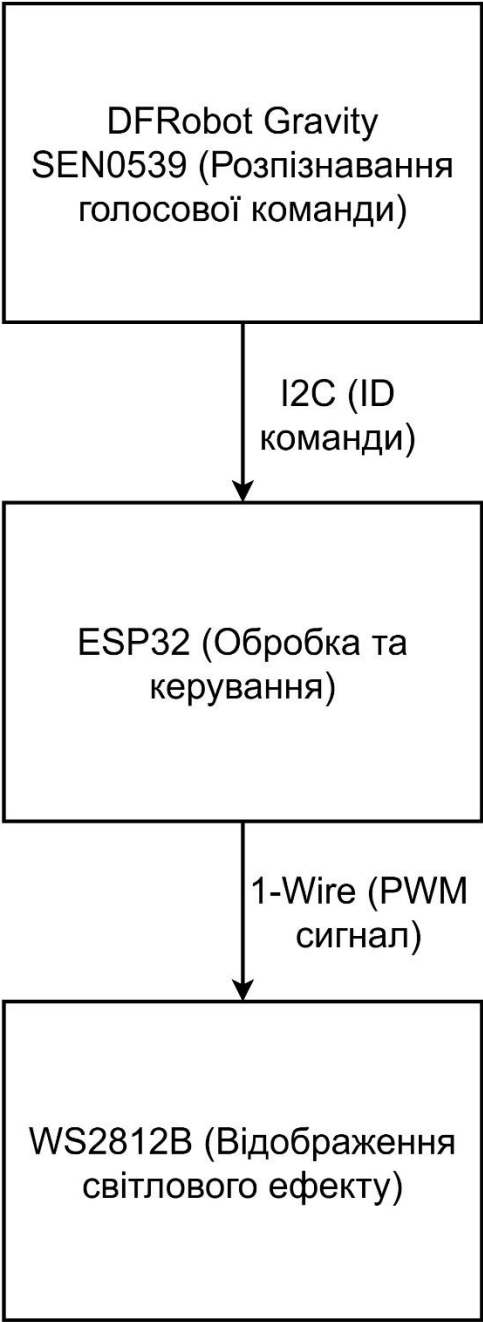
Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32

Схема роботи системи

ІАЛЦ.468300.005 Д1

Аркушів 1

Київ 2026



					ІАЛЦ.468300.005 Д1				
					Офлайн-система голосового керування адресною LED стрічкою на базі ESP32	Літ.		Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата					
Розробив		Приймак Ю.П.							
Перевінив		Молчанов О.А.							
						Арк.		Аркушів	
					Схема роботи системи	КПІ ім. Ігоря Сікорського, ФІСПІМ КВ-23			
Н. контр.		Клятченко Я.М.							
Затв.		Романківніч В.О.							

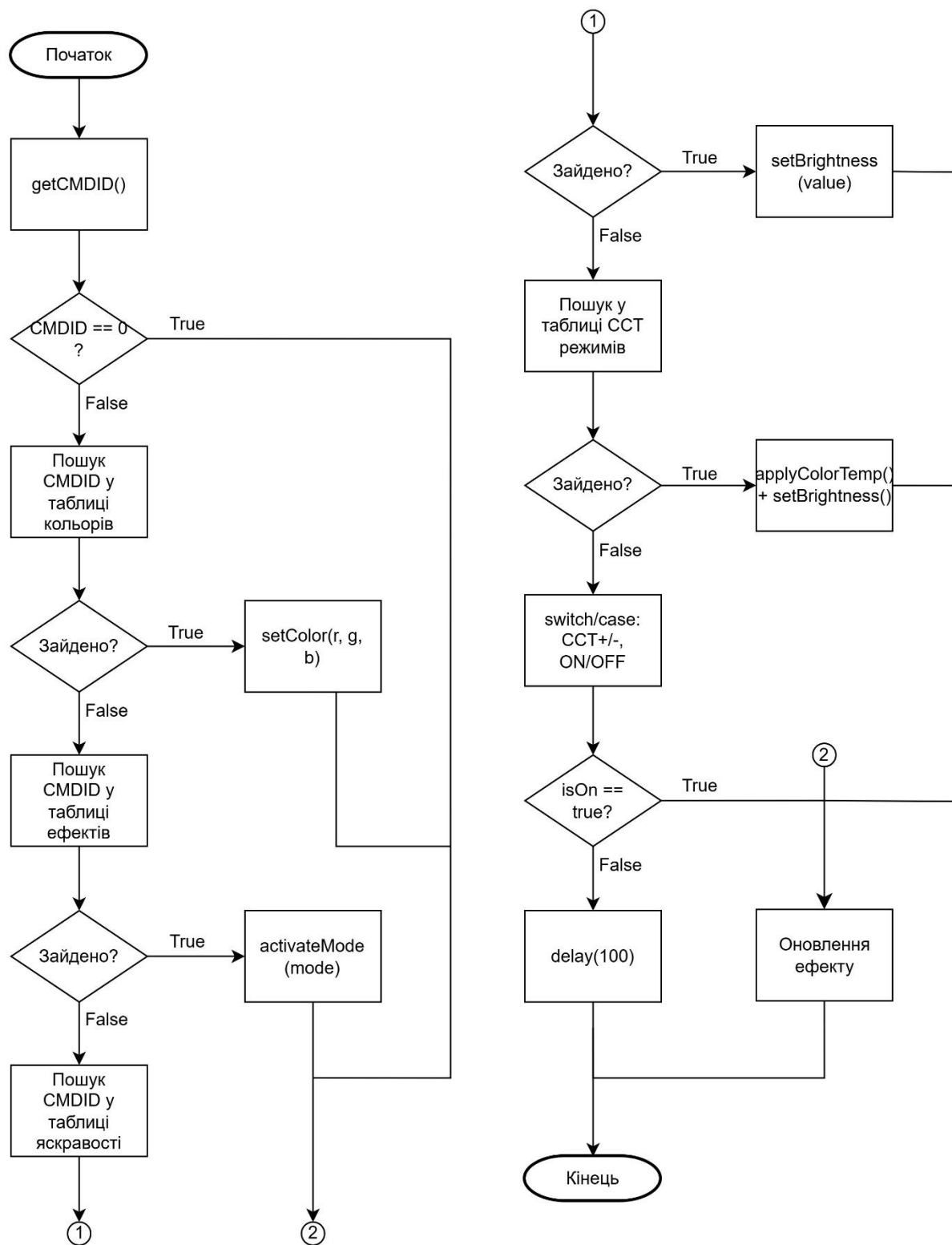
ДОДАТОК Б

Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32

Обробка голосових команд. Схема алгоритму
ІАЛЦ.468300.006 Д2

Аркушів 1

Київ 2026



Зм.	Арк.	№ докум.	Підпис	Дата	
Розробив		Приймак Ю.П.			
Перевірив		Молчанов О.А.			
Н. контр.		Клятчченко Я.М.			
Затв.		Романківнич В.О.			

ІАЛЦ.468300.006 Д2

Офлайн-система голосового
керування адресною LED
стрічкою на базі ESP32

Обробка голосових команд.
Схема алгоритму

Літ.			Маса		Масштаб	
Арк.			Аркушів			
КПІ ім. Ігоря Сікорського, ФПСІМ КВ-23						

ДОДАТОК В

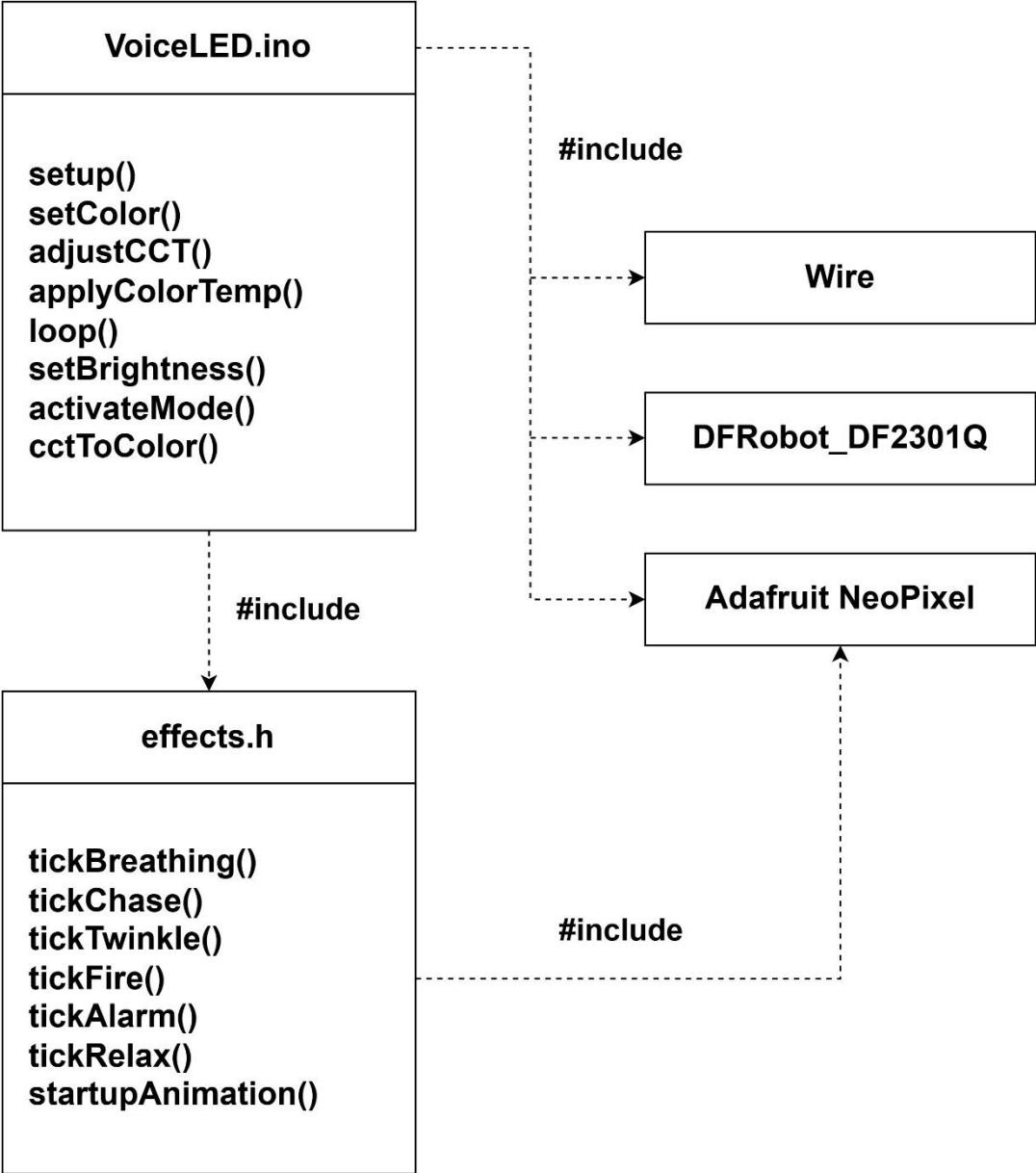
Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32

Взаємодія модулів. Схема структурна

ІАЛЩ.468300.007 ДЗ

Аркушів 1

Київ 2026



						ІАЛЦ.468300.007 ДЗ			
						Офлайн-система голосового керування адресною LED стрічкою на базі ESP32			
Зм.	Арк.	№ докум.	Підпис	Дата					
						Взаємодія модулів. Схема структурна			
Розробив		Приймак Ю.П.							
Перевірив		Молчанов О.А.				КПІ ім. Ігоря Сікорського, ФПСІМ КВ-23			
Н. контр.		Клятчєнко Я.М.							
Затв.		Романкевич В.О.							

ДОДАТОК Г

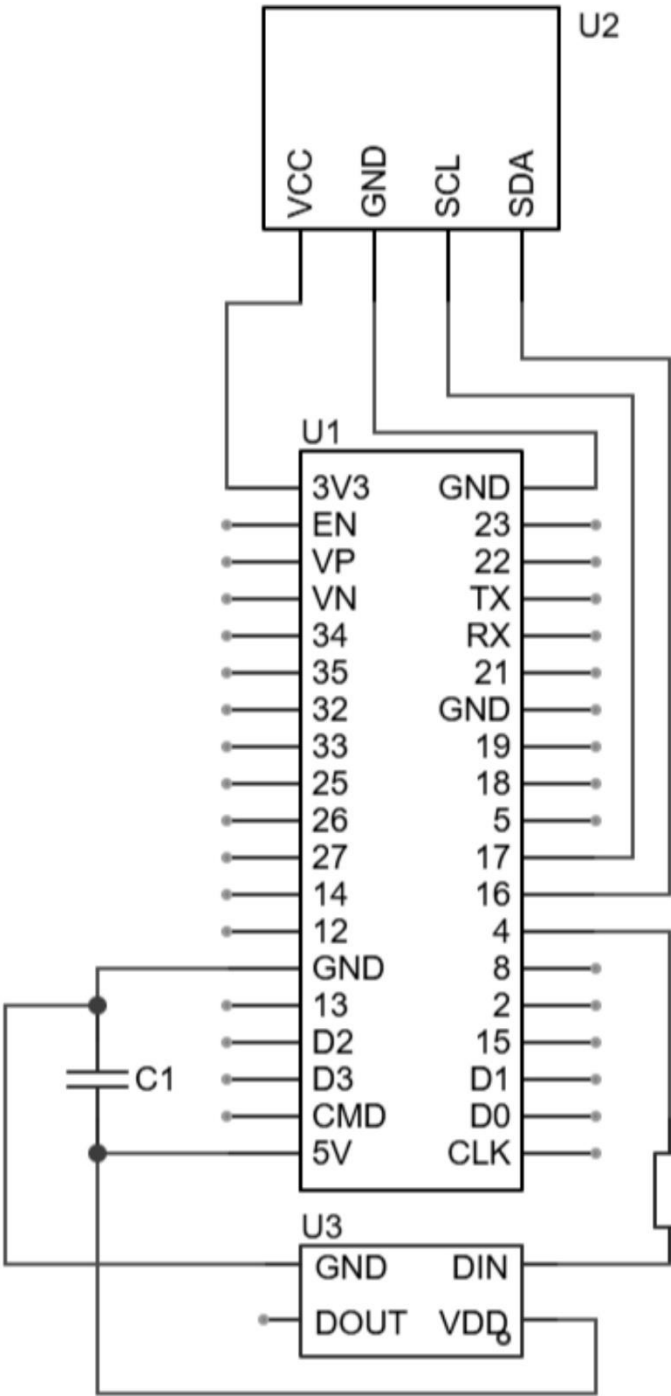
Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32

Схема електрична принципова

ІАЛЦ.468300.008 ЕЗ

Аркушів 2

Київ 2026



					ІАЛЦ.468300.008 ЕЗ				
					Офлайн-система голосового керування адресною LED стрічкою на базі ESP32	Літ.		Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата					
Розробив	Приймак Ю.П.								
Перевірив	Молчанов О.А.								
Н. контр.	Клятьченко Я.М.				Схема електрична принципова	Арк.		Аркушів	
Затв.	Романкевич В.О.					КПІ ім. Ігоря Сікорського, ФПСІМ КВ-23			

[illegible]

ДОДАТОК Д

Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32

Презентація

ІАЛЩ.468300.009 Д5

Аркушів 9

Київ 2026

Офлайн-система голосового керування адресною LED стрічкою на базі ESP32

Виконала: студентка IV курсу, групи
КВ-23

Приймак Юліанна Павлівна

Керівник: асистент каф. СПІСКС, д.ф.
Молчанов Олексій Андрійович

Вступ

Системи керування LED стрічками з голосовим вводом класифікують за:

- інтерфейсом керування (Wi-Fi, Bluetooth, Offline)
- типом LED стрічок (аналогові, адресні, білі)
- методом обробки команд (хмарні сервіси або на пристрої)
- ступенем відкритості

Онлайн-системи передають сигнал на зовнішні сервери — забезпечують високу точність, але залежать від інтернету та хмарних помічників.

Офлайн-системи виконують розпізнавання на вбудованому чипі (DTW, HMM, нейромережі) — повністю автономні, проте вільно доступних рішень на ринку практично немає.

Для обґрунтування розробки власної системи проведено порівняльний аналіз представників різних категорій контролерів.

Мета

Метою дипломного проекту є розробка офлайн-системи керування адресною LED стрічкою з використанням розпізнавання голосових команд на базі мікроконтролера ESP32 та модуля DFRobot Gravity SEN0539, що забезпечує повністю автономну роботу без підключення до інтернету чи зовнішніх сервісів та усуває основні недоліки існуючих комерційних рішень — залежність від інтернету, обмежену кількість голосових команд та відсутність підтримки адресних RGB стрічок.

Наявні рішення



Наявні рішення

Існуючі системи керування LED освітленням можна розділити на три категорії:

- WiFi контролери (Gledopto GL-C-310WL, GL-C-211WL) — широкий функціонал але повна залежність від інтернету для голосового керування через Alexa
- Bluetooth контролери (SUPERNIGHT) — незалежні від інтернету але керування через смартфон, лише аналогові стрічки
- Офлайн голосові контролери (Voiceemity DW SD/SI) — автономне голосове керування але лише білі стрічки, лише 8 команд

Жодне з рішень не поєднує офлайн голосове керування з підтримкою адресних RGB стрічок



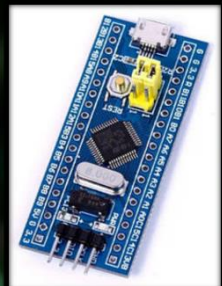
Обрані засоби



Arduino UNO — недостатньо: RAM лише 2 КБ — замало для ефектів та I2C буферів одночасно. Тактова частота 16 МГц не справляється з паралельною обробкою голосу та LED.

STM32 — надлишково: Висока продуктивність, але складне середовище розробки, потрібен окремий програматор та значно вища ціна.

ESP32 — оптимальний вибір: Достатня тактова частота та RAM для паралельної роботи. Пряма сумісність з DFRobot через 3.3V логіку — перетворювачі рівнів не потрібні.





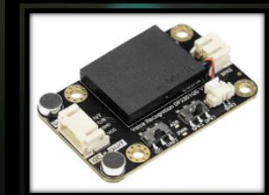
Обрані засоби



Elechouse V3 — не підходить: Залежний від голосу користувача, потребує тренування кожної команди. Лише 7 одночасно активних команд — може бути замало для керування освітленням.



LD3320 — не підходить: Незалежний від голосу, але низька точність в умовах шуму, відсутність якісної документації та потреба в додаткових компонентах.



DFRobot Gravity SEN0539 — оптимальний вибір: Незалежний від голосу, офлайн розпізнавання на нейронмережевому процесорі. 121 вбудована команда + 17 слотів для користувацьких. Детальна документація та офіційна бібліотека для Arduino IDE.



Обрані засоби



APA102 — не підходить: Висока частота оновлення через SPI, але два дроти даних займають більше GPIO пінів — ускладнює підключення з кількома периферійними пристроями. Вища ціна та нижча доступність.

SK6812 — не підходить: Схожа на WS2812B, але додатковий білий канал ускладнює керування без реальної необхідності для даного проекту.

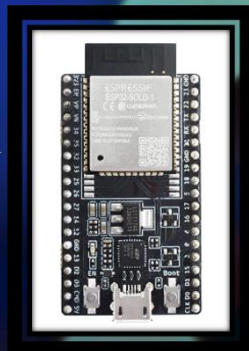
WS2812B — оптимальний вибір: Один дріт даних, проста інтеграція, широка доступність та низька ціна. Оптимальний баланс функціональності та простоти для навчального проекту.



Обрані засоби



I2C



1-Wire



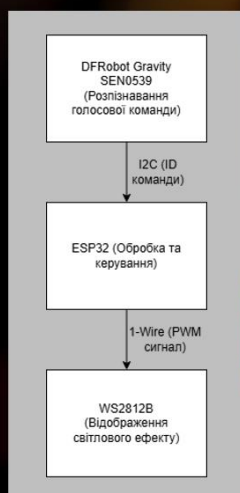
SEN0539 - ESP32 — інтерфейс I2C.

Сумісна 3.3В логіка — перетворювачі рівнів не потрібні. Живлення від вбудованого стабілізатора ESP32 (3В).

ESP32 - WS2812B — однодротовий інтерфейс 1-Wire.

Бібліотека Adafruit NeoPixel з офіційною підтримкою ESP32. Живлення стрічки від 5В через USB.

Принцип роботи системи



Після підключення живлення система ініціалізує всі компоненти та переходить у режим очікування.

SEN0539 постійно аналізує вхідний аудіосигнал у пошуку команди пробудження "Hello robot". Після її розпізнавання модуль починає аналізувати наступну команду користувача та передає її числовий ідентифікатор до ESP32 через I2C.

ESP32 зчитує ідентифікатор у головному циклі програми, знаходить відповідний світловий ефект у файлі `effects.h` та формує керуючий сигнал для стрічки.

WS2812B отримує сигнал через 1-Wire — кожен піксель зчитує свої 24 біти та передає решту наступному в ланцюжку.

Єдиний засіб взаємодії користувача з системою — голос.

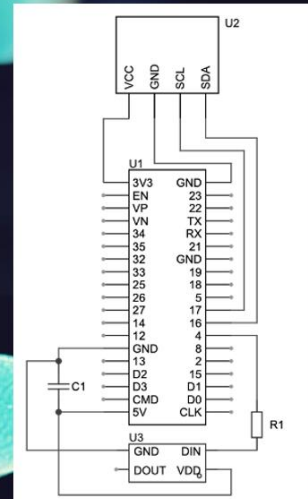
Підключення компонентів

Підключення голосового модуля до мікроконтролера

DFRobot SEN0539	ESP32	Призначення
VCC	3.3V	Живлення
GND	GND	Спільна земля
SDA	GPIO 16	Лінія даних I2C
SCL	GPIO 17	Тактова лінія I2C

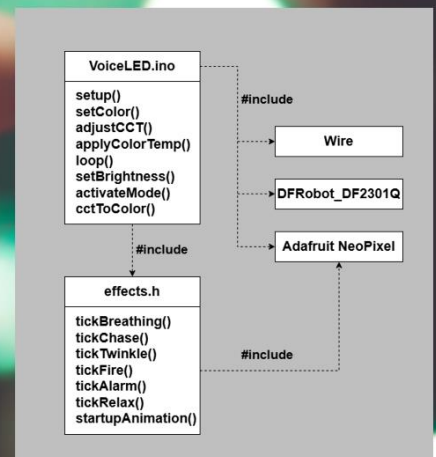
Підключення LED стрічки до мікроконтролера

WS2812B	ESP32	Призначення
VCC	5V	Живлення
GND	GND	Спільна земля
DIN	GPIO 4	Лінія даних 1-Wire



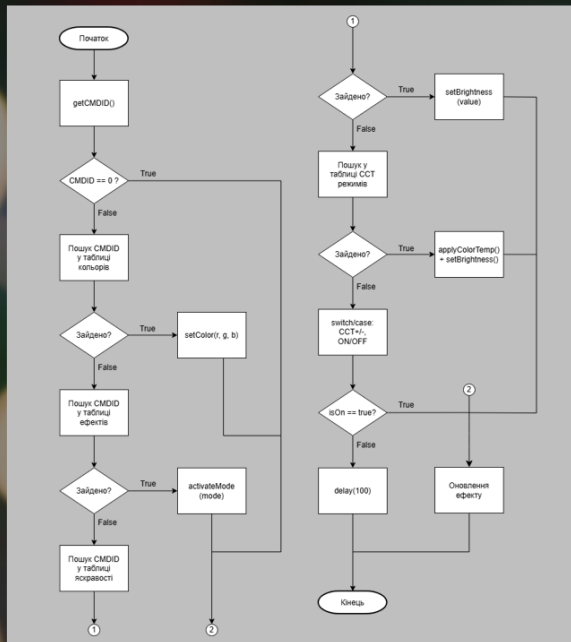
Програмна реалізація

Програмне забезпечення реалізовано мовою C++ в середовищі Arduino IDE. Воно складається з двох файлів — головного файлу керування VoiceLED.ino та файлу, що містить світлові ефекти effects.h. Така структура забезпечує модульність проекту та спрощує можливість його розширення у майбутньому. Додавання нового ефекту потребує змін лише у файлі effects.h без необхідності модифікації файлу з основною логікою програми. З бібліотек було використано: Adafruit NeoPixel для контролю LED стрічки, DFRobot_DF2301Q та Wire для голосового модуля.



Обробка ГОЛОСОВИХ КОМАНД

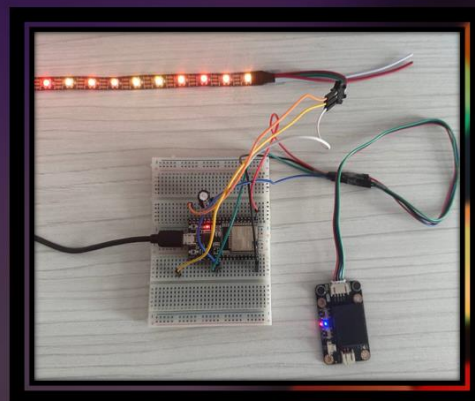
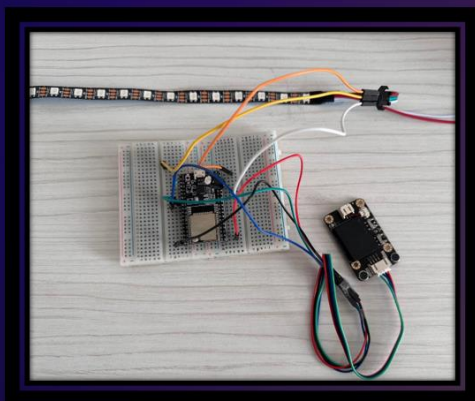
Після отримання ідентифікатора команди від модуля SEN0539 через I2C, ESP32 послідовно перевіряє чотири таблиці команд — кольори, ефекти, яскравість та ССТ режими. При знаходженні відповідності виконується відповідна дія. Команди ON/OFF та регулювання ССТ обробляються окремо через switch/case. Усі анімаційні ефекти оновлюються неблокуючим способом через millis() що забезпечує безперервне отримання нових команд.



Результати тестування

Тестування	Результат
<u>Точність розпізнавання</u>	<u>Середня точність 94% на вибірці з 10 команд</u>
<u>Дистанція</u>	<u>Стабільна робота до 3 м (100%) та до 5 м (80%)</u>
<u>Час відгуку</u>	<u>Обробка команди ESP32 в середньому 0.6 мс, загальний час відгуку ~600 мс</u>
<u>Стійкість до шуму</u>	<u>Точність 100% при інструментальній музиці, 80% при мовному фоні</u>
<u>Стабільність</u>	<u>60 хвилин безперервної роботи без жодних збоїв</u>

Зовнішній вигляд системи

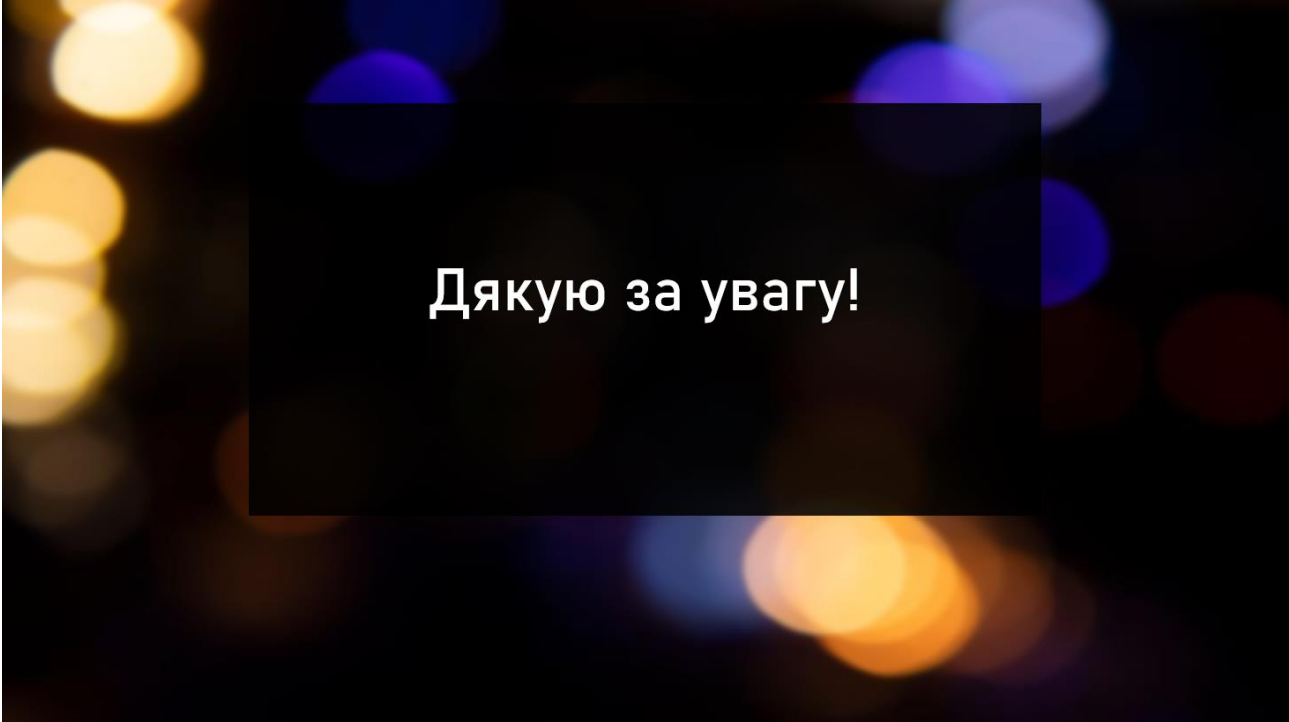


Висновки

У дипломному проєкті розроблено офлайн-систему голосового керування адресною LED-стрічкою. В ході роботи:

- Проаналізовано існуючі рішення — встановлено що жодне не поєднує офлайн керування з адресними RGB стрічками
- Обґрунтовано вибір компонентів — ESP32, DFRobot SEN0539, WS2812B
- Розроблено ПЗ з 27 голосовими командами та 6 динамічними ефектами
- Проведено тестування — точність 94%, дистанція до 3 м, стабільна робота 60 хв

Розроблена система є єдиною серед розглянутих що поєднує офлайн голосове керування, SI розпізнавання та адресні RGB стрічки з відкритою архітектурою для подальшого розширення.

The image features a dark background with out-of-focus light spots in warm yellow and cool blue tones, creating a bokeh effect. A solid black rectangular box is centered horizontally and vertically, containing the text "Дякую за увагу!".

Дякую за увагу!

ДОДАТОК Е

Офлайн-система голосового керування адресною LED-стрічкою на базі ESP32

Лістинг коду

ІАЛЦ.468300.010 Д6

Аркушів 6

Київ 2026

“VoiceLED.ino”

```
#include "DFRobot_DF2301Q.h"
#include <Adafruit_NeoPixel.h>
#include "effects.h"

#define NUM_LEDS 60
#define DATA_PIN 4
#define MAX_BRIGHTNESS 255
#define MIN_BRIGHTNESS 5
#define BRIGHTNESS_STEP 30
#define CCT_STEP 10

Adafruit_NeoPixel strip(NUM_LEDS, DATA_PIN, NEO_GRB + NEO_KHZ800);
DFRobot_DF2301Q_I2C asr;

bool isOn = true;
int currentBrightness = 80;
uint32_t currentColor = 0;
int colorTemp = 50;

enum Mode {
  MODE_STATIC,
  MODE_BREATHING,
  MODE_CHASE,
  MODE_TWINKLE,
  MODE_FIRE,
  MODE_ALARM,
  MODE_RELAX
};
Mode currentMode = MODE_STATIC;

unsigned long lastUpdate = 0;
int breathBrightness = 0;
bool breathDirection = true;
int chasePosition = 0;
bool alarmState = false;
int relaxBrightness = 5;
bool relaxDirection = true;

struct ColorCommand {
  uint8_t cmdID;
  uint8_t r, g, b;
};

const ColorCommand colorCommands[] = {
  {116, 255, 0, 0},
  {117, 255, 128, 0},
  {118, 255, 255, 0},
  {119, 0, 255, 0},
  {120, 0, 255, 255},
  {121, 0, 128, 255},
  {122, 153, 51, 255},
  {123, 255, 255, 255},
};
const int colorCount = sizeof(colorCommands) / sizeof(colorCommands[0]);

struct ModeCommand {
  uint8_t cmdID;
  int cctTemp;
  int brightness;
};

const ModeCommand modeCommands[] = {
  {113, 100, 200},
  {114, 20, 15},
  {128, 90, 150},
  {129, 10, 150},
};
const int modeCount = sizeof(modeCommands) / sizeof(modeCommands[0]);

struct BrightnessCommand {
  uint8_t cmdID;
```

```

int value;
bool isAbsolute;
};

const BrightnessCommand brightnessCommands[] = {
{105, BRIGHTNESS_STEP, false},
{106, -BRIGHTNESS_STEP, false},
{107, MAX_BRIGHTNESS, true},
{108, MIN_BRIGHTNESS, true},
};
const int brightnessCount = sizeof(brightnessCommands) / sizeof(brightnessCommands[0]);

struct EffectCommand {
uint8_t cmdID;
Mode mode;
int brightness;
};

const EffectCommand effectCommands[] = {
{5, MODE_TWINKLE, -1},
{6, MODE_FIRE, -1},
{7, MODE_BREATHING, -1},
{8, MODE_CHASE, -1},
{9, MODE_ALARM, 200},
{10, MODE_RELAX, -1},
};
const int effectCount = sizeof(effectCommands) / sizeof(effectCommands[0]);

struct EffectTick {
Mode mode;
void (*tickFn)();
unsigned long interval;
};

const EffectTick effectTicks[] = {
{MODE_BREATHING, tickBreathing, 15},
{MODE_CHASE, tickChase, 40},
{MODE_TWINKLE, tickTwinkle, 80},
{MODE_FIRE, tickFire, 120},
{MODE_ALARM, tickAlarm, 300},
{MODE_RELAX, tickRelax, 30},
};
const int tickCount = sizeof(effectTicks) / sizeof(effectTicks[0]);

uint32_t cctToColor(int temp) {
temp = constrain(temp, 0, 100);
uint8_t r = map(temp, 0, 100, 255, 180);
uint8_t g = map(temp, 0, 100, 180, 220);
uint8_t b = map(temp, 0, 100, 80, 255);
return strip.Color(r, g, b);
}

void setColor(uint8_t r, uint8_t g, uint8_t b) {
currentMode = MODE_STATIC;
currentColor = strip.Color(r, g, b);
if (isOn) { strip.fill(currentColor); strip.show(); }
}

void setBrightness(int brightness) {
currentBrightness = constrain(brightness, MIN_BRIGHTNESS, MAX_BRIGHTNESS);
strip.setBrightness(currentBrightness);
strip.show();
}

void applyColorTemp() {
currentMode = MODE_STATIC;
currentColor = cctToColor(colorTemp);
if (isOn) { strip.fill(currentColor); strip.show(); }
}

void adjustCCT(int delta) {
colorTemp = constrain(colorTemp + delta, 0, 100);
applyColorTemp();
}

```

```

void activateMode(Mode mode, int brightness = -1) {
    currentMode = mode;
    isOn = true;
    breathBrightness = 0;
    breathDirection = true;
    chasePosition = 0;
    alarmState = false;
    relaxBrightness = 5;
    relaxDirection = true;
    if (brightness >= 0) strip.setBrightness(brightness);
    else strip.setBrightness(currentBrightness);
}

void setup() {
    Serial.begin(115200);
    Serial.println("STARTING...\n");
    strip.begin();
    strip.setBrightness(currentBrightness);
    strip.clear();
    strip.show();
    strip.fill(strip.Color(0, 0, 255));
    strip.show();
    Serial.println("LEDS INITIALIZED\n");

    delay(2000);
    Wire.begin(16, 17);
    while (!(asr.begin())) {
        Serial.println("Communication with device failed");
        strip.fill(strip.Color(255, 0, 0));
        strip.show();
        delay(3000);
    }
    Serial.println("DFROBOT INITIALIZED\n");

    asr.setVolume(3);
    asr.setMuteMode(0);
    asr.setWakeTime(20);

    startupAnimation();
    Serial.println("READY FOR WORK");
}

void loop() {
    uint8_t CMDID = asr.getCMDID();

    if (CMDID != 0) {
        unsigned long responseStart = millis();

        for (int i = 0; i < colorCount; i++) {
            if (CMDID == colorCommands[i].cmdID) {
                setColor(colorCommands[i].r, colorCommands[i].g, colorCommands[i].b);
                Serial.print("Color: "); Serial.println(CMDID);
                CMDID = 0; break;
            }
        }

        for (int i = 0; i < effectCount; i++) {
            if (CMDID == effectCommands[i].cmdID) {
                activateMode(effectCommands[i].mode, effectCommands[i].brightness);
                Serial.print("Effect: "); Serial.println(CMDID);
                CMDID = 0; break;
            }
        }

        for (int i = 0; i < brightnessCount; i++) {
            if (CMDID == brightnessCommands[i].cmdID) {
                if (brightnessCommands[i].isAbsolute)
                    setBrightness(brightnessCommands[i].value);
                else
                    setBrightness(currentBrightness + brightnessCommands[i].value);
                Serial.print("Brightness: "); Serial.println(CMDID);
                CMDID = 0; break;
            }
        }
    }
}

```

```

}

for (int i = 0; i < modeCount; i++) {
    if (CMDID == modeCommands[i].cmdID) {
        colorTemp = modeCommands[i].cctTemp;
        applyColorTemp();
        setBrightness(modeCommands[i].brightness);
        Serial.print("Mode: "); Serial.println(CMDID);
        CMDID = 0; break;
    }
}

switch (CMDID) {
    case 109: adjustCCT(+CCT_STEP); Serial.println("CCT+"); break;
    case 110: adjustCCT(-CCT_STEP); Serial.println("CCT-"); break;
    case 111: adjustCCT(+100); Serial.println("CCT max"); break;
    case 112: adjustCCT(-100); Serial.println("CCT min"); break;
    case 103:
        isOn = true;
        currentMode = MODE_STATIC;
        strip.setBrightness(currentBrightness);
        strip.fill(currentColor);
        strip.show();
        Serial.println("ON");
        break;
    case 104:
        isOn = false;
        currentMode = MODE_STATIC;
        strip.clear();
        strip.show();
        Serial.println("OFF");
        break;
    default:
        if (CMDID != 0) {
            Serial.print("Unknown command: "); Serial.println(CMDID);
        }
        break;
}

Serial.print("Response time: ");
Serial.print(millis() - responseStart);
}

if (!isOn && currentMode != MODE_ALARM) {
    delay(100);
    return;
}

unsigned long now = millis();
for (int i = 0; i < tickCount; i++) {
    if (currentMode == effectTicks[i].mode) {
        if (now - lastUpdate > effectTicks[i].interval) {
            effectTicks[i].tickFn();
            lastUpdate = now;
        }
        break;
    }
}
}

```

“effects.h”

```
#ifndef EFFECTS_H
#define EFFECTS_H

#include <Adafruit_NeoPixel.h>
#include <math.h>

extern Adafruit_NeoPixel strip;
extern int currentBrightness;
extern uint32_t currentColor;
extern int relaxBrightness;
extern bool relaxDirection;
extern bool alarmState;
extern int breathBrightness;
extern bool breathDirection;
extern int chasePosition;

void tickBreathing() {
  strip.setBrightness(breathBrightness);
  strip.fill(currentColor);
  strip.show();
  if (breathDirection) {
    breathBrightness += 3;
    if (breathBrightness >= currentBrightness) breathDirection = false;
  } else {
    breathBrightness -= 3;
    if (breathBrightness <= 0) breathDirection = true;
  }
}

void tickChase() {
  strip.clear();
  strip.setPixelColor(chasePosition, currentColor);
  for (int t = 1; t <= 4; t++) {
    int pos = (chasePosition - t + strip.numPixels()) % strip.numPixels();
    uint8_t r = ((currentColor >> 16) & 0xFF) / (t * 2);
    uint8_t g = ((currentColor >> 8) & 0xFF) / (t * 2);
    uint8_t b = ((currentColor) & 0xFF) / (t * 2);
    strip.setPixelColor(pos, strip.Color(r, g, b));
  }
  strip.show();
  chasePosition = (chasePosition + 1) % strip.numPixels();
}

void tickTwinkle() {
  int pixel = random(strip.numPixels());
  if (random(2)) {
    strip.setPixelColor(pixel, currentColor);
  } else {
    strip.setPixelColor(pixel, 0);
  }
  strip.show();
}

void tickFire() {
  for (int i = 0; i < strip.numPixels(); i++) {
    int flicker = random(80, 180);
    int green = random(10, 50);
    strip.setPixelColor(i, strip.Color(flicker, green, 0));
  }
  strip.show();
}

void tickAlarm() {
  if (alarmState) {
    strip.fill(strip.Color(255, 0, 0));
  } else {
    strip.clear();
  }
  strip.show();
  alarmState = !alarmState;
}
```

```
void tickRelax() {
  strip.setBrightness(relaxBrightness);
  strip.fill(strip.Color(255, 147, 41));
  strip.show();
  if (relaxDirection) {
    relaxBrightness++;
    if (relaxBrightness >= 60) relaxDirection = false;
  } else {
    relaxBrightness--;
    if (relaxBrightness <= 5) relaxDirection = true;
  }
}

void startupAnimation() {
  for (int i = 0; i < strip.numPixels(); i++) {
    strip.setPixelColor(i, strip.Color(0, 100, 255));
    strip.setPixelColor((i + 1) % strip.numPixels(), strip.Color(0, 50, 128));
    strip.show();
    delay(40);
  }
  strip.clear();
  strip.show();
}

#endif
```