

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

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)

“ ___ ” червня 2025 р.

Дипломний проєкт

на здобуття ступеня бакалавра
за освітньо-професійною програмою

«Системне програмування та спеціалізовані комп'ютерні системи»

зі спеціальності

123 «Комп'ютерна інженерія»

на тему: Система налаштування механічних клавіатур на мікроконтролерах ESP32

Виконала: студентка IV курсу, групи КВ-11
(шифр групи)

_____ Нестерук Анастасія Олександрівна _____
(прізвище, ім'я, по батькові) (підпис)

Керівник ст. викладач каф. СПСКС, к.т.н. Коляда К.В. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з нормоконтролю, доц. каф. СПСКС, к.т.н., доц. Клятченко Я.М. _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент доц. каф. ОТ, к.т.н., доц., Марковський Олександр Петрович _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

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

Студентка _____
(підпис)

Київ – 2025 року

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

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)

«___» _____ 2025 р.

ЗАВДАННЯ

на дипломний проєкт студентки

Нестерук Анастасії Олександрівни

(прізвище, ім'я, по батькові)

1. Тема проєкту «Система налаштування механічних клавіатур на мікроконтролерах ESP32» _____

керівник проєкту ст. викл. каф. СПСКС, к.т.н. Коляда Костянтин Вячеславович,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від _____

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту див. Технічне завдання _____

4. Зміст пояснювальної записки _____

- 1) Аналіз наявних рішень та обґрунтування теми дипломного проєкту
- 2) Аналіз обраних технологій для розробки
- 3) Опис та тестування системи налаштування механічних клавіатур

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

- 1) Схема роботи системи налаштування механічних клавіатур на мікроконтролерах ESP32. Схема структурна
- 2) Зв'язок між модулями програмного коду клавіатури. Схема структурна
- 3) Алгоритм сканування матриці. Схема алгоритму
- 4) Алгоритм прошивки пристрою. Схема алгоритму

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., к. т. н., доцент каф. СП і СКС		

7. Дата видачі завдання 30 жовтня 2024 р

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Видача завдання на дипломне проектування	30.10.2024	
2.	Вивчення літератури за тематикою проекту	1.02.2025	
3.	Розроблення та узгодження технічного завдання	15.02.2025	
4.	Аналіз існуючих рішень	1.03.2025	
5.	Підготовка матеріалів першого розділу дипломного проекту	15.03.2025	
6.	Підготовка матеріалів другого розділу дипломного проекту	1.04.2025	
7.	Розробка програмної частини проекту	15.04.2025	
8.	Підготовка матеріалів третього розділу дипломного проекту	7.05.2025	
9.	Підготовка графічної частини дипломного проекту	15.05.2025	
10.	Попередній огляд матеріалів диплому на кафедрі	20.05.2025	
	Оформлення документації дипломного проекту	30.05.2025	

Студентка

(підпис)

Анастасія НЕСТЕРУК

(ініціали, прізвище)

Керівник проекту

(підпис)

Костянтин КОЛЯДА

(ініціали, прізвище)

АНОТАЦІЯ

Бакалаврський дипломний проєкт включає пояснювальну записку (54 с., 21 рис. 1 табл., 3 додатки, 20 слайдів презентації).

Об'єкт розробки – створення системи для налаштування клавіатур, побудованих на мікроконтролерах ESP32.

Система дозволяє: задавати параметри клавіатури, такі як кількість рядків і стовпців, призначення пінів, розкладку клавіш та запис макросів, а також проводити тестування роботи клавіатури, генерувати файл прошивки та автоматично завантажувати його на пристрій. У процесі розробки використані наступні технології: мови програмування C# та C++, платформа WPF, інструменти командного рядка Arduino CLI та esptool.

В ході розробки:

- проведено аналіз існуючих засобів налаштування клавіатур;
- Сформульовано основні вимоги до розроблюваної системи;
- Розроблено структуру системи налаштування клавіатур;
- Обрано необхідні інструменти та технології для реалізації системи;
- Створено зручний графічний інтерфейс розроблюваної системи;
- Розроблено основний функціонал системи для налаштування клавіатур побудованих на мікроконтролерах ESP32;
- Протестовано та відлагоджено роботу розробленої системи.

Розроблена система налаштувань дозволить зробити процес конфігурації клавіатури простішим та доступнішим, що дасть змогу користувачам зменшити час, витрачений на налаштування.

Ключові слова: клавіатура, прошивка, конфігурація, ESP32, C#, WPF, Arduino CLI, esptool.

ABSTRACT

Bachelor's degree project includes an explanatory note (54 p., 21 figures, 1 table, 3 appendices, 20 slides).

The object of development is to create a system for customizing keyboards built on ESP32 microcontrollers.

The system allows: set keyboard parameters, such as the number of rows and columns, pin assignment, key layout and macro recording, as well as test the keyboard operation, generate a firmware file and automatically upload it to the device. The following technologies were used in the development process: C# and C++ programming languages, WPF platform, Arduino CLI and esptool command line tools.

In the course of development:

- An analysis of existing keypad customization tools was conducted;
- The basic requirements for the developed system were formulated;
- The structure of the keyboard customization system is developed;
- The necessary tools and technologies for the system implementation were selected;
- A user–friendly graphical interface of the developed system was created;
- The basic functionality of the system for configuring keyboards built on ESP32 microcontrollers has been developed;
- Tested and debugged the developed system;

The developed configuration system will make the process of keyboard configuration easier and more accessible, which will allow users to reduce the time spent on configuration.

Keywords: keyboard, firmware, configuration, ESP32, C#, WPF, Arduino CLI, esptool.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.467200.002 ТЗ	Система налаштування механічних клавіатур на мікроконтролерах ESP32	4		
			Технічне завдання			
	A4	ІАЛЦ.467200.003 ТП	Система налаштування механічних клавіатур на мікроконтролерах ESP32	2		
			Відомість технічного проекту			
	A4	ІАЛЦ.467200.004 ПЗ	Система налаштування механічних клавіатур на мікроконтролерах ESP32	54		
			Пояснювальна записка			
	A3	ІАЛЦ.467200.005 Д1	Система налаштування механічних клавіатур на мікроконтролерах ESP32	1		
			Принцип роботи системи.			
			Схема структурна			

					ІАЛЦ.467200.001 ОА		
Змін.	Арк.	№ докум.	Підпис	Дата			
Розробив	Нестерук А.О.				Літ.	Аркуш	Аркушів
Перевірив	Коляда К.В.					1	2
Консульт.					Опис альбому і.м. Ігоря Сікорського, ФПМ КВ-11		
Н. контроль	Клятченко Я.М.						
Зав. каф.	Романкевич В. О.						

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

					<i>ІАЛЦ. 467200.002 ТЗ</i>			
Змін	Арк.	№ докум.	Підпис	Дата	Система налаштування механічних клавіатур на мікроконтролерах ESP32 Технічне завдання	Літ.	Аркуш	Аркушів
Розробив		Нестерук А.О.						
Перевірив		Коляда К.В.					1	4
Н. контроль		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФПМ КВ-11		
Затвердив		Романкевич В.О.						

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Система налаштування механічних клавіатур на мікроконтролерах ESP32».

Галузь застосування: розробка та налаштування периферійних пристроїв для професійних та спеціалізованих застосувань, оптимізація робочих процесів через індивідуальні рішення.

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 проєкту є розробка універсальної системи для налаштування клавіатур на базі ESP32, що надає користувачам можливість персоналізувати свої пристрої відповідно до індивідуальних потреб і вимог.

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

- Сумісність з Windows 10/11
- Інтуїтивний користувацький інтерфейс
- Підтримка взаємодії з мікроконтролером через USB CDC

					<i>ІАЛЦ. 467200.002 ТЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		2

- Налаштування клавіатури: підтримка до чотирьох шарів розкладки та можливість створення макросів.
- Збереження зроблених налаштувань.
- Прошивка мікроконтролера.

5.2. Вимоги до апаратного забезпечення

- Наявність USB-портів та Bluetooth з'єднання для підключення.
- Достатня кількість вбудованої пам'яті для збереження налаштувань та прошивок.
- Процесор: не нижче Intel Core i3 або AMD Ryzen 3
- Оперативна пам'ять: 8 Гб.
- Простір на диску: 64 Гб.
- Тип системи: 64-розрядна операційна система.

5.3. Вимоги до програмного та апаратного забезпечення користувача

- Операційна система Windows 10/11
- Мікроконтролер сімейства ESP32 з Bluetooth
- Наявність можливості Bluetooth з'єднання
- Наявність драйверів для обраного мікроконтролера
- Оперативна пам'ять: 4 Гб
- Вільне місце на диску: 10 Гб

					<i>ІАЛЦ. 467200.002 ТЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		3

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Видача завдання на дипломне проєктування	30.10.2024
2.	Вивчення літератури за тематикою проекту	1.02.2025
3.	Розроблення та узгодження технічного завдання	15.02.2025
4.	Аналіз існуючих рішень	1.03.2025
5.	Підготовка матеріалів першого розділу дипломного проекту	15.03.2025
6.	Підготовка матеріалів другого розділу дипломного проекту	1.04.2025
7.	Розробка програмної частини проєкту	15.04.2025
8.	Підготовка матеріалів третього розділу дипломного проекту	7.05.2025
9.	Підготовка графічної частини дипломного проекту	15.05.2025
10.	Попередній огляд матеріалів диплому на кафедрі	20.05.2025
11.	Оформлення документації дипломного проекту	30.05.2025

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	3
ВСТУП	4
1. АНАЛІЗ НАЯВНИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ	6
1.1 Аналіз існуючих систем для налаштування та прошивки механічних клавіатур.....	6
1.1.1 Набір інструментів Quantum Mechanical Keyboard.....	6
1.1.2 Додаткова прошивка для QMK VIA.....	12
1.1.3 кросплатформний графічний інтерфейс vial	15
1.1.4 Проєкт Zephyr Mechanical Keyboard Firmware	17
1.1.5 Прошивка КМК на основі Python.....	20
1.2 Переваги та недоліки існуючих рішень	22
1.3 Обґрунтування теми дипломного проєкту.....	23
2. АНАЛІЗ ОБРАНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ.....	25
2.1 Мова програмування C#	25
2.2 Графічний фреймворк Windows Presentation Foundation	26
2.3 Середовище розробки Visual Studio	28
2.4 Система контролю версій Git та платформа GitHub.....	30
2.5 Додаткові інструменти	31
2.5.1 Arduino CLI	31
2.5.2 Estptool.....	32
2.6 Підтримувані мікроконтролери	33
3. ОПИС ТА ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ НАЛАШТУВАНЬ МЕХАНІЧНИХ КЛАВІАТУР	36
3.1 Структура системи	36
3.2 Структура програмного коду клавіатури	37

					<i>ІАЛЦ. 467200.004 ПЗ</i>			
Змін	Арк.	№ докум.	Підпис	Дата	Система налаштування механічних клавіатур на мікроконтролерах ESP32 Пояснювальна записка	Літ.	Аркуш	Аркушів
Розробив		Нестерук А.О.						
Перевірив		Коляда К.В.					1	54
Н. контроль		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФПМ КВ-11		
Затвердив		Романкевич В.О.						

3.3 Тестування системи.....	40
3.4 Можливості подальшого вдосконалення системи	50
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	53

Додатки

Додаток А. Копії графічного матеріалу.

- 1) ІАЛЦ.467200.005 Д1. Схема роботи системи налаштування механічних клавіатур на мікроконтролерах ESP32. Схема структурна
- 2) ІАЛЦ.467200.006 Д2. Зв'язок між модулями програмного коду клавіатури. Схема структурна
- 3) ІАЛЦ.467200.007 Д3. Алгоритм сканування матриці. Схема алгоритму.
- 4) ІАЛЦ.467200.008 Д4. Алгоритм прошивки пристрою. Схема алгоритму

Додаток Б. Фрагменти програмного коду

Додаток В. Презентація

					<i>ІАЛЦ. 467200.004 ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ARM – сімейство 32- та 64-бітних мікропроцесорів з архітектурою RISC, розроблене компанією Arm Holdings

AVR – сімейство восьмибітових мікроконтролерів розроблених, компанією Atmel

BLE – бездротова технологія Bluetooth з низьким рівнем енергоспоживанням

CLI (Command Line Interface) – спосіб взаємодії між людиною і комп'ютером шляхом надсилання комп'ютеру команд, що являють собою послідовність символів.

DIY (do it yourself) – вид діяльності, за якого людина самостійно виготовляє для власного використання ті речі, предмети та вироби, які зазвичай виготовляються промисловістю або майстрами-професіоналами

ESP32 – сімейство мікроконтролерів розроблених компанією Espressif Systems

IDE (Integrated Development Environment) – комплекс програмних засобів, який використовують програмісти для розроблення програмного забезпечення

MSYS2 (Minimal SYStem 2) – програмна платформа для Windows, яка надає Unix-подібне середовище для розробки та компіляції програмного забезпечення

QMK (Quantum Mechanical Keyboard) – набір інструментів з відкритим вихідним кодом для розробки та кастомізації прошивок механічних клавіатур на основі AVR та ARM мікроконтролерів

ZMK (Zephyr Mechanical Keyboard) – прошивка з відкритим вихідним кодом для механічних клавіатур, створена на базі операційної системи реального часу Zephyr RTOS

Змін.	Арк.	№ докум.	Підпис	Дата

ВСТУП

Сучасний світ розвивається стрімкими темпами, а комп'ютерні технології стають невід'ємною частиною нашого повсякдення, а комп'ютерні пристрої відіграють ключову роль у різних сферах життя, забезпечуючи зручну й ефективну взаємодію з цифровим середовищем. Особливе місце серед них посідають засоби введення інформації, зокрема клавіатури, які залишаються основним інструментом для роботи з комп'ютерами навіть у наші дні.

Останніми роками механічні клавіатури здобули значну популярність завдяки своїй надійності, тактильному комфорту та широким можливостям персоналізації. Вони стали улюбленим вибором тих, хто цінує не лише зручність, а й можливість адаптувати пристрій до власних потреб. Сучасний ринок пропонує широкий асортимент клавіатур із розширеним функціоналом, однак такі рішення часто мають високу вартість, а бюджетні варіанти, у свою чергу, поступаються у гнучкості налаштувань, що змушує користувачів обирати між ціною та можливостями.

Багато ентузіастів знаходять вихід у самостійному створенні клавіатур із окремих компонентів, що дозволяє досягти оптимального балансу між вартістю та функціоналом. У таких проєктах значну роль відіграє програмне забезпечення, яке забезпечує конфігурацію пристрою. Проте більшість доступних інструментів для налаштування кастомних клавіатур потребують багато часу в освоєнні, спеціальних знань у програмуванні та розуміння різних технічних аспектів, що робить їх недоступними для широкого кола користувачів.

Ця проблема підкреслює потребу в розробці інтуїтивно зрозумілих і водночас потужних систем налаштування, які б спростили процес персоналізації клавіатур і зробили його доступним навіть для тих, хто не має глибокого технічного досвіду. Такі інструменти можуть усунути бар'єри, пов'язані зі складністю програмування, і відкрити нові можливості для користувачів, які прагнуть адаптувати пристрої до своїх потреб без зайвих зусиль.

					<i>ІАЛЦ. 467200.004 ПЗ</i>	Арк.
						4
Змін.	Арк.	№ докум.	Підпис	Дата		

Метою цього проєкту є створення системи налаштування механічних клавіатур із дружнім інтерфейсом, яка дозволить легко змінювати розкладки, створювати макроси та керувати основними функціями пристрою. Очікується, що таке рішення підвищить зручність персоналізації, зробить її доступною для ширшої аудиторії та покращить загальний досвід використання клавіатур, задовольняючи потреби як початківців, так і досвідчених користувачів.

					<i>ІАЛЦ. 467200.004 ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		5

1. АНАЛІЗ НАЯВНИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ

Перший розділ проєкту присвячений огляду існуючих систем налаштування клавіатур, у ньому розглянуті найпопулярніші рішення, що пропонують широкий функціонал для налаштування. Проведено аналіз їхніх можливостей, а також розглянуто переваги та недоліки кожного підходу, що дозволяє оцінити сильні та слабкі сторони сучасних рішень і визначає доцільність розробки нової системи, яка пропонує унікальні функції або покращену зручність використання.

1.1 Аналіз існуючих систем для налаштування та прошивки механічних клавіатур

1.1.1 Набір інструментів Quantum Mechanical Keyboard

Quantum Mechanical Keyboard – це набір інструментів з відкритим вихідним кодом, який дозволяє користувачам налаштовувати прошивку своєї механічної клавіатури. QMK базується на старій прошивці TMK Keyboard Firmware, розширює її можливості, даючи змогу користувачам налаштовувати роботу кожної клавіші, додавати складні макроси та інше. QMK підтримує безліч різних моделей клавіатур, як комерційних так і користувацьких рішень. Завдяки своїй гнучкості та широким можливостям налаштування, QMK став популярним вибором серед ентузіастів.

Опис деяких функцій QMK:

- Розширені клавіші: Забезпечується підтримка зміни шарів з використанням подвійної дії – одна фізична клавіша може виконувати функцію перемикання шару та водночас надсилати сканкод. Передбачено як класичні методи перемикання, так і комбіновані.
- Bluetooth: підтримка Bluetooth для клавіатури (Для Bluetooth 2.1 QMK підтримує модулі RN-42. Для більш сучасних протоколів BLE наразі безпосередньо підтримується лише Adafruit Bluefruit SPI Friend).

					<i>ІАЛЦ. 467200.004 ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		6

- Розкладки (Layouts): Фізичне розташування клавiш (layout) абстрагується від логічної розкладки (keymap), що дозволяє використовувати один набір сканкодів із кількома різними фізичними конфігураціями клавіатури.
- Апаратні макроси: Реалізується запис і відтворення послідовностей клавiш без залежності від операційної системи користувача.
- Динамічні макроси: Підтримується створення тимчасових макросів без необхідності перепрошивки пристрою.
- Leader Key: Використовується спеціальна клавiша-ініціатор, після якої активується послідовність дій або макрос.
- Tap Dance: Одна клавiша здатна виконувати різні функції залежно від кількості послідовних натискань.
- Key Lock: Передбачено можливість фіксації клавiші в натиснутому стані для тривалого утримання.
- Аудіо: Здійснюється виведення звукових сигналів при натисканні клавiш, перемиканні шарів або інших діях. Підтримується MIDI та музичний режим.
- Auto Shift: При короткому натисканні надсилається звичайний символ, при тривалому – символ із модифікатором Shift.
- Bootmagic: Підтримується зміна конфігурацій клавіатури через спеціальні комбінації клавiш під час завантаження, наприклад, перестановка функцій CapsLock та Control.
- Термінал (CLI): Реалізовано інтерфейс командного рядка для керування внутрішніми параметрами клавіатури.
- OLED та LCD-дисплеї: Здійснюється підтримка підключення дисплеїв (HD44780, OLED) для виведення стану системи або кастомної інформації.
- Підсвічування: Реалізується підтримка як однотонного LED-підсвічування, так і індивідуального RGB-підсвічування клавiш через матрицю.

- **Velocikey:** Дозволяється динамічна зміна швидкості анімації підсвічування відповідно до швидкості друку (WPM).
- **Керування мишею (Mouse Keys):** Дозволяється керування курсором безпосередньо з клавіатури.
- **Зовнішні координатні пристрої:** Підтримується підключення пристроїв введення типу трекпоінт або PS/2 миші.
- **Енкодери:** Реалізовано підтримку поворотних енкодерів з можливістю налаштування дій за обертанням або натисканням [1].

QMK працює на більшості USB-сумісних мікроконтролерів архітектур AVR та ARM, за умови наявності достатнього обсягу флеш-пам'яті. Для AVR – чіпів зазвичай потрібно щонайменше 32 КБ, проте при значному відключенні функцій прошивка може вміститися навіть у 16 КБ. У випадку ARM-контролерів рекомендується мати щонайменше 64 КБ флеш-пам'яті. Сумісність окремих функцій QMK залежить від конкретного мікроконтролера, а деякі функції можуть бути частково або повністю несумісні з певними моделями [1].

Загалом існує декілька інструментів QMK, одні з них це – QMK Configurator, QMK Toolbox і QMK MSYS.

QMK Configurator – це вебінструмент, який дозволяє налаштовувати розкладки клавіатур та генерувати прошивки на їх основі. Однак він має обмеження: не підтримується генерація прошивок для клавіатур із контролерами, що відрізняються від тих, для яких вони були розроблені, наприклад, для контролера RP2040 на платі, призначеній для Pro Micro. У таких випадках необхідно застосовувати інструменти командного рядка [1].

Щоб розпочати налаштування, треба обрати необхідний пристрій у списку «KEYBOARD», після цього з'явиться короткий опис, який може допомогти в разі виникнення проблем.

Наступним кроком є вибір найбільш підходящої розкладки клавіатури, яку можна налаштувати через інтерактивну карту, на якій користувач може призначати функції клавіш кількома способами: перетягуючи їх з карти кодів,

вибираючи коди для клавіш прямо на макеті, або навіть натискаючи фізичні клавіші для автоматичного призначення функцій. Після налаштувань можна експортувати готову розкладку у файл, щоб далі продовжити роботу.

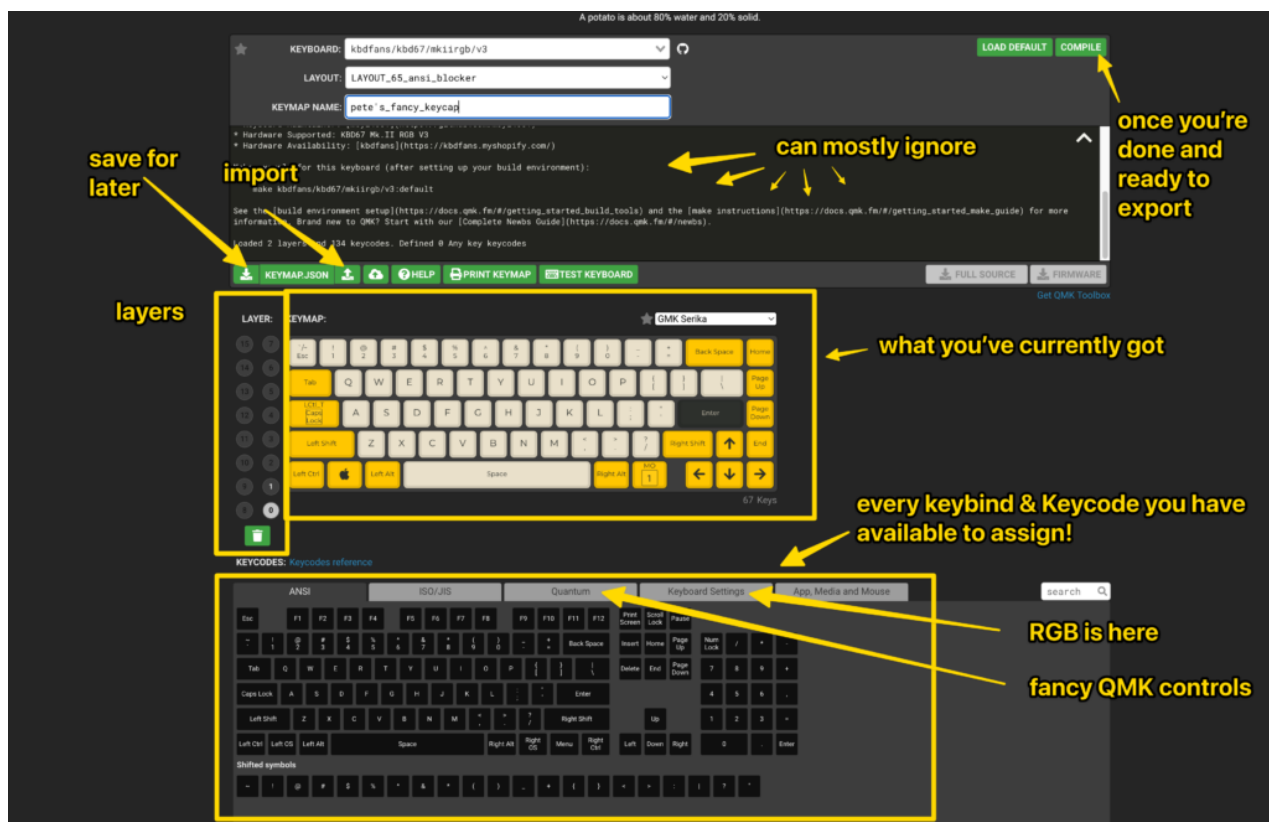


Рисунок 1 – Зовнішній вигляд QMK Configurator

Для її компіляції достатньо натиснути кнопку «Compile», після чого з'явиться можливість завантажити мікропрограму у форматах .bin або .hex.

QMK Configurator також дозволяє працювати з шарами клавіатури, додаючи гнучкості у програмуванні та налаштуванні. Підтримка розширених кодів клавіш, таких як підсвічування, Mod-Tap або одноразові клавіші, розширює функціональність і дозволяє точніше налаштувати клавіатуру під особисті потреби. Процес налаштувань супроводжується інтуїтивно зрозумілим інтерфейсом, що дозволяє без проблем змінювати функції клавіш, налаштовувати рівень підсвічування та інші параметри, наприклад, налаштування RGB підсвітки чи функцій для миші.

QMK Toolbox – це програмне забезпечення для «перепрошивки» мікропрограми клавіатури.

Змін.	Арк.	№ докум.	Підпис	Дата

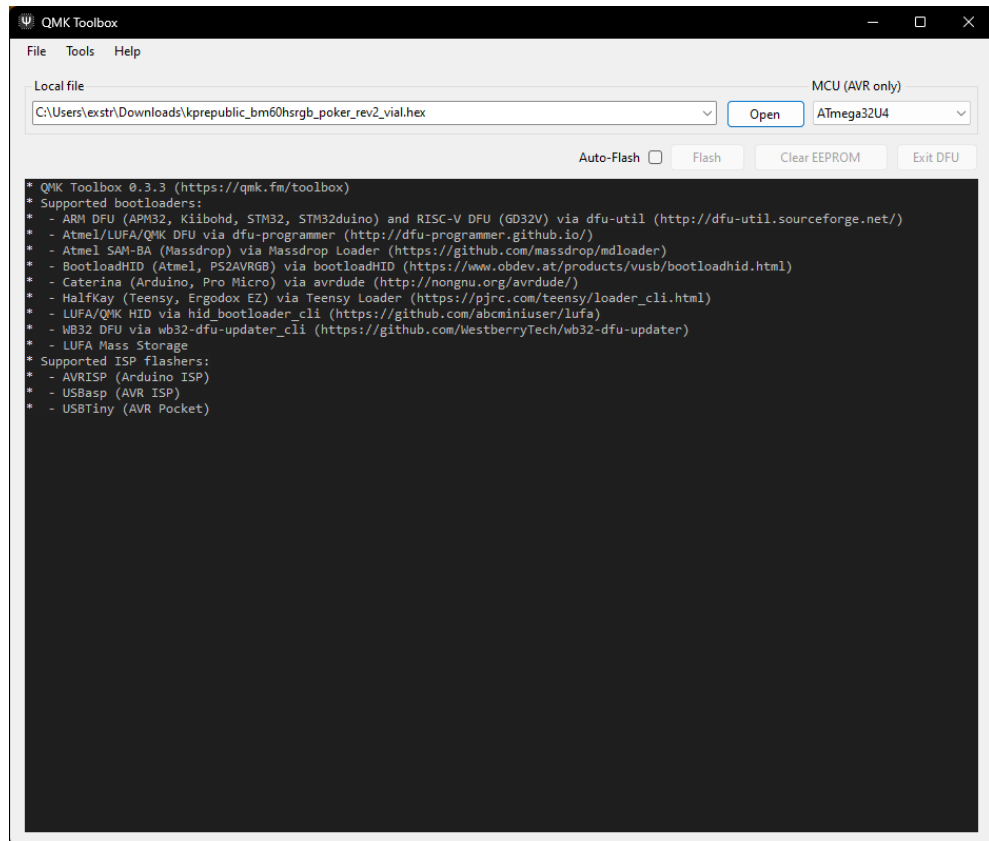


Рисунок 2 – Зовнішній вигляд QMK Toolbox

Процес прошивки клавіатури є важливим етапом налаштування і QMK Toolbox є одним з найзручніших способів це зробити. Це безкоштовне програмне забезпечення, доступне для операційних систем Windows та Mac, яке дозволяє зручно та швидко завантажувати мікропрограму на клавіатуру.

Після першого запуску QMK Toolbox на Windows програма автоматично пропонує встановити необхідні драйвери, або можна зробити це вручну через контекстне меню програми. Для користувачів Mac цей етап не потрібен.

Наступним кроком є завантаження файлу прошивки, це може бути файл формату .bin або .hex, який можна вибрати у відповідному розділі програми.

Перед початком прошивки слід переконатися, що вибрано правильний мікроконтролер (MCU), відповідний моделі клавіатури. У більшості випадків Toolbox автоматично визначає необхідний контролер, однак у разі потреби його можна обрати вручну, інформацію про потрібний MCU можна знайти у файлі rules.mk у репозиторії клавіатури на GitHub.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ. 467200.004 ПЗ

Арк.

10

Для прошивки необхідно перевести клавіатуру в режим завантажувача, після чого має відобразити повідомлення про успішне підключення пристрою. Якщо всі попередні етапи були виконано правильно, потрібно натиснути кнопку «Flash», щоб розпочати процес прошивки. Процес триває кілька секунд, і після її завершення з'явиться повідомлення про відключення пристрою, що свідчить про успішне оновлення мікропрограми.

QMK Toolbox значно спрощує процес прошивки клавіатури, дозволяючи швидко й ефективно оновлювати її програмне забезпечення та адаптувати пристрій під конкретні потреби користувача.

QMK MSYS – це спеціалізоване програмне середовище, засноване на дистрибутиві MSYS2, призначене для розробки, компіляції та завантаження прошивок QMK в операційній системі Windows. Воно надає зручну командну оболонку з підтримкою POSIX-інтерфейсу, попередньо налаштованими шляхами та інтеграцією з усіма необхідними інструментами, включно з git, make, компіляторами для архітектур AVR і ARM, а також утилітами прошивки [2].

Першим кроком у робочому процесі є ініціалізація системи. У терміналі слід ввести команду `qmk setup`, яка завантажить офіційний репозиторій прошивки з GitHub і виконає встановлення всіх залежностей.

Подальша робота здійснюється всередині директорії `qmk_firmware`. Тут зберігаються всі основні компоненти прошивки: структура каталогів клавіатур, файли конфігурації, вихідні коди логіки роботи клавіш і макросів. Для кожної клавіатури створюється власна директорія, всередині якої містяться технічні параметри схеми, налаштування підключення та опис поведінки клавіш.

Основні параметри вказуються у файлі `rules.mk`, де задаються модель мікроконтролера, тактова частота, обсяг завантажувача та інші опції компіляції. У файлі `config.h` вказуються розміри матриці клавіатури, номери пінів введення-виведення, налаштування затримок, режимів зв'язку та інше. Поведінка клавіш і шарів визначається в директорії `keymaps`, де кожна піддиректорія відповідає окремій розкладці клавіатури. Ці розкладки описуються за допомогою файлів

мовою C, у яких задаються дії клавіш, комбінації, макроси та логіка перемикання між шарами.

Після завершення всіх налаштувань виконується збірка прошивки. Для цього використовується команда: `qmk compile -kb <шлях_до_клавіатури> -km <ім'я_розкладки>`, де шлях до клавіатури відповідає структурі директорій всередині `keyboards/`, а ім'я розкладки збігається з назвою директорії всередині `keymaps/`. У результаті компіляції створюється бінарний файл у форматі `.hex` або `.bin`, готовий для прошивки пристрою.

Для прошивки мікроконтролера після компіляції можна використовувати вбудовану команду «`qmk flash`», без необхідності використовувати сторонні програми. Попередньо клавіатуру необхідно перевести в режим завантажувача, що залежить від типу використовуваної плати (наприклад, на Pro Micro із завантажувачем Caterina досить двічі натиснути кнопку Reset).

Команда прошивки має такий вигляд: `qmk flash -kb <шлях_до_клавіатури> -km <ім'я_розкладки>`.

Альтернативно, можна об'єднати компіляцію і прошивку однією командою: `qmk compile -kb <шлях_до_клавіатури> -km <ім'я_розкладки>:flash`.

Середовище QMK MSYS автоматично підбирає потрібний завантажувач і параметри, виходячи з типу мікроконтролера, зазначеного у файлі `rules.mk`.

Як наслідок, уся робота – від завантаження вихідних кодів і налаштування оточення до компіляції та завантаження прошивки в пристрій – може бути виконана в межах одного середовища, без необхідності залишати інтерфейс QMK MSYS. Це робить процес максимально автоматизованим, відтворюваним і доступним навіть для розробників-початківців.

1.1.2 Додаткова прошивка для QMK – VIA

VIA – додаткова прошивка для QMK, яка дозволяє змінювати налаштування клавіатури без перепрошивки. Всі налаштування зберігаються

безпосередньо на клавіатурі, тому вони залишаються активними навіть після відключення пристрою чи підключення до іншого комп'ютера [3].

Для використання VIA існує два варіанти: вебзастосунок і десктопна програма.

- Вебзастосунок є зручним варіантом, оскільки не потребує встановлення – всі оновлення та нові функції доступні автоматично, без дій з боку користувача. Проте він працює лише в браузерях на базі Chrome, які підтримують WebHID.
- Десктопний застосунок VIA Configurator app – це окрема програма, яка насправді є обгорткою над вебзастосунком, тому для її роботи також потрібне підключення до Інтернету. Однак вона може бути трохи надійнішою у виявленні клавіатури на деяких системах. Програма доступна для Windows, macOS та Linux.

Перевагою VIA є постійна пам'ять на клавіатурі, що дозволяє використовувати збережені налаштування навіть на інших пристроях. Це зручно для людей, які часто перемикаються між різними комп'ютерами.

VIA значно спрощує процес налаштування, оскільки не потребує технічних знань. Завдяки конфігуратору можна легко змінювати функції клавіш, слої та підсвітку, навіть без досвіду програмування. Всі зміни застосовуються миттєво, а також можна зберігати різні профілі для різних умов використання – роботи, ігор чи програмування.

Змін.	Арк.	№ докум.	Підпис	Дата



Рисунок 3 – Зовнішній вигляд via configurator

VIA працює шляхом обміну даними з вбудованим програмним забезпеченням, яке працює на пристрої, і надсилає йому команди через USB. Увімкнення функції VIA у QMK дає змогу спілкуватися з Конфігуратором VIA та зберігати розкладки клавіш та інші налаштування [3].

Якщо клавіатура не має попередньо встановленої підтримки VIA, користувач може налаштувати її самостійно в середовищі QMK. Для цього необхідно:

- створити окрему розкладку клавіш у каталозі keymaps/via;
- додати до неї файл rules.mk із рядком `VIA_ENABLE = yes`;
- створити файл keymap.c або keymap.json з відповідним макросом розкладки;
- у файлі info.json вказати унікальні значення `vid` і `pid`, які дозволяють VIA Configurator розпізнати пристрій [3].

Після цього прошивка компілюється з розкладкою via, що дозволяє змінювати клавіатуру динамічно – без потреби подальших перепрошивок.

Змін.	Арк.	№ докум.	Підпис	Дата

Хоча VIA спрощує налаштування, деякі функції QMK, наприклад, Tap Dance, не підтримуються в VIA, і для повного доступу до можливостей QMK може знадобитися перепрошивка.

Загалом, VIA є зручним і потужним інструментом для тих, хто хоче швидко налаштувати свою клавіатуру без перепрошивки кожного разу, з підтримкою профілів і постійною пам'яттю для різних сценаріїв використання.

1.1.3 Кросплатформний графічний інтерфейс – vial

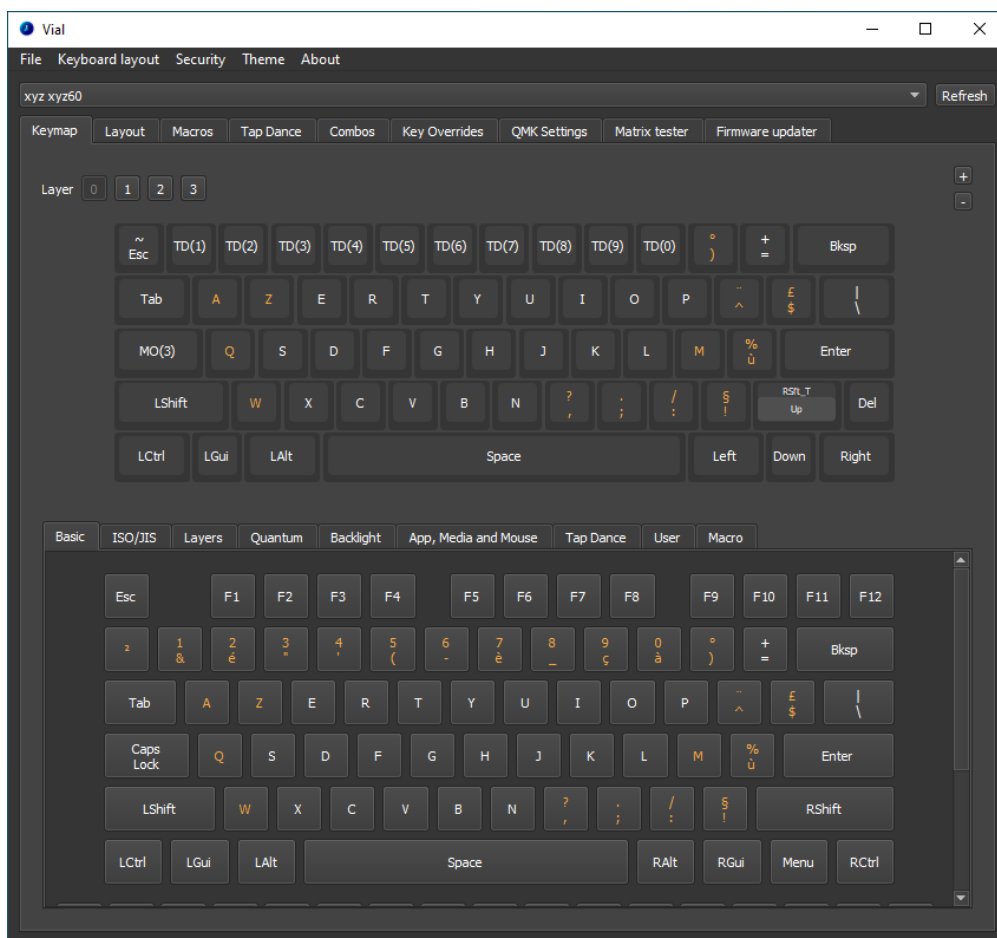


Рисунок 4 – Зовнішній вигляд vial configurator

Vial – це кросплатформний графічний інтерфейс з відкритим вихідним кодом, що є форком QMK і дозволяє налаштовувати клавіатури в режимі реального часу. Завдяки простому та інтуїтивно зрозумілому інтерфейсу, програма дозволяє змінювати різні параметри без необхідності програмування. Як і у випадку з VIA, основною перевагою Vial є миттєве застосування змін: усі дії, виконані в програмі, відразу відображаються на клавіатурі. Це включає зміну

Змін.	Арк.	№ докум.	Підпис	Дата

призначення клавіш, створення макросів, додавання шарів для зміни функціональності залежно від активного режиму. Такі зміни застосовуються у реальному часі, що значно спрощує процес налаштування [4].

На відміну від VIA, Vial пропонує ширший набір функцій. Він підтримує такі можливості, як Tap Dance, Combos, Key Overrides, налаштування до шістнадцяти шарів, розширену підтримку макросів, використання енкодерів, редагування ключових кодів та використання raw HID у режимі реального часу. Також передбачена підтримка RGB-підсвічування, що робить програму універсальним інструментом для повного контролю над поведінкою клавіатури [4].

Для того щоб налаштувати Vial, необхідно виконати наступні кроки:

- клонувати репозиторій Vial-QMK у незалежний каталог;
- у середовищі QMK MSYS виконати команду `make git-submodule` для ініціалізації всіх необхідних підмодулів;
- створити розкладку `vial`, скопіювавши вміст каталогу `keymaps/default` у `keymaps/vial`;
- додати до файлу `rules.mk` директиви `VIA_ENABLE = yes` та `VIAL_ENABLE = yes`;
- розмістити файл `vial.json` із визначенням клавіатури у каталозі `keymaps/vial/` (його можна згенерувати вручну або завантажити із VIA);
- згенерувати унікальний UID запуском скрипту `python3 util/vial_generate_keyboard_uid.py`, після чого вставити його до файлу `config.h` разом із директивою `#pragma once`;
- для активації захисту вказати координати комбінації розблокування за допомогою макросів `VIAL_UNLOCK_COMBO_ROWS` та `VIAL_UNLOCK_COMBO_COLS` у `config.h`, або, якщо захист не потрібен – додати до `rules.mk` директиву `VIAL_INSECURE = yes` [4].

Після завершення підготовчих етапів виконується складання прошивки командою `make keyboards/your_keyboard:vial`, де `your_keyboard` – шлях до

Змін.	Арк.	№ докум.	Підпис	Дата

потрібного каталогу. У результаті створюється готова до завантаження прошивка [4].

У підсумку, наведений процес дозволяє повністю підготувати клавіатуру до взаємодії з програмним забезпеченням Vial. Завдяки широкому набору функціональних можливостей, які включають роботу з макросами, RGB-підсвічуванням, енодерами та багатошаровими розкладками, Vial є потужним інструментом для досвідчених користувачів і розробників. Попри це, слід враховувати, що інтерфейс програми може виявитися дещо складним для новачків, тоді як VIA залишається привабливим рішенням для налаштування клавіатур із мінімальними зусиллями.

1.1.4 Проєкт Zephyr Mechanical Keyboard Firmware

Zephyr Mechanical Keyboard Firmware – це програмне забезпечення для клавіатури з відкритим вихідним кодом, побудоване на основі операційної системи реального часу (RTOS) проєкту Zephyr™. ZMK можна використовувати як для дротових, так і для бездротових пристроїв введення [5].

Основні функції та можливості ZMK:

- Підтримка бездротових та дротових розділених клавіатур: ZMK підтримує як бездротові клавіатури через Bluetooth (BLE), так і дротові варіанти підключення через USB.
- Енергоефективність: Вбудовані механізми зниження енергоспоживання дозволяють значно продовжити термін роботи на батареї, зокрема завдяки сплячим режимам та енергозберігаючим режимам для периферійних пристроїв.
- Підтримка енодерів та підсвічування: Прошивка підтримує енодери та RGB-підсвічування на основі світлодіодів для додаткової кастомізації.

Змін.	Арк.	№ докум.	Підпис	Дата

- Підтримка різноманітних пристроїв: ZMK може працювати з різними типами пристроїв, такими як сенсорні панелі Multitouch, миші та інші вказівні пристрої.
- Налаштування та конфігурації: Користувачі можуть створювати та зберігати власні налаштування клавіатури, визначати карти клавіш і шари, а також використовувати макроси і комбінації клавіш.
- Гнучке керування поведінкою клавіш: ZMK підтримує утримання натискань, Tap-Dances, Sticky клавіші (One Shot), комбінації клавіш та макроси для розширеної функціональності [5].

ZMK підтримує широкий спектр 32-розрядних мікроконтролерів, зокрема чіпи від Nordic Semiconductor, такі як nRF52840, nRF52833 та nRF5340. Також є підтримка мікроконтролерів STM32, зокрема моделей STM32F411CEU6 та STM32F401CEU6 на платах BlackPill. Крім того, ZMK підтримує мікроконтролери RP2040, такі як SparkFun Pro Micro RP2040, Adafruit QT Py RP2040 і Seeeduno XIAO RP2040. Водночас ZMK не підтримує мікроконтролери на базі 8-розрядних чіпів, таких як ATmega32U4, оскільки Zephyr™ RTOS, на якому базується прошивка, підтримує тільки 32-розрядні та 64-розрядні платформи [5].

Для налаштування клавіатури на базі ZMK необхідно створити окремий репозиторій на GitHub, у якому зберігатиметься конфігурація користувача. Налаштування виконується за допомогою спеціального скрипту, що автоматизує процес. Під час його запуску користувач обирає модель клавіатури, контролер та вводить дані для інтеграції з GitHub. Після ініціалізації репозиторію можна редагувати файл `keymap.overlay`, змінювати розкладку клавіш і вмикати або вимикати необхідні функції ZMK. Після збереження змін GitHub Actions автоматично компілює прошивку [5].

Згенерований файл прошивки можна завантажити з розділу «Actions» у репозиторії. Для прошивки контролер потрібно перевести в режим завантаження, після чого скопіювати відповідний UF2-файл у з'явлений у

системі USB-накопичувач. Після завершення процесу мікроконтролер автоматично перезапуститься, і клавіатура буде готова до роботи [5].

У разі використання розділеної клавіатури кожену половину потрібно прошити окремо, проте лише головний контролер (зазвичай у лівій частині) передаватиме натискання клавіш через USB або Bluetooth [5].

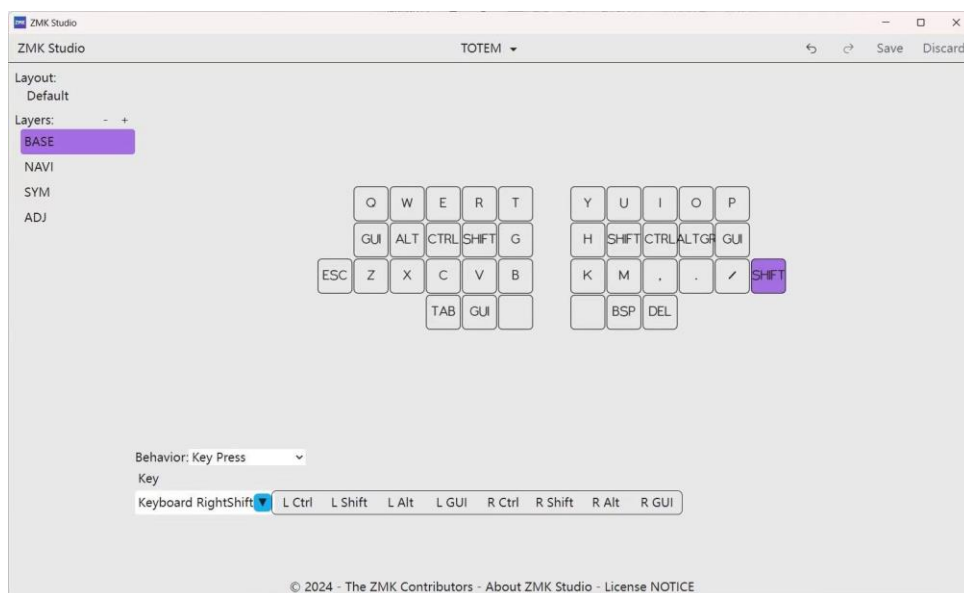


Рисунок 4 – Зовнішній вигляд ZMK studio

Також від ZMK існує графічний інтерфейс для налаштування ZMK Studio – це інструмент, який дозволяє користувачам змінювати шари клавіатури на пристроях, що працюють на базі прошивки ZMK, без необхідності перепрошивки. Це досягається завдяки поєднанню вдосконалень прошивки та спеціального фронтенд-додатку для взаємодії з клавіатурою [5].

Основні можливості ZMK Studio включають зміну ключових відображень під час використання клавіатури, підключення через USB та Bluetooth, призначення поведінки клавіш, а також конфігурацію слоїв та умовних переходів. Щоб скористатися можливостями ZMK Studio, потрібно вносити певні зміни до прошивки, що дозволить ZMK Studio вносити зміни під час роботи [5].

ZMK Studio доступний як вебзастосунок для браузерів, а також як нативний застосунок для Windows, Linux та macOS, який можна завантажити з

Змін.	Арк.	№ докум.	Підпис	Дата

сторінки завантажень. Вебзастосунок підтримує підключення через USB, тоді як нативні додатки також підтримують Bluetooth на Linux.

Використання ZMK Studio значно спрощує процес налаштування та персоналізації клавіатури, роблячи його більш інтуїтивно зрозумілим та доступним для користувачів.

1.1.5 Прошивка КМК на основі Python

КМК – це прошивка для механічних клавіатур, яка працює на платформі CircuitPython, і працює з багатьма контролерами, що підтримують CircuitPython. Процес її налаштування передбачає завантаження прошивки, копіювання файлів на USB-накопичувач клавіатури і створення основного скрипта для визначення функціональності клавіатури [6].

З КМК можна змінювати розкладки клавіш, додавати макроси, а також налаштовувати шари, HoldTap, TapDance і RGB-підсвічування. Є підтримка енкодерів, трекболів, стенографічного режиму та деяких інших пристроїв. Окрім цього, можна працювати з медіаклавішами, налаштовувати енергозбереження і навіть робити split клавіатури [6].

КМК підтримує мікроконтролери, такі як nice!nano, ItsyBitsy M4 Express, RP2040 та Adafruit ItsyBitsy nRF52840 Express. Для інших мікроконтролерів потрібно, щоб на них був встановлений CircuitPython, а також щоб вони мали мінімум 256 КБ флеш-пам'яті і підтримували HID через USB або Bluetooth [6].

Один із плюсів КМК – це можливість попередньої компіляції, що пришвидшує завантаження і зменшує використання пам'яті, що є важливим для пристроїв з обмеженим обсягом флеш-пам'яті. Можна компілювати вручну через утиліту mru-cross або завантажити вже готову версію з офіційного репозиторію [6].

Хоча КМК не має власної графічної оболонки, існує стороннє програмне забезпечення, таке як ROG, яке спрощує процес налаштування клавіатури. ROG

дозволяє працювати з розкладками через зручний інтерфейс, а також створювати прошивки без необхідності знання Python.

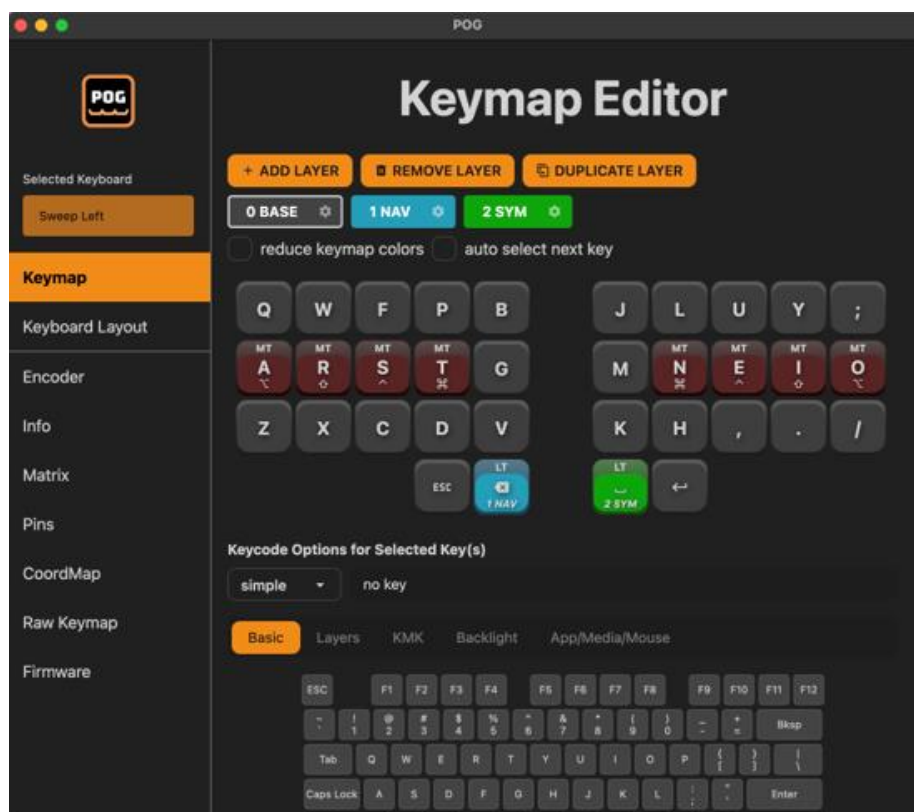


Рисунок 6 – Зовнішній вигляд POG

Для налаштування, потрібно встановити CircuitPython на мікроконтролер та підключити його до комп'ютера, відкривши POG. Вибравши накопичувач, можна перевірити чи є вже встановлений KMK, якщо ні, то POG автоматично завантажить і встановить його [7].

Далі потрібно вибрати тип розводки: матричну чи пряму, вказати напрямок діодів і налаштувати виводи. Також важливо правильно визначити координати клавіш, щоб прошивка коректно обробляла натискання. Для цього можна використовувати текстову область в POG, натискаючи клавіші по черзі [7].

Налаштування розкладки в POG дозволяє змінювати положення, вигляд клавіш, їхній розмір і навіть обертання. Користувач може створювати різні варіанти розкладок для різних потреб, наприклад, між ISO та ANSI Enter. Для редагування функцій клавіш можна вибирати їх в інтерфейсі та призначати відповідні команди або коди [7].

Змін.	Арк.	№ докум.	Підпис	Дата

1.2 Переваги та недоліки існуючих рішень

Кожне розглянуте, в попередньому розділі, рішення має свої переваги та недоліки.

QMK є беззаперечним лідером у сфері створення та налаштування кастомних клавіатур. Він поєднує в собі низку зручних інструментів, що підходять користувачам із різними потребами та рівнем досвіду. Наприклад, QMK Configurator і QMK Toolbox є чудовим вибором для нескладного, базового налаштування пристрою без необхідності працювати з кодом. Натомість QMK MSYS відкриває доступ до значно ширшого функціоналу, оскільки дозволяє взаємодіяти безпосередньо з вихідним кодом прошивки. Це надає змогу не лише точно налаштувати поведінку клавіатури, а й створювати повністю індивідуальні рішення, адаптовані до конкретних завдань користувача.

Водночас існують і певні недоліки. Окрім необхідності безпосередньої роботи з кодом під час створення власних рішень, QMK має обмежену підтримку бездротового з'єднання. Офіційно підтримуються лише модуль RN-42 та мікроконтролер Adafruit Feather 32u4 Bluefruit LE, який є досить габаритним і дорогим. Це значно ускладнює розробку повноцінної бездротової клавіатури, що робить QMK не надто зручним варіантом для таких рішень. Крім того, популярні конфігуратори VIA та Vial, що базуються на прошивці QMK, мають ті самі обмеження – вони не підтримують бездротового з'єднання, тому для налаштування клавіатури необхідно підключатися до неї у дротовому режимі.

Натомість ZMK має протилежну ситуацію, він має підтримку багатьох MCU з можливістю бездротового з'єднання, і на сьогодні є найкращим вибором для створення бездротових клавіатур. Проте ZMK не має зручного графічного інтерфейсу – усе налаштування здійснюється через конфігураційні файли, що суттєво підвищує поріг входження навіть для досвідчених користувачів, а ZMK Studio, як і VIA, працює лише з прошитими пристроями.

KMK у поєднанні з ROG є непоганим рішенням, яке надає зручний і інтуїтивно зрозумілий інтерфейс для початкового налаштування клавіатури.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ. 467200.004 ПЗ

Арк.

22

Проте сам POG – доволі новий інструмент, і на даному етапі він ще не підтримує багато важливих функцій, таких як Bluetooth, mod-tap, tap dance, макроси та послідовності. У разі потреби в розширених можливостях користувачеві доведеться знову звертатися до редагування коду безпосередньо у КМК, що суттєво ускладнює процес для новачків.

1.3 Обґрунтування теми дипломного проекту

Аналіз існуючих рішень показує, що вони пропонують широкий функціонал не лише для роботи з готовими пристроями, а й для повноцінного налаштування клавіатур з нуля. Проте, цей процес часто супроводжується необхідністю занурення в програмний код, а також постійним перемиканням між різними інструментами в процесі роботи. Крім того, не всі наявні рішення підтримують створення бездротових клавіатур, що є суттєвим обмеженням для сучасних користувачів.

Таким чином, існує потреба в більш гнучкому та інтуїтивно зрозумілому рішенні, яке поєднує широкі можливості налаштування з простим і зручним інтерфейсом, а також забезпечує повноцінну підтримку бездротових клавіатур. Створення такої системи дозволить суттєво спростити процес конфігурації клавіатур, зробивши його доступним навіть для користувачів без досвіду роботи з прошивками або програмуванням.

У якості основної апаратної платформи для клавіатур планується використання мікроконтролерів сімейства ESP32. Це серія недорогих, але потужних мікроконтролерів, розроблених компанією Espressif Systems, які поєднують високу обчислювальну продуктивність із широкими можливостями бездротового зв'язку. ESP32 мають вбудовану підтримку Bluetooth та Bluetooth Low Energy (LE), що дає змогу створювати повністю бездротові рішення без потреби у зовнішніх модулях. Крім того, вони оснащені достатнім обсягом оперативної та флеш-пам'яті, що дозволяє реалізовувати складну логіку роботи

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ. 467200.004 ПЗ

Арк.

23

клавіатур, включаючи макроси, багаторівневі шари, а також підтримку спеціалізованих функцій [8].

Ще однією суттєвою перевагою ESP32 є їх низьке енергоспоживання, що особливо важливо для автономної роботи бездротових пристроїв. Завдяки підтримці режимів енергозбереження (sleep/ultra-low power) такі мікроконтролери можуть функціонувати на одному заряді акумулятора протягом тривалого часу. Важливо й те, що ESP32 мають велику різноманітність модифікацій від базових моделей до покращених варіантів із додатковими портами, датчиками та розширеними можливостями вводу/виводу, що відкриває шлях до розробки різнопланових і функціонально багатих клавіатур [8].

Втім, попри всі технічні переваги, ESP32 залишається відносно «молодим» рішенням і наразі не має повноцінної підтримки у більшості популярних прошивок для клавіатур. Наприклад, хоча операційна система Zephyr, на якій базується ZMK, формально підтримує ESP32, сам ZMK офіційно не підтримує дані мікроконтролери. Подібна ситуація спостерігається і з прошивкою KMK, яка працює на базі CircuitPython.

Існує декілька користувацьких проєктів, що намагаються реалізувати підтримку ESP32, серед яких варто відзначити RMK (Rust Keyboard Firmware)[9] та MK32[10]. Втім, такі ініціативи часто мають низку суттєвих обмежень: відсутній графічний інтерфейс, неповна документація, обмежена підтримка моделей ESP32, а активна розробка наразі відсутня.

Необхідно передбачити створення десктопного застосунку для Windows, який дозволить налаштовувати механічні клавіатури на базі мікроконтролерів сімейства ESP32. Система об'єднуватиме всі необхідні інструменти для конфігурації та прошивки клавіатур, а також забезпечуватиме інтуїтивно зрозумілий графічний інтерфейс. Таке рішення суттєво спростить процес кастомізації клавіатур і зробить його доступним для ширшого кола користувачів, одночасно розширюючи можливості використання платформи ESP32 у сфері DIY-проєктів.

2. АНАЛІЗ ОБРАНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ

Цей розділ присвячений огляду обраних технологій, необхідних для реалізації проєкту.

2.1 Мова програмування C#

C# – це сучасна, об'єктно-орієнтована мова програмування, яка була розроблена компанією Microsoft як частина платформи .NET, призначена для розробки надійних, безпечних і масштабованих застосунків. Вона поєднує в собі кращі риси таких мов, як C++, Java та Visual Basic, і водночас вводить унікальні концепції та інструменти, що робить її однією з найпопулярніших мов програмування в різних галузях [11].

C# підтримує принципи об'єктно-орієнтованого програмування, такі як інкапсуляція, спадкування та поліморфізм, що дозволяє створювати модульний і повторно використовуваний код. Класи та інтерфейси забезпечують чітку організацію програми, полегшуючи її супровід та модернізацію. Мова є строго типізованою, що зменшує ймовірність помилок через неправильне використання типів даних, виявляючи проблеми ще на етапі компіляції, автоматичне керування пам'яттю через збирач сміття звільняє пам'ять від непотрібних об'єктів, що спрощує розробку програмного забезпечення та зменшує ризик витоків пам'яті [11].

Особливістю C# є його тісна інтеграція з .NET – сучасною кросплатформною програмною платформою від Microsoft, яка включає набір бібліотек, компіляторів та середовище виконання Common Language Runtime (CLR) для розробки та виконання додатків різного призначення. Завдяки інтеграції з .NET, C# забезпечує розробку продуктивних і безпечних програм для Windows, macOS, Linux, Android, iOS та хмарних сервісів [11].

Платформа .NET надає широкі можливості для створення різноманітних додатків. Наприклад, ASP.NET застосовується для розробки масштабованих вебзастосунків і API, Entity Framework – для роботи з базами даних, WPF і WinForms – для настільних додатків із графічним інтерфейсом, а .NET MAUI

дозволяє створювати кросплатформні мобільні додатки. Окремо варто відзначити інтеграцію з Microsoft Azure, що робить C# популярним вибором для хмарних рішень [11].

Сучасні версії C#, включають записи (records), покращені шаблони відповідності (pattern matching), асинхронне програмування з async/await та елементи функціонального програмування [11].

Завдяки своїй гнучкості, потужним можливостям і простоті використання, C# залишається однією з найпопулярніших мов програмування у світі. Її багатий набір інструментів та бібліотек забезпечує комфортну роботу в різних галузях – від розробки мобільних додатків і ігор до складних корпоративних систем.

2.2 Графічний фреймворк Windows Presentation Foundation

Windows Presentation Foundation (WPF) – це сучасний фреймворк для створення настільних додатків із графічним інтерфейсом користувача у середовищі Windows. Він є частиною платформи .NET і використовує незалежний від роздільної здатності, векторно-орієнтований рушій рендерингу, який дозволяє максимально ефективно використовувати можливості сучасного графічного обладнання. Завдяки цьому WPF забезпечує розробникам широкі можливості для створення гнучких, інтерактивних і візуально привабливих інтерфейсів [12].

Однією з ключових особливостей WPF є використання Extensible Application Markup Language (XAML) – декларативної мови розмітки на основі XML, що дозволяє описувати зовнішній вигляд додатка, структуру його елементів, їх властивості та стилі. XAML дозволяє організовувати інтерфейс у вигляді дерева вкладених елементів, яке забезпечує логічну та зручну модель для побудови складних вікон і панелей керування. Окрім XAML, для реалізації логіки взаємодії використовується код на мові C#, який зв'язується з розміткою за допомогою механізму code-behind. Для цього застосовується атрибут x:Class

Змін.	Арк.	№ докум.	Підпис	Дата

та метод `InitializeComponent()`, який викликається в конструкторі класу та забезпечує ініціалізацію компонентів, описаних у XAML [12].

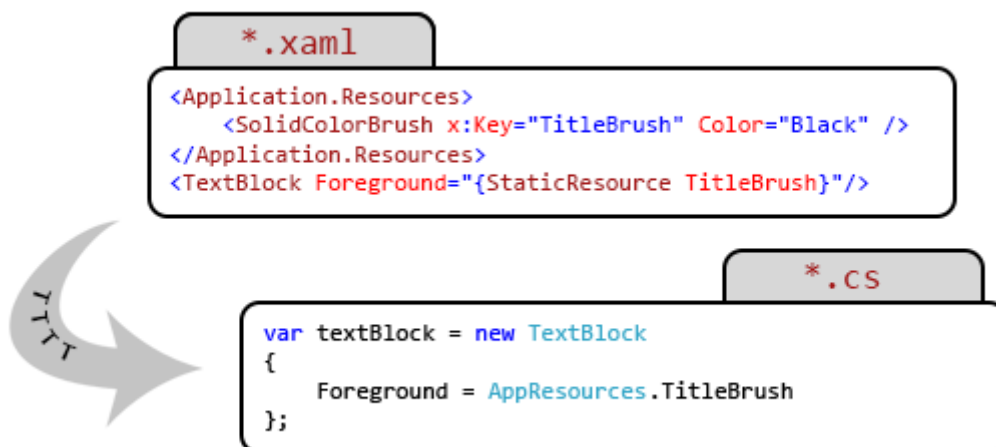


Рисунок 7 – Зв'язок між XAML-ресурсами і C#-кодом за допомогою механізму code-behind

Основою будь-якого інтерфейсу у WPF є елементи управління – класи, які об'єднують графічне представлення та функціональність. До них належать як стандартні елементи, такі як кнопки, текстові поля та списки, так і складніші компоненти, які можна гнучко налаштувати за допомогою стилів, шаблонів та анімацій. Завдяки цьому розробник отримує повний контроль над зовнішнім виглядом і поведінкою елементів без необхідності змінювати їхню базову логіку, що дозволяє створювати інтерфейси будь-якої складності та дизайну [12].

Однією з важливих особливостей WPF є підтримка прив'язки даних (Data Binding), що дає змогу синхронізувати відображення інформації в інтерфейсі з її джерелом автоматично, без необхідності вручну оновлювати значення. Це значно спрощує розробку динамічних форм і дозволяє легко реалізовувати інтерактивні елементи, які реагують на зміни даних у режимі реального часу [12].

Програмування у WPF базується на знайомих для розробників .NET принципах: створенні екземплярів класів, налаштуванні їхніх властивостей, виклику методів і обробці подій. WPF реалізований як частина .NET і використовує переважно типи з простору імен `System.Windows`. Також WPF має підтримку маршрутизованих подій, що дозволяють подіям передаватися через

Змін.	Арк.	№ докум.	Підпис	Дата

ієрархію елементів інтерфейсу. Завдяки цьому обробку дій користувача, таких як клацання кнопок чи введення тексту, можна централізувати або делегувати іншим елементам, що значно підвищує гнучкість і зручність організації логіки взаємодії [12].

Для більш складних сценаріїв взаємодії передбачено систему команд (Commands) – механізм, який відокремлює логіку виконання дії від способу її виклику. Це дає змогу використовувати одну й ту саму команду для кількох різних елементів інтерфейсу, наприклад, викликати команду копіювання або вставлення як за допомогою кнопки, так і через меню або комбінацію клавіш [12].

WPF забезпечує сучасний, гнучкий і потужний підхід до розробки настільних додатків у середовищі Windows, дозволяючи ефективно поєднувати декларативне описання інтерфейсу та імперативну програмну логіку.

2.3 Середовище розробки Visual Studio

Visual Studio – це потужне інтегроване середовище розробки (IDE), створене компанією Microsoft, яке широко використовується для створення програмного забезпечення на платформах Windows, Android, iOS та web. Це універсальний інструмент, який підтримує різноманітні мови програмування, серед яких C#, C++, Python, JavaScript, TypeScript та інші. Завдяки своїй функціональності та гнучкості, Visual Studio є одним з найпопулярніших середовищ серед розробників у всьому світі [13].

Однією з основних переваг Visual Studio є зручна система управління проєктами та файлами. Середовище дозволяє легко організувати великі проєкти, структуровано розміщуючи вихідні файли, ресурси, бібліотеки та конфігураційні документи. Вбудований редактор коду забезпечує підсвічування синтаксису, автоматичне доповнення коду та зручну навігацію по проєкту [13].

Особливе значення має система налагодження, що дозволяє запускати застосунки у режимі відладки, встановлювати контрольні точки, переглядати значення змінних, виконувати покрокове виконання коду та аналізувати стек

викликів. Це дозволяє ефективно знаходити та виправляти помилки на ранніх етапах розробки [13].

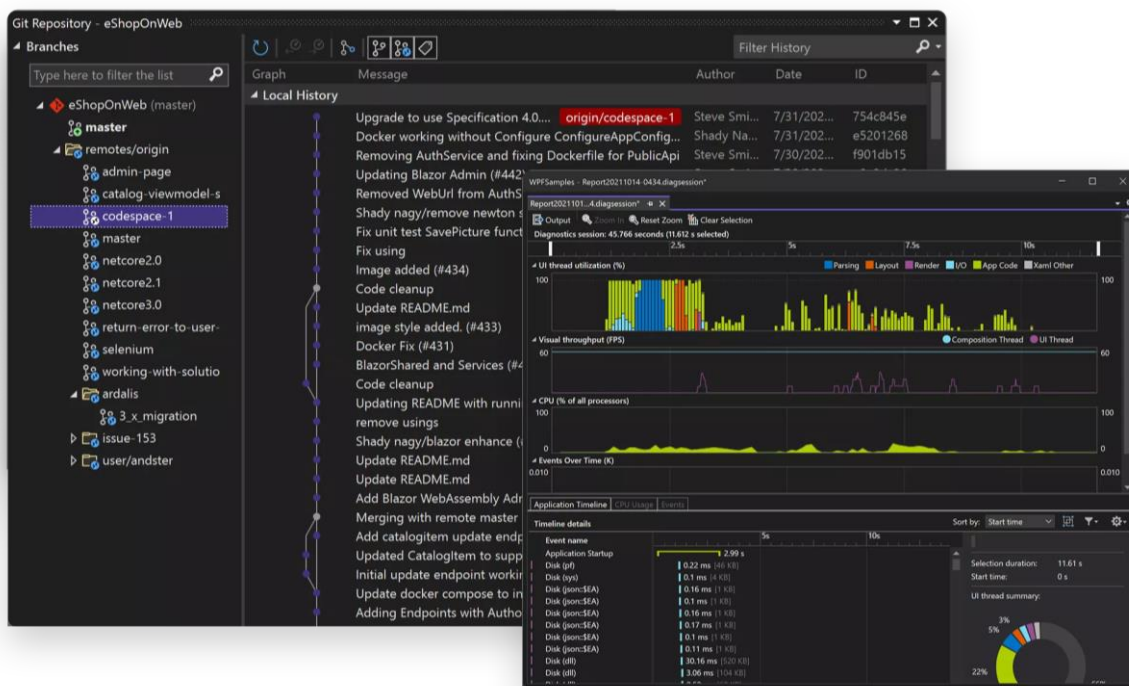


Рисунок 8 – Visual Studio

Visual Studio також має широкий набір інструментів для тестування, створення звітів про помилки та аналізу продуктивності застосунків. Середовище підтримує інтеграцію з системами контролю версій, такими як Git, що дає можливість зручно організувати колективну розробку, відстежувати зміни в коді та працювати з віддаленими репозиторіями [13].

Крім цього, Visual Studio надає можливість розширення функціональності за допомогою модулів і плагінів, що дозволяє адаптувати середовище до потреб конкретного проєкту. Власна екосистема розширень охоплює інструменти для роботи з базами даних, розгортання застосунків, хмарних рішень, аналізу коду та інших завдань [13].

У рамках дипломного проєкту Visual Studio було використано як основний інструмент для написання програмного коду, його тестування та налагодження. Завдяки гнучкості налаштувань, широким можливостям для розробки та підтримці сучасних технологій, це середовище забезпечило

Змін.	Арк.	№ докум.	Підпис	Дата

ефективну організацію роботи над системою, а також зручний контроль за всіма етапами створення програмного забезпечення.

2.4 Система контролю версій Git та платформа GitHub

Під час розробки програмного забезпечення важливо забезпечити збереження історії змін, можливість повернення до попередніх версій та ефективну організацію спільної роботи над кодом. Для цього використовуються системи контролю версій, серед яких найбільш популярною є Git.

Git – це сучасна розподілена система контролю версій, яка дозволяє зберігати повну історію змін проєкту на кожному локальному пристрої розробника. Завдяки цьому забезпечується автономність роботи: всі операції над історією, створення гілок, порівняння версій та об'єднання змін можна виконувати локально, без постійного з'єднання з сервером [14].

У системі Git кожна зафіксована зміна проєкту зберігається у вигляді коміту (commit) – зафіксованого стану файлів на певному етапі розробки. Коміт містить інформацію про автора, дату та опис внесених змін, що дозволяє відстежувати історію роботи над проєктом та за потреби повертатися до попередніх версій. Для організації паралельної роботи Git підтримує гілки (branches) – незалежні лінії розробки, які дають змогу одночасно створювати новий функціонал, виправляти помилки чи експериментувати, не зачіпаючи основної версії програми. По завершенні роботи зміни з різних гілок можуть бути об'єднані за допомогою операції злиття (merge), що особливо актуально для командних проєктів [14].

Для зберігання репозиторію було обрано платформу GitHub. Це платформа для розміщення Git-репозиторіїв, яка надає зручний вебінтерфейс для перегляду коду, відстеження змін, ведення документації та обговорення технічних питань. Крім того, GitHub дозволяє налаштовувати права доступу до репозиторію, створювати pull requests – запити на об'єднання змін, а також інтегрувати сторонні сервіси для автоматизації тестування та розгортання додатків [15].

					<i>ІАЛЦ. 467200.004 ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		30

Основні переваги використання GitHub:

- Надійне віддалене зберігання коду.
- Відкритий або приватний доступ до проєкту.
- Зручний перегляд історії змін.
- Організація спільної роботи кількох розробників.
- Інструменти для контролю якості коду та автоматизації.

Оскільки більшість популярних проєктів для створення та налаштування механічних клавіатур є відкритими, було прийнято рішення, що й система, розроблена в межах дипломного проєкту, буде доступною для спільноти. Для цього створено публічний репозиторій на GitHub, де зберігаються вихідні файли, супровідна документація та історія змін. Це дає змогу не лише забезпечити надійне зберігання коду та організувати роботу над проєктом, а й надати можливість усім зацікавленим користувачам ознайомитися з реалізованою системою. Таким чином, проєкт долучається до сучасної практики відкритого програмного забезпечення, що сприяє його подальшому розвитку, вдосконаленню та популяризації.

2.5 Додаткові інструменти

У процесі налаштування клавіатур одним із важливих етапів є створення відповідної прошивки та її подальше завантаження на мікроконтролер. Для цього було обрано відповідні інструменти, які забезпечують зручну та ефективну роботу з мікроконтролерами сімейства ESP32.

2.5.1 Arduino Command Line Interface

Arduino CLI – це офіційний інструмент для роботи з платформою Arduino через командний рядок, без використання графічного середовища розробки. Цей інструмент надає розробникам гнучкість і можливість автоматизувати робочі процеси, що особливо корисно для великих проєктів або інтеграції зі сторонніми системами [16].

Змін.	Арк.	№ докум.	Підпис	Дата

Arduino CLI підтримує всі основні функції середовища Arduino IDE: керування платами та бібліотеками, створення, компіляцію та завантаження прошивок на мікроконтролери, а також моніторинг серійного порту для перегляду даних у реальному часі. Робота з інструментом здійснюється через текстові команди в терміналі, що дозволяє інтегрувати його у системи безперервної інтеграції (CI/CD), сценарії автоматизації, або використовувати в середовищах, де графічний інтерфейс недоступний [16].

Перевагою Arduino CLI є його відкритий вихідний код і кросплатформенність – він працює на Windows, macOS та Linux. Крім того, цей інструмент дає можливість гнучко налаштовувати параметри компіляції, завантаження та налаштування плат [16].

Arduino CLI активно застосовується для розробки вбудованих систем, тестування, створення власних систем складання проєктів і підключення Arduino до сторонніх редакторів коду та IDE.

У межах дипломного проєкту Arduino CLI використовується як інструмент для компіляції прошивки клавіатури для мікроконтролера. Програмний код клавіатури написаний на спрощеній версії мови C/C++, яку використовує Arduino.

2.5.2 Утиліта Esptool

Esptool – це утиліта з відкритим програмним кодом на базі Python, яка призначена для прошивки, налаштування та взаємодії з мікроконтроллерами від Espressif [17].

Esptool працює через завантажувач ПЗП, вбудований у мікроконтролер, або використовує власну флеш-заглушку – невелику програму, що тимчасово замінює стандартний завантажувач для виправлення його обмежень та розширення можливостей [17].

Основні функції включають:

- Зчитування, запис, стирання та перевірка двійкових даних, що зберігаються у флеш-пам'яті.

- Зчитування характеристик чіпа та інших даних (MAC-адреса або ідентифікатор флеш-чіпа).
- Зчитування та запис одноразово програмованих eFuses.
- Підготовка бінарних образів, готових до прошивки.
- Аналіз, збирання та об'єднання бінарних зображень.

Esptool можна використовувати як командний рядок, що є основним способом роботи з утилітою, так і інтегрувати як бібліотеку в Python-скрипти для автоматизації процесів прошивання та обслуговування мікроконтролерів [17].

У межах дипломного проєкту esptool застосовується для об'єднання файлів прошивки, згенерованих після компіляції за допомогою Arduino CLI, в один суцільний образ, та для подальшого завантаження цього файлу в пам'ять мікроконтролера, забезпечуючи швидке й надійне завантаження коду безпосередньо на пристрій.

2.6 Підтримувані мікроконтролери

Компанія Espressif Systems є відомим виробником мікроконтролерів та бездротових комунікаційних рішень. Найбільш популярною лінійкою мікроконтролерів компанії є серія ESP32, яка включає кілька підсерій, а саме:

- ESP32: Перший чіп компанії з вбудованими модулями Wi-Fi та Bluetooth Classic/LE. Побудований на архітектурі Xtensa LX6. Завдяки універсальності та достатній продуктивності підходить для широкого кола загальних IoT-проєктів.
- ESP32-S: Високопродуктивні чіпи з підтримкою Wi-Fi та Bluetooth LE. Мають розширені можливості для обробки мультимедіа.
- ESP32-C: серія, заснована на відкритій архітектурі RISC-V. Підтримує Wi-Fi 4/6, Bluetooth LE, Zigbee та Thread. Оптимізована для енергоефективних IoT-пристроїв.

- ESP32-H: Мікроконтролери без Wi-Fi, але з підтримкою Bluetooth LE, Zigbee та Thread. Призначені для пристроїв з низьким енергоспоживанням.
- ESP32-P: Високопродуктивні мікроконтролери без бездротових інтерфейсів. Призначені для обробки зображень, НМІ та периферійних обчислень.

У розроблюваній системі передбачена підтримка таких мікроконтролерів, як класичний ESP32, а також моделі ESP32-C3, ESP32-C6, ESP32-S3 та ESP32-H2 у форматах mini та supermini.



Рисунок 9 – ESP32-C6-Zero Mini

Використання зменшених мікроконтролерів має численні переваги. Завдяки зменшеним розмірам плати такі мікроконтролери дозволяють суттєво оптимізувати загальні габарити пристрою та є зручними для монтажу, а доступна вартість мініатюрних модулів робить їх більш вигідними, водночас енергоефективність міні-версій у порівнянні з повнорозмірними модулями досягається завдяки спрощеним апаратним можливостям, що зменшує загальне споживання енергії й дозволяє клавіатурі працювати довше від автономного живлення. При цьому зменшення обчислювальної потужності у таких мікроконтролерах цілком задовольняє вимоги клавіатур, де немає потреби у високій продуктивності для обробки великих обсягів даних.

Змін.	Арк.	№ докум.	Підпис	Дата

Таблиця 1 – Основні характеристики обраних моделей мікроконтролерів

Параметри		ESP32	ESP32-C3 mini	ESP32-C6 mini	ESP32-S3 mini	ESP32-H2 mini
Архітектура		Xtensa LX6 (2 ядра)	RISC-V (1 ядро)	RISC-V (1 ядро)	Xtensa LX7 (2 ядра)	RISC-V (1 ядро)
Частота процесора		до 240 МГц	до 160 МГц	до 160 МГц	до 240 МГц	до 96 МГц
Розмір пам'яті	Flash	4 МБ	4 МБ	4 МБ	4 МБ	2/4 МБ
	ROM	–	384 КБ	320 КБ	384 КБ	128 КБ
	SRAM	520КБ	400 КБ	512 КБ (HP) 16 КБ (LP)	400 КБ	320 КБ
Кількість контактів		30/38	15	22	24	19
Споживання енергії	Активний режим	90-130 мА	30-50 мА	30-60 мА	80-120 мА	20-50 мА
	Режим сну	до 20 мкА	~5 мкА	~15 мкА	~43 мкА	~10 мкА

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ. 467200.004 ПЗ

Арк.

35

3. ОПИС ТА ТЕСТУВАННЯ СИСТЕМИ НАЛАШТУВАНЬ МЕХАНІЧНИХ КЛАВІАТУР

3.1 Структура системи

У цьому пункті описано основні модулі системи та їх призначення. Загальна схема роботи системи див. ІАЛЦ.467200.005 Д1.

MainMenu. Цей модуль являє собою головне меню, містить три кнопки, за допомогою яких користувач може перейти до конфігуратора, флешера або тестера кнопок.

Flasher. Відповідає за завантаження прошивки на мікроконтролер через СОМ-порт. Flasher перевіряє наявність вибраного СОМ-порту, мікроконтролера, файлу прошивки, та виконує прошивку за допомогою esptool та виводить інформацію про хід процесу в вікно зневадження. Повний алгоритм роботи цього модуля наведено в ІАЛЦ.467200.008 Д4.

Keyboard_Tester. Дозволяє перевірити коректність зроблених налаштувань. Модуль відслідковує натискання клавіш, підсвічуючи відповідні кнопки у графічному інтерфейсі.

KeyboardSettings. У цьому файлі зберігаються налаштування клавіатури, такі як назва, модель мікроконтролера, кількість рядків і колонок, обрані піни, розкладки та макроси.

Configurator_ColRow_Select. Сторінка вибору кількості рядків і колонок для матриці клавіатури. Дозволяє користувачу ввести бажані значення та відображає попередній вигляд матриці, після чого зберігає їх в KeyboardSettings.

Configurator_Pins_Select. Забезпечує вибір пінів для рядків та стовпців згідно обраного мікроконтролера. Містить словник доступних пінів для кожної плати та динамічно генерує необхідну кількість полів для вибору пінів на основі заданих у Configurator_ColRow_Select кількості рядків і стовпців.

Змін.	Арк.	№ докум.	Підпис	Дата

Configurator_Layers. Дозволяє користувачу вводити розкладку клавіш для кожного шару клавіатури. Введені дані в текстовому вигляді нормалізуються та валідуються, після чого зберігаються у двовимірний масив у KeyboardSettings.

Configurator_Macros. Відповідає за створення та запис макросів клавіш для клавіатури. Реалізує фіксацію послідовності натискань та відпускань клавіш через Windows-hook. Здійснює нормалізацію імен клавіш відповідно до проєктного формату, перевіряє їх на відповідність допустимим значенням та зберігає макроси у KeyboardSettings як послідовність дій із значеннями клавіші та типом події.

Configurator_Save_firmware. Модуль забезпечує генерацію прошивки на основі налаштувань користувача. З цією метою генерується тимчасовий файл Settings.h, який містить усіх налаштування мовою C++. Далі використовується Arduino-cli для компіляції коду клавіатури, після чого отримані бінарні файли об'єднуються за допомогою esptool в один файл, який зберігається в обране користувачем місце. Крім цього, модуль дає змогу зберегти налаштування у форматі .json для подальшого використання чи обміну.

Styles. Містить опис стилів, які використовуються для створення графічного інтерфейсу.

3.2 Структура програмного коду клавіатури

Як вже зазначалося програмний код для клавіатури написаний на спрощеній мові C++, яку використовує Arduino, та має наступну структуру (див. ІАЛЦ.467200.006 Д2):

Файл main.ino є основним файлом для прошивки мікроконтролера, який реалізує Bluetooth-клавіатуру. Він відповідає за ініціалізацію апаратного забезпечення, сканування матриці клавіш, обробку натискань, перемикання шарів клавіатури, виконання макросів та відправлення даних через BLE.

Змін.	Арк.	№ докум.	Підпис	Дата

- `Setup` – викликає `setupPins()` для налаштування пінів матриці клавіатури, встановлює початковий шар та ініціалізує задачу для роботи Bluetooth (`bluetoothTask`).
- `Loop` – основний цикл, який перевіряє підключення до BLE і, якщо макроси не виконуються, викликає сканування матриці клавіатури.
- `setupPins` – налаштовує піни рядків як входи з підтяжкою до високого рівня, і піни стовпців як виходи, які за замовчуванням у високому стані. Ініціалізує масив `lastButtonState` для відстеження стану кнопок.
- `scanMatrix` – сканує матрицю клавіш, перевіряючи стан кожного піну. Виявляє натискання кнопок і виконує відповідну дію: перемикає шару, виконання макросу або відправлення звіту про натискання. Увесь алгоритм роботи цієї функції зображений на ІАЛЦ.467200.007 ДЗ в додатку А.
- `handleLayerSwitch` – Обробляє перемикає між шарами клавіатури.
- `sendKeyReport` – Формує і відправляє BLE-звіт (`InputReport`) про натискання клавіші.

`BLEConnection.h` відповідає за ініціалізацію та керування Bluetooth Low Energy (BLE) підключенням для ESP32, який працює як BLE HID-пристрій. Файл містить масив `REPORT_MAP`, який описує структуру HID-звітів для клавіатури. Ця структура визначає, які саме клавіші підтримуються, як обробляються комбінації натискань та в який спосіб ця інформація надсилається. Також є два класи `BLEServerCallbacks` і `BLECharacteristicCallbacks`. Перший клас відповідає за підключення або відключення пристрою. Як тільки встановлюється з'єднання, програма оновлює відповідний стан і дозволяє передавати дані. Якщо ж зв'язок розривається, передавання тимчасово вимикається. Інший клас стежить за тим, коли на пристрій надходять дані у зворотному напрямку, наприклад – сигнал про ввімкнення або вимкнення світлодіодів Caps Lock чи Num Lock. Уся робота з налаштуванням та запуском Bluetooth відбувається у функції `bluetoothTask`, в якій послідовно створюється сервер для з'єднання, реєструються обробники подій, формується сам HID-пристрій, додаються характеристики для обміну

даними та виставляються службові параметри. Також налаштовується захист з'єднання, щоб підключатися могли лише дозволені пристрої. Далі програма готує службову інформацію, яку бачать інші пристрої під час пошуку, та вмикає передачу цієї інформації у мережу. Після запуску всіх служб програма переходить у стан очікування, де безперервно працює доти, поки її не зупинять.

Файл Reports.h відповідає за опис структури даних, якими обмінюється клавіатура з підключеним пристроєм під час роботи. Структура InputReport визначає вигляд звіту, який клавіатура надсилає на пристрій. У цьому звіті зберігається інформація про модифікатори (наприклад, Shift, Ctrl чи Alt), службовий байт, і список із шести клавіш, що можуть бути натиснуті одночасно. Така кількість обрана відповідно до стандарту HID для клавіатур, який дозволяє передавати до шести одночасних натискань.

Друга структура OutputReport містить лише одне поле, яке призначене для зберігання стану світлодіодів на клавіатурі. Сюди можуть надходити сигнали про увімкнення чи вимкнення таких індикаторів, як Caps Lock, Num Lock або Scroll Lock.

Модуль Macros.h відповідає за обробку та виконання макросів. Він містить масив вказівників на макроси, функцію handleMacro – для визначення індексу макросу та executeMacro, яка виконує макрос, обробляючи послідовність команд: формує HID-звіти для натискань/відпускання клавіш, обробляє паузи, та відправляє звіти.

Файл Keycodes.h визначає усі можливі коди клавіш, спеціальних команд та модифікаторів для роботи клавіатури. Тут зібрані як стандартні HID-коди клавіш, так і службові значення для перемикання шарів, запуску макросів та виконання додаткових команд.

Settings.h є тимчасовим файлом і генерується окремо під час кожної компіляції й містить усі унікальні налаштування конкретної клавіатури, так як: назва пристрою, тип мікроконтролера, кількість рядків і стовпців, пін-коди,

Змін.	Арк.	№ докум.	Підпис	Дата

кількість шарів і їхні розкладки. Також у цьому файлі задаються макроси та послідовності їхніх комбінацій для подальшого виконання.

3.3 Тестування системи

Після запуску системи користувач потрапляє на головний екран, який є центральною точкою навігації в програмі. На цьому екрані доступні чотири основні пункти: Set Configuration, Flasher, Install Drivers та Keyboard Tester. Усі зазначені елементи забезпечують доступ до відповідних функцій програми. Також на екрані присутня емблема GitHub, яка дозволяє перейти до репозиторію проекту.

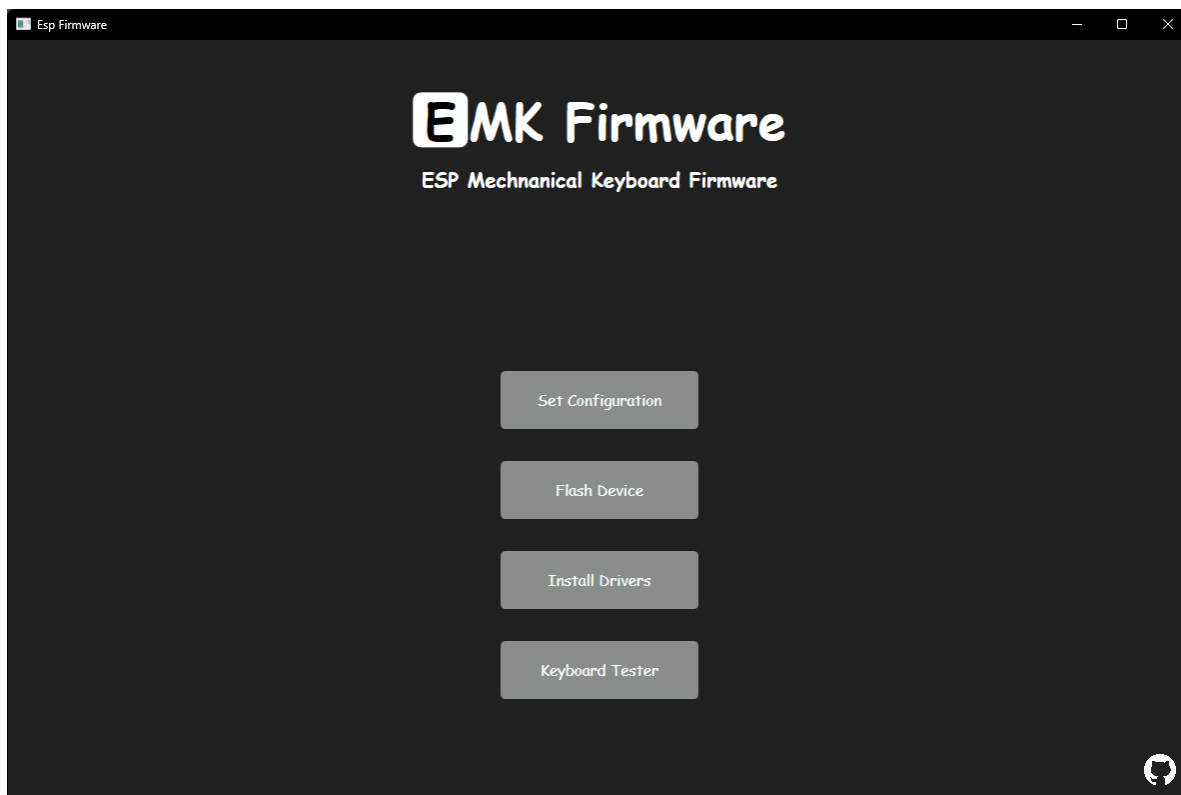


Рисунок 10 – Головне меню системи

Для початку роботи слід встановити драйвера для мікроконтролерів серії ESP32. Для цього потрібно натиснути кнопку Install Drivers, після чого на комп'ютер користувача буде встановлено драйвер CP210x USB to UART Bridge VCP Drivers. Це драйвер для пристроїв на основі чипів CP210x, що дозволяє перетворити USB-з'єднання на звичайний послідовний порт, що дає змогу комп'ютеру встановлювати послідовний зв'язок з мікроконтролерами.

Змін.	Арк.	№ докум.	Підпис	Дата

Після завершення встановлення драйверів можна переходити до створення налаштувань клавіатури за допомогою розділу Set Configuration.

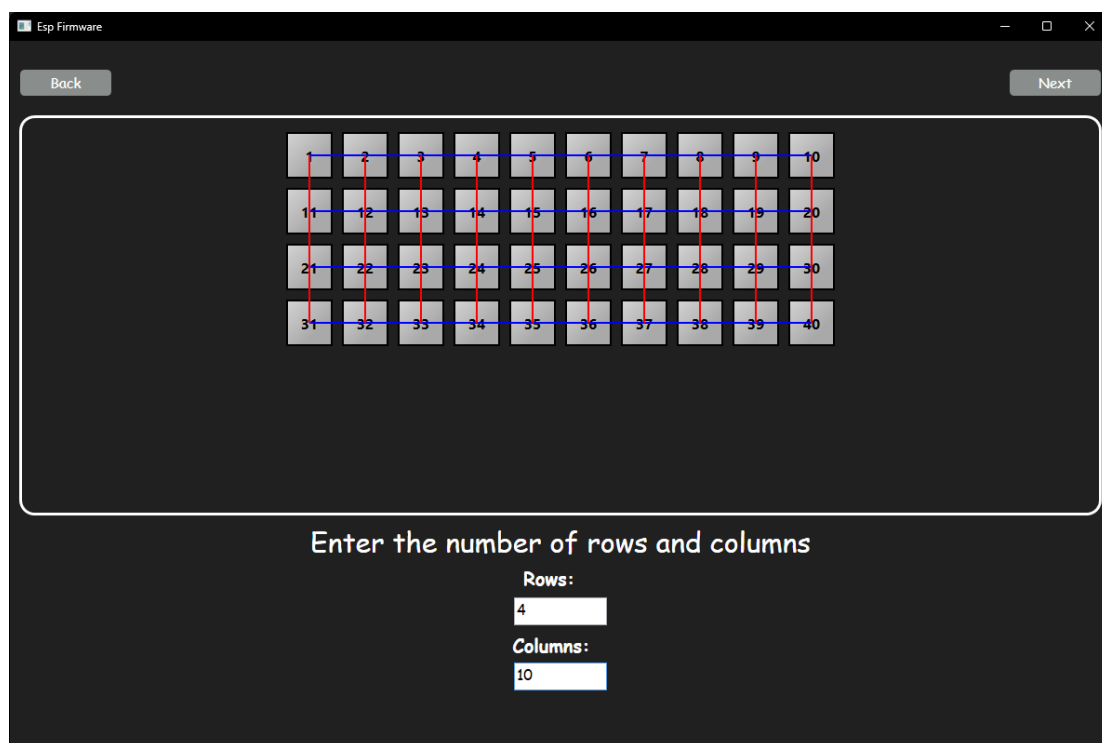


Рисунок 11 – Сторінка налаштування рядків/стовпців

Першим етапом налаштування є задання розміру матриці клавіатури. На цьому кроці користувач повинен ввести кількість стовпців і рядків, що відповідають фізичній структурі його клавіатури. Ці значення використовуються для формування основи майбутньої конфігурації: вони впливають на кількість пінів, які потрібно буде призначити, а також на розмір розкладок і кількість клавіш. Правильне введення цих параметрів є необхідним для коректної роботи всієї системи.

Для задання розміру матриці на сторінці передбачено два відповідних поля: Rows і Columns, де користувач має вказати бажані значення. Перехід до наступного етапу стане доступним лише після того, як обидва поля будуть заповнені – залишити їх порожніми неможливо. Крім цього, на сторінці розміщена область, у якій у спрощеному вигляді відображається структура заданої матриці. Ця візуалізація слугує для наочності та допомагає користувачеві краще уявити, як буде з'єднана матриця у кінцевому пристрої.

Змін.	Арк.	№ докум.	Підпис	Дата

Наступним етапом є вибір мікроконтролера та призначення пінів. Користувач повинен обрати модель мікроконтролера з переліку запропонованих варіантів, відповідно до тієї, яка використовується у його клавіатурі.

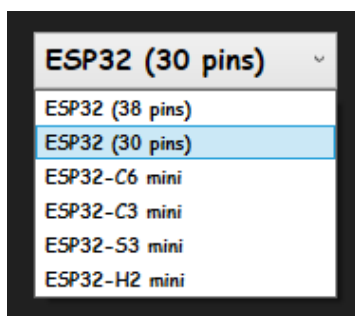


Рисунок 12 – Меню вибору мікроконтролера

Після вибору мікроконтролера відкривається можливість призначити піни для кожного рядка і стовпця матриці. Кількість доступних полів для введення пінів відповідає значенням, вказаним на попередньому етапі під час задання розміру матриці. Для зручності користувача на сторінці також відображається схема вибраного мікроконтролера, на якій позначено, які піни бажано використовувати, а які – не рекомендується. Це дозволяє уникнути помилок при конфігуруванні та спрощує процес налаштування.

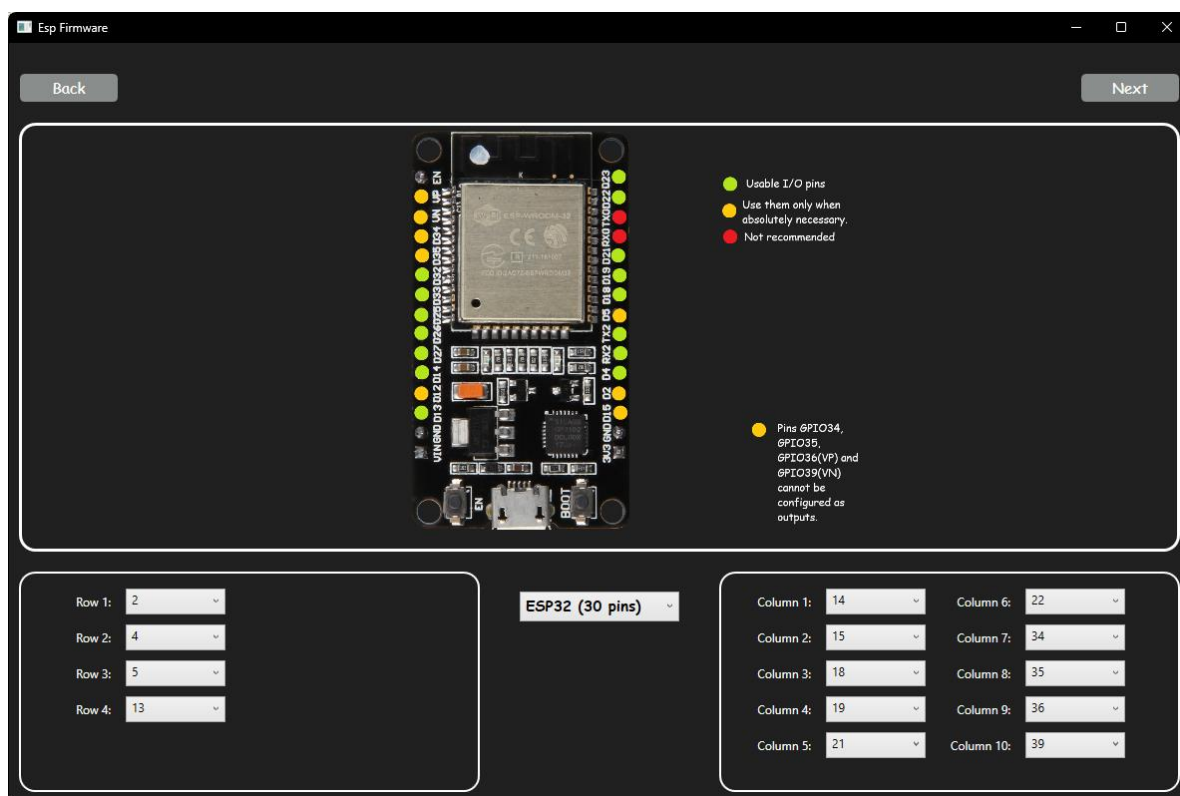


Рисунок 13 – Сторінка налаштування пінів мікроконтролера

Змін.	Арк.	№ докум.	Підпис	Дата

клавiші активації макросів M0()-M9()), клавiші для перемикання між шарами – TO(), TT(), MO()) та інші.

Введена розкладка відображається в окремому вікні, що дозволяє користувачу візуально оцінити результат і переконатися у правильності введених даних.

Далі користувач має можливість визначити до десяти макросів, їх налаштування не є обов'язковими і цей крок можна пропустити, якщо така функціональність не потрібна. Макроси дозволяють записати послідовності натискань клавiш, які потім можуть автоматично відтворюватися при натисканні призначеної клавiші.

Для запису макросу потрібно обрати відповідний номер макросу, та натиснути кнопку «Record Macros», після цього активується режим запису, під час якого інтерфейс буде тимчасово заблокований для уникнення сторонніх дій. А у текстовому полі з'явиться повідомлення про початок запису.

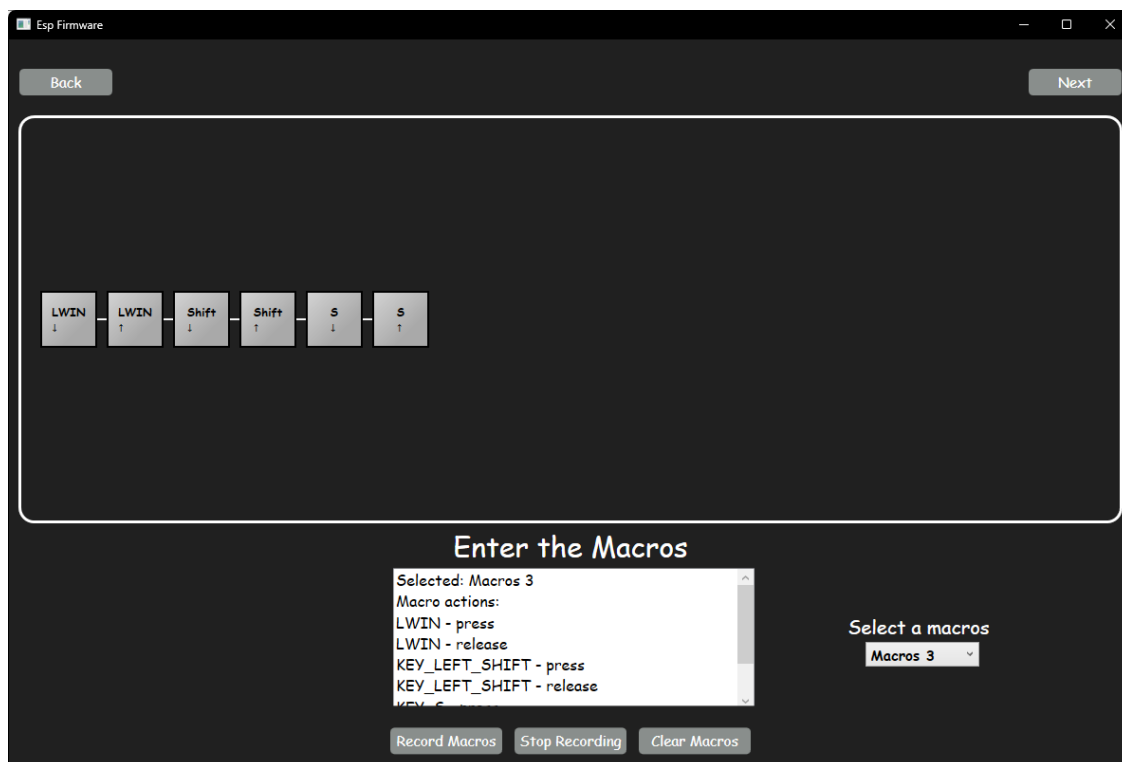


Рисунок 16 – Сторінка запису макросів

Для запису потрібно просто ввести бажану комбінацію клавiш на клавіатурі – усі натискання будуть автоматично збережені в порядку їх

Змін.	Арк.	№ докум.	Підпис	Дата

виконання. Коли послідовність завершено, потрібно натиснути кнопку «Stop Recording», після чого запис буде зупинено, а інтерфейс знову стане доступним для взаємодії. Якщо користувач бажає перезаписати макрос, для цього передбачена кнопка «Clear Macros», яка дозволяє стерти раніше записану послідовність і створити нову. Записана послідовність відразу ж буде відображена на екрані, щоб користувач міг її перевірити.

Останнім етапом конфігурації є створення прошивки. Перед початком компіляції користувачу обов'язково потрібно задати назву клавіатури в спеціальному полі «Keyboard Name», ця ім'я буде відображатися як назва пристрою при підключенні клавіатури.

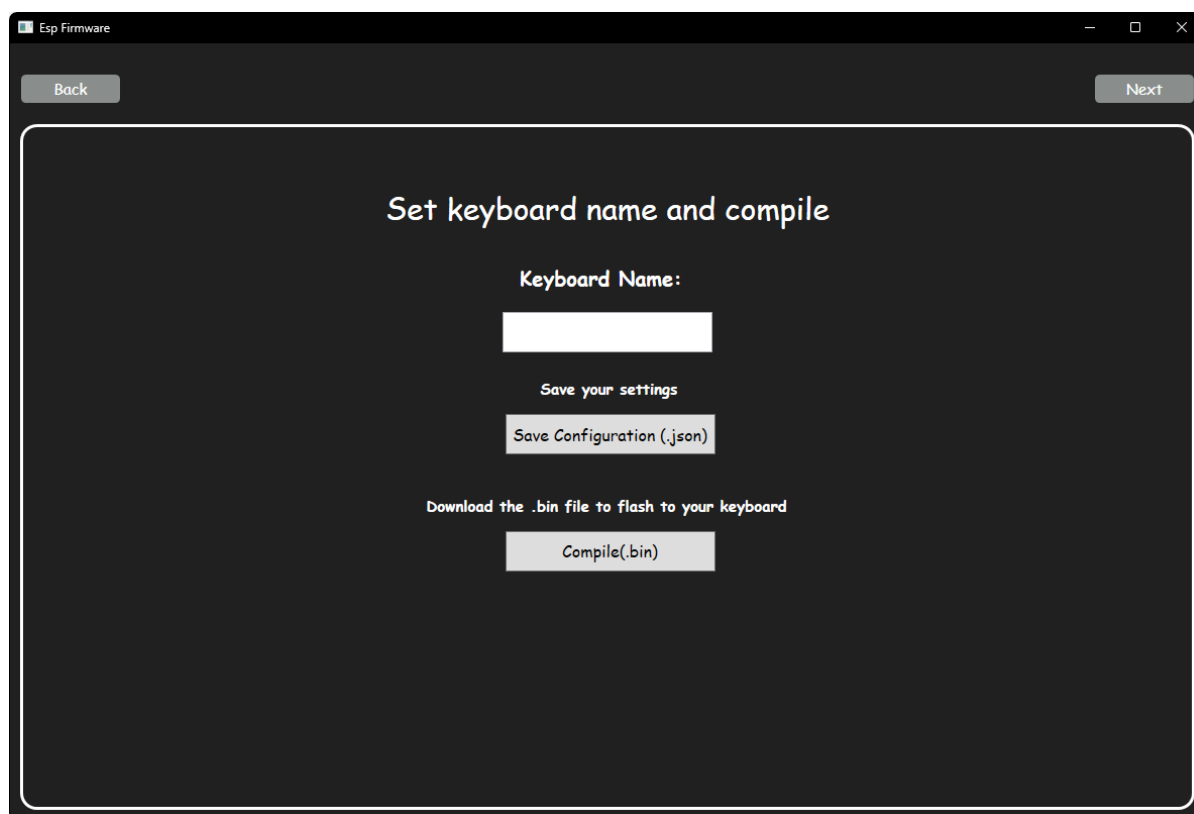


Рисунок 17 – Сторінка компіляції прошивки

Також, за потреби користувач може завантажити створену конфігурацію у форматі .json. Це може бути корисно як для резервного збереження налаштувань, так і для подальшої роботи з іншими конфігураторами або редакторами. Приклад структури .json файлу:

```
{  
  «KeyboardName»: «ESP-WROOM»,
```

Змін.	Арк.	№ докум.	Підпис	Дата

```

«Microcontroller»: «ESP32 (30 pins)»,
«Rows»: 3,
«Columns»: 3,
«RowPins»: [2, 4, 5],
«ColumnPins»: [12, 13, 14],
«LayersCount»: 4,
«Layouts»: [
  [
    [«KEY_Q», «KEY_W», «KEY_E»],
    [«KEY_A», «KEY_S», «KEY_D»],
    [«KEY_M0», «KEY_T2», «KEY_Z»]
  ],
  null,
  [
    [«KEY_1», «KEY_2», «KEY_3»],
    [«KEY_4», «KEY_5», «KEY_6»],
    [«KEY_M0», «KEY_T0», «KEY_0»]
  ],
  null
],
«Macros»: {
  «Macros 3»: [
    {«Key»: «LWIN», «Action»: «press»},
    {«Key»: «LWIN», «Action»: «release»},
    {«Key»: «KEY_LEFT_SHIFT», «Action»: «press»},
    {«Key»: «KEY_LEFT_SHIFT», «Action»: «release»},
    {«Key»: «KEY_S», «Action»: «press»},
    {«Key»: «KEY_S», «Action»: «release»}
  ],

```

Змін.	Арк.	№ докум.	Підпис	Дата

```

«Macros 5»: [
  {«Key»: «KEY_A», «Action»: «press»},
  {«Key»: «KEY_S», «Action»: «press»},
  {«Key»: «KEY_A», «Action»: "release"},
  {"Key": "KEY_S", "Action": "release"}
]
}
}

```

Для створення прошивки на основі заданої конфігурації користувач повинен натиснути кнопку «Compile (.bin)». Після цього одразу розпочинається процес компіляції прошивки, який здійснюється автоматично на основі введених параметрів. Після успішного завершення користувачеві буде запропоновано обрати місце для збереження згенерованого файлу з прошивкою у форматі .bin.

На цьому етапі процес конфігурації завершується, і користувач отримує готовий бінарний файл прошивки, який можна записати безпосередньо на мікроконтролер.

Для цього слід повернутися до головного меню програми та натиснути кнопку «Flash Device».

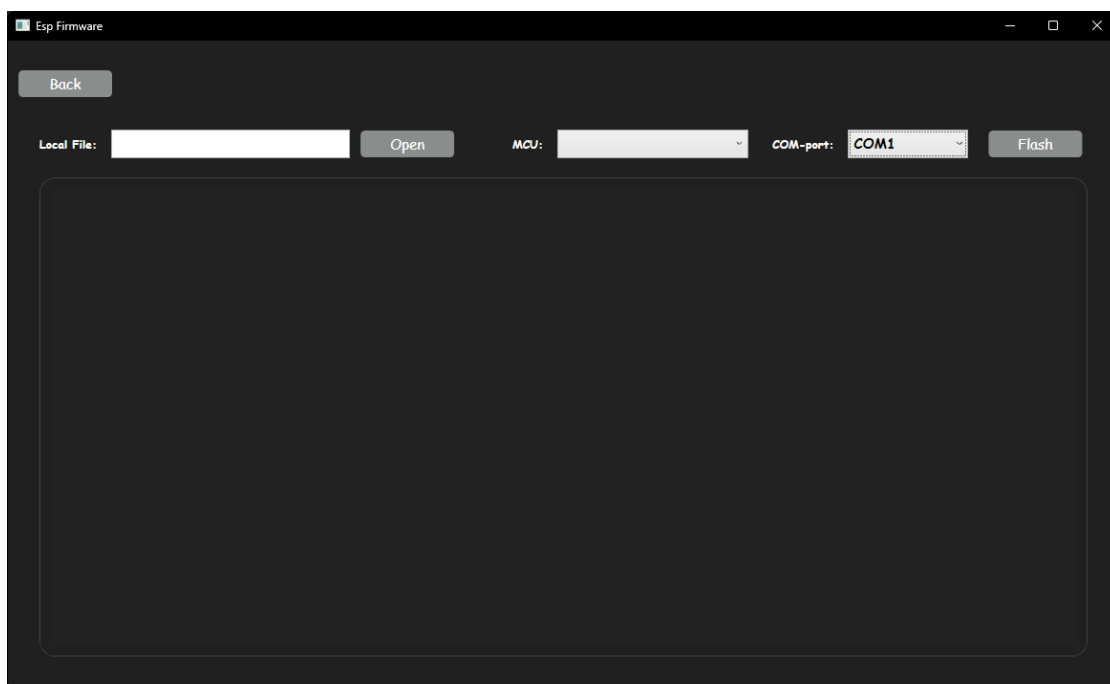


Рисунок 18 – Сторінка завантажувача

Змін.	Арк.	№ докум.	Підпис	Дата

Інтерфейс завантажувача поділена на дві основні області: керуюча панель та вікно виводу знизу неї. Керуюча панель складається з поля для вибору файлу прошивки, випадаюче меню для вибору моделі мікроконтролера, а також список доступних СОМ-портів.

При виборі файлу прошивки користувач може завантажити лише файл формату .bin, оскільки тільки бінарні файли використовуються для безпосереднього запису на мікроконтролер. Також користувач має обрати потрібну для прошивки модель мікроконтролера та СОМ-порт, до якого підключено пристрій. Система автоматично сканує всі доступні порти, тому після підключення плати відповідний СОМ-порт з'являється у списку автоматично. Коли усі параметри задані можна почати процес прошивки натиснувши на кнопку «Flash», esptool, який відповідає за завантаження, автоматично переводить мікроконтролер у режим завантаження та запускає передачу прошивки. Уся інформація про перебіг процесу виводить у вікно виводу.

Рисунок 19 – Процес прошивки плати

Після завершення прошивки користувач може підключитися до свого пристрою через Bluetooth.

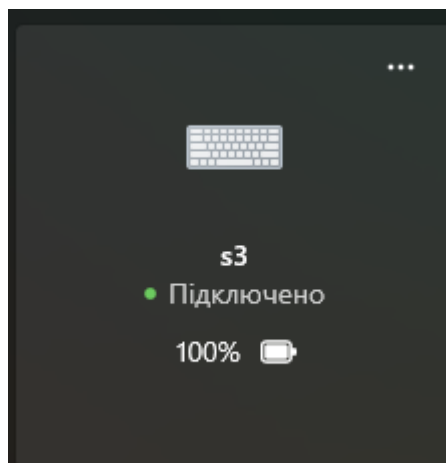


Рисунок 20 – Приєднаний пристрій

Для перевірки роботи клавіатури одразу після прошивки користувач може скористатися Keyboard Tester, перейшовши до нього з головного меню. На сторінці відображається повноцінна клавіатура з повним набором можливих клавіш. Усі клавіші мають візуальне представлення, і при натисканні фізичної клавіші на підключеній клавіатурі відповідна кнопка на екрані підсвічується – жовтим кольором при натисканні й зеленим при утриманні.

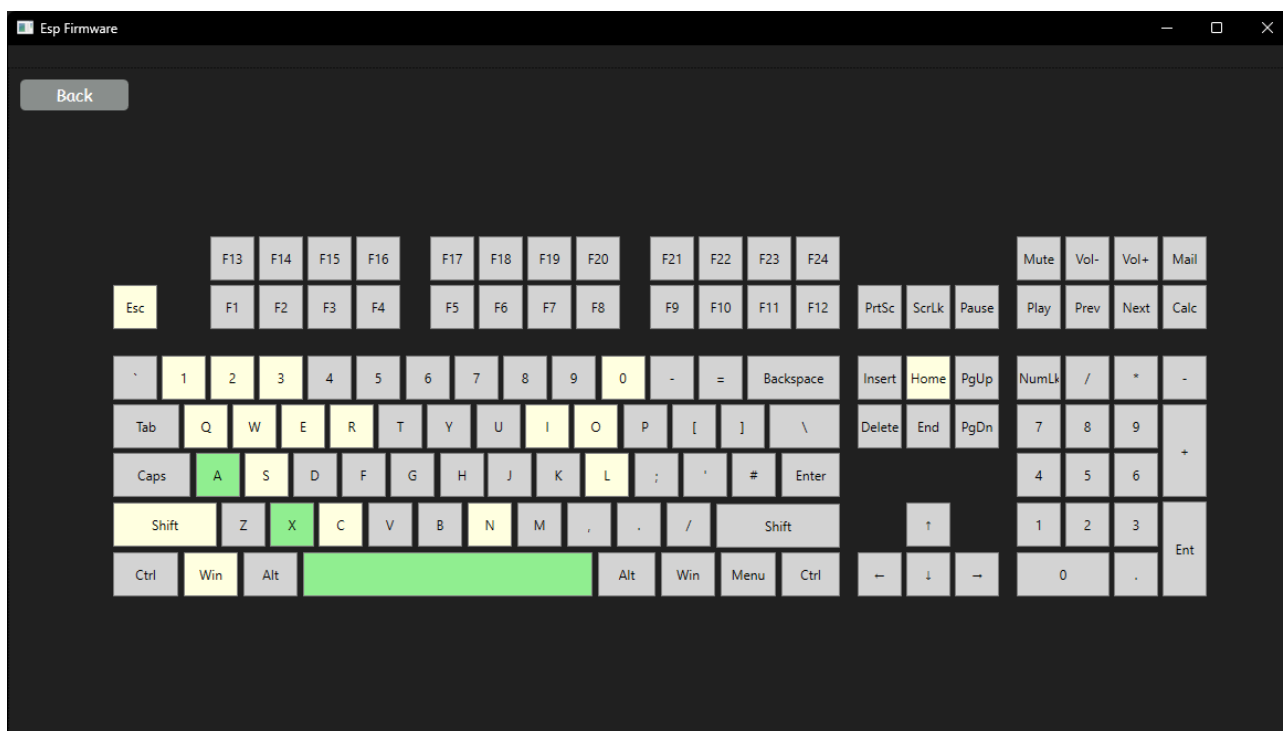


Рисунок 21 – Тестер натискань

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ. 467200.004 ПЗ

Арк.

49

Це дозволяє користувачу переконатися, що натискання коректно зчитуються мікроконтролером та відповідають заданій розкладці.

3.4 Можливості подальшого вдосконалення системи

Розроблена система має широкі можливості для подальшого вдосконалення. В першу чергу це розширення функціональних можливостей налаштування клавіатур:

- Підтримка більшої кількості шарів.
- Додавання таких функцій як Tap Dance, Combos, Key Overrides і тд.
- Підтримка поворотних енкодерів, екранами або іншими додатковими елементами, а також можливість їх простого налаштування через інтерфейс.

Також важливим є вдосконалення графічного інтерфейсу, для його покращення можлива реалізація інтерактивного редактора де користувач зможе вручну визначати положення клавіш, змінювати їх розміри та форму шляхом перетягування. Це дозволить враховувати індивідуальні форм-фактори клавіатур і надасть додаткову інформацію для створення 3D-моделі корпусу або інших його частин.

Для досвідчених користувачів, які потребують більших можливостей, доцільним буде додавання вбудованого командного рядка, що дозволить безпосередньо взаємодіяти з кодом прошивки та гнучко адаптувати її під специфічні вимоги.

Такі вдосконалення значно розширяють можливості системи та зроблять її більш привабливою і функціональною для широкого кола користувачів.

Змін.	Арк.	№ докум.	Підпис	Дата

ВИСНОВКИ

При виконанні дипломного проєкту була розроблена система налаштувань механічних клавіатур для мікроконтролерів ESP32. Система покликана усунути бар'єри, пов'язані зі складністю програмування та розробки DIY-клавіатур, надаючи користувачам зручний і зрозумілий інтерфейс для їх налаштування та прошивки.

Система має всі необхідні засоби для повного циклу розробки. Вона дає змогу задавати базові параметри клавіатури, такі як: розмір матриці, призначення пінів, створення розкладок клавіш та запис макросів, а також має засоби створення та завантаження прошивки. Для реалізації функціональності клавіатури розроблено спеціальну мікропрограму, яка генерується автоматично відповідно до параметрів, заданих користувачем. Компіляції відбувається у фоновому режимі за допомогою вбудованих інструментів, що забезпечує простоту та швидкість прошивки без необхідності встановлення сторонніх засобів або ручного втручання в код. Важливим також є процес завантаження прошивки на плату, з цією метою в систему інтегровано спеціальний завантажувач, побудований на основі інструменту esptool, що забезпечує швидке й надійне прошивання плати всього за кілька кліків.

Для перевірки працездатності пристрою після прошивки в систему було вбудовано спеціальний модуль тестування клавіш, який дозволяє швидко переконатися в коректності роботи пристрою без потреби у зовнішніх утилітах чи складному обладнанні.

Розроблена система має великий потенціал для подальшого вдосконалення, що дає змогу позиціонувати проєкт як універсальний інструмент для налаштування та програмування кастомних клавіатур, який зможе задовольнити потреби як новачків, так і досвідчених користувачів.

Отже розроблена система є ефективним інструментом для конфігурування та програмування механічних клавіатур. Вона поєднує інтуїтивно зрозумілий інтерфейс із широкими функціональними можливостями, що дозволяє

					<i>ІАЛЦ. 467200.004 ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		51

користувачам з різним рівнем підготовки швидко створювати, налаштовувати та перевіряти власні пристрої. Система усуває складність програмування, автоматизує всі етапи конфігурації та прошивки, а також забезпечує наочне тестування. Завдяки цьому вона не лише спрощує реалізацію DIY-проектів, але й відкриває нові можливості для розвитку індивідуальних рішень у сфері ввідних пристроїв, розширюючи потенціал використання мікроконтролерів ESP32 у користувацьких і професійних розробках.

					<i>ІАЛЦ. 467200.004 ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		52

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Quantum Mechanical Keyboard. URL: <https://docs.qmk.fm/> (Дата звернення: 10.03.2025)
2. QMK MSYS. URL: <https://msys.qmk.fm/> (Дата звернення: 10.03.2025)
3. Via. URL: https://caniusevia.com/docs/configuring_qmk (Дата звернення: 11.03.2025)
4. Vial. URL: <https://get.vial.today/> (Дата звернення: 11.03.2025)
5. Zephyr Mechanical Keyboard Firmware. URL: <https://zmk.dev/docs> (Дата звернення: 12.03.2025)
6. КМК: Clackety Keyboards Powered by Python. URL: https://github.com/KMKfw/kmk_firmware/tree/main/docs/en (Дата звернення: 13.03.2025)
7. POG. URL: <https://pog.heaper.de/> (Дата звернення: 13.03.2025)
8. ESP SOCs | Espressif Systems. URL: <https://www.espressif.com/en/products/socs> (Дата звернення: 14.03.2025)
9. Rust keyboard firmware library. URL: <https://haobogu.github.io/rmk/> (Дата звернення: 14.03.2025)
10. MK32 - Simple BLE keyboard for ESP32. URL: <https://github.com/Galzai/MK32/wiki> (Дата звернення: 14.03.2025)
11. A tour of the C# language. URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/overview> (Дата звернення: 24.04.2025)
12. Desktop Guide (WPF .NET). URL: <https://learn.microsoft.com/en-gb/dotnet/desktop/wpf/overview/?view=netdesktop-9.0> (Дата звернення: 24.04.2025)
13. Visual Studio. URL: <https://visualstudio.microsoft.com/vs/> (Дата звернення: 24.04.2025)
14. About Git. URL: <https://git-scm.com/about> (Дата звернення: 25.04.2025)
15. About GitHub. URL: <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git> (Дата звернення: 25.04.2025)

16.Arduino CLI. URL: <https://docs.arduino.cc/arduino-cli/> (Дата звернення:
26.04.2025)

17.Esptool.py Documentation. URL: <https://docs.espressif.com/projects/esptool/>
(Дата звернення: 26.04.2025)

					<i>ІАЛЦ. 467200.004 ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		54

Додаток А

Додаток В