

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет

імені Петра Могили

Факультет комп'ютерних наук

Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри,
д-р техн. наук, проф.

_____ І. М. Журавська

« __ » _____ 2024 р.

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

Інтерактивне табло на адресних світлодіодах

Спеціальність 123 Комп'ютерна інженерія

123 – КБР.ПЗ.00 – 405.22010517

Студент

 В. В. Петіков
підпис

« __ » _____ 202__ р.

Керівник ст. викладач

_____ Б. Г. Салтовський
підпис

« __ » _____ 202__ р.

Миколаїв – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Зав. кафедри _____ І. М. Журавська

« _____ » _____ 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної бакалаврської роботи

Видано студенту групи 405 факультету комп'ютерних наук

Петікову Володимирі Вікторовичу
(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Інтерактивне табло на адресних світлодіодах

Затверджена наказом по ЧНУ від 30.01.2024 №17.

2. Строк представлення кваліфікаційної роботи « _____ » _____ 20__ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Прилад має отримувати данні про тивоги через API, після цього лунає сигнал динаміка та загорається адресний світлодіод в залежності від типу тривоги. На кожен тип тривоги виділено окремий світлодіод який загорається своїм окремим кольором.

4. Перелік питань, що підлягають розробці

Всебічний огляд апаратних платформ. Опис та аналіз контролера ESP32, адресних світлодіодів, звукової плати, модуля живлення і API UkraineAlarm. Вивчення алгоритмів та методів інтеграції. Створення моделі прототипу. Розробка макетних і принципових схем. Складання та монтаж прототипу. Налаштування і оптимізація апаратної платформи. Розробка програмного забезпечення для пристрою. Спеціальний розділ з охорони праці. Висновки роботи. Перелік використаних джерел.

5. Перелік графічних матеріалів

Зображення звукової плати, модуля живлення, адресних світлодіодів.

Зображення схеми пінів контролера ESP32 CH340 Type-C.

Зображення макетної та принципової схеми підключення компонентів.

Зображення функціонування програмно-апаратного комплексу.

6. Завдання до спеціальної частини

Розглянути основні методи ідентифікації аварійних ситуацій за допомогою адресних світлодіодів. Зробити розрахунки оптимальної кількості адресних світлодіодів для захисних сховищ. Опрацювати рекомендації що до впровадження інтерактивних табло на адресних світлодіодах в захисні сховища. Описати основні навички та інструктаж персоналу з використання нових технологій.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
ст. викладач О. В. Макарова	кафедра екології Медичного інституту ЧНУ ім. Петра Могили	Спеціальна частина з охорони праці

Керівник роботи

Ст. викл. каф. комп. інженерії Салтовський Борис Григорович

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Петіков Володимир Вікторович

(прізвище, ім'я, по батькові студента)



(підпис)

Дата видачі завдання « ____ » _____ 20 ____ р.

КАЛЕНДАРНИЙ ПЛАН

Виконання кваліфікаційної роботи

Тема: Інтерактивне табло на адресних світлодіодах

№	Найменування роботи	Початок	Закінчення	Примітки
	Розробка та затвердження завдання на виконання КБР			Виконав
	Огляд літератури за темою роботи			Виконав
	Складання календарного плану КБР			Виконав
	Аналіз предметної області			Виконав
	Аналіз існуючих технічних рішень			Виконав
	Підбір необхідних для реалізації системи компонентів			Виконав
	Проектування прототипу системи оповіщення на базі ESP-32			Виконав
	Реалізація програмної частини системи оповіщення на базі ESP-32			Виконав
	Тестування та відлагодження отриманого прототипу. Аналіз результатів			Виконав
	Оформлення результатів дослідження			Виконав
	Підготовка презентації			Виконав
	Попередній захист		0	Виконав
	Рецензування			Виконав
	Завершення оформлення КБР та презентації	6 6	8 6	Виконав
	Захист кваліфікаційної роботи			Виконав

Розробив здобувач ВО Петіков Володимир Вікторович _____

(прізвище, ім'я, по батькові)

(підпис)

«__» _____ 20__ р.

Керівник роботи Ст. викл. каф. комп. інженерії Б. Г. Салтовський _____

(підпис)

«__» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи
«Інтерактивне табло на адресних світлодіодах»

Студент: Петіков Володимир Вікторович

Керівник: Ст. викл. каф. комп. інженерії Салтовський Б. Г.

Бакалаврська робота присвячена розробці інтерактивного табло на адресних світлодіодах для оснащення захисних сховищ на основі ESP32. Розглянуто наявні в наш час методи та способи проєктування систем з використанням мікроконтролера ESP32. Практичне значення отриманих результатів полягає у дослідженні особливостей створення та реалізації ефективного приладу-інформатора на основі ESP32, адресних світлодіодів аудіоплати з динаміком, модулем живлення та інтеграції їх з API.

Пояснювальна записка бакалаврської роботи складається зі вступу, чотирьох розділів, висновків та додатків. У вступі визначається актуальність теми, сформульовані мета, об'єкт, предмет та завдання дослідження та розроблення бакалаврської роботи. У першому розділі досліджуються різноманітні технології сповіщення про небезпеку та їх використання, що є об'єктом дослідження; проводиться аналіз існуючих прототипів і технічних рішень систем з використанням ESP32. У другому розділі наведені дані про підходи до проєктування системи та алгоритм її створення. У третьому розділі наведені дані про реалізацію програмно-апаратного комплексу та результати тестування. Четвертий розділ присвячений охороні праці в контексті інтерактивних табло на адресних світлодіодах. У висновках наведено аналіз виконаної роботи та отриманих результатів дослідження та розроблення.

У додатку А наведена довідка про унікальність, а в Б наведено програмний код, що використовувався в проєкті. В цілому, бакалаврська робота без додатків містить 71 сторінки, 39 рисунків, 2 таблиці, 29 джерел посилання.

Ключові слова: інтерактивне табло, адресні світлодіоди, ESP32, звукова плата, програмно-апаратний комплекс, API.

ABSTRACT

of the Bachelor's Thesis
«Interactive scoreboard on addressable LEDs»

Student: Petikov Volodymyr

Consultant: Senior Lecturer of the Department of Computer Engineering, B.G.
Saltovsky.

The bachelor's thesis is dedicated to the development of an interactive display using addressable LEDs for equipping protective shelters based on the ESP32. The existing methods and approaches to designing systems using the ESP32 microcontroller are examined. The practical significance of the obtained results lies in the study of the features of creating and implementing an effective information device based on the ESP32, addressable LEDs, an audio board with a speaker, a power module, and their integration with APIs.

The explanatory note of the bachelor's thesis consists of an introduction, four chapters, conclusions, and appendices. The introduction outlines the relevance of the topic, formulates the aim, object, subject, and objectives of the research and development of the bachelor's thesis. The first chapter investigates various danger notification technologies and their applications, which is the object of the study; it analyzes existing prototypes and technical solutions of systems using the ESP32. The second chapter provides data on approaches to system design and the algorithm for its creation. The third chapter presents data on the implementation of the hardware-software complex and the results of testing. The fourth chapter is devoted to occupational safety in the context of interactive displays with addressable LEDs. The conclusions provide an analysis of the completed work and the obtained research and development results.

In Annex A contains a uniqueness report, and Appendix B provides the software code used in the project. In total, the bachelor's thesis, excluding appendices, contains 71 pages, 39 figures, 2 tables, 29 references sources.

Keywords: interactive display board, addressable LEDs, ESP32, audio board, hardware-software complex, API.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП	5
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ НАДАННЯ ІНФАРМАЦІЇ ПРО ТРИВОГУ ТА ОГЛЯД ІСНУЮЧИХ ПРОТОТИПІВ	7
1.1 Використання системи сповіщення повітряної тривоги.....	9
1.2 Способи сповіщення про повітряні тривоги	10
1.3 Аналіз існуючих систем, прототипів	15
1.4 Висновки до розділу 1	24
РОЗДІЛ 2 МОДЕЛЮВАННЯ ТА РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО МОДУЛЮ	25
2.1 Методи проектування системи та процес її розробки.....	25
2.2 Обґрунтування причин вибору програмно-апаратних засобів для управління системою оповіщень.....	27
2.3 Вибір адресних світлодіодів	39
2.4 Вибір звукової плати.....	41
2.5 Живлення пристрою	43
2.6 Програмне забезпечення	45
2.7 UkraineAlarm API	48
2.8 Висновки до розділу 2	49
РОЗДІЛ 3 АПАРАТНА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЄКТУ НА ESP- 32, АДРЕСНИХ СВІТЛОДІОДІВ, ЗВУКОВОЇ ПЛАТИ ТА API.....	50
3.1 Під'єднання усіх компонентів пристрою.....	50
3.2 Використання UkraineAlarm API.....	54
3.3 Налаштування Arduino IDE.....	55
3.4 Завантаження коду на ESP32	62
3.5 Опис програмної частини приладу	63
3.6 Можливі модифікації приладу.....	65
3.7 Висновки до розділу 3	66
ВИСНОВКИ.....	68

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	69
ДОДАТОК А ДОВІДКА ПРО ПЕРЕВІРКУ НА УНІКАЛЬНІСТЬ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ	72
ДОДАТОК Б КОД ДЛЯ ІНТЕРАКТИВНОГО ТАБЛА НА АДРЕСНИХ СВІТЛОДІОДАХ.....	73

ПЕРЕЛІК СКОРОЧЕНЬ

ADC	– Analog-to-digital converter
API	– Application programming interface
DAC	– Digital to analog converter
GPIO	– General-purpose input/output
GSM	– Global System for Mobile Communications
I2C	– Inter-Integrated Circuit
IDE	– Integrated Development Environment
LED	– Light-emitting diode
NTP	– Network Time Protocol
OLED	– Organic light-emitting diode
PWM	– Pulse-width modulation
RGB	– Red Green Blue
RTC	– Real-Time Clock
SMS	– Short Message Service
SoC	– System-on-a-Chip
SPI	– Serial Peripheral Interface
SPI	– Serial Peripheral Interface
UART	– Universal asynchronous receiver/transmitter
UDP	– User Datagram Protocol
URL	– Uniform Resource Locator
WiFi	– Wireless Fidelity

ВСТУП

Актуальність дослідження. Актуальність теми кваліфікаційної роботи, а саме інтерактивне табло на адресних світлодіодах для оснащення захисних сховищ виражається в тому, що в сучасному світі зростає значення безпеки та ефективного управління тривожними ситуаціями. Застосування технології адресних світлодіодів та інтерактивних табло для цієї мети дозволяє створювати ефективні пристрої для оперативного сповіщення про аварійні стани, що є критичним у випадках екстрених ситуацій.

Технічний прогрес у сфері світлодіодних технологій та бездротового зв'язку дозволяє реалізувати цю ідею на практиці, створюючи недорогі та ефективні засоби для інформування персоналу та громадськості про початок, тип і закінчення тривоги в реальному часі. Забезпечення безпеки та швидкого реагування на аварійні ситуації стає важливим завданням у сучасному світі, тому розробка таких інноваційних пристроїв має значний практичний і соціальний вплив.

Мета та завдання дослідження:

Мета даної кваліфікаційної роботи полягає у розробці та впровадженні інтерактивного табло на адресних світлодіодах для оснащення захисних сховищ, з метою швидкого та ефективного сповіщення про початок, тип і закінчення тривоги, забезпечуючи безпеку та ефективність управління аварійними ситуаціями.

Завдання: Провести аналіз існуючих систем оповіщення про тривогу та на прикладах вже існуючих екземплярів, змоделювати свою версію. Проаналізувати методи та способи отримання інформації про тривоги для оснащення свого пристрою. На основі всього цього зробити свій функціонуючий пристрій.

Об'єкт дослідження: Технологія оповіщення про небезпеку через інтерактивне табло на адресних світлодіодах.

Предмет дослідження: Розробка пристрою-інформера для оснащення захисних сховищ, який за допомогою інформаційного табло та звукових

сигналів повідомлятиме про початок, тип і закінчення тривоги в реальному часі.

Практичне значення результатів дослідження та розробки:

Результати дослідження та розроблення такого пристрою мають велике практичне значення для забезпечення безпеки в захисних сховищах та інших об'єктах. Цей пристрій допоможе оперативно та ефективно інформувати персонал та мешканців про тривожні ситуації, забезпечуючи швидку реакцію на них та мінімізацію можливих наслідків. Таке рішення дозволить створювати недорогі, не великі за розміром та ефективні пристрої для інформування громадян про тип та наявність небезпек у сховищах та інших об'єктах, що має важливе значення для загальної безпеки та захисту населення.

РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ НАДАННЯ ІНФАРМАЦІЇ ПРО ТИВОГУ ТА ОГЛЯД ІСНУЮЧИХ ПРОТОТИПІВ

Системи сповіщення повітряної тривоги є важливими елементами національної безпеки, забезпечуючи оперативне інформування населення про небезпеку повітряних нападів. Ці системи використовують різноманітні технічні рішення для максимального охоплення і своєчасного попередження громадян.

Сучасні системи сповіщення повітряної тривоги базуються на технологіях, які дозволяють забезпечити високу надійність і швидкість передачі сигналів тривоги. Огляд літератури показує, що існує кілька основних підходів до реалізації таких систем: традиційні сирени, радіопередавачі, мобільні додатки та інтегровані системи оповіщення через інтернет-протоколи.

Традиційні системи оповіщення включають використання гучних сирен, розміщених у стратегічних точках міста або населених пунктів. Ці системи мають перевагу у високій надійності та простоті обслуговування. Вони здатні передавати звукові сигнали, які можуть бути почуті на великій відстані, що дозволяє ефективно охопити значні території.

З розвитком технологій, системи оповіщення повітряної тривоги почали інтегрувати нові технічні рішення. Наприклад, використання радіопередавачів дозволяє передавати сигнали тривоги безпосередньо на радіоприймачі громадян. Також, широкого розповсюдження набули мобільні додатки, які дозволяють отримувати сповіщення про повітряну тривогу через смартфони та інші мобільні пристрої. Такі додатки можуть використовувати GPS для визначення місцезнаходження користувача та надавати точну інформацію про рівень загрози у конкретному регіоні.

Патентний аналіз показує, що існує значна кількість запатентованих технологій, спрямованих на підвищення ефективності систем сповіщення повітряної тривоги. Наприклад, патенти на технології інтеграції систем оповіщення з мережами мобільного зв'язку дозволяють передавати сигнали

тривоги безпосередньо на мобільні телефони користувачів. Інші патенти зосереджуються на удосконаленні апаратних компонентів, таких як нові види сирен або датчиків, здатних виявляти загрози більш точно і швидко.

Перспективи розвитку систем сповіщення повітряної тривоги пов'язані з подальшою інтеграцією сучасних технологій. Використання Інтернету речей (IoT) може значно підвищити ефективність таких систем. Наприклад, датчики, розміщені у різних точках міста, можуть автоматично виявляти загрози і передавати сигнали тривоги на центральний сервер. Цей сервер, у свою чергу, може миттєво інформувати населення через різноманітні канали зв'язку, включаючи мобільні додатки, соціальні мережі та інші засоби комунікації.

Системи сповіщення повітряної тривоги є критично важливими для забезпечення безпеки населення. Огляд літератури та патентний аналіз показують, що існує багато перспективних напрямків розвитку цих систем. Інтеграція сучасних технологій, таких як мобільні додатки та Інтернет речей, може значно підвищити їх ефективність та надійність, що особливо важливо в умовах зростаючих загроз повітряних нападів.

Сучасні методи надання інформації про повітряну тривогу охоплюють різноманітні технології та підходи. Вони мають на меті забезпечити швидке та ефективне оповіщення населення про небезпеку, використовуючи як традиційні, так і новітні технологічні рішення.

В даний час існують різні види систем попередження про повітряні загрози, які використовуються для швидкого надання людям інформації та забезпечення їхньої безпеки, ось деякі з цих прототипів:

- Системи які автоматично виявляють газ у повітрі: Ці системи можуть сказати вам, чи є в повітрі газ, який може бути шкідливий для вас. Наприклад, існують системи, які можуть виявляти викиди CO₂, газоподібного амонію та інших шкідливих речовин.

- Системи мережевого спостереження: Це системи, які використовують безліч датчиків та пристроїв для спостереження за ефіром та

відправлення попереджень, коли щось змінюється. Ці системи можна комбінувати із системами управління аваріями та пожежної безпеки.

– Системи моніторингу якості повітря: Вони використовуються для спостереження за повітрям та виявлення будь-яких змін, які можуть означати щось погане, наприклад збільшення забруднення або ризику пожежі.

– Мобільні програми та інтерфейси: Деякі прототипи ґрунтуються на додатках, які дозволяють користувачам отримувати інформацію про ефір, при необхідності подавати сигнал тривоги та щось із цим робити.

Ці прототипи використовуються у різних галузях, таких як промисловість, медицина, комунальне господарство та інших, щоб забезпечити безпеку людей та можливість швидко відреагувати у разі небезпеки. Люди почали більше використовувати ці прототипи в різних сферах, так як вони більш дешеві в розробці та моделюванні ніж вже представлені на ринку екземпляри.

1.1 Використання системи сповіщення повітряної тривоги

Система повітряного сповіщення є ключовим інструментом безпеки людей під час обстрілів та інших надзвичайних ситуацій. Якщо ви використовуєте її правильно, це дійсно може знизити ймовірність того, що люди та речі будуть поранені чи пошкоджені.

Коли система повітряної тривоги виявляє небезпеку, вона відразу надсилає сигнали, що дозволяє людям швидко зробити необхідні кроки для забезпечення безпеки себе та своїх близьких. Ці системи можуть надсилати оповіщення різними способами, наприклад, розмовляти гучномовцем, надсилати текстові повідомлення на телефон, радіо або телевізор.

Життя завжди в небезпеці, тому нам потрібно більше систем повітряного сповіщення. Система передбачає не тільки сповіщення людей про небезпеку, а й розгляд різних ситуацій та забезпечення того, щоб ми добре на них реагували.

Наприклад, ви можете налаштувати системи, які зможуть автоматично включати пожежну сигналізацію, аварійне освітлення або інші заходи безпеки, щоб допомогти людям швидше та краще реагувати на небезпеку.

Сучасні технології дозволяють з'єднувати системи повітряного сповіщення з іншими безпековими системами, наприклад, з моніторингом об'єктів або автоматизованими системами управління будівлею. Це надає повний спосіб забезпечити безпеку речей і допомагає нам швидше та краще реагувати на небезпеку.

Всі ці моменти показують, наскільки важливо створювати та вдосконалювати системи повітряного сповіщення у світі. Слідкувати за технічними оновленнями, розповсюджувати інформацію та працювати з поліцейськими та фахівцями з безпеки — все це важливі частини цього.

1.2 Способи сповіщення про повітряні тривоги

1.2.1 Візуальне оповіщення

Візуальна демонстрація - це один із видів передачі інформації про початок та кінець повітряних тривог, який базується на використанні світлових сигналів або візуальних засобів сповіщення.

Основні елементи цієї системи включають світлові маячки (рис. 1.1), які розміщуються на видимих місцях, щоб світити або миготіти певним кольором і привертати увагу громадян. Світлові сигнали відносяться до способів передачі інформації за допомогою світлових ефектів. Це може бути основним засобом сповіщення про різноманітні події, включаючи повітряні тривоги.



Рисунок 1.1 – Світловий маячок

Це спеціальні пристрої, які розміщуються на видимих місцях, таких як дахи будівель, вежі, транспортні засоби тощо, і світяться або миготять певним кольором, щоб привернути увагу і повідомити про надзвичайну ситуацію, в тому числі повітряну тривогу.

Для більш компактних систем сповіщення про небезпеку використовуються світлодіоди, а саме адресні світлодіоди WS2812 (рис. 1.2) з яких можна робити цілі стрічки з виводом свого кольору на окремий світлодіод.

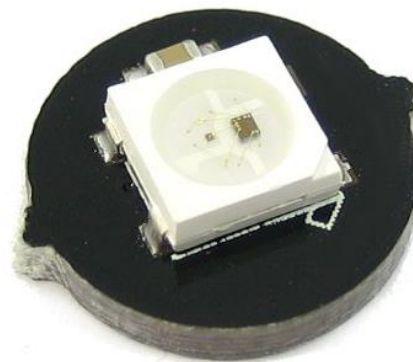


Рисунок 1.2 – Адресний світлодіод WS2812

Стрічки з адресними світлодіодами можна використовувати як суцільне табло[1][3] з виводом графічної інформації, такої як текст або

суцільну карту України. Наприклад, світлодіодні стрічки з адресними світлодіодами WS2812 стали популярними для створення компактних систем сповіщення про небезпеку. Такі стрічки можуть відображати різні кольори і навіть текстову або графічну інформацію. Наприклад, на (рис. 1.3) ви бачите карту України, яка відтворюється на адресних світлодіодах.



Рисунок 1.3 – Карта України на адресних світлодіодах

Щоб зробити систему сповіщення більш інтерактивною, світлодіоди можна розмістити на готовому макеті карти, де кожен світлодіод відіграє роль маркера, що змінює свій колір в області, де почалась тривога (рис. 1.4). Такий підхід дозволяє швидко та чітко передавати інформацію про надзвичайні ситуації та реагувати на них.



Рисунок 1.4 – Карта тревог України

Загалом, технології візуальної демонстрації на основі світлових сигналів постійно розвиваються, забезпечуючи ефективні та інноваційні методи сповіщення про небезпеку і покращуючи безпеку громадян у разі надзвичайних ситуацій.

1.2.2 Звукове оповіщення

Звукове оповіщення - це ще один спосіб передачі інформації про початок та кінець повітряних тривог. Цей метод базується на використанні звукових сигналів та акустичних засобів сповіщення. Основні елементи звукового оповіщень включають:

– Сирени і дзвони: У багатьох містах та на промислових об'єктах встановлені сирени (рис. 1.5) або дзвони, які активуються під час тривоги [2]. Ці звукові сигнали можуть мати різний тон та інтенсивність в залежності від типу надзвичайної ситуації.



Рисунок 1.5 – Сирена повітряної тривоги С-40К

– Гучномовці: На вулицях, в громадських місцях або в будівлях можуть бути встановлені гучномовці (рис. 1.6), через які розповсюджуються голосові повідомлення про тривогу та інструкції щодо дій громадян.



Рисунок 1.6 – Гучномовець

– Аудіо системи в транспорті: У громадському транспорті також можуть бути встановлені аудіо системи, які автоматично відтворюють звукові повідомлення про надзвичайні ситуації.

– Мобільні додатки та SMS-повідомлення: Деякі системи тривоги також можуть використовувати мобільні додатки або надсилати SMS-повідомлення громадянам з інформацією про тривогу та інструкціями. Доне з таких застосунків “Air Alert!” (рис. 1.7). Це застосунок яких сповіщує користувача про початок тривоги та кінець. Він має голосові повідомлення для кращого повідомлення ситуації. Цей застосунок є як і у мобільному додатку, так і у вигляді сайту. Також він має можливість інформувати про такі види загроз як: повітряна загроза, загроза артилерії, хімічна тривога, радіаційна небезпека та вулечні бої.

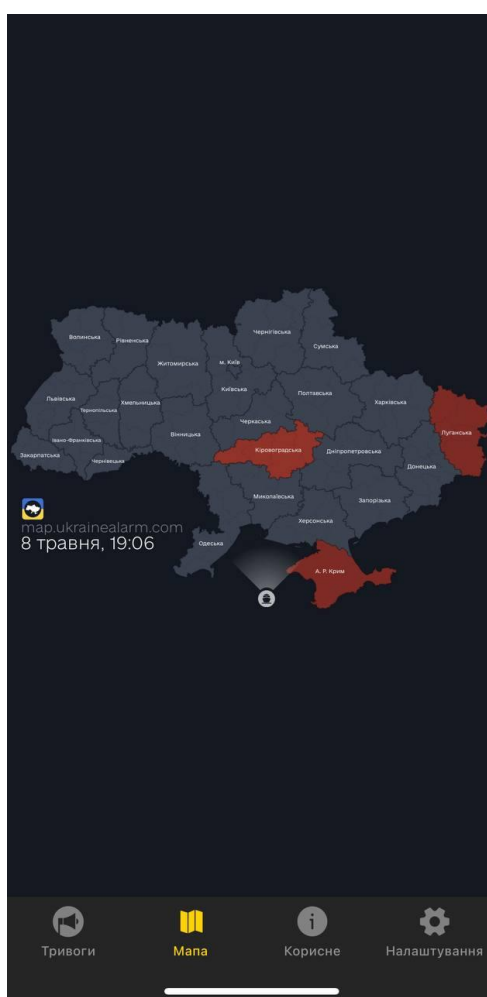


Рисунок 1.7 – Air Alert!

1.3 Аналіз існуючих систем, прототипів

В сучасному світі, де технології швидко розвиваються, важливо постійно аналізувати існуючі системи та прототипи для вдосконалення та

створення нових рішень. Особливо це стосується сфери безпеки та управління аварійними ситуаціями. З метою покращення швидкості та ефективності сповіщення про надзвичайні події, у даному дослідженні проводиться аналіз існуючих систем та прототипів інтерактивних табло на адресних світлодіодах. Цей аналіз є необхідним етапом у розробці та впровадженні нової системи, що спрямована на підвищення рівня безпеки та ефективності управління в надзвичайних ситуаціях.

1.3.1 Мапа тривоги на базі ESP32

Один з вже існуючих прототипів систем сповіщення тривоги є «Мапа тривоги на базі ESP32» [7]. Мапа тривоги на базі ESP32 є одним із ключових елементів інтерактивного табло на адресних світлодіодах, яке представлено у цій роботі. Цей проект поєднує в собі сучасні технології мікроконтролерів, адресних світлодіодів та системи сповіщення тривоги для створення інтерактивного та інформативного пристрою.

Давайте детально розглянемо процес створення мапи тривоги на базі ESP32. Цей процес включає у себе вибір та підготовку необхідних компонентів, програмування мікроконтролера, налаштування з'єднання та завантаження програмного забезпечення.

Розглянемо компоненти які використовуються у цій роботі:

ESP32 (ESP-WROOM-32): Це мікроконтролер з вбудованим Wi-Fi і Bluetooth, який використовується для керування світлодіодною адресною стрічкою та іншими компонентами системи (рис. 1.8).

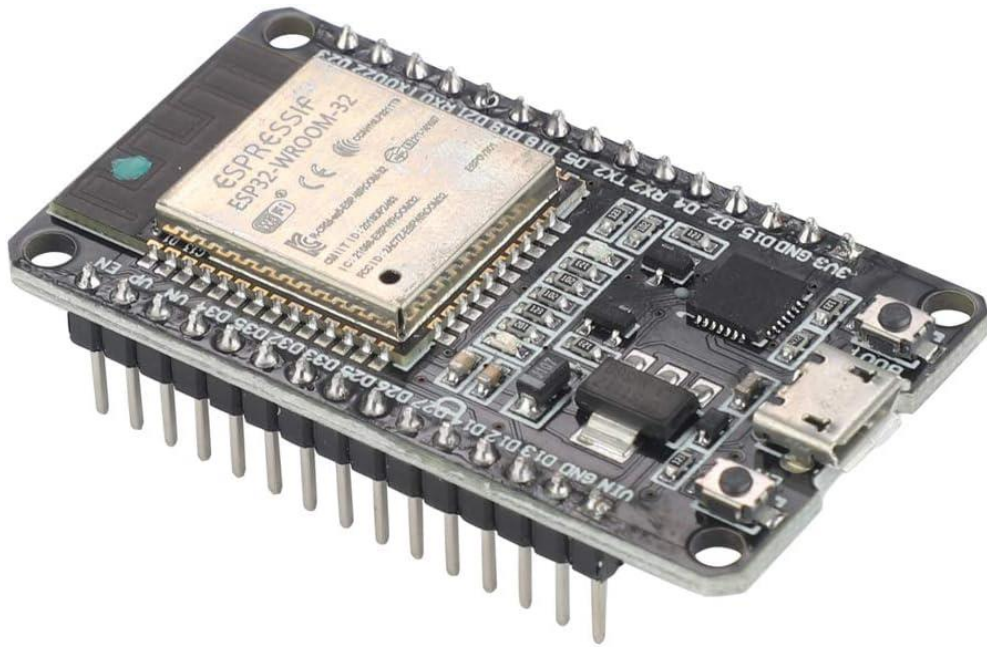


Рисунок 1.8 – ESP32 (ESP-WROOM-32)

Адресна LED стрічка: Світлодіодна стрічка, а саме WS2812B RGB (рис. 1.9) у якій кожен світлодіод має свою унікальну адресу. Це дозволяє керувати окремими світлодіодами та створювати різноманітні ефекти та зображення.



Рисунок 1.9 – Адресна LED стрічка WS2812B RGB

Резистор (опціонально): Резистор може використовуватися для стабілізації напруги або для інших цілей у системі, які вимагають обмеження струму.

Дисплей (опціонально): Додатковий елемент для відображення інформації або статусу системи. Наприклад, OLED дисплей може використовуватися для відображення тексту, зображень або анімацій.

Акумулятор та зарядка (опціонально): Акумулятор і зарядка дозволяють системі працювати незалежно від джерела живлення. Це може бути корисно для мобільних додатків або у разі відключення живлення.

Підвищувач напруги (опціонально): Якщо потрібно живити компоненти, які вимагають вищої напруги, такі як дисплей або інші пристрої, може використовуватися підвищувач напруги.

Динамік (опціонально): Додатковий звуковий елемент, який може використовуватися для візуально-аудіо тривоги або повідомлення користувачу про події у системі.

Вимикач (опціонально): Вимикач може використовуватися для управління живленням або іншими функціями системи.

Ці компоненти можуть бути використані окремо або в поєднанні залежно від потреб і вимог проєкту. Наприклад, у мінімальній конфігурації може бути лише ESP32 та адресна LED стрічка, тоді як у максимальній конфігурації використовуються всі перераховані компоненти для більш розширених функцій та можливостей системи.

Схема підключення компонентів виглядає наступним чином:

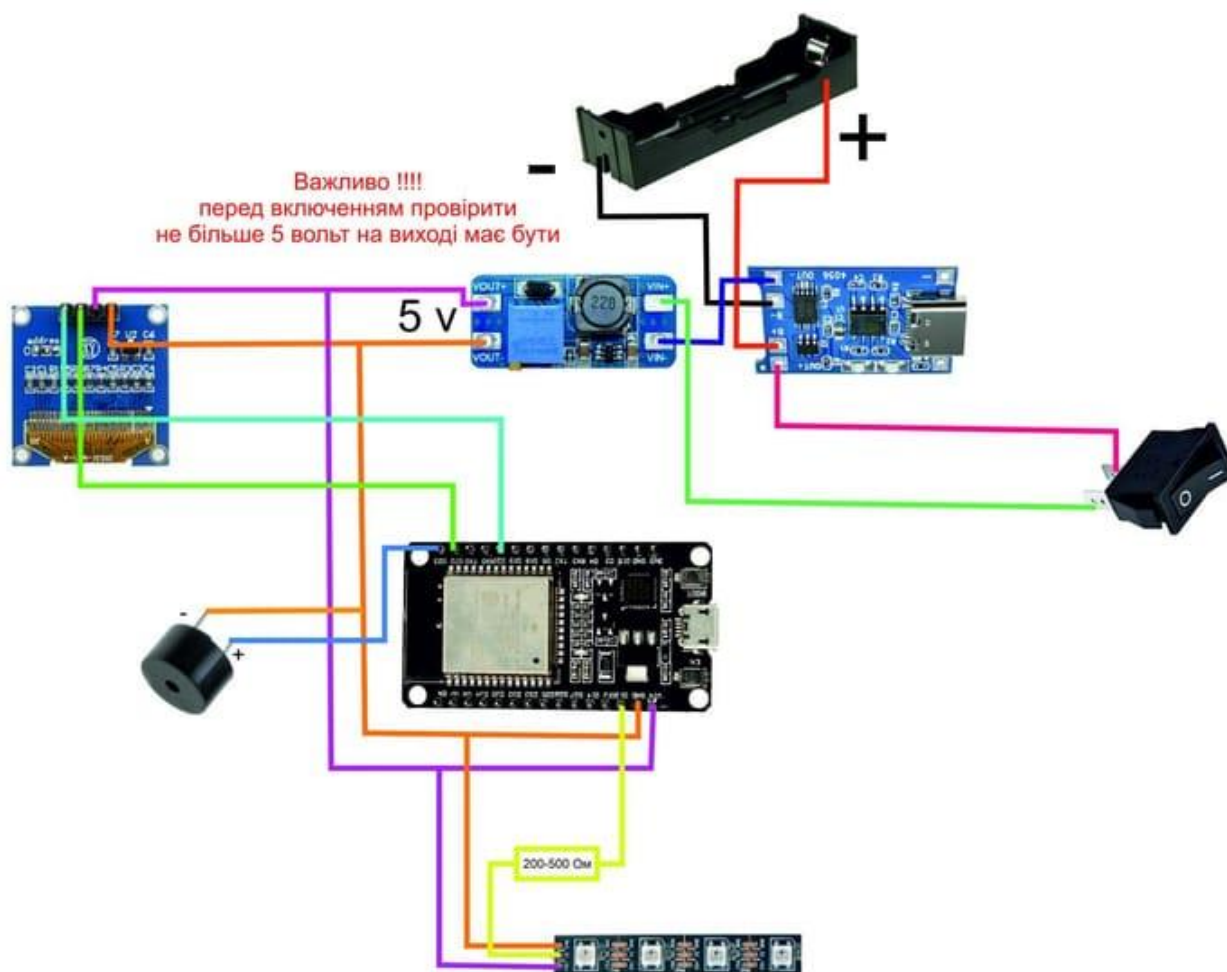


Рисунок 1.10 – Схема підключення максимальної комплектації

Існує кілька готових наборів кодів для роботи з мапою. Основний набір компонентів включає плату на базі ESP32, резистор (рекомендується) та адресу LED стрічку. У розширеному комплекті використовуються: плата ESP32, адреса LED стрічка, дисплей, акумулятор, підвищувач напруги, зарядний пристрій для акумулятора, звуковий сигнал (у багатьох випадках не працює), вимикач та резистор (рекомендується).

В цій моделі використовуються адреса LED стрічка WS2812B RGB (рис.1.9) яка була перепаяна та зібрана згідно с розташуванням областей на мапі України (рис.1.11). На рисунку також зображено послідовність підключення цих світлодіодів.



Рисунок 1.11 – Розташування та підключення світлодіодів

1.3.2 Портативний пристрій для сигналізації повітряної тривоги, який можна під'єднати до зовнішньої сирени

Розглянемо створений прототип пристрою для сигналізації повітряної тривоги, який можна під'єднати до зовнішньої сирени [29]. Існують автономні сирени, вебсайти та мобільні додатки, що сповіщають про початок та кінець тривоги, але це персональний портативний пристрій, що може працювати як самостійно, так і з зовнішньою сиреною в громадських місцях.

Для перетворення цього в серійний продукт необхідно ще багато зусиль, особливо щодо безпеки. Проте, це може бути цікавим проектом.



Рисунок 1.12 – Портативний пристрій для сигналізації повітряної тривоги

Основні функції пристрою охоплюють:

- З'єднання з WiFi мережею;
- використання NTP для корекції часу;
- індикацію стану тривоги за допомогою RGB LED;
- відтворення звукового сигналу під час активації або деактивації тривоги через динамік (у реальних умовах можна використовувати сирену);
- можливість вимкнення звуку динаміка за допомогою перемикача;
- налаштування робочих годин пристрою (в умовах експлуатації він повинен працювати цілодобово).

Наступна версія (рис 1.13) буде більш автономною з використанням двох акумуляторів 18650 та GSM модуля.

Для створення базової версії нам знадобляться:

- Модуль ESP32 WROOM DevKit;
- RGB LED індикатор;
- динамік;
- перемикач на дві або три позиції;
- резистори на 220 Ом;
- провідники-перехідники;

- USB-кабель для підключення та живлення мікроконтролера;
- корпус, надрукований на 3D-принтері.

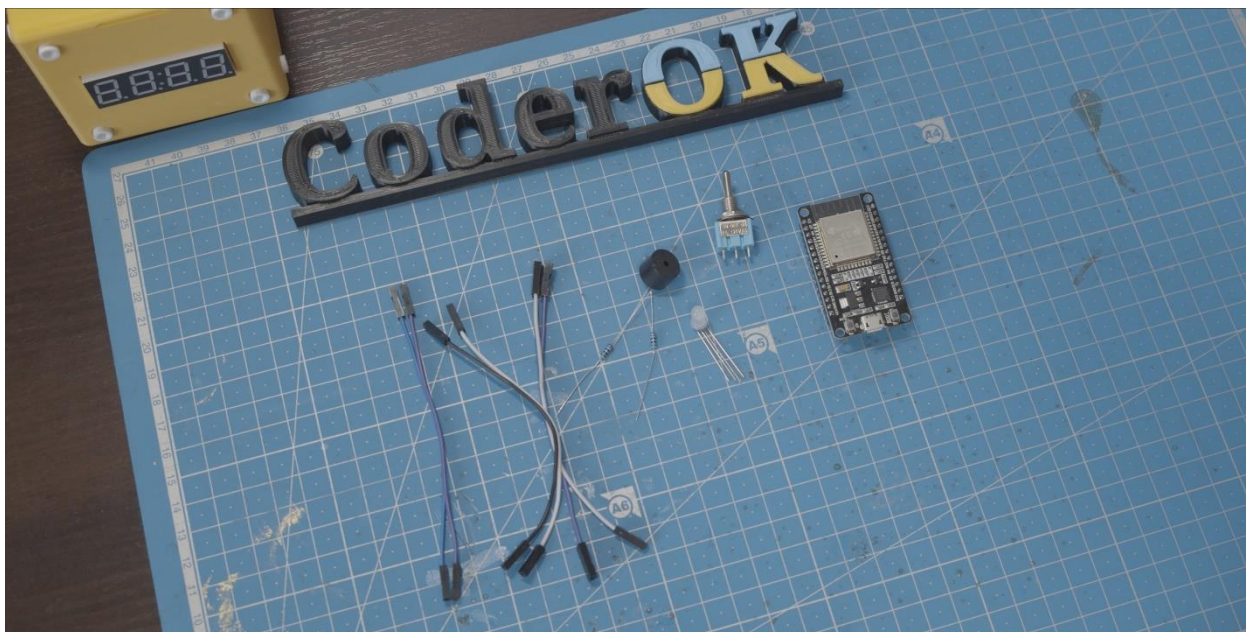


Рисунок 1.13 – Вдосконалена версія пристрою

Корпус буде нагадувати старі сигналізації на магазинах (рис. 1.14). Пристрій можна зробити автономним, додавши дві батареї 18650 та GSM-модем для роботи в умовах можливого відключення електроживлення.

Для 3D-модельовання використовую Blender, потужний опенсорс-інструмент, який не надто складний для освоєння.

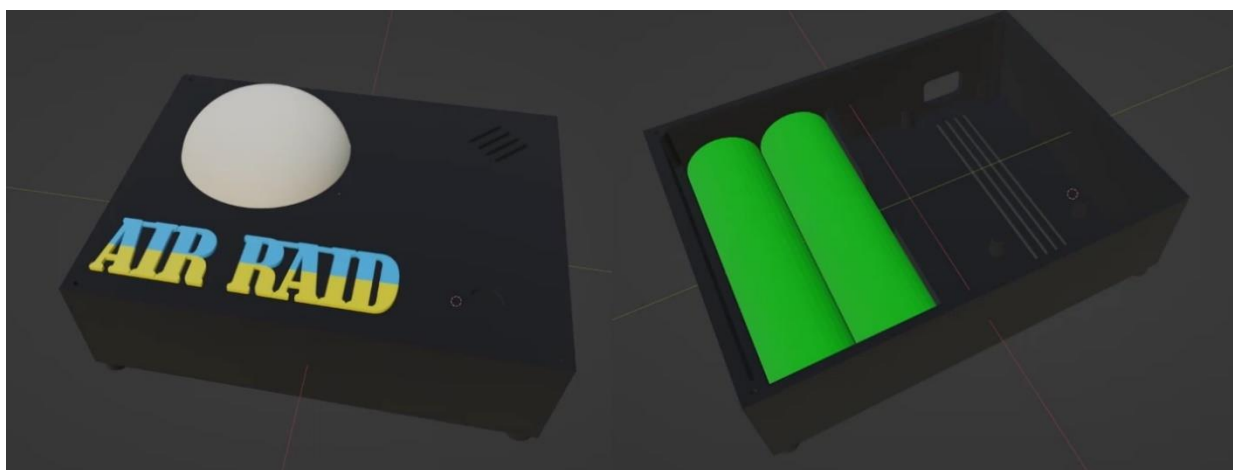


Рисунок 1.14 – 3D модель корпусу

Пристрій базується на ESP32-WROOM-32, який має двоядерний 32-розрядний процесор Xtensa LX6 з тактовою частотою 160/240 МГц та

вбудованою флеш-пам'яттю об'ємом 4 Мб. Це більше ніж достатньо для більшості проєктів. Вбудований Wi-Fi та Bluetooth додають необхідних можливостей без додаткових модулів.

Програмування ESP32 зазвичай здійснюється на C/C++, але можна використовувати MicroPython, Lua та інші мови. Електрична схема проста: мікроконтролер живиться від 5V блока живлення або USB. Резистори на 220 Ом між контактами йдуть до анодів RGB-світлодіода. Пін D13 контролює червоний світлодіод, а пін D12 — зелений. Катод підключається до негативного контакту мікроконтролера.

Перемикач встановлюється між динаміком та мікроконтролером (рис 1.15). Плюсовий провід йде через перемикач до динаміка, а мінусовий — безпосередньо до динаміка.

Для підключення зовнішньої сирени потрібне реле на 5V із зовнішнім джерелом живлення, оскільки мікроконтролер може видавати лише 3,3V. Сигнальний пін реле підключається до мікроконтролера. Коли подається сигнал HIGH, реле підключає сирену до 220V змінного струму; сигнал LOW відключає її.

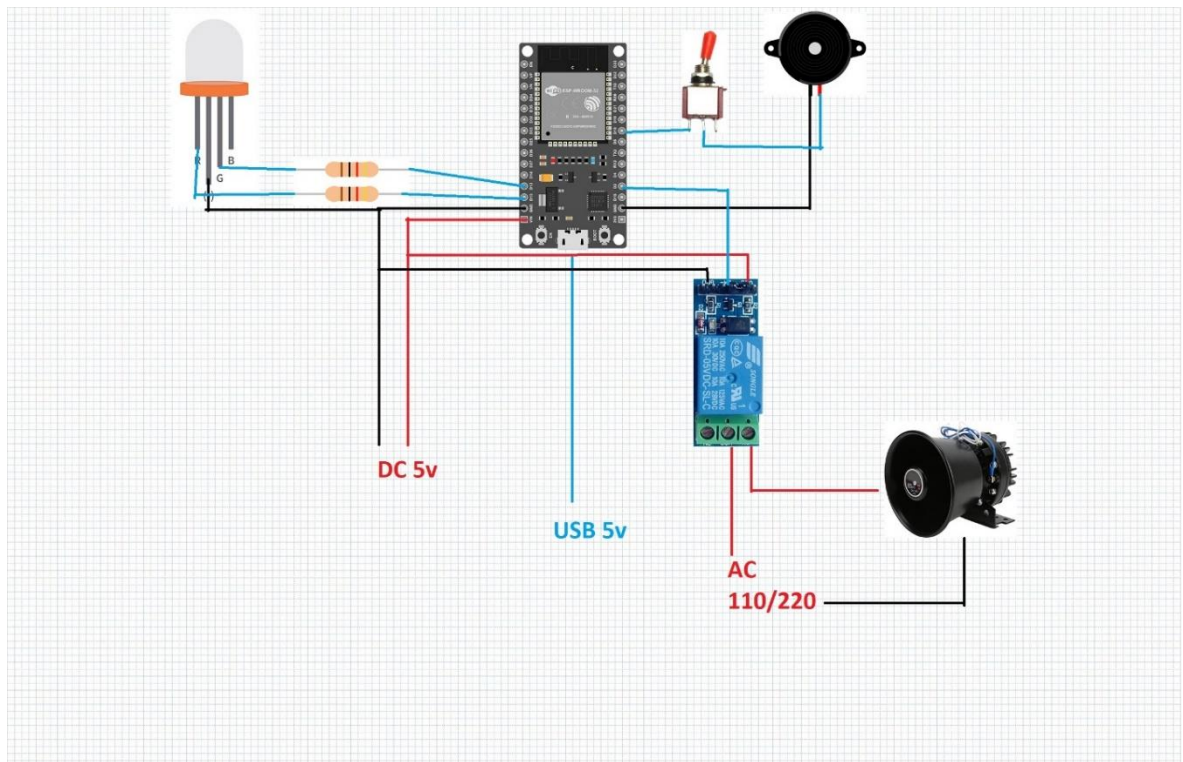


Рисунок 1.15 – Схема підключення пристрою до зовнішньої сирени

Для управління пристроєм створюється чотири класи: Music, Alert API, NTP та Logic. Вбудований RTC у ESP32 може дрейфувати з часом, тому використовуватимемо NTP для синхронізації. HTTP GET запити для API будуть містити API Key у заголовку. Отримані відповіді у форматі JSON десеріалізуватимемо за допомогою бібліотек, таких як ArduinoJson.

Основний клас Logic керуватиме пристроєм, оновлюватиме NTP, відправлятиме API запити, відтворюватиме звуки та контролюватиме RGB LED. Для зовнішньої сирени буде потрібен додатковий цифровий вихід для керування реле.

1.4 Висновки до розділу 1

Перший розділ включає огляд різних методів та технологій для передачі інформації про тривоги, включаючи візуальну та слухову демонстрацію, а також аналіз існуючих систем і прототипів для сповіщення про надзвичайні ситуації.

У розділі було розглянуто важливі елементи візуальної демонстрації, такі як світлові маячки та адресні світлодіоди, що є ефективними засобами для привертання уваги громадян у разі тривоги. Також була розглянута слухова демонстрація за допомогою сирен, гучномовців та інших акустичних засобів.

Висновок розділу 1 полягає у тому, що на основі ретельного аналізу різних методів та технологій можна побачити, що для ефективного сповіщення про тривоги використання комбінації візуальних і слухових засобів разом з програмним забезпеченням на платі ESP32 є дуже ефективним підходом.

Також було розглянуто декілька зроблених проєктів з використанням ESP32 та визначено різні підходи для проєктування систем з використанням API.

РОЗДІЛ 2 МОДЕЛЮВАННЯ ТА РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО МОДУЛЮ

Ключовими аспектами проектування є розробка алгоритму функціонування програмного модуля та вибір відповідних технологій, які органічно впишуться у загальну логіку проекту. Від цього залежатиме швидкодія програми та її функціонування в цілому.. Ретельно розроблений алгоритм дозволить оптимізувати процеси обробки даних, забезпечуючи своєчасне та точне реагування на тривожні ситуації.

Крім того, вибір відповідних технологій та компонентів забезпечить надійне функціонування системи навіть у складних умовах. Важливо також врахувати енергоефективність і стабільність роботи всіх елементів пристрою, оскільки це безпосередньо впливає на довговічність і безпеку користувачів.

Таким чином, проектування такого пристрою вимагає комплексного підходу, включаючи детальний аналіз вимог, вибір відповідних технологій і компонентів, та ретельну розробку алгоритмів. Це забезпечить ефективну, надійну та безпечну роботу системи сповіщення тривоги, сприяючи захисту та безпеці людей у захисних сховищах.

2.1 Методи проектування системи та процес її розробки

Під час розробки сучасного апаратно-програмного модуля ключовим етапом є використання керуючих пристроїв, таких як мікроконтролери. Вони виступають як інтерфейс між користувачем та системою.

У роботі над інтерактивним табло на адресних світлодіодах для захисних сховищ, логіка роботи базується на використанні мікроконтролерів як керуючих пристроїв. Ці мікроконтролери виступають інтерфейсом між пристроєм і користувачем. Деякі з них мають вже вбудовані модулі передачі сигналів.

Технології віддаленого зв'язку, такі як Wi-Fi та Bluetooth, дозволяють забезпечити взаємодію між мобільними телефонами та іншими пристроями, такими як адресні світлодіоди у нашому випадку. Це реалізується за

допомогою протоколів віддаленого зв'язку, які забезпечують обмін інформацією та дистанційне керування через мобільні додатки або інші пристрої, такі як пульт або комп'ютер. Такий підхід дозволяє створювати ефективні та зручні для користувачів системи сповіщень.

Давайте проаналізуємо основні технології для дистанційного обміну даними, які доступні для використання без спеціальних дозволів і є безкоштовними для створення нових продуктів (рис 2.1).

Technology	Wi-Fi	Bluetooth	ZigBee
Standard	802.11b	802.15.1	802.15.4
Target Application	Web, Email, Video	Voice, Cable replacement	Monitoring and control
Flash Size	1MB+	250KB+	25KB - 70KB
Battery Life (days)	0.5 - 5	1 - 7	100 - 1,000+
Basic Network	32 / ...	7 / ...	255 / 65535
Throughput (kbps)	11,000+	1,000	20 - 250
Range (m)	1 - 100+	1 - 100+	1 - 75+
Advantage	Speed, flexibility (IP)	Price, convenience	Low power, simple

Рисунок 2.1 – Бездротові технології та стандарти передачі даних.

Wi-Fi – це технологія бездротових локальних мереж, що ґрунтується на стандартах IEEE 802.11. Логотип Wi-Fi є зареєстрованою торговою маркою організації Wi-Fi Alliance. Назва Wi-Fi, утворена від англійського виразу Wireless Fidelity ("бездротова точність"), позначає набір стандартів для передачі цифрових даних через радіохвилі. Основні частоти, на яких працює Wi-Fi, – 2,4 ГГц (2412 МГц-2472 МГц) і 5 ГГц (5160-5825 МГц).

Сигнал Wi-Fi може передаватися на великі відстані, навіть якщо має низьку потужність, але для його отримання на значних відстанях від звичайного маршрутизатора необхідна антена з високим коефіцієнтом посилення, наприклад, параболічна антена або Wi-Fi гармата.

Будь-яке обладнання, що відповідає стандарту IEEE 802.11, щоб отримати дозвіл на використання логотипу Wi-Fi, може протестувати та сертифікуватись через Wi-Fi Alliance.

Переваги Wi-Fi:

1. Можливість створення мережі без прокладки кабелю, що знижує витрати на встановлення та розширення мережі. Бездротові мережі можуть обслуговувати зони, де кабелі прокласти неможливо, як-от на відкритих просторах або в будівлях з історичною цінністю;
2. забезпечення доступу до мережі мобільних пристроїв;
3. широка поширеність пристроїв Wi-Fi на ринку. Сумісність обладнання гарантується обов'язковою сертифікацією з логотипом Wi-Fi;
4. мобільність. Користувачі можуть користуватися Інтернетом у зручному місці, не будучи прив'язаними до одного місця. У зоні покриття Wi-Fi кілька користувачів можуть одночасно підключатися до Інтернету з різних пристроїв [8].

На сьогоднішній день пропускна здатність Wi-Fi значно зросла (рис 2.2).

	WiFi 5	WiFi 6	WiFi 6E	WiFi 7
Дата запуску	2013	2019	2021	2024 (попередньо)
IEEE стандарт	802.11ac	802.11ax	802.11ax	802.11be
Максимальна швидкість	3.5 Гбіт/с	9.6 Гбіт/с	9.6 Гбіт/с	46 Гбіт/с
Частота	5 ГГц	2.4 ГГц, 5 ГГц	2.4 ГГц, 5 ГГц, 6 ГГц	2.4 ГГц, 5 ГГц, 6 ГГц
Ширина каналу	20, 40, 80, 80+80, 160 МГц	20, 40, 80, 80+80, 160 МГц	20, 40, 80, 80+80, 160 МГц	До 320 МГц
Модуляція	256-QAM OFDM	1024-QAM OFDMA	1024-QAM sOFDMA	4096-QAM OFDMA (with extensions)
MIMO	4x4 MIMO DL MIMO	8x8 UL/DL MU-MIMO	8x8 UL/DL MU-MIMO	16x16 UL/DL MU-MIMO
RU	/	RU	RU	Multi-RUs
MAC	/	/	/	MLO

Рисунок 2.2 – Порівняння стандартів Wi-Fi

2.2 Обґрунтування причин вибору програмно-апаратних засобів для управління системою оповіщень

Було встановлено, що технології для системи сповіщення про небезпеку базуються на різноманітних датчиках і пристроях, керування якими здійснюється через глобальну мережу. Основними засобами зв'язку для таких систем можуть бути контролери, що ґрунтуються на:

- Raspberry Pi Pico W
- ESP8266
- ESP32
- WeMos D1 Wi-Fi

На сьогоднішній день ESP32 є одним з найбільш популярних мікроконтролерів з підтримкою Wi-Fi. У цьому проекті планується створити систему, де компоненти будуть взаємодіяти через об'єкт управління, який буде представлений мікроконтролером.

ESP8266 - це мікроконтролер, розроблений китайською компанією Espressif Systems, який має вбудований інтерфейс Wi-Fi. Недоліком цього варіанту є в тому що цей мікроконтролер не має вбудованої флеш-пам'яті у системі на кристалі (SoC), і програми користувача запускаються зовнішньою флеш-пам'яттю через інтерфейс SPI [17].

У 2014 році цей мікроконтролер здобув популярність завдяки випуску перших продуктів на його основі за дуже низькою ціною.

У весняний період 2016 року розпочалося виробництво ESP8285, який поєднує можливості ESP8266 з 1 Мб флеш-пам'яті. Восени 2015 року компанія Espressif представила новинку - чіп ESP32 та модулі, побудовані на його основі.

Мікроконтролер має такі характеристики:

- робоча частота 80 МГц, але можливий розгін до 160 МГц;
- процесор Tensilica 32-розрядний;
- підтримує Wi-Fi стандарту IEEE 802.11 b/g/n і включає в себе захист WEP, а також WPA/WPA2;
- 14 портами вводу-виводу (11 з яких можна використовувати), підтримкою SPI, I2S, UART, 10-розрядним аналого-цифровий перетворювач, але I2C доступний тільки через біт-банг;
- робочий діапазон живлення від 2,2 до 3,6 В;
- споживання електроенергії: до 215 мА у режимі передачі, 100 мА у режимі прийому, 70 мА у режимі очікування;

- підтримує три рівні низької потужності, включаючи режими сну модему (15 мА), легкий сон (0,4 мА) і глибокий сон (15 мА);
- використовує зовнішню SPI ROM для виконання програм, з можливістю завантаження програми в кеш інструкцій. Завантаження є апаратним і не вимагає додаткових налаштувань від програміста;
- підтримує до 16 МБ зовнішньої пам'яті програм з можливістю стандартного, подвійного або чотириразового інтерфейсу SPI.

Інформацію щодо електричних параметрів, розеток та схем підключення можна знайти в документах 0A-ESP8266EX__Datasheet та 0B-ESP8266__System_Description, які входять до комплекту документації Espressif SDK.

Програмне джерело мікроконтролера ESP8266 визначається станом портів GPIO0, GPIO2 і GPIO15 під час включення (наприклад, при подачі живлення). Два основних режими роботи полягають у виконанні коду через інтерфейс UART (коли GPIO0 = 0, GPIO2 = 1 і GPIO15 = 0) або із зовнішньої пам'яті (коли GPIO0 = 1, GPIO2 = 1 і GPIO15 = 0).

Засоби програмування програмного забезпечення для мікроконтролера включають у себе наступне:

- А) компілятори: У складі програмного пакету розробника (SDK) включає компілятор Xtensa LX106, що базується на GNU Compiler Collection та має відкритий вихідний код. Різні версії SDK можуть включати різні збірки цього компілятора зі своїми унікальними налаштуваннями та функціоналом.
- Б) бібліотеки для взаємодії з периферійними пристроями контролера, а також роботи зі стеками WiFi та TCP/IP. Ці бібліотеки дозволяють розробникам легко взаємодіяти з різними аспектами контролера, такими як введення-виведення, мережеві операції тощо.
- В) засоби для завантаження виконуваних файлів у пам'ять програм мікроконтролера. Ці інструменти дозволяють ефективно завантажувати

та виконувати програми на мікроконтролері, забезпечуючи правильну роботу інтегрованих програмних рішень.

Необов'язковий IDE.

Espressif пропонує свій набір інструментів розробки безкоштовно. До цього набору входять компілятор GCC, бібліотеки від Espressif, а також утиліта для завантаження XTCOM. Бібліотеки надаються у вигляді скомпільованих файлів без доступу до вихідного коду. Espressif підтримує дві версії SDK: одну на базі власної операційної системи (ОСРВ) та іншу на базі зворотного виклику.

Окрім офіційного SDK, існує декілька альтернативних проєктів SDK. Ці пакети використовують бібліотеки Espressif або пропонують власні варіанти, створені через зворотну інженерію.

"esp-open-sdk" представляє собою покращену версію Espressif SDK, яка містить компілятор GCC та певні бібліотеки Espressif. Цей інструмент доступний лише для Linux.

Неофіційний комплект розробки, створений Михайлом Григор'євим включає інсталятор для Windows, власний компілятор GCC з інтеграцією у графічну IDE Eclipse, документацію Espressif та сучасні набори бібліотек, також різноманітні утиліти. Цей пакет має російськомовний форум для підтримки.

"Arduino IDE для ESP8266" - це додаток до Arduino IDE, який дозволяє програмувати ESP8266 так само, як будь-який інший модуль Arduino. Цей пакет включає в себе можливості мережі ESP8266, компілятор GCC та програму для завантаження мікропрограм ESPTool.

"Набір інструментів GNU для ESP8266" можна легко інтегрувати у середовище Visual Studio для зручного програмування і розробки з ESP8266.

"ESP8266 GCC Toolchain" розроблений Максом Філіпповим.

"Sming" - це проєкт, який надає сумісні з Arduino бібліотеки на базі стандартних бібліотек Espressif, проте без використання середовища Arduino.

Інфраструктура для мережі.

Один зі звичайних способів використання ESP8266 як бази для Internet of Things часто пов'язаний з його встановленням у будинках або офісах. У таких випадках, пристрій зазвичай підключається до домашньої або офісної локальної мережі через роутер з доступом до Інтернету. Користувач може управляти цим пристроєм за допомогою планшета або комп'ютера через місцеву мережу або віддалено через Інтернет.

Розглядаючи ESP32 - це ряд доступних за невеликі гроші, але потужних чіпів від компанії Espressif Systems. Це системний чіп із вбудованими контролерами Wi-Fi, Bluetooth і Thread. Моделі ESP32 і ESP32-S мають процесори з архітектурою Tensilica, а ESP32-C і ESP32-H використовують ядра з відкритою архітектурою RISC-V. У мікросхемі вже вбудовані різноманітні радіочастотні компоненти: балансувальний трансформатор, перемикачі антенних каналів, фільтри та інші модулі керування сигналами. ESP32 був розроблений компанією з Шанхаю і виготовляється TSMC на технологіях 40 нм і 28 нм. Ця лінійка є спадкоємцем чіпів ESP8266.

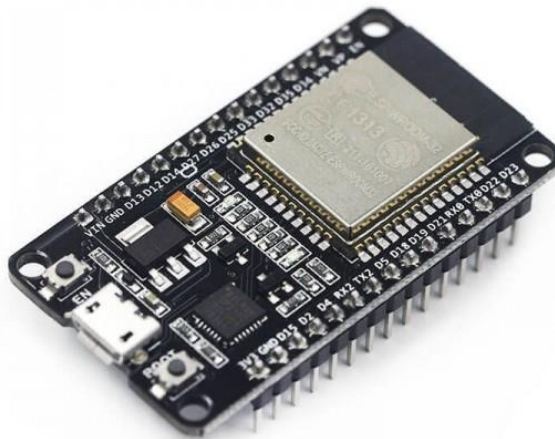


Рисунок 2.3 – Плата ESP32-WROOM

Плата ESP32 є удосконаленою версією попереднього мікроконтролера ESP8266. Незважаючи на те, що не так давно була випущена оновлена версія плати, багато розробників віддають перевагу саме ESP32 через покращене

ядро з більш швидким Wi-Fi, Bluetooth 4.2 і різними входами/виходами (аналоговими/цифровими). Між цими двома платами існують численні відмінності (рис.2.4).

Specifications	ESP8266	ESP32
MCU	Xtensa® Single-Core 32-bit L106	Xtensa® Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Yes, HT20	Yes, HT40
Bluetooth	None	Bluetooth 4.2 and below
Typical Frequency	80 MHz	160 MHz
SRAM	160 kBytes	512 kBytes
Flash	SPI Flash , up to 16 MBytes	SPI Flash , up to 16 MBytes
GPIO	17	36
Hardware / Software PWM	None / 8 Channels	1 / 16 Channels
SPI / I2C / I2S / UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	None	1
Ethernet MAC Interface	None	1
Touch Sensor	None	Yes
Temperature Sensor	None	Yes
Working Temperature	-40°C – 125°C	-40°C – 125°C

Рисунок 2.4 – Порівняння плат ESP8266 та ESP32

Одним з головних недоліків ESP8266 є обмежена кількість доступних контактів на платі [18]. У випуску ESP32 цей аспект було вдосконалено, піни стали більшими за кількістю, а також набули додаткових функцій [9].

Основні види плат ESP32

ESP32-WROOM: Ця плата володіє не лише вбудованим Wi-Fi та Bluetooth, а й можливістю використання широкого спектру периферійних пристроїв (рис. 2.3). Її висока продуктивність і низьке енергоспоживання роблять її ідеальним вибором для IoT-проектів, які вимагають постійного з'єднання з мережею та взаємодії з різними сенсорами та актуаторами.

ESP32-WROVER: Особливість цієї моделі полягає у великій кількості оперативної пам'яті, що дає змогу виконувати більш складні завдання, такі як обробка великих обсягів даних або використання ресурсів-інтенсивних алгоритмів машинного навчання. Це робить ESP32-WROVER популярним вибором серед розробників IoT-пристроїв, які потребують великої продуктивності.

ESP32-PICO: Ця компактна плата зберігає всі основні функції ESP32, але з мініатюрним розміром і вбудованою антеною. Вона ідеально підходить для проектів, де обмежений обсяг простору або потрібна висока мобільність пристрою, наприклад, носимі пристрої або розумний одяг.

ESP32-CAM: Ця модель має вбудовану камеру, що відкриває безліч можливостей для застосування у відеонагляді, створенні систем розпізнавання обличчя, відеочатів та інших додатків, які потребують візуального взаємодії з оточуючим середовищем.

ESP32-S2: Ця модель, не маючи модуля Bluetooth, вирішує завдання, пов'язані з підключенням до Інтернету через USB-інтерфейс. Її швидкість та низьке енергоспоживання дозволяють створювати ефективні IoT-пристрої, зокрема пристрої з обміном даними через USB або мережу Wi-Fi.

Модуль ESP32 CH340 Type-C.

Модуль ESP32 CH340 Type-C є мікроконтролерним рішенням, що базується на чипі ESP32 та має USB-інтерфейс, завдяки чипу CH340. Цей модуль відрізняється великими можливостями для розробки різноманітних проектів у сфері Інтернету речей (IoT) завдяки високій продуктивності ESP32, який включає в себе двоядерний процесор, підтримку Wi-Fi та Bluetooth.

Технічна реалізація цього проекту передбачає використання мікроконтролера ESP32 (рис. 2.5). Це дозволяє забезпечити високу продуктивність, ефективне з'єднання з мережею та легку інтеграцію з іншими пристроями.

Модуль ESP32 CH340 Type-C ідеально підходить для різних застосувань, зокрема: розумні домашні пристрої (автоматизація та контроль побутових систем), бездротові сенсорні мережі (моніторинг та збір даних з різних сенсорів), промислові системи автоматизації (контроль і управління виробничими процесами) та прототипування IoT-пристроїв (розробка та тестування нових технологічних рішень).



Рисунок 2.5 – Плата ESP32 CH340 Type-C

Характеристики модуля ESP32 CH340 Type-C наведені у табл. 2.1 [10]

Таблиця 2.1 – Характеристики модуля ESP32 CH340 Type-C

Параметр	Значення
Мікроконтролер	ESP32
Процесор	Двоядерний Tensilica LX6
Тактова частота процесора	До 240 МГц
Флеш-пам'ять	4 МБ (варіанти можуть мати більше або менше)
Оперативна пам'ять	520 КБ SRAM
Wi-Fi	802.11 b/g/n
Bluetooth	Bluetooth 4.2 BR/EDR і BLE
Інтерфейси	UART, SPI, I2C, I2S, PWM, ADC, DAC
Кількість GPIO пінів	34
Аналогові входи	18 (ADC)
Аналогові виходи	2 (DAC)

Параметр	Значення
Максимальна напруга GPIO	3.3 В
USB інтерфейс	USB Type-C з чіпом CH340 для UART зв'язку
Підтримка MicroPython	Так
Підтримка Arduino IDE	Так
Робоча напруга	3.0-3.6 В (зазвичай 3.3 В)
Живлення через USB	5 В
Робоча температура	-40°C до +85°C

Підсистема I2S в ESP32 також пропонує високошвидкісну шину, яка підключена безпосередньо до оперативної пам'яті, забезпечуючи прямий доступ до пам'яті. Це дозволяє налаштувати підсистему I2S ESP32 для апаратно контрольованого надсилання або отримання паралельних даних [11].

Особливості ESP32 CH340 Type-C [12].

Мікроконтролер і управління:

- Двоядерний (або одноядерний) 32-розрядний процесор Tensilica Xtensa LX6 з тактовою частотою 160 або 240 МГц і до 600 DMIPS (Dhrystone MIPS);
- надмалопотужний співпроцесор;
- пам'ять: 520 КБ пам'яті SRAM.

Бездротові технології:

- Wi-Fi: 802.11 b/g/n;
- Bluetooth: v4.2 BR/EDR і BLE.

Периферійні інтерфейси:

- 12-розрядний АЦП до 18 каналів;
- 2 × 8-біт ЦАП;
- 10 × портів для підключення ємнісних датчиків;
- 4 × головний інтерфейс SPI (провідні пристрої);

- головний інтерфейс $2 \times I2S$;
- головний інтерфейс $2 \times I2C$;
- інтерфейс $3 \times UART$;
- хост-контролер SD/SDIO/CE-ATA/MMC/eMMC;
- підпорядковані контролери SDIO/SPI (відомі пристрої) ;
- інтерфейс Ethernet MAC із виділеною підтримкою DMA та IEEE 1588 Precision Time Protocol;
- шина CAN 2.0;
- ІЧ-пульт дистанційного керування (передавач/приймач, до 8 каналів) ;
- можливість підключення двигунів і світлодіодів через ШІМ вихід;
- датчик Холла;
- аналоговий підсилювач малої потужності.

Безпека:

- Підтримуються всі функції безпеки IEEE 802.11, включаючи WPA, WPA/WPA2 і WAPI;
- безпечне завантаження;
- шифрування флешки;
- апаратне криптографічне прискорення: AES, SHA-2, RSA, криптографія з еліптичною кривою (ECC), апаратний генератор випадкових чисел.

Управління живленням:

- Лінійний регулятор з низьким падінням напруги;
- індивідуальне живлення для RTC;
- споживання 5-2,5 мкА в режимі «глибокий сон»;
- пробудження від переривання GPIO, таймера, вимірювання АЦП, переривання датчика ємнісного датчика;
- робоча напруга від 2,2 до 3,6 В;
- робоча температура від $-40\text{ }^{\circ}\text{C}$ до $+125\text{ }^{\circ}\text{C}$.

Максимальна швидкість передачі даних:

- 150 Мбіт/с при 11n HT40;
- 72 Мбіт/с при 11n HT20;
- 54 Мбіт/с при 11g;
- 11 Мбіт/с при 11b.

Максимальна потужність передачі:

- 19,5 дБм при 11b;
- 16,5 дБм при 11g;
- 15,5 дБм при 11n.

Мінімальна чутливість приймача:

- 98 дБм.

Стабільна пропускна здатність UDP:

- 135 Мбіт/с.

Корпус:

- ESP32 поставляється в корпусі QFN з 48 контактами по периметру і одним великим радіатором по центру, який також використовується для сигнального заземлення.

Внутрішня пам'ять:

- 448 КБ ПЗУ для завантаження та основних функцій;
- 520 КБ (8 КБ швидкої пам'яті RTC включено) вбудована SRAM для даних та інструкцій;
- 8 КБ SRAM в RTC для швидкої пам'яті та використання під час глибокого сну;
- 8 КБ SRAM в RTC як вільна пам'ять під час глибокого сну;
- 1 КБ eFuse, з яких 256 біт використовуються для системи (MAC-адреса та конфігурація мікросхеми), а ще 768 біт зарезервовано для клієнтських програм.

Підтримка зовнішньої пам'яті:

- До 16 МБ зовнішньої флеш-пам'яті з апаратним шифруванням на основі AES.

- Виконання коду та зчитування даних підтримується на флеш-пам'яті та SRAM.

- Запис даних підтримується тільки на SRAM.

ESP32-WROVER:

- Інтегрує зовнішній SPI-флеш 4-16 МБ;

- 4 МБ SPI Flash може бути картою пам'яті з підтримкою доступу до 8, 16 і 32 біт.

ESP32 CH340 Type-C має додаткові особливості і можливості, які варто врахувати:

- Додаткова пам'ять: Окрім флеш-пам'яті SPI у діапазоні від 4 до 16 МБ, ESP32-WROVER включає також 4-8 МБ PSRAM для додаткового місця в пам'яті. Це дозволяє зберігати більше даних і виконувати більш складні програми;

- генератор кварцу: Прошивка ESP32 Wi-Fi/BT підтримує лише кварцовий генератор з тактовою частотою 40 МГц, що забезпечує стабільну роботу пристрою;

- RTC та управління низьким споживанням: ESP32 має різні режими живлення, які дозволяють зменшувати споживання енергії в залежності від потреби програми. Режими включають активний режим, режим сну модему, легкий сплячий режим, гібернацію та режим глибокого сну;

- шаблони сну: ESP32 підтримує різні шаблони сну, які дозволяють ефективно використовувати енергію, залежно від конкретного застосування. Наприклад, шаблон асоційованого сну змінює живлення між активним режимом, модемом і режимом легкого сну, тоді як шаблон датчика ULP зберігає енергію, коли основний процесор перебуває в стані сну;

- модульні плати SMT: Модулі SMT на основі ESP32 відрізняються компактністю та містять SoC ESP32, що полегшує їх інтеграцію в інші пристрої. Це забезпечує зручність при розробці складних систем;

–

- виміряні конструкції інверсної F-антени: Деякі модулі ESP32 CH340 Туре-С мають виміряні конструкції інверсної F-антени, що сприяє кращому відстеженню антени на друкованій платі. Це може покращити якість бездротового зв'язку.

2.3 Вибір адресних світлодіодів

Адресний світлодіод - це пристрій, який може відобразити різні кольори світла і робити це у відповідності зі сигналом, що йому надходить. Його часто використовують у світлодіодних матричних дисплеях, які можуть відобразити зображення або інформацію за допомогою багатьох світлодіодів, кожен з яких контролюється окремо.

Ці адресні світлодіоди зазвичай використовують технологію керування світлодіодами, таку як WS2812B, WS2813, APA102 і т.д. Ці пристрої мають вбудований мікроконтролер, який дозволяє керувати кожним світлодіодом окремо через один або декілька каналів передачі даних. Таке керування дозволяє створювати складні ефекти, змінювати кольори і яскравість світла в реальному часі.

У нашому процесі вибору адресних світлодіодів для проекту ми ретельно вивчили кілька ключових параметрів, таких як кольорова гама, яскравість, споживана потужність і зручність управління. Після ретельного аналізу ми прийшли до висновку, що WS2812B RGB є найкращим варіантом для наших потреб з наступних причин.

Ці світлодіоди мають вражаючий спектр кольорів і можливість індивідуального керування кожним діодом, що дозволяє створювати дивовижні світлові ефекти з великою точністю. Крім того, їхня висока яскравість забезпечує чітке відтворення кольорів та інтенсивне світіння, що важливо для нашого проекту.

Однією з переваг WS2812B RGB є легкість управління, особливо за допомогою мікроконтролера, що робить їх ідеальними для створення

інтерактивних світлових систем. Також важливою є їхня надійність та довговічність, що робить цей тип світлодіодів популярним серед розробників.

Загалом, після уважного вивчення всіх аспектів ми впевнено вирішили, що WS2812B RGB є найбільш оптимальним вибором для нашого проекту з адресними світлодіодами.

Проект передбачає використання інформаційного табло для відображення важливої інформації про тривожні ситуації, такі як початок тривоги, її тип і закінчення. Воно складається з п'яти світлодіодів WS2812B RGB (рис. 2.6). Інформаційне табло в рамках цього проекту є ключовим елементом для візуального сповіщення користувачів про тривожні ситуації у захисних сховищах.

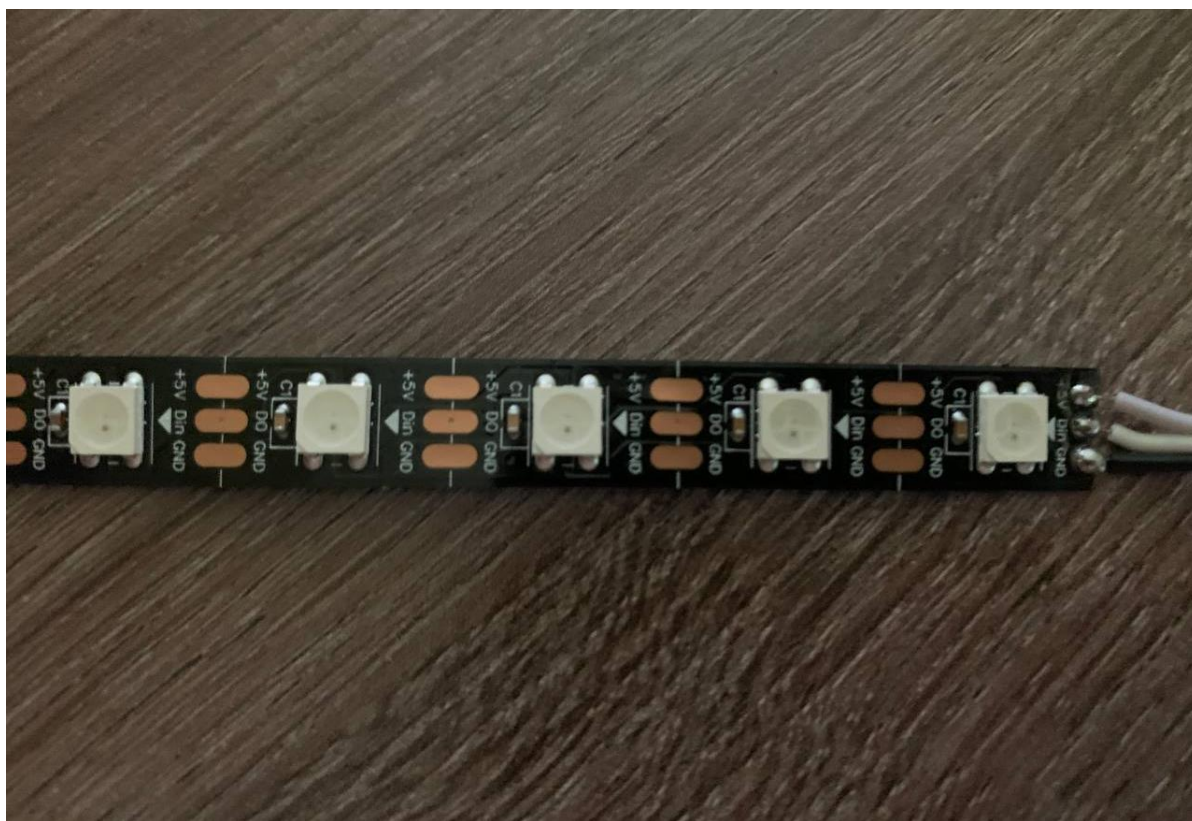


Рисунок 2.6 – Світлодіоди WS2812B RGB

Його функції та характеристики розглядаються з декількох аспектів:

- Відображення початку тривоги: Інформаційне табло має здатність миттєво відобразити початок тривоги на екрані. Це може бути представлено текстовою інформацією (наприклад, "Початок повітряної

- тривоги") або символами, що відображають тип тривоги (наприклад, зображення хмари для повітряної тривоги);
- інформація про тип тривоги: Крім того, інформаційне табло відображає конкретний тип тривоги, що дозволяє користувачам швидко розпізнати ситуацію. Наприклад, для хімічної тривоги може бути показано відповідний символ або текстове повідомлення про цей тип небезпеки;
 - індикація закінчення тривоги: Після того, як тривога закінчується або ситуація повертається до нормального стану, інформаційне табло відображає повідомлення про це. Наприклад, "Тривога припинена" або "Ситуація під контролем";
 - вивід помилок: В проєкті буде виділено один світлодіод на вивід інформації. Коли прилад не отримав данні в API, буде загоратись один світлодіод білим кольором;
 - візуальний дизайн: Інформаційне табло має забезпечувати чітке і зрозуміле візуальне відображення інформації. Екран може бути розділеним на сегменти для різних типів інформації, використовувати яскраві кольори для привертання уваги та використовувати світлодіоди або інші світлові елементи для підсвічування.

Інформаційне табло у цьому проєкті є важливим засобом комунікації з користувачами захисних сховищ, що дозволяє їм швидко та ефективно реагувати на тривожні ситуації та забезпечує безпеку та зручність в умовах небезпеки.

2.4 Вибір звукової плати

Звукова плата MAX98357, вироблена компанією Maxim Integrated, є потужним інструментом для відтворення аудіо сигналів у високій якості через динаміки або навушники. Вона володіє вбудованим підсилювачем звуку, що забезпечує чітке і потужне звучання без необхідності в додаткових пристроях для підсилення сигналу.

Однією з головних переваг плати є її компактні розміри, які роблять її ідеальним вибором для вбудовування в різноманітні пристрої з обмеженим простором. Крім того, вона характеризується низьким рівнем споживання енергії, що робить її ефективним рішенням для портативної техніки і мобільних пристроїв.

Не менш важливою є висока якість звуку, яку забезпечує ця плата. Вона здатна передавати аудіо сигнали без спотворень і втрати деталей, що робить її популярним вибором серед виробників аудіоапаратури та спеціалізованих пристроїв звукового відтворення.

Узагальнюючи, звукова плата MAX98357 є надійним інструментом з високою якістю звуку, компактними розмірами і низьким споживанням енергії, що робить її привабливим вибором для широкого спектру застосувань у сучасній аудіотехніці.

Звукове супроводження в цьому проєкті буде забезпечується звуковою платою MAX98357 (рис. 2.7) з вбудованим динаміком. Ця звукова плата відповідає за відтворення звукових сигналів, таких як тривожні сигнали або повідомлення про тривоги, для користувачів захисних сховищ.

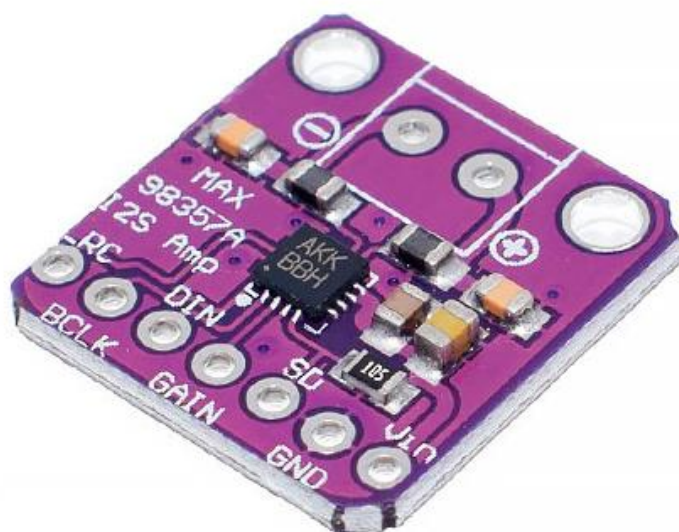


Рисунок 2.7 – Звукова плата MAX98357 з динаміком

Основні характеристики звукової плати MAX98357 та її динаміка включають:

- Висока якість звуку: MAX98357 підтримує відтворення високоякісного аудіо з низькими спотвореннями та шумами, що дозволяє чітко передавати звукові сигнали;
- інтегрований динамік: Наявність вбудованого динаміка на платі дозволяє спростити підключення та реалізацію звукового відтворення без необхідності використання додаткових зовнішніх пристроїв;
- легка інтеграція: Звукова плата MAX98357 забезпечує просте підключення до мікроконтролера ESP32 або іншого контролера через інтерфейси, такі як SPI або I2S, що робить її легкою у використанні та інтеграції в проєкт;
- низьке споживання енергії: Плата має оптимізовану систему керування енергією, що дозволяє зменшити споживання енергії при відтворенні звуку, що є важливим для мобільних та бездротових пристроїв;
- доступність додаткових функцій: Звукова плата може підтримувати різні додаткові функції, такі як контроль гучності, еквайзери та інші налаштування звуку, що дозволяє налаштовувати звукове відтворення під конкретні потреби проєкту.

Таким чином, використання звукової плати MAX98357 з динаміком забезпечить надійне та якісне звукове супроводження в цьому проєкті, що допоможе ефективно комунікувати тривожні сигнали та повідомлення користувачам у захисних сховищах.

2.5 Живлення пристрою

Для живлення нашого пристрою для захисних сховищ ми використовуємо модуль живлення для макетної плати MB-102 (5V / 3.3V) (рис. 2.8).



Рисунок 2.8 – Модуль живлення MB-102 (5V / 3.3V)

Цей модуль забезпечує стабільне живлення з напругою 5V або 3.3V, що дозволяє ефективно живити електронні компоненти пристрою і забезпечує його надійну роботу.

Звичайно, пристрій для захисних сховищ може мати різні види модифікацій живлення, а також комбіновані варіанти, що враховують конкретні потреби та умови експлуатації:

- Акумуляторне живлення: Простий та зручний варіант, де пристрій живиться вбудованим акумулятором, який можна заряджати через USB або інші джерела. Це забезпечує мобільність і автономність роботи пристрою.
- Сонячні панелі: Для пристроїв, які розташовані у відкритому просторі або на відкритих площах, можна використовувати сонячні панелі як додаткове джерело живлення. Це дозволяє зменшити залежність від мережі електроживлення.
- Зовнішнє живлення: Для пристроїв, що використовуються в стаціонарних умовах або мають постійний доступ до мережі електроживлення, можна використовувати зовнішнє джерело енергії, щоб забезпечити стабільне та безперебійне живлення.

- Резервне живлення: Для підвищення надійності пристрою можна використовувати резервне живлення, таке як резервний акумулятор або генератор, який автоматично включається у випадку відключення основного джерела живлення.
- Комбіновані варіанти: Такі варіанти можуть поєднувати декілька джерел живлення, наприклад, акумулятор з можливістю зарядки через сонячні панелі або зовнішнє джерело енергії. Це дозволяє оптимізувати ефективність живлення в різних умовах.

Кожен з цих варіантів має свої переваги та недоліки і може бути вибраний залежно від конкретних вимог до пристрою, умов експлуатації та бюджету проєкту.

2.6 Програмне забезпечення

Програмне забезпечення Arduino, що має відкритий код, спрощує процес написання та завантаження коду на плату [18]. Інтегроване середовище розробки (IDE) цієї програми підходить для всіх основних операційних систем, таких як Windows, Mac OS X та Linux, і підтримує мови програмування C і C++. Термін "скетчі" використовується для програм, створених в Arduino IDE; ці невеликі програми зберігаються у текстовому форматі з розширенням .ino. Редактор IDE має різні функції, включаючи редагування тексту, пошук та заміну, а також область повідомлень для зворотного зв'язку під час роботи з програмами. Консоль відображає повідомлення від програмного забезпечення, включаючи повністю оформлені повідомлення про помилки та іншу інформацію. Налаштована плата та послідовний порт відображаються у правому нижньому куті вікна, а панель інструментів надає можливість перевірки, завантаження, створення, відкриття та збереження програм, а також відкриття послідовного монітора.



Рисунок 2.9 – вигляд Arduino IDE

Бібліотеки додають додатковий функціонал для використання в програмах, наприклад, для роботи з обладнанням або обробки даних. Для використання бібліотеки в програмі, потрібно вибрати її в меню "Скетч" -> "Імпорт бібліотеки". Це автоматично додасть один або кілька операторів `#include` у верхню частину програми та скомпілює бібліотеку разом з програмою. Оскільки бібліотеки завантажуються на мікроконтролер разом з програмою, вони збільшують розмір програми. Якщо програмі більше не потрібна бібліотека, оператори `#include` можна видалити з верхньої частини коду.

У довідці є список доступних бібліотек. Деякі з них вже включені до програмного забезпечення Arduino, а інші можна завантажити з різних джерел або використовувати Менеджер бібліотек. Починаючи з версії 1.0.5 IDE, також є можливість імпортувати бібліотеку з zip-файлу та використовувати її у власних програмах.

В програмній структурі Arduino виокремлюють дві основні функції: `setup()`, що викликається при старті програми для початкової ініціалізації змінних та контактів, і `loop()`, яка виконує послідовний цикл операцій для активного управління платою та надання зворотного зв'язку. Функція `setup()`

активується один раз при включенні живлення або перезавантаженні, в той час як `loop()` продовжує свою роботу, забезпечуючи постійну діяльність програми на платі Arduino.

2.6.1 Мови програмування

Мова програмування є засобом спілкування програмістів з комп'ютерами, представлений набором інструкцій, які виконують конкретні завдання у таких мовах як C, C++, Java, Python та інші. Ці мови зазвичай використовуються для створення настільних програм, веб-сайтів та мобільних додатків. Особливість мови програмування високого рівня полягає у її легкості у розумінні, написанні та підтримці. Наприклад, мови, як Python, Java, JavaScript, PHP, C#, C++, створені для розробки зручних програм та веб-сайтів, що вимагає наявності компілятора або інтерпретатора для перекладу коду на машинну мову для виконання програми.

Зокрема, C++ є розширенням мови C, з властивостями об'єктно-орієнтованого та загального програмування, що дозволяє маніпулювати пам'яттю на низькому рівні. Використання C++ в програмному забезпеченні, особливо для вбудованого програмування, обмежених пристроїв та програмування великих систем, забезпечує високу швидкість, продуктивність та ефективність. Не зважаючи на це, C++ використовується практично в усіх галузях, від програмних засобів баз даних до операційних систем та ігор.

Ця мова має важливе значення в інтернеті речей (IoT), де низькі ресурси та обмежені можливості живлення вимагають ефективного програмування. Багато пристроїв IoT використовують C або C++ через їх компактність і здатність працювати в умовах з обмеженими ресурсами. Крім того, мова програмування для платформи Arduino, що базується на C/C++, є простою для освоєння та дозволяє використовувати бібліотеки AVR Libc для розробки програм для мікроконтролерів.

2.7 UkraineAlarm API

API UkraineAlarm надає інформацію в режимі реального часу про повітряні тривоги та інші надзвичайні ситуації в Україні [19]. Цей сервіс є надзвичайно важливим для інформування громадян та влади про потенційні загрози, що дозволяє їм своєчасно вживати необхідних заходів безпеки.

Основні функції API UkraineAlarm включають оповіщення в реальному часі, які надають миттєві повідомлення про повітряні тривоги та інші надзвичайні ситуації, а також можуть бути налаштовані для різних регіонів і типів загроз. Покриття сервісу охоплює всі регіони України, надаючи детальну інформацію про статус тривог у різних місцях. Доступ до даних здійснюється через різні кінцеві точки, такі як поточні тривоги, історичні дані та інформація по конкретних регіонах, з підтримкою кількох форматів даних, таких як JSON і XML, для легкого інтегрування з іншими додатками.

API UkraineAlarm також має можливість інтеграції у мобільні додатки, вебсайти та інші платформи для надання користувачам актуальної інформації, що є корисним для державних установ, екстрених служб та загального населення. Типові кінцеві точки, що надаються API UkraineAlarm, можуть включати: поточні тривоги (GET /api/v1/alerts/current), тривоги по регіонах (GET /api/v1/alerts/{region_id}) та історичні дані (GET /api/v1/alerts/history).

Приклади використання API включають інтеграцію з мобільними додатками, що сповіщають користувачів про наближення повітряних тривог і надають інструкції щодо безпеки; державні портали, що інтегрують API для надання громадянам актуальної інформації та координації екстрених служб; а також медіа та новинні ресурси, що відображають актуальні оновлення про статус повітряних тривог для інформування громадськості.

Переваги використання API UkraineAlarm включають підвищення безпеки населення завдяки своєчасним сповіщенням, що допомагають зменшити кількість постраждалих, дозволяючи людям вчасно знайти укриття; інформоване прийняття рішень, оскільки влада може використовувати дані для прийняття обґрунтованих рішень щодо реагування на надзвичайні ситуації

та розподілу ресурсів; а також підвищення обізнаності громадськості, що збільшує підготовленість населення під час кризових ситуацій.

API UkraineAlarm є життєво важливим інструментом для управління та розповсюдження оповіщень про надзвичайні ситуації в Україні. Завдяки наданню даних у режимі реального часу, він підвищує безпеку населення, підтримує роботу урядових та екстрених служб і інформує громадськість про критичні ситуації.

2.8 Висновки до розділу 2

Під час розробки апаратної частини було проведено аналіз декількох мікроконтролерів з урахуванням їхніх переваг та недоліків. Після уважного вивчення оптимальним варіантом виявився мікроконтролер ESP32 CH340 Type-C.

Також було детально розглянуто можливі компоненти та програмне забезпечення для створення модулю системи оповіщення, і були обрані наступні найоптимальніші варіанти:

- Адресні світлодіоди WS2812B RGB;
- звукова плата MAX98357;
- модуль живлення MB-102;
- мова програмування C++ для реалізації алгоритму на модулі ESP32;
- використання UkraineAlarm API для отримання актуальної інформації про ситуацію в регіоні;
- використання середовища розробки Arduino IDE.

РОЗДІЛ 3 АПАРАТНА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЄКТУ НА ESP-32, АДРЕСНИХ СВІТЛОДІОДІВ, ЗВУКОВОЇ ПЛАТИ ТА API

Складові програмно апаратного комплексу:

- Мікроконтролер ESP32;
- адресні світлодіоди WS2812B RGB;
- звукова плата MAX98357 з невеликим динаміком;
- модуль живлення MB-102;
- макетна плата (не обов'язково)
- кабелі Dupont для з'єднання прототипу приладу;
- виготовлений на 3D-принтері корпус (можливе для удосконалення приладу);
- отримати доступ від розробників UkraineAlarm (а саме код доступу для користування API та Документацію);
- середовище Arduino IDE для розробки, написання та завантаження програми на мікроконтролер.

3.1 Під'єднання усіх компонентів пристрою

Послідовність підключення компонентів.

Модуль живлення:

- Підключіть модуль живлення до джерела живлення (наприклад, через USB або інший відповідний роз'єм);
- встановіть перемикач на модулі живлення на 5V (якщо стрічка WS2812B потребує 5V).

Підключення ESP32:

- Підключіть пін GND ESP32 до GND модуля живлення;
- підключіть пін 3.3V ESP32 до 3.3V модуля живлення.

Підключення світлодіодної стрічки WS2812B:

- Підключіть пін GND стрічки до GND модуля живлення;
- підключіть пін VCC стрічки до 5V модуля живлення;

- підключіть пін DIN (дані) стрічки до обраного GPIO піну на ESP32 (наприклад, GPIO 18).

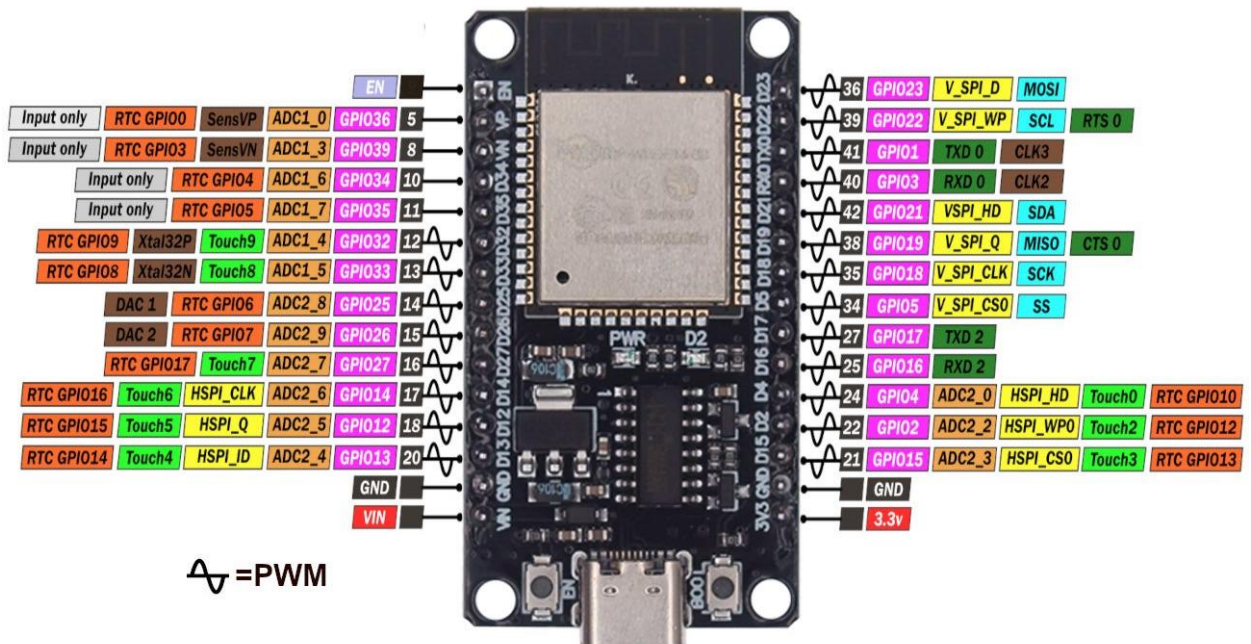


Рисунок 3.1 – схема пінів ESP32 CH340 Type-C

Підключення аудіоусилювача MAX98357A:

- Підключіть пін GND аудіоусилювача до GND модуля живлення;
- підключіть пін VIN аудіоусилювача до 5V модуля живлення;
- підключіть пін DIN аудіоусилювача до відповідного піну ESP32 для I2S сигналу (наприклад, GPIO 25);
- підключіть пін BCLK (bit clock) аудіоусилювача до піну I2S BCLK на ESP32 (наприклад, GPIO 26);
- підключіть пін LRC (left-right clock) аудіоусилювача до піну I2S LRCLK на ESP32 (наприклад, GPIO 27);
- якщо хочете керувати режимами підсилення, підключіть пін GAIN до GPIO ESP32 або залиште непідключеним для використання значення за замовчуванням;
- пін SD (Shutdown) можна підключити до GPIO ESP32 для керування вимкненням підсилювача або підключити до 5V для постійної роботи.

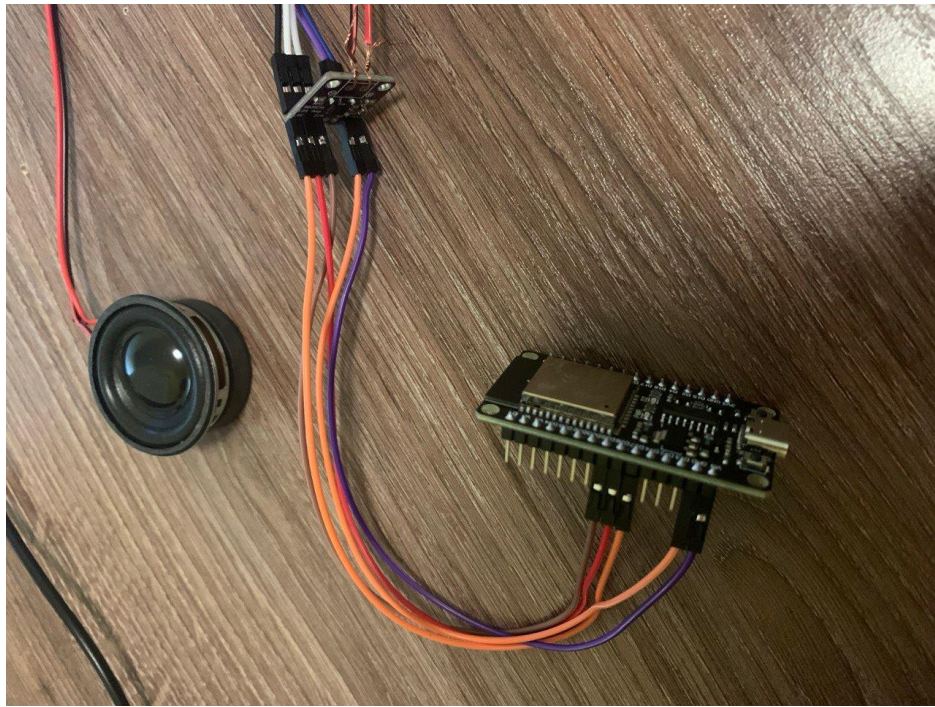


Рисунок 3.2 – з'єднання аудіоусилювача з платою ESP32

Підключення контактів між мікроконтролером та усіма компонентами приладу здійснюється з використанням кабелів типу Dupont (рис. 3.3).



Рисунок 3.3 – вигляд кабелів типу Dupont

Також під'єднання пінів усіх компонентів зазначено у табл. 3.1.

Таблиця 3.1 – Підключень пінів

Компонент	Пін на компоненті	Пін на ESP32 або модуля живлення
ESP32	GND	GND (модуль живлення)
	3.3V	3.3V (модуль живлення)
Світлодіодна стрічка WS2812B	GND	GND (модуль живлення)
	VCC	5V (модуль живлення)
	DIN	GPIO 18 (ESP32)
Аудіоусилювач MAX98357A	GND	GND (модуль живлення)
	VIN	5V (модуль живлення)
	DIN	GPIO 25 (ESP32)
	BCLK	GPIO 26 (ESP32)
	LRC	GPIO 27 (ESP32)
	GAIN	(опціонально) GND або GPIO (ESP32)
	SD	(опціонально) 5V або GPIO (ESP32)
Модуль живлення	GND	GND (всі компоненти)
	5V	5V (всі компоненти)
	3.3V	3.3V (ESP32)

Якщо не плануєте використовувати керування підсиленням, залиште пін GAIN не підключеним, або підключіть до GND для встановлення підсилення за замовчуванням.

Також якщо хочете, щоб підсилювач завжди був увімкненим, підключіть пін SD (Shutdown) до 5V. Якщо хочете керувати ним, підключіть до відповідного GPIO піну на ESP32.

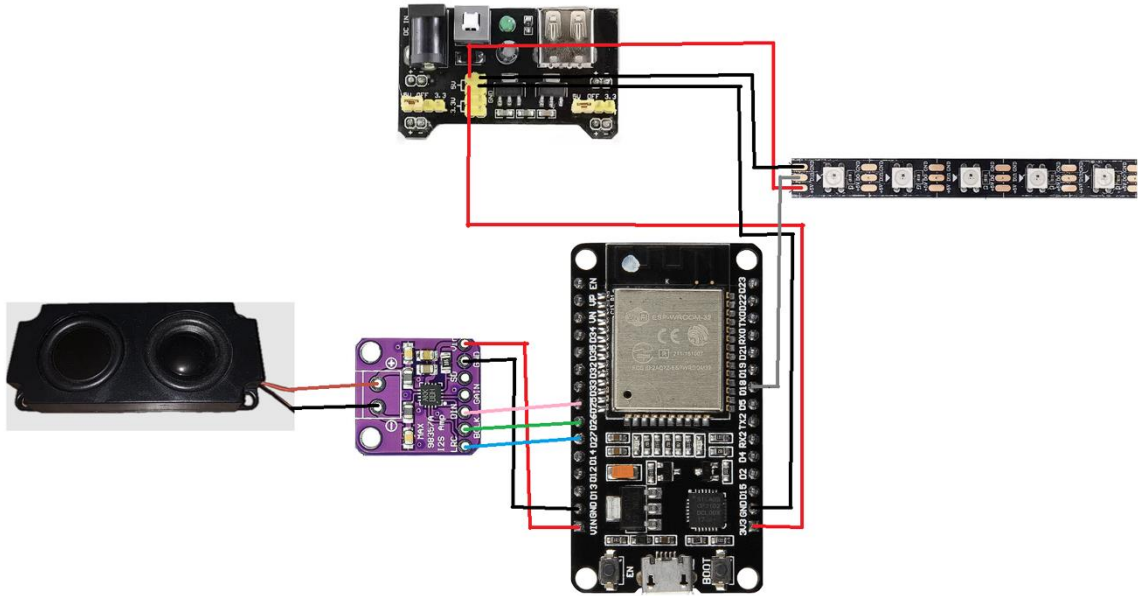


Рисунок 3.4 – схема підключення пристрою

3.2 Використання UkraineAlarm API

Перед тим як почати роботу з кодом та усією програмною частиною проєкту, треба отримати доступ до документації по цій API та сам код доступу без якого ви не зможете нею користуватися.

Для того аби отримати доступ потрібно зайти на офіційний сайт UkraineAlarm (рис 3.5) та залишити заявку на отримання доступу [20].

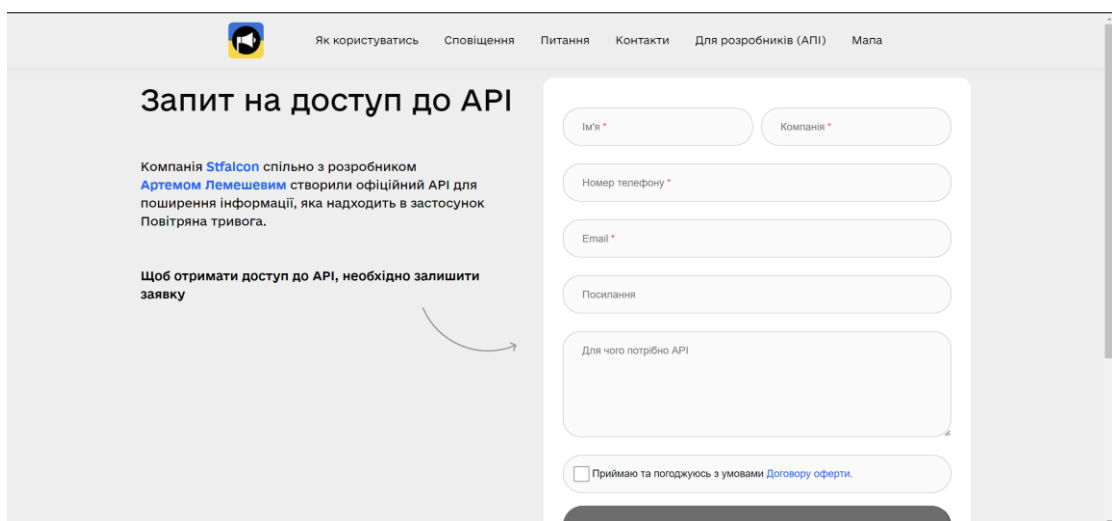


Рисунок 3.5 – сторінка сайту UkraineAlarm

Після того як вам нададуть доступ, вам на пошту прийде повідомлення з вашим кодом доступу до API та інструкцією (рис 3.6).

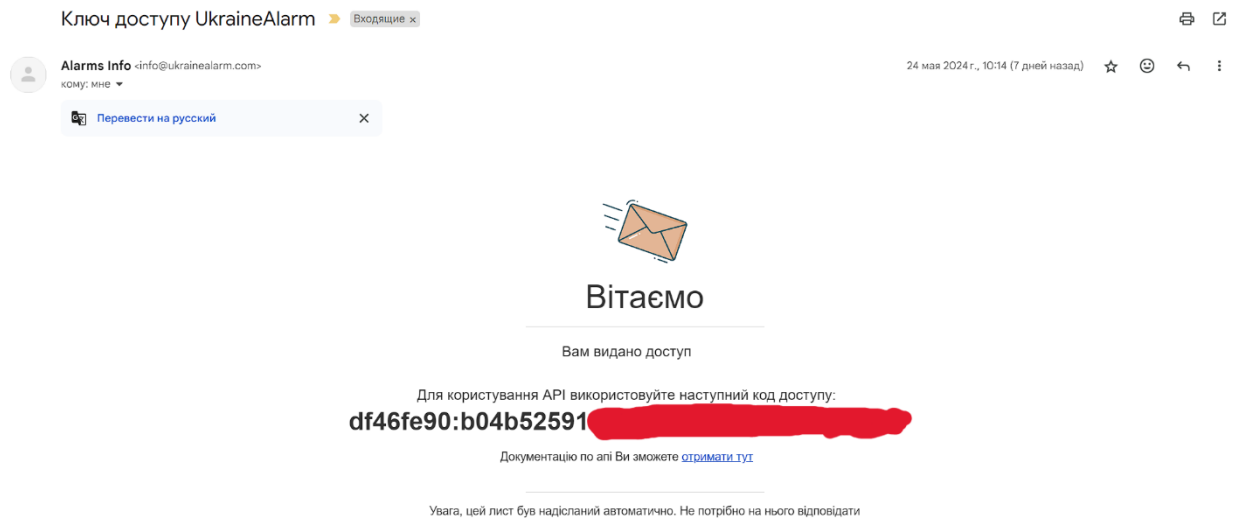


Рисунок 3.6 – повідомлення з кодом доступу

Коли ми отримали це повідомлення ми можемо спокійно використовувати це API в нашому пристрої для отримання інформації яка нас цікавить, а саме про почато, кінець та вид тривоги в заданій області або місті.

3.3 Налаштування Arduino IDE

Для програмування плати ESP32 у роботі використовується середовище Arduino IDE. Це потужний інструмент, який надає зручний інтерфейс для розробки програмного забезпечення для мікроконтролерів ESP32.

Розглянемо основні кроки програмування в Arduino IDE для даного проекту.

Завантаження та встановлення Arduino IDE:

Почніть з завантаження та встановлення Arduino IDE на ваш комп'ютер з офіційного сайту Arduino. Перейдіть на офіційний сайт Arduino IDE. Виберіть версію для вашої операційної системи (Windows, macOS, Linux) та завантажте інсталяційний файл (рис 3.7).

Downloads



Рисунок 3.7 – сторінка завантаження Arduino IDE

Після завантаження запусить інсталяційний файл і дотримуйтесь інструкцій на екрані, щоб завершити встановлення Arduino IDE на вашому комп'ютері.

Додавання підтримки плати ESP32:

Відкрийте Arduino IDE. Переходимо у меню File та знаходимо Preferences (рис 3.8).

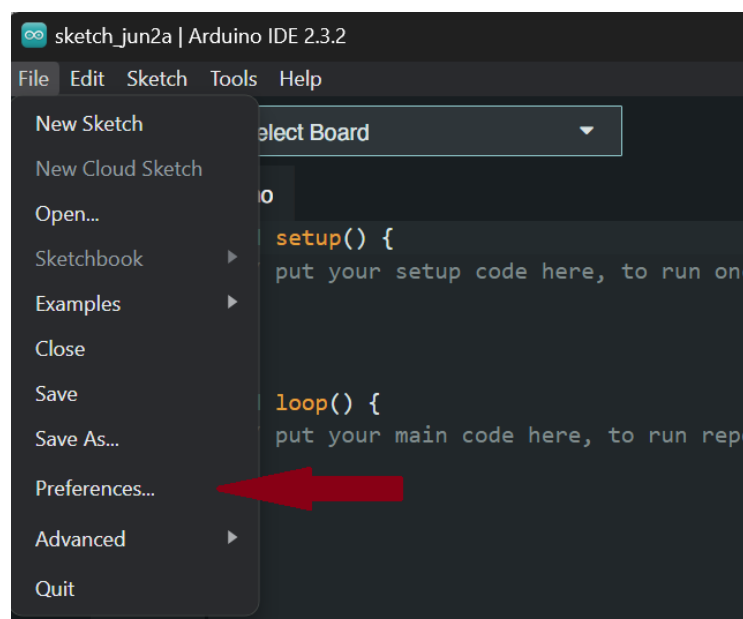


Рисунок 3.8 – меню File

У вікні Preferences знаходимо поле Additional Boards Manager (рис 3.9).
У поле вводимо URL: https://dl.espressif.com/dl/package_esp32_index.json.
Натискаємо ОК, щоб зберегти налаштування.

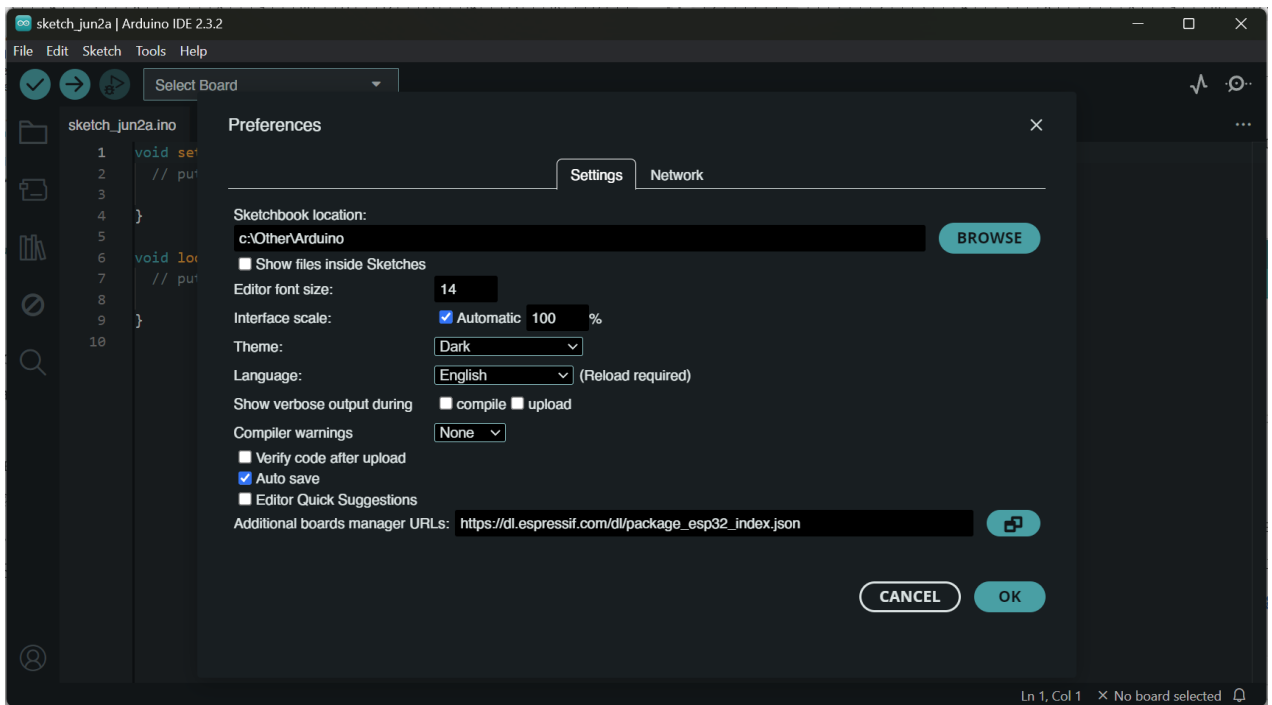


Рисунок 3.9 – вікно Preferences

Тепер встановлюємо підтримку ESP32. Переходимо у меню Tools > Board > Boards Manager (рис 3.10).

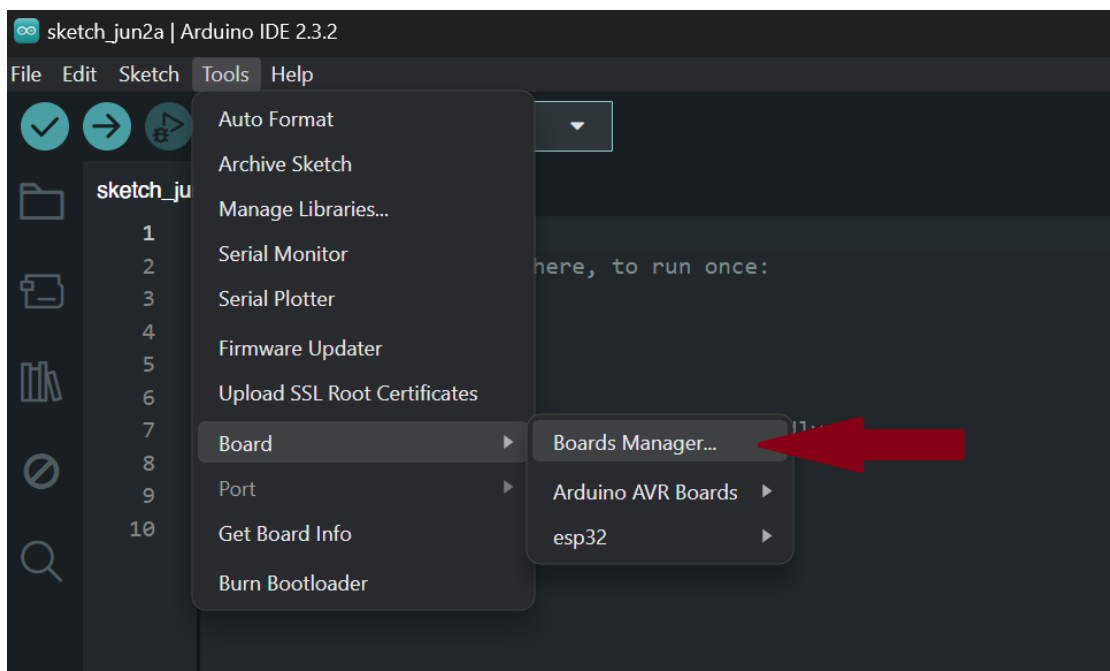


Рисунок 3.10 – Boards Manager

У вікні Boards Manager знаходимо esp32 у списку та натисніть Install. Це завантажить і встановить пакет плат ESP32 (рис. 3.11).

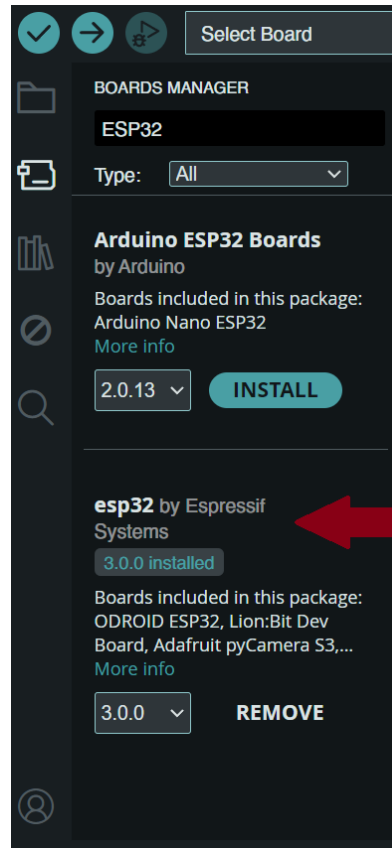


Рисунок 3.11 – завантаження пакету плат ESP32

Після встановлення пакету плат ESP32, переходимо у меню Tools > Board та вибираємо нашу плату ESP32 зі списку (наприклад, ESP32 Dev Module) (рис. 3.12).

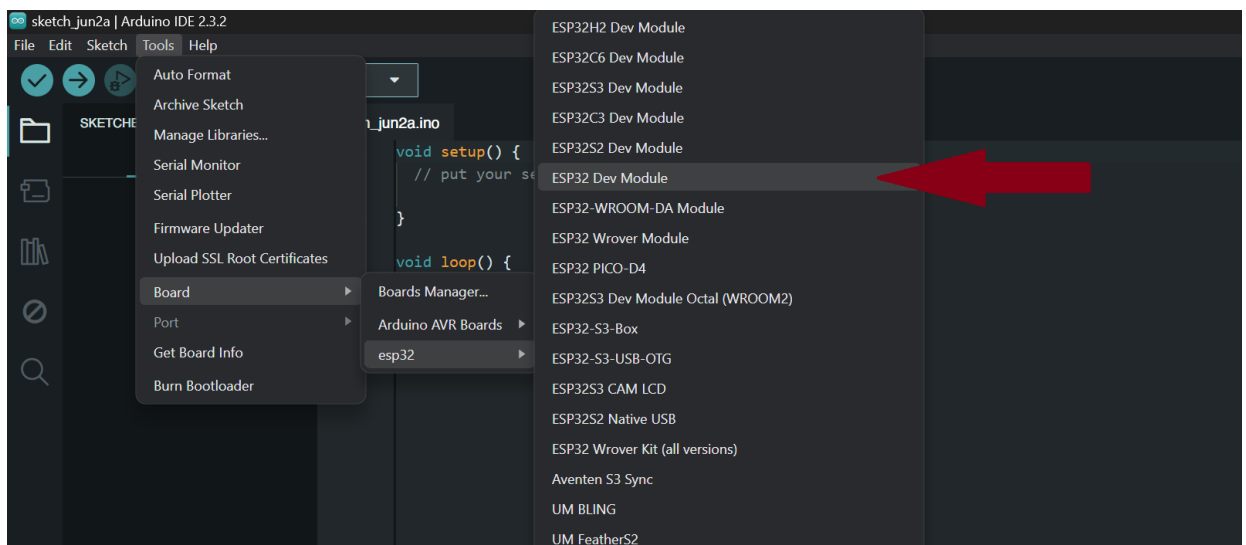


Рисунок 3.12 – вибір плати ESP32 Dev Module

3.3.1 Бібліотеки в Arduino IDE

Розробка проектів на мікроконтролерах ESP32 у середовищі Arduino IDE є захоплюючим завданням, яке потребує використання різноманітних інструментів та ресурсів. Одними з ключових інструментів, які полегшують процес програмування та роботи з компонентами, є бібліотеки Arduino IDE. Давайте розглянемо різні бібліотеки, які використовуються у проекті для створення інтерактивного табло на адресних світлодіодах.

Кожна бібліотека має свої унікальні можливості та функції, які допомагають забезпечити ефективну та функціональну роботу системи. У цьому контексті ми розглянемо бібліотеки для роботи з JSON-даними, керування адресованими світлодіодами, віддаленого оновлення програмного забезпечення, роботи з Wi-Fi, шинами зв'язку та іншими. Розглянемо їхні основні функції, використання та користь у контексті проекту.

Вивчення цих бібліотек дозволить зрозуміти, як ефективно використовувати їх у власних проектах для досягнення певних функціональних цілей та підвищення якості роботи програмних рішень для ESP32

Розглянемо ці бібліотеки:

- ArduinoJson - це потужна бібліотека для роботи з JSON-даними у середовищі Arduino IDE. Вона дозволяє легко обробляти, створювати та аналізувати JSON-дані, що широко використовуються у веб-програмуванні та обміні даними між пристроями. ArduinoJson знаходить застосування у проектах, де потрібно передавати структуровані дані, такі як налаштування, статуси сенсорів, параметри віддаленого керування тощо. Бібліотека підтримує як статичне, так і динамічне виділення пам'яті, що дозволяє ефективно використовувати обмежені ресурси мікроконтролера. Вона також забезпечує високопродуктивний парсинг і генерацію JSON, що робить її придатною для ресурсомістких проектів;

- Adafruit NeoPixel - це бібліотека, створена для управління адресованими світлодіодами (наприклад, WS2812B RGB LED), яка дозволяє створювати різні світлові ефекти, змінювати кольори та яскравість світлодіодів. Ця бібліотека широко використовується в проектах, де використовуються світлодіодні стрічки для створення інтерактивного освітлення, індикації стану, анімацій тощо. Вона надає можливість керувати кожним світлодіодом окремо, створювати складні анімації та ефекти, а також синхронізувати світлодіоди для досягнення однорідного освітлення. Крім того, бібліотека включає функції для оптимізації енергоспоживання, що є важливим для портативних проектів;
- I2S - це драйвер для роботи з аудіо через протокол I2S (Inter-IC Sound) на мікроконтролерах ESP32. Він підтримує передачу та прийом аудіо даних, забезпечуючи високу якість звуку. Використовується в проектах, де потрібна робота з аудіо, наприклад, у цифрових аудіо системах, радіо або записуючих пристроях. Драйвер I2S дозволяє підключати ESP32 до зовнішніх аудіо ЦАПів (цифро-аналогових перетворювачів) і АЦП (аналогово-цифрових перетворювачів), що забезпечує можливість створення складних аудіо систем. Бібліотека дозволяє налаштовувати різні параметри аудіо потоку, такі як частота дискретизації, кількість каналів і розрядність;
- WiFi - це вбудована бібліотека для роботи з Wi-Fi на мікроконтролерах ESP32. Вона надає функції для підключення до мережі Інтернет, налаштування параметрів мережі та обміну даними через Wi-Fi. Ця бібліотека необхідна для роботи з Wi-Fi, наприклад, для з'єднання з веб-сервером, відправки даних до хмари або отримання оновлень з Інтернету. Вона підтримує різні режими роботи, такі як STA (клієнт), AP (точка доступу) та STA+AP (комбінований режим), що дозволяє реалізовувати гнучкі мережеві топології. Бібліотека також забезпечує

захист даних через WPA/WPA2 шифрування та підтримує різні методи автентифікації;

- **WiFiClient** - це бібліотека, яка дозволяє створювати клієнтські з'єднання через TCP/IP протокол для взаємодії з сервером по мережі. Використовується в проектах, де потрібно встановити зв'язок з віддаленим сервером для обміну даними, наприклад, у системах IoT, клієнт-серверних додатках тощо. Бібліотека підтримує основні функції роботи з мережею, такі як відкриття та закриття з'єднань, відправка та отримання даних. Вона також дозволяє працювати з декількома з'єднаннями одночасно, що робить її придатною для складних мережевих додатків. Бібліотека легко інтегрується з іншими мережевими бібліотеками, такими як **HTTPClient**, що дозволяє створювати комплексні мережеві рішення;
- **HTTPClient** - це бібліотека для здійснення HTTP-запитів у середовищі Arduino IDE. Вона підтримує методи GET, POST, PUT та інші, дозволяючи взаємодіяти з веб-сервісами, API та віддаленими серверами. Використовується в проектах, де необхідно отримувати або надсилати дані через HTTP протокол, наприклад, для доступу до веб-ресурсів, RESTful API або відправки телеметрії. Бібліотека підтримує роботу з HTTP заголовками, дозволяючи налаштовувати запити відповідно до вимог сервера. Вона також надає засоби для обробки відповідей сервера, включаючи коди стану та отримані дані. **HTTPClient** підтримує захищені з'єднання через HTTPS, що забезпечує безпеку даних під час передачі.

Використання цих бібліотек у проекті для створення інтерактивного табло на адресних світлодіодах дозволяє значно спростити процес розробки та підвищити функціональність системи. Кожна з них забезпечує конкретні можливості, необхідні для роботи з мікроконтролерами ESP32, такі як керування світлодіодами, робота з мережами, обробка аудіо та даних у

форматі JSON. Це дозволяє розробникам зосередитися на створенні більш складних та інтерактивних проєктів, використовуючи зручні та ефективні інструменти.

Завантаження бібліотек.

Переходимо у меню Sketch > Include Library > Manage Libraries (рис. 3.13). У вікні Library Manager знаходимо всі бібліотеки що нас цікавлять та натискаємо Install для встановлення бібліотеки.

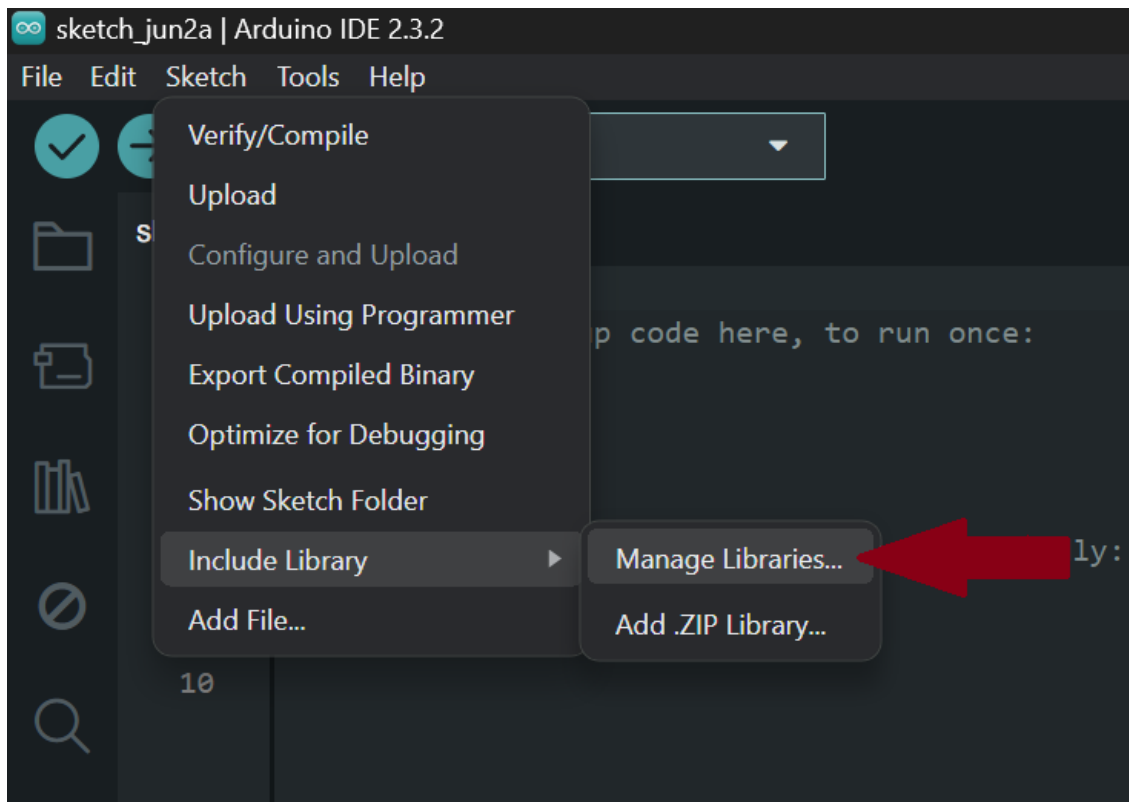


Рисунок 3.13 – вибір плати ESP32 Dev Module

3.4 Завантаження коду на ESP32

Для завантаження коду на плату ESP32, спочатку підключимо плату до комп'ютера за допомогою USB-кабеля. Переконайтеся, що кабель не лише живить плату, але також підтримує передачу даних (деякі кабелі призначені тільки для зарядки і не підтримують передачу даних). Якщо ви підключаєте ESP32 до комп'ютера вперше, можливо, знадобиться встановити драйвери. Більшість плат ESP32 використовують чіп CH340 або CP2102 для USB-

з'єднання. Ви можете знайти драйвери на сайті виробника вашої плати або на сторінках, що присвячені драйверам для CH340 або CP2102.

Після підключення плати відкрийте Arduino IDE. Перейдіть у меню Tools > Port. Ви побачите список доступних портів. Виберіть COM-порт, до якого підключена ваша плата ESP32. Зазвичай, якщо у вас підключена тільки одна плата, новий порт з'явиться у списку після підключення плати. Наприклад, виберіть COM3 (Windows) або /dev/ttyUSB0 (Linux) або /dev/cu.SLAB_USBtoUART (macOS).

Перейдіть у меню Tools > Board і виберіть вашу плату ESP32 зі списку (наприклад, ESP32 Dev Module). Після вибору порту і плати натисніть кнопку Upload у верхньому лівому куті Arduino IDE. Ця кнопка позначена стрілкою вправо (вказує на функцію завантаження коду). Arduino IDE почне компіляцію вашого коду. Ви побачите повідомлення в нижній частині вікна IDE, яке вказує на процес компіляції та завантаження. Якщо виникають помилки під час компіляції, переконайтеся, що ваш код правильний та всі необхідні бібліотеки встановлені.

Після завершення компіляції, Arduino IDE завантажить код на плату ESP32. Ви побачите повідомлення "Connecting..." в консолі IDE. Можливо, вам доведеться натиснути кнопку BOOT на вашій платі ESP32, щоб увійти в режим завантаження. Тримайте цю кнопку натиснутою, доки Arduino IDE не почне завантаження коду (якщо цього вимагає ваша плата). Після успішного завантаження коду ви побачите повідомлення Done uploading. у нижній частині Arduino IDE. Плата ESP32 автоматично перезавантажиться та почне виконувати завантажений код.

3.5 Опис програмної частини приладу

Цей код є програмою для платформи Arduino, яка здійснює наступні функції: підключення до Wi-Fi мережі, завантаження та обробка даних з API, налаштування світлодіодної стрічки та I2S для звуку, а також виконання основного циклу з обробкою отриманих даних. Використовуються бібліотеки

WiFi і HTTPClient для підключення до Wi-Fi та здійснення HTTP-запитів, підключається до мережі з SSID "Apple_Wi-Fi" та паролем "88888887". Програма отримує дані з API за URL "https://api.ukrainealarm.com/api/v3/alerts/17" за допомогою HTTP GET-запиту та використовує ключ API для авторизації.

Світлодіодна стрічка WS2812 з 10 пікселями ініціалізується, а I2S налаштовується для передачі звуку з частотою 44100 Гц і 16-бітними зразками. Кожні 20 секунд відправляється запит до API і обробляються отримані дані. Залежно від типу попередження (alertType), змінюються кольори світлодіодів і відтворюються звукові сигнали. Типи попереджень включають "UNKNOWN", "NoAlerts", "AIR", "ARTILLERY", "URBAN_FIGHTS", "CHEMICAL", "NUCLEAR". Відповідні кольори для кожного типу попередження та звукові сигнали налаштовані в програмі (рис 3.14).

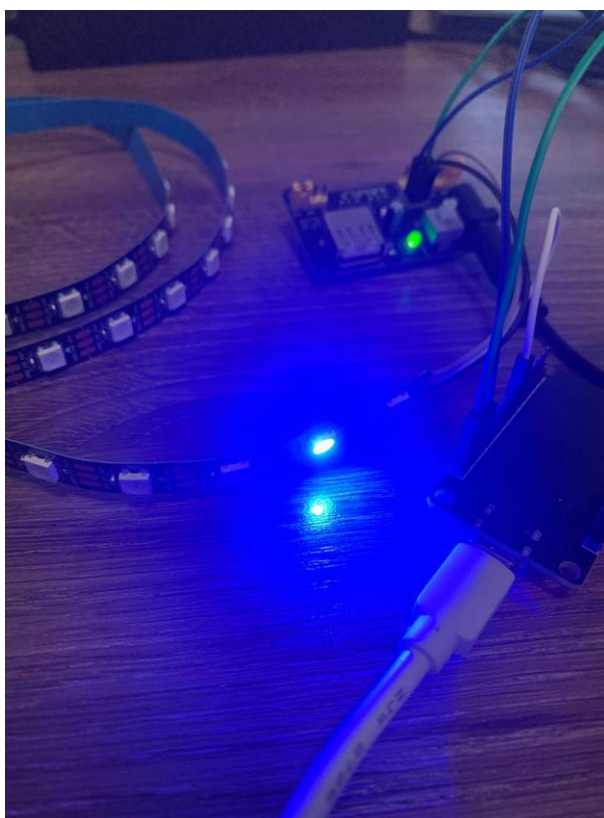


Рисунок 3.14 – сповіщення про повітряну небезпеку

Також реалізован вивід помилки при не отриманні приладом інформації від API. Загорається перший світлодіод білим кольором (рис 3.15).

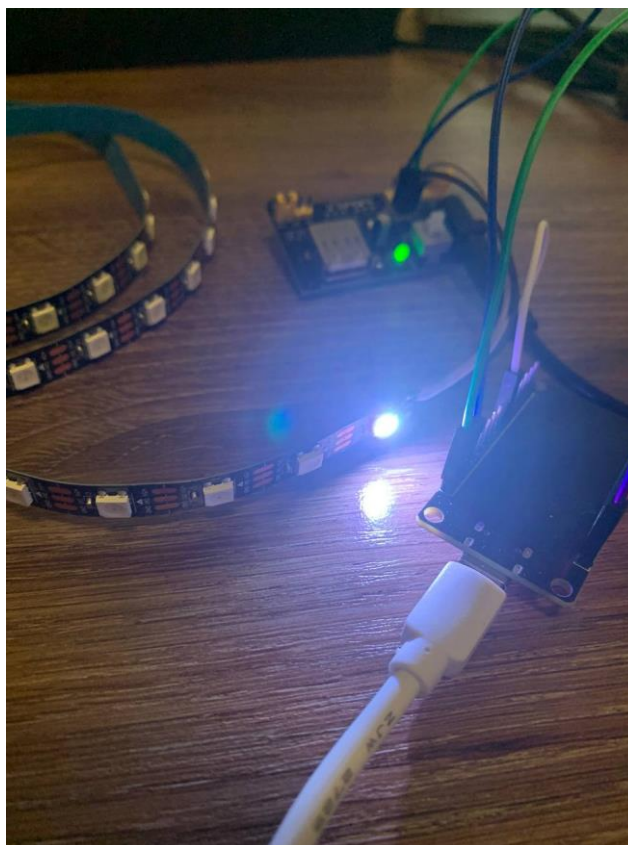


Рисунок 3.15 – вивід помилики при отриванні інформації

Програма також включає допоміжні функції: `playTone(uint16_t frequency, uint32_t duration)` для генерації звукового сигналу заданої частоти та тривалості, `getApiData(const char* url)` для виконання HTTP GET-запиту до зазначеного URL і повернення отриманих даних, `sendDataToServer(const String& data)` для відправлення даних на локальний сервер для налагодження, та `processJsonData(const String& jsonData)` для обробки отриманих JSON даних і повернення типу попередження. Програма реалізована для моніторингу сигналів тривоги, відображення їх за допомогою світлодіодів та звукового сигналу, що може бути корисним для оповіщення про різні небезпеки.

3.6 Можливі модифікації приладу

На основі отриманого приладу можна зробити кілька модифікацій для покращення функціональності та ефективності пристрою. По-перше, можна розглянути використання альтернативних видів живлення компонентів, що збільшить автономність пристрою, наприклад при живленні від батареї. Заміна

існуючих компонентів на більш енергоефективні допоможе зменшити споживання енергії. Та можливе встановлення перемикача живлення, для варіанта коли пристрій буде живитися від аккумулятора.

По-друге, можна розширити світлодіодну стрічку, збільшивши кількість світлодіодів. Це дозволить використовувати більше індикаторів для різних станів або попереджень. Для цього потрібно адаптувати код для роботи з більшою кількістю світлодіодів, щоб вони відповідали новим вимогам.

Також варто розглянути покращення звукової системи, наприклад, використання більш потужного динаміка. Це буде корисно в умовах шумного середовища, де важливо мати чіткий звуковий сигнал.

Додатково, можна додати інтерфейси для комунікації з іншими пристроями, такі як Bluetooth. Це дозволить інтегрувати прилад у розумний будинок або інші IoT системи, що значно підвищить його універсальність.

Захист від зовнішніх впливів також є важливим аспектом. Використання корпусу, який захищає електроніку від пилу та вологи, збільшить надійність пристрою при використанні в різних умовах. Це особливо актуально для пристроїв, що працюють на відкритому повітрі або в середовищах з підвищеною вологістю.

Оптимізація коду допоможе зменшити затримки та підвищити швидкість обробки даних. Можна використовувати багатопоточність або інші методи для покращення продуктивності пристрою. Наприклад, розділення обробки даних і керування світлодіодами на різні потоки дозволить зменшити час реакції на події.

Усі ці модифікації допоможуть зробити пристрій більш універсальним та ефективним для різних завдань, підвищивши його надійність та функціональність.

3.7 Висновки до розділу 3

У процесі розробки інтерактивного табло на базі мікроконтролера ESP32 було використано різноманітні бібліотеки Arduino IDE для спрощення

програмування та забезпечення багатофункціональності системи. До них входять ArduinoJson для роботи з JSON-даними, Adafruit NeoPixel для керування світлодіодами, I2S для обробки аудіо даних, WiFi та WiFiClient для підключення до мережі, і HTTPClient для здійснення HTTP-запитів.

Основні етапи роботи включали завантаження відповідних бібліотек через Library Manager в Arduino IDE, підключення ESP32 до комп'ютера та встановлення драйверів, вибір відповідного порту і плати в Arduino IDE, а також завантаження коду на плату через USB-з'єднання та налагодження можливих помилок.

Сам код включав різні функції, такі як підключення до Wi-Fi мережі, здійснення HTTP-запитів для отримання даних про безпеку, обробка цих даних, керування світлодіодами та аудіосистемою. Також були реалізовані функції для відтворення звукових сигналів та обробки JSON даних.

Потенційні модифікації пристрою включають використання альтернативних джерел живлення для підвищення автономності, розширення світлодіодної стрічки, покращення звукової системи, додавання інтерфейсів для комунікації з іншими пристроями, захист пристрою від зовнішніх впливів та оптимізація коду для підвищення продуктивності.

Усі ці кроки спрямовані на створення більш універсального та ефективного пристрою для моніторингу та оповіщення про безпеки, а також для подальшого розвитку та модифікацій.

ВИСНОВКИ

У ході написання моєї бакалаврської роботи було проведено детальний аналіз різних джерел та патентних робіт, що підтвердило доцільність створення інтерактивного табло на базі адресних світлодіодів. Було розроблено поетапний план розвитку цього проєкту.

Аналітичний огляд наявних рішень продемонстрував, що використання ESP32 є найбільш ефективним для цього проєкту завдяки його функціональним можливостям, зокрема швидкості, компактності та підтримці різних компонентів, таких як світлодіодні стрічки і різні протоколи передачі даних.

Для реалізації проєкту було обрано Arduino IDE через її значну підтримку бібліотек і простоту використання, що значно полегшує розробку та інтеграцію різноманітних функціональних модулів.

Створення цього приладу дозволяє розробити економічно вигідне рішення для систем сповіщення про тривоги або інших інформаційних систем. Зібраний модуль здатний ефективно виконувати свої функції, хоча можливі оновлення компонентів для підвищення продуктивності та якості.

Після завершення роботи було створено програмно-апаратний прилад інтерактивного табло на базі ESP32 для відображення сповіщень з використанням API, керування світлодіодною стрічкою та відтворення звукових сигналів.

Зібрано прототип програмно-апаратного модуля для відображення сповіщень із використанням світлодіодів та звукових сигналів. Продемонстровано переваги модуля ESP32 у порівнянні з іншими моделями цієї серії.

Апаратно-програмний модуль може працювати від різноманітних джерел живлення. Цей проєкт можна інтегрувати в ширший комплекс, використовуючи додаткові датчики та модулі. Розробка може сприяти зростанню популярності WiFi-модулів ESP32 та технологій сигналізації тривоги, заохочуючи більшу кількість людей до їх застосування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Led display and method for operating an led display : patent WO/2020/053137 Germany : G09G 3/3225 2016.1 G09G 3/20 2006.1. No. PCT/EP2019/073974 ; applied on 09.09.2019 ; published on 19.03.2020. URL: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2020053137&_cid=P11-LVZGF9-70963-1.
2. Alarm siren : patent 89114069 eu : G10K 7/06. No. 0411168 ; applied on 29.07.1989 ; published on 06.02.1991, Bulletin no. 89114069. 7 p. URL: https://patentscope.wipo.int/search/ru/detail.jsf?docId=EP12116422&_cid=P11-LVZFUN-57913-1.
3. Led light address emitter, vehicle networking system and electronic map drawing method : patent 103400506 China : G08G 1/13 H04L 29/08. No. 201310325601.5 ; applied on 30.07.2013 ; published on 10.02.2016. URL: https://patentscope.wipo.int/search/ru/detail.jsf?docId=CN97898734&_cid=P11-LVZGZZ-82941-1.
4. Riley T. M. Multiple images as a function of leds viewed during vibratio. *Sage journals*. 1977. Vol. 19, no. 1. URL: <https://doi.org/10.1177/00187208770190010>.
5. Meng M. Application programming interface documentation: what do software developers want?. *Sage journals*. 2017. Vol. 48, no. 3. URL: <https://journals.sagepub.com/doi/10.1177/0047281617721853>.
6. Defensebridge. Anti-Aircraft warfare: safeguarding the skies from aerial threats | defensebridge. *Homepage / Defensebridge*. URL: <https://defensebridge.com/article/anti-aircraft-warfare-safeguarding-the-skies-from-aerial-threats.html>.
7. Дмитро. Мапа тривоги своїми руками на базі esp32! *Друкарня*. URL: <https://drukarnia.com.ua/articles/мапа-тривоги-своими-руками-на-базі-esp32-гайд-для-новачків-NW25t#heading-2-1523>.

8. Neil Cameron. Electronics Projects with the ESP8266 and ESP32: Building Web Pages, Applications, and WiFi Enabled Devices. December 18, 2020, p. 723.
9. Craig Hunt. TCP/IP Network Administration (3rd Edition; O'Reilly Networking). April 1, 2002, p. 746.
10. Cuno Pfister. Getting Started with the Internet of Things: Connecting Sensors And Microcontrollers To The Cloud. June 21, 2011, p. 194.
11. Sayak Boral. Article: What is ESP32 and Why Is It Best for IoT Projects? May 25, 2020. URL: <https://www.iottechrends.com/what-is-esp32/>.
12. A New Dimension of Air-Based Threats | DGAP. *German Council on Foreign Relations / DGAP*. URL: <https://dgap.org/en/research/publications/new-dimension-air-based-threats>.
13. Cameron N. Electronics Projects with the ESP8266 and ESP32. Berkeley, CA : Apress, 2021. URL: <https://doi.org/10.1007/978-1-4842-6336-5>.
14. The complete ESP32 projects guide | espressif systems. *Wireless SoCs, Software, Cloud and AIoT Solutions / Espressif Systems*. URL: <https://www.espressif.com/en/news/complete-esp32-projects-guide>.
15. 160+ ESP32 projects, tutorials and guides with arduino IDE | random nerd tutorials. *Random Nerd Tutorials*. URL: <https://randomnerdtutorials.com/projects-esp32/>.
16. Bielicki T. Install software on computer : arduino ide: stm32 arduino ide. Independently Published, 2021.
17. Marco Schwartz. Internet of Things with ESP8266. 2016, 226p
18. Офіційний сайт Arduino. URL: <https://www.arduino.cc>. (дата звернення: 21.03.2022).
19. Головна | повітряна тривога. *Повітряна Тривога*. URL: <https://www.ukrainealarm.com/>.
20. Запит на доступ до API. *UkraineAlarm*. URL: <https://api.ukrainealarm.com/>.

21. 熊宇. Design of remote alarm system for gas detection in living room based on ESP32. *Computer science and application*. 2019. Vol. 09, no. 11. P. 2010–2019. URL: <https://doi.org/10.12677/csa.2019.911226>.
22. Hazard warning. *Anaesthesia*. 1983. Vol. 38, no. 3. P. 310. URL: <https://doi.org/10.1111/j.1365-2044.1983.tb14022.x>.
23. 48 × 48 pixelated addressable full-color micro display based on flip-chip micro LEDs / Y. Li et al. *Applied optics*. 2019. Vol. 58, no. 31. P. 8383. URL: <https://doi.org/10.1364/ao.58.008383>.
24. Volders L. ESP32 simplified. Lulu Press, Inc., 2020.
25. Cameron N. ESP32 formats and communication. Berkeley, CA : Apress, 2023. URL: <https://doi.org/10.1007/978-1-4842-9376-8>.
26. Abed I., Naser H. ESP32 microcontroller based smart power meter system design and implementation. *Al-Rafidain engineering journal (AREJ)*. 2020. Vol. 25, no. 2. P. 137–145. URL: <https://doi.org/10.33899/rengj.2020.127111.1038>.
27. Mini alarm siren. *Elderly care*. 1995. Vol. 7, no. 5. P. 30. URL: <https://doi.org/10.7748/elc.7.5.30.s41>.
28. Visualizing air raid sirens in ukraine. *Observable*. URL: <https://observablehq.com/@mourner/sirens>.
29. DIY Portable Air Raid Alert – портативний пристрій для сигналізації повітряної тривоги, який можна під'єднати до зовнішньої сирени. *DOU*. URL: <https://dou.ua/forums/topic/48082/>.

ДОДАТОК А

ДОВІДКА ПРО ПЕРЕВІРКУ НА УНІКАЛЬНІСТЬ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

бакалаврської кваліфікаційної роботи на тему:
«Інтерактивне табло на адресних світлодіодах»

студента спеціальності 123 «Комп'ютерна інженерія», 405 групи
Петіков Володимир Вікторович
прізвище, ім'я, по-батькові

Перевірку тексту здійснено сервісом: онлайн-сервіс Unicheck

Результат перевірки тексту бакалаврської кваліфікаційної роботи: схожість
складає 2,96 %.

The screenshot shows the Unicheck report interface. At the top, it displays the Unicheck logo and the document name 'Петіков - КБР - плагіат'. Below this, there are two columns of metadata: user name (Борис Салтовський), check ID (1016335066), check date (08.06.2024 14:03:10 EEST), document type (Doc vs Internet), report date (08.06.2024 15:05:26 EEST), and user ID (100012260). A summary bar indicates a 2.96% similarity, with a breakdown of 1% from internet sources and 405 matches. It also shows 0% for quotes and 0% for removed sources. A 'Модифікації' (Modifications) section indicates 1 symbol was replaced.

Здобувач:

підпис

В. В. Петіков
ініціали, прізвище

Керівник:

Ст. викл. каф. комп. інженерії

_____ Б. Г. Салтовський
підпис ініціали, прізвище

Дата: «__» _____ 2024 р.

ДОДАТОК Б

КОД ДЛЯ ІНТЕРАКТИВНОГО ТАБЛА НА АДРЕСНИХ СВІТЛОДІОДАХ

```
#include <Adafruit_NeoPixel.h>
#include <driver/i2s.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>

// Налаштування Wi-Fi, замінюємо xxxx на ваш пароль від Wi-Fi та міняємо назву на свою
const char* ssid = "Apple_Wi-Fi";
const char* password = "xxxxxxxx";

// Налаштування API, замінюємо xxxx на ваш API ключ
const char* apiUrl = "https://api.ukrainealarm.com/api/v3/alerts/17";
const char* apiKey = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";

// Локальні налаштування сервера для відладки
const char* server_ip = "192.168.1.189";
const int server_port = 3338;

// Налаштування світлодіодної стрічки
#define PIN_WS2812 18
#define NUM_PIXELS 10
#define I2S_DIN 25
#define I2S_BCLK 26
#define I2S_LRCLK 27

Adafruit_NeoPixel strip(NUM_PIXELS, PIN_WS2812, NEO_GRB + NEO_KHZ800);
WiFiClient client;

void setup() {
  Serial.begin(115200);

  // Підключення до Wi-Fi
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
```

```
}

// Налаштування I2S для звукового виводу
i2s_config_t i2s_config = {
    .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_TX),
    .sample_rate = 44100,
    .bits_per_sample = I2S_BITS_PER_SAMPLE_16BIT,
    .channel_format = I2S_CHANNEL_FMT_RIGHT_LEFT,
    .communication_format = I2S_COMM_FORMAT_I2S,
    .intr_alloc_flags = ESP_INTR_FLAG_LEVEL1,
    .dma_buf_count = 8,
    .dma_buf_len = 64,
    .use_apll = false,
    .tx_desc_auto_clear = true,
    .fixed_mclk = 0
};

i2s_pin_config_t pin_config = {
    .bck_io_num = I2S_BCLK,
    .ws_io_num = I2S_LRCLK,
    .data_out_num = I2S_DIN,
    .data_in_num = I2S_PIN_NO_CHANGE
};

i2s_driver_install(I2S_NUM_0, &i2s_config, 0, NULL);
i2s_set_pin(I2S_NUM_0, &pin_config);
}

void loop() {
    // Запит даних з API кожні 20 секунд
    String jsonData = getApiData(apiUrl);

    if (jsonData.length() == 0) {
        // Обробка пустих даних
        strip.setPixelColor(0, strip.Color(255, 255, 255));
        strip.show();
        delay(2000);
    } else {
        // Обробка отриманих даних
        String alertType = processJsonData(jsonData);
        if (alertType == "UNKNOWN") {
            strip.setPixelColor(0, strip.Color(255, 0, 0));
        }
    }
}
```

```
strip.show();
playTone(1000, 100);
delay(1000);
playTone(1000, 500);
} else if (alertType == "NoAlerts") {
strip.setPixelColor(0, strip.Color(0, 0, 0));
strip.show();
delay(1000);
} else if (alertType == "AIR") {
strip.setPixelColor(1, strip.Color(0, 0, 255));
strip.show();
playTone(1000, 100);
delay(1000);
playTone(1000, 500);
} else if (alertType == "ARTILLERY") {
strip.setPixelColor(2, strip.Color(255, 0, 0));
strip.show();
playTone(1000, 100);
delay(1000);
playTone(1000, 500);
} else if (alertType == "URBAN_FIGHTS") {
strip.setPixelColor(3, strip.Color(255, 0, 255));
strip.show();
playTone(1000, 100);
delay(1000);
playTone(1000, 500);
} else if (alertType == "CHEMICAL") {
strip.setPixelColor(4, strip.Color(255, 255, 0));
strip.show();
playTone(1000, 100);
delay(1000);
playTone(1000, 500);
} else if (alertType == "NUCLEAR") {
strip.setPixelColor(5, strip.Color(0, 255, 0));
strip.show();
playTone(1000, 100);
delay(1000);
playTone(1000, 500);
} else {
strip.setPixelColor(6, strip.Color(0, 255, 0));
}
}
```



```
// Очищення світлодіодів перед повторенням циклу
strip.clear();
strip.show();
delay(10000);
}

void playTone(uint16_t frequency, uint32_t duration) {
    const int sampleRate = 44100;
    const int samples = (sampleRate * duration) / 1000;
    const int16_t amplitude = 30000;

    for (int i = 0; i < samples; i++) {
        int16_t sample = amplitude * sin(2 * PI * frequency * i / sampleRate);
        size_t bytes_written;
        i2s_write(I2S_NUM_0, &sample, sizeof(sample), &bytes_written, portMAX_DELAY);
    }
}

String getApiData(const char* url) {
    HTTPClient http;
    String payload;

    http.begin(url);
    http.addHeader("Accept", "application/json");
    http.addHeader("Content-Type", "application/json");
    http.addHeader("Authorization", apiKey);

    int httpCode = http.GET();
    if (httpCode > 0) {
        if (httpCode == HTTP_CODE_OK) {
            payload = http.getString();
        } else {
            Serial.printf("HTTP GET request failed with code: %d\n", httpCode);
        }
    } else {
        Serial.printf("HTTP GET request failed: %s\n", http.errorToString(httpCode).c_str());
    }

    http.end();
    return payload;
}
```

```
void sendDataToServer(const String& data) {
    if (!client.connected()) {
        if (client.connect(server_ip, server_port)) {
            client.println(data);
        }
    } else {
        client.println(data);
    }
}

String processJsonData(const String& jsonData) {
    StaticJsonDocument<1024> doc;
    DeserializationError error = deserializeJson(doc, jsonData);

    if (error) {
        Serial.print("deserializeJson() failed: ");
        Serial.println(error.f_str());
        return "";
    }

    const char* exampleValue = doc[0]["activeAlerts"][0]["type"];
    sendDataToServer(exampleValue);

    if (strcmp(exampleValue, "UNKNOWN") == 0) {
        return "UNKNOWN";
    } else if (strcmp(exampleValue, "AIR") == 0) {
        return "AIR";
    } else if (strcmp(exampleValue, "ARTILLERY") == 0) {
        return "ARTILLERY";
    } else if (strcmp(exampleValue, "URBAN_FIGHTS") == 0) {
        return "URBAN_FIGHTS";
    } else if (strcmp(exampleValue, "CHEMICAL") == 0) {
        return "CHEMICAL";
    } else if (strcmp(exampleValue, "NUCLEAR") == 0) {
        return "NUCLEAR";
    } else {
        return "NoAlerts";
    }
}
```