

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


Інтерактивна мапа тривоги України на базі

мікроконтролера ESP32

Спеціальність 123 Комп'ютерна інженерія

123 – КБР.01 – 405.22010603

Студент


_____ Р. О. Горгун
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

Горгуну Руслану Олександровичу

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Інтерактивна мапа тривоги України на базі мікроконтролера ESP32

Затверджена наказом по ЧНУ ім. Петра Могили від 30.01.2024 № 17.

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

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

Створення апаратного та програмного забезпечення для інтерактивної мапи тривоги України на базі мікроконтролера ESP32, яке дозволить оперативно інформувати населення про актуальні небезпеки та тривоги. Вхідними даними роботи є специфікація вимог, що описує характеристики зазначеного апаратного та програмного забезпечення, а також дані з систем сповіщення та датчиків, які будуть інтегровані для отримання інформації в режимі реального часу.

4. Перелік питань, що підлягають розробці

Провести аналіз існуючих систем інформування про тривоги та небезпеки; вивчити можливості мікроконтролера ESP32 для створення інтерактивних мап; розробити програмне забезпечення для інтерактивної мапи тривоги; інтегрувати мапу з системами сповіщення для отримання даних у режимі реального часу.

5. Перелік графічних матеріалів

Зображення ESP32, звукової плати, світлодіодів.

Зображення схеми пінів ESP32, звукової плати, світлодіодів.

Зображення блок-схеми роботи пз, зображення uml діаграми прецедентів.

Зображення макетної схеми, принципової електричної схеми та схеми розміщення на друкованій платі.

6. Завдання до спеціальної частини

Проаналізувати та розробити рекомендації з охорони праці при створенні і експлуатації інтерактивної мапи тривоги України на базі мікроконтролера ESP32. Робота включає аналіз потенційних небезпек, організацію безпечних умов праці, заходи захисту при роботі з електронними компонентами, пожежну безпеку, медичні аспекти та ергономіку, а також психологічний клімат і стресостійкість.

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

Горгун Руслан Олександрович

(прізвище, ім'я, по батькові студента)



(підпис)

Дата видачі завдання « ____ » _____ 20 ____ р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Інтерактивна мапа тривоги України на базі мікроконтролера ESP32

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КР	21.12.23	25.12.23	Виконав
2	Огляд літератури за темою роботи	16.01.24	20.02.24	Виконав
3	Складання календарного плану БКР	22.02.24	08.03.24	Виконав
4	Аналіз предметної області	25.02.24	12.03.24	Виконав
5	Розробка проєктних рішень	25.02.24	09.03.24	Виконав
6	Моделювання та конструювання АПЗ	20.02.24	27.02.24	Виконав
7	Перевірка працездатності, тестування та апробація розробленого АПЗ, аналіз результатів тестування	01.03.24	04.03.24	Виконав
8	Відгук керівника КР	07.06.24	07.06.24	Виконав
9	Оформлення БКР та презентації	15.03.24	21.04.24	Виконав
10	Попередній захист	28.05.24	04.06.24	Виконав
11	Рецензування	08.06.24	12.06.24	Виконав
12	Завершення оформлення КР та презентації	10.06.24	13.06.24	Виконав
13	Захист бакалаврської кваліфікаційної роботи	24.06.24	26.06.24	Виконав

Розробив здобувач ВО Горгун Руслан Олександрович
(прізвище, ім'я, по батькові)

(підпис)

« ____ » _____ 20__ р.

Керівник роботи Ст. викл. каф. комп. інженерії Б. Г. Салтовський
(посада, прізвище, ім'я, по батькові)

_____ (підпис)

« ____ » _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи
«Інтерактивна мапа тривог України на базі мікроконтролера ESP32»
Студент 405 гр.: Горгун Руслан Олександрович
Керівник: Ст. викл. Салтовський Б. Г.

Бакалаврська робота присвячена розробці інтерактивної мапи тривог України на базі мікроконтролера ESP32. Метою є створення пристрою-інформеру про тривоги, який буде доступним за ціною.

Апаратна частина складається з мікроконтролера ESP32, адресних світлодіодів WS2812B, звукової I2S плати MAX98357 та динаміку. Програмний код реалізовано на C++. Код обробляє дані отримані з API та керує світлодіодами, показуючи кольором стан тривоги у кожній області.

Пояснювальна записка бакалаврської роботи складається зі вступу, чотирьох розділів, висновків та двох додатків. У першому розділі проведено огляд сучасних систем оповіщення населення, аналіз вимог до інтерактивної мапи тривог, а також розглянуто апаратні та програмні компоненти, необхідні для її створення. Другий розділ присвячено проектуванню та моделюванню пристрою. Обґрунтовано обрання технологій, розроблено схему з'єднання компонентів, розглянуто ідеї щодо вдосконалення пристрою, та принцип роботи програмного забезпечення. У третьому розділі описано апаратно-програмну реалізацію системи. Подано детальний опис програмної частини розробки. Четвертий розділ присвячено охороні праці. Висновки містять підсумок проведеного дослідження та результати реалізації проекту.

Додаток А містить довідку про перевірку на унікальність пояснювальної записки, а додаток Б – лістинг коду для інтерактивної мапи тривог на базі ESP32.

В цілому бакалаврська робота без додатків містить 63 сторінку, 44 рисунків, 5 таблиці, 22 джерел посилання.

Ключові слова: *інтерактивна мапа, ESP32, WS2812B, Arduino IDE, API, C++.*

ABSTRACT

of the Bachelor's Thesis

«Interactive alarm map of Ukraine based on ESP32 microcontroller»

Student: Horhun Ruslan

Supervisor: Senior Lecturer of the Department of Computer Engineering, Borys Saltovskyy

The aim of this thesis is to create an affordable alarm informer device. The hardware consists of an ESP32 microcontroller, WS2812B addressable LEDs, an I2S sound board MAX98357, and a speaker. The software code, implemented in C++, processes data received from an API and controls the LEDs to display the alarm status in each region with different colors.

The explanatory note of the bachelor's thesis consists of an introduction, four chapters, conclusions, and appendices. The first chapter provides an overview of modern public alert systems, analyzes the requirements for an interactive alarm map, and reviews the hardware and software components needed to create it. The second chapter is dedicated to the design and modeling of the device. It includes the rationale for the chosen technologies, the development of the component connection scheme, ideas for device improvement, and the principle of software operation. The third chapter describes the hardware and software implementation of the system, providing a detailed description of the software development. The fourth chapter focuses on occupational safety. The conclusions summarize the research conducted and the results of the project implementation.

Appendix A contains a report on the uniqueness check of the explanatory note, while Appendix B includes the code listing for the interactive alarm map based on ESP32.

In total, the bachelor's thesis, excluding appendices, consists of 63 pages, 44 figures, 5 tables, and 22 references.

Keywords: *interactive map, ESP32, WS2812B, Arduino IDE, API, C++.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП	4
1 СИСТЕМИ ОПОВІЩЕННЯ НАСЕЛЕННЯ.....	6
1.1 Огляд існуючих рішень	7
1.2 Аналіз вимог до інтерактивної мапи тривоги	12
1.3 Апаратні компоненти.....	14
1.4 Програмні компоненти	18
Висновок до розділу 1	22
2 ПРОЕКТУВАННЯ ТА МОДЕЛЮВАННЯ ПРИСТРОЮ	23
2.1 Обґрунтування обраних технологій	23
2.2 Схема з'єднання компонентів	37
2.3 Ідеї щодо вдосконалення пристрою	37
2.4 Принцип роботи програмного забезпечення.....	39
Висновок до розділу 2	44
3 АПАРАТНО-ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ	46
3.1 Апаратна частина	46
3.2 Кошторисна вартість.....	53
3.3 Отримання даних з API.....	54
3.4 Встановлення необхідних бібліотек.....	56
3.5 Програмна частина.....	58
Висновок до розділу 3	62
ВИСНОВКИ.....	63
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	64
ДОДАТОК А Довідка про перевірку на унікальність пояснювальної записки	67
ДОДАТОК Б Лістинг коду для інтерактивної мапи тривоги на базі esp32	68

ПЕРЕЛІК СКОРОЧЕНЬ

ГІС	– Геоінформаційна система
РЛС	– Радіолокаційна станція
API	– Application Programming Interface
3D	– 3-dimensional
GPIO	– General Purpose Input/Output
GSM	– Global System for Mobile Communications
IDE	– Integrated Development Environment
I2S	– Inter-IC Sound
JSON	– JavaScript Object Notation
RGB	– Red, Green, Blue
SMS	– Short Message Service
UART	– Universal Asynchronous Receiver/Transmitter
UML	– Unified Modeling Language
URL	– Uniform Resource Locator
Wi-Fi	– Wireless Fidelity

ВСТУП

В умовах сучасної ситуації в Україні важливо забезпечити оперативне інформування населення про небезпеки, що виникають внаслідок збройного конфлікту. Ці виклики вимагають швидкої реакції та відповідних заходів, що дозволяють мінімізувати ризики для цивільного населення. Ефективне поширення інформації про тривоги та небезпеки є одним з ключових аспектів захисту цивільного населення. Одним з перспективних рішень у цій сфері є створення інтерактивної мапи тривоги на базі сучасних технологій.

Використання інтерактивної мапи дозволить оперативно інформувати громадян про поточні загрози та забезпечить їх необхідною інформацією для прийняття своєчасних рішень. Інтеграція з різними датчиками та системами сповіщення дозволить отримувати дані в режимі реального часу, що є критично важливим для ефективного реагування на небезпеки.

Тема даної дипломної роботи: "Інтерактивна мапа тривоги України на базі мікроконтролера ESP32".

Мета дослідження: Розробити інтерактивну мапу тривоги України на базі мікроконтролера ESP32, яка дозволить оперативно інформувати населення про актуальні небезпеки та тривоги і буде доступною практично кожному.

Завдання дослідження:

1. Провести аналіз існуючих систем інформування про тривоги та небезпеки.
2. Вивчити можливості мікроконтролера ESP32 для створення інтерактивних мап.
3. Розробити програмне забезпечення для інтерактивної мапи тривоги.
4. Інтегрувати мапу з системами сповіщення для отримання даних у режимі реального часу.
5. Створити повністю функціональний пристрій на базі розробленої системи.

Об'єкт дослідження: Системи оперативного інформування населення про тривоги та небезпеки в умовах збройного конфлікту.

Предмет дослідження: Технології та методи створення інтерактивної мапи тривоги на базі мікроконтролера ESP32, а також їх впровадження для підвищення ефективності інформування населення.

Актуальність даного дослідження зумовлена потребою у швидкому та надійному інформуванні населення про потенційні загрози в умовах збройного конфлікту. Використання сучасних технологій дозволяє створити ефективні системи, які можуть врятувати життя та забезпечити безпеку громадян. Інтерактивна мапа тривоги, розроблена на базі мікроконтролера ESP32, надасть можливість своєчасно отримувати інформацію про небезпеки та оперативно реагувати на них.

Розробка такої системи дозволить забезпечити швидкий доступ до інформації про тривоги, що є вкрай важливим для оперативного реагування та забезпечення безпеки населення. Інтерактивна мапа буде інтегрована з різними датчиками та системами сповіщення, що дозволить отримувати дані в режимі реального часу і оперативно передавати їх користувачам. Це сприятиме підвищенню рівня захищеності громадян та дозволить мінімізувати ризики, пов'язані з виникненням надзвичайних ситуацій.

1 СИСТЕМИ ОПОВІЩЕННЯ НАСЕЛЕННЯ

Система оповіщення – це комплекс організаційно-технічних заходів, пристроїв, технічних методів оповіщення, апаратури, засобів і каналів зв'язку, призначених для своєчасного донесення сигналів і повідомлень з питань цивільної оборони до центральних і місцевих органів виконавчої влади, підприємств, установ, організацій і населення.

Як нескладно здогадатися, сигнал повітряної тривоги надходить у тому випадку, якщо військові фіксують загрозу з повітря. Виявленням таких загроз займаються підрозділи радіолокації. Для виявлення загроз можуть бути задіяні як станції РЛС (рис. 1.1), так й інші засоби розвідки включаючи супутникові. Причому погрози фіксуються не лише на території держави, за якою противник вирішив завдати удару, а й за її межами. Наприклад, якщо кошти розвідки зафіксували виліт літаків супротивника чи пуск ракет, населення буде відразу ж попереджено.



Рисунок 1.1 – Екран стеження за повітряним простором: червоні мітки - повітряні об'єкти, кільця позначають відстань

Після виявлення загрози військові оцінюють ситуацію, наприклад маршрут ракети або авіації. Під час цього враховується можливе відхилення курсу, можливість зникнення ракети з радарів тощо. Таким чином обчислюються регіони, які можуть опинитися під ударом. Після цього місцеву владу по спецз'язку попереджають про небезпеку, внаслідок чого вони включають сирени. Від моменту виявлення загрози до включення сигналу тривоги може пройти кілька хвилин.

Таким чином, сигнал тривоги говорить про те, що в бік регіону насувається ворожа авіація або вже випущена ракета. Тривога включається у всіх містах, що лежать на шляху ракети або авіації, а також ліворуч та праворуч від них, на випадок, якщо ракета (авіація) змінить свій курс. Після того, як ракета пролітає регіон, сирена в ньому вимикається, а в регіонах, до яких вона ще не долетіла, сигнал тривоги продовжує працювати.

У всіх регіонах країни тривога вимикається лише після того, як ракета потрапить у ціль або буде збита системами ППО, але за умови, що військові не виявили нових загроз. Наприклад, якщо у повітрі курсує авіація противника, сигнал тривоги може тривати до того часу, поки літаки не повернуться на місця дислокації.

1.1 Огляд існуючих рішень

Традиційні системи оповіщення про небезпеку включають сирени, радіо та телебачення. Сирени (рис. 1.2), що розташовані в містах та селах, є найпоширенішим методом попередження населення про небезпеку. Їх звукові сигнали можуть попереджати про різні види загроз, такі як повітряні атаки, природні катастрофи або інші надзвичайні ситуації. Основна перевага сирен полягає у їхній здатності миттєво охоплювати велику аудиторію, оскільки вони не залежать від індивідуальних пристроїв. Проте, сирени мають обмежене покриття, особливо у віддалених районах, і не можуть передати детальну інформацію про загрозу.



Рисунок 1.2 – Система оповіщення цивільного населення в Києві

Радіо та телебачення використовуються для передачі більш детальної інформації про небезпеку. Радіо та телевізійні станції можуть переривати свої програми для екстрених повідомлень. Переваги цього методу включають можливість передати детальну інформацію та широке охоплення завдяки високій поширеності радіо і телевізорів. Однак ці системи залежать від електроенергії та інфраструктури, і не всі громадяни можуть бути біля пристроїв у момент передавання сигналу.

Текстові повідомлення (SMS) також використовуються для масового відправлення повідомлень на мобільні телефони. Цей метод має високу ймовірність досягнення конкретного адресата та можливість передати конкретні інструкції. Проте затримки в доставці через навантаження мережі та залежність від мобільного покриття і функціонування операторів є значними недоліками.

Недоліки традиційних систем оповіщення включають обмежену можливість передачі детальної інформації, залежність від наявності та

функціонування інфраструктури, а також вразливість до технічних збоїв та цілеспрямованих атак на інфраструктуру.

Сучасні технології дозволяють створювати більш ефективні та надійні системи оповіщення [1, 2]. Одним з таких рішень є мобільні застосунки для оповіщення (рис. 1.3). Ці застосунки інформують користувачів про небезпеку через push-повідомлення. Перевагами цього методу є оперативність та миттєвість передачі інформації, можливість передати детальну інформацію та інструкції, а також інтерактивні можливості, включаючи інтеграцію з мапами та іншими сервісами. Однак мобільні додатки залежать від інтернет-з'єднання і вимагають, щоб користувачі встановили застосунок на свої пристрої.

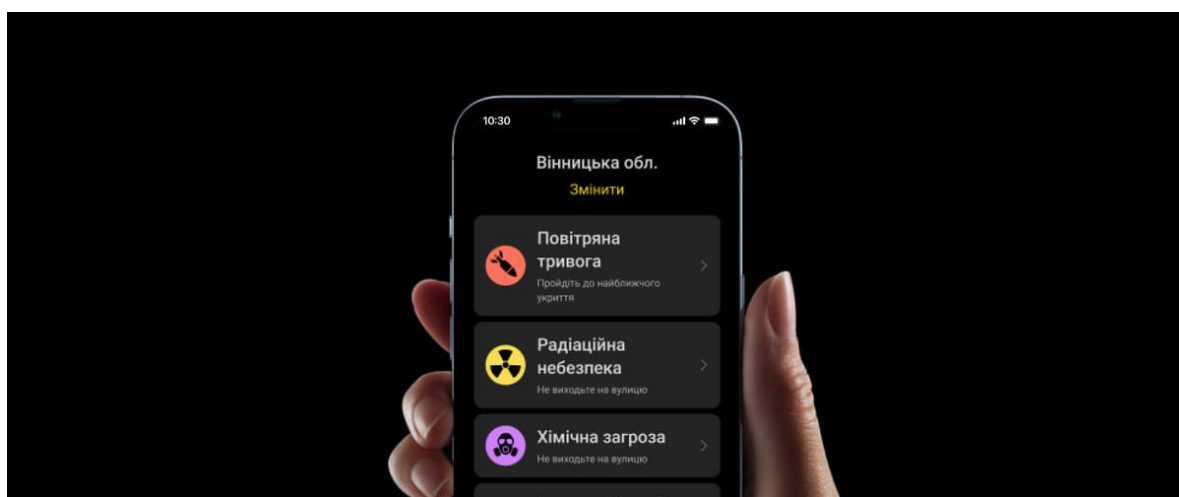


Рисунок 1.3 – Інтерфейс застосунку «Повітряна тривога»

Інтернет-ресурси та соціальні мережі також використовуються для швидкого поширення інформації. Інтернет-сайти урядових та неурядових організацій публікують інформацію про загрози, надаючи доступ до широкої аудиторії та можливість розміщення детальних звітів та інструкцій. Проте цей метод вимагає активного пошуку інформації користувачами та залежить від наявності інтернет-з'єднання.

Інтерактивні мапи [3], що використовують геоінформаційні системи (ГІС), відображають зони небезпеки в реальному часі. Перевагами використання ГІС є візуалізація даних, що спрощує розуміння ситуації, можливість відображення різних типів загроз та їх динаміки, а також інтеграція з іншими інформаційними джерелами та системами. Проте

Інтерактивні мапи залежать від точності та актуальності даних, а також потребують надійної інтернет-інфраструктури для постійного оновлення інформації.

Аналогічні інтерактивні мапи можна знайти на маркетплейсах, таких як olx або prom. Основним їх недоліком є ціна, вони стоять дуже дорого і не кожний може дозволити собі купити ці прилади.

Одна з таких мап (рис. 1.4) підключається до мережі Wi-Fi та оновлює дані. Має декілька режимів яскравості залежно від часу, після 21:00 стає менш яскравим, та більш яскравим о 07:00 ранку. Живиться через USB роз'єм, що дозволяє підключити її до павербанку, ноутбуку, тощо.

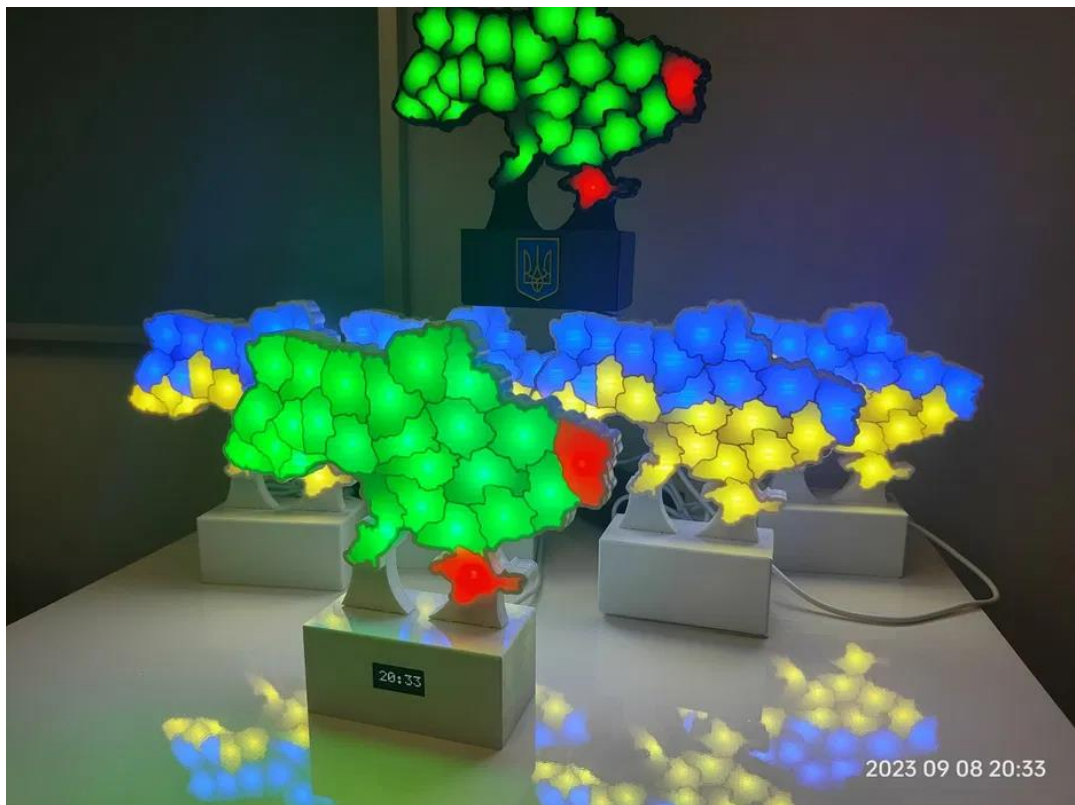


Рисунок 1.4 – Інтерактивна мапа тривоги України з вбудованим годинником

Інша мапа (рис. 1.5) має дерев'яний корпус, живиться через кабель/акумулятор та керується через телеграм.



Рисунок 1.5 – Інтерактивна мапа тривоги України

Порівняння та аналіз ефективності сучасних технологій показують, що мобільні застосунки та інтерактивні мапи мають значні переваги у швидкості та детальності передачі інформації, але стикаються з технічними та інфраструктурними викликами. Оперативність мобільних застосунків та соціальних мереж забезпечує швидку передачу інформації, проте може бути обмежена технічними умовами, такими як інтернет-з'єднання та доступність пристроїв. Детальність інформації, яку надають системі оповіщення із використанням сучасних технологій, дозволяє користувачам бачити візуалізацію загроз. Однак доступність цих технологій залежить від інтернет-з'єднання.

1.2 Аналіз вимог до інтерактивної мапи тривоги

Для розробки інтерактивної мапи тривоги на базі мікроконтролера ESP32 необхідно врахувати наступні вимоги, які поділяються на функціональні та нефункціональні.

1.2.1 Функціональні вимоги

Відображення стану тривоги для кожного регіону

Інтерактивна мапа повинна мати можливість відображати стан тривоги для кожного регіону України. Кожен регіон буде представлений світлодіодом, який змінює свій колір залежно від поточного стану тривоги. Наприклад, зелений колір може означати відсутність небезпеки, жовтий - підвищену готовність, червоний – безпосередню загрозу. Це дозволить користувачам швидко та інтуїтивно отримувати інформацію про стан безпеки в різних частинах країни.

Оновлення даних у реальному часі

Інформація про тривоги повинна оновлюватися автоматично при надходженні нових даних. Це вимагає створення системи, яка здатна постійно моніторити дані з віддалених серверів та оперативно оновлювати відображення на мапі. Такий підхід забезпечить актуальність інформації та своєчасне інформування користувачів про зміни у ситуації.

Підключення до інтернету через Wi-Fi

Для отримання даних про тривоги з віддалених серверів система повинна мати можливість підключення до інтернету через Wi-Fi. Мікроконтролер ESP32, оснащений вбудованим модулем Wi-Fi, забезпечить стабільне та швидке з'єднання з мережею, що дозволить системі оперативно отримувати нові дані та оновлювати відображення тривоги.

1.2.2 Нефункціональні вимоги

Надійність

Система повинна бути стійкою до помилок та забезпечувати коректне відображення даних навіть при втраті з'єднання з інтернетом. Це означає, що система повинна мати механізми для автоматичного відновлення з'єднання та збереження останніх отриманих даних у разі збоїв. Надійність також включає захист від неправдивих сповіщень та гарантію точності переданої інформації.

Енергоефективність

Використання компонентів з низьким енергоспоживанням є критично важливим для забезпечення тривалої автономної роботи системи. Мікроконтролер ESP32 відомий своїм низьким енергоспоживанням, що дозволить зменшити загальні витрати енергії. Додатково, система може бути оснащена енергоефективними світлодіодами та іншими компонентами, щоб забезпечити мінімальне споживання електроенергії.

Простота використання

Система повинна бути легкою у налаштуванні та інтеграції. Це включає зручний інтерфейс для початкового налаштування з'єднання з Wi-Fi, можливість автоматичного оновлення програмного забезпечення та доступні інструкції для користувачів. Простота використання також передбачає інтуїтивно зрозуміле відображення інформації, яке не вимагає додаткових пояснень або навчання для розуміння стану тривоги.

Аналіз вимог до інтерактивної мапи тривоги на базі мікроконтролера ESP32 дозволяє визначити ключові аспекти, які необхідно врахувати під час розробки системи. Функціональні вимоги зосереджуються на забезпеченні актуальності та точності інформації про тривоги, а також на інтерактивному відображенні стану тривоги для кожного регіону. Нефункціональні вимоги акцентують увагу на надійності, енергоефективності та простоті використання системи. Виконання цих вимог забезпечить створення ефективної та надійної

системи оповіщення, яка буде корисною для населення, рятувальних служб та органів влади.

1.3 Апаратні компоненти

За основу апаратної частини інтерактивної мапи тривоги України було обрано мікроконтролер **ESP32** (рис. 1.6).

ESP32 [4] має значно вищу продуктивність у порівнянні з **ESP8266**. ESP32 оснащений двоядерним процесором, що дозволяє одночасно виконувати більше завдань, що особливо важливо для обробки великої кількості даних. Крім того, ESP32 має більшу кількість GPIO пінів, що спрощує підключення великої кількості світлодіодів та інших компонентів, необхідних для функціонування інтерактивної мапи.

Також розглядалися варіанти з використанням мікроконтролерів **Arduino Uno** або **Arduino Nano**. Однак ці мікроконтролери не мають вбудованого Wi-Fi модуля, що є критичним для інтерактивної мапи, яка повинна мати можливість отримувати та обробляти дані в режимі реального часу. Для реалізації Wi-Fi функціональності на базі Arduino Uno або Arduino Nano потрібно було б додатково використовувати Wi-Fi модуль, наприклад, ESP8266, що ускладнило б конструкцію та збільшило загальну вартість проекту.

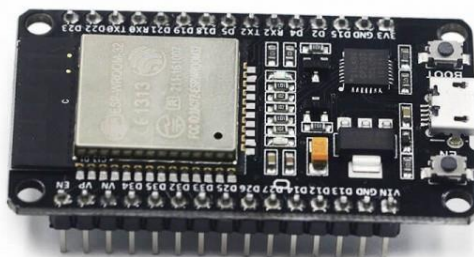


Рисунок 1.6 – Зовнішній вигляд ESP32

За візуальний сигнал будуть відповідати адресні RGB світлодіоди **WS2812B** (рис. 1.7) [5]. Їх обрання обґрунтоване тим, що вони вимагають лише одного цифрового піну для передачі даних, що значно спрощує процес підключення та зменшує кількість необхідних проводів. Крім того WS2812B відомі своєю надійністю та довговічністю.



Рисунок 1.7 – Адресні світлодіоди WS2812B

Візуальний сигнал дублюється звуком через звукову плату **MAX98357** (рис. 1.8), яка керується через протокол I2S [6].



Рисунок 1.8 – Звукова плата MAX98357

Звук виводиться через динамік потужністю 0,5 Вт з опором 8 Ом (рис. 1.9).



Рисунок 1.9 – Динамік

Для обмеження струму до світлодіодів у системі буде використано резистор стандартний вивідний **MFR-25FRF52-330R Yageo** (рис. 1.10).



Рисунок 1.10 – Резистор MFR-25FRF52-330R Yageo

Для стабілізації живлення ESP32 підібрано конденсатор **EEU-FC1C101** (рис. 1.11).

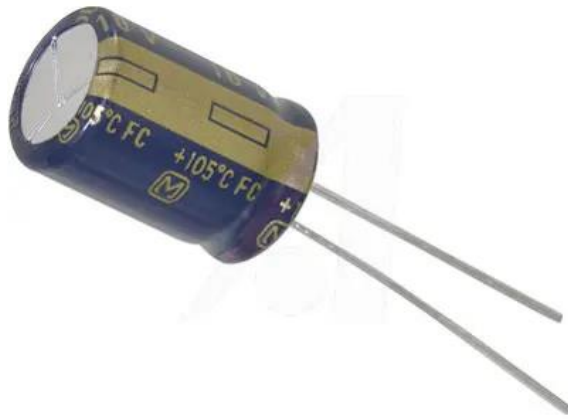


Рисунок 1.11 – Конденсатор EEU-FC1C101

У якості джерела живлення для такої системи найкращим варіантом буде **блок живлення 5V/3A** (рис. 1.12). Він має більшу потужність у порівнянні з **блоком живлення 5V/2A**, що дозволить підключити значно більшу кількість світлодіодів та звуковий компонент.



Рисунок 1.12 – Блок живлення 5V/3A

Корпус розробки – роздрукована на 3D принтері мапа України з адміністративним поділом на області (рис. 1.13). В кожній комірці знаходиться адресний світлодіод WS2812B.

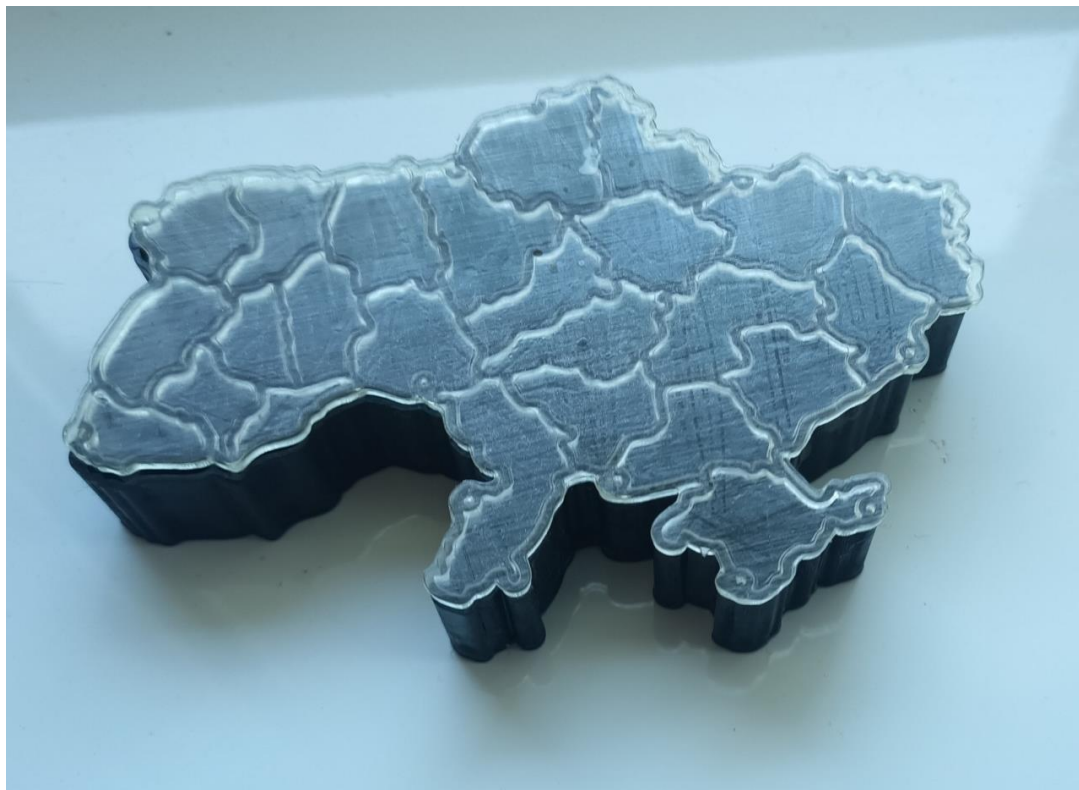


Рисунок 1.13 – Корпус пристрою

Обрана апаратна платформа забезпечує високу продуктивність, надійність та гнучкість інтерактивної мапи тривоги України, дозволяючи ефективно обробляти та відображати інформацію в режимі реального часу. Крім того, усі компоненти є доступними за ціною.

1.4 Програмні компоненти

Arduino IDE — це програмне забезпечення для написання, редагування та завантаження програмного коду на плату Arduino. Воно використовує мову програмування C/C++ з деякими спрощеннями, адаптованими для роботи з апаратним забезпеченням Arduino. Основні функції Arduino IDE включають текстовий редактор з підсвічуванням синтаксису, компілятор для перетворення коду у машинний код, завантажувач для відправки скомпільованого коду на плату Arduino через USB, а також серіал монітор для

перегляду та налагодження даних, що передаються з плати на комп'ютер. Arduino IDE відома своєю простотою використання, доступністю для новачків, підтримкою численних бібліотек для різних датчиків і модулів, кросплатформеністю (Windows, macOS, Linux), а також великою спільнотою користувачів, які діляться прикладами коду та проектами [7].

PlatformIO є потужною платформою для програмування мікроконтролерів, яка підтримує багато платформ та мікроконтролерів, включаючи Arduino, ESP32, STM32 та інші. Вона забезпечує інтегроване управління проектами, налагодження, тестування та безперервну інтеграцію, а також можливість використання зовнішніх бібліотек та залежностей через PlatformIO Library Manager. Проте, PlatformIO має більш складний інтерфейс та налаштування у порівнянні з Arduino IDE.

MicroPython використовує інтерпретовану мову програмування, що дозволяє писати та налагоджувати код безпосередньо на мікроконтролері, з простим та зрозумілим синтаксисом Python. Це забезпечує швидкий цикл розробки та можливість швидко змінювати та тестувати код. Однак, MicroPython менш ефективний в порівнянні з нативним кодом C/C++, що може бути критичним для проектів з високими вимогами до продуктивності, і має обмежену кількість бібліотек порівняно з Arduino IDE.

1.4.1 Встановлення та налаштування Arduino IDE

1) Перейдіть на офіційний сайт Arduino: <https://www.arduino.cc/en/software> та завантажте інсталятор (рис. 1.14).

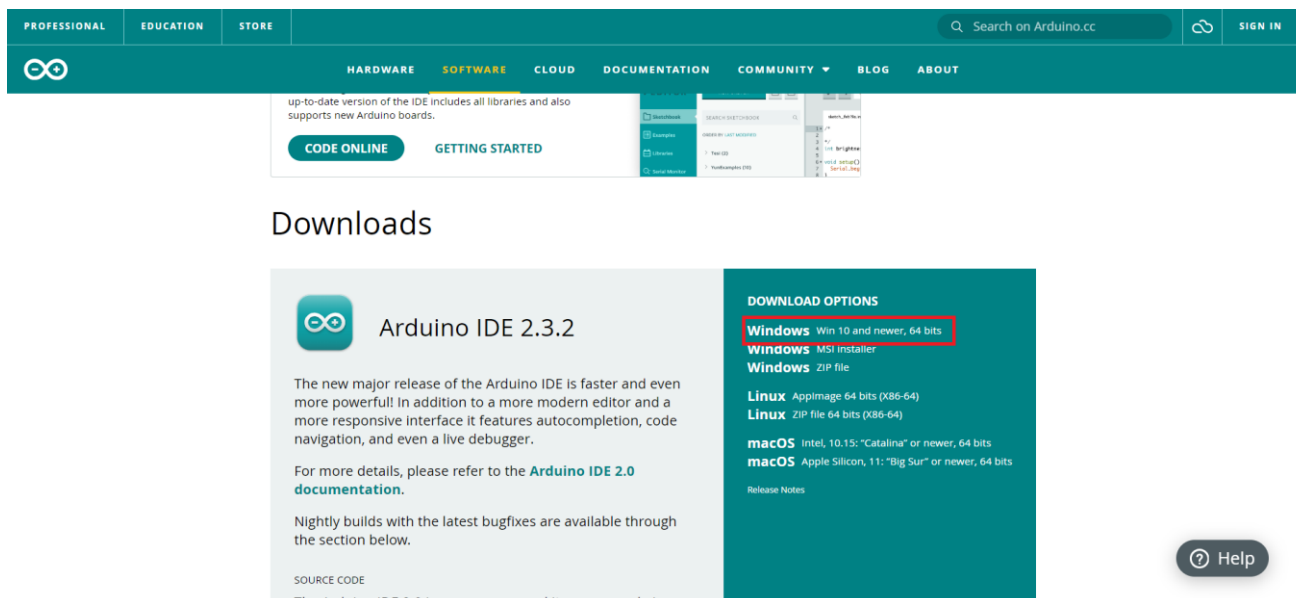


Рисунок 1.14 – Офіційний сайт Arduino

2) Відкрийте завантажений файл і слідуйте інструкціям на екрані для завершення встановлення. Відкрийте Arduino IDE (рис. 1.15).

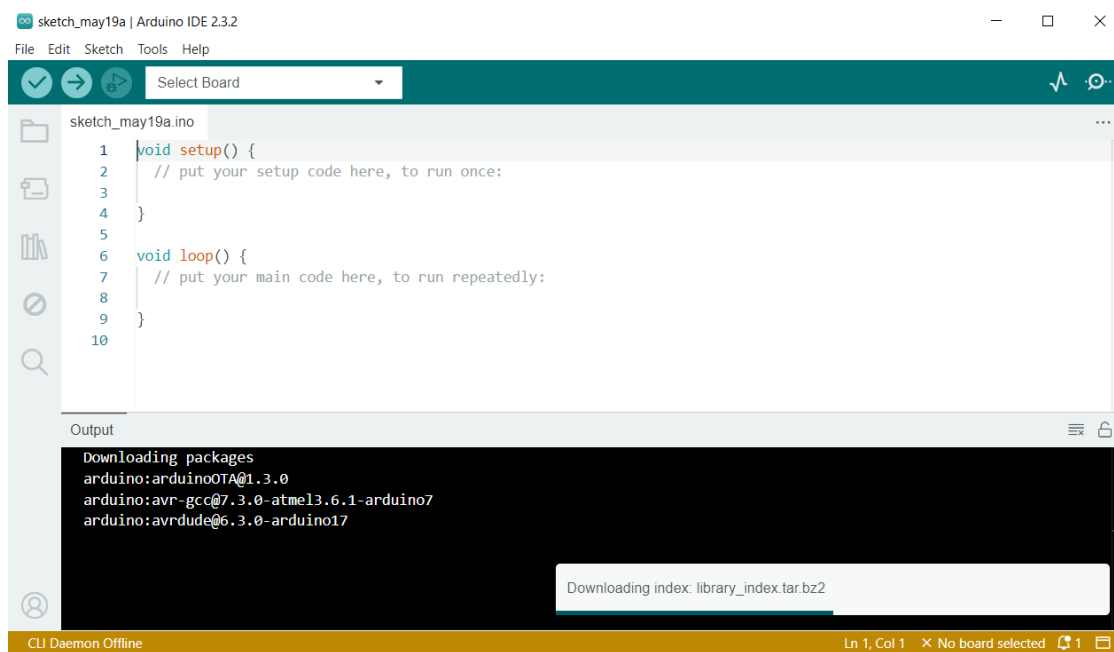


Рисунок 1.15 – Вікно програми Arduino IDE

3) Перейдіть до File -> Preferences. У полі "Additional Boards Manager URLs" додайте наступний URL: https://dl.espressif.com/dl/package_esp32_index.json (рис. 1.16) [8].

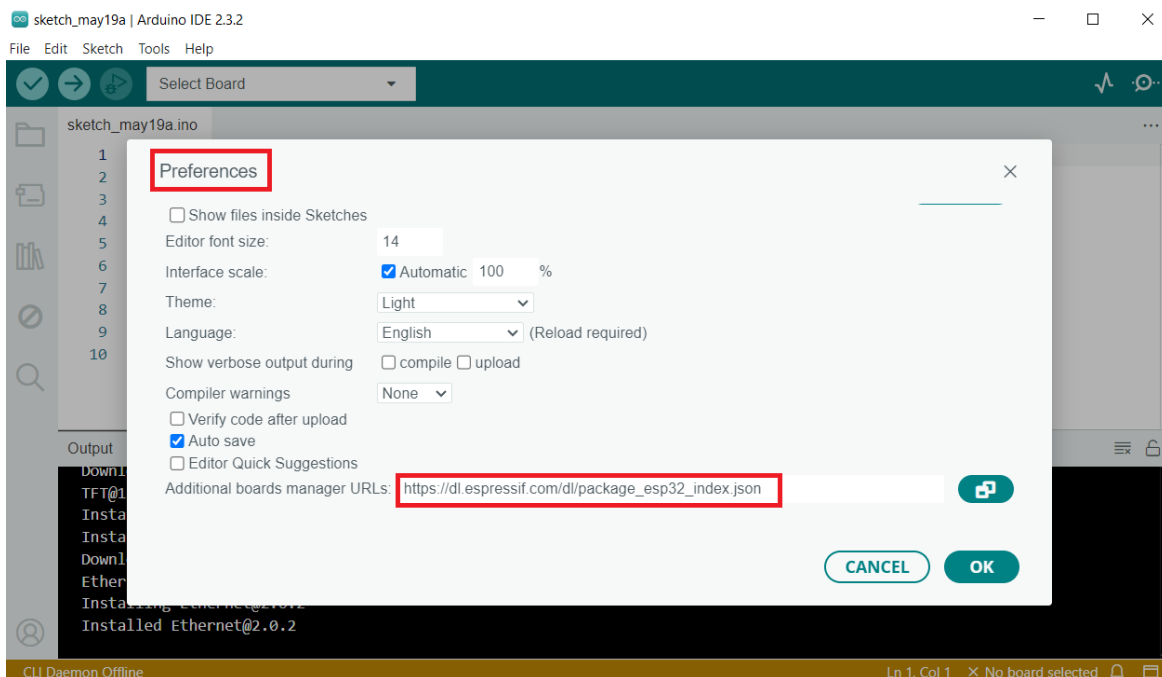


Рисунок 1.16 – Додавання URL платформи

4) Перейдіть до Tools -> Board -> Boards Manager. У вікні, що з'явилося, знайдіть "esp32" і натисніть кнопку Install (рис. 1.17).

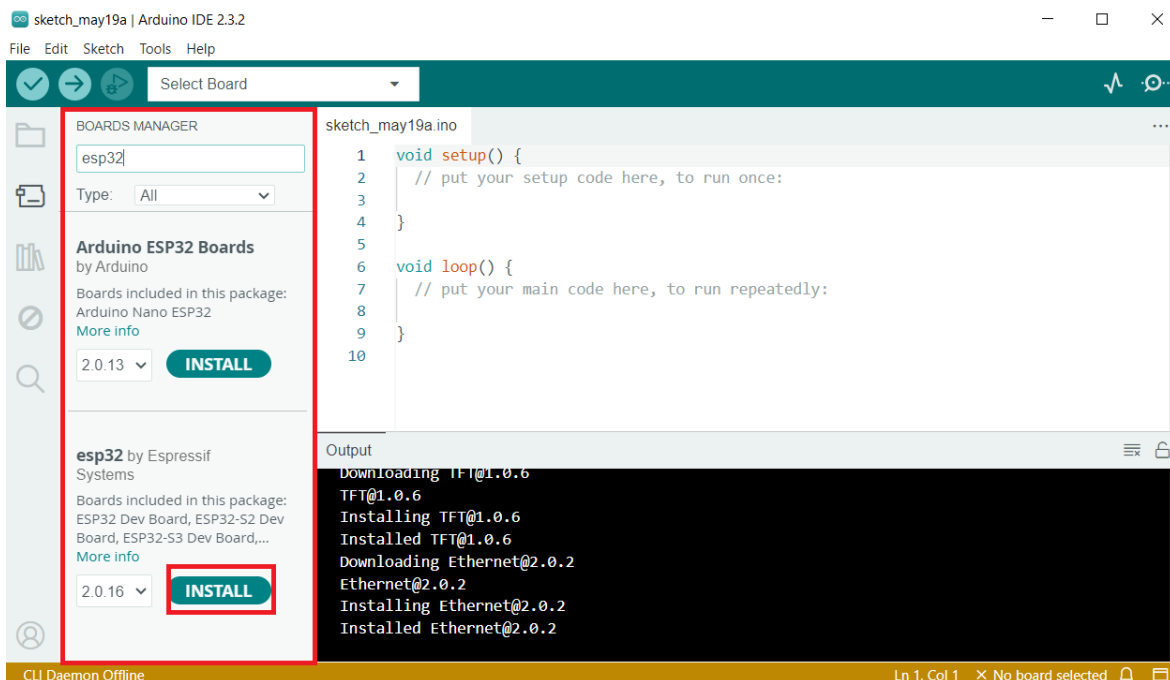


Рисунок 1.17 – Встановлення плати ESP32

Тепер Arduino IDE готове до роботи з мікроконтролером ESP32, залишилося лише підключити плату до PC та перевірити правильність роботи.

Висновок до розділу 1

Розробка інтерактивної мапи тривоги на базі мікроконтролера ESP32 дозволяє створити ефективну систему оповіщення про загрози для населення України. Функціональні вимоги до системи передбачають відображення стану тривоги для кожного регіону в режимі реального часу, підключення до інтернету через Wi-Fi та зручність в налаштуванні і використанні. Нефункціональні вимоги акцентують увагу на надійності, енергоефективності та простоті інтеграції системи.

Для реалізації апаратної частини було обрано мікроконтролер ESP32 завдяки його високій продуктивності, наявності вбудованого Wi-Fi модуля та достатній кількості GPIO пінів для підключення світлодіодів та інших компонентів. Візуальне відображення забезпечуватимуть адресні світлодіоди WS2812B, а за аудіо сповіщення відповідатиме звукова плата MAX98357. Додаткові компоненти, такі як резистори, конденсатори та потужний блок живлення, забезпечать стабільність і надійність роботи системи.

Програмна частина проекту включає використання Arduino IDE для розробки та налагодження програмного забезпечення. Arduino IDE забезпечує простоту використання, зручний текстовий редактор з підсвічуванням синтаксису, компілятор і завантажувач, що дозволяє швидко розробляти та тестувати код на ESP32.

Загалом, обрана апаратна та програмна платформа забезпечує високу продуктивність, надійність та енергоефективність інтерактивної мапи тривоги. Це дозволяє ефективно обробляти та відображати актуальну інформацію про стан безпеки в різних регіонах України, роблячи систему корисною для населення, рятувальних служб та органів влади.

2 ПРОЕКТУВАННЯ ТА МОДЕЛЮВАННЯ ПРИСТРОЮ

2.1 Обґрунтування обраних технологій

У цьому розділі буде розглянуто та обґрунтовано вибір технологій, які були застосовані у процесі розробки проекту. Буде проведено аналіз кожної з обраних технологій з точки зору її функціональних можливостей, продуктивності, надійності та відповідності вимогам проекту. Це дозволить не лише зрозуміти, чому саме ці технології були обрані, але й продемонструвати їх доцільність та ефективність у контексті вирішення конкретних завдань проекту.

2.1.1 Технологія бездротового зв'язку

Обрання технології бездротового зв'язку – ключовий етап у проектуванні. Такі технології забезпечують передачу даних між пристроями без використання фізичних кабелів. Вони є основою для побудови ефективних та гнучких мереж, що дозволяють інтегрувати різноманітні пристрої та системи в єдину інформаційну інфраструктуру. У цьому розділі буде розглянуто основні технології бездротового зв'язку: Wi-Fi, GSM, Bluetooth, ZigBee та LoRaWAN (табл. 2.1). На основі аналізу буде обрано технологію, що буде використовуватись у подальшому проектуванні та реалізації.

Таблиця 2.1 – Порівняння бездротових технологій

№	Feature	Wi-Fi (IEEE802 11B)	Bluetooth (IEEE 802.15.1)	ZigBee (IEEE 802.15.4)	GSM/GPRS CDMA/1X RTT
1	Radio	DSSS#	FHSS	DSSS	–
2	Data Rate	11 M bps	1 M bps	250 kbps	64-128+
3	Nodes Per Master	32	7	64000	1
4	Slave Enumeration Latency	Up to 3s	Up to 10s	30 ms	–

№	Feature	Wi-Fi (IEEE802 11B)	Bluetooth (IEEE 802.15.1)	ZigBee (IEEE 802.15.4)	GSM/GPRS CDMA/1X RTT
5	Data type	Video, audio, graphics pictures, file	Audio, graphics, pics, files	Small data packet	Wide Area Voice and Data
6	Range (M)	100	10	70	10+
7	Extendibility	Roaming possible	No	Yes	Yes
8	Battery Type	AAA battery power	Li-ion battery pack	Alkaline batteries	Nickel cadmium
9	Battery Life	Hours	1 week	>1 year	1.7
10	Complexity	Complex	Very complex	Simple	Simple
11	Frequency	2.4GHz	2.4GHz	850-930MHz	1,710-1,785MHz
12	Security	Authentication service set ID (SSID)	64bit, 128bit	128bit AES and application layer user defined	–
13	Operating Environment	+32°F to +104°F	+32°F to +122°F	+32°F to +131°F	-30°C to +85°C
14	Success metrics	Speed, flexibility	Compatible, easy to handle	Reliability, power, cost	Reach, quality

Wi-Fi (Wireless Fidelity) є однією з найпоширеніших технологій бездротового зв'язку, яка забезпечує високошвидкісну передачу даних на коротких та середніх відстанях [9]. Ця технологія використовується в локальних мережах (LAN) і забезпечує з'єднання між пристроями, такими як комп'ютери, смартфони, планшети та роутери.

Основні переваги Wi-Fi включають високу швидкість передачі даних (до декількох Гбіт/с у сучасних стандартах, таких як Wi-Fi 6), відносно велика зона покриття (до 100 метрів в приміщенні) та здатність одночасно підтримувати велику кількість підключених пристроїв. Проте, Wi-Fi споживає значну кількість енергії, що може бути критичним для пристроїв з обмеженим ресурсом батареї.

Wi-Fi доцільно використовувати у проектах, де необхідна висока пропускна здатність та покриття великих площ, наприклад, у домашніх та офісних мережах, кафе, бібліотеках, аеропортах тощо. Також Wi-Fi підходить для потокового передавання відео, онлайн-ігор та інших додатків, які потребують великої швидкості та низької затримки.

GSM (Global System for Mobile Communications) – це стандарт для мобільного зв'язку, який забезпечує передачу голосу та даних на великих відстанях [10]. GSM використовується переважно в мобільних телефонах та інших пристроях, які підключаються до мобільних мереж.

Основні переваги GSM включають широке покриття (фактично по всьому світу), можливість роботи в умовах руху (наприклад, під час подорожей автомобілем чи поїздом) та підтримка голосових викликів та передачі даних. Недоліком є порівняно низька швидкість передачі даних (до декількох сотень кбіт/с у стандарті GSM і до декількох Мбіт/с у новіших поколіннях, таких як 3G або 4G).

GSM підходить для проектів, де необхідно забезпечити мобільний зв'язок на великих відстанях, таких як системи моніторингу транспорту, мобільні телефони, системи віддаленого контролю та управління, а також у проектах Інтернету речей (IoT), де важлива широкомасштабна мережа з глобальним покриттям.

Bluetooth – це технологія бездротового зв'язку на короткі відстані, призначена для обміну даними між пристроями. Вона широко використовується в пристроях особистого користування, таких як гарнітури, клавіатури, миші, смарт-годинники та фітнес-трекери.

Основні переваги Bluetooth включають низьке енергоспоживання, простота налаштування та використання, а також підтримка низькошвидкісної передачі даних на відстанях до 10 метрів (для стандарту Bluetooth Low Energy – BLE). Недоліком є обмежена зона покриття та порівняно невисока швидкість передачі даних (до 3 Мбіт/с у стандарті Bluetooth 4.0).

Bluetooth доцільно використовувати у проектах, де важливим є низьке енергоспоживання та невелика зона покриття, наприклад, у бездротових периферійних пристроях, носимих гаджетах, системах "розумного" дому та персональних медичних пристроях.

ZigBee – це технологія бездротової мережі, призначена для додатків з низьким енергоспоживанням і малою швидкістю передачі даних. Вона використовується у мережах датчиків та інших пристроїв, що потребують тривалого автономного живлення.

Основні переваги ZigBee включають дуже низьке енергоспоживання, можливість побудови мереж з великою кількістю пристроїв (до 65000 вузлів у мережі) та висока стійкість до збоїв завдяки мережевій топології "mesh". Недоліком є низька швидкість передачі даних (до 250 кбіт/с) та обмежена зона покриття (до 100 метрів у приміщенні).

LoRaWAN (Long Range Wide Area Network) – це технологія бездротової мережі з великою дальністю дії, призначена для додатків Інтернету речей (IoT), які потребують передачі даних на великі відстані з низьким енергоспоживанням.

Основні переваги LoRaWAN включають дуже велику дальність передачі (до 15 км у сільських умовах), низьке енергоспоживання та можливість підтримки великої кількості пристроїв у мережі. Недоліком є низька швидкість передачі даних (до 50 кбіт/с) та обмежені можливості щодо передачі великих обсягів даних.

LoRaWAN доцільно використовувати у проектах, де необхідна передача даних на великі відстані з мінімальним енергоспоживанням, таких як системи моніторингу навколишнього середовища, сільське господарство, розумні міста, віддалені системи моніторингу та управління, а також інші додатки Інтернету речей (IoT).

Кожна з розглянутих технологій бездротового зв'язку має свої унікальні переваги та недоліки, що робить їх підходящими для різних типів проектів. Wi-Fi підходить для проектів з високими вимогами до швидкості передачі

даних та покриття. GSM ідеально підходить для мобільних додатків з широким покриттям. Bluetooth використовується в додатках з низьким енергоспоживанням та короткими відстанями. ZigBee оптимальний для мереж з великою кількістю пристроїв і низьким енергоспоживанням. LoRaWAN доцільний для проектів з передачею даних на великі відстані з мінімальним енергоспоживанням.

Для цього проекту підходить технологія Wi-Fi (рис. 2.1), оскільки вона забезпечує високу швидкість передачі даних та достатнє покриття. Wi-Fi дозволить створити надійне та швидкісне з'єднання між пристроями, що є критично важливим для успішної реалізації проекту. Це також забезпечить зручність підключення і можливість інтеграції з існуючими мережами.

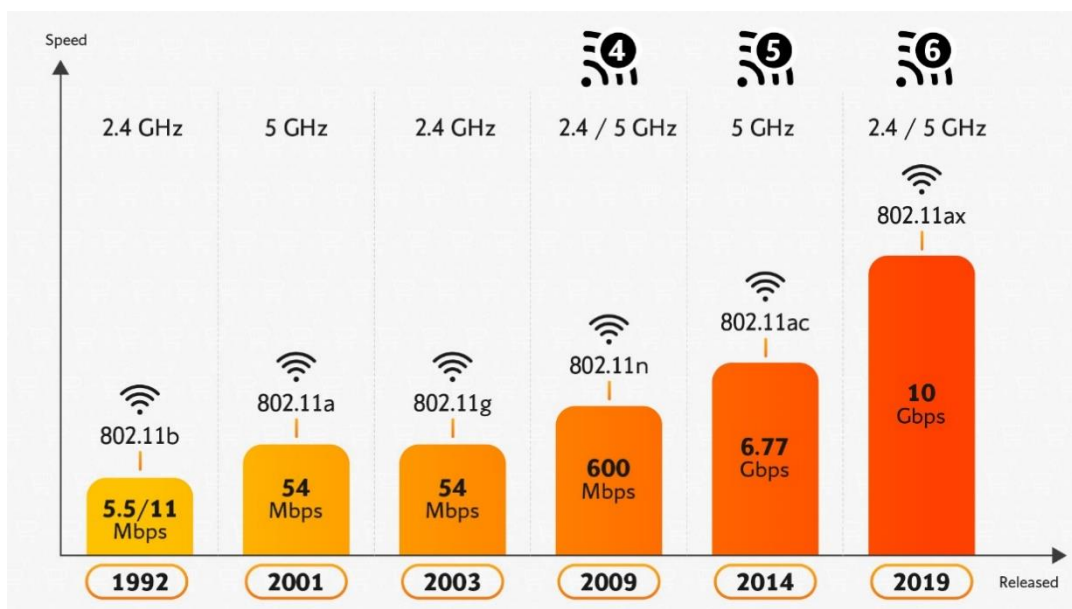


Рисунок 2.1 – Еволюція бездротових стандартів Wi-Fi

Проте, при вдосконаленні приладу можна розглянути можливість використання технології GSM. Це дозволить забезпечити ширше покриття та мобільність рішення, особливо в тих випадках, коли доступ до Wi-Fi мережі може бути обмеженим.

2.1.2 Мікроконтролер

ESP32, ESP8266 і Arduino UNO – це три популярні мікроконтролери, кожен з яких має свої унікальні характеристики (рис. 2.2), які роблять їх придатними для певних застосувань.

SPECS/BOARD	ESP32	ESP8266	ARDUINO UNO
Number of Cores	2	1	1
Architecture	32 Bit	32 Bit	8 Bit
CPU Frequency	160 MHz	80 MHz	16 MHz
WiFi	YES	YES	NO
BLUETOOTH	YES	NO	NO
RAM	512 KB	160 KB	2 KB
FLASH	16 MB	16 MB	32 KB
GPIO PINS	36	17	14
Busses	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S	SPI, I2C, UART
ADC Pins	18	1	6
DAC Pins	2	0	0

Рисунок 2.2 – Порівняльна таблиця ESP32, ESP8266 та Arduino UNO

ESP32 (рис. 2.3) має два ядра, що робить його значно потужнішим у порівнянні з однопроцесорними ESP8266 та Arduino UNO. Багатоядерність дозволяє ESP32 ефективніше виконувати паралельні завдання. ESP8266 та Arduino UNO оснащені одним ядром. Обидва використовують різну архітектуру: ESP8266 – 32-бітну, а Arduino UNO – 8-бітну. 32-бітна архітектура ESP8266 забезпечує вищу продуктивність і можливість обробки складніших завдань. ESP32 працює на частоті 160 МГц, що є найвищим показником серед трьох мікроконтролерів. Це забезпечує йому високий рівень продуктивності та швидку обробку даних. ESP8266 має частоту 80 МГц, що також є досить високим показником, але все ж поступається ESP32. Arduino UNO з частотою 16 МГц значно поступається обом ESP32 і ESP8266, що обмежує його застосування в проектах, які вимагають високої швидкості обробки.

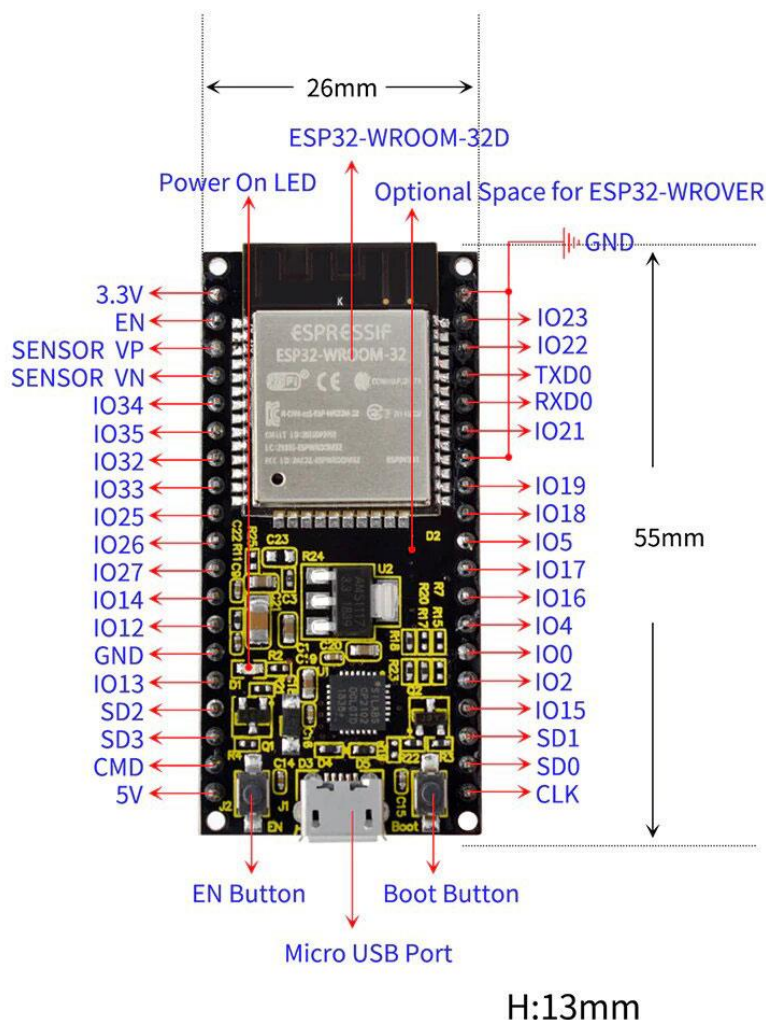


Рисунок 2.3 – Розташування та призначення пінів на ESP32

ESP32 підтримує як Wi-Fi, так і Bluetooth, що робить його надзвичайно універсальним для розробки бездротових додатків. ESP8266 підтримує тільки Wi-Fi, але це також робить його досить потужним інструментом для бездротових проектів. Arduino UNO не має вбудованої підтримки Wi-Fi або Bluetooth, що обмежує його можливості для бездротових додатків без додаткових модулів.

Відносно оперативної пам'яті, ESP32 має 512 КБ, що значно більше, ніж у ESP8266 (160 КБ) та Arduino UNO (2 КБ). Це дозволяє ESP32 обробляти більші обсяги даних і підтримувати складніші програми. FLASH-пам'ять у ESP32 і ESP8266 однакова – 16 МБ, що значно більше ніж 32 КБ у Arduino UNO. Це дозволяє зберігати більше програм та даних.

Кількість GPIO пінів також є важливим фактором. ESP32 має 36 GPIO пінів, що значно більше, ніж у ESP8266 (17) та Arduino UNO (14). Це забезпечує більшу гнучкість для підключення різних датчиків та пристроїв.

Щодо шин, ESP32 підтримує SPI, I2C, UART, I2S і CAN, що робить його дуже універсальним. ESP8266 підтримує SPI, I2C, UART і I2S, що також робить його досить гнучким, але з меншою кількістю опцій порівняно з ESP32. Arduino UNO підтримує тільки SPI, I2C і UART, що обмежує його можливості підключення.

В ESP32 є 18 ADC пінів, що значно більше ніж у ESP8266 (1) та Arduino UNO (6). Крім того, ESP32 має 2 DAC піни, в той час як ESP8266 і Arduino UNO взагалі не мають DAC пінів. Отже, ESP32 є найбільш потужним і універсальним мікроконтролером серед трьох розглянутих (табл. 2.2).

Таблиця 2.2 – Технічні характеристики ESP32

Параметр	Опис
Архітектура	Xtensa 32-bit LX6 dual-core
Тактова частота	До 240 МГц
Оперативна пам'ять (RAM)	520 КБ SRAM
Вбудована пам'ять (ROM)	448 КБ ROM
Флеш-пам'ять	До 4 МБ (залежно від моделі)
Wi-Fi	802.11 b/g/n
Bluetooth	Bluetooth 4.2 та BLE
GPIO	До 34 програмованих входів/виходів
Інтерфейси	SPI, I2C, I2S, UART, SDIO, PWM, ADC, DAC
Робоча напруга	2.2–3.6 В
Споживана потужність	Споживання в режимі сну: < 5 μ А, при активній роботі: до 500 мА
Температурний діапазон	-40°C до +125°C
Розміри модуля	Приблизно 25.5 мм x 18 мм (може варіюватися залежно від виробника)
Підтримувані середовища розробки	Arduino IDE, ESP-IDF, MicroPython, PlatformIO
Додаткові функції	Вбудований датчик Холла, датчик температури, підтримка апаратного шифрування, акселератор криптографії

Параметр	Опис
Роз'єм для антени	Вбудована антена та роз'єм для зовнішньої антени (u.FL)
Сертифікація	CE, FCC, RoHS

ESP32 обраний завдяки своїй високій продуктивності, багатоядерності, широким можливостям бездротового зв'язку (Wi-Fi та Bluetooth), великій кількості GPIO пінів та підтримці різних інтерфейсів. З його високою тактовою частотою, значною кількістю оперативної пам'яті та розширеними можливостями обробки даних, ESP32 є найпотужнішим і універсальним мікроконтролером серед розглянутих варіантів, що робить його ідеальним для цього проекту.

2.1.3 Світлодіод

WS2812B, SK6812 і APA102 (рис. 2.4) є типами адресованих світлодіодів, відомих також як LED пікселі. Вони використовуються для створення кольорових ефектів у різноманітних проектах, включаючи підсвічування, декоративні ефекти та світлові інсталяції.

	WS2812B	SK6812	APA102
Control	one-wire interface	one-wire interfacer	standard SPI
Per meter options	30/60/144	30/60/144	30/60/144
Current Draw / LED	50mA @ 5V	40mA @ 5V	50mA @ 5V
Color Channel	8-Bit PWM	8-Bit PWM	8-bit PWM
Brightness Control	PWM Based	PWM based	5-bit PWM
PWM Frequency	400Hz	1.2kHz	20kHz
Time taken to change LED (450 LED strip)	0.225 Milliseconds	0.225 Milliseconds	10 Milliseconds
LED Type	RGB	RGBW	RGB

Рисунок 2.4 – Порівняльна таблиця WS2812B, SK6812 та APA102

WS2812B (рис. 2.5) є одним із найпоширеніших типів світлодіодів, які використовуються в різноманітних проектах завдяки своїй простоті та доступності. Він використовує одно-провідний інтерфейс для керування, що значно спрощує підключення великої кількості світлодіодів в ланцюг. Кожен світлодіод споживає 50 мА при напрузі 5В, а для кожного кольорового каналу

використовується 8-бітний ШІМ, що забезпечує достатньо плавне регулювання яскравості. Частота ШІМ у WS2812B складає 400 Гц, що є достатнім для більшості статичних і простих динамічних ефектів, але не підходить для додатків з високими вимогами до швидкості оновлення, таких як POV ефекти.

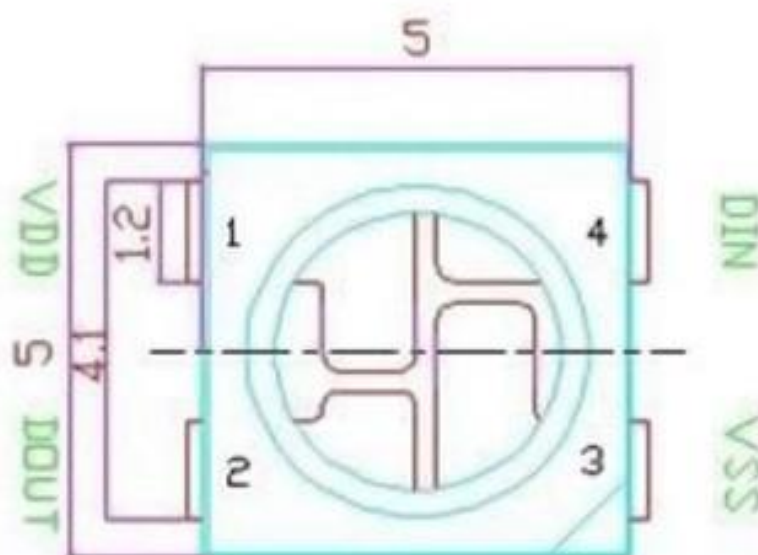


Рисунок 2.5 – Розміри та розташування пінів на WS2812B

SK6812 є майже ідентичним клоном WS2812B, але з кількома вдосконаленнями. Одним з основних покращень є збільшена частота ШІМ до 1.2 кГц, що забезпечує більш плавне регулювання яскравості та кращу якість зображення при швидких змінах кольорів. Також SK6812 підтримує додатковий білий світлодіод у кожному пікселі, що дозволяє створювати RGBW-ефекти з більш точними кольорами та покращеною яскравістю. Як і WS2812B, SK6812 використовує одно-провідний інтерфейс, що робить його сумісним з існуючими ланцюгами WS2812B без потреби у зміні програмного забезпечення.

APA102, на відміну від WS2812B і SK6812, використовує стандартний SPI інтерфейс для керування світлодіодами. Це значно спрощує питання синхронізації та дозволяє керувати світлодіодами незалежно від таймінгів, що є важливим при використанні з SBC (одноплатними комп'ютерами), які

працюють під управлінням операційних систем. АРА102 має значно вищу частоту ШІМ – 20 кГц, що робить його ідеальним для POV-додатків та інших проектів, де висока частота оновлення є критичною. Крім того, час реакції АРА102 значно менший (10 мс у порівнянні з 0.225 мс у WS2812В та SK6812), що забезпечує швидше оновлення даних і покращену візуальну якість при швидких змінах кольорів.

Однією з основних переваг WS2812В є його популярність та велика кількість доступних бібліотек і прикладів для різних мікроконтролерів, таких як Arduino, що робить його доступним навіть для початківців. Однак, зважаючи на відносно низьку частоту ШІМ, цей світлодіод не є найкращим вибором для додатків з високими вимогами до швидкості оновлення.

SK6812 надає додаткові можливості завдяки підтримці RGBW та вищій частоті ШІМ, що робить його більш універсальним у порівнянні з WS2812В. Він є гарним вибором для тих, хто потребує додаткової яскравості та точності кольорів, особливо в проектах, де використовується біле світло.

АРА102, хоч і є дорожчим, надає найкращі можливості з точки зору частоти оновлення та якості зображення. Використання SPI інтерфейсу забезпечує стабільну та швидку передачу даних, що робить його ідеальним для професійних додатків та проектів з високими вимогами до швидкості оновлення та якості зображення.

WS2812В є чудовим варіантом для простих і недорогих проектів, де висока частота оновлення не є критичною (табл. 2.3).

Таблиця 2.3 – Технічні характеристики WS2812В

Параметр	Опис
Робоча напруга	3.5...5.3В
Максимальний струм	60 мА
Інтенсивність випромінювання червоний колір (625 Нм)	550-700 мкд
Інтенсивність випромінювання зелений колір (520 Нм)	1100-1400 мкд
Інтенсивність випромінювання синій колір (470 Нм)	200-400 мкд

Параметр	Опис
Кут випромінювання	120 °
Тип корпусу	5050

WS2812B відрізняється своєю доступністю та простотою підключення завдяки одно-провідному інтерфейсу. Він підтримується численними бібліотеками та прикладами для популярних мікроконтролерів, що робить його зручним для початківців. Попри нижчу частоту ШІМ, WS2812B ідеально підходить для бюджетних проектів, де швидкість оновлення не є основною вимогою.

2.1.4 Звукова плата

MAX98357 (рис. 2.6) – це високоефективний цифровий підсилювач класу D, який приймає вхідний сигнал через інтерфейс I2S. Він був обраний через свої відмінні характеристики: висока ефективність, низька споживана потужність і зручний інтерфейс I2S, що забезпечує високу якість звуку. Плата має компактний дизайн.

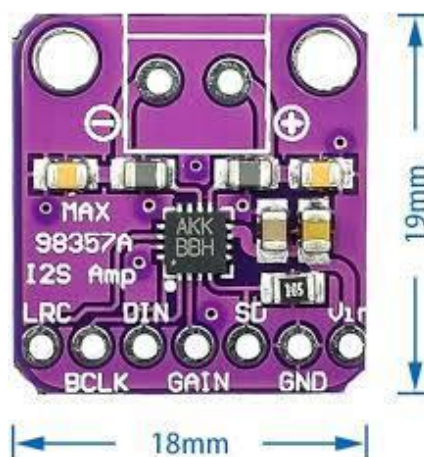


Рисунок 2.6 – Розташування та призначення пінів на MAX98357

Основні характеристики MAX98357 включають цифровий вхід I2S, що підтримує стандартний аудіоінтерфейс і дозволяє легко інтегрувати плату з багатьма сучасними мікроконтролерами і процесорами, які мають I2S виходи. Підсилювач класу D забезпечує високий коефіцієнт корисної дії, що знижує тепловиділення і споживану потужність, що важливо для портативних і

енергоєфективних пристроїв. MAX98357 оптимізований для роботи від акумуляторів, що робить його ідеальним для мобільних застосувань. Підсилювач забезпечує низький рівень спотворень і шуму, що дозволяє отримати чистий і якісний звук. Плата має просту схему підключення і не вимагає додаткових компонентів для базової роботи, що полегшує процес інтеграції в проекти.

I2S (Inter-IC Sound) – це стандартний цифровий інтерфейс для передачі аудіосигналів між різними інтегральними схемами в цифрових аудіопристроях. Інтерфейс був розроблений компанією Philips для забезпечення простого і ефективного способу передачі стереоаудіо. I2S забезпечує синхронну передачу даних, де окремі лінії використовуються для передачі синхронізаційних сигналів і даних. Це дозволяє уникнути проблем із синхронізацією, які часто виникають при використанні асинхронних інтерфейсів.

Основні компоненти інтерфейсу I2S включають три основні сигнали: сигнал синхронізації (BCLK), сигнал вибору каналу (LRCLK) і дані (SDATA). BCLK визначає частоту передачі даних, LRCLK вказує, який канал передається (лівий або правий), а SDATA містить власне аудіодані. Важливою особливістю I2S є можливість передачі високоякісного аудіо з низькими спотвореннями і затримками, що робить його ідеальним для використання в сучасних цифрових аудіопристроях.

Однією з ключових переваг I2S є його широка підтримка серед різноманітних аудіо компонентів, що дозволяє легко інтегрувати його в різноманітні системи. I2S – відмінний вибір для проектів, де важлива якість звуку (рис. 2.7) і сумісність з іншими цифровими аудіопристроями.

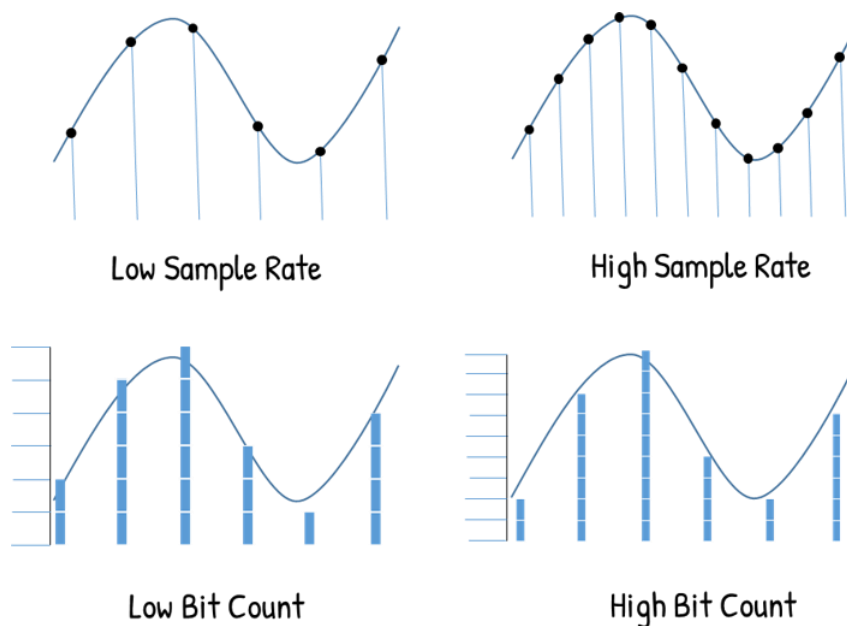


Рисунок 2.7 – Високоякісне аудіо з I2S

Таким чином, вибір плати MAX98357 з інтерфейсом I2S забезпечує високу ефективність, низьке споживання енергії та відмінну якість звуку, що робить її ідеальним рішенням для сучасних аудіопроектів (табл. 2.4). Інтерфейс I2S, у свою чергу, забезпечує надійну і якісну передачу аудіосигналів.

Таблиця 2.4 – Технічні характеристики MAX98357

Параметр	Опис
Вихідна потужність	3.2Вт на навантаженні 4Ω, 10% THD, 1.8Вт на навантаженні 8Ω, 10% THD, при живленні від 5В джерела
PSRR	77 dB типовий @ 1KHz
I2S частота семплювання	Від 8КГц до 96КГц
Сигнал MCLK	Не потрібно
Шумозаглушення	Клацання
Посилення	3dB, 6dB, 9dB, 12dB, 15dB
Захист	Від перевантаження та перегріву

Проаналізувавши технічні характеристики апаратного забезпечення та схеми розташування пінів можна перейти до проектування схеми з'єднання компонентів.

2.2 Схема з'єднання компонентів

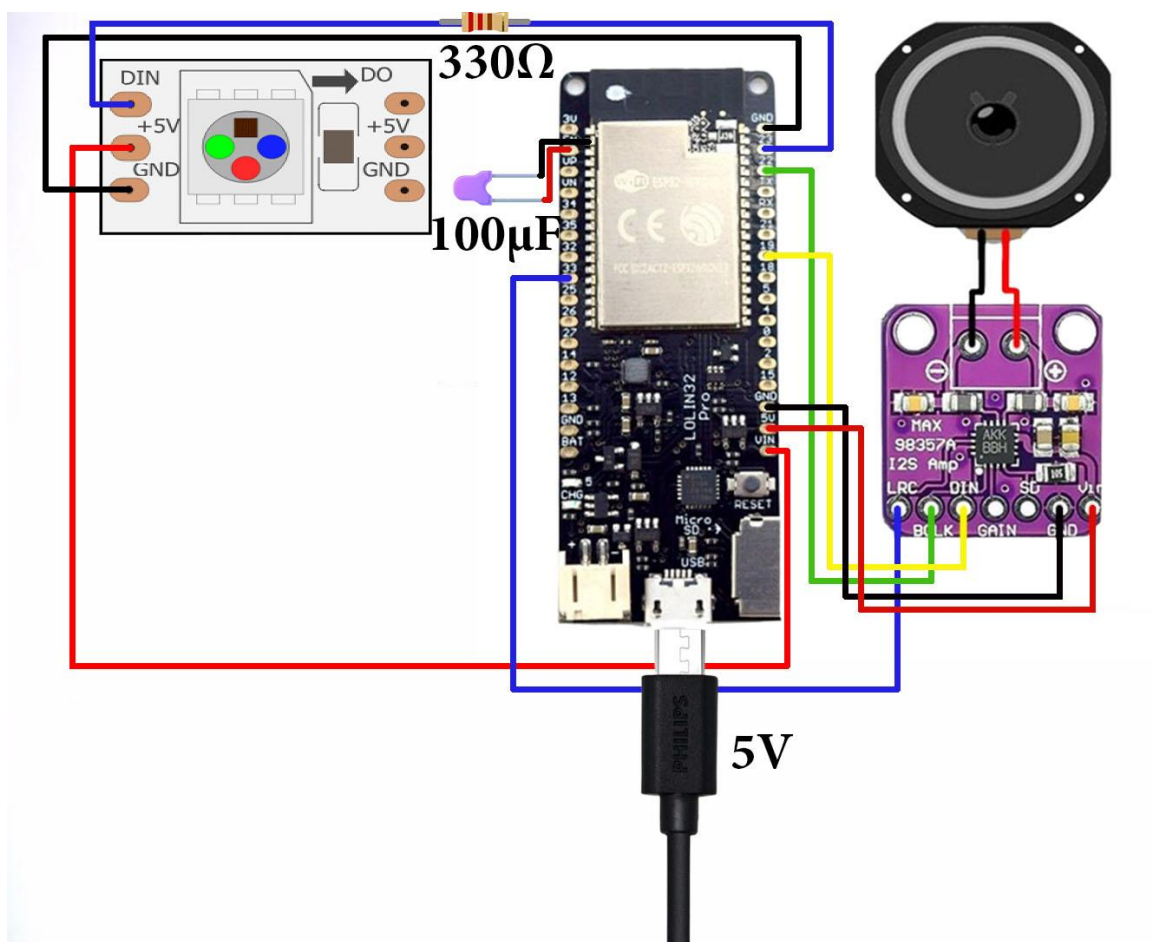


Рисунок 2.8 – Схема підключення апаратного забезпечення

ESP32 отримує живлення 5V. Вивід 5V ESP32 підключений до VDD WS2812B і VIN MAX98357, а GND до GND обох компонентів і динаміка. GPIO 25 ESP32 підключений до DIN WS2812B через резистор 330Ω. Конденсатор 100 μF підключений між VDD і GND WS2812B для стабілізації живлення. Лінії GPIO 26 (SD), GPIO 25 (LRC) і GPIO 22 (BCLK) ESP32 з'єднані з відповідними входами SD, LRC і BCLK MAX98357. Виходи LOUT і ROUT MAX98357 підключені до динаміка.

2.3 Ідеї щодо вдосконалення пристрою

Напрямами подальшого дослідження можуть бути можливості безпосереднього отримання необхідної інформації через мобільну мережу. Може бути використаний як готовий модем, що приєднується за протоколом

USB, так і окрема плата, що передає данні за допомогою UART або I2C. По характеристикам підходить, наприклад, плата SIMcom SIM7600E-H (рис. 2.9), яка крім мобільного зв'язку дозволяє через супутникову систему місцезнаходження GPS автоматично визначати територіальну громаду, для якої актуальна інформація про тривоги [11].



Рисунок 2.9 – Плата SIMcom SIM7600E-H

Також щоб зменшити вартість обладнання, можна замінити MAX98357 на DFPlayer Mini (рис. 2.10) [12].

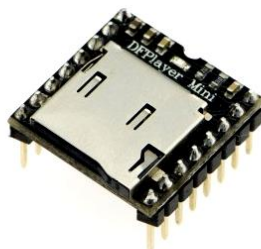


Рисунок 2.10 – Плата DFPlayer Mini

На цій платі всі звукові сигнали записуються на MicroSD картку у форматі MP3. Вибір необхідного файлу здійснюється по протоколу UART. Ця плата підтримує частоти дискретизації 8/11.025/12/16/22.05/24/32/44.1/48 кГц, має 24bit-розрядний ЦАП, підтримка динамічного діапазону: 90 дБ, співвідношення SNR: 85dB. Повністю підтримує FAT16, FAT32 файлову систему, підтримка MicroSD карти до 32 Гб. Перевагою такої реалізації є низька ціна та простота програмування вибору звукових сигналів. До недоліків можна віднести підвищену складність загального проекту та необхідність купівлі MicroSD картки.

Оскільки у військовий час не завжди є доступ до електроенергії, актуальним є пошук рішення енергетичної незалежності пристрою від електромережі. Найпростіший варіант – використання пауербанків, наприклад, Xiaomi Power bank 10000mAh 22.5W (рис. 2.11).



Рисунок 2.11 – Xiaomi Power bank 10000mAh 22.5W

Використання пристроїв безперервного живлення може бути розглянуто, але слід враховувати, що на відміну від пауербанків така модифікація суттєво підвищить вартість інтерактивного інформера.

2.4 Принцип роботи програмного забезпечення

Програмне забезпечення для інтерактивної мапи тривоги на базі ESP32 інтегрує кілька важливих функцій: підключення до мережі, отримання даних про тривоги, обробка цих даних, управління світлодіодами для візуалізації стану тривоги, та відтворення звукових сигналів за допомогою звукової плати MAX98357. Блок-схема алгоритму програмного забезпечення зображено на рис. 2.12.

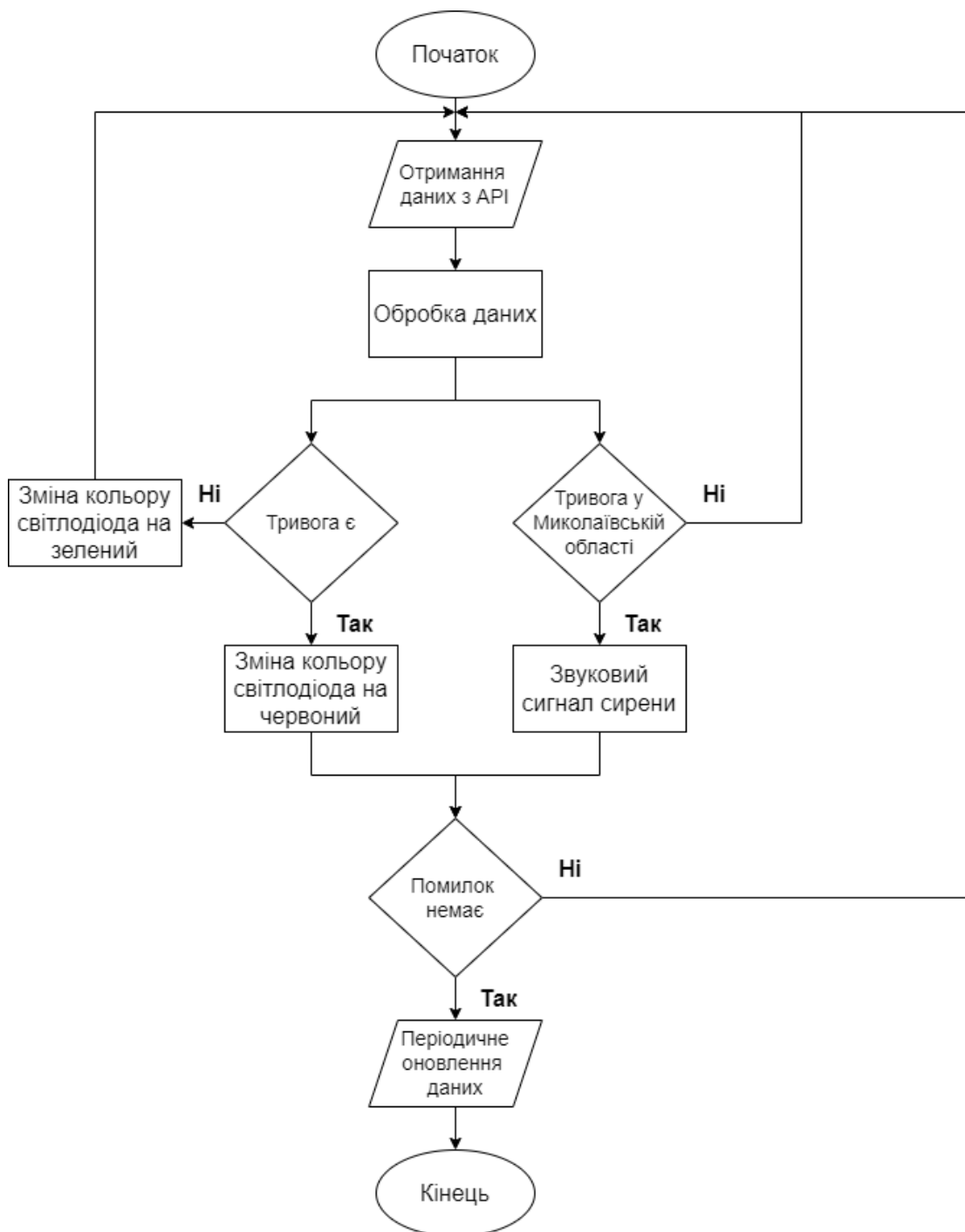


Рисунок 2.12 – Блок-схема алгоритму ПЗ

1. Ініціалізація системи

Програмне забезпечення починає свою роботу з ініціалізації мікроконтролера ESP32, що включає налаштування всіх необхідних периферійних пристроїв. Піни ESP32 налаштовуються для управління світлодіодами WS2812B та для зв'язку із звуковою платою MAX98357. ESP32

підключається до Wi-Fi мережі для доступу до Інтернету. Це підключення необхідне для отримання актуальної інформації про стан тривоги через API. Встановлюються бібліотеки для управління світлодіодами (наприклад, FastLED) і звуковим модулем (Adafruit_I2S), а також бібліотека для роботи з Wi-Fi.

2. Отримання даних про тривоги

Основна функція програмного забезпечення полягає в отриманні актуальної інформації про стан тривоги. Це здійснюється шляхом періодичних HTTP-запитів до API за адресою `map.ukrainealarm.com`. API надає дані у форматі JSON, які містять інформацію про поточний стан тривоги у різних регіонах України.

3. Обробка отриманих даних

Після отримання даних з API розбираються JSON-дані, щоб виділити інформацію про кожен регіон. Перевіряє, чи активна тривога в кожному регіоні. Для цього аналізуються відповідні поля в JSON-даних, що вказують на наявність або відсутність тривоги.

4. Управління світлодіодами

На основі оброблених даних про тривоги програмне забезпечення змінює стан світлодіодів WS2812B. Кожен світлодіод відповідає за певний регіон на мапі. Якщо в регіоні активна тривога, світлодіод світиться червоним кольором. Якщо тривоги немає - зеленим або іншим кольором. Резистори використовуються для стабілізації живлення світлодіодів, що забезпечує їх коректну роботу.

5. Звукове оповіщення

Для звукового сповіщення програмне забезпечення використовує плату MAX98357. Якщо у Миколаївській області активна тривога, MAX98357 відтворює попередньо записаний звуковий файл (сирену). Звуковий файл зберігається у вбудованій пам'яті ESP32 або на підключеному зовнішньому сховищі. ESP32 зчитує звукові файли з пам'яті і передає їх на MAX98357 для

відтворення, що дозволяє відтворювати, зупиняти або перемикати звукові файли.

6. Обробка помилок та відновлення з'єднання

Якщо з'єднання з Wi-Fi втрачається, програма намагається підключитися знову. У випадку помилки при запиті до API, програма робить повторні спроби отримати дані і продовжує свою роботу після відновлення доступу.

7. Періодичне оновлення даних

Програмне забезпечення налаштоване на періодичне оновлення даних через певні інтервали часу (кожні 60 секунд). Програма виконує повторні запити до API, щоб отримати актуальні дані про стан тривоги. Отримані нові дані обробляються, і на основі них оновлюється стан світлодіодів та звукове сповіщення.

Діаграма прецедентів (рис. 2.13), також відома як діаграма варіантів використання або Use Case Diagram – це тип UML (Unified Modeling Language) діаграми, яка відображає взаємодію користувачів (акторів) із системою для виконання певних дій (прецедентів або варіантів використання). Вона призначена для опису функціональних вимог до системи з точки зору кінцевих користувачів.

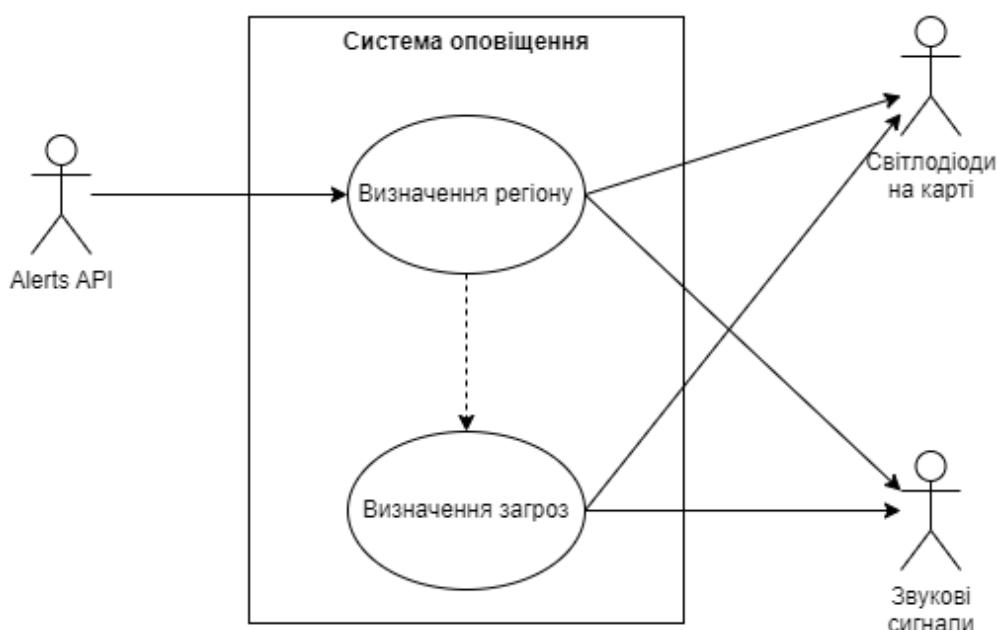


Рисунок 2.13 – UML діаграма прецедентів

Alerts API взаємодіє з системою оповіщення шляхом надсилання даних про тривоги. Цей компонент відповідає за отримання актуальної інформації про стан тривоги через HTTP-запити до API.

Система оповіщення – основна система, що включає кілька підсистем. Визначення регіону приймає дані від API і ідентифікує регіон, для якого призначена інформація.

Визначення загроз на основі отриманих даних аналізує наявність тривоги в конкретному регіоні. Це передбачає розбір JSON-даних і визначення активної тривоги.

Світлодіоди на карті отримують дані від підсистеми визначення загроз і змінюють свій стан залежно від наявності тривоги. Якщо в регіоні активна тривога, відповідний світлодіод світиться червоним кольором; якщо тривоги немає – зеленим.

Звукові сигнали використовують дані від підсистеми визначення загроз для відтворення звукового сигналу (сирени) у разі активної тривоги у Миколаївській області.

Діаграма прецедентів надає візуальне уявлення про те, як різні компоненти системи взаємодіють між собою, що допомагає зрозуміти загальну архітектуру та логіку роботи системи.

2.4.1 Опис бібліотек Arduino IDE

WiFi.h – ця бібліотека дозволяє мікроконтролеру ESP32 підключатися до Wi-Fi мережі, що необхідно для доступу до Інтернету і виконання HTTP-запитів до API. Основні функції включають налаштування мережі SSID та паролю, підключення до мережі та перевірку статусу підключення.

HTTPClient.h – бібліотека, що забезпечує можливість виконання HTTP-запитів, таких як GET, для отримання даних з веб-серверів. Вона використовується для відправки запитів до API тривоги та отримання JSON-даних з відповіддю.

FastLED.h – ця бібліотека використовується для управління адресованими світлодіодами, такими як WS2812B. Вона забезпечує прості засоби для налаштування кольорів кожного світлодіода та оновлення їх стану, що дозволяє легко візуалізувати стан тривог на мапі.

Adafruit_I2S.h – ця бібліотека, використовується для роботи з I2S інтерфейсом. З її допомогою можна передавати аудіо дані до підсилювача, такого як MAX98357.

ArduinoJson.h – ця бібліотека використовується для парсингу JSON-даних, отриманих від API. Вона дозволяє легко розбирати JSON-структури та отримувати доступ до необхідних даних, таких як стан тривог для кожного регіону.

ПЗ інтерактивної мапи тривог на базі ESP32 забезпечує постійний моніторинг стану тривог в усіх регіонах України, візуалізує цю інформацію за допомогою світлодіодів та доповнює її звуковими сигналами (якщо тривога відбувається у Миколаївській області). Така система дозволяє оперативно реагувати на зміни в стані тривог та забезпечує наочне і звукове сповіщення користувачів.

Висновок до розділу 2

Проаналізувавши технічні характеристики апаратного забезпечення, схеми підключення компонентів та функції програмного забезпечення для інтерактивної мапи тривог на базі ESP32, можна зробити кілька ключових висновків. ESP32, завдяки своїй потужній архітектурі та широкому набору інтерфейсів, є оптимальним вибором для керування системою. Він підтримує підключення до Wi-Fi, що дозволяє отримувати актуальну інформацію про стан тривог через API. Світлодіоди WS2812B забезпечують яскраве і наочне відображення стану тривог у різних регіонах України, а модуль MAX98357 додає можливість звукового сповіщення у випадку, якщо тривога лунає у Миколаївській області.

Програмне забезпечення включає наступні етапи: ініціалізацію системи, підключення до Wi-Fi, отримання та обробку даних про тривоги, управління світлодіодами та відтворення звукових сигналів. Для реалізації цих функцій використовуються різні бібліотеки Arduino IDE, такі як WiFi.h для підключення до мережі, HTTPClient.h для виконання HTTP-запитів, FastLED.h для управління світлодіодами, Adafruit_I2S.h для роботи зі звуковим модулем та ArduinoJson.h для парсингу JSON-даних.

Ця інтеграція апаратного та програмного забезпечення забезпечує постійний моніторинг стану тривоги, візуалізацію цієї інформації за допомогою світлодіодів та звукове сповіщення про тривоги у визначених регіонах. Система дозволяє оперативно реагувати на зміни в стані тривоги, забезпечуючи наочне і звукове сповіщення користувачів. Завдяки стабільному живленню та механізмам обробки помилок, система залишається надійною та ефективною в роботі.

3 АПАРАТНО-ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

У цьому розділі реалізовано апаратно-програмний комплекс, що відповідає вимогам до компонентів та функціональності пристрою, базуючись на висновках минулих розділів.

3.1 Апаратна частина

Головним елементом або «мозком» пристрою є мікроконтролер ESP32 (рис. 3.1). Безпосередньо до нього будуть підключатися усі модулі та плати.

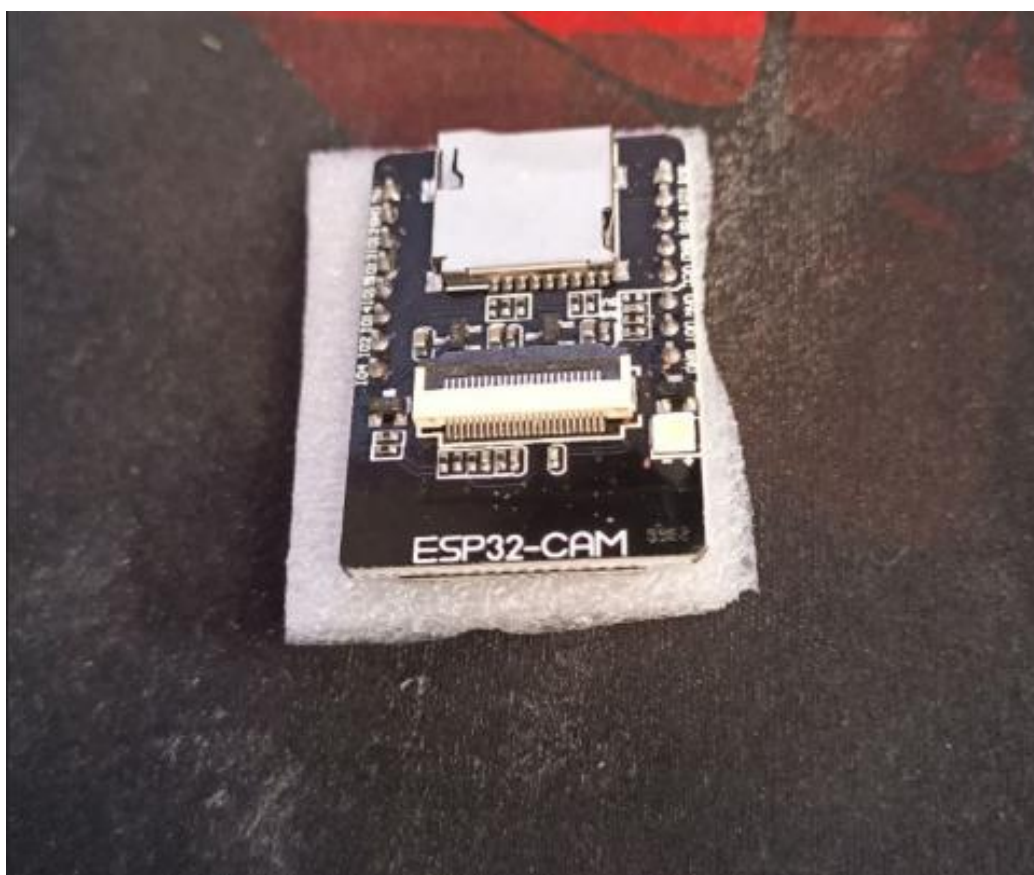


Рисунок 3.1 – Мікроконтролер ESP32

З'єднання буде відбуватися за допомогою кабелів типу Dupont. Кабелі типу Dupont (іноді їх називають Jumper Wires) широко використовуються для прототипування електронних схем, особливо з мікроконтролерами (рис. 3.2).

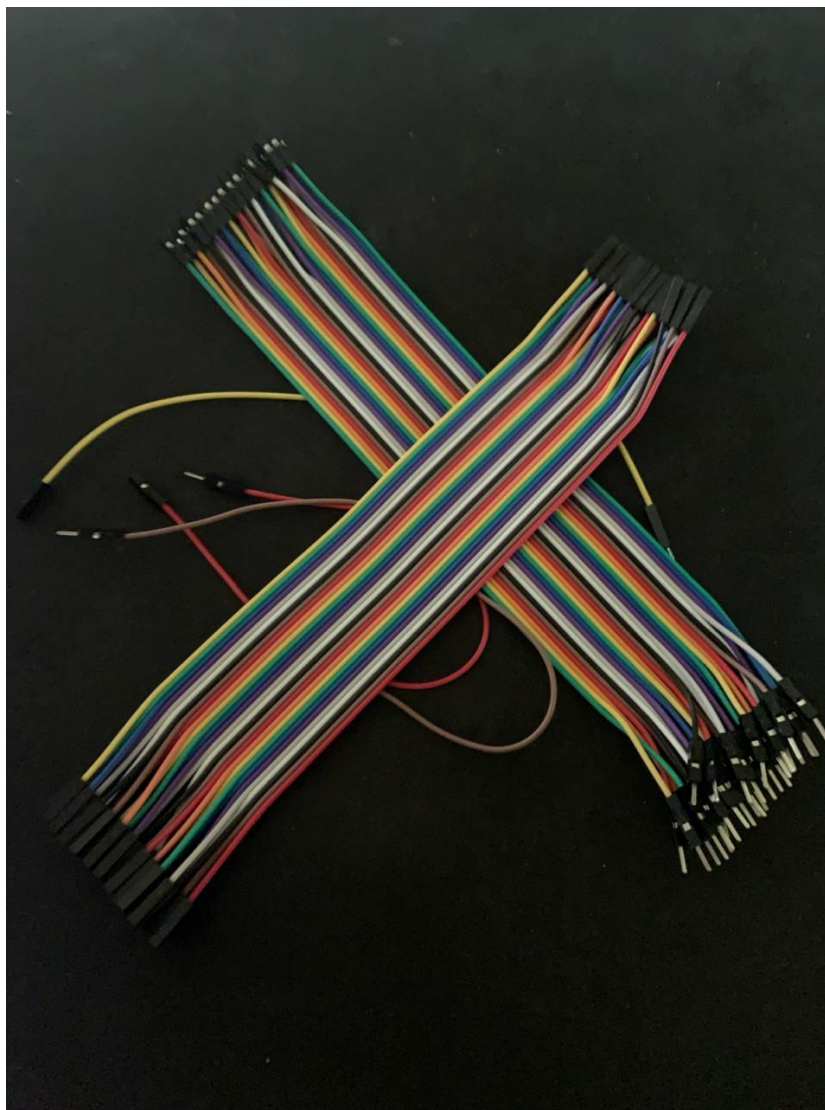


Рисунок 3.2 – Кабелі типу Dupont

Вони забезпечують швидке з'єднання без пайки. Є три основні типи: Male-to-Male (M-M) для з'єднання між піновими роз'ємами на макетній платі або компонентами, Male-to-Female (M-F) для з'єднання пінів мікроконтролера з іншими компонентами, і Female-to-Female (F-F) для з'єднання пінів між собою. Кабелі бувають різної довжини (зазвичай від 10 см до 30 см), різних кольорів для зручного маркування, і зазвичай виготовляються з мідного проводу для хорошої провідності.

Схеми з'єднання будуть побудовані у Fritzing. Fritzing — це вільне програмне забезпечення з відкритим вихідним кодом, призначене для проектування електронних схем і друкованих плат (PCB). Воно створене для того, щоб полегшити процес проектування та створення прототипів

електронних пристроїв як для початківців, так і для професіоналів. Інтерфейс Fritzing інтуїтивно зрозумілий і дозволяє користувачам легко моделювати схеми, розробляти макети на макетних платах, генерувати схеми з'єднань і проектувати друковані плати.

Однією з основних переваг Fritzing є його здатність автоматично генерувати схеми та макети з фізичних прототипів. Це значно спрощує процес документування проектів і підготовки їх до виробництва. Програма підтримує експорт файлів у форматі, що використовується для виробництва друкованих плат, а також створення документації, що робить її корисною для розробників, які працюють над реальними продуктами.

Крім того, Fritzing має велику бібліотеку компонентів, що дозволяє швидко знайти та додати необхідні елементи до схеми. Користувачі також можуть створювати свої власні компоненти, якщо вони не знайдуть потрібні у вбудованій бібліотеці. Це програмне забезпечення широко використовується в освітніх цілях, оскільки воно дозволяє студентам та початківцям легко зрозуміти основи електроніки і створювати власні проекти.

Підключення адресного світлодіоду WS2812B до ESP32 зображено на рис. 3.3.

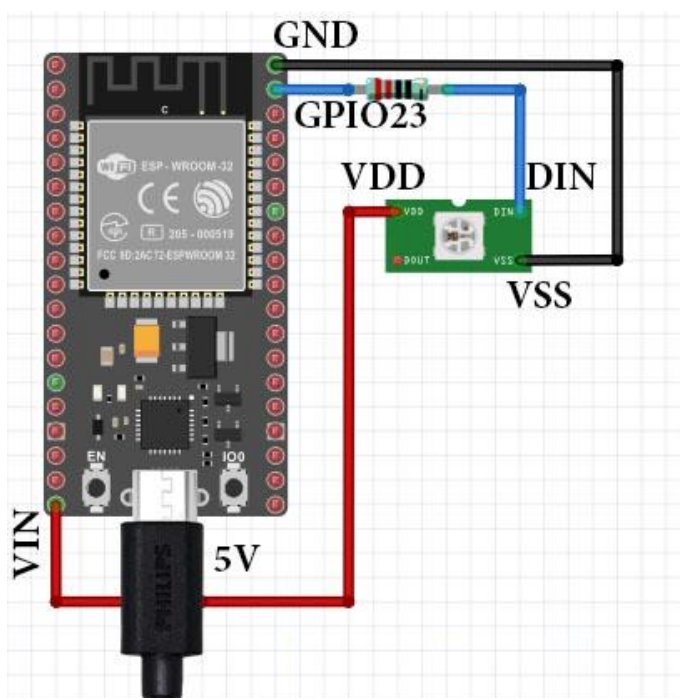


Рисунок 3.3 – Макетна схема підключення ESP32 та WS2812B

Живлення світлодіода забезпечується від джерела 5V (VIN): червоний провід підключений до контакту "+5V" на світлодіоді WS2812B. Контакт "GND" ESP32 з'єднаний з контактом "GND" на світлодіоді та контактом "GND" на джерелі живлення. Дані передаються через жовтий провід, що підключений до контакту "GPIO23" ESP32 і через резистор 330 Ом до контакту "DIN" світлодіода WS2812B.

Підключення звукової I2S плати (рис. 3.4) до ESP32 зображено на рис 3.5.

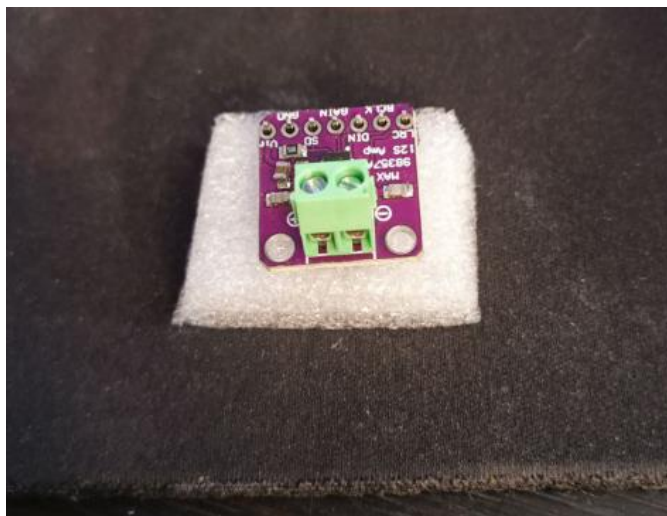


Рисунок 3.4 – Звукова плата MAX98357

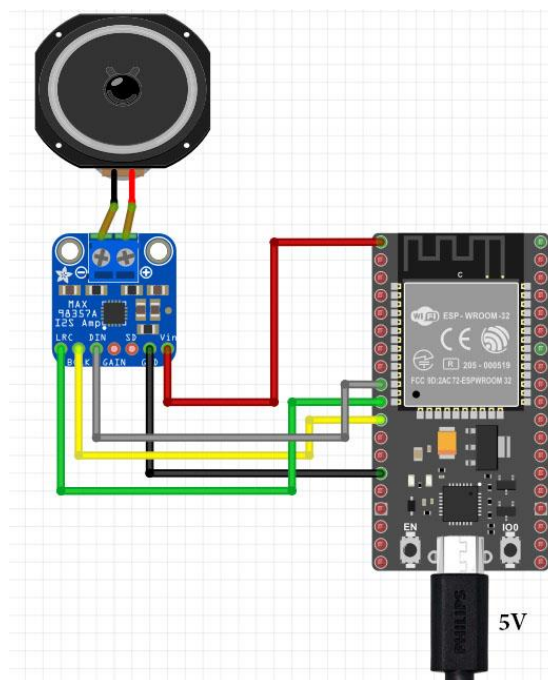


Рисунок 3.5 – Макетна схема підключення ESP32, MAX98357 та динаміку

Живлення забезпечується від ESP32: контакт "5V" (червоний провід) підключений до контакту "VIN" на MAX98537, а контакт "GND" (чорний провід) з'єднаний з контактом "GND" на MAX98537. Дані передаються через сірий провід від контакту "GPIO25" ESP32 до контакту "DIN" на MAX98537, зелений провід від контакту "GPIO26" ESP32 до контакту "LRC" на MAX98537, і жовтий провід від контакту "GPIO27" ESP32 до контакту "BCLK" на MAX98537. Динамік підключений до відповідних виходів на MAX98537.

На рис. 3.6 представлена макетна схема підключення всіх компонентів інформеру на макетній платі. Вона використовується для візуального розуміння з'єднань та допомагає уникнути помилок при підключенні компонентів.

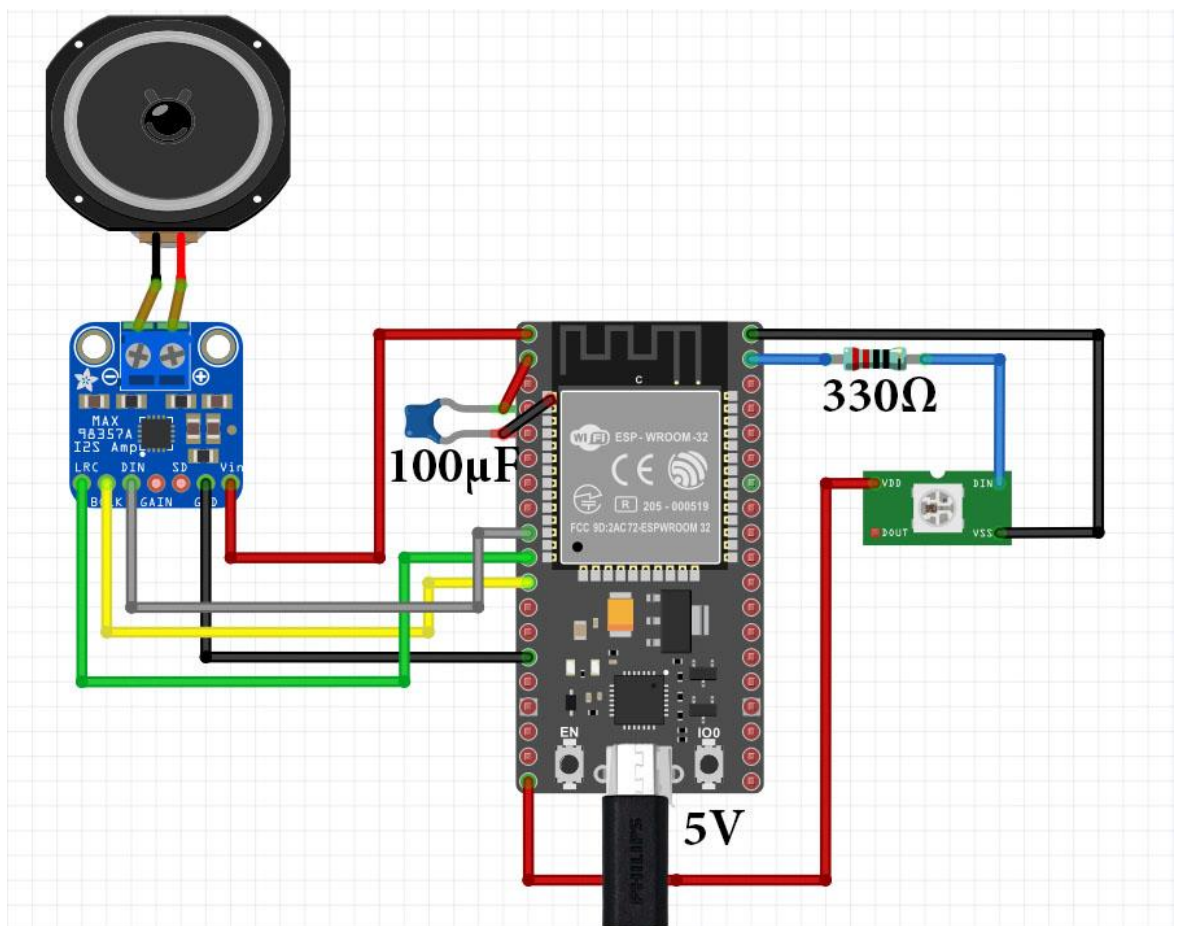


Рисунок 3.6 – Макетна схема інформеру

На рис. 3.7 зображена принципова електрична схема, яка відображає логічні з'єднання між усіма компонентами інформеру. Це основний документ,

який використовується для аналізу та перевірки правильності електричних з'єднань у схемі.

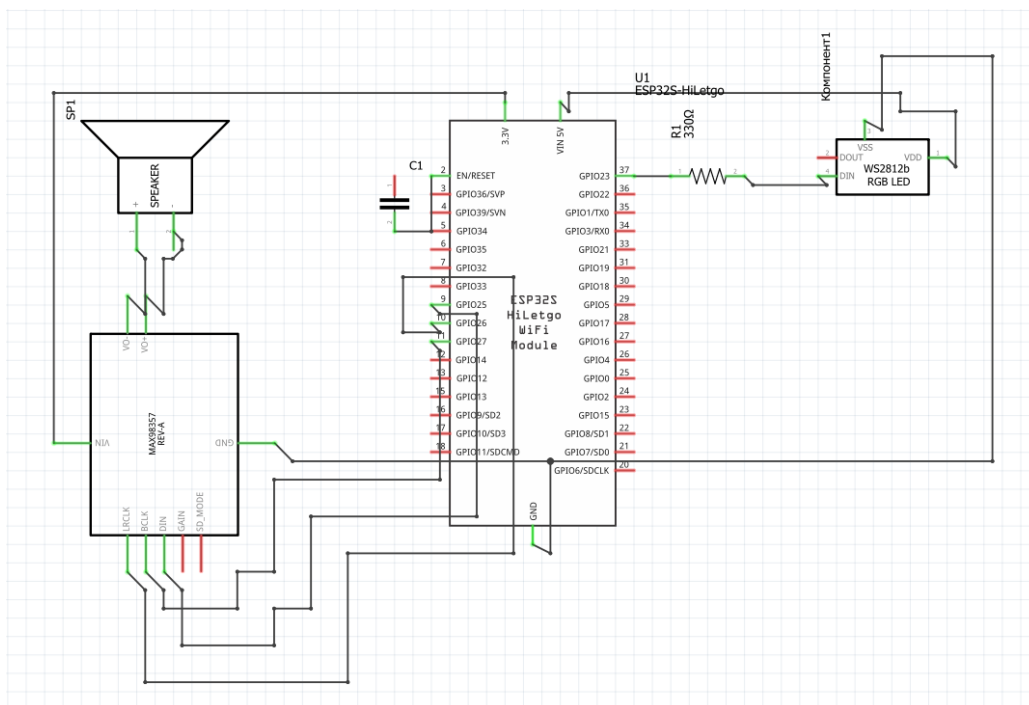


Рисунок 3.7 – Принципова електрична схема

Рис. 3.8 показує розміщення компонентів на друкованій платі. Це важливий етап проектування, який дозволяє підготувати пристрій до виробництва та забезпечити правильне з'єднання всіх компонентів на фізичному рівні.

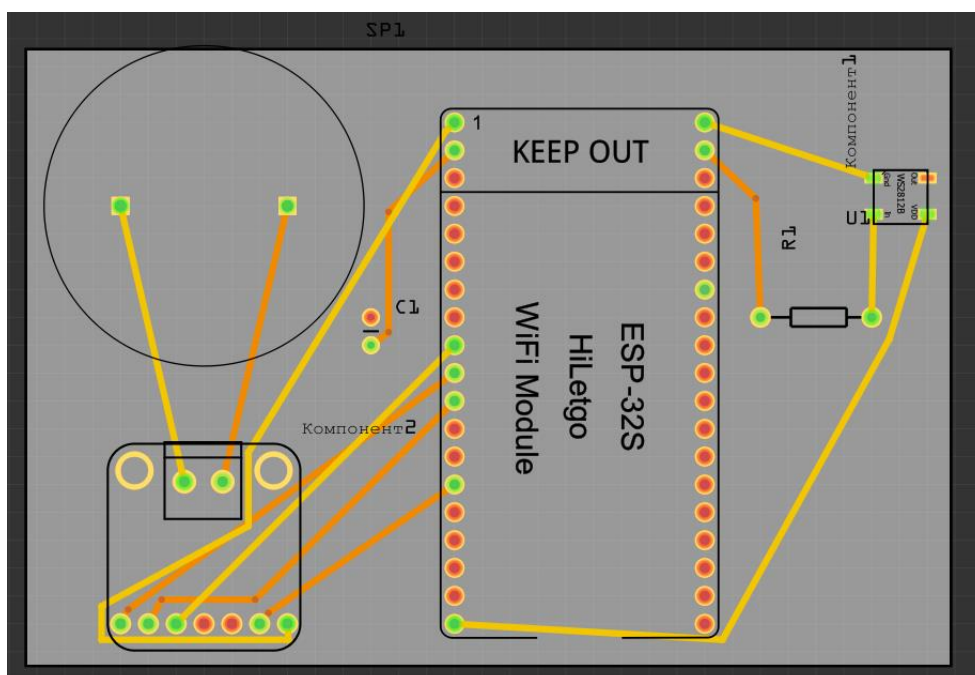


Рисунок 3.8 – Розміщення на друкованій платі

У цьому підрозділі було розглянуто процес підключення основних компонентів інформеру на базі мікроконтролера ESP32. Було детально описано використання макетних схем для початкового прототипування, принципів електричних схем для розуміння та аналізу логічних з'єднань, а також розміщення компонентів на друкованій платі для підготовки до виробництва. Використання програмного забезпечення Fritzing значно спрощує процес проектування та документування електронних схем, роблячи його доступним як для початківців, так і для професіоналів.

3.1.1 Корпус розробки

Корпусом пристрою було обрано надруковану на 3D принтері мапу України, що складається з двох частин. На рис. 3.9 зображено верхню частину надруковану з прозорого пластику.

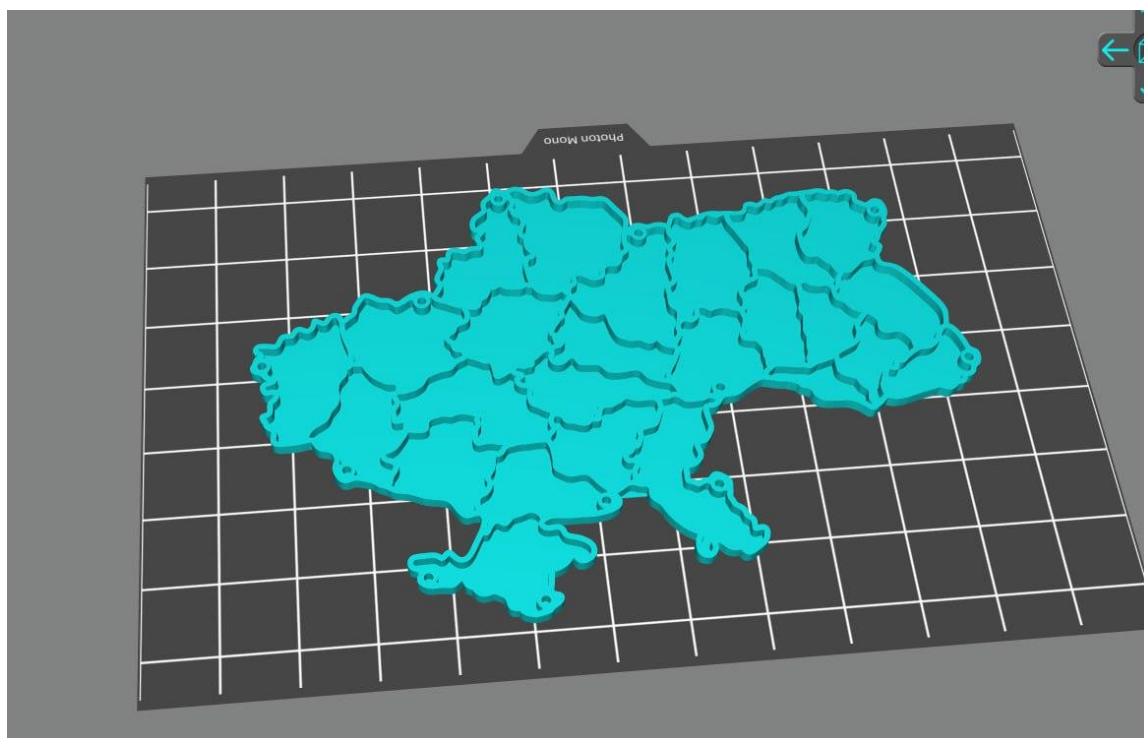


Рисунок 3.9 – Верхня частина корпусу

Нижня частина корпусу (рис. 3.10) є віддзеркаленою копією верхньої, надрукована з чорного пластику.



Рисунок 3.10 – Нижня частина корпусу

Кожен регіон являє собою комірку для адресного світлодіоду WS2812B.

3.2 Кошторисна вартість

Оскільки основною метою проекту є створення відносно дешевого інформеру – необхідним кроком є створення кошторису, що дозволить оцінити приблизну вартість пристрою загалом та кожного компонента окремо. Кошторис наведено у табл 3.1.

Таблиця 3.1 – Кошторис

Найменування	К-ть	Од. Вим	Ціна, грн	Вартість, грн
ESP32	1	шт.	229	229
WS2812B	25	шт.	4,5	112,5
MAX98357	1	шт.	76	76
ХНХDZ-27мм Динамік 8Ом, 0.25Вт	1	шт.	18	18
Резистор 330 кОм 5% 1Вт вив.	10	шт.	1,2	1,2
Конденсатор 100мкФ 16В 105С 1	10	шт.	45	45
Кабель Dupont 20 см.	40	шт.	75	75
			РАЗОМ	556,7

На основі розробленого кошторису, загальна вартість компонентів для створення інформеру складає 556,7 гривень. Найдорожчими елементами є ESP32 (229 грн) та набір світлодіодів WS2812B (112,5 грн). Інші компоненти, такі як аудіопідсилювач MAX98357, динамік, резистори, конденсатори та кабелі, мають меншу вартість, але також відіграють важливу роль у функціонуванні пристрою. Враховуючи всі витрати, можна стверджувати, що даний інформер є відносно недорогим, що відповідає основній меті проєкту – створення бюджетного пристрою.

3.3 Отримання даних з API

Перед тим як розпочати роботу з програмною частиною пристрою потрібно спочатку отримати токен для того щоб працювати з **API**.

API (Application Programming Interface) – це набір правил і механізмів, який дозволяє різним програмам взаємодіяти одна з одною. API виступає посередником, який дозволяє одній програмі запитувати дані або функції в іншій програмі або сервісу та отримувати необхідні результати. Основними поняттями API є **ендпоінти**, **запити**, **відповіді** та **токени доступу**.

Ендпоінт – це конкретна URL-адреса для запиту до API.

Запит – це HTTP-запит (GET, POST, PUT, DELETE), який відправляється до API для отримання або надсилання даних.

Відповідь – дані, які повертаються від API у відповідь на запит, зазвичай у форматі JSON або XML

Токен доступу – унікальний ключ для автентифікації і авторизації запитів до API.

У цій роботі використано API сайту map.ukrainealarm.com (рис. 3.11).

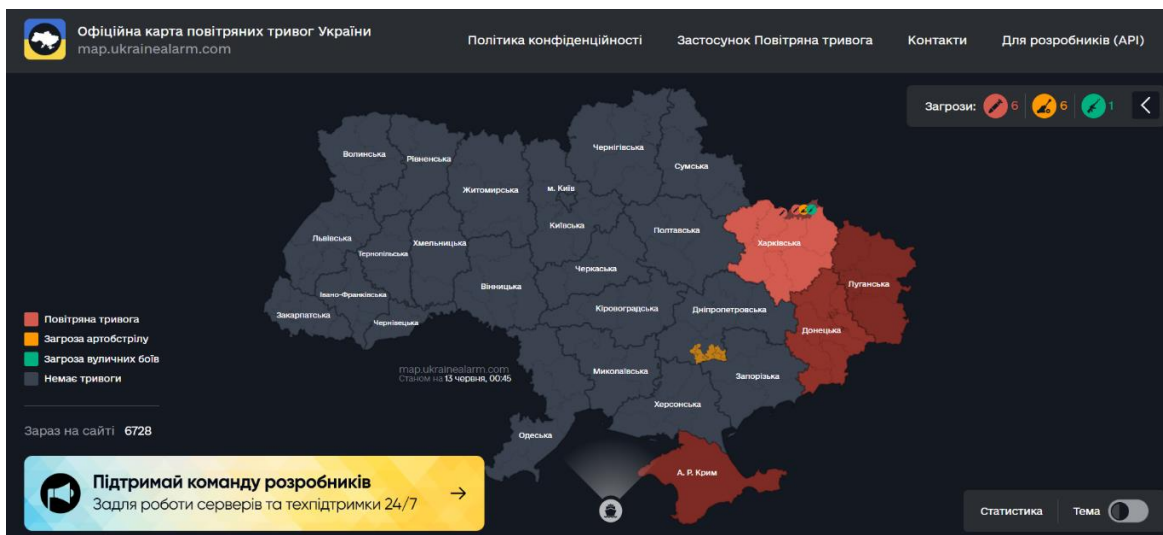


Рисунок 3.11 – Головна сторінка сайту

Якщо перейти за посиланням «Для розробників (API)», відкриється сторінка з формою (рис. 3.12), після заповнення якої, на електронну пошту буде надіслано відповідний токен (рис. 3.13).

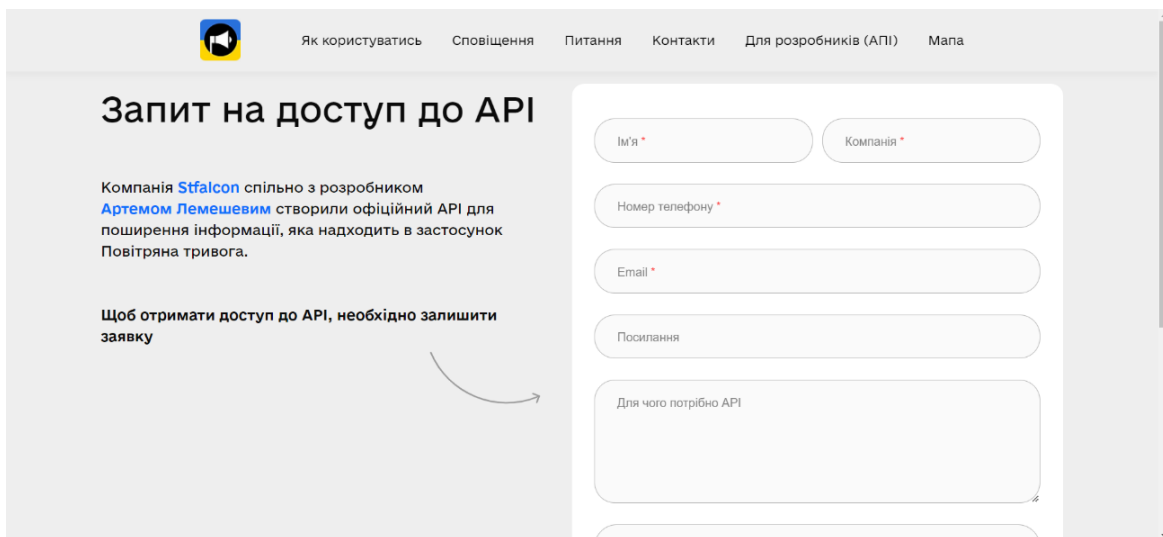


Рисунок 3.12 – Форма запиту на доступ до API

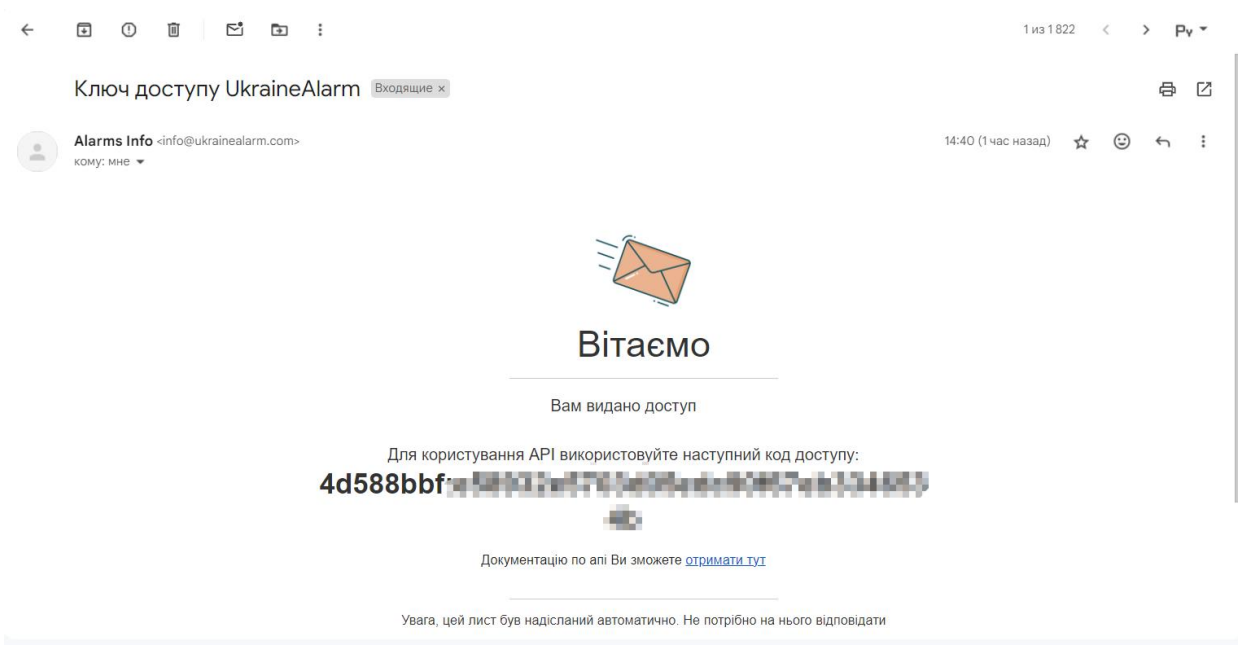


Рисунок 3.13 – Успішно отриманий токен

Після отримання токена можна будувати повноцінну програмну частину системи. За допомогою нього можна отримувати, обробляти дані та відтворювати те, що було отримано, безпосередньо на апаратній частині пристрою.

3.4 Встановлення необхідних бібліотек

Бібліотека — це набір коду, який надає додаткові функції для використання в програмуванні, наприклад, під час роботи з обладнанням або маніпулювання даними. Встановити бібліотеку можна трьома способами: за допомогою менеджера бібліотек, із .zip архіву, вручну.

3.4.1 Менеджер бібліотек

Для того щоб встановити бібліотеку цим способом потрібно:

1. Відкрити Library Manager: на панелі меню обрати Tools > Manage Libraries...
2. Відфільтрувати доступні бібліотеки, ввівши щось (наприклад, назву бібліотеки) у текстове поле над бібліотеками в списку.

- Знайти бібліотеку в результатах пошуку. Результати відображаються в алфавітному порядку, тому можливо знадобитися прокрутити список вниз.
- Знайти бібліотеку, яку потрібно встановити. Можна переглянути опис та автора. Коли бібліотека була знайдена, потрібно натиснути «Встановити». За замовчуванням вибрано останню версію.

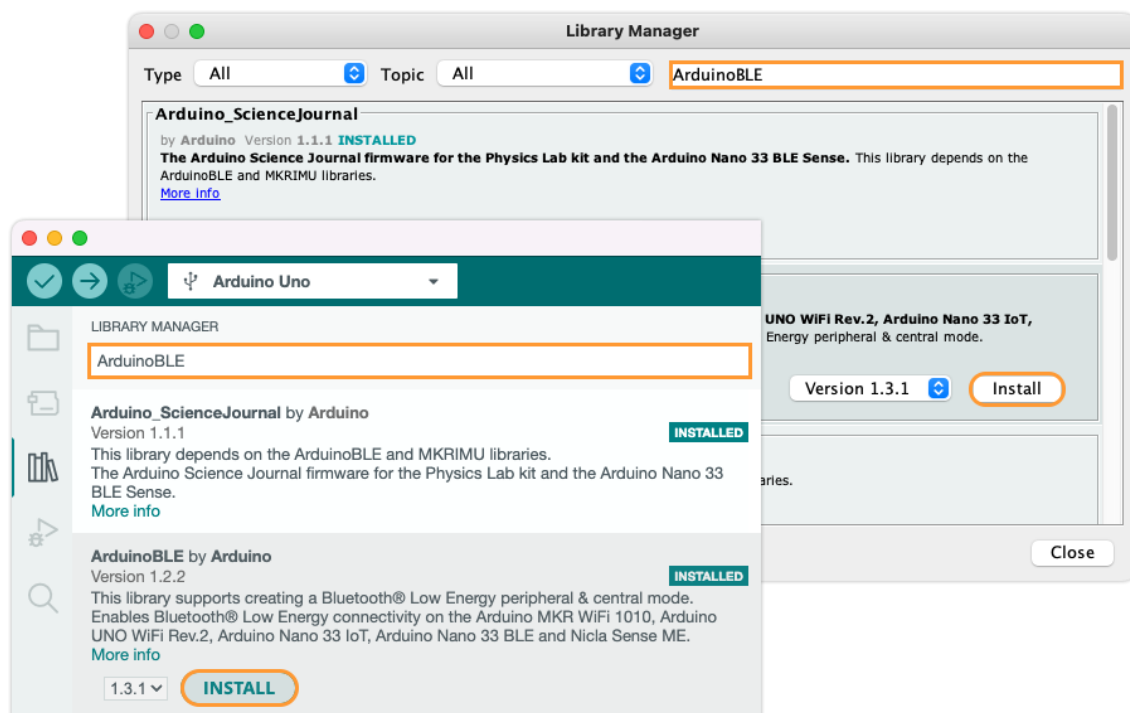


Рисунок 3.14 – Приклад встановлення бібліотеки Arduino IDE, використовуючи Library Manager

- Дочекатися завершення інсталяції.

3.4.2 Імпорт із .zip архіву

Встановлення бібліотеки способом імпортування із .zip архіву передбачає наступні кроки:

- На панелі меню обрати Sketch > Include Library > Add .ZIP Library...
- Вам буде запропоновано вибрати бібліотеку, яку ви хочете додати. Перейдіть до розташування файлу .zip і відкрийте його.
- У більшості випадків може знадобитися перезапустити Arduino IDE, щоб бібліотека стала доступною.

3.4.3 Ручна установка

Наступні кроки описують процес інсталяції бібліотеки до Arduino IDE вручну:

1. Потрібно перейти до каталогу, у якому знаходиться файл бібліотеки .zip.
2. Розпакувати файл .zip.
3. Вибрати головну папку (вона має містити назву бібліотеки) і перемістити її до папки «libraries».
4. Перезапустити Arduino IDE.

3.5 Програмна частина

Код для проекту інтерактивної мапи тривоги на базі ESP32 написаний на мові програмування C++ з використанням Arduino IDE.

C++ є однією з найпотужніших і найпоширеніших мов програмування, створеною Б'ярне Страуструпом як розширення мови C. Використання C++ дозволяє ефективно працювати з апаратними ресурсами ESP32, забезпечуючи високий рівень продуктивності.

Arduino IDE використовує спрощений синтаксис C++. Використовуючи який, можна писати програми, які виконують періодичні HTTP-запити до API для отримання даних про тривоги, оброблять ці дані для визначення стану тривоги у різних регіонах України та керувати світлодіодами і звуковим модулем на основі цих даних.

Опис програмної частини розробки:

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <FastLED.h>
#include <Adafruit_I2S.h>
```

На початку коду підключаємо необхідні бібліотеки для роботи з Wi-Fi, виконання HTTP-запитів, обробки JSON-даних, керування світлодіодами та звуковим модулем.

Для підключення до інтернету та отримання даних про тривоги необхідно задати параметри Wi-Fi мережі, адресу API та токен.

```
const char* ssid = "YOUR_SSID";  
const char* password = "YOUR_PASSWORD";  
  
const char* apiUrl = "http://map.ukrainealarm.com/api/alarms";  
const char* apiKey = "YOUR_TOKEN";
```

Далі налаштовуємо пін для світлодіодів та їх кількість, а також ініціалізуємо звуковий модуль.

```
#define LED_PIN    5  
#define NUM_LEDS  25  
CRGB leds[NUM_LEDS];
```

```
Adafruit_I2S i2s;
```

Задаємо максимальну довжину відповіді від API та інтервал оновлення даних (60 секунд).

```
const int capacity = JSON_ARRAY_SIZE(NUM_LEDS) + JSON_OBJECT_SIZE(1);
```

```
const long interval = 60000;  
unsigned long previousMillis = 0;
```

У функції setup() виконується ініціалізація світлодіодів, підключення до Wi-Fi та ініціалізація звукового модуля.

```
void setup() {  
  Serial.begin(115200);  
  
  FastLED.addLeds<NEOPIXEL, LED_PIN>(leds, NUM_LEDS);  
  
  WiFi.begin(ssid, password);  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(1000);  
    Serial.println("Connecting to WiFi...");  
  }  
  Serial.println("Connected to WiFi");  
  
  if (!i2s.begin()) {  
    Serial.println("Failed to initialize I2S");  
    while (1);  
  }  
}
```

Основний цикл програми перевіряє, чи настав час для оновлення даних, і викликає функцію updateAlarms() для оновлення стану тривоги.

```
void loop() {  
  unsigned long currentMillis = millis();
```



```
if (currentMillis - previousMillis >= interval) {  
    previousMillis = currentMillis;  
    updateAlarms();  
}  
}
```

Функція `updateAlarms()` виконує HTTP-запит до API для отримання даних про тривоги, обробляє отримані JSON-дані та оновлює стан світлодіодів.

```
void updateAlarms() {  
    if (WiFi.status() == WL_CONNECTED) {  
        HTTPClient http;  
        http.begin(apiUrl);  
        int httpResponseCode = http.GET();  
  
        if (httpResponseCode > 0) {  
            String response = http.getString();  
            DynamicJsonDocument doc(capacity);  
            deserializeJson(doc, response);  
  
            JSONArray alarms = doc["alarms"];  
            for (int i = 0; i < NUM_LEDS; i++) {  
                bool alarmActive = alarms[i]["alarm_active"];  
                if (alarmActive) {  
                    leds[i] = CRGB::Red;  
                } else {  
                    leds[i] = CRGB::Green;  
                }  
                FastLED.show();  
  
                // Звукове сповіщення для Миколаївської області  
                if (i == 15 && alarmActive) { // Припускаємо, що 15-ий індекс відповідає  
Миколаївській області  
                    playSiren();  
                }  
            }  
        } else {  
            Serial.print("Error on HTTP request: ");  
            Serial.println(httpResponseCode);  
        }  
        http.end();  
    } else {  
        Serial.println("WiFi not connected");  
        WiFi.reconnect();  
    }  
}
```

Функція `playSiren()` генерує простий тон, який відтворюється за допомогою звукового модуля. Ця функція використовується для звукового сповіщення при активній тривозі в Миколаївській області.

```
void playSiren() {  
    const int frequency = 1000; // Частота тону  
    const int duration = 5000; // Тривалість у мілісекундах  
    const int sampleRate = 44100; // Частота дискретизації  
  
    for (int i = 0; i < (duration * sampleRate) / 1000; i++) {  
        int16_t sample = 32767 * sin(2 * PI * frequency * i / sampleRate);  
        i2s.write(sample);  
    }  
}
```

Мова програмування C++ та середовище розробки Arduino IDE дозволяють легко реалізувати інформер про тривоги, забезпечуючи високу продуктивність і гнучкість у роботі з апаратними ресурсами.

3.5.1 Завантаження коду на ESP32

Щоб завантажити код на ESP32 за допомогою Arduino IDE, спочатку потрібно підключити ESP32 до комп'ютера за допомогою USB-кабелю. Відкрити Arduino IDE та обрати плату ESP32 зі списку плат у меню Tools -> Board. Потім обрати відповідний COM-порт у меню Tools -> Port, який відповідає підключеній платі.

Відкрити або створити новий скетч у Arduino IDE. Наприклад, можна використовувати стандартний приклад "Blink", який можна знайти у меню File -> Examples -> 01.Basics -> Blink. Цей приклад змусить вбудований світлодіод на ESP32 мигати.

Потрібно переконатися, що обрана плата і порт правильні, після чого натиснути кнопку Upload (стрілка вправо) у верхній частині вікна Arduino IDE. Далі можна побачити, як компілюється код і завантажується на ESP32. Під час цього процесу на платі будуть блимати світлодіоди, що вказують на передачу даних.

Після завершення завантаження буде виведено повідомлення Done uploading в нижній частині вікна Arduino IDE. Тепер код виконуватиметься на

ESP32, і можна буде побачити очікувану поведінку, наприклад, миготіння світлодіода у випадку з прикладом "Blink".

Висновок до розділу 3

У цьому розділі було детально розглянуто апаратно-програмну реалізацію інформеру на базі мікроконтролера ESP32. Описано основні компоненти, такі як ESP32, адресні світлодіоди WS2812B, аудіопідсилювач MAX98357, динамік та інші елементи. Було створено кошторис, що підтвердив бюджетний характер проекту з загальною вартістю 556,7 грн. Основними витратами є ESP32 і набір світлодіодів WS2812B, що відповідає цілям створення доступного інформеру.

Завдяки використанню програми Fritzing, спрощено процес проектування схем та документування, що дозволяє ефективно створювати прототипи як для початківців, так і для професіоналів. Крім того, описано процедуру отримання токена доступу до API сайту map.ukrainealarm.com, що є критичним для подальшої роботи з програмною частиною пристрою.

Для розробки програмного забезпечення були встановлені необхідні бібліотеки через Library Manager, імпортовані з .zip архіву та встановлені вручну, що забезпечило гнучкість у налаштуванні середовища розробки Arduino IDE. Було створено програмну частину, що використовує мову C++ та Arduino IDE для взаємодії з апаратною частиною, виконання HTTP-запитів до API, обробки даних та керування світлодіодами і звуковим модулем. Це дозволяє періодично оновлювати дані про тривоги в Україні, відображати їх на світлодіодах та надавати звукове сповіщення при активних тривогах.

Ці кроки гарантують, що всі компоненти працюватимуть злагоджено і відповідають функціональним вимогам проекту. В цілому, проект інформеру є завершеним з урахуванням бюджетних обмежень і високої функціональності, що забезпечує його ефективне використання в реальних умовах.

ВИСНОВКИ

У процесі дослідження було вирішено низку завдань, що дозволили створити інтерактивну мапу, яка здатна оперативно відображати інформацію про тривоги в різних регіонах України.

Проведено аналіз існуючих систем інформування, що дозволило визначити їхні переваги та недоліки, а також сформулювати вимоги до нової системи. Вивчено можливості мікроконтролера ESP32, який обрано як основу для розробки інтерактивної мапи. Його високі технічні характеристики, вбудований Wi-Fi модуль та достатня кількість GPIO пінів забезпечують необхідну функціональність та надійність роботи.

Розроблено програмне забезпечення для інтерактивної мапи тривог, що включає ініціалізацію системи, підключення до мережі, отримання та обробку даних про тривоги через API, управління світлодіодами WS2812B та відтворення звукових сигналів за допомогою модуля MAX98357. Це забезпечує візуальне та звукове інформування населення в режимі реального часу.

Інтеграція з системами сповіщення дозволила створити систему, яка оперативно реагує на зміну стану безпеки, підвищуючи рівень захищеності громадян. Проект продемонстрував можливість створення бюджетного інформеру, загальна вартість якого склала 556,7 грн, що робить його доступним для широкого кола користувачів.

Результати роботи підтверджують, що інтерактивна мапа тривог на базі мікроконтролера ESP32 є ефективним інструментом для своєчасного інформування населення про небезпеки. Це сприяє оперативному реагуванню на надзвичайні ситуації, мінімізуючи ризики для життя та здоров'я громадян. Використання сучасних технологій у поєднанні з низькою вартістю та простотою налаштування робить цю систему перспективним рішенням для захисту населення в умовах збройного конфлікту та інших кризових ситуацій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ALARM NOTIFICATION SYSTEM AND ALARM NOTIFICATION METHOD : patent 2019004350 Japan : G08B 25/10, G05B 23/02, H04M 11/00, H04W 4/12, H04W 92/08. Applied on 16.06.2017 ; published on 10.01.2019. URL: https://patentscope.wipo.int/search/en/detail.jsf?docId=JP275176693&_cid=P12-LXFS38-06710-1 (date of access: 19.05.2024).
2. ALARM DISPLAY MAP SYSTEM AND MAP DISPLAY SYSTEM : patent 2000285338 Japan : G08B 23/00, G06F 3/00, G08B 21/00, G08B 25/00, G09B 29/00. Applied on 30.03.1999 ; published on 13.10.2000. URL: https://patentscope.wipo.int/search/en/detail.jsf?docId=JP268476390&_cid=P12-LXFSBH-11415-1 (date of access: 19.05.2024).
3. INTERACTIVE GEOSPATIAL MAP : patent 3133510 eu : G06F 17/30, G06F 3/0481, G06T 11/60, G09B 29/10. Applied on 16.08.2016 ; published on 22.02.2017. URL: https://patentscope.wipo.int/search/en/detail.jsf?docId=EP192502543&_cid=P10-LXFSAO-79100-1 (date of access: 19.05.2024).
4. Cameron N. ESP32 Microcontroller. ESP32 Formats and Communication. Berkeley, CA, 2023. P. 1–54. URL: https://doi.org/10.1007/978-1-4842-9376-8_1 (date of access: 19.05.2024).
5. Interfacing WS2812b LED Ring with ESP32 URL: <https://www.circuits-diy.com/interfacing-ws2812b-led-ring-with-esp32/> (last accessed: 20.05.2024).
6. Wootton C. Audio and Inter-IC Sound (I2S). Samsung ARTIK Reference. Berkeley, CA, 2016. P. 351–368. URL: https://doi.org/10.1007/978-1-4842-2322-2_22 (date of access: 19.05.2024).
7. Arduino Documentation URL: <https://docs.arduino.cc/> (date of access: 20.05.2024).
8. Installing the ESP32 Board in Arduino IDE (Windows, Mac OS X, Linux) URL: <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/> (date of access: 20.05.2024).

9. Ghadage S., Joshi S., Thorat S. Wi-Fi Technology. International Journal for Research in Applied Science and Engineering Technology. 2022. Vol. 10, no. 12. P. 1068–1069. URL: <https://doi.org/10.22214/ijraset.2022.48112> (date of access: 20.05.2024).
10. Breaking GSM. IEEE Spectrum. 2009. Vol. 46, no. 12. P. 13. URL: <https://doi.org/10.1109/mspec.2009.5340238> (date of access: 20.05.2024).
11. Send HTTPS request from SIM 7600x LTE Module URL: <https://dogcomp.medium.com/send-https-request-from-sim-7600x-lte-module-4f76be19e900> (date of access: 21.05.2024).
12. DF Player Mini Interface with ESP32 URL: <https://www.hackster.io/munir03125344286/df-player-mini-interface-with-esp32-f1efca> (date of access: 21.05.2024).
13. Sirens URL: https://www.researchgate.net/publication/338869206_Sirens (last accessed: 21.05.2024).
14. The Evolution of Air Threats in Future Conflicts URL: https://www.researchgate.net/publication/375975202_The_Evolution_of_Air_Threats_in_Future_Conflicts (date of access: 21.05.2024).
15. Lussier F. Future Air and Missile Threats. RAND Corporation, 2002. URL: <https://doi.org/10.7249/rb3023> (date of access: 21.06.2024).
16. Contemporary threats to the state air defence system URL: https://www.researchgate.net/publication/328217076_CONTEMPORARY_THREATS_TO_THE_STATE_AIR_DEFENCE_SYSTEM (date of access: 22.05.2024).
17. ESP32 HTTPS Requests (Arduino IDE) URL: <https://randomnerdtutorials.com/esp32-https-requests/> (date of access: 22.05.2024).
18. ESP32 HTTP GET with Arduino IDE (OpenWeatherMap.org and ThingSpeak) URL: <https://randomnerdtutorials.com/esp32-http-get-open-weather-map-thingspeak-arduino/> (date of access: 22.05.2024).
19. ESP-IDF Programming Guide URL: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html> (date of access: 23.05.2024).

20. ESP32 Tutorials URL: <https://esp32io.com/esp32-tutorials>
(date of access: 23.05.2024).
21. Volders L. ESP32 Simplified. Lulu Press, Inc., 2020.
22. AIR RAID ALERT. Miriam Hearing Sister. 2022. P. 91–92. URL:
<https://doi.org/10.2307/j.ctv2wdtm3r.65> (date of access: 24.06.2024).

ДОДАТОК А

Довідка про перевірку на унікальність пояснювальної записки

бакалаврської кваліфікаційної роботи на тему:
«Інтерактивна мапа тривоги України на базі мікроконтролера ESP32»

студента спеціальності 123 «Комп'ютерна інженерія», 405 групи
Горгун Руслан Олександрович
прізвище, ім'я, по-батькові

Перевірку тексту здійснено сервісом: онлайн-сервіс Unicheck

Результат перевірки тексту бакалаврської кваліфікаційної роботи: схожість складає 1,2 %.

UNICHECK
by Turnitin

Ім'я користувача: Борис Салтовський ID перевірки: 1016366159
Дата перевірки: 16.06.2024 22:53:18 EEST Тип перевірки: Doc vs Internet
Дата звіту: 16.06.2024 22:56:27 EEST ID користувача: 100012260

Назва документа: AP_Gorgun_diplom (1)
Кількість сторінок: 21 Кількість слів: 9137 Кількість символів: 66300 Розмір файлу: 85.25 KB ID файлу: 1016172361

1.2% Схожість
Найбільша схожість: 0.27% з Інтернет-джерелом ([https://krs.chmnu.edu.ua/jspui/bitstream/123456789/2379/1/402%20%](https://krs.chmnu.edu.ua/jspui/bitstream/123456789/2379/1/402%20%20))

1.2% Джерела з Інтернету 228 Сторінка 23

Пошук збігів з Бібліотекою не проводився

0% Цитат
Вилучення цитат вимкнено
Вилучення списку бібліографічних посилань вимкнено

0% Вилучень
Немає вилучених джерел

Здобувач:


підпис

Р. О. Горгун
ініціали, прізвище

Керівник:

Ст. викл. каф. комп. інженерії

_____ Б. Г. Салтовський
підпис ініціали, прізвище

Дата: «__» _____ 2024 р.

2024 р

Горгун Р. О.

123 – КБР.01 – 405.22010603

ДОДАТОК Б

Лістинг коду для інтерактивної мапи тривоги на базі esp32

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <FastLED.h>
#include <Adafruit_I2S.h>

// Wi-Fi налаштування
const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";

// Адреса API
const char* apiUrl = "http://map.ukrainealarm.com/api/alarms";
const char* apiKey = "YOUR_TOKEN";

// Налаштування для світлодіодів
#define LED_PIN 5
#define NUM_LEDS 25
CRGB leds[NUM_LEDS];

// Ініціалізація звукового модулю
Adafruit_I2S i2s;

// Максимальна довжина відповіді від API
const int capacity = JSON_ARRAY_SIZE(NUM_LEDS) +
JSON_OBJECT_SIZE(1);

// Інтервал оновлення даних (у мілісекундах)
```

```
const long interval = 60000; // 60 секунд
unsigned long previousMillis = 0;

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

  // Ініціалізація світлодіодів
  FastLED.addLeds<NEOPIXEL, LED_PIN>(leds, NUM_LEDS);

  // Підключення до Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  // Ініціалізація звукового модулю
  if (!I2S.begin()) {
    Serial.println("Failed to initialize I2S");
    while (1);
  }
}

void loop() {
  unsigned long currentMillis = millis();

  // Перевірка, чи настав час для оновлення даних
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
```

```
updateAlarms();
}
}

void updateAlarms() {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(apiUrl);
        int httpResponseCode = http.GET();

        if (httpResponseCode > 0) {
            String response = http.getString();
            DynamicJsonDocument doc(capacity);
            deserializeJson(doc, response);

            // Парсинг JSON даних
            JsonArray alarms = doc["alarms"];
            for (int i = 0; i < NUM_LEDS; i++) {
                bool alarmActive = alarms[i]["alarm_active"];
                if (alarmActive) {
                    leds[i] = CRGB::Red;
                } else {
                    leds[i] = CRGB::Green;
                }
            }
            FastLED.show();

            // Звукове сповіщення для Миколаївської області
            if (i == 15 && alarmActive) { // Припускаємо, що 15-ий індекс
                // відповідає Миколаївській області
                playSiren();
            }
        }
    }
}
```

```
    }  
  }  
  } else {  
    Serial.print("Error on HTTP request: ");  
    Serial.println(httpResponseCode);  
  }  
  http.end();  
} else {  
  Serial.println("WiFi not connected");  
  WiFi.reconnect();  
}  
}  
  
void playSiren() {  
  // Звуковий файл повинен бути заздалегідь збережений у форматі WAV  
у пам'яті ESP32  
  // В цьому прикладі ми просто генеруємо простий тон (бо i2s.begin() не  
підтримує play функції)  
  
  const int frequency = 1000; // Частота тону  
  const int duration = 5000; // Тривалість у мілісекундах  
  const int sampleRate = 44100; // Частота дискретизації  
  
  for (int i = 0; i < (duration * sampleRate) / 1000; i++) {  
    int16_t sample = 32767 * sin(2 * PI * frequency * i / sampleRate);  
    i2s.write(sample);  
  }  
}
```