

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Обслуговування комп'ютерних систем та мереж»

Група: 4КС-57

Дипломний проєкт

здобувача освіти денної форми навчання

КС.57.03.000.ДП

**ГЕРГАНОВА ВЛАДИСЛАВА
ПИЛИПОВИЧА**

м. Одеса
2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Обслуговування комп'ютерних систем та мереж»

Група: 4КС-57

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

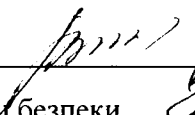
**Розробка фізичної карти
повітряних тривог**

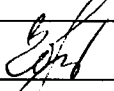
Проектний матеріал складається з пояснювальної записки на 81 сторінках та графічного (презентаційного) матеріалу на 10 аркушах (слайдах).

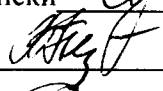
Дипломник  (Герганов В. П.)

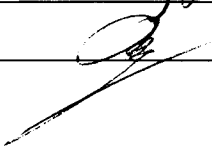
Керівник  (Кіреєв І. А.)

Консультанти:

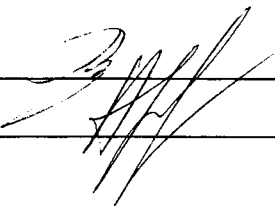
з економічного розділу  (Іванченков В. С.)

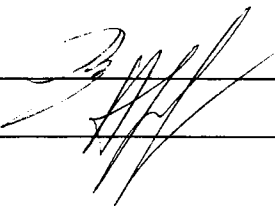
з розділу охорони праці та техніки безпеки  (Чорновол Н. І.)

з нормоконтролю  (Петрашова В. І.)

старший консультант  (Кривченко Ю. В.)

До захисту допущений

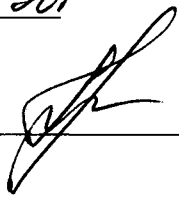
Голова циклової комісії  (Кривченко Ю. В.)

Завідувач відділення  (Скорнякова О. В.)

Захист «19» 06 2024 р.

Протокол ЕК № 3

Оцінка ЕК 5 (вдуже добре) 90%

Секретар ЕК 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення	<u>комп'ютерних систем</u>	Комісія	<u>КТ та ПП</u>
Спеціальність	<u>123 «Комп'ютерна інженерія»</u>		
Освітня програма	<u>«Обслуговування комп'ютерних систем та мереж»</u>		

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР

Беркань І. В.

“ 15 ” 01 2024 року

ЗАВДАННЯ

на дипломний проєкт

Здобувачеві освіти Герганову Владиславу Пилиповичу

1. Тема проєкту Розробка фізичної карти повітряних тривог

Затверджена наказом по коледжу від “ 02 ” листопада 2023 р., наказ № 244-А2-ОД

2 Термін здачі закінченого проєкту 10.06.2024

3. Вихідні дані до проєкту

1. Програмована плата на базі ESP32. 2. Адресна світлодіодна стрічка WS2812B 5V.

2. Тактова кнопка бхбх12. 4. Контролер заряду Li-Ion акумуляторів на базі TP4056 з Type-C.

5. Підвищуючий DC/DC перетворювач на MT3608. 6. Пасивний зумер. 7. Перемикач KCD5-102

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

1. Аналіз предметної області. 2. Технології та засоби розробки (проєктування).

3. Компонентна база. 4. Проєктування пристрою. 5. Проєктування прошивки.

6. Збірка пристрою. 7. Розробка прошивки. 8. Тестування створеного пристрою.

9. Економічний розрахунок. 10. Аспекти охорона праці та техніки безпеки

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

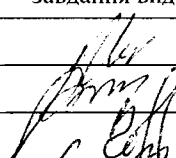
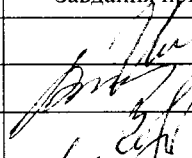
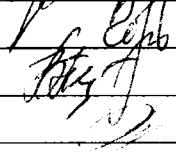
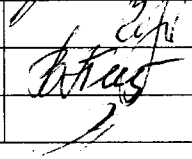
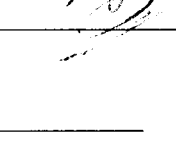
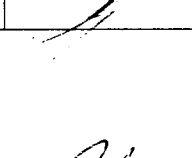
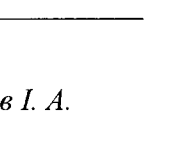
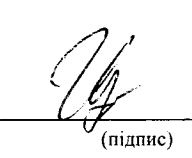
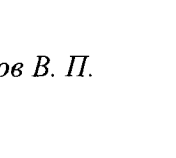
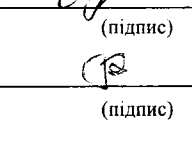
Презентація Power Point – 10 слайдів

!Основні відомості про проєкт; Вибір елементної бази; Вибір мікроконтролера;

Структурна схема карти повітряних тривог; Схема підключення компонентів до ESP32;

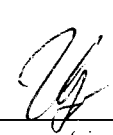
Підключення до Wi-Fi через вебсервер; Тестування пристрою)

6. Консультанти по проєкту, із зазначенням розділів проєкту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Кірсєв І. А.		
Економічний розділ	Іванченков В. С.		
Розділ охорони праці	Чорновол Н. І.		
Нормоконтроль	Петрашова В. І.		
Старший консультант	Кривченко Ю. В.		

7. Дата видачі завдання 15.01.2024

Керівник Кірсєв І. А.


(підпис)

Завдання прийняв до виконання Герганов В. П.


(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проєкту (роботи)	Термін виконання етапів дипломного проєкту (роботи)	Відмітка про виконання
1	Формування вступу	29.04.24	Виконано
2	Аналіз предметної області	10.05.24	Виконано
3	Підбір технічної літератури	19.05.24	Виконано
4	Вибір технологій та засобів проєктування	20.05.24	Виконано
5	Вибір компонентної бази	22.05.24	Виконано
6	Проєктування архітектури пристрою	24.05.24	Виконано
7	Реалізація моделі пристрою	26.05.24	Виконано
8	Розробка прошивки	28.05.24	Виконано
9	Тестування створеного пристрою	29.05.24	Виконано
10	Оформлення пояснювальної записки	31.05.24	Виконано
11	Оформлення графічної (презентаційної) частини	01.06.24	Виконано
12	Економічний розрахунок	02.06.24	Виконано
13	Опис охорони праці та техніки безпеки	09.06.24	Виконано
14	Аналіз результатів проєктування	13.06.24	Виконано
15	Підготовка доповіді для захисту	16.06.24	Виконано

Дипломник 
(підпис)

Керівник 
(підпис)

[illegible]

ЗМІСТ

ВСТУП	7
1 ОСНОВНИЙ РОЗДІЛ	8
1.1 Аналіз предметної області	8
1.1.1 Огляд існуючих рішень	8
1.1.2 Технології та засоби проєктування	11
1.1.3 Вибір комплектуючих	26
1.2 Проєктування моделі пристрою	32
1.2.1 Технічне завдання на розробку	32
1.2.2 Проєктування моделі пристрою	32
1.3 Реалізація пристрою	34
1.3.1 Підготовка та налаштування середовища розробки Thonny IDE	34
1.3.2 Збір пристрою	35
1.3.3 Розробка прошивки	42
1.4 Тестування пристрою	52
2 ЕКОНОМІЧНИЙ РОЗДІЛ	53
2.1 Резюме	53
2.2 Вартісна оцінка розробленого пристрою	53
2.3 Прогноз обсягів продаж даного виробу	56
3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	60
3.1 Вступ	60
3.2 Аналіз небезпечних та шкідливих чинників, що впливають на працівника	60
3.3 Розробка заходів з охорони праці	61
3.3.1 Гігієнічне нормування параметрів повітря робочої зони	62
3.3.2 Освітлення	63
3.3.3 Шум	63
3.3.4 Електробезпека	64
3.3.5 Організація робочого місця	65
3.4 Пожежна безпека	65

ВИСНОВКИ	68
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	69
ДОДАТОК А. Програмний код основної логіки веб-застосунку	71
ДОДАТОК Б. Слайди мультимедійної презентації	76

					КС 57. 03 000. 00 ДП ПЗ	Арк.
						6
Ізм.	Лист	№ докум.	Підпис	Дата		

ВСТУП

Розробка фізичної карти повітряних тривог в Україні є актуальною та важливою, оскільки вона забезпечує оперативне інформування населення про ракетну небезпеку, що є життєво важливим для забезпечення безпеки громадян у воєнний час. Ця система дозволяє миттєво сприймати інформацію про тривогу через візуальне та звукове сповіщення, що може врятувати життя. Водночас, така карта може слугувати елементом дизайну, інтегруючись у простір та стаючи частиною інтер'єру, що підкреслює соціальну відповідальність та освітній аспект, нагадуючи про важливість об'єднання спільноти у важкі часи. Така карта є не лише інструментом безпеки, але й символом стійкості та єдності нації.

Метою даної роботи є розробка фізичної карти повітряних тривог, яка дозволить оперативно інформувати населення про ракетну небезпеку в різних регіонах України. Основними завданнями проєкту є створення інтуїтивно зрозумілої візуалізації зон небезпеки, інтеграція звукового сигналу для підвищення уваги до тривоги. Крім того, проєкт передбачає можливість використання карти як елемента дизайну, що додасть їй додаткову цінність та функціональність. Важливою частиною роботи також є забезпечення простоти використання системи для всіх категорій населення.

При дослідженні основних проблем предметної області, аналізі аналогів та засобів розробки використано системний підхід. Для реалізації фізичної моделі пристрою використано мікроконтролер ESP-WROOM-32 на базі ESP32.

Результатом розробки є фізична карта повітряних тривог, яка служить як ефективний засіб для оперативного сповіщення населення про повітряні тривоги.

					КС 57. 03 000. 00 ДП ПЗ	Арк.
						7
Ізм.	Лист	№ докум.	Підпис	Дата		

1 ОСНОВНИЙ РОЗДІЛ

1.1 Аналіз предметної області

1.1.1 Огляд існуючих рішень

Розглянемо існуючі альтернативи або аналогічні пристрої, схожі на той, який розробляється:

Веб-карта тривоги, яка доступна через браузер на мобільних телефонах та комп'ютерах.

Мобільний застосунок «Повітряна тривога» для сповіщення про тривоги, для смартфонів, планшетів і т.д.

Звуковий оповіщувач на базі мікроконтролера ESP8266, який генерує звукові сигнали після початку тривоги.

Розпочнемо з огляду першого альтернативного рішення - веб-сервісу для відстеження тривог «alerts.in.ua».

1 Аналог – Веб-карта тривоги «alerts.in.ua» – це інтерактивна веб-карта тривоги, яка дозволяє користувачам візуалізувати та відстежувати стан повітряних тривог в реальному часі через браузер на мобільному телефоні чи комп'ютері [1].

На рис. 1.1 зображено веб-карту.

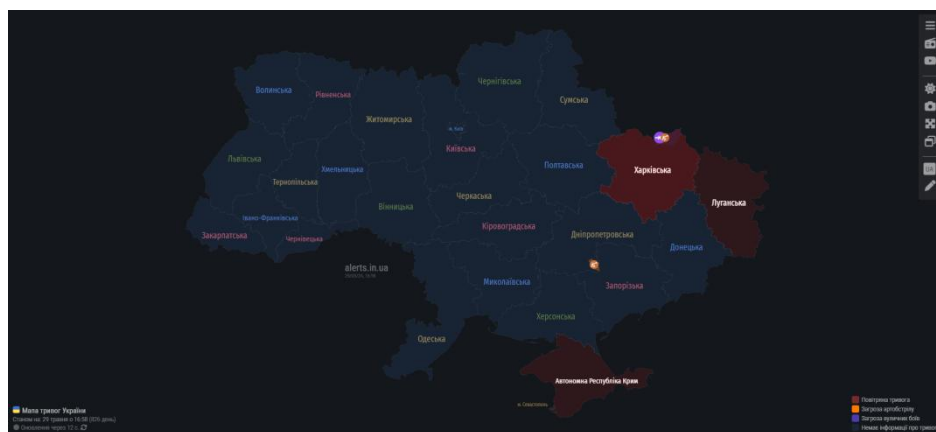


Рисунок 1.1. Веб-карта повітряних тривог

Розглянемо переваги веб-карти:

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						8
Ізм.	Лист	№ докум.	Підпис	Дата		

1. Доступність: Карта доступна з будь-якого пристрою, що має інтернет-з'єднання
2. Деталізація: Дозволяє відображати інформацію не лише про тривоги в цілому по області, а й окремо для конкретних населених пунктів.
3. Інформативність: Користувачі можуть отримати більшу інформацію про тривогу (тривалість тривоги, час тривоги, поточні загрози)

Недоліки веб-карти:

1. Оповіщення: Не подає звукових сигналів при оголошенні повітряної тривоги.

Розглянемо другий аналог – мобільний застосунок «Повітряна тривога».

2 Аналог – це мобільний застосунок, розроблений для надання миттєвих сповіщень про повітряну тривогу в обраному користувачем місті чи регіоні, має вбудовану карту повітряних тривог в застосунок [2].

На рис. 1.2 зображено мобільний застосунок «Повітряна тривога».

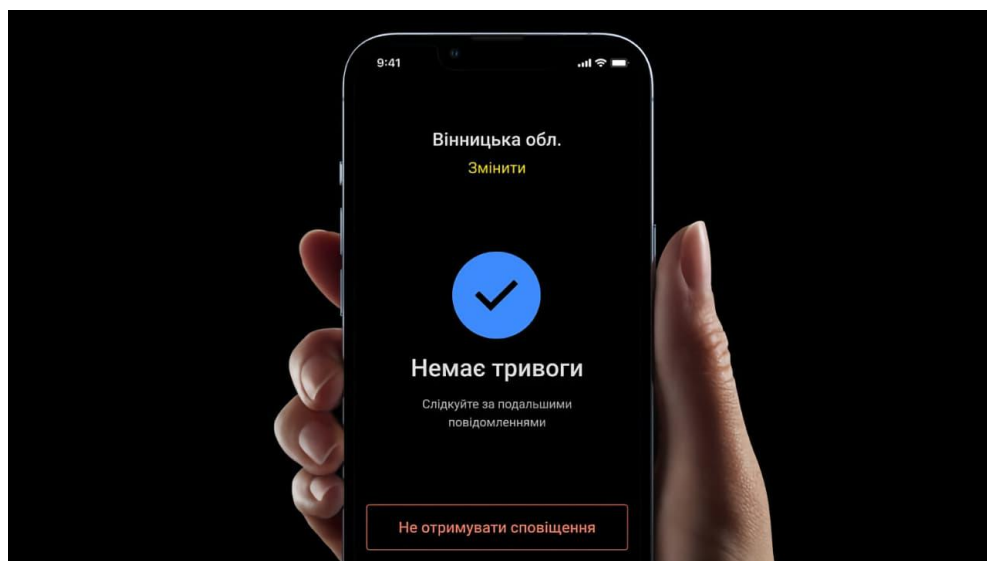


Рисунок 1.2. Мобільний застосунок «Повітряна тривога»

Переваги мобільного застосунку «Повітряна тривога»:

1. Сповіщення в реальному часі: Застосунок надсилає сповіщення про повітряну тривогу, як тільки вона оголошується
2. Робота в беззвучному режимі: Навіть якщо смартфон перебуває в беззвучному режимі, застосунок може гучно сповістити про тривогу.

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						9
Ізм.	Лист	№ докум.	Підпис	Дата		

3. Вбудована карта: Мобільний застосунок «Повітряна тривога» включає вбудовану карту тривоги, яка дозволяє користувачам візуально відстежувати повітряні тривоги по всій Україні

Третій аналог пристрою – «Персональний оповіщувач про повітряні тривоги».

3 Аналог – це проєкт «Персональний оповіщувач про повітряні тривоги» розроблений автором каналу «Pavel Fomuchov» на платформі «YouTube». Коли починається тривога в заданій в скетчі області, зумер(п'єзовипромінювач) видаватиме звук, схожий на звук сирени [3].

На рис. 1.3 зображено пристрій.

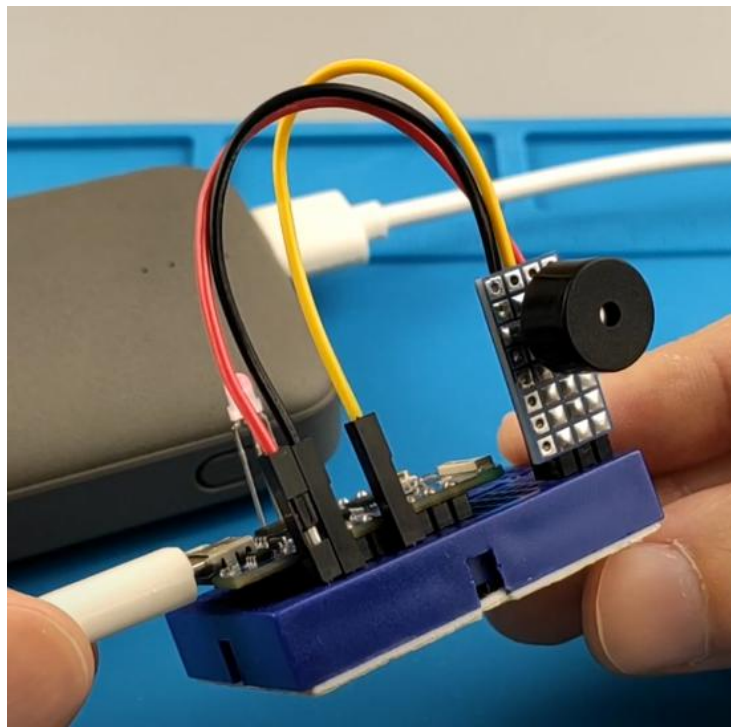


Рисунок 1.3. Персональний оповіщувач про повітряні тривоги

Переваги:

1. Енергоефективність: Споживає мало енергії

Розглянемо недоліки:

1. Інформування: Система не забезпечує повної карти тривоги по всій Україні, а лише інформує про стан тривоги в обраній області

2. Налаштування пристрою: Для підключення до Wi-Fi, та щоб обрати свою область потрібно самому змінювати код прошивки.

Проаналізуємо порівняльну таблицю аналогів.

Порівняльна таблиця аналогів представлена на таблиці 1.1.

Таблиця 1.1. Порівняльна характеристика аналогів

Характеристика	Веб-карта	Мобільний застосунок	Персональний оповіщувач про повітряні тривоги
Наявність звукового оповіщення	ні	так	так
Наявність повної карти тривог	так	так	ні
Використання як елемент дизайну	ні	ні	ні

Враховуючи аналіз існуючих рішень, можна зробити висновок, що розробка фізичної карти повітряних тривог, яка розглядається в цій роботі, є доцільною. Цей проєкт пропонує інтуїтивно зрозумілу та ефективну звукову, та графічну систему оповіщення, яка також може слугувати як елемент дизайну, що додає естетичну цінність.

1.1.2 Технології та засоби проєктування

Для створення фізичної карти повітряних тривог можна використовувати різні технології та платформи. Вони мають свої переваги та недоліки. Розглянемо три основні технології, які можна використати для створення такого проєкту: Arduino, Raspberry Pi, ESP32.

Технологія 1 – Arduino.

Arduino – це ефективний засіб розробки програмованих електронних пристроїв, які, на відміну від персональних комп'ютерів, орієнтовані на тісну взаємодію з навколишнім світом. Arduino являє собою відкриту програмовану

апаратну платформу для роботи з різноманітними фізичними об'єктами і складається з простої плати з мікроконтролером, а також спеціального середовища розробки для написання програмного забезпечення мікроконтролера.

Arduino можна використовувати для створення інтерактивних систем, керованих різними датчиками та перемикачами. Такі системи, у свою чергу, можуть керувати роботою різних індикаторів, двигунів та інших пристроїв. Проєкти Arduino можуть бути як самостійними, так і взаємодіяти з програмним забезпеченням, що працює на персональному комп'ютері (наприклад, з програмами Flash, Processing, MaxMSP). Будь-яку плату Arduino можна зібрати власноруч або придбати готовий пристрій; середовище розробки для програмування такої плати має відкритий вихідний код і є повністю безкоштовним.

Мова програмування Arduino є реалізацією схожої апаратної платформи "Wiring", заснованої на середовищі програмування мультимедіа "Processing" [4].

Arduino включає в себе різні моделі плат, серед яких найбільш популярні: Arduino Uno, Arduino Mega, Arduino Nano, Arduino Leonardo та інші. Кожна з них має свої особливості і може використовуватися для різних типів проєктів. Arduino Uno є найпопулярнішою моделлю завдяки своїй універсальності та простоті у використанні. Вона має 14 цифрових входів/виходів, 6 аналогових входів, 16 МГц кварцовий осцилятор, USB-роз'єм для програмування та живлення, а також роз'єм для зовнішнього живлення.

Основні характеристики плати Arduino Uno:

- Мікроконтролер: ATmega328P (8-бітове AVR ядро);
- Робоча напруга: 5V;
- Вхідна напруга, що рекомендується: 7-12V;
- Аналогові входи: 6 (A0 – A5);
- Цифрові входи/виходи: 14 (з яких 6 можуть використовуватися як виходи ШІМ);
- Струм по цифровим входам/виходам: 40 мА;
- Струм по 3.3V входу: 50 мА;

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						12
Ізм.	Лист	№ докум.	Підпис	Дата		

- Flash: 32 КБ (з яких 0.5 КБ використовується завантажувачем);
- SRAM: 2 КБ;
- EEPROM: 1 КБ;
- Частота (тактова частота): 16 МГц.

Arduino Uno також має різні інтерфейси, такі як USB-підключення, роз'єм живлення, заголовок ICSP та кнопку скидання¹. Він містить все необхідне підтримки мікроконтролера; просто підключіть його до комп'ютера за допомогою кабелю USB або запитуйте від адаптера AC-to-DC або батареї, щоб почати роботу [5].

На рис. 1.4 зображено мікроконтролер Arduino Uno.



Рисунок 1.4. Arduino Uno

Переваги Arduino:

1. Легкість у використанні для новачків.
2. Велика кількість навчальних матеріалів та прикладів.
3. Широкий спектр датчиків і модулів, що підтримуються.
4. Відкритий вихідний код та активна спільнота.

Недоліки:

1. Обмежені обчислювальні можливості порівняно з іншими платформами.
2. Обмежений обсяг пам'яті та швидкість обробки даних.
3. Можливі труднощі при реалізації складних проектів з великим обсягом обробки даних.

Технологія 2 – Raspberry Pi.

Raspberry Pi – це мініатюрний одноплатний комп'ютер, який легко поміститься на долоні дорослої людини. Незважаючи на свої скромні розміри, плата має високу продуктивність, що дозволяє їй вийти на один рівень зі стаціонарними ПК. Цей компактний пристрій став справжньою революцією у світі технологій, поєднуючи в собі потужність, портативність і доступність.

Спочатку Raspberry Pi був розроблений як навчальний посібник з інформатики. Ідея полягала в тому, щоб надати студентам та ентузіастам недорогий інструмент для вивчення основ програмування, комп'ютерних наук та електроніки. Але сама концепція виявилася настільки вдалою, що за кілька років міні-комп'ютер став популярним у дуже широких колах. Його використовують не лише в освітніх закладах, а й у промисловості, для створення розумних будинків, у якості медіа-центрів, ретро-ігрових консолей та навіть у наукових експериментах.

З плином часу Raspberry Pi пережив кілька модифікацій, кожна з яких відрізнялася від попередника якимось параметром. Наприклад, модель Raspberry Pi 4 має більш потужний процесор, більше оперативної пам'яті та підтримує два дисплеї з роздільною здатністю 4K. А модель Raspberry Pi Zero – це ультракомпактна та дешева версія, ідеальна для вбудованих проектів. Такий підхід дозволив регулювати вартість виробу залежно від потреб користувача, що також позитивно позначилося на популярності пристрою. Кожен може вибрати модель, яка найкраще відповідає його завданням і бюджету.

Вся лінійка Raspberry Pi використовує процесори з ARM-архітектурою, яка зарекомендувала себе з найкращого боку. ARM-процесори відомі своєю енергоефективністю та хорошим співвідношенням продуктивності до споживаної енергії. Це робить Raspberry Pi ідеальним для проектів, де важливі компактність і низьке енергоспоживання, наприклад, для робототехніки або портативних пристроїв.

Однією з найпопулярніших моделей є Raspberry Pi 3 B+, зовнішній вигляд якої показано на рис. 1.5. Ця модель має чотириядерний процесор Broadcom

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						14
Ізм.	Лист	№ докум.	Підпис	Дата		

BCM2837B0 з тактовою частотою 1,4 ГГц, 1 ГБ оперативної пам'яті, вбудований Wi-Fi і Bluetooth, а також повнорозмірний порт HDMI і чотири USB-порти. Незважаючи на те, що зараз доступні новіші моделі, Raspberry Pi 3 B+ все ще широко використовується через свою надійність, достатню потужність для багатьох завдань і більш низьку ціну порівняно з новішими версіями.



Рисунок 1.5. Raspberry Pi 3 B+

Спільнота Raspberry Pi є однією з найактивніших і найбільш сприятливих у світі технологій. Користувачі діляться своїми проектами, надають навчальні посібники та допомагають один одному вирішувати проблеми. Офіційний сайт та форуми Raspberry Pi Foundation пропонують безліч ресурсів, від базових уроків до складних проектів. Крім того, існує величезна кількість додаткових компонентів і аксесуарів, таких як камери, сенсори, дисплеї та корпуси, що дозволяють адаптувати Raspberry Pi під будь-які потреби.

Raspberry Pi також відіграє важливу роль у просуванні ідеї відкритого програмного та апаратного забезпечення. Більшість програмного забезпечення для Raspberry Pi є відкритим, а схеми самої плати доступні для перегляду. Це не тільки дозволяє спільноті покращувати і розширювати можливості пристрою, але й навчає принципам прозорості та співпраці в технологічному світі.

У майбутньому очікується, що Raspberry Pi продовжить свій розвиток, пропонуючи ще більшу продуктивність, нові функції та, можливо, навіть спеціалізовані версії для конкретних завдань, таких як машинне навчання або обробка відео. При цьому фонд Raspberry Pi Foundation залишається вірним

своїй основній місії - зробити обчислювальну техніку та цифрову творчість доступними для всіх [6].

Повні характеристики Raspberry Pi 3 Model B+:

Процесор:

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC;
- Чотириядерний, 1,4 ГГц (на відміну від 1,2 ГГц у Raspberry Pi 3B);
- Покращена схема управління живленням та тепловіддачею для стабільної роботи на вищих частотах.

Оперативна пам'ять:

- 1 ГБ LPDDR2 SDRAM;
- Працює на частоті 900 МГц.
- Графічний процесор:
- Broadcom VideoCore IV;
- Підтримує OpenGL ES 2.0, 1080p30 H.264/MPEG-4 AVC високопрофільне декодування;
- Здатний забезпечувати відкриту графіку GL ES 2.0, апаратне прискорення OpenVG 1.1, 1080p30 H.264.

Мережеві можливості:

- 2.4ГГц і 5ГГц IEEE 802.11.b/g/n/ac бездротовий LAN (оновлено до ас у порівнянні з Raspberry Pi 3B);
- Bluetooth 4.2, BLE (Bluetooth Low Energy);
- Гігабітний Ethernet через USB 2.0 (максимальна пропускна здатність до 300 Мбіт/с);

USB-порти:

- 4 x USB 2.0 порти.

GPIO (General Purpose Input/Output):

- 40-контактний роз'єм GPIO;
- З них: 27 програмованих GPIO, різні інтерфейси (I²C, SPI, UART);
- Можливість живлення 5В, 3.3В, підключення до землі.

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						16
Ізм.	Лист	№ докум.	Підпис	Дата		

Мультимедіа:

- Повнорозмірний HDMI порт (підтримка до 1080p60);
- 4-полюсний стерео аудіо та композитний відеопорт;
- Інтерфейс CSI для підключення камери Raspberry Pi;
- Інтерфейс DSI для підключення сенсорного дисплею Raspberry Pi.

Живлення:

- 5В постійного струму через роз'єм micro USB або GPIO;
- Рекомендована потужність: мінімум 2,5А для повного навантаження;
- Підтримка Power over Ethernet (PoE) за допомогою окремо придбаного PoE HAT.

Зберігання даних:

- Слот для карт пам'яті microSD;
- Підтримка завантаження з USB-накопичувачів.
- Розміри та вага:
- 88 x 58 x 19,5 мм;
- Приблизно 50 г.

Операційна система:

- Офіційно підтримується Raspbian (на основі Debian);
- Підтримка інших дистрибутивів: Ubuntu MATE, Snappy Ubuntu Core, Windows 10 IoT Core, LibreELEC, OSMC, RISC OS.

Додаткові функції:

- Роз'єм для камери Raspberry Pi (CSI);
- Роз'єм для дисплея Raspberry Pi (DSI);
- Апаратне декодування H.264, MPEG-4;
- Апаратне кодування H.264 (до 1080p30).

Переваги:

1. Висока обчислювальна потужність: Завдяки потужному процесору та великому обсягу оперативної пам'яті, Raspberry Pi може виконувати складні завдання, включаючи обробку зображень, роботу з базами даних та запуск веб-серверів.

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						17
Ізм.	Лист	№ докум.	Підпис	Дата		

2. Можливість запуску операційних систем: Raspberry Pi підтримує різні операційні системи, такі як Raspbian (офіційна ОС на базі Linux), Ubuntu, Windows 10 IoT Core та інші. Це робить його дуже гнучким для різних типів проектів.
3. Велика кількість портів для підключення периферійних пристроїв: Raspberry Pi має багато портів для підключення різноманітних пристроїв, включаючи USB, HDMI, Ethernet та GPIO.
4. Широкий спектр застосувань: Raspberry Pi можна використовувати для створення медіацентрів, домашніх серверів, систем автоматизації, IoT-проектів та багато іншого.

Недоліки:

1. Вища складність у використанні для новачків: Raspberry Pi потребує більше знань для налаштування та конфігурації операційних систем і програмного забезпечення, що може бути складним для новачків.
2. Вища вартість у порівнянні з Arduino: Raspberry Pi коштує дорожче за Arduino, що може бути важливим фактором при виборі платформи для простих проектів.
3. Потреба додаткових знань для налаштування: Для повного використання можливостей Raspberry Pi необхідні знання в області Linux, мережевих з'єднань та програмування.

Технологія 3 – ESP32.

ESP32 – це потужна, недорога і енергоефективна система на чипі (SoC), яка стала справжнім проривом у світі Інтернету речей (IoT) та вбудованих систем. Розроблена компанією Espressif Systems, ця плата пропонує унікальне поєднання продуктивності, гнучкості та низького енергоспоживання, що робить її ідеальною для широкого спектру застосувань.

Плата ESP32 базується на мікропроцесорі Xtensa LX6, який може працювати в одно- або двоядерній конфігурації з тактовою частотою до 240 МГц [7]. Це значне поліпшення в порівнянні з її попередником, ESP8266, який мав лише одне ядро з частотою 80 МГц. Висока обчислювальна потужність ESP32

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						18
Ізм.	Лист	№ докум.	Підпис	Дата		

дозволяє їй обробляти складні алгоритми в режимі реального часу, наприклад, для машинного навчання, обробки голосу або керування роботами.

Однією з найвизначніших особливостей ESP32 є її вражаюча кількість вбудованих периферійних пристроїв. Вона оснащена 34 програмованими GPIO (General Purpose Input/Output) контактами, які підтримують різні протоколи: I²C, SPI, I²S, UART. Це дозволяє підключати широкий спектр датчиків, дисплеїв і актуаторів без додаткових мікросхем. ESP32 також має 12-бітові АЦП (аналогово-цифрові перетворювачі) і 8-бітові ЦАП (цифро-аналогові перетворювачі), що робить її відмінним вибором для проектів, які вимагають точного вимірювання або генерації аналогових сигналів [8].

В епоху IoT, з'єднання відіграють вирішальну роль, і ESP32 тут виступає яскравим прикладом. Цей чіп підтримує Wi-Fi (стандарти 802.11 b/g/n) на частоті 2,4 ГГц з максимальною швидкістю передачі даних до 150 Мбіт/с, а також класичний Bluetooth і Bluetooth Low Energy (BLE). Завдяки цьому, ESP32 може безпосередньо взаємодіяти зі смартфонами, ноутбуками та іншими IoT-пристроями без потреби у додаткових компонентах. Окрім того, деякі моделі ESP32, наприклад ESP32-S3, також підтримують стандарт IEEE 802.15.4, який застосовується у протоколах Zigbee та Thread, розширюючи можливості для підключення.

ESP32 вражає своєю енергоефективністю. У режимі глибокого сну споживання становить лише 2,5 мкА, роблячи чіп ідеальним для батарейних пристроїв. Чіп має декілька режимів енергозбереження та може динамічно змінювати частоту процесора та вимикати неактивні периферійні пристрої для оптимізації тривалості роботи від батареї.

Щодо пам'яті та зберігання, ESP32 зазвичай оснащений 4 МБ вбудованої флеш-пам'яті, хоча деякі моделі можуть мати до 16 МБ. Це дозволяє зберігати об'ємні програми, конфігураційні дані та навіть файли веб-інтерфейсу безпосередньо на пристрої. Крім того, ESP32 має 520 КБ SRAM для програмних змінних та буферів, а моделі, такі як ESP32-S3, мають додаткові 8 МБ PSRAM для більшої гнучкості [7].

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						19
Ізм.	Лист	№ докум.	Підпис	Дата		

Програмування ESP32 є інтуїтивним та адаптивним. Офіційне середовище розробки ESP-IDF (Espressif IoT Development Framework) базується на FreeRTOS, що дозволяє виконувати багатозадачність у реальному часі, що є критично важливим для IoT-застосувань. Для тих, хто шукає більш високий рівень абстракції, ESP32 також підтримується в Arduino IDE, що робить її доступною для новачків та швидкого створення прототипів. Крім того, ESP32 сумісна з MicroPython та CircuitPython, дозволяючи програмувати на високорівневій мові Python.

Застосування ESP32 є дуже різноманітними. Цей чіп є популярним вибором для створення розумних домашніх систем, таких як термостати, системи безпеки та контролери освітлення. У промисловості він використовується для моніторингу обладнання, автоматизації процесів та управління енергоспоживанням. У медичній сфері ESP32 застосовується у носимих пристроях для моніторингу життєво важливих показників та системах віддаленого спостереження за пацієнтами. Також чіп знайшов застосування у цифровому мистецтві, живлячи інтерактивні інсталяції та світлодіодні дисплеї.

Спільнота навколо ESP32 є великою та активною. Espressif підтримує форуми, де розробники обмінюються проектами, ставлять запитання та розв'язують проблеми. Існує багато онлайн-курсів, які охоплюють від базових до складних тем, включаючи машинне навчання на ESP32. Популярні платформи електроніки, такі як Adafruit та SparkFun, пропонують широкий вибір плат ESP32 та сумісних модулів.

Підсумовуючи, ESP32 не просто черговий мікроконтролер; це потужна, енергоефективна та надзвичайно гнучка платформа, яка змінила підходи в світі IoT та вбудованих систем. Від розумних будинків до промислового Інтернету речей, від носимих пристроїв до інтерактивного мистецтва – можливості ESP32 обмежені лише уявою розробників. Ця платформа відкриває безмежні горизонти для інновацій та творчості, дозволяючи створювати рішення, які раніше здавалися неможливими. Завдяки своїй гнучкості, ESP32 є ідеальним вибором

для широкого спектру застосувань, від простих DIY проектів до складних промислових систем.

На рис. 1.6 зображено мікроконтролер ESP32.



Рисунок 1.6. «Мікроконтролер ESP32»

Повні характеристики:

Процесор:

- Двоядерний або одноядерний Xtensa® LX6 32-бітний мікропроцесор;
- Тактова частота до 240 МГц;
- Продуктивність: до 600 DMIPS (Millions of Instructions Per Second - мільйонів інструкцій в секунду);
- Ультранизьке енергоспоживання (ULP) співпроцесор.

Пам'ять:

- 448 КБ вбудованої ROM (Read-Only Memory);
- 520 КБ вбудованої SRAM (Static Random Access Memory);
- До 16 МБ зовнішньої флеш-пам'яті;
- До 8 МБ зовнішньої PSRAM (на ESP32-S3).

Бездротовий зв'язок:

- Wi-Fi: 802.11 b/g/n, 2,4 ГГц;
- Швидкість передачі даних по Wi-Fi: до 150 Мбіт/с;
- Bluetooth® v4.2 BR/EDR та Bluetooth Low Energy (BLE);
- Вбудовані антени або роз'єм для зовнішньої антени.

Периферійні інтерфейси:

- 34 програмованих контакти GPIO;
- 4 канали SPI;
- канали I²C;
- канали I²S;
- канали UART ;
- 12 ємнісних сенсорних каналів;
- 2 канали ЦАП (8-біт);
- 18 каналів АЦП (12-біт);
- 2 контролери CAN 2.0;
- Інтерфейс камери (8-біт паралельний).

Інші функції:

- Інтерфейс SD/SDIO/MMC Host;
- Пульсово-широтна модуляція (PWM) / Контролер двигуна;
- Інфрачервоний віддалений контролер (до 8 каналів);
- Таймери: 4 x 64-біт генеральні таймери, 2 x ШІМ таймери, 1 x системний таймер.

Енергоспоживання:

- 5 В живлення через USB або VIN контакт;
- 3,3 В живлення через 3V3 контакт;
- Струм в активному режимі: ~260 мА (при 3,3 В, 240 МГц);
- Струм в режимі глибокого сну: ~2,5 мкА.

Розміри (варіюються залежно від версії модуля):

- ESP32-WROOM-32: 25,5 x 18 x 3,1 мм;
- ESP32-PICO-D4: 7 x 7 мм (чип).

Програмування:

- Через USB (з використанням UART-USB мосту);
- Через JTAG для розширеного налагодження;
- Оновлення програмного забезпечення через Wi-Fi (OTA - Over-The-Air).

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						22
Ізм.	Лист	№ докум.	Підпис	Дата		

Розробка програмного забезпечення:

- ESP-IDF (Espressif IoT Development Framework) - офіційна середовища розробки;
- Підтримка Arduino IDE;
- Підтримка MicroPython та CircuitPython;
- OpenThread підтримка на ESP32-C6 [7].

Переваги ESP32:

1. Ціна: Низька вартість робить ESP32 доступним для різноманітних проектів.
2. Розмір: Компактний розмір дозволяє використовувати ESP32 в мініатюрних пристроях.
3. Вбудований WiFi та Bluetooth: Вбудовані модулі WiFi та Bluetooth спрощують створення бездротових рішень.
4. Енергозбереження: ESP32 має кілька режимів енергозбереження, що дозволяє знижувати споживання енергії, особливо важливо для автономних пристроїв.
5. Інтеграція з середовищем розробки: Підтримка Arduino IDE, PlatformIO та інших середовищ спрощує програмування та налагодження.
6. Багатофункціональні інтерфейси: Підтримка численних периферійних інтерфейсів, таких як UART, SPI, I2C, PWM, ADC, DAC, та сенсорний інтерфейс.

Недоліки ESP32:

1. Обмежена потужність процесора: Двоядерний процесор менш потужний, ніж в Raspberry Pi що обмежує можливості для складних обчислювальних завдань.
2. Відсутність повноцінної операційної системи: Працює без повноцінної операційної системи, що може бути обмежуючим фактором для деяких проектів.

Проаналізуємо порівняльну таблицю технологій.

Порівняльна таблиця технологій представлена на таблиці 1.2.

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						23
Ізм.	Лист	№ докум.	Підпис	Дата		

Таблиця 1.2. Порівняльна таблиця технологій

Параметр	Arduino	Raspberry Pi	ESP32
Тип пристрою	Мікроконтролер	Одноплатний комп'ютер	Мікроконтролер
Процесор	ATmega328 (Uno)	ARM Cortex-A53 (Pi 4)	Dual-core Xtensa LX6
Тактова частота	16 МГц	1.5 ГГц	До 240 МГц
Пам'ять (RAM)	2 КБ (Uno)	До 8 ГБ (Pi4)	520 КБ
Пам'ять (Flash)	32 КБ (Uno)	microSD до 256 ГБ	До 16 МБ
Виходи/входи (GPIO)	14 цифрових, 6 аналогових	До 40	До 36
Інтерфейси зв'язку	UART, I2C, SPI	UART, I2C, SPI, HDMI, USB	UART, I2C, SPI, Bluetooth, Wi-Fi
Підключення до Інтернету	Немає	Ethernet, Wi-Fi, USB адаптер	Вбудований Wi-Fi
Bluetooth	Немає	Через USB адаптер	Вбудований Bluetooth
Операційна система	Немає	Linux (Raspberry Pi OS)	Немає
Енергоспоживання	Низьке	Високе	Низьке
Ціна	Від \$20	Від \$35	Від \$5
Розмір	Маленький	Більший за Arduino	Маленький
Спеціалізація	Прості проекти, DIY	Комплексні проекти	IoT, бездротові проекти
Підтримка периферії	Обмежена	Широка	Широка
Підтримка бібліотек	Велика кількість	Велика кількість	Велика кількість

Висновки:

Arduino: Найкраще підходить для простих проектів і навчання основам мікроконтролерів. Завдяки низькому енергоспоживанню і простоті у використанні, це ідеальний вибір для новачків і для невеликих DIY проектів. Arduino має велику спільноту користувачів і багато бібліотек, що спрощує розробку і впровадження нових ідей. Однак, його обчислювальна потужність обмежена, що робить його менш підходящим для складніших завдань.

Raspberry Pi: Це більш потужний і універсальний пристрій, що підходить для комплексніших проектів, які потребують значних обчислювальних ресурсів. Raspberry Pi може працювати як повноцінний комп'ютер, підтримує різні операційні системи (включаючи спеціально розроблену Raspberry Pi OS), і має широкий спектр інтерфейсів для підключення периферії. Це робить його ідеальним для створення медіацентрів, домашніх серверів, навчальних проектів з програмування і робототехніки. Проте, він споживає більше енергії, ніж Arduino чи ESP32, і має більші розміри.

ESP32: Найкращий вибір для проектів Інтернету речей (IoT) завдяки вбудованим модулям Wi-Fi і Bluetooth, що значно спрощує створення бездротових пристроїв. Він має низьке енергоспоживання і високу продуктивність для своїх розмірів, що робить його ідеальним для мобільних і компактних пристроїв. Крім того, можливість писати код на MicroPython додає зручності у розробці, дозволяючи швидко і ефективно впроваджувати нові функції. Вбудовані функції і низька ціна роблять ESP32 чудовим вибором для проектів з обмеженим бюджетом, які потребують високої продуктивності.

Я вибрав ESP32 через його ціну, співвідношення продуктивності та ціни, компактність, низьке енергоспоживання, вбудований Wi-Fi, а також можливість писати код на MicroPython. Це дозволяє мені розробляти ефективні і функціональні IoT пристрої з мінімальними затратами та максимальними можливостями.

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						25
Ізм.	Лист	№ докум.	Підпис	Дата		

1.1.3 Вибір комплектуючих

Для реалізації індикації повітряних тривог на фізичній карті було обрано світлодіодну стрічку WS2812B. Цей вибір обумовлений кількома важливими характеристиками та перевагами, які забезпечують високу ефективність і зручність у використанні порівняно з іншими аналогами.

Основні характеристики WS2812B:

- Тип діодів: WS2812B, інтегровані RGB LED з драйверами;
- Напруга живлення: 5 В;
- Споживана потужність: до 60 мА на кожен діод при максимальній яскравості;
- Керування: Один цифровий пін для передачі даних;
- Протокол: Інтерфейс управління даними з послідовною передачею;
- Розміри: 5 x 5 мм (кожен діод);
- Колір: Підтримка повного спектру кольорів RGB (24 біти на піксель) [9].

Переваги WS2812B:

1. Інтеграція драйверів: Кожен світлодіод WS2812B має вбудований драйвер, що значно спрощує схему підключення та управління. Це зменшує кількість необхідних компонентів та спрощує монтаж.
2. Легкість у керуванні: Завдяки використанню одного цифрового піна для керування всією стрічкою, суттєво спрощується програмна частина проекту. Цей підхід дозволяє легко змінювати кольори та ефекти за допомогою програмного забезпечення.
3. Компактність: Невеликі розміри діодів (5 x 5 мм) дозволяють розмістити значну кількість світлодіодів на обмеженій площі, що є важливим для детального відображення інформації на карті.
4. Надійність і довговічність: Світлодіодні стрічки WS2812B відомі своєю високою надійністю та тривалим терміном служби, що є критичним фактором для систем, які повинні працювати безперебійно та точно.

На рис. 1.7 зображено світлодіодну стрічку WS2812B.



Рисунок 1.7. Світлодіодна стрічка WS2812B

Для забезпечення звукового сповіщення про повітряні тривоги в проєкті було обрано пасивний зумер. Цей вибір обумовлений декількома важливими характеристиками та перевагами, які роблять пасивний зумер оптимальним рішенням порівняно з іншими аналогами.

Пасивний зумер – це компонент, який використовується для створення звукових сигналів у різних електронних пристроях. Він перетворює електричну енергію на звукові коливання і є незамінним у таких проєктах, як системи сповіщення, електронні іграшки, охоронні системи, світлотехніка та побутова техніка [10].

Технічні характеристики пасивного зумера:

- Напруга живлення: 3-12 В;
- Опір: 15 Ом;
- Розміри: 12 x 7,5 мм;
- Спосіб підключення: Зумер має два контакти, позначені на платі “+” і “-”.

Для правильного підключення зумера потрібно дотримуватися полярності [10].

Переваги пасивного зумера:

1. Гнучкість у налаштуванні звуку: Пасивний зумер дозволяє створювати звукові сигнали різної частоти та тривалості, що робить його ідеальним для індивідуальних потреб кожного проєкту [11].

2. Енергоефективність: Пасивні зумери споживають значно менше енергії, ніж їх активні аналоги, що дозволяє збільшити час автономної роботи системи.
3. Простота інтеграції: Легке підключення до мікроконтролерів, таких як Arduino, ESP32 без необхідності використання додаткових компонентів.
4. Компактність: Малі розміри зумера дозволяють його використання у пристроях з обмеженим простором.
5. Доступність: Пасивні зумери є більш доступними за ціною, порівняно з активними моделями, що робить їх вигідним вибором для багатьох проектів [10].

Пасивний зумер може бути використаний у пристрої для створення звукових сповіщень, які будуть активовані в реальному часі відповідно до даних про повітряні тривоги. Його можна програмувати для відтворення різних звукових сигналів, що дозволить користувачам легко розрізняти між різними типами сповіщень.

На рис. 1.8 зображено пасивний зумер.



Рисунок 1.8. Пасивний зумер

Для живлення пристрою було обрано акумулятор типу 18650. Це літій-іонний акумулятор, який широко використовується в різних електронних пристроях, зокрема в портативній електроніці, ліхтарях, ноутбуках та інших пристроях, що потребують надійного та тривалого джерела живлення. Однією з основних переваг акумулятора 18650 є його висока ємність, що дозволяє забезпечити тривале автономне живлення системи без необхідності частого

перезаряджання. Ці акумулятори також відомі своєю стабільністю напруги, що є критично важливим для безперебійної роботи електронних компонентів. Крім того, компактні розміри акумулятора дозволяють інтегрувати його в різні конструкції без значного збільшення розмірів пристрою [12].

На рис. 1.9 зображено Li-іон акумулятор формату 18650.



Рисунок 1.9. Li-іон акумулятор формату 18650

Однак акумулятор 18650 має номінальну напругу 3.7 В, що не є достатнім для багатьох електронних компонентів, які потребують живлення 5 В. Для вирішення цієї проблеми використовується підвищуючий модуль MT3608. Цей модуль призначений для перетворення напруги з нижчого рівня до вищого, забезпечуючи стабільні 5 В для живлення всієї системи. Модуль MT3608 відрізняється своєю високою ефективністю перетворення енергії, що дозволяє зменшити втрати енергії і забезпечити максимальну продуктивність системи. Він має регульовану вихідну напругу, що дозволяє налаштовувати його під конкретні потреби проекту. Це робить його універсальним і зручним у використанні компонентом для багатьох електронних проектів.

Характеристики MT3608:

- Вхідна напруга: 2 - 24 В;
- Вихідна напруга: 2.5 - 28 В (регульована);
- Максимальний струм навантаження: 1 А;
- Максимальний вихідний ток: 2 А;

- Ефективність перетворення: до 93%;
- Розміри: 36 x 17 x 14 мм [13].

На рис. 1.10 зображено підвищуючий модуль MT3608.

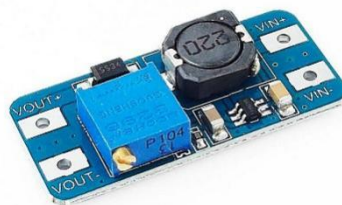


Рисунок 1.10. Підвищуючий модуль MT3608

Щоб забезпечити безпечну та ефективну зарядку акумулятора 18650, використовується плата заряду TP4056. Ця плата спеціально розроблена для зарядки літій-іонних акумуляторів і забезпечує оптимальний режим зарядки з урахуванням усіх вимог до безпеки. Плата TP4056 має вбудовані механізми захисту від перезарядки та короткого замикання, що гарантує безпечну роботу системи навіть у разі несправностей. Вона підключається до стандартного USB джерела живлення, що робить процес зарядки зручним і доступним. Крім того, плата обладнана світлодіодними індикаторами, які дозволяють легко контролювати процес зарядки та стан акумулятора.

Характеристики плати TP4056:

- Вхідна напруга: 4 - 8 В;
- Напруга вимкнення батареї під час заряджання: 4.2 В;
- Напруга вимкнення батареї при розряді: 2.5 В;
- Струм заряду: 50 - 1000 мА;
- Максимальний струм розряду: 3.5 А;
- Діапазон робочих температур: -10...+125°C [14].

На рис. 1.11 зображено плату заряду TP4056.

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						30
Ізм.	Лист	№ докум.	Підпис	Дата		



Рисунок 1.11 Плата заряду TP4056

Для вмикання та вимикання пристрою додав перемикач. Це дозволяє легко контролювати подачу живлення на пристрій.

На рис. 1.12 зображено перемикач (тумблер) KCD5-102.



Рисунок 1.12. Перемикач (тумблер) KCD5-102

Також додано дві тактові кнопки до пристрою:

Одна кнопка призначена для запуску точки доступу Wi-Fi. При її натисканні, пристрій створює власну Wi-Fi мережу для підключення до веб-серверу.

Друга кнопка використовується для керування режимом підсвітки.

На рис. 1.13 зображено тактову кнопку.



Рисунок 1.13. Тактова кнопка

1.2 Проєктування моделі пристрою

1.2.1 Технічне завдання на розробку

До пристрою, що розробляється, ставляться наступні вимоги:

1. Надійність та швидкодія системи: Система повинна працювати безперебійно та забезпечувати швидке оновлення статусу тривоги в кожній області, оскільки від цього залежить безпека громадян.
2. Точність відображення даних: Карта повинна точно відображати статус повітряної тривоги для кожної області України, використовуючи актуальні дані з офіційних джерел.
3. Ефективна світлодіодна індикація: Кожна область на карті повинна бути представлена світлодіодами, які чітко відображають статус тривоги (червоний - тривога, зелений – немає тривоги).
4. Якісне звукове оповіщення: Бузер повинен генерувати різні звукові сигнали для оголошення тривоги та її відбою, які будуть достатньо гучними та розпізнаваними.
5. Енергоефективність: ESP32 та інші компоненти повинні оптимально використовувати енергію, забезпечуючи тривалу роботу пристрою.
6. Простота в налаштуванні: Програмне забезпечення для ESP32 має бути легким у встановленні та налаштуванні, щоб користувачі могли швидко підключити пристрій до Wi-Fi.
7. Резервне живлення: Пристрій повинен мати можливість роботи від батареї або акумулятора.

1.2.2 Проєктування моделі пристрою

Пристрій, що розробляється, буде включати створення фізичного пристрою на базі ESP32. Для написання коду використовуватиметься мова програмування MicroPython в середовищі розробки Thonny IDE.

MicroPython – це інтерпретована мова програмування високого рівня, оптимізована для мікроконтролерів і вбудованих систем. Основні переваги

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						32
Ізм.	Лист	№ докум.	Підпис	Дата		

MicroPython полягають у його простоті, ефективності та широких можливостях для роботи з апаратним забезпеченням.

Переваги MicroPython:

1. Простота та читабельність коду: MicroPython успадкував багато характеристик від Python, зокрема простий синтаксис і високу читабельність коду. Це робить його зручним для розробників різного рівня досвіду, що особливо важливо для швидкого прототипування і розробки.
2. Велика кількість бібліотек: Для MicroPython доступна велика кількість бібліотек і модулів, які дозволяють легко додавати нові функції до проекту. Наприклад, існують бібліотеки для роботи з мережами, сенсорами, дисплеями, файловими системами та іншими периферійними пристроями. Це значно спрощує процес розробки, оскільки не потрібно писати код з нуля для багатьох стандартних задач.
3. Асинхронне програмування: MicroPython підтримує асинхронне програмування, що дозволяє ефективно керувати ресурсами мікроконтролера і забезпечувати високу продуктивність. Асинхронність особливо корисна у випадках, коли необхідно обробляти множинні події або запити одночасно, що є частою вимогою в IoT-проектах.
4. Компактність та ефективність: MicroPython спеціально розроблений для роботи в обмежених умовах, таких як мікроконтролери з обмеженою оперативною пам'яттю та процесорною потужністю. Це робить його ідеальним вибором для проектів, які використовують мікроконтролери.
5. Підтримка широкого спектру мікроконтролерів: MicroPython підтримує роботу з багатьма популярними мікроконтролерами, такими як ESP32, STM32, та інші. Це забезпечує гнучкість у виборі апаратної платформи для проекту.
6. Велика спільнота розробників: Спільнота розробників, яка підтримує MicroPython, активно створює та ділиться прикладами коду, бібліотеками, і навчальними матеріалами. Це значно спрощує процес навчання та вирішення проблем, які можуть виникнути під час розробки.

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						33
Ізм.	Лист	№ докум.	Підпис	Дата		

1.3 Реалізація пристрою

1.3.1 Підготовка та налаштування середовища розробки Thonny IDE

Thonny IDE – це інтегроване середовище розробки, яке було створено з метою надання зручного інструменту для програмування на Python, особливо для початківців. Це IDE підтримує MicroPython, що робить його ідеальним для розробки проектів на мікроконтролерах, таких як ESP32.

Основною перевагою Thonny IDE є його простота та інтуїтивно зрозумілий інтерфейс, що дозволяє легко писати, тестувати та відлагоджувати код. Воно має вбудований інтерпретатор Python, який дозволяє виконувати код безпосередньо у середовищі IDE. Також Thonny надає можливість легко встановлювати та керувати пакетами Python, що робить його корисним для широкого спектру проектів.

Для роботи з ESP32 та MicroPython, Thonny IDE дозволяє легко завантажувати скрипти на мікроконтролер та взаємодіяти з ним через серійний порт. Це робить процес розробки значно швидшим та ефективнішим.

Thonny IDE сумісне з операційними системами Windows, Mac OS X та Linux і вже встановлене за замовчуванням на Raspberry Pi OS. Це відкрите програмне забезпечення, тому його можна безкоштовно завантажити та використовувати для своїх проектів.

Використання Thonny IDE для розробки на ESP32 з MicroPython хорошим вибором, оскільки воно спрощує процес програмування та відлагодження, а також підтримує всі необхідні інструменти для роботи з мікроконтролерами [15].

Щоб завантажити Thonny IDE потрібно перейти за посиланням на офіційний сайт [16]. Далі потрібно обрати потрібну версію для своєї операційної системи, та завантажити слідуючи інструкціям.

Для роботи та розпізнавання плати ESP32 потрібно встановити драйвер CH341SER

Далі потрібно зайти в саму Thonny IDE, зайти до меню Інструменти – Параметри. У вікні, що відкрилося потрібно перейти до вкладки Інтерпретатор,

далі слід обрати інтерпретатор для ESP32, обрати потрібний COM Port в який підключено ESP32, та встановити прошивку.

Для зручності можна відкрити провідник усередині Thonny IDE, для цього потрібно натиснути кнопку «Вид», та у діалоговому вікні натиснути «Файли».

1.3.2 Збір пристрою

Спочатку підключимо акумулятор 18650 до плати заряду TP4056 вбудованим захистом, щоб забезпечити безпечне і стабільне заряджання. Після цього налаштуємо підключення від плати заряду до підвищуючого перетворювача MT3608, який дозволить збільшити напругу до необхідного рівня для живлення пристрою. Цей процес включає кілька кроків, які детально розглянемо для забезпечення коректної роботи всієї системи.

Для початку потрібно підключити акумулятор 18650 до плати заряду TP4056. Плата заряду TP4056 є спеціальним контролером, який дозволяє заряджати літій-іонні акумулятори, такі як 18650, через Type-C. Ця плата також має вбудовану систему захисту, яка запобігає перезарядженню, надмірному розряду і короткому замиканню акумулятора, що забезпечує додаткову безпеку під час заряджання.

На платі TP4056 є контакти, позначені як "В+" і "В-", які призначені для підключення акумулятора. Позитивний контакт акумулятора 18650 підключається до контакту "В+" на платі TP4056, а негативний контакт акумулятора підключається до контакту "В-". Це підключення забезпечує можливість заряджання акумулятора через плату TP4056, коли вона підключена до джерела живлення через Type-C.

На рис. 1.14 зображено схему підключення акумулятора до плати TP4056.

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						35
Ізм.	Лист	№ докум.	Підпис	Дата		

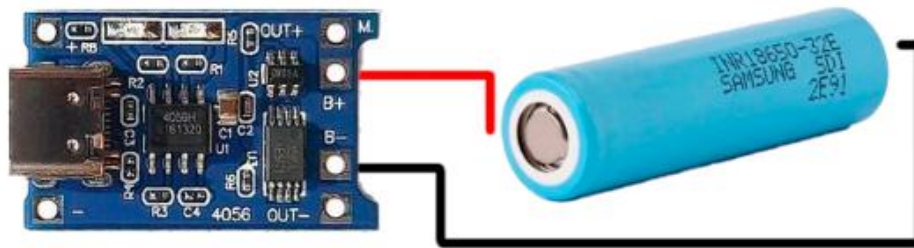


Рисунок 1.14 Схема підключення акумулятора до плати TP4056

На платі TP4056 є контакти, позначені як "B+" і "B-", які призначені для підключення акумулятора. Позитивний контакт акумулятора 18650 підключається до контакту "B+" на платі TP4056, а негативний контакт акумулятора підключається до контакту "B-". Це підключення забезпечує можливість заряджання акумулятора через плату TP4056, коли вона підключена до джерела живлення через Type-C. Після підключення акумулятора до плати заряду необхідно підключити плату заряду до підвищуючого перетворювача MT3608. MT3608 є DC-DC перетворювачем, який здатен підвищувати вхідну напругу до необхідного рівня, що дозволяє жити різноманітні пристрої. На платі TP4056 є контакти "OUT+" і "OUT-", які видають напругу акумулятора. Ці контакти потрібно підключити до входів "IN+" і "IN-" на платі MT3608 відповідно. Тобто, контакт "OUT+" на TP4056 підключається до контакту "IN+" на MT3608, а контакт "OUT-" на TP4056 підключається до контакту "IN-" на MT3608.

На рис. 1.15 зображено схему підключення акумулятора з платою TP4056 до підвищуючого модуля MT3608.

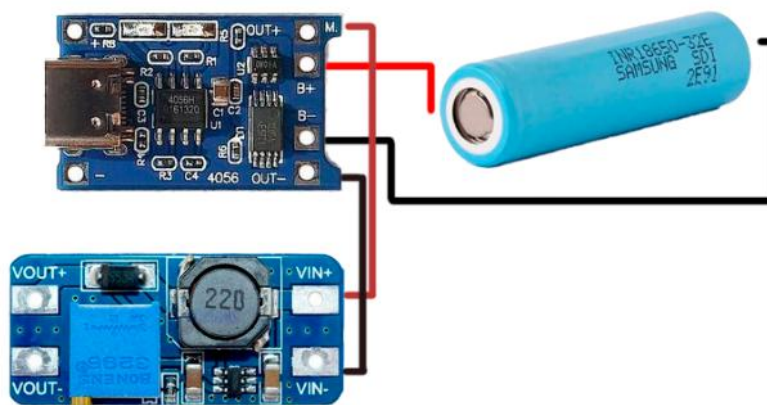


Рисунок 1.15. Схема підключення акумулятора з платою TP4056 до підвищувачого модуля MT3608

Для підвищення напруги треба повертати потенціометр проти годинникової стрілки. Перед першим використанням треба повернути потенціометр проти годинникової стрілки на 15-20 оборотів. Для перевірки напруги рекомендується використовувати мультиметр. З його допомогою можна дізнатись яка вихідна напруга на платі MT3608.

Для підключення вимикача до схеми, щоб можна було включати і вимикати пристрій, вимикач потрібно розмістити між платою заряду TP4056 і підвищувачим перетворювачем MT3608.

На рис. 1.16 зображено схему підключення кнопки.

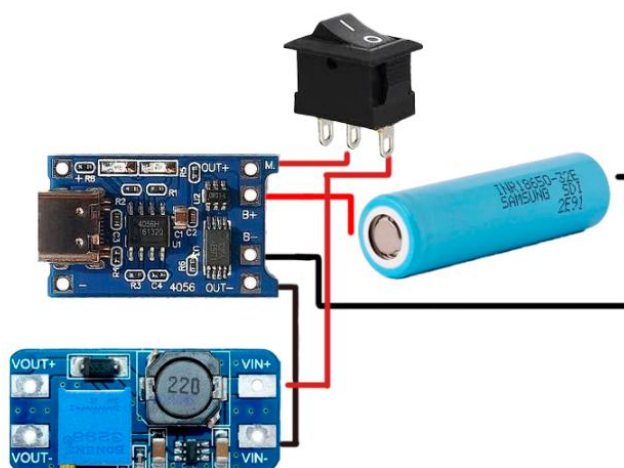


Рисунок 1.16. Схема підключення кнопки

Наступним етапом є підключення розробленої схеми живлення до мікроконтролерної плати ESP32. Схема живлення складається з літій-іонного акумулятора 18650, модуля заряджання TP4056 з захистом та підвищувального перетворювача напруги MT3608.

Підключення виконується наступним чином:

Вихідні контакти MT3608:

1. VOUT+ з'єднується з пін VIN на платі ESP32. Цей пін розрахований на вхідну напругу 5В.
2. VOUT- (негативний вихід або "земля") підключається до будь-якого з пінів GND на ESP32. На платі є кілька таких пінів, всі вони еквівалентні.

Важливо зазначити, що неправильне підключення може призвести до серйозних наслідків:

1. Подача напруги від MT3608 на пін 3.3V або будь-який GPIO пін ESP32 пошкодить плату.
2. Переплутані VOUT+ і VOUT- викличуть коротке замикання.
3. Неправильна полярність акумулятора 18650 може пошкодити TP4056 або навіть спричинити загоряння.

На рис. 1.17 зображено схему підключення живлення до плати ESP32.

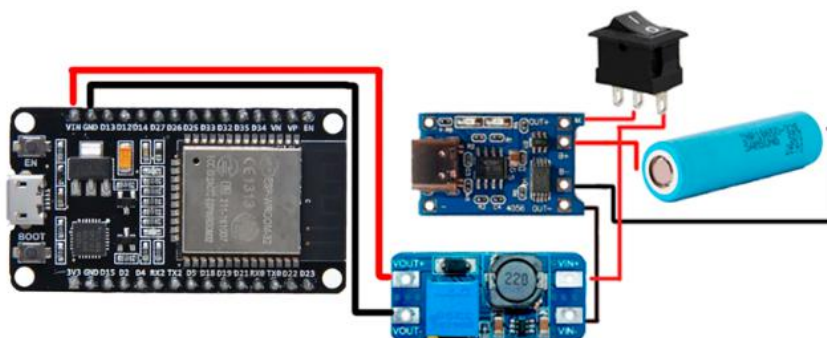


Рисунок 1.17. Схема підключення живлення до плати ESP32

Наступним етапом є підключення світлодіодної стрічки WS2812B до плати ESP32. WS2812B – це популярний тип адресних світлодіодів, також відомих під торговою маркою NeoPixel. Кожен світлодіод у стрічці містить вбудований драйвер, що дозволяє індивідуально керувати кольором та яскравістю кожного пікселя.

Контакти на стрічці WS2812B:

1. VCC (або +5V): Це позитивний вхід живлення. Стрічка WS2812B вимагає стабільного джерела живлення 5В. Важливо забезпечити достатню потужність блоку живлення, оскільки кожен світлодіод може споживати до 60 мА при максимальній яскравості білого кольору. Наприклад, стрічка з 60 світлодіодами може споживати до 3,6 А.
2. GND (або -): Це негативний вхід або "земля". Він повинен бути з'єднаний з GND джерела живлення та GND мікроконтролера для створення спільної точки відліку напруги.
3. DIN (або DATA IN): Це вхід даних. Саме через цей контакт мікроконтролер передає інструкції про колір та яскравість кожного світлодіода. У WS2812B використовується однопровідний послідовний протокол, що дозволяє керувати довгими ланцюжками світлодіодів лише одним сигнальним проводом.
4. DOUT (або DATA OUT): Це вихід даних, присутній на деяких стрічках. Він дозволяє з'єднувати кілька стрічок послідовно, передаючи дані від виходу DOUT однієї стрічки до входу DIN наступної. У багатьох випадках цей контакт може бути відсутній, оскільки той самий функціонал забезпечується останнім світлодіодом у стрічці.

На рис. 1.18 зображено схему підключення світлодіодної стрічки до плати ESP32.

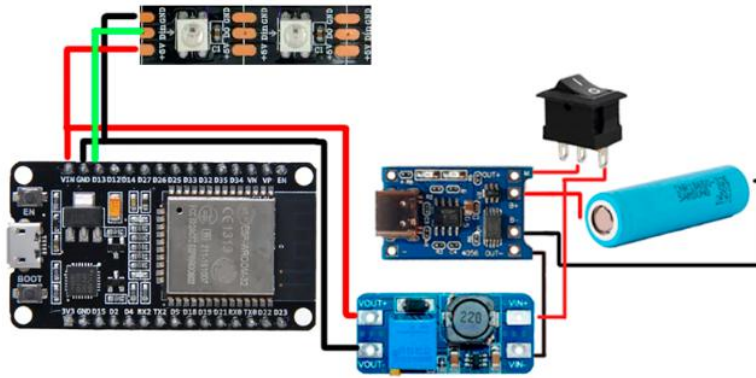


Рисунок 1.18. Схема підключення світлодіодної стрічки до плати ESP32

Наступним етапом є підключення зумера до схеми з мікроконтролером ESP32 та світлодіодною стрічкою WS2812B. Зумер – це пристрій, який перетворює електричні сигнали в звукові коливання, дозволяючи генерувати різноманітні звуки.

Існують два основні типи зумерів:

Активний зумер – містить вбудований генератор, що дозволяє йому генерувати звук самостійно, коли до нього подається постійна напруга. Для його роботи не потрібно додаткових сигналів або спеціального драйвера [17].

Контакти:

1. VCC – підключається до позитивного полюса джерела живлення, який йде з модуля MT3608
2. GND – підключається до негативного полюса (земля, GND) плати ESP32

Пасивний зумер – не має вбудованого генератора і потребує зовнішнього сигналу для створення звуку. Він працює лише тоді, коли отримує змінний струм або спеціально сформований сигнал (наприклад, PWM) [18].

Контакти:

1. VCC – підключається до позитивного полюса джерела живлення.
2. GND – підключається до негативного полюса (земля, GND).

Підключення до ESP32:

Для підключення пасивного зумера до плати ESP32 треба підключити контакт VCC до любого GPIO контакта на платі ESP32, а землю (GND) підключити до контакту GND на платі.

На рис. 1.19 зображено схему підключення зумера до плати ESP32.

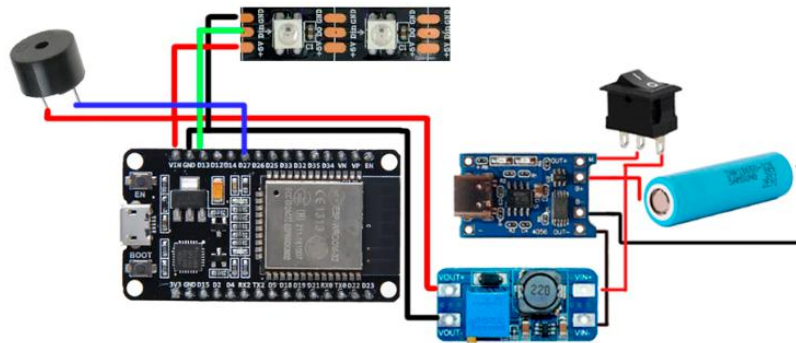


Рисунок 1.19. Схема підключення зумера до плати ESP32

Останнім кроком є підключення звичайної тактової кнопки, для цього:

1. Один контакт кнопки підключається до пін GND на платі ESP32.
2. Інший контакт кнопки з'єднується з вільним GPIO піном на ESP32 через Цей пін буде використовуватися для зчитування стану кнопки.
3. При натисканні кнопки на GPIO пін буде подаватися високий рівень напруги (логічна 1). При відпусканні кнопки, на пін знову буде подаватися низький рівень (логічна 0).
4. У програмному забезпеченні потрібно налаштувати обраний GPIO пін як вхідний та реалізувати опитування його стану в циклі або за допомогою переривань.

На рис. 1.20 зображено схему підключення тактової кнопки до схеми.

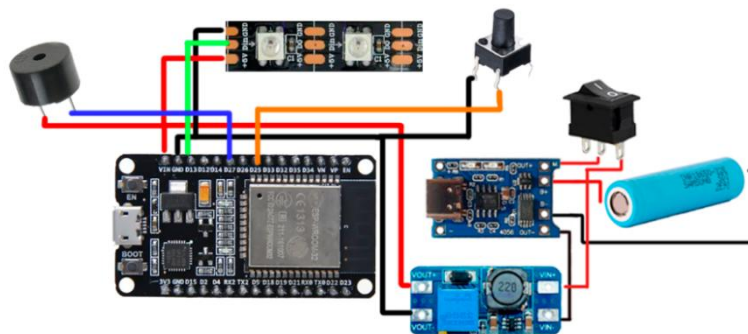


Рисунок 1.20. Схема підключення тактової кнопки до схеми

1.3.3 Розробка прошивки

На початку коду відбувається імпорт необхідних бібліотек та модулів. Це критично важливий етап, адже без потрібних бібліотек програма не зможе працювати належним чином.

Нижче наведено код імпорту модулів:

```
import network
import socket
from time import sleep, time
import esp
import gc
import machine
import json
import _thread
from neopixel import NeoPixel
import urequests
from machine import Pin, PWM
```

Опишемо цей код:

Бібліотека `network` дозволяє працювати з мережевими підключеннями, такими як WiFi. Вона надає функції для сканування доступних мереж, підключення до них та керування з'єднанням.

Модуль `socket` забезпечує низькорівневий доступ до мережесокетів, що дозволяє програмі відправляти та отримувати дані через мережу. Це важливо для налаштування веб-сервера для конфігурації WiFi.

Функції `sleep` та `time` з модуля `time` використовуються для затримки виконання коду та отримання поточного часу відповідно. Це необхідно для реалізації певних затримок та таймаутів у програмі.

Бібліотека `esp` забезпечує доступ до низькорівневих функцій мікроконтролера ESP32, на якому працює програма. Наприклад, за допомогою функції `osdebug(None)` вимикається виведення налагоджувальної інформації в консоль для економії пам'яті.

Модуль `gc` (garbage collector) використовується для управління пам'яттю та збирання сміття, що дозволяє ефективно використовувати обмежені ресурси пам'яті ESP32.

Бібліотека `machine` надає доступ до апаратних ресурсів мікроконтролера, таких як GPIO (General Purpose Input/Output), що дозволяє керувати світлодіодами та іншими пристроями, підключеними до виводів ESP32.

Модуль `json` забезпечує кодування та декодування даних у форматі JSON, який використовується для зберігання та читання налаштувань WiFi.

Бібліотека `_thread` дозволяє створювати та керувати потоками в програмі, що забезпечує можливість одночасного виконання декількох задач, наприклад, блимання світлодіода та очікування підключення до WiFi.

Модуль `neopixel` містить клас `NeoPixel`, який використовується для керування адресними світлодіодами `NeoPixel`. Це дозволяє індивідуально керувати кольором кожного світлодіода.

Бібліотека `urequests` забезпечує функціональність для виконання HTTP-запитів, що необхідно для отримання даних про стан тривог у різних областях України з API `alerts.net.ua`.

Далі у коді визначаються константи `PIN` та `NUMPIXELS`, які вказують номер виводу GPIO, до якого підключені світлодіоди `NeoPixel`, та кількість світлодіодів відповідно.

Нижче наведено код констант:

```
PIN = 12
NUMPIXELS = 25
```

Опишемо цей код:

Масив `states` містить назви всіх областей України, які будуть відображатися на карті тривог.

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						43
Ізм.	Лист	№ докум.	Підпис	Дата		

Створюється об'єкт пр класу NeoPixel з вказаними параметрами PIN та NUMPIXELS. Цей об'єкт буде використовуватися для керування світлодіодами NeoPixel.

Нижче наведено код створення класу NeoPixel:

```
esp.osdebug(None)
gc.collect()

station = network.WLAN(network.STA_IF)
if station.isconnected():
    station.active(False)

PIN = 12
NUMPIXELS = 25

states = ["Закарпатська область", "Івано-Франківська область", "Тернопільська область", "Львівська область", "Волинська область", "Рівненська область", "Житомирська область", "Київська область", "Чернігівська область", "Сумська область", "Харківська область", "Луганська область", "Донецька область", "Запорізька область", "Херсонська область", "Автономна Республіка Крим", "Одеська область", "Миколаївська область", "Дніпропетровська область", "Полтавська область", "Черкаська область", "Кіровоградська область", "Вінницька область", "Хмельницька область", "Чернівецька область"]

pin = Pin(PIN, Pin.OUT)
np = NeoPixel(pin, NUMPIXELS)
```

Опишемо цей код:

1. `esp.osdebug(None)`: вимикає відлагодні повідомлення для зменшення виводу інформації в консоль, що допомагає уникнути зайвого шуму під час роботи програми.
2. `gc.collect()`: викликає збір сміття, що звільняє невикористовувану пам'ять, забезпечуючи стабільну роботу системи.
3. `station = network.WLAN(network.STA_IF)`: ініціалізує WiFi модуль в режимі станції (STA).
4. `if station.isconnected(): station.active(False)`: при включенні пристрою вимикає WiFi модуль, якщо він уже підключений.
5. `PIN = 12`: визначає номер піну для підключення світлодіодної стрічки.
6. `NUMPIXELS = 25`: визначає кількість світлодіодів у стрічці.
7. `states`: список областей України, для яких буде відображатися статус повітряної тривоги.
8. `pin = Pin(PIN, Pin.OUT)`: налаштовує пін для виводу.

9. np = NeoPixel(pin, NUMPIXELS): ініціалізує світлодіодну стрічку NeoPixel.

Нижче наведено код управління світлодіодами:

```
def turn_off_leds():
    for i in range(NUMPIXELS):
        np[i] = (0, 0, 0)
    np.write()

turn_off_leds()

led = machine.Pin(2, machine.Pin.OUT)

wifi_config_file = 'config.json'

def load_wifi_config():
    try:
        with open(wifi_config_file, 'r') as f:
            config = json.load(f)
            return config['ssid'], config['password']
    except OSError:
        return None, None

def save_wifi_config(ssid, password):
    config = {'ssid': ssid, 'password': password}
    with open(wifi_config_file, 'w') as f:
        json.dump(config, f)

def blink_red_error():
    station = network.WLAN(network.STA_IF)
    while not station.isconnected():
        for i in range(25):
            np[i] = (125, 0, 0)
            np.write()
            sleep(0.5)
        for i in range(25):
            np[i] = (0, 0, 0)
            np.write()
            sleep(0.5)

def reset_wifi_config(pin):
    print("Кнопка сброса нажата, сброс настроек WiFi...")
    try:
        import os
        os.remove(wifi_config_file)
    except OSError:
        pass
    machine.reset()
```

Опишемо цей код:

1. turn_off_leds(): вимикає всі світлодіоди на стрічці, встановлюючи їх значення в (0, 0, 0) (чорний колір) і оновлює стрічку за допомогою np.write().
2. led = machine.Pin(2, machine.Pin.OUT): налаштовує пін для виводу, який буде використовуватися для індикації стану WiFi підключення.

3. `wifi_config_file = 'config.json'`: визначає назву файлу, в якому будуть зберігатися WiFi налаштування.
4. `load_wifi_config()`: функція для завантаження WiFi налаштувань з файлу `config.json`. Повертає SSID та пароль або None, якщо файл не знайдено.
5. `save_wifi_config(ssid, password)`: зберігає WiFi налаштування (SSID та пароль) у файл `config.json`.
6. `blink_red_error()`: функція для мигання червоними світлодіодами у випадку, якщо не вдається підключитися до WiFi. Світлодіоди мигають з інтервалом у 0.5 секунди.
7. `reset_wifi_config(pin)`: функція для скидання WiFi налаштувань при натисканні кнопки. Видаляє файл `config.json` і перезавантажує мікроконтролер.

Нижче наведено код конфігурації WiFi:

```
html_template = """<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Налаштування WiFi ESP32</title>
  </head>
  <body>
    <h1>Налаштування WiFi</h1>
    <form action="/configure" method="post">
      <label for="ssid">SSID:</label>
      <select id="ssid" name="ssid">
        {options}
      </select><br><br>
      <label for="password">Пароль:</label>
      <input type="password" id="password" name="password"><br><br>
      <input type="submit" value="Отправить">
    </form>
  </body>
</html>
"""

def scan_networks():
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    networks = wlan.scan()
    options = ''
    for ssid, bssid, channel, RSSI, authmode, hidden in networks:
        options += f'<option value="{ssid.decode()}">{ssid.decode()}</option>'
    return options

ssid_options = scan_networks()

addr = socket.getaddrinfo('0.0.0.0', 80)[0][-1]
s = socket.socket()
```

```
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
s.bind(addr)
s.listen(1)
```

Опишемо цей код.

Код налаштовує веб-сервер на ESP32, використовуючи модуль socket. Для цього створюється сокет s, який прив'язується до IP-адреси 0.0.0.0 та порту 80 (стандартний порт для HTTP).

HTML-шаблон html_template містить розмітку веб-сторінки для вибору SSID (назви WiFi-мережі) та введення пароля. Шаблон містить поле select для вибору SSID та поле input для введення пароля.

На рис. 1.22 зображено веб-сторінку підключення до ESP32 до WiFi.

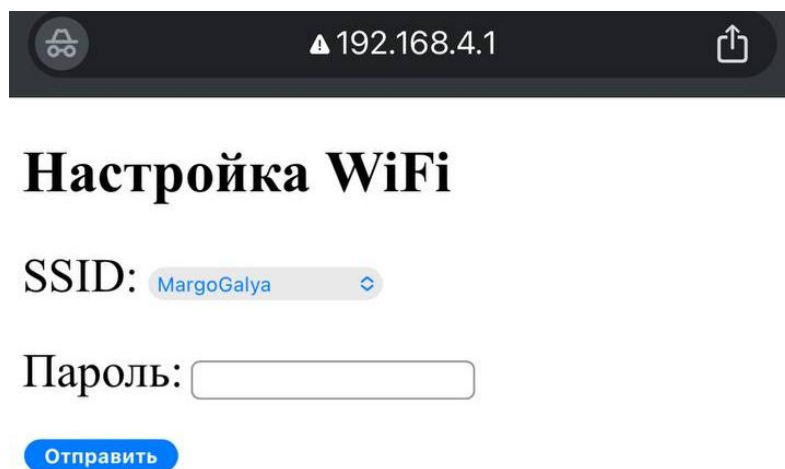


Рисунок 1.22. Веб-сторінка підключення до ESP32 доWiFi

Нижче наведено код підключення до WiFi та налаштування точки доступу:

```
ssid, password = load_wifi_config()

if ssid and password:
    station = network.WLAN(network.STA_IF)
    station.active(True)
    while not station.isconnected():
        try:
            station.connect(ssid, password)
            sleep(5)
        except:
            _thread.start_new_thread(blink_red_error, ())
            sleep(10)

    connected = True
else:
    ap = network.WLAN(network.AP_IF)
    ap.active(True)
```

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						47
Ізм.	Лист	№ докум.	Підпис	Дата		


```

ap.config(essid='ESP32-Setup', password='12345678')

ip = ap.ifconfig()[0]
print('Точка доступа запущена. IP адрес:', ip)
connected = False

```

Опишемо цей код.

Якщо конфігурація WiFi була знайдена, код намагається підключитися до збереженої мережі. Якщо підключення не вдається, викликається функція `blink_red_error()` в окремому потоці за допомогою `_thread.start_new_thread(blink_red_error, ())`. Це дозволяє програмі продовжувати виконання основного циклу, поки блимає світлодіод.

Якщо конфігурації WiFi немає, то код налаштовує точку доступу WiFi з назвою "ESP32-Setup" та паролем "12345678". Ця точка доступу буде використовуватися для налаштування WiFi-з'єднання через веб-інтерфейс.

1. `ssid, password = load_wifi_config()`: завантажує WiFi налаштування з файлу.
2. `if ssid and password`: якщо налаштування існують, мікроконтролер намагається підключитися до WiFi:
3. `station = network.WLAN(network.STA_IF)`: ініціалізує WiFi модуль у режимі станції.
4. `station.active(True)`: активує WiFi модуль.
5. `while not station.isconnected()`: у циклі намагається підключитися до WiFi.
6. `station.connect(ssid, password)`: намагається підключитися до мережі з заданими SSID і паролем.
7. `sleep(5)`: затримка на 5 секунд перед повторною спробою підключення.
8. `except`: у випадку помилки запускає потік з функцією `blink_red_error` і чекає 10 секунд перед повторною спробою.
9. `connected = True`: встановлює прапорець підключення, якщо підключення успішне.

Якщо налаштування не завантажені:

1. `ap = network.WLAN(network.AP_IF)`: ініціалізує WiFi модуль у режимі точки доступу.
2. `ap.active(True)`: активує точку доступу.

3. `ap.config(essid='ESP32-Setup', password='12345678')`: налаштовує точку доступу з заданими SSID і паролем.
4. `ip = ap.ifconfig()[0]`: отримує IP адресу точки доступу.
5. `print('Точка доступа запущена. IP адрес:', ip)`: виводить IP адресу точки доступу.
6. `connected = False`: встановлює прапорець відсутності підключення.

Нижче наведено код підключення до WiFi і API:

```
def connect_to_wifi(ssid, password):
    station = network.WLAN(network.STA_IF)
    station.active(True)
    station.connect(ssid, password)

    start_time = time()
    timeout = 5

    while not station.isconnected():
        if time() - start_time > timeout:
            print('Подключение не удалось')
            station.active(False)
            return False
        sleep(1)
    print('Успешное подключение')
    save_wifi_config(ssid, password)
    return True

def connect_api():
    print("Подключение к API...")
    api_url = "http://alerts.net.ua/alerts_statuses_v3.json"
    try:
        response = urequests.get(api_url)
        if response.status_code == 200:
            data = response.json()
            for i, oblast in enumerate(oblasti):
                status = data["states"].get(oblast, False)
                if status:
                    print(f"{oblast}: {status}")
                np[i] = (125, 0, 0) if status else (0, 125, 0)
                np.write()
                sleep(0.1)
```

Опишемо цей код.

Якщо функція `connect_to_wifi(ssid, password)` успішно підключається до вказаної WiFi-мережі, то вона зберігає дані SSID та пароль у файл `config.json` за допомогою функції `save_wifi_config(ssid, password)`. Після цього відправляється HTTP-відповідь клієнту з повідомленням про успішне підключення до WiFi та вимикається точка доступу за допомогою `ap.active(False)`.

Якщо підключення до WiFi не вдалося, то відправляється HTTP-відповідь з повідомленням про помилку та інструкціями для повторної спроби.

Якщо функція `connect_to_wifi(ssid, password)` успішно підключається до вказаної WiFi-мережі, то вона зберігає дані SSID та пароль у файл `config.json` за допомогою функції `save_wifi_config(ssid, password)`. Після цього відправляється HTTP-відповідь клієнту з повідомленням про успішне підключення до WiFi та вимикається точка доступу за допомогою `ap.active(False)`.

Якщо підключення до WiFi не вдалося, то відправляється HTTP-відповідь з повідомленням про помилку та інструкціями для повторної спроби.

Після обробки запиту з'єднання з клієнтом закривається за допомогою `cl.close()`.

Функція `connect_api()` відправляє GET-запит до API за допомогою `urequests.get(api_url)`. Якщо відповідь успішна (статус-код 200), то функція розбирає JSON-дані з відповіді та перевіряє статуси тривоги для кожної області.

Для кожної області функція встановлює відповідний колір світлодіода NeoPixel залежно від стану тривоги. Якщо в області є тривога, світлодіод встановлюється на червоний колір (125, 0, 0), інакше - на зелений (0, 125, 0). Після встановлення кольору світлодіода, функція `np.write()` застосовує ці зміни до фізичних світлодіодів.

Якщо під час виконання `connect_api()` виникає виняткова ситуація (наприклад, помилка мережі або неможливість отримати дані з API), то всі світлодіоди NeoPixel блимають червоним кольором, сигналізуючи про проблему.

Нижче наведено код перевірки підключення до WiFi:

```
while True:
    if connected:
        connect_api()
        sleep(15)

    if not connected:
        try:
            cl, addr = s.accept()
            request = cl.recv(1024)
            request = request.decode('utf-8')

            if 'POST /configure' in request:
                _, data = request.split('\r\n\r\n')
```

					КС 57. 03 001. 00 ДП ПЗ	Арк.
						50
Ізм.	Лист	№ докум.	Підпис	Дата		

```

        params = data.split('&')
        ssid = params[0].split('=')[1]
        password = params[1].split('=')[1]

        if connect_to_wifi(ssid, password):
            response = "HTTP/1.1 200 OK\r\nContent-Type: text/html; charset=UTF-8\r\n\r\n"
            response += "<html><body><h1>Подключение к WiFi успешно!</h1><p>Отключаю точку доступа...</p></body></html>"

            cl.send(response.encode('utf-8'))
            cl.close()

            sleep(3)
            ap.active(False)

        else:
            response = "HTTP/1.1 200 OK\r\nContent-Type: text/html; charset=UTF-8\r\n\r\n"
            response += f"<html><body><h1>Не удалось подключиться. Попробуйте еще раз.</h1><p>Перейдите по адресу <a href='http://{ip}'>http://{ip}</a></p></body></html>"
            else:
                network.WLAN(network.STA_IF).active(False)
                ap.active(True)

            response = "HTTP/1.1 200 OK\r\nContent-Type: text/html; charset=UTF-8\r\n\r\n"
            response += html_template.format(options=ssid_options)

            cl.send(response.encode('utf-8'))
            cl.close()
    except Exception as e:
        print('Ошибка:', e)
    try:
        cl.close()
    except Exception as e:
        print('Ошибка при закрытии соединения:', e)

```

Опишемо цей код.

Цей код знаходиться в нескінченному циклі `while True`. Він перевіряє, чи встановлено підключення до WiFi (`if connected:`). Якщо підключення встановлено, то викликається функція `connect_api()` для отримання даних про стан тривог з API, після чого робиться затримка на 15 секунд (`sleep(15)`).

Якщо підключення до WiFi відсутнє (`if not connected:`), код очікує на підключення клієнта до веб-сервера за допомогою `cl, addr = s.accept()`. Після отримання запиту від клієнта (`request = cl.recv(1024)`), код перевіряє, чи це запит на конфігурацію WiFi (`if 'POST /configure' in request:`).

Якщо це запит на конфігурацію WiFi, код витягує дані SSID та пароля з запиту та намагається підключитися до вказаної мережі за допомогою функції `connect_to_wifi(ssid, password)`. Залежно від результату підключення, відправляється відповідне повідомлення клієнту.

Якщо це не запит на конфігурацію WiFi, код деактивує режим станції (STA), активує режим точки доступу (AP) та генерує HTML-код веб-сторінки налаштувань WiFi з доступними мережами.

Після обробки запиту, відповідь відправляється клієнту за допомогою `cl.send(response.encode('utf-8'))`, а з'єднання закривається (`cl.close()`).

Будь-які виняткові ситуації, що виникають під час обробки запиту, обробляються в блоці `except` та виводяться в консоль [19, 20, 21, 22].

1.4 Тестування пристрою

Переконаємось в працездатності реалізованого пристрою.

На рис. 1.23 зображено пристрій в працюючому стані під час повітряної тривоги.



Рисунок 1.23. Пристрій в працюючому стані під час повітряної тривоги

2 ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

Темою даного проекту є розробка фізичної карти повітряних тривог. Метою цього проекту є розробка фізичної карти повітряних тривог, яка дозволить оперативно інформувати населення про ракетну небезпеку в різних регіонах України. Основні завдання включають створення зрозумілої візуалізації зон небезпеки та інтеграцію звукового сигналу для привернення уваги до тривоги. Крім того, проект передбачає використання карти як елементу дизайну, що додасть їй додаткову цінність і функціональність. Важливою частиною роботи є забезпечення простоти використання системи для всіх категорій населення.

2.2 Вартісна оцінка розробленого пристрою

У даному розділі визначається вартісна оцінка розробленого пристрою. Спочатку визначається калькуляція розробленого виробу укрупненим методом через вартість покупних комплектуючих елементів і виробів, для визначення якої складаємо перерахування елементів і виробів на основі відомості специфікацій (принципової схеми) по формі, приведеної в таблиці 2.1.

					КС 57. 03 002. 00 ДП ПЗ	Арк.
						53
Ізм.	Лист	№ докум.	Підпис	Дата		

Таблиця 2.1. Розрахунок відомості покупних комплектуючих елементів

Найменування, тип, модель	Од.вим	Норма витрат на виріб	Ціна , грн.	Вартість комплектуючих
Мікроконтроллер ESP32	шт..	1	200	250
Світлодіодна стрічка WS2812B	шт..	1	100	100
Плата заряду TP4056	шт..	1	25	25
Підвищуючий DC/DC перетворювач на MT3608	шт..	1	20	20
Акумулятор Samsung 32E	шт..	1	100	100
Перемикач KCD5-102.	шт..	1	10	10
Тактова кнопка 6x6x12	шт..	2	2	4
Корпус	шт..	1	250	250
Загальна вартість покупних комплектуючих елементів				709
Транспортні витрати (10%)				70,9
Всього (В_{пк})				770,9

Калькуляцію планової собівартості розробленого виробу розраховуємо з використанням методу питомих ваг і структури собівартості аналогічної продукції (див. табл. 2.1).

Тому що, проєктований виріб відноситься до радіоелектронної апаратури, то:

Питома вага матеріалу $\rightarrow \alpha_m = 20\%$;

Питома вага покупних виробів $\rightarrow \alpha_{пк} = 62\%$

Питома вага основної заробітної плати $\rightarrow \alpha_{озп} = 18\%$

Таблиця 2.2 Калькуляція планової собівартості

Найменування статті витрат	Значення статті, грн.	Розрахунок
1. Сировина і матеріал	248,67	$V_m = \alpha_m * V_{пк} / \alpha_{пк}$ $V_m = 20 * 770,9 / 62$
2. Комплектуючі вироби і покупні напівфабрикати	770,9	$V_{пк} = \text{см. табл. 2.1}$
3. Основна заробітна плата	223,80	$V_{оз} = \alpha_{озп} * V_{пк} / \alpha_{пк}$ $V_{оз} = 18 * 770,9 / 62$
4. Додаткова заробітна плата	89,52	$V_{дз} = 0,4 * V_{оз}$ $V_{дз} = 0,4 * 223,80$
5. Відрахування до єдиного соцфонду	68,93	$V_{ес} = (V_{оз} + V_{дз}) * 0,22$ $V_{ес} = (223,80 + 89,52) * 0,22$
6. Загально-виробничі витрати	290,94	$V_{заг.вир} = (1,2 \dots 1,5) * V_{оз}$ $V_{заг.вир} = 1,3 * 223,80$
7. Виробнича собівартість	1 692,76	$C_{вир} = V_m + V_{пк} + V_{оз} + V_{дз} + V_{ес} + V_{заг.вир}$
8. Адміністративні витрати	67,14	$V_a = V_{оз} * 0,3$ $V_a = 223,80 * 0,3$
9. Витрати на збут	33,85	$V_{зб} = C_{вир} * 0,02$ $V_{зб} = 1692,76 * 0,02$
10. Інші операційні витрати	16,92	$V_{оп} = C_{вир} * 0,01$ $V_{оп} = 1692,76 * 0,01$
Повна собівартість	1 810,68	$C_{пов.} = C_{вир} + V_a + V_{зб} + V_{оп}$ $C_{пов.} = 1\,692,76 + 67,14 + 33,8552 + 16,9276$

Розмір планового прибутку, що включається в ціну, визначаємо по формулі:

$$\Pi = (C_{\text{пов}} * p) / 100\%$$

$$\Pi = (1810,68 * 10\%) / 100\% = 181,06,$$

де

p-планова рентабельність продукції (10%...30%).

Оптову ціну виробу визначаємо по формулі:

$$C_o = C_{\text{пов}} + \Pi$$

$$C_o = 1810,68 + 181,06 = 1991,74$$

Ціну реалізації виробу встановлюємо з урахуванням ПДВ:

$$C_p = C_o + \Pi_3,$$

де

Π_3 - податкове зобов'язання з ПДВ:

$$\Pi_3 = C_o * 0,2.$$

$$\Pi_3 = 1\,991,74 * 0,2 = 398,34$$

Звідси:

$$C_p = 1991,74 + 398,34 = 2\,390,08$$

2.3 Прогноз обсягів продаж даного виробу

Отримана в таблиці 2.2 повна собівартість являє собою витрати виготовлення (Спк) одиниці виробу для даного року виробництва. Запропонуємо прогноз обсягів продажів даного виробу на другій стадії життєвого циклу виробу «Виробництво» з розподілом по роках (прогноз продажів передбачаємо на 4 роки).

Характерні зони промислового випуску виробу представлені на рисунку 2.1.

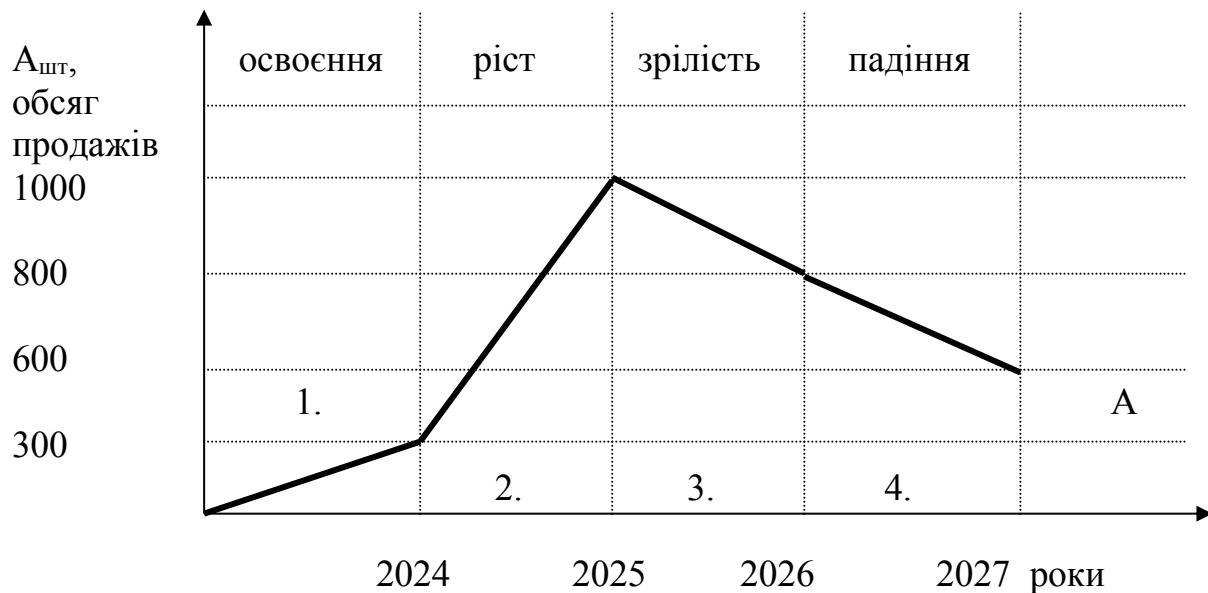


Рисунок 2.1. Зони промислового випуску виробу

В 2024 році обсяг продажів передбачається в розмірі 300 шт під замовлення.

В наступному році прогнозується збільшення обсягу продажів, тому витрати виробництва визначаємо по формулі:

$$C_{\text{пов } i+1} = C_{\text{пов } i} \left(\frac{A_i}{A_{i-1}} \right)^{0,23},$$

де

A_i – обсяг продажів (виробництва) у 1 рік розрахункового періоду, шт.;

A_{i+1} – обсяг продажів (I+1)-ом року, шт.;

0,23 – показник ступеня, що характеризує вплив росту обсягів виробництва на собівартість продукції.

Звідси випливає, що

$$C_{\text{пов } 2025} = 1810,68 \cdot (300/1000)^{0,23} = 1372,71 \text{ грн.}$$

При відсутності росту обсягів виробництва, тобто якщо обсяг продажів або не змінюється або зменшується в наступному році, витрати виробництва приймаються на рівне попереднього року.

$$C_{\text{пов } 2025} = 1372,71 \text{ грн.}$$

Плановий прибуток, що включається в оптову ціну підприємства, для наступного року при збільшенні обсягу продажів, визначаємо по формулі:

$$P_{i+1} = C_{ni+1} * \frac{\rho}{100}$$

Звідси:

$$P_{2025} = 1372,71 * 10 / 100 = 137,27 \text{ грн.}$$

Оптову ціну підприємства в наступні роки розрахункового періоду визначаємо по формулі:

$$C_{O_{i+1}} = C_{ni+1} + P_{i+1}$$

Звідси:

$$C_{2025} = 1372,71 + 137,27 = 1509,98 \text{ грн.}$$

Податкове зобов'язання визначається по формулі:

$$Pz_{i+1} = C_{O_{i+1}} * 0,2$$

Звідси:

$$Pz_{2025} = 1509,98 * 0,2 = 302 \text{ грн.}$$

Ціну реалізації одиниці продукції в наступні роки визначаємо по формулі:

$$C_{P_{i+1}} = C_{O_{i+1}} + Pz_{i+1}$$

Звідси:

$$C_{p2025} = 1509,98 + 458,05 = 1968.03 \text{ грн.}$$

Вартісну оцінку результатів за розрахунковий період (Рт) визначаємо по формулі:

$$P_T = \sum_{i=t_p}^{t_k} A_i * C_{P_i} * \alpha_i$$

де

t_p, t_k – відповідно розрахунковий і кінцевий рік розрахункового періоду;

C_{pi} – ціна реалізації в і-тім році, грн.;

A_i – обсяг продажів у і-тім році, грн.;

α_i – коефіцієнт, що включає фактор часу, тобто коефіцієнт приведення різночасних витрат і результатів до розрахункового року.

Коефіцієнт α_i визначаємо по формулі:

$$\alpha_i = |1 + E_H|^{t_p - t_i},$$

де

					КС 57. 03 002. 00 ДП ПЗ	Арк.
						58
Ізм.	Лист	№ докум.	Підпис	Дата		

E_n – норматив ефективності капітальних вкладень, $E_n = 0,1$;

t_p – розрахунковий рік розрахункового періоду;

t_i – i -й рік розрахункового періоду, витрати і результати якого приводяться до розрахункового року.

Вартісну оцінку за розрахунковий період визначаємо по формі, приведеної в таблиці 2.3

Таблиця 2.3. Розрахунок вартісної оцінки результатів.

Найменування показника	Позначення	Розрахунок виробничого періоду				Всього
		1-й	2-й	3-й	4-й	
Обсяг продажів, шт	A_i	300	1000	800	600	2700
Ціна реалізації, грн.	C_{pi}	2 390,08	1968.03	1968.03	1968.03	-
Вартісна оцінка результатів, грн.	$A_i * C_{pi}$	717024	1968030	1574424	1180818	-
Коефіцієнт, що враховує фактор часу	α_i	0,98	0,86	0,75	0,69	-
Вартісна оцінка результатів з урахуванням фактора часу, грн.	$A_i * C_{pi} * \alpha_i$	702683,52	1692505,8	1180818	814764.42	4390771.74

Виробництво дає змогу одержати дохід за 4 роки 4390.77 тис. грн.

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

3.1 Вступ

Закон України "Про охорону праці" регулює питання безпеки та гігієни праці на всіх підприємствах, установах і організаціях, незалежно від форм власності та видів діяльності. Він покладає на власника або уповноважені ним органи обов'язок забезпечувати належні умови праці, полегшувати їх, оздоровлювати навколишнє середовище та дбати про дотримання правил безпеки та інструкцій з техніки безпеки.

Координацію цієї діяльності здійснює служба охорони праці підприємства. Завдання та функції служби охорони праці викладені в Типовому положенні про службу охорони праці, затвердженому наказом Держнаглядохоронпраці (ДНАОП 0.00-4.21-93).

Працівники також несуть відповідальність за дотримання вимог охорони праці, визначених нормативними документами. У сучасних умовах кожному працівнику необхідно постійно підтримувати високий фізичний, психологічний та професійний рівень, запобігати виникненню випадків травматизму та професійних захворювань.

У розділі охорона праці дипломного проєкту розглянуті питання безпеки та гігієни праці на підприємствах, пов'язані з роботою з комп'ютерною технікою. Оскільки під час розробки програмного забезпечення, веб-додатків, пристроїв розробники (програмісти) проводять значну частину свого часу за комп'ютером, важливо приділити увагу створенню безпечних та комфортних умов праці для них.

3.2 Аналіз небезпечних та шкідливих чинників, що впливають на працівника

Шкідливі фактори під час роботи за комп'ютером. На сьогодні виявлено прямий зв'язок між використанням ПК та низкою захворювань, таких як погіршення зору, болі в спині і шиї, болі в кистях, ліктях та плечах, порушення

					КС 57. 03 003. 00 ДП ПЗ	Арк.
						60
Ізм.	Лист	№ докум.	Підпис	Дата		

сну, хронічні головні болі, нудота, слабкість, стресові стани, захворювання шкіри, вроджені аномалії, провокування епілептичних нападів, інсульту та інші захворювання. Виникли також дві нові хвороби: "синдром комп'ютерного зору" та "інтернет-синдром".

Основні симптоми синдрому комп'ютерного зору включають втому очей, двоїння в очах (диплопія), порушення кольоросприйняття, слезотечу. Інтернет-синдром характеризується сильною залежністю та втратою контролю над своїми діями при тривалій роботі за комп'ютером. В дипломному проєкті враховано працю користувача ПК, тому умови праці та забезпечення безпеки його роботи здійснюються відповідно до всіх норм і вимог безпечного використання комп'ютерів.

3.3 Розробка заходів з охорони праці

Трудова діяльність користувачів комп'ютерів відбувається в певному виробничому середовищі, яке безпосередньо впливає на їхній функціональний стан. Найбільш значущими факторами виробничого середовища, що впливають на них, є фізичні фактори, такі як електромагнітні хвилі різних частотних діапазонів, електростатичні поля, шум, параметри мікроклімату та широкий спектр світлотехнічних показників. Вплив хімічних та, особливо, біологічних факторів виробничого середовища на користувачів комп'ютерів є значно меншим.

Сприятливі санітарні умови створюються шляхом забезпечення працівників зручними робочими місцями, повітрям з нормованими параметрами, належною освітленістю, захистом від шуму та вібрацій, захистом від дії шкідливих речовин та випромінювань, робочим одягом та різними засобами індивідуального захисту, а також наявністю побутових приміщень та спеціальних служб, призначених для створення безпечних та нормальних умов праці

					<i>КС 57. 03 003. 00 ДП ПЗ</i>	Арк.
						61
Ізм.	Лист	№ докум.	Підпис	Дата		

3.3.1 Гігієнічне нормування параметрів повітря робочої зони

Значущим фактором, що впливає на продуктивність і безпеку праці, є виробничий мікроклімат. Цей мікроклімат визначається такими параметрами, як температура і вологість повітря, швидкість руху повітряних мас, а також інтенсивність радіаційного впливу. Всі ці показники повинні відповідати вимогам ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень що наведено в таблиці 3.1. Висока температура в поєднанні з високою вологістю повітря значно впливає на працездатність оператора. За таких умов різко збільшується час сенсомоторних реакцій, порушується координація рухів і зростає кількість помилок.

Таблиця 3.1. Вимоги до параметрів мікроклімату у виробничому приміщенні

Параметри мікроклімату	Значення параметрів	
	Узимку	Улітку
1. Температура, °C	22-24	23-25
2. Швидкість повітряних мас, м/с	0.1	0.1-0.2
3. Відносна вологість, %	40-60	40-60

Висока температура не тільки негативно впливає на фізичні аспекти працездатності, але й має суттєвий вплив на психологічні функції людини. Зокрема, зменшується обсяг запам'ятовуваної інформації, знижується здатність до асоціацій, погіршується якість асоціативних і рахункових операцій, а також знижується рівень уваги. Відносна вологість у межах 40-60% незначно впливає на стан людини, але при вологості 95-100% механізм регуляції потовиділення майже повністю вимикається, що швидко призводить до перегрівання організму.

Для підтримки необхідної температури й вологості в робочих приміщеннях використовуються системи опалення й кондиціонування. Ці системи забезпечують постійне й рівномірне нагрівання, циркуляцію, а також очищення повітря від пилу й шкідливих речовин. Важливою складовою є також вентиляція, яка забезпечує підтримання гігієнічного складу повітря, видалення шкідливих

газів, пари та пилу. Вентиляція – це регульований повітрообмін у приміщенні, і вона може бути як природною, так і штучною.

Природна вентиляція підрозділяється на аерацію й провітрювання. Провітрювання здійснюється за рахунок відкриття вікон і дверей для забезпечення природного потоку повітря. Механічна вентиляція використовує вентилятори, які постачають повітря з чистих зон до робочих місць і видаляють забруднене повітря. При цьому повітря може піддаватися різній обробці: підігріву, зволоженню, осушенню, очищенню від пилу тощо.

3.3.2 Освітлення

У кожному приміщенні, де розташовуються робочі місця для співробітників, які працюють за комп'ютером, повинні бути присутні як природні, так і штучні джерела освітлення. Штучне освітлення у кімнатах з відеодисплейними терміналами (ВДТ) слід організовувати у вигляді комбінованої системи з використанням люмінесцентних ламп у світильниках загального освітлення. На робочих місцях повинна бути забезпечена рівномірна освітленість за допомогою переважно відбитого або розсіяного світлорозподілу. Світлові відблиски з клавіатури, екрана та інших частин ВДТ, що потрапляють в очі користувача, мають бути відсутні.

Природне та штучне освітлення повинні бути збалансовані для мінімізації відблисків на екрані та клавіатурі. Штучне освітлення повинно включати комбінацію загального та локального освітлення з використанням люмінесцентних ламп, що забезпечують рівномірне розподілення світла. Це дозволить створити комфортні умови для зорової роботи та знизити зорову напругу. Норма освітленості на робочих місцях становить 300-500 люксів.

3.3.3 Шум

В робочих приміщеннях компанії основні джерела акустичних шумів – це персональні електронно-обчислювальні машини (ПЕОМ). Ці машини також створюють шуми електромагнітного походження через коливання елементів електромеханічних пристроїв під впливом змінних магнітних полів. Крім того, у

					<i>КС 57. 03 003. 00 ДП ПЗ</i>	Арк.
						63
Ізм.	Лист	№ докум.	Підпис	Дата		

приміщеннях виникає структурний шум, який випромінюється поверхнями коливних конструкцій стін, перекриттів і перегородок будівлі у звуковому діапазоні частот. Постійний шум може призводити до втоми слуху, зниження здатності до звукового сприйняття, а також значної втоми всього організму.

Для зменшення або усунення негативного впливу шумів, доцільно ізолювати робочі приміщення, розміщуючи їх у частинах будинку, віддалених від міського шуму, з вікнами, що виходять у двір. Також важливо перевіряти герметичність корпусів комп'ютерів і своєчасно змінювати вентилятори охолодження. Шум також знижується завдяки зеленим насадженням, які поглинають звуки.

Допустимий рівень шуму при розумовій праці, що вимагає зосередженості, для програміста складає 50 дБ. Для зменшення шуму та вібрації в приміщенні устаткування, апарати й прилади встановлюються на спеціальні фундаменти й амортизаційні прокладки. Якщо стіни та стелі приміщення є джерелами шуму, вони повинні бути облицьовані звукопоглинаючим матеріалом.

3.3.4 Електробезпека

Основними причинами електротравматизму є:

- присутність напруги, навіть якщо вона відключена;
- несподіване виникнення напруги через пошкодження ізоляції в місцях, де в нормальних умовах її не повинно бути;
- контакт струмопровідного обладнання з проводом, що знаходиться під напругою.

Для попередження уражень електричним струмом необхідно строго та повністю дотримуватись правил проведення робіт і технічної експлуатації. Слід виключити можливість доступу оператора до частин обладнання, що працюють під небезпечною напругою, до неізольованих частин, призначених для роботи при малій напрузі, і не підключених до захисного заземлення. Також важливо підключати електроживлення до ПЕОМ через розетку за допомогою спеціальної вилки із заземлюючим контактом.

Електромережі, електроприлади та апаратура повинні використовуватися тільки в справному стані, з дотриманням рекомендацій виробників.

3.3.5 Організація робочого місця

Облаштування та організація робочого місця з використанням відеодисплейних терміналів (ВДТ) повинні забезпечувати відповідність усіх конструкційних елементів робочого місця та їх взаємне розташування ергономічним вимогам, враховуючи характер і особливості трудової діяльності, як зазначено в ДСанПіН 3.3.2.-007-98. Конструкція робочого місця і його елементів (сидіння, органи керування, засоби відображення інформації) повинна враховувати антропометричні, фізіологічні та психологічні вимоги, а також характер роботи. Робочі меблі мають бути адаптовані для індивідуального налаштування, що дозволяє забезпечити зручну і правильну позу працівника.

Робочий стіл повинен бути пофарбований матовою фарбою, щоб уникнути відблисків і підвищити комфорт зорової роботи. Дисплей слід розташовувати таким чином, щоб верхній край екрана був на рівні очей або трохи нижче, на відстані приблизно 70 см, що відповідає рекомендованому діапазону від 60 до 90 см. Частота оновлення екрана повинна бути не менше 100 Гц, що перевищує мінімальні вимоги у 70 Гц, для зменшення зорової втоми і мерехтіння.

Для зниження нервово-емоційного напруження та запобігання втомі рекомендується регулярно виконувати комплекс вправ, спрямованих на поліпшення мозкового кровообігу та подолання негативних наслідків гіподинамії. Ці вправи викладені у Державних санітарних правилах і нормах роботи з візуальними терміналами електронно-обчислювальних машин (ДСанПіН 3.3.2.007-98).

3.4 Пожежна безпека

Пожежна небезпека – це ймовірність виникнення або розвитку пожежі в будь-якій речовині, процесі чи стані. Пожежі завжди несуть загрозу, якщо не життю та здоров'ю людей, то значними матеріальними збитками.

					<i>КС 57. 03 003. 00 ДП ПЗ</i>	Арк.
						65
Ізм.	Лист	№ докум.	Підпис	Дата		

Меблі та обладнання повинні бути розташовані так, щоб забезпечувати вільний прохід для евакуації до вихідних дверей, ширина якого має бути не менше 1 метра. Евакуаційні шляхи і виходи повинні бути завжди вільними і не зашарашеними. У разі виявлення пошкоджень електромереж, вимикачів, розеток або інших електропристроїв необхідно негайно вимкнути їх і вжити заходів для забезпечення пожежної безпеки. Документи, папір та інші горючі матеріали повинні зберігатися на відстані не менше 1 метра від електрощитів, 0,5 метра від електросвітильників, 0,6 метра від пожежних сповіщувачів та 0,15 метра від приладів центрального опалення. Відстань від найдальшої точки приміщення до вогнегасника не повинна перевищувати 20 метрів.

На рисунку 3.1 зображено первинні засоби пожежогасіння.



Рисунок 3.1. Первинні засоби пожежогасіння

В службових приміщеннях забороняється використовувати тимчасові електромережі, прокладати електричні проводи безпосередньо на горючій основі, експлуатувати електроприлади з механічними пошкодженнями, зашарашувати підходи до засобів пожежогасіння, курити або використовувати легкозаймисті рідини, виконувати вогневі та зварювальні роботи без спеціального дозволу, вмикати електронагрівальні прилади без негорючих підставок або в місцях, де це заборонено, та зашарашувати евакуаційні шляхи та виходи.

В усіх офісних приміщеннях біля телефонів повинні бути таблички із номером телефону пожежної служби. Якщо на поверсі працює більше 25 осіб, потрібно розробити та вивісити плани евакуації. Якщо на поверсі більше 50 осіб,

крім плану евакуації, повинна бути розроблена інструкція з дій персоналу для безпечної та швидкої евакуації.

Нар рисунку 3.2 зображено основні дії при пожежі.



Рисунок 3.2 Основні дії при пожежі

План евакуації включає в себе схематичне зображення будівлі або поверху з чітким зазначенням шляхів евакуації, місць розташування пожежного обладнання, таких як вогнегасники, пожежні крани, а також місць збору після евакуації. Цей план повинен бути розроблений з урахуванням специфіки будівлі та кількості людей, що в ній перебувають

Нар рисунку 3.3 зображено план евакуації.



Рисунок 3.3. План евакуації

ВИСНОВКИ

У дипломному проєкті створено пристрій для відображення повітряних тривог, що дозволяє ефективно інформувати користувачів про небезпеку в режимі реального часу. У створеному пристрої реалізований функціонал відображення тривог за допомогою світлодіодів, що відповідають різним областям України, а також видача звукового сигналу бужером для області, де знаходиться користувач, при оголошенні тривоги. Пристрій підключається до Інтернету через Wi-Fi для отримання актуальних даних про тривоги, обробляє ці дані та керує світлодіодами і бужером на їх основі.

Для реалізації проєкту було задіяно мікроконтролер ESP32, який відповідає за обробку даних і керування світлодіодами та бужером. Програмне забезпечення написано на мові MicroPython, що дозволяє легко інтегруватися з різними периферійними пристроями. Середовище розробки Thonny IDE використовувалося для написання та налагодження коду, що забезпечило швидку та ефективну роботу над проєктом. Дані про повітряні тривоги отримувалися через API або спеціальні сервіси, що надають таку інформацію.

В пояснювальній записці розглянуті всі питання, передбачені технічним завданням на дипломне проєктування. Проведено аналіз предметної області, детально описано технології та засоби, які використовувалися при створенні проєкту, проведено розрахунок економічної ефективності створення та впровадження пристрою, а також розглянуті питання охорони праці та наведений перелік використаних джерел.

Розробка пристрою для відображення повітряних тривог з використанням ESP32 і MicroPython дозволила створити надійне та ефективне рішення для інформування про небезпеку. Це сприятиме підвищенню поінформованості та безпеки користувачів, забезпечуючи швидку реакцію на повітряні тривоги. Застосовані технології та інструменти довели свою ефективність і практичність, що дозволяє рекомендувати їх для подібних проєктів у майбутньому.

					КС 57. 03 000. 00 ДП ПЗ	Арк.
						68
Ізм.	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Мапа тривоги України [Веб-сайт]. URL: <https://alerts.in.ua/>.
2. Застосунок «Повітряна тривога». [Веб-сайт]. URL: <https://www.ukrainealarm.com/>.
3. Персональный оповещатель о воздушных тревогах на ESP8266. [Веб-сайт]. URL: <https://www.youtube.com/watch?v=G29C8wKiC1o&t=75s>
4. Что такое Arduino? [Веб-сайт]. URL: <https://doc.arduino.ua/ru/about/>.
5. Arduino Uno. [Веб-сайт]. URL: <https://components101.com/microcontrollers/arduino-uno>.
6. Raspberry Pi [Веб-сайт]. URL: <https://3d-diy.ru/wiki/arduino-platy/obzor-platy-raspberry-pi/>.
7. ESP32 Series. Datasheet [Веб-сайт]. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
8. ESP32. Technical Reference Manual [Веб-сайт]. URL: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf.
9. Адресные светодиоды. WS2812B, WS2812S [Веб-сайт]. URL: <https://3d-diy.ru/wiki/components/adresnye-svetodiody-ws2812b-ws2812s/>.
10. Пассивный зумер, пьезоэлектрический, 3-12 В, для Arduino та інших мікроконтролерів [Веб-сайт]. URL: <https://arduinokit.com.ua/ua/p1592114002-passivnyj-zummer-pezoelektricheskij.html>.
11. Модуль з динаміком KY-006 (пасивний) [Веб-сайт]. URL: <https://arduino.ua/prod1153-modul-s-dinamikom-ky-006-passivnyi>.
12. АКУМУЛЯТОРИ 18650 - ДЛЯ ЧОГО ВОНИ ПОТРІБНІ [Веб-сайт]. URL: <https://iq-fishing.com.ua/ua/akkumulyatory-18650-dlia-cheho-ony-nuzhny/>.
13. MT3608 Підвищувач регульований перетворювач DC-DC [Веб-сайт]. URL: <https://myproject.com.ua/mt3608-pidvischujuchij-regulovaniy-dc-dc-modul-ua.html>.
14. Контролер заряду Li-Ion акумуляторів на базі TP4056 Type-C [Веб-сайт]. URL: <https://www.mini-tech.com.ua/ua/modul-zaryada-li-lon-tp4056-type-c>.

15. Getting Started with Thonny MicroPython (Python) IDE for ESP32 and ESP8266 [Веб-сайт]. URL: <https://randomnerdtutorials.com/getting-started-thonny-micropython-python-ide-esp32-esp8266/>.
16. Thonny [Веб-сайт]. URL: <https://thonny.org/>.
17. Активний зуммер [Веб-сайт]. URL: <https://electromicro.ru/resources/wiki/arduino-lessons/arduino-lessons-zummer-active/>.
18. П'єзоелектричний пасивний зумер 1407 [Веб-сайт]. URL: <https://myproject.com.ua/pzoelektrichnij-pasivnij-zumer-1407-ua.html>.
19. Азаров О. Д., Гарнага В. А., Клятченко Я. М., Тарасенко В. П. Комп'ютерна схемотехніка: підручник. – Вінниця: ВНТУ, 2018.
20. Дейбук В.Г., Деревянчук О.В., Г.О. Кравченко. Віртуальний електронний практикум: Навчальний посібник. – Чернівці: Чернівецький національний університет, 2021.
21. В.В. Макаренко. Вступ до техніки вимірювань: Навчальний посібник. – Київ: КПІ ім. Ігоря Сікорського, 2021.
22. Сторчак К.П., Тушич А.М., Срібна І.М., Яковенко Н.Д., Кравець Д.В. Технології інтернет речей: Навчальний посібник. – Київ: Державний університет телекомунікацій навчально-науковий інститут інформаційних технологій, 2021.

ДОДАТОК А. Програмний код основної логіки веб-застосунку

```
import network
import socket
from time import sleep, time
import esp
import gc
import machine
import json
import _thread
from neopixel import NeoPixel
import urequests
from machine import Pin, PWM
esp.osdebug(None)
gc.collect()

station = network.WLAN(network.STA_IF)
if station.isconnected():
    station.active(False)

PIN = 12
NUMPIXELS = 25

states = ["Закарпатська область", "Івано-Франківська область", "Тернопільська область",
"Львівська область", "Волинська область", "Рівненська область", "Житомирська область",
"Київська область", "Чернігівська область", "Сумська область", "Харківська область",
"Луганська область", "Донецька область", "Запорізька область", "Херсонська область",
"Автономна Республіка Крим", "Одеська область", "Миколаївська область",
"Дніпропетровська область", "Полтавська область", "Черкаська область", "Кіровоградська
область", "Вінницька область", "Хмельницька область", "Чернівецька область"]

pin = Pin(PIN, Pin.OUT)
np = NeoPixel(pin, NUMPIXELS)

def turn_off_leds():
    for i in range(NUMPIXELS):
        np[i] = (0, 0, 0)
    np.write()

turn_off_leds()

led = machine.Pin(2, machine.Pin.OUT)

wifi_config_file = 'config.json'

def load_wifi_config():
    try:
        with open(wifi_config_file, 'r') as f:
            config = json.load(f)
            return config['ssid'], config['password']
    except OSError:
        return None, None

def save_wifi_config(ssid, password):
    config = {'ssid': ssid, 'password': password}
    with open(wifi_config_file, 'w') as f:
        json.dump(config, f)
```



```

def blink_red_error():
    station = network.WLAN(network.STA_IF)
    while not station.isconnected():

        for i in range(25):
            np[i] = (125, 0, 0)
            np.write()
            sleep(0.5)
        for i in range(25):
            np[i] = (0, 0, 0)
            np.write()
            sleep(0.5)

def reset_wifi_config(pin):
    print("Кнопка сброса нажата, сброс настроек WiFi...")
    try:
        import os
        os.remove(wifi_config_file)
    except OSError:
        pass
    machine.reset()

ssid, password = load_wifi_config()

if ssid and password:
    station = network.WLAN(network.STA_IF)
    station.active(True)
    while not station.isconnected():
        try:
            station.connect(ssid, password)
            sleep(5)
        except:
            _thread.start_new_thread(blink_red_error, ())
            sleep(10)

    connected = True
else:

    ap = network.WLAN(network.AP_IF)
    ap.active(True)
    ap.config(essid='ESP32-Setup', password='12345678')

    ip = ap.ifconfig()[0]
    print('Точка доступа запущена. IP адрес:', ip)
    connected = False

def blink_led():
    while True:
        station = network.WLAN(network.STA_IF)
        if not station.isconnected():
            led.value(not led.value())
            sleep(0.5)
        else:
            led.value(1)
            break

_thread.start_new_thread(blink_led, ())

```

```

html_template = """<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">

        <title>Настройка WiFi ESP32</title>
    </head>
    <body>
        <h1>Настройка WiFi</h1>
        <form action="/configure" method="post">
            <label for="ssid">SSID:</label>
            <select id="ssid" name="ssid">
                {options}
            </select><br><br>
            <label for="password">Пароль:</label>
            <input type="password" id="password" name="password"><br><br>
            <input type="submit" value="Отправить">
        </form>
    </body>
</html>
"""

def scan_networks():
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    networks = wlan.scan()
    options = ''
    for ssid, bssid, channel, RSSI, authmode, hidden in networks:
        options += f'<option value="{ssid.decode()}">{ssid.decode()}</option>'
    return options

ssid_options = scan_networks()

addr = socket.getaddrinfo('0.0.0.0', 80)[0][-1]
s = socket.socket()
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
s.bind(addr)
s.listen(1)

def connect_to_wifi(ssid, password):
    station = network.WLAN(network.STA_IF)
    station.active(True)
    station.connect(ssid, password)

    start_time = time()
    timeout = 5

    while not station.isconnected():
        if time() - start_time > timeout:
            print('Подключение не удалось')
            station.active(False)
            return False
        sleep(1)
    print('Успешное подключение')
    save_wifi_config(ssid, password)
    return True

def connect_api():
    print("Подключение к API...")
    api_url = "http://alerts.net.ua/alerts_statuses_v3.json"

```

```

try:
    response = urequests.get(api_url)
    if response.status_code == 200:
        data = response.json()
        for i, oblast in enumerate(states):
            status = data["states"].get(oblast, False)

            if status:
                print(f"{oblast}: {status}")
                np[i] = (125, 0, 0) if status else (0, 125, 0)
                np.write()
                sleep(0.1)
            else:
                print(f"Ошибка при запросе к API: {response.status_code}")
except:
    for i in range(10):
        for i in range(25):
            np[i] = (125, 0, 0)
            np.write()
            sleep(0.5)
        for i in range(25):
            np[i] = (0, 0, 0)
            np.write()
            sleep(0.5)

while True:
    if connected:
        connect_api()
        sleep(15)

    if not connected:
        try:
            cl, addr = s.accept()
            request = cl.recv(1024)
            request = request.decode('utf-8')

            if 'POST /configure' in request:
                _, data = request.split('\r\n\r\n')
                params = data.split('&')
                ssid = params[0].split('=')[1]
                password = params[1].split('=')[1]

                if connect_to_wifi(ssid, password):
                    response = "HTTP/1.1 200 OK\r\nContent-Type: text/html; charset=UTF-
8\r\n\r\n"
                    response += "<html><body><h1>Подключение к WiFi
успешно!</h1><p>Отключаю точку доступа...</p></body></html>"

                    cl.send(response.encode('utf-8'))
                    cl.close()

                    sleep(3)
                    ap.active(False)

                else:
                    response = "HTTP/1.1 200 OK\r\nContent-Type: text/html; charset=UTF-
8\r\n\r\n"

```

```

        response += f"<html><body><h1>Не удалось подключиться. Попробуйте
еще раз.</h1><p>Перейдите по адресу <a
href='http://{ip}'>http://{ip}</a></p></body></html>"
    else:
        network.WLAN(network.STA_IF).active(False)
        ap.active(True)

    response = "HTTP/1.1 200 OK\r\nContent-Type: text/html; charset=UTF-
8\r\n\r\n"

    response += html_template.format(options=ssid_options)

    cl.send(response.encode('utf-8'))
    cl.close()
except Exception as e:
    print('Ошибка:', e)
try:
    cl.close()
except Exception as e:
    print('Ошибка при закрытии соединения:', e);

```

Дипломний проєкт на тему: «Розробка фізичної карти повітряних тривог»

Виконав: Герганов В.П.
Керівник: Кіреєв І. А.

Основні відомості про проєкт

Проєкт полягає у створенні фізичного пристрою для відображення повітряних тривог в режимі реального часу.

Мета пристрою – підвищити поінформованість та безпеку користувачів під час загроз, забезпечуючи швидке інформування про повітряні тривоги. Пристрій відображає тривоги за допомогою світлодіодів, що відповідають різним областям України, та видає звуковий сигнал зумером для області, де знаходиться користувач, при оголошенні тривоги



Вибір елементної бази



Світлодіодна стрічка WS2812B



Плата зарядки TP4056



Підвищуючий DC/DC перетворювач
MT3608



Акумулятор 18650 Samsung 32e

Вибір елементної бази



Пасивний зумер



Перемикач



Мікроконтролер ESP32



Тактова кнопка

Вибір мікроконтролера

Для проєкту було обрано мікроконтролер ESP32, який є потужним і гнучким рішенням для реалізації IoT-проєктів. ESP32 має вбудований Wi-Fi і Bluetooth, що дозволяє легко підключатися до мережі та отримувати актуальні дані про повітряні тривоги в режимі реального часу.

ESP32 підтримує програмування на мові MicroPython, що забезпечує зручність і простоту написання коду. Основні переваги ESP32 включають високу продуктивність при низькому енергоспоживанні, можливість роботи з різними периферійними пристроями, компактні розміри та вбудований Wi-Fi модуль.

Завдяки відкритій екосистемі з великою кількістю бібліотек та прикладів, ESP32 забезпечує надійність та ефективність, необхідні для реалізації проєкту з відображення повітряних тривог.



Структурна схема карти повітряних тривог

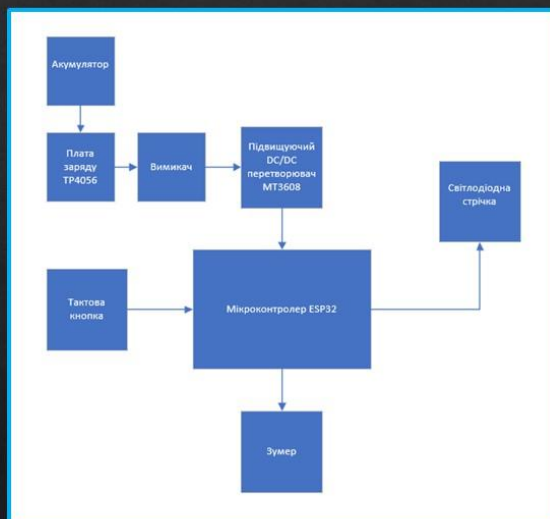


Схема підключення компонентів до ESP32

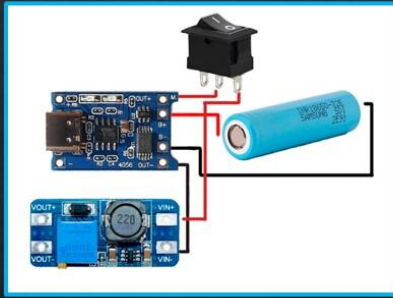


Схема підключення акумулятора до плати заряду, DC/DC перетворювача, і перемикача

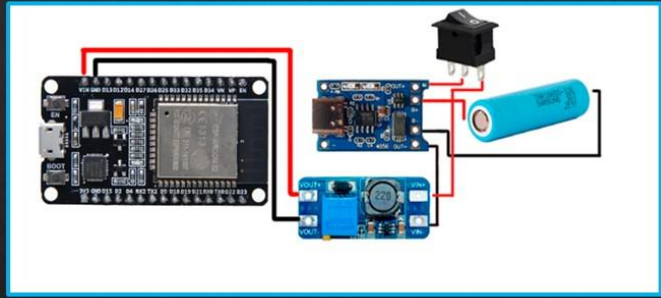


Схема підключення акумулятора с минулої схеми до плати ESP32

Схема підключення компонентів до ESP32

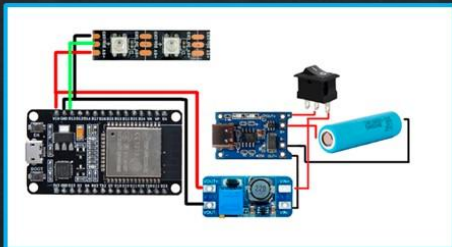


Схема підключення світлодіодної стрічки до плати ESP32

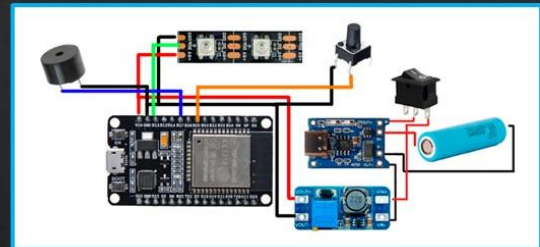


Схема підключення кнопки та зумера до плати ESP32

Підключення до Wi-Fi через вебсервер

Для зручності підключення пристрою до Wi-Fi, використовується вебсервер, який запускається при першому включенні пристрою. Спочатку ESP32 запускає Wi-Fi точку доступу, користувач підключається до неї, далі через будь-який браузер треба зайти на адресу 192.168.4.1 та ввести необхідні дані для підключення до Wi-Fi мережі.



Отримання даних про повітряні тривоги

Для отримання актуальних даних про повітряні тривоги використовується API alerts.net.ua. Код написано на мові програмування MicroPython. Після підключення до Wi-Fi мікроконтролер відправляє HTTP-запит до API і отримує дані про поточні повітряні тривоги. Якщо відповідь успішна, статус кожної області перевіряється і оновлюється стан світлодіодів: якщо в області тривога – вмикається червоний світлодіод, якщо тривоги немає - зелений

```
def connect_api():
    print("Підключення к API...")
    api_url = "http://alerts.net.ua/alerts_statuses_v3.json"
    # Відправка запроса
    try:
        response = urequests.get(api_url)
        if response.status_code == 200:
            data = response.json()
            # Перевірка статусу областей
            for i, oblast in enumerate(states):
                status = data["states"].get(oblast, False)
                if status:
                    print(f"{oblast}: {status}")
                    np[i] = (125, 0, 0) if status else (0, 125, 0)
                    np.write()
                    sleep(0.1)
            else:
                print(f"Помилка при запиті до API: {response.status_code}")
```

Тестування пристрою



Режим 1 – Тривоги



Режим 2 – прапор України

Дякую за увагу!

ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Герганова Владислава Пилиповича

(прізвище, ім'я та по батькові)

Спеціальність: _____

Освітня програма: *123 «Комп'ютерна інженерія»*

«Обслуговування комп'ютерних систем та мереж»

Тема дипломного проекту: *Розробка фізичної карти повітряних тривог*

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка до дипломного проекту містить 81 сторінок. У пояснювальній записці описано етапи розробки фізичної карти повітряних тривог на базі ESP32. Графічна частина складається з окремих слайдів, оформлених у вигляді презентації, передбачених технічним завданням. Якість виконання пояснювальної та слайдів добра.

б) самостійність роботи над проектом: Протягом виконання дипломного проекту здобувач освіти Герганов Владислав поступово та послідовно виконував всі етапи, проявляв ініціативу в створенні загальної концепції та реалізації роботи. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника.

в) теоретична підготовка випускника (випускниці): Здобувач освіти Герганов Владислав під час роботи над дипломним проектом вивчив достатньо багато літературних та інтернет-джерел за даною тематикою. Вважаю, що теоретична підготовка дипломника достатня і він готовий до захисту проекту.

г) вміння розв'язувати виробничі та конструкторські питання Під час виконання дипломного проекту здобувач освіти Герганов Владислав показів вміння організовано працювати над поставленим завданням, застосовувати знання у галузі електротехніки, схемотехніки та програмування, розробляти, встановлювати та налаштовувати спеціалізоване програмне забезпечення, оформлювати слайди та складати презентації, користуючись сучасними комп'ютерними програмними засобами, такими як Microsoft PowerPoint, Microsoft Visio та ін.

Оцінка розрахункової частини Відмінно
Оцінка графічної частини Добре
Загальна оцінка Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту Кіреєв Ігор Анатолійович

Місце роботи і посада керівника дипломного проекту к.т.н., доцент кафедри «Кібербезпеки та технічного захисту інформації» Державного університету інтелектуальних технологій і зв'язку

Підпис 

«10» 06 2024 р.

РЕЦЕНЗІЯ

на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Герганова Владислава Пилиповича

(прізвище, ім'я та по батькові)

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма «Комп'ютерна графіка і Web-дизайн»

Керівник дипломного проекту (роботи) Кіреєв Ігор Анатолійович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка фізичної карти повітряних тривог

Обсяг розрахунково-пояснювальної записки 81 сторінок

Обсяг графічної (презентаційної) частини 10 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту завданню

Представлений на рецензію дипломний проект відповідає затвердженій темі та виконаний відповідно технічному завданню. Дипломний проект присвячений проблемі особистого оповіщення про повітряні тривоги та складається з пояснювальної записки, додатку з програмним кодом та мультимедійної презентації, що містить приклади роботи програми.

б) характеристика виконання кожного розділу дипломного проекту

Пояснювальна записка складається з основного розділу (аналізу предметної області, проектування пристрою, реалізації пристрою, тестування пристрою), економічного розділу, розділу охорони праці та додатків. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора КТ. Економічний розділ проекту містить розрахунок витрат на НДР та реалізацію проекту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту

Графічна частина складається з 10 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять ілюстративні схеми, скріншоти роботи програмного застосунку, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки добра, розробку виконано у повному обсязі.

г) перелік позитивних якостей дипломного проекту *Реалізовано фізичний пристрій карти повітряних тривог, що базується на сучасній платформі ESP32.*

Корпус пристрою має мінімалістичний дизайн та було роздруковано на 3D принтері. Пристрій використовує адресну світлодіодну стрічку для індикації областей, що знаходяться під загрозою. Також пристрій має порт для зарядки акумулятору.

д) основні недоліки дипломного проекту

Пристрій надає загальну інформацію про статус повітряної тривоги в областях, але додатково не інформує про тип загрози.

Апаратна частина розробки недостатньо обґрунтована у тексті пояснювальної записки

Оцінка розрахункової частини

Відмінно

Оцінка графічної частини

Добре

Загальна оцінка

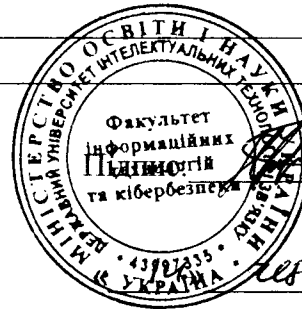
Відмінно

Прізвище, ім'я, по батькові рецензента

Царьов Роман Юрійович

Місце роботи і посада рецензента

Державний університет інтелектуальних технологій і зв'язку, ст. викладач, зав. кафедри комп'ютерної інженерії та інформаційних систем



Сергій 2024 р.

Ім'я користувача:
Катерина Григоріївна Краснокутська

ID перевірки:
1016357726

Дата перевірки:
13.06.2024 18:09:46 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
13.06.2024 18:10:54 EEST

ID користувача:
100011688

Назва документа: 4КС-57_Герганов_В

Кількість сторінок: 49 Кількість слів: 9381 Кількість символів: 67509 Розмір файлу: 3.05 MB ID файлу: 1016162066

7.02% Схожість

Найбільша схожість: 0.71% з Інтернет-джерелом (<http://www.kk.nau.edu.ua/article/2059>)

7.02% Джерела з Інтернету

740

Сторінка 51

Не знайдено джерел з Бібліотеки

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи

3

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Герганов Владислав Пилипович,
здобувач освіти гр. 4КС-57, та

Кірсєв Ігор Анатолійович,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка фізичної карти повітряних тривог» (автор роботи – Герганов В.П., керівник роботи – Кірсєв І.А.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

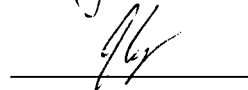
Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Герганов В.П. /

Керівник



/ Кірсєв І.А. /

«10» червня 2024 р.