Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та безпеки

(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

IOT-НАВІГАТОР ДЛЯ ПОШУКУ ЗАГУБЛЕНИХ РЕЧЕЙ

IOT NAVIGATOR FOR FINDING LOST THINGS

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІс-21
Кочергін Богдан Володимирович

(підпис)

Керівник:
к.т.н., доцент
Костючко Сергій Миколайович

(підпис)

Кваліфікаційну роботу
допущено до захисту
« 04 » червня 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *комп'ютерної інженерії та безпеки*
Ступінь вищої освіти: *бакалавр*
Галузь знань: *12 Інформаційні технології*
Спеціальність: *123 Комп'ютерна інженерія*
Освітня програма: *«Комп'ютерна інженерія»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ доц. Т. ТЕРЛЕЦЬКИЙ
« 10 » 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Кочергіну Богдану Володимировичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *IoT-навігатор для пошуку загублених речей*

Керівник роботи *к.т.н., доц. Костючко Сергій Миколайович*

затверджені наказом закладу вищої освіти від «04» січня 2025 року № 11/01-02
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 11.06.2025р.

3. Вихідні дані до роботи *джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування.*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Вступ
Аналіз теми та історія розвитку
Проєктування апаратної складової пристрою
Практична реалізація IoT-навігатора для пошуку загублених речей
Висновки

5. Перелік графічного (ілюстративного) матеріалу:
Існуючі рішення
Використані технології
Архітектура системи
Інтерфейс системи
Схема роботи програмного продукту

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз теми та історія розвитку	*Костючко С.М., доцент*		
Проєктування апаратної складової пристрою	*Костючко С.М., доцент*		
Практична реалізація IoT-навігатора для пошуку загублених речей	*Костючко С.М., доцент*		
Нормоконтроль	*Багнюк Н.В., доцент*		
Гарант ОП	*Лавренчук С.В., доцент*		
Показник запозичень тексту	***___%***		
Академічна доброчесність	*Міскевич О.І.,ст.викладач*		

7. Дата видачі завдання 10.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	*Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень*	до 10.02.2025 р.	Виконано
2.	*Аналіз теми та історія розвитку, проєктування апаратної складової пристрою*	до 02.03.2025 р.	Виконано
3.	*Практична реалізація IoT-навігатора для пошуку загублених речей*	до 02.04.2025 р.	Виконано
4.	*Висновки та пропозиції*	до 10.04.2025 р.	Виконано
5.	*Формування списку використаних джерел*	до 15.04.2025 р.	Виконано
6.	*Формування додатків*	до 02.05.2025 р.	Виконано
7.	*Оформлення ілюстративного матеріалу*	до 10.05.2025 р.	Виконано
8.	*Представлення остаточного варіанту кваліфікаційної роботи керівникові*	до 15.05.2025 р.	Виконано
9.	*Нормоконтроль*	до 30.05.2025 р.	Виконано
10	*Інструментальна перевірка на академічний плагіат*	до 03.06.2025 р	Виконано
11.	*Здача кваліфікаційної роботи та всіх супровідних документів на кафедру*	до 10.06.2025 р.	Виконано

Здобувач вищої освіти _____ (підпис) Кочергін Б.В. (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ (підпис) Костючко С.М. (прізвище, ініціали)

АНОТАЦІЯ

Кочергін Б. В. IoT-навігатор для пошуку загублених речей. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

Метою роботи є розробка компактного та енергоефективного IoT-пристрою для пошуку загублених речей із використанням звукової індикації та можливістю віддаленого налаштування через вебінтерфейс.

Об'єктом дослідження є пристрої Інтернету речей (IoT), призначені для взаємодії з особистими речами у повсякденному житті.

Робота включає аналіз літератури, обґрунтування вибору компонентів, моделювання схеми, програмування мікроконтролера та тестування пристрою.

Перший розділ присвячено аналізу технологій BLE, Wi-Fi та ультразвукових сенсорів, а також актуальності створення трекерів для локалізації предметів.

У другому розділі обґрунтовано вибір апаратного та програмного забезпечення, зокрема мікроконтролера ESP8266, сенсора KY-037, зумера, Arduino IDE та Fritzing. Проведено порівняння з аналогами і визначено доцільність використання вибраних засобів.

У третьому розділі описано архітектуру пристрою, реалізацію апаратної та програмної частини, створення вебінтерфейсу та результати тестування. Система показала стабільну роботу, можливість налаштування чутливості й звукового сигналу, а також автономність при живленні від акумулятора.

Пристрій може бути використаний у повсякденному житті, освітніх проектах або інтегрований у комерційні IoT-рішення для відстеження. Він поєднує в собі доступність, гнучкість та простоту використання.

Ключові слова: інтернет речей, IoT-навігатор, ESP8266, KY-037, вебінтерфейс, мікроконтролер, arduino IDE, пошук речей.

ANNOTATION

Kocherhin B. Iot navigator for finding lost things. Manuscript.

Bachelor's qualification thesis of the Educational Program «Computer Engineering», specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

The aim of the thesis is to develop a compact and energy-efficient IoT device for locating lost items, using sound indication and remote configuration via a web interface.

The object of the research is Internet of Things (IoT) devices designed for everyday interaction with personal items.

The research involves literature analysis, selection and justification of components, circuit modeling, microcontroller programming, and device testing.

The first chapter provides an overview of modern sensor technologies such as BLE, Wi-Fi, and ultrasonic sensors, and highlights the relevance and current state of research in the field.

The second chapter substantiates the choice of hardware and software components, including the ESP8266 microcontroller, KY-037 sound sensor, buzzer, Arduino IDE, and Fritzing. Comparative analysis of available alternatives is provided, and the selection of components is justified.

The third chapter focuses on practical implementation, including architecture design, component integration, firmware development, and testing. The developed system demonstrated stable operation, remote configuration capability, and effective sound-based activation.

The device can be used in daily life, educational projects, or integrated into commercial IoT tracking solutions. It combines accessibility, flexibility, and ease of use.

Keywords: internet of things, IoT navigator, ESP8266, KY-037, web interface, item tracking, microcontroller, arduino IDE.

ЗМІСТ

ВСТУП

Актуальність теми. Кількість особистих гаджетів і речей постійно зростає, проблема їхньої втрати стає дедалі гострішою. Щодня люди витрачають багато часу на пошук втрачених ключів, гаманців, телефонів та інших предметів. Рішенням цієї проблеми може стати застосування технологій Інтернету речей (IoT), які дозволяють створювати інтелектуальні системи для пошуку речей у реальному часі. Розробка простого, компактного й доступного пристрою для пошуку загублених речей на базі технологій бездротового зв'язку є актуальною задачею як з точки зору зручності користувача, так і розвитку сучасних технічних рішень у галузі IoT.

Метою роботи є розробка прототипу IoT-навігатора для пошуку загублених речей, що працює на базі контролера ESP8266 з можливістю дистанційного налаштування чутливості через вебінтерфейс.

Об'єкт дослідження – апаратні та програмні рішення для побудови систем Інтернету речей.

Предмет дослідження – методи створення та програмування компактного IoT-пристрою для пошуку загублених речей на базі ESP8266.

Завдання, які необхідно виконати:

– реалізувати прототип пристрою для пошуку загублених речей із використанням технологій Інтернету речей;

– розробити електричну схему пристрою та програмне забезпечення для керування його роботою;

– дослідити особливості роботи контролера ESP8266, сумісність з різними типами сенсорів та аналіз бездротових технологій передачі даних;

– запропонувати оптимальні рішення для забезпечення стабільної та енергоефективної роботи пристрою.

Результати роботи доповідалися на міжнародній науково-практичній конференції молодих вчених та студентів «Програмне та апаратне забезпечення в інформаційних технологіях», Луцьк, 6 травня 2025 року [1].

РОЗДІЛ 1
АНАЛІЗ ТЕМИ ТА ІСТОРІЯ РОЗВИТКУ

1.1 Актуальність теми і обґрунтування вибору

У сучасному світі значна кількість людей щоденно користується великою кількістю особистих речей, таких як ключі, гаманці, мобільні телефони, навушники, пульти дистанційного керування, картки доступу та зарядні пристрої. З огляду на ритм життя, що постійно прискорюється, ці речі часто губляться або залишаються в несподіваних місцях, що створює додаткові незручності. Пошук загублених предметів забирає багато часу та зусиль, а іноді може спричинити значний стрес. Дослідження показують, що протягом життя людина витрачає в середньому 6,5 місяців на пошуки загублених речей, що підтверджує масштаб цієї проблеми [2].

Проблема загублених речей є особливо актуальною для людей, які ведуть активний спосіб життя, мають багато щоденних завдань або часто пересуваються. У таких умовах потреба в ефективному рішенні для швидкого пошуку особистих речей стає очевидною. Одним із перспективних напрямів, що дозволяє створити таке рішення, є застосування технологій Інтернету речей (IoT). Завдяки цим технологіям можна створювати пристрої, які забезпечують відстеження та пошук речей у реальному часі, значно спрощуючи їх локалізацію.

Останні роки демонструють активний розвиток IoT-рішень, що пояснюється їхньою компактністю, енергоефективністю та здатністю до бездротового підключення. Очікується, що кількість пристроїв, підключених до Інтернету, стрімко зростатиме, досягаючи близько 30 мільярдів одиниць до 2030 року. Це свідчить про широкі перспективи впровадження подібних пристроїв у повсякденне життя [3].

Слід зазначити, що серед наявних рішень для пошуку речей, таких як Bluetooth-трекери чи GPS-маячки, існують певні обмеження. Зокрема, Bluetooth-трекери мають обмежений радіус дії, що не завжди дозволяє вчасно знайти предмет [4], а GPS-пристрої потребують значних витрат енергії та фінансових

ресурсів для підтримки зв'язку з супутниками. Через це постає потреба у створенні альтернативного рішення, що забезпечить ефективний пошук речей із мінімальними витратами енергії та доступною вартістю.

Одним із можливих підходів до створення такого пристрою є використання сучасних мікроконтролерів, що поєднують компактність, низьке енергоспоживання та можливість бездротового підключення [5]. Такі пристрої можуть працювати спільно з мобільними додатками або веб-інтерфейсами, дозволяючи користувачам легко визначати місцезнаходження загублених предметів. Для сигналізації про знайдений об'єкт можна застосовувати різні методи, зокрема звукові сигнали, які активуються голосовими командами або іншими звуковими тригерами [6].

Таким чином, розробка пристрою для пошуку загублених речей є актуальною через поширеність проблеми втрати особистих предметів, потребу в її ефективному вирішенні та перспективність впровадження IoT-технологій у повсякденне життя.

1.2 Мікроконтролери та їх історія

Мікроконтролери відіграють важливу роль у сучасному світі електроніки, адже без них було б неможливо уявити розвиток автоматизованих систем, Інтернету речей та багатьох інших технологічних рішень. По суті, мікроконтролер – це маленький «мозок», що керує роботою пристрою, виконуючи задані команди та обробляючи інформацію від підключених до нього датчиків. Вони відіграють ключову роль у керуванні різноманітними процесами в техніці, автомобільній галузі, системах розумного дому, а також знаходять застосування в медицині та космонавтиці [7].

Історія мікроконтролерів бере свій початок ще у 1971 році, коли компанія Intel представила перший мікропроцесор Intel 4004, розроблений інженерами Тедом Хоффом, Федеріко Фаггіном та Станлі Мазором. Цей пристрій став першим кроком на шляху до створення повноцінних мікроконтролерів [8]. У 1974

році компанія Texas Instruments випустила TMS1000 – перший справжній мікроконтролер, що містив у собі процесор, пам'ять і засоби введення/виведення. Саме цей пристрій заклав основу для подальшого розвитку мікроконтролерів [9].

Протягом 1980-х років мікроконтролери активно вдосконалювалися, що сприяло їхньому ширшому використанню в різних галузях. Зокрема, одним із ключових покращень стало вдосконалення їхнього програмування. Спочатку програмування вимагало використання складних інструментів і мало обмежену кількість циклів перезапису даних. Компанія Microchip Technology зробила вагомий внесок у розвиток цієї галузі, коли у 1985 році випустила мікроконтролери серії PIC. Вони відзначалися простотою у програмуванні завдяки використанню спеціального програматора та базового інтерфейсу для налаштування команд. Крім того, PIC-контролери мали покращену структуру пам'яті, що дозволяло зручніше працювати зі складними алгоритмами та прискорювало процес налагодження пристроїв. Такі нововведення значно полегшили роботу інженерів і відкрили можливості для створення більш гнучких електронних рішень [10].

Однак ранні мікроконтролери мали значне обмеження – їхнє перепрограмування було складним і незручним, оскільки вимагало фізичного вилучення мікросхеми з плати для оновлення даних. Цю проблему було вирішено в 1988 році, коли компанія Intel представила мікроконтролер Intel 8051F із вбудованою флеш-пам'яттю. Це нововведення дозволило багаторазово перепрограмовувати пристрій без потреби його демонтажу, що значно полегшило процес розробки й тестування електронних проєктів [11].

Згодом технології продовжили розвиватися, і мікроконтролери ставали дедалі потужнішими, компактнішими та енергоефективнішими. Вони отримали можливість працювати з різними типами датчиків, керувати двигунами, екранами, світлодіодами й навіть бездротовими комунікаціями. Особливо великий прорив відбувся на початку 2000-х років, коли з'явилися платформи, які зробили роботу з мікроконтролерами доступною не лише для інженерів, а й для аматорів. Важливу роль у популяризації мікроконтролерів відіграли платформи

Arduino, що з'явилися у 2005 році. Arduino була створена Массимо Банзі, Девідом Куартілльєсом, Томом Ігоєм, Джанлукою Мартіно та Девідом Мелісом в Італії як навчальний проєкт для студентів Італійського інституту Interaction Design. На відміну від стандартних контролерів, Arduino відзначалась відкритим апаратним забезпеченням, спрощеною мовою програмування та широкою підтримкою спільноти, що дозволило інженерам і аматорам швидко створювати електронні проєкти. Саме це зробило платформу надзвичайно популярною серед новачків і професіоналів у галузі електроніки [12].

Серед сучасних мікроконтролерів особливої уваги заслуговують ті, що мають вбудовані засоби бездротового зв'язку, наприклад, Wi-Fi або Bluetooth. Першим мікроконтролером з інтегрованими модулями Wi-Fi і Bluetooth став ESP32, випущений у 2016 році компанією Espressif Systems. Цей контролер отримав велику популярність завдяки своїй багатофункціональності, невисокій вартості та зручності програмування. ESP32 став проривом у створенні IoT-проєктів, оскільки дозволив спростити реалізацію бездротових рішень без необхідності підключення додаткових модулів зв'язку. Завдяки цьому мікроконтролери стали ключовими елементами сучасної електроніки, що забезпечують ефективне керування пристроями та автоматизацію багатьох процесів. Їхній розвиток триває й сьогодні, а завдяки постійному вдосконаленню вони стають ще потужнішими, розширюючи межі можливостей сучасних технологій.

1.3 Дослідження різних сенсорів

Сенсори є важливою складовою сучасних електронних пристроїв. Вони виконують функцію збору даних про навколишнє середовище або стан самої системи, у якій встановлені. Завдяки сенсорам можна автоматизувати безліч процесів, підвищити ефективність роботи обладнання та забезпечити високий рівень контролю над різними об'єктами. У багатьох галузях промисловості,

медицини, транспорту та побуту сенсори стали незамінними елементами сучасних технологічних рішень.

Функціональне призначення сенсорів полягає у перетворенні фізичних величин, таких як температура, тиск, вологість, світло, звук чи магнітне поле, у електричні сигнали, які можуть бути оброблені електронною системою. Це дозволяє пристроям зчитувати зміни в середовищі та відповідно реагувати на них. Завдяки цьому сенсори широко застосовуються в системах безпеки, автоматизації, контролю параметрів довкілля, а також у пристроях для моніторингу стану здоров'я людини [13].

Існує безліч різновидів сенсорів, кожен з яких виконує свої унікальні функції. Наприклад, температурні сенсори призначені для вимірювання температури та широко використовуються в системах клімат-контролю, холодильниках або бойлерах. Датчики освітлення регулюють рівень яскравості в приміщеннях, що дозволяє заощаджувати електроенергію. У промисловості активно застосовуються датчики тиску, які забезпечують контроль над роботою гідравлічних систем та механізмів. Є також сенсори руху, які фіксують переміщення об'єктів і часто застосовуються в охоронних системах.

Серед різноманіття сенсорів особливе місце займають звукові сенсори, які чутливо реагують на звукові хвилі. Вони знайшли застосування у великій кількості різних сфер, починаючи від систем розпізнавання голосу і закінчуючи пристроями, що активуються звуком, як-от автоматичні освітлювальні прилади чи охоронні системи, які реагують на раптові гучні сигнали.

Звукові сенсори вирізняються низкою переваг, які роблять їх особливо зручними у випадках, коли важливо зафіксувати наявність або місцеперебування об'єкта за допомогою звукових сигналів. Такі сенсори здатні реагувати як на конкретний рівень гучності, так і на певну частоту звуку, що дозволяє налаштувати їх для спрацьовування на специфічні команди чи гучні сигнали. Завдяки цьому вони стали популярними в системах голосового управління та у приміщеннях, де використання інших методів виявлення є проблематичним. Додатковою перевагою є простота монтажу, доступна вартість і швидке

реагування на звукові подразники, що забезпечує ефективність їх використання в різноманітних технологічних рішеннях.

Порівняно з багатьма іншими видами сенсорів, звукові мають відносно невисоку вартість, що дозволяє їх широко використовувати навіть у бюджетних пристроях. Простота їх конструкції та легкість підключення значно полегшують процес створення різноманітних IoT-рішень для повсякденного використання.

1.4 Дослідження літератури

Bluetooth повністю змінив уявлення про бездротовий зв'язок, ставши гідною альтернативою дротовим з'єднанням і зробивши взаємодію між пристроями набагато зручнішою. Від початку свого існування наприкінці 1990-х років ця технологія стрімко розвивалась, перетворившись на універсальний стандарт для передачі даних. У цьому огляді основна увага зосереджена на еволюції Bluetooth, з особливим акцентом на Bluetooth Low Energy (BLE) – ключовому рішенні для енергоефективних бездротових систем.

Хоча створення Bluetooth – це результат командної роботи, першовідкривачем основної технології вважається Якобус Корнеліс Хаартсен, нідерландський інженер з Ericsson Mobile. Першу версію Bluetooth 1.0 було розроблено як заміну незручним дротам у гарнітурах та мобільних телефонах. Починаючи з 2000-х років, ця технологія швидко набула популярності, зокрема завдяки функції передачі даних.

Справжній прорив стався у 2010 році, коли з'явився BLE – варіант Bluetooth, створений спеціально для пристроїв із низьким енергоспоживанням. Ще на початку свого розвитку BLE був оцінений як перспективна технологія для IoT-систем. Наприклад, дослідження Gomez та ін. [14] підтверджувало, що завдяки мізерному споживанню енергії BLE чудово підходить для носимих пристроїв, систем розумного дому та інших «розумних» рішень.

BLE працює на частоті 2.4 ГГц – в тому ж ISM-діапазоні, що й Bluetooth Classic. Але на відміну від класичної версії, BLE був розроблений із нуля як

окремий протокол. Він не сумісний із BR/EDR (Bluetooth Basic Rate/Enhanced Data Rate), однак може функціонувати паралельно. Його головною перевагою стала економія енергії без втрати якості зв'язку на короткій дистанції.

Mikhaylov та колеги [15] провели ґрунтовне порівняння BLE з іншими протоколами, такими як IEEE 802.15.4 та SimpliciTI. У своїй роботі вони довели, що BLE має кращу енергоефективність, менші затримки та хорошу пропускну здатність (до 320 кбіт/с), що робить його надзвичайно привабливим для використання у пристроях з обмеженим ресурсом батареї.

У контексті IoT BLE відіграє ключову роль. Сьогодні екосистема Інтернету речей – це сукупність розумних пристроїв, сенсорів, програмного забезпечення та мереж, які спільно працюють для збору, обробки та обміну даними. До таких пристроїв належать не лише побутові гаджети, але й обладнання для промисловості, медицини, транспорту. Усі вони оснащені сенсорами, здатними фіксувати параметри навколишнього середовища, і виконавчими механізмами, що реагують на отриману інформацію.

Для з'єднання між собою ці пристрої використовують різні протоколи, і саме BLE забезпечує ідеальний баланс між енергоспоживанням та стабільністю передачі даних на близькій відстані. Завдяки цьому Bluetooth Low Energy активно використовується в IoT-пристроях, де ключовими є автономність і компактність.

Основна мета згаданого наукового огляду – простежити, як змінювалася технологія Bluetooth від моменту її створення до сьогоднішнього дня, а також оцінити її внесок у розвиток IoT. Автори аналізують не лише технічні характеристики BLE, але й описують труднощі, з якими стикаються розробники при впровадженні цієї технології, та шляхи їх подолання. Наприклад, Darroudi і Gomez розглядали питання створення BLE Mesh – технології, яка дозволяє з'єднувати багато пристроїв в одну мережу для покриття великих площ [16]. У новіших роботах, таких як дослідження Natgunanathan та ін. [17], аналізується практичне застосування BLE Mesh, зокрема його переваги та потенційні обмеження.

Зрозуміти роль BLE у сучасній цифровій екосистемі – значить отримати уявлення про те, як бездротовий зв'язок став основою більшості сучасних технологій. Завдяки низькому енергоспоживанню та високій надійності, BLE дозволяє створювати пристрої, які можуть працювати місяцями або навіть роками без потреби заряджання. Це особливо актуально для носимих гаджетів, сенсорів для «розумного» дому та інших компактних рішень.

Проте важливо пам'ятати, що BLE – лише одна з багатьох технологій у сфері бездротового зв'язку. Наприклад, Raza та ін. [18] досліджували LPWAN – мережі з великою зоною покриття та низьким енергоспоживанням, які можуть доповнювати BLE в умовах, де потрібен більший радіус дії. У свою чергу, Cui та співавтори акцентують увагу на безпеці BLE в контексті використання у «розумних містах» – зростаючій сфері, де велика кількість пристроїв працюють на основі BLE. Забезпечення надійного захисту даних та пристроїв – один із головних викликів на шляху подальшого впровадження BLE в критичних сферах.

Додатково варто згадати аналітичні праці, присвячені практичному порівнянню BLE з іншими IoT-рішеннями. Зокрема, у статті Grigorios Koulouras, Stylianos Katsoulis і Fotios Zantalis [19] наведено ґрунтовний аналіз переваг BLE у порівнянні з традиційними підходами до бездротового зв'язку в Інтернеті речей. Автори наводять практичні приклади використання BLE у сфері створення трекерів, особливо підкреслюючи його ефективність в умовах обмеженого енергоспоживання. У статті BLE описується як технологія, яка забезпечує стабільну роботу пристрою при надзвичайно малій витраті енергії, що дає змогу пристроям функціонувати на одному заряді батареї впродовж тривалого часу – іноді до кількох місяців або навіть років.

Крім того, у роботі Kostiantyn Oliynyk [20] детально досліджено технологію Wi-Fi як один із варіантів реалізації бездротового зв'язку в IoT-системах. Автор підкреслює, що Wi-Fi має кілька суттєвих переваг, серед яких – високі швидкості передачі даних та великий радіус дії, що дозволяє об'єднувати пристрої навіть у масштабних інфраструктурах, таких як великі офіси, підприємства чи виробничі комплекси. Завдяки цьому Wi-Fi стає зручним

інструментом у тих випадках, коли потрібно швидко передавати великі обсяги інформації або забезпечити стабільне з'єднання на відстані. Проте, як зазначає автор, ці переваги супроводжуються істотним недоліком – підвищеним енергоспоживанням. Це означає, що пристрої на базі Wi-Fi потребують більшого живлення, а отже, менш придатні для автономної тривалої роботи від батареї. У контексті побудови мобільних або портативних пристроїв, де критично важливо забезпечити тривалий час автономної роботи, ця особливість Wi-Fi може бути вирішальним фактором при виборі технології. Тому, хоча Wi-Fi і залишається потужним інструментом у сфері IoT, його доцільність використання залежить від специфіки задачі та вимог до енергоспоживання.

Також заслуговує на увагу публікація E&ICT Academy, IIT Kanpur [21], у якій детально розглянуто особливості роботи та використання ультразвукових сенсорів в Інтернеті речей. Автори цієї праці акцентують увагу на здатності ультразвукових сенсорів визначати відстань до об'єктів з високою точністю, що особливо корисно в умовах, де візуальне або радіочастотне позиціонування неможливе або ускладнене. Завдяки властивості ультразвукових хвиль проникати через певні типи середовищ, ці сенсори часто використовуються в складних або нестандартних умовах, наприклад, у промислових цехах, приміщеннях із підвищеним рівнем пилу або шуму, а також у логістиці для контролю рівня заповнення складських ємностей.

Попри це, автори звертають увагу на низку суттєвих обмежень, пов'язаних з ультразвуковими сенсорами. Зокрема, їх інтеграція в IoT-пристрої потребує ретельного налаштування, використання відповідних алгоритмів фільтрації сигналу та додаткових мікроконтролерів для обробки зібраних даних. Крім того, вартість таких сенсорів у поєднанні з додатковими апаратними складовими може істотно вплинути на загальну вартість кінцевого продукту. Це ускладнює їх використання в побутових пристроях або недорогих рішеннях, орієнтованих на масового користувача. Таким чином, хоча ультразвукові сенсори і мають значні технічні переваги, їх застосування вимагає зваженого підходу, особливо при розробці простих, компактних і економічно доцільних IoT-пристроїв.

На підставі аналізу згаданих джерел можна зробити висновок, що жодна з технологій не є універсальною. BLE є найкращим у контексті енергоощадності, Wi-Fi – у швидкості передачі даних, а ультразвук – у точності. Тому при створенні IoT-навігаторів варто шукати баланс між функціональністю, вартістю, енергоспоживанням і можливістю реалізації в конкретному середовищі застосування.

РОЗДІЛ 2
ПРОЄКТУВАННЯ АПАРАТНОЇ СКЛАДОВОЇ ПРИСТРОЮ

2.1 Теоретичне обґрунтування загальної архітектури пристрою

Основний принцип роботи пристрою базується на тому, що користувач подає команду (наприклад, плескає в долоні), а пристрій розпізнає цей звук і активується. У відповідь він подає звуковий сигнал, що допомагає знайти загублену річ. Ми обрали саме звукову активацію, оскільки це зручніше для користувача – він може подати команду в будь-який момент, не витрачаючи час на пошук смартфона чи інших пристроїв. Крім того, цей метод є простим і не потребує додаткових дій, як-от встановлення додаткових додатків чи натискання кнопок. Достатньо звичайного плеску, щоб пристрій миттєво зреагував, видаючи чіткий звуковий сигнал, який приверне увагу власника.

Звуковий сигнал має значні переваги перед іншими методами, такими як Bluetooth або GPS. Наприклад, GPS є ефективним для пошуку великих об’єктів, але в умовах приміщення його точність значно знижується, що робить його малопридатним для пошуку дрібних речей. Bluetooth, у свою чергу, працює лише в межах певної зони дії, і якщо користувач перебуває поза цим радіусом, пошук буде неможливим. Натомість звук не має таких обмежень – він розповсюджується у просторі незалежно від перешкод та відстані в межах кімнати, що робить його універсальним і зручним рішенням.

Однією з головних переваг звукового сигналу є його природність для людини. Ми інтуїтивно орієнтуємося на звук, коли щось шукаємо, наприклад, коли кличемо домашню тварину або реагуємо на дзвінок телефону. Тому використання саме звукового методу є найбільш логічним для швидкого виявлення загубленої речі. До того ж, людський слух чудово адаптований до сприйняття звуків у навколишньому середовищі, що робить цей метод особливо ефективним.

Ще один важливий фактор – це економія енергії. Звуковий сигнал споживає мінімальну кількість енергії, що дозволяє пристрою працювати довше без частих

підзарядок або заміни батарейок. На відміну від пристроїв, які використовують постійний зв'язок через Bluetooth або Wi-Fi, наш навігатор активується лише у відповідь на команду користувача, що значно подовжує його автономність.

Також варто зазначити, що пристрій може бути корисним для людей з порушеннями зору. Завдяки звуковому сигналу вони зможуть легко знаходити свої речі без необхідності переглядати екран смартфона чи читати повідомлення. Це робить IoT-навігатор універсальним і доступним для широкого кола користувачів.

Таким чином, вибір звукового сигналу є оптимальним рішенням для розробки IoT-навігатора, оскільки він поєднує в собі простоту, ефективність, енергоощадність і доступність для різних категорій користувачів.

2.2 Аналіз та обґрунтування вибору мікроконтролера

При розробці будь-якого електронного пристрою, особливо такого, що належить до категорії IoT-рішень, одним із ключових етапів є вибір апаратної основи – контролера, який керуватиме роботою всієї системи. Саме від нього залежить, наскільки стабільно, енергоефективно та функціонально працюватиме пристрій. У нашому випадку мова йде про створення IoT-навігатора для пошуку загублених речей, отже, контролер повинен бути не лише компактним, а й мати вбудований модуль бездротового зв'язку, підтримку роботи з сенсорами, можливість енергоощадного режиму та достатню кількість портів для підключення периферії.

Одним із найвідоміших та найпоширеніших мікроконтролерів, з якого починають багато розробників, є Arduino Uno (рис. 2.1). Цей контролер розроблений компанією Arduino.cc, побудований на базі мікроконтролера ATmega328P і має достатньо зручний форм-фактор. Arduino Uno підтримується великою кількістю бібліотек, має просте підключення до ПК через USB, а також активно використовується в навчальних цілях. Проте його недоліком є відсутність вбудованих засобів бездротового зв'язку – для реалізації Wi-Fi або

Bluetooth підключення необхідно додатково використовувати зовнішні модулі, що ускладнює конструкцію пристрою, збільшує його розміри, споживання енергії та ціну. До того ж, Arduino Uno досить габаритний, що робить його не найкращим вибором для компактних IoT-рішень.

Рисунок 2.1 – Мікроконтролер Arduino Uno [22]

Іншим потенційним варіантом був Arduino Nano (рис. 2.2) – це компактніша версія Arduino Uno, яка зберігає більшість функцій оригінальної плати. Вона зручна для розміщення у малих пристроях, підтримує стандартні інтерфейси, має схожий мікроконтролер ATmega328.

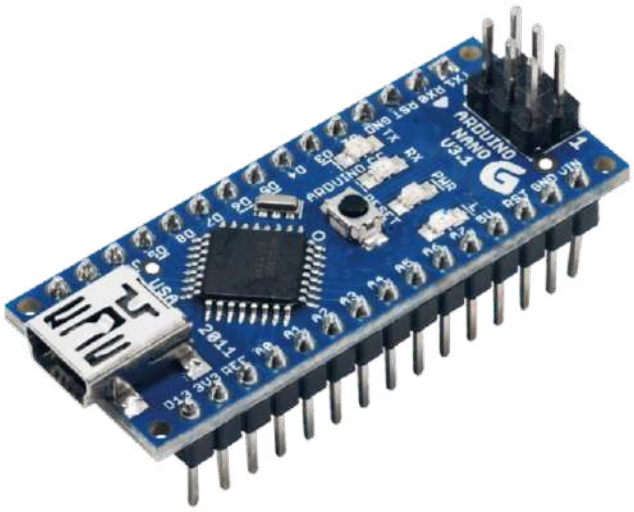

Рисунок 2.2 – Мікроконтролер Arduino Nano [22]

Однак, як і у випадку з Uno, для додавання бездротових функцій потрібен окремий модуль, наприклад ESP-01 (рис. 2.3) або NRF24L01 (рис. 2.4). Це не лише створює потребу у складніших схемах підключення, а й додає потенційних точок відмови, що є критичним для пристроїв, що мають працювати стабільно і довго в автономному режимі.

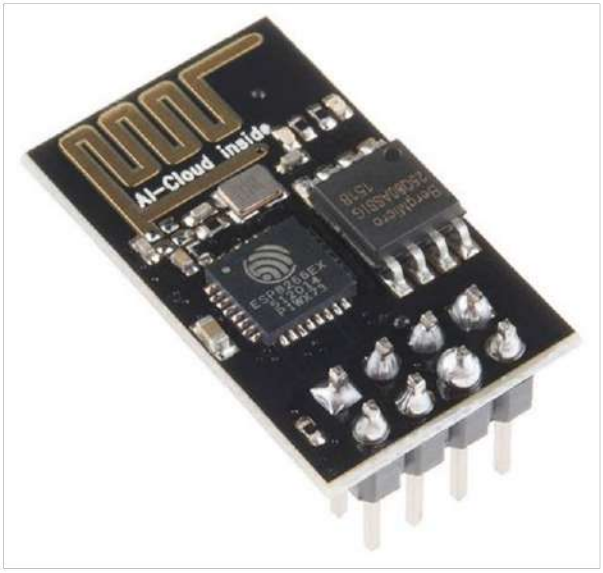

Рисунок 2.3 – Wi-Fi модуль ESP-01 [22]

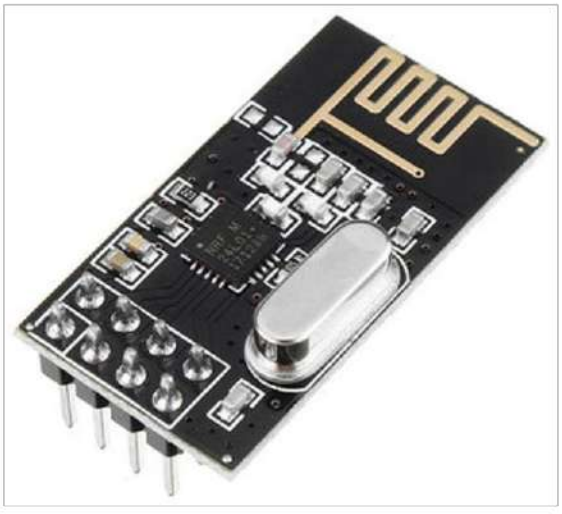

Рисунок 2.4 – Wi-Fi модуль NRF24L01 [22]

Більш потужною альтернативою є ESP32 (рис. 2.5) – мікроконтролер, розроблений компанією Espressif. Він має два ядра, працює на частоті до 240 МГц, підтримує Wi-Fi та Bluetooth, має значно більшу кількість вхідних/вихідних

портів, а також можливість підключення до більш складних пристроїв. ESP32 ідеально підходить для завдань з високими вимогами до обчислювальної потужності. Але в нашому випадку така потужність є надлишковою, а отже – неефективною. Крім того, ESP32 є дорожчим і споживає більше енергії, ніж простіші рішення. Для пристрою, який повинен працювати тривалий час від батарейки та виконувати відносно прості задачі, ESP32 видається занадто ресурсоємним.

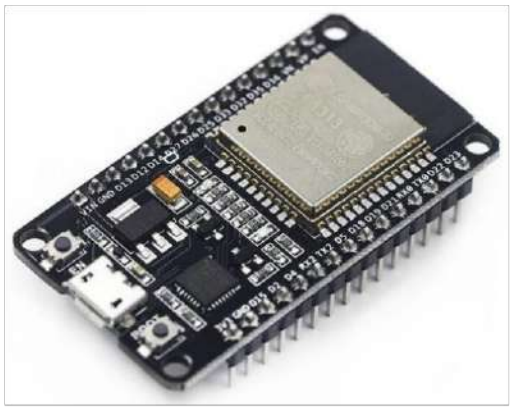

Рисунок 2.5 – Мікроконтролер ESP32 [22]

Ще одним варіантом, який заслуговує на увагу, є ESP8266 mini (рис. 2.6) – мікроконтролер, розроблений компанією Espressif, який швидко завоював популярність серед розробників IoT-проєктів завдяки своїй простоті, доступності та широкому функціоналу.

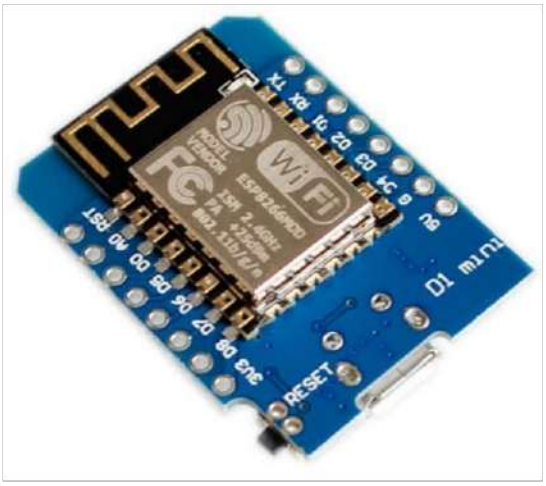

Рисунок 2.6 – Мікроконтролер ESP8266 mini [22]

Він побудований на 32-бітному процесорі Tensilica L106 із тактовою частотою до 160 МГц і має до 4 МБ флеш-пам'яті, а головною його перевагою є вбудований модуль Wi-Fi, який підтримує стандарти IEEE 802.11 b/g/n, що дозволяє реалізовувати бездротову передачу даних без необхідності у зовнішніх модулях. Підтримка таких інтерфейсів, як UART, SPI, I2C, PWM, а також цифрових і аналогових входів/виходів забезпечує сумісність з більшістю датчиків, що робить його універсальним для побудови різних систем. До того ж, його компактні розміри та низьке енергоспоживання дають змогу застосовувати його у портативних пристроях з автономним живленням, а можливість переходу в режим сну сприяє значному збільшенню тривалості роботи на батареї. У розробці ESP8266 вирізняється ще й тим, що активно підтримується спільнотою: доступні численні бібліотеки, приклади, інструкції, що спрощують інтеграцію та знижують поріг входу для новачків. Контролер легко програмується через Arduino IDE, що забезпечує зручність та гнучкість у створенні прошивок. У порівнянні з Arduino Uno та Nano, ESP8266 має беззаперечну перевагу – наявність вбудованого Wi-Fi, що дозволяє уникнути використання додаткових модулів і тим самим зменшує розміри схеми та енергоспоживання. У порівнянні з ESP32, який є потужнішим і має Bluetooth, ESP8266 поступається за продуктивністю, однак зважаючи на простоту поставленого завдання та необхідність економного використання енергії, ESP8266 виявляється більш доцільним вибором. Його баланс між функціональністю, енергоефективністю та простотою використання робить його оптимальним варіантом для реалізації нашого IoT-навігатора для пошуку загублених речей.

2.3 Техніко-функціональне обґрунтування вибору елементної бази

Окрім контролера, важливою частиною будь-якого IoT-пристрою є набір допоміжних компонентів, які забезпечують взаємодію з навколишнім середовищем і користувачем. Для реалізації нашого IoT-навігатора було обрано кілька ключових елементів: мікрофонний сенсор, зумер та модуль живлення.

Кожен з цих компонентів було підібрано на основі аналізу кількох альтернатив, враховуючи сумісність, простоту інтеграції, енергоспоживання, розміри та надійність.

Як засіб виявлення звуку було обрано мікрофонний модуль KY-037 (рис. 2.7). Цей модуль має регульовану чутливість, оснащений підсилювачем та компаратором, що дозволяє фіксувати звуки вище певного рівня гучності. KY-037 добре працює з мікроконтролерами серії ESP і має просте підключення – лише декілька контактів (живлення, земля, аналоговий та цифровий вихід).

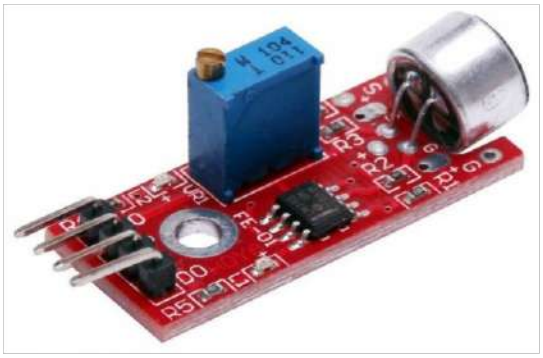

Рисунок 2.7 – Мікрофон KY-037 [22]

Альтернативами могли б бути модулі MAX9814 (рис. 2.8) або KY-038. Проте KY-038 має гіршу чутливість та стабільність, а MAX9814 – хоч і забезпечує кращу якість звуку, значно дорожчий і вимагає більш складного підключення. Враховуючи потребу у простому, компактному та енергоефективному рішенні, KY-037 є оптимальним вибором.

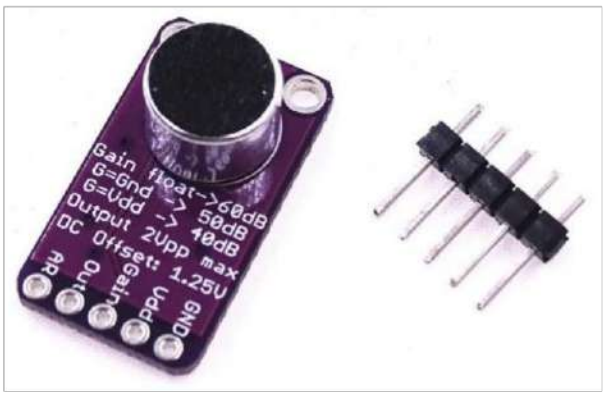

Рисунок 2.8 – Мікрофон MAX9814 [22]

Як елемент оповіщення використовується активний зумер що зображений на рисунку 2.9 з трьома контактами (GND, I/O, VCC). Цей тип зумера дозволяє подавати логічний сигнал з мікроконтролера без потреби в генерації частоти, що спрощує схему. Звук, який видає зумер, є достатньо гучним, щоб бути почутим навіть у шумному середовищі. Серед альтернатив були пасивні зумери, але вони потребують ШІМ-сигналу, що створює додаткове навантаження на контролер. Отже, активний зумер є кращим варіантом у контексті нашого проєкту.

Рисунок 2.9 – Активний зумер [22]

Живлення пристрою реалізується за допомогою Li-Po акумулятора (рис. 2.10), що підключається через спеціалізований модуль Wemos D1 mini Battery Shield (рис. 2.11).

Рисунок 2.10 – Li-Po акумулятор [22]

Цей модуль дозволяє легко підключити акумулятор до мікроконтролера, забезпечуючи не лише стабілізоване живлення, а й можливість заряджання акумулятора безпосередньо в пристрої через micro-USB порт. Battery Shield обладнаний вбудованим захистом від перенапруги, перерозряду та короткого замикання, що робить його безпечним для використання навіть у тривалих автономних проєктах.

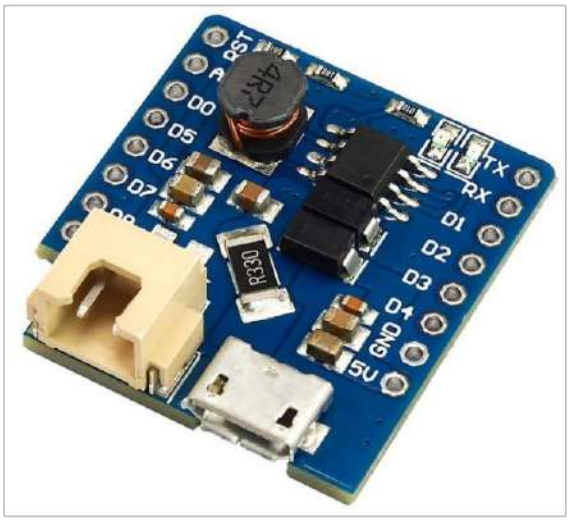

Рисунок 2.11 – Модуль живлення Wemos D1 mini Battery Shield [22]

Порівняно з традиційними кнопковими батарейками типу CR2032, акумуляторна система на базі Li-Po значно ефективніша – вона забезпечує більшу ємність, довший термін роботи та можливість багаторазового використання без заміни елементів живлення. Такий підхід також дозволяє спростити експлуатацію пристрою, адже для його підзарядки не потрібно знімати корпус або розбирати схему. Battery Shield ідеально інтегрується з контролером Wemos D1 mini (на базі ESP8266), оскільки був розроблений саме для сумісності з цією платформою. Завдяки цьому можна уникнути необхідності додаткових стабілізаторів напруги чи зарядних модулів, що зменшує складність схеми та загальні габарити пристрою. Таким чином, вибір акумуляторного живлення з використанням Wemos D1 mini Battery Shield є практичним рішенням, яке поєднує енергоефективність, компактність та зручність обслуговування IoT-пристрою.

2.4 Вибір програмного забезпечення

Під час розробки IoT-навігатора для пошуку загублених речей особливу увагу було приділено вибору відповідного програмного забезпечення, яке дозволило б не лише створити електронну схему пристрою, а й написати програмний код для керування всіма його компонентами. У результаті аналізу доступних інструментів було обрано дві ключові програми – Fritzing для візуального моделювання електронної схеми та Arduino IDE для написання, компіляції та завантаження прошивки на мікроконтролер.

Програма Fritzing – це зручний графічний інструмент, розроблений для візуального проєктування електронних схем, особливо популярний серед розробників прототипів і початківців у галузі мікроелектроніки. Однією з ключових переваг Fritzing є можливість створення схем з використанням зображень реальних компонентів, таких як Arduino, ESP-плати, резистори, сенсори та інші модулі. Це дозволяє легко уявити, як виглядатиме зібраний пристрій у реальному житті. Fritzing також дозволяє генерувати друковані плати (PCB) для подальшого виготовлення, що може стати в нагоді при розширенні або тиражуванні проєкту. У порівнянні з іншими CAD-програмами для електроніки, такими як KiCad або Eagle, Fritzing є більш інтуїтивно зрозумілим, але водночас має обмежений функціонал для складних багаторівневих плат. У нашому випадку його простота та швидкість візуалізації виявилися вирішальними факторами.

Щодо програмування мікроконтролера, було обрано Arduino IDE – офіційне середовище розробки для платформ Arduino та сумісних мікроконтролерів, таких як ESP8266. Arduino IDE підтримує мову програмування на основі C/C++, має простий інтерфейс, автоматичну компіляцію та зручну функцію завантаження коду на мікроконтролер. Програма також підтримує великий вибір бібліотек, що значно спрощує інтеграцію модулів, таких як датчики, дисплеї або мережеві компоненти. У порівнянні з альтернативами, такими як PlatformIO, яка надає більш розширені функції (наприклад, підтримку автодоповнення, налагодження, вбудоване тестування),

Arduino IDE залишається простішим у налаштуванні та освоєнні, що є вагомим аргументом на користь її використання у навчальних і прототипних проєктах.

Таким чином, вибір Fritzing і Arduino IDE для реалізації проєкту був зумовлений поєднанням зручності, простоти використання, широкої підтримки спільноти розробників і відповідності поставленим завданням. Обидві програми є відкритими або безкоштовними для навчального використання, що також відповідає критеріям доступності й ефективності при створенні індивідуального IoT-пристрою.

РОЗДІЛ 3
ПРАКТИЧНА РЕАЛІЗАЦІЯ IOT-НАВІГАТОРА ДЛЯ ПОШУКУ ЗАГУБЛЕНИХ РЕЧЕЙ

3.1 Архітектура пристрою та взаємодія компонентів системи

Архітектура IoT-навігатора для пошуку загублених речей базується на модульному підході, який забезпечує простоту збирання, гнучкість у конфігурації та легкість у подальшому масштабуванні пристрою. У центрі архітектури – мікроконтролер WeMos D1 mini на базі чипа ESP8266, який забезпечує обробку сигналів, керування виконавчими модулями та бездротову комунікацію з іншими пристроями.

Фізична структура пристрою складається з кількох основних компонентів, що взаємодіють між собою через стандартні пінові інтерфейси. До контролера WeMos D1 mini зверху підключається модуль живлення Battery Shield, який забезпечує стабільне живлення від Li-Po акумулятора, з можливістю одночасної зарядки та роботи пристрою. Завдяки стандартному розташуванню контактів, Battery Shield щільно прилягає до основної плати та не потребує додаткових дротів або роз'ємів, що мінімізує габарити пристрою та підвищує надійність.

До аналогового входу A0 мікроконтролера підключається мікрофон KY-037, який реагує на зміни рівня звукового тиску. Цей датчик забезпечує подачу аналогового сигналу, що далі аналізується прошивкою пристрою для визначення наявності звукового тригера. Після фіксації голосового або шумового сигналу, пристрій реагує – надсилає команду на виконавчий модуль, яким є зумер, підключений до цифрового піну D6 (GPIO12). Зумер генерує гучний звуковий сигнал, що дозволяє знайти загублений предмет.

Кожен із компонентів має чітко визначену роль у системі та взаємодіє через прості сигнальні та живильні інтерфейси. Структурно, пристрій має одноблокову логіку: центральний мікроконтролер отримує вхідний аналоговий сигнал, обробляє його відповідно до закладеної логіки в програмному коді, і, у випадку спрацювання умов, керує виконавчим пристроєм. Усі компоненти об'єднані на

рівні логіки керування через мікроконтролер, що дозволяє централізовано реалізовувати розширення – наприклад, додавання Bluetooth або датчиків руху, зміну логіки обробки сигналу тощо.

Крім того, важливу роль відіграє Wi-Fi модуль, інтегрований у чип ESP8266. Він забезпечує обмін даними між пристроєм та смартфоном або сервером, дозволяючи реалізовувати сценарії керування на відстані. Наприклад, у майбутньому можливо реалізувати web-інтерфейс або мобільний застосунок, який надсилатиме команду на активацію зумера через локальну мережу.

Загальна архітектура пристрою є прикладом класичної IoT-системи: датчик, контролер, виконавчий механізм і інтерфейс бездротового зв'язку. Такий підхід дозволяє підтримувати компактність, енергоефективність і функціональну завершеність, необхідну для реального використання в повсякденному житті. У наступних підрозділах розглянуто реалізацію апаратної частини та особливості програмної логіки, що забезпечує взаємодію між компонентами згідно з цією архітектурою.

3.2 Проєктування схеми пристрою, схема підключення

Розробка схеми пристрою є одним із ключових етапів у створенні будь-якого електронного проєкту, адже саме на цьому етапі визначається, як компоненти будуть взаємодіяти між собою, до яких пінів буде підключено живлення, передача даних, керування і які інтерфейси використовуватимуться. У контексті IoT-навігатора на базі ESP8266 важливим є точне розуміння розпіновки та особливостей кожного GPIO-виводу, оскільки будь-яка помилка може призвести до того, що контролер просто не запуститься або працюватиме нестабільно.

Контролер WeMos D1 mini має на платі пінову нумерацію, яка не збігається з фактичними номерами виводів мікросхеми ESP8266. На платі контакти позначені як D0–D8, проте ці маркування відповідають певним номерам GPIO (рисунок 3.1). Наприклад, пін D1 – це GPIO5, а D2 – GPIO4. У коді Arduino IDE

допустимо звертатись як до D1, так і до GPIO5, однак варто бути уважним і не плутати ці позначення. Крім того, деякі пінові контакти можуть мати ще третє ім'я – TX і RX (GPIO1 і GPIO3), які використовуються для серійної комунікації. Наприклад, digitalWrite(TX, LOW) – це допустимий виклик у коді.

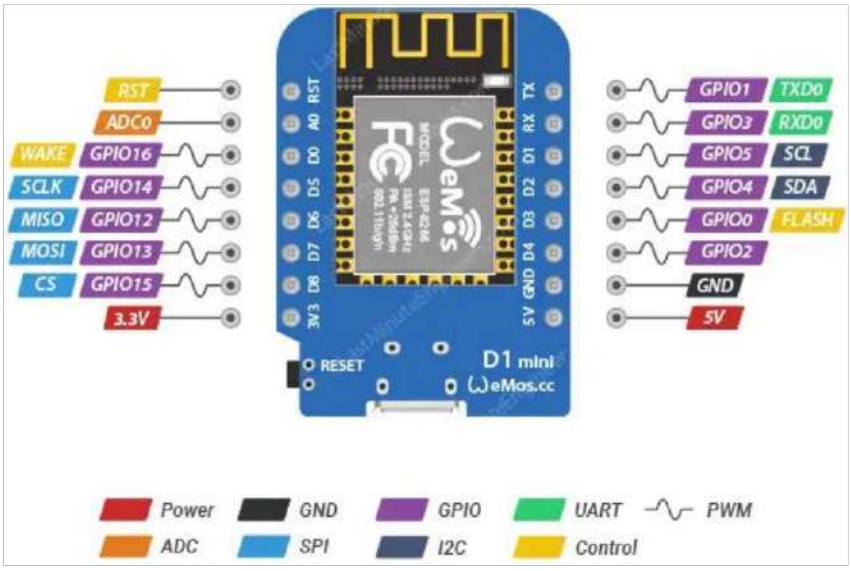

Рисунок 3.1 – Нумерація пінів контролера WeMos D1 mini [23]

Деякі пінові лінії мають системне значення – під час запуску ESP8266 вони повинні бути у певному логічному стані. Якщо під'єднати до них компонент, який порушить цей стан, мікроконтролер не зможе завантажити прошивку. Наприклад, GPIO0 (D3) та GPIO2 (D4) повинні бути у високому стані (HIGH) при завантаженні, а GPIO15 (D8) – у низькому (LOW), інакше завантаження буде заблоковано. Саме тому ці пінові лінії вже підтягнуті резисторами на платі до відповідного рівня, але при підключенні зовнішніх пристроїв до них варто враховувати, як ці пристрої впливають на напругу на піні. На рисунку 3.2 зображено лінії до яких безпечно можна підключати компоненти і до яких потрібно звернути увагу перед підключенням.

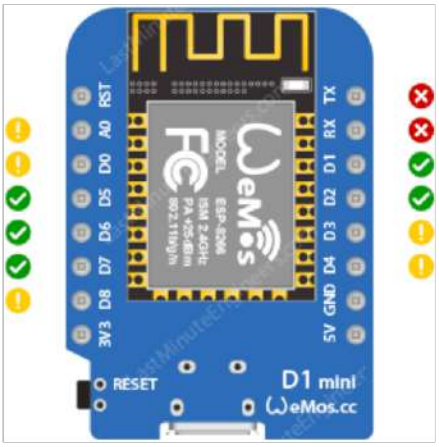

Рисунок 3.2 – Безпечні виходи на контролері [23]

Особливу увагу слід звернути на пін GPIO16 (D0). Його не можна використовувати для ШІМ-сигналів або переривань. Зате цей пін підходить для виходу з режиму сну, що дозволяє реалізовувати енергоощадний режим роботи пристрою. Також варто знати, що світлодіод на платі (LED_BUILTIN) знаходиться на GPIO2, і його логіка інверсна: при LOW світлодіод вмикається. Ще одна особливість ESP8266 – це те, що під час старту більшість пінів автоматично піднімаються у високий стан (HIGH). Тобто, якщо до них підключені, наприклад, реле або транзистори, це може спричинити їхнє короткочасне спрацьовування. Найбільш «спокійними» піновими лініями вважаються D1 (GPIO5) та D2 (GPIO4) – саме їх доцільно використовувати для підключення критичних компонентів.

Щодо можливостей підключення до Wi-Fi, то ESP8266 підтримує декілька режимів роботи: режим клієнта (STA), режим точки доступу (AP) та комбінований режим STA + AP. У першому випадку мікроконтролер підключається до вже існуючої мережі, у другому – сам створює Wi-Fi мережу, до якої можуть підключатися інші пристрої. Це дозволяє реалізовувати автономну або гібридну модель роботи, в залежності від сценарію застосування. Зокрема, у режимі AP пристрій може бути доступним навіть у польових умовах, без наявності зовнішньої інфраструктури.

Таким чином, врахування особливостей пінів, правильне призначення логічних рівнів, уникнення конфліктів при запуску та грамотне використання

режимів Wi-Fi – усе це дозволяє забезпечити стабільну, надійну та енергоефективну роботу IoT-навігатора на базі ESP8266.

На рисунку 3.3 зображена схема підключення пристрою, яка передбачає логічну та компактну інтеграцію всіх основних компонентів. У центрі конструкції знаходиться контролер WeMos D1 mini, на який зверху встановлюється Battery Shield – спеціальний модуль живлення, що дозволяє підключити Li-Po акумулятор, одночасно забезпечуючи зарядку акумулятора через micro-USB порт та стабілізацію живлення. Такий підхід дає змогу створити зручну модульну систему, в якій живлення та контролер з'єднуються без потреби додаткових проводів, завдяки стандартному гребінцю контактів.

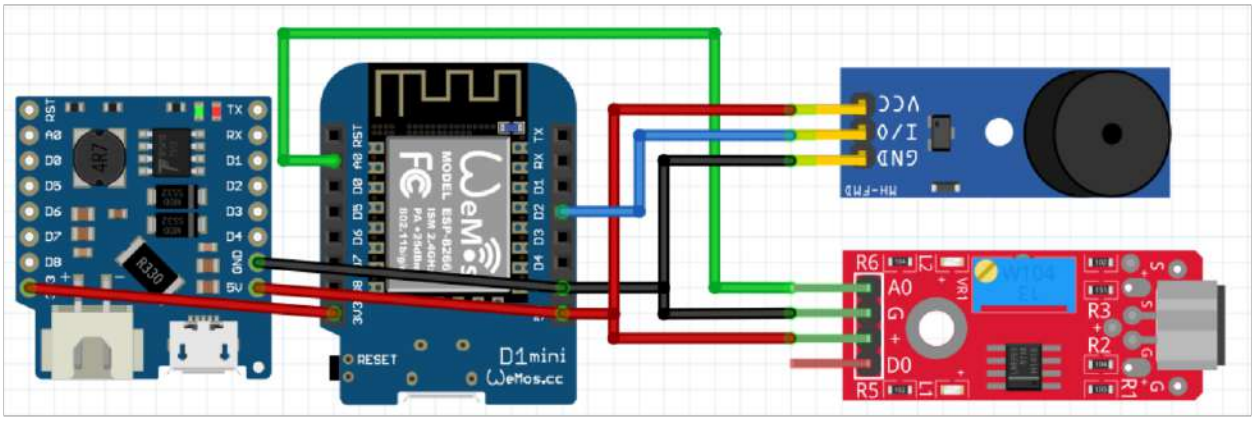

Рисунок 3.3 – Схема підключення IoT-навігатора

Кожен з компонентів – мікрофон KY-037 та зумер – має бути підключений до спільної «землі» (GND), що забезпечує стабільну електричну схему. Аналогічно, до контактів з живленням під'єднуються живильні лінії модулів. Правильне підключення живлення є важливим для стабільної роботи, оскільки перевищення або нестача напруги може спричинити помилки в роботі сенсорів або самого мікроконтролера.

Мікрофон KY-037, який ми використовуємо як сенсор, має вихід аналогового типу. Тому його сигнальний контакт підключається до аналогового входу A0 на платі WeMos D1 mini. Два інших контакти мікрофона – живлення (VCC) та земля (GND) – підключаються відповідно до пінів 5V та GND

контролера. Такий підхід дозволяє зчитувати значення звукового тиску у вигляді напруги та аналізувати їх у коді мікроконтролера.

Що стосується зумера, то для його керування обрано пін D2 (GPIO4). Цей пін рекомендований для подачі сигналу керування через свою «нейтральність»: він не бере участі в процесі завантаження прошивки, не має вимог до рівня сигналу при старті мікроконтролера та не конфліктує з системними процесами. Для його живлення використовується пін 5V на платі WeMos D1 mini, який забезпечує стабільну напругу, достатню для роботи малопотужного звукового модуля. Таким чином, зумер має три підключення: один контакт – до цифрового керуючого виходу D2, другий – до загального контакту GND, а третій – до джерела живлення 5V.

Узагальнюючи, можна сказати, що схема підключення побудована на базових принципах зручності, компактності та електричної стабільності. Вона є логічно впорядкованою, не перевантажена зайвими елементами та дозволяє без проблем масштабувати або змінювати конфігурацію пристрою в майбутньому, якщо з’явиться потреба у підключенні нових модулів.

3.3 Реалізація апаратної частини пристрою

Після завершення етапу проєктування схеми та визначення архітектури пристрою, розпочалася безпосередня реалізація апаратної частини IoT-навігатора для пошуку загублених речей. Цей етап передбачає не лише фізичне складання компонентів, а й врахування практичних нюансів, які впливають на стабільність, зручність та довговічність роботи пристрою в реальних умовах експлуатації.

Початковим завданням стало підготувати всі необхідні модулі та забезпечити їхню справність. До складу апаратної частини увійшли мікроконтролер WeMos D1 mini, звуковий сенсор KY-037, активний зумер з трьома контактами (VCC, GND, SIGNAL) та Battery Shield, який слугує модулем живлення на основі акумулятора типу Li-Po 3.7 В. Кожен з елементів перед

складанням був протестований окремо, щоб унеможливити проблеми після інтеграції всієї системи.

Процес складання розпочався з фіксації Battery Shield поверх плати контролера WeMos D1 mini. Ці два модулі мають однакове розташування пінів, що дозволяє здійснити їх з'єднання без жодного дроту – через стандартний набір гребінчастих контактів. Такий метод не лише значно полегшує збирання, а й підвищує надійність, адже зменшується кількість потенційно ненадійних з'єднань. Крім того, Battery Shield оснащений вбудованим зарядним модулем, який дозволяє заряджати акумулятор безпосередньо через micro-USB роз'єм – це дуже зручно, особливо під час тестування чи польових умов експлуатації.

Наступним кроком стало підключення звукового сенсора KY-037. Цей модуль має три виводи: VCC, GND та A0 (аналоговий вихід). Контакт VCC був підключений до виводу 5V на платі контролера, GND – до загальної «землі», а аналоговий вихід – до входу A0. Сигнальний дріт було максимально скорочено, щоб уникнути впливу шумів і спотворень при передачі аналогового сигналу. Крім того, для покращення фіксації та надійності контакту всі з'єднання були закріплені пайкою. Пайка проводилася з використанням свинцевмісного припою з флюсом, що гарантує гарну змочуваність і довговічність з'єднань.

Що стосується зумера, то його керуючий пін був підключений до D2 (GPIO4), оскільки цей пін є «спокійним» при старті системи і не конфліктує з іншими модулями. Як і у випадку з мікрофоном, живлення зумера подавалося з виводу 5V, а «земля» – до загального GND. Оскільки зумер має активний вбудований генератор, він не потребує ШІМ або частотної генерації з боку контролера – достатньо подати логічний високий або низький рівень на пін SIGNAL. Це значно спрощує програмну частину та дозволяє легко тестувати пристрій у польових умовах. Підключення зумера також здійснювалося методом пайки, що забезпечило міцність фіксації та унеможливило випадкове від'єднання дротів при транспортуванні.

Особливу увагу було приділено організації з'єднань. Усі з'єднання були підібрані відповідної довжини, згруповані в пучки та закріплені термозбіжними

трубками або пластиковими хомутами. Це дозволило уникнути хаотичного розташування проводів, що могло б призвести до механічного навантаження на контакти або короткого замикання. Також для тестування був підготовлений макетний корпус, у якому зручно розмістити всі компоненти, не порушуючи ергономіки та забезпечуючи швидкий доступ до основних елементів під час налаштування.

Весь процес складання супроводжувався регулярним візуальним контролем і перевіркою за допомогою мультиметра. Особливо це стосується перевірки живлення, правильності підключення пінів та відсутності коротких замикань. Такий підхід дозволив вчасно виявити та усунути потенційні помилки ще до подачі живлення на плату, що суттєво знизило ризик пошкодження електронних компонентів.

На завершення етапу складання апаратної частини був виконаний перший запуск пристрою. Плата була підключена до комп'ютера через micro-USB кабель для подачі живлення та одночасного завантаження прошивки. Пристрій успішно завантажився, коректно визначив мікрофонний сигнал та активував зумер при досягненні порогового значення. Це підтвердило працездатність усіх компонентів та правильність виконаного монтажу.

Отже, реалізація апаратної частини пристрою була виконана з урахуванням усіх технічних та ергономічних вимог. У процесі складання використовувались стандартні методи з'єднання, контроль якості пайки, оптимізація маршруту дротів та тестування на кожному етапі. Це дозволило створити стабільну, компактну та легко обслуговувану систему, яка повністю готова до інтеграції з програмною частиною.

3.4 Розробка програмного забезпечення

Програмне забезпечення є невід'ємною складовою розробленого пристрою. Воно забезпечує логіку взаємодії компонентів, обробку вхідних даних із мікрофона, прийняття рішень, а також активацію виконавчого елементу –

зумера. Саме завдяки прошивці пристрій набуває функціональності та здатен виконувати покладені на нього завдання. Тому процес створення програмної частини включає як розробку алгоритму, так і безпосереднє програмування мікроконтролера у відповідному середовищі.

Принцип роботи програмного забезпечення полягає в безперервному зчитуванні значення з аналогового входу A0, до якого підключено мікрофон KY-037. У разі перевищення певного порогового значення, що свідчить про гучний звук чи голосову команду, активується зумер. При цьому алгоритм повинен мати можливість налаштування чутливості, а також враховувати захист від хибних спрацювань – наприклад, короткочасних шумів або імпульсних перешкод. На рисунку 3.4 подано загальну схему роботи програмного забезпечення.

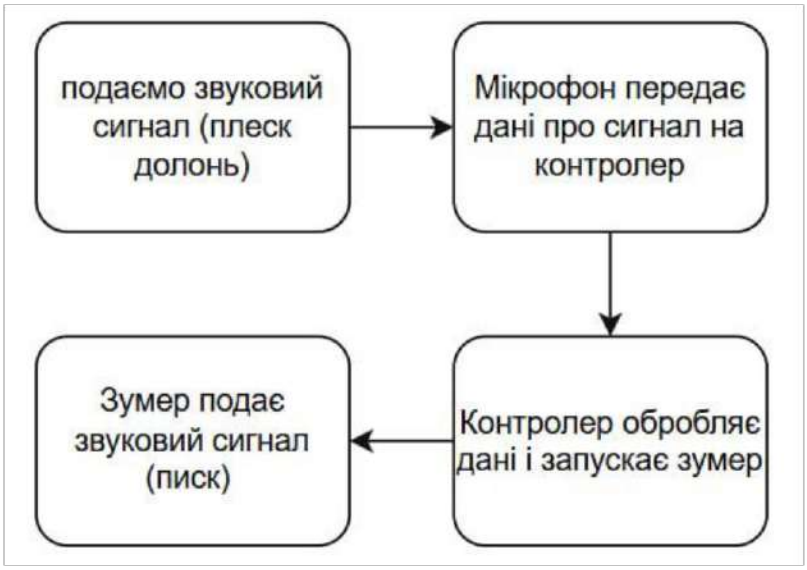

Рисунок 3.4 – Блок-схема логіки роботи прошивки

Для розробки коду було обрано середовище Arduino IDE, яке є зручним і доступним інструментом для програмування мікроконтролерів ESP8266. Перед початком написання програми необхідно встановити Arduino IDE (рекомендується версія 1.8.x або 2.0.x), а також додати підтримку плати WeMos D1 mini до середовища. Це здійснюється через вкладку налаштувань, де в поле «Additional Board Manager URLs» потрібно додати адресу «http://arduino.esp8266.com/stable/package_esp8266com_index.json» (рис. 3.5).

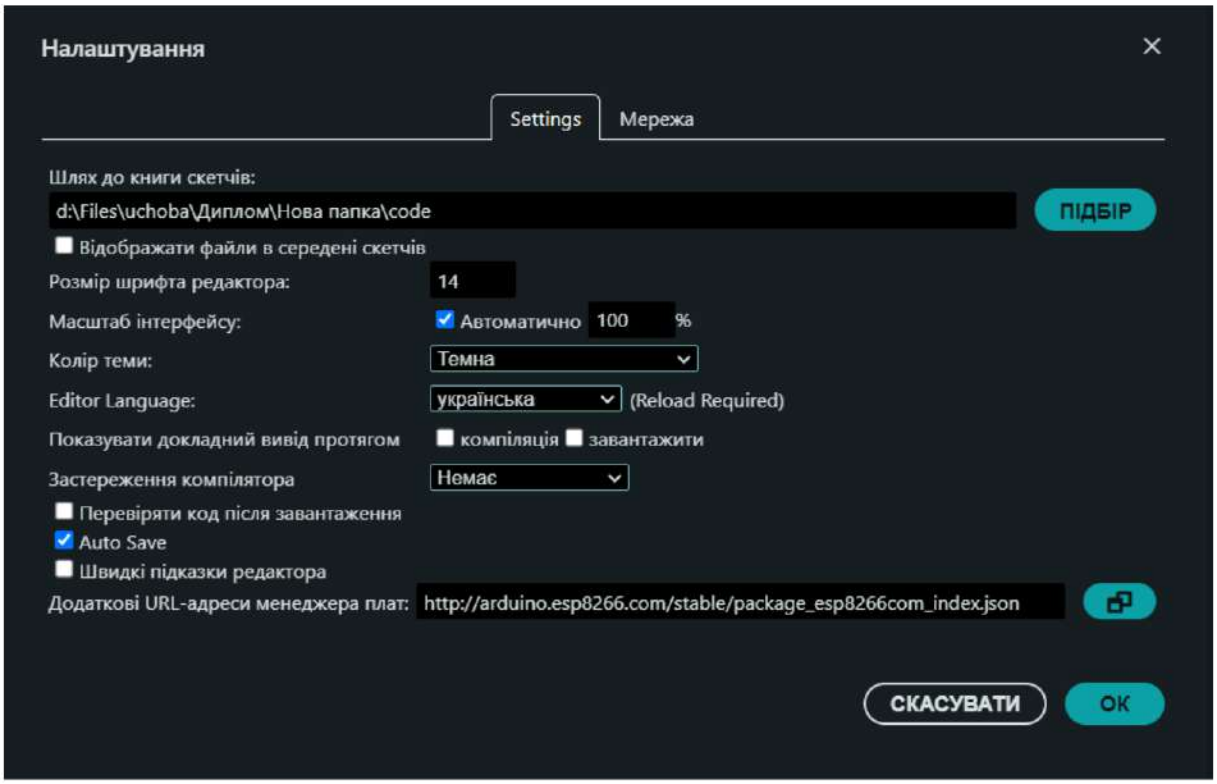

Рисунок 3.5 – Налаштування Arduino IDE

Після цього через менеджер плат встановлюється бібліотека «ESP8266 by ESP8266 Community» (рис. 3.6), а в меню вибору плат необхідно обрати «LOLIN(WEMOS) D1 R2 & mini» (рис. 3.7).

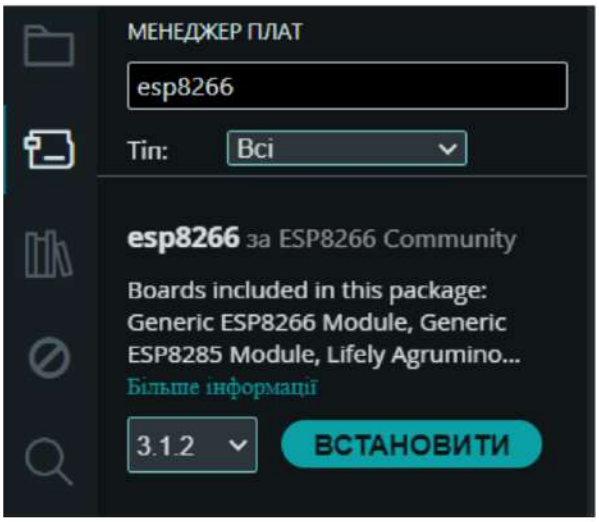

Рисунок 3.6 – Встановлення бібліотеки «ESP8266 by ESP8266 Community»

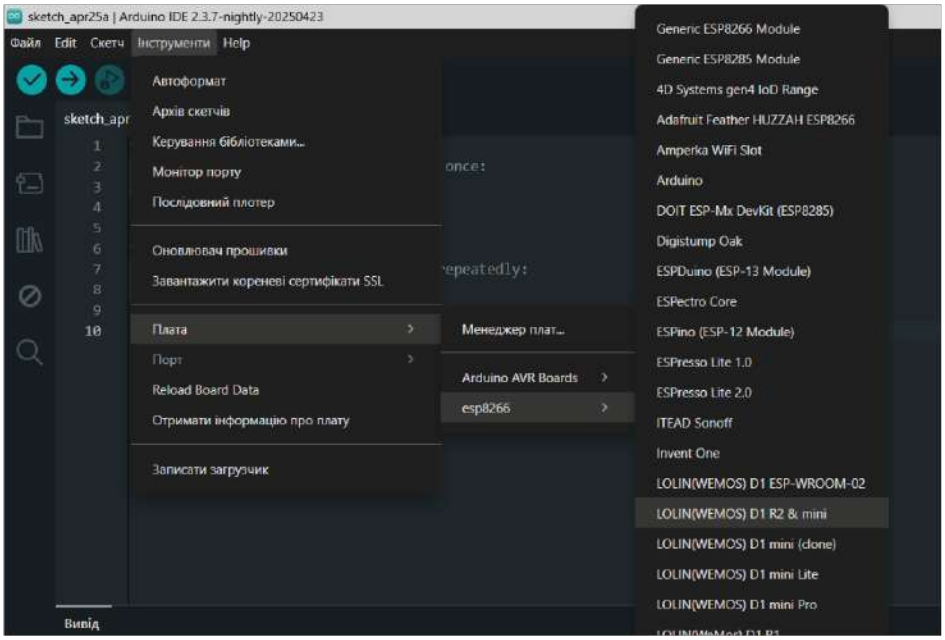

Рисунок 3.7 – Вибір плати

Код розпочинається з оголошення ключових змінних і констант. Ми присвоїли SOUND_SENSOR_PIN аналоговому входу A0, до якого підключено мікрофон KY-037, а BUZZER_PIN – піну D2, який відповідає за керування зумером. Також оголошено змінні soundThreshold та activationDuration – порогове значення та тривалість звучання сигналу відповідно. Ці змінні використовуються для визначення чутливості спрацювання пристрою та забезпечення гнучкості його налаштувань.

Після базової ініціалізації у функції setup() виконується налаштування пінів як вхідних і вихідних, а також ініціалізується Wi-Fi у режимі точки доступу (режим AP), що дозволяє створити локальну мережу з назвою «LostFinder». Це дає змогу керувати пристроєм через мобільний телефон або інший гаджет без потреби в зовнішній інфраструктурі. У рамках налаштування також створюється простий, але функціональний веб-сервер, доступний за IP-адресою 192.168.4.1. Сам веб-інтерфейс дозволяє переглядати поточне значення сигналу з мікрофона, змінювати поріг чутливості, керувати тривалістю звучання зумера, а також вручну активувати або вимикати пристрій. На рисунку 3.8 зображено приклад інтерфейсу вебсторінки.

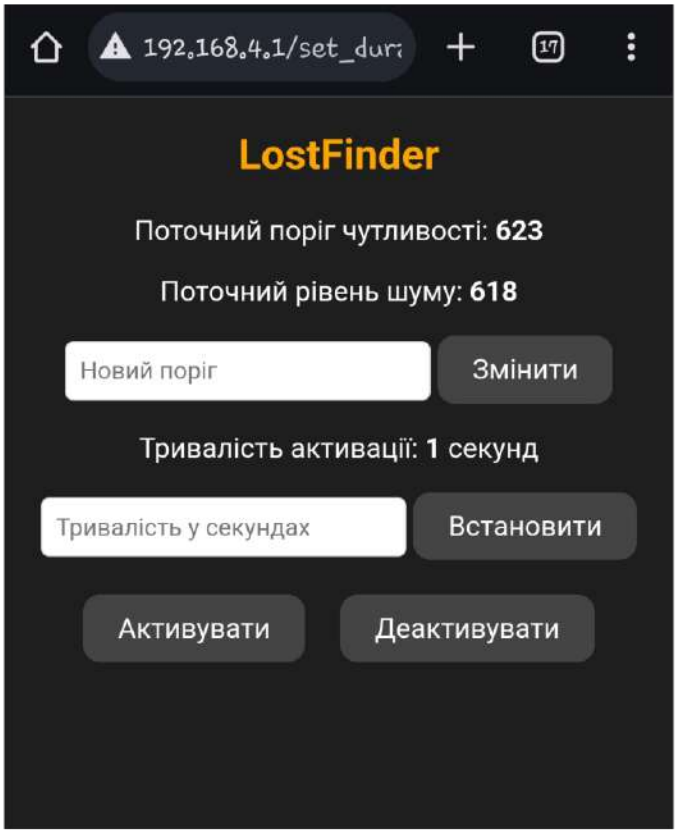

Рисунок 3.8 – Веб-інтерфейс керування пристроєм

Основна логіка програми зосереджена у функції loop(). На кожній ітерації циклу здійснюється зчитування аналогового значення з мікрофона, порівняння його з встановленим порогом, а також визначення, чи потрібно активувати зумер. Якщо значення перевищує поріг і пристрій неактивний, зумер вмикається, а після закінчення заданого часу – вимикається. При цьому індикаторний світлодіод (LED_BUILTIN) також вмикається, сигналізуючи про активний стан.

У коді реалізовано також обробку HTTP-запитів до сервера. Зокрема, функції handleSet() та handleSetDuration() дозволяють динамічно змінювати значення порогу чутливості та тривалості звучання зумера, що значно підвищує зручність користування. Інші функції (handleActivate() та handleDeactivate()) забезпечують ручне вмикання або вимикання зумера зі сторінки керування.

Цей підхід дозволяє не лише спростити інтерфейс користувача, а й забезпечити інтерактивність, не потребуючи фізичних кнопок чи дисплеїв. Особливо важливо, що код організовано структуровано, а всі основні параметри

винесено як змінні, що дозволяє легко змінювати поведінку пристрою без потреби в повному перезаписі програми.

Під час розробки програмного забезпечення особливу увагу було приділено тестуванню. Спочатку функціональність перевірялась на симуляторі в Arduino IDE, де ми мали змогу аналізувати значення сигналу в режимі реального часу. Згодом програму було завантажено в контролер, і тестування продовжилося на реальному пристрої. Під час тестування ми перевіряли реакцію на різні рівні звуку, стабільність мережевого з'єднання, коректність налаштування через веб-інтерфейс та відповідність логіки очікуваній поведінці. Результати показали, що пристрій надійно реагує на гучні звуки, не реагує на фонові шуми, а веб-інтерфейс працює стабільно й без затримок.

Загалом створене програмне забезпечення повністю виконує покладені на нього функції, є масштабованим і може бути розширене додатковими можливостями в майбутньому. Повний програмний код розміщено у додатку А.

3.5 Тестування та аналіз результатів

Після завершення процесу розробки апаратної частини та написання програмного забезпечення наступним важливим етапом стала перевірка працездатності пристрою в реальних умовах. Тестування – це необхідний крок, який дозволяє впевнитися у правильності роботи пристрою, виявити можливі недоліки та провести оптимізацію параметрів для досягнення найкращої ефективності.

Підготовка до тестування розпочалася з налаштування тестового середовища. Як середовище розробки використовувалась Arduino IDE версії 2.1.1, що працювала на операційній системі Windows 11. Прошивка завантажувалась безпосередньо через micro-USB порт контролера WeMos D1 mini. Пристрій тестувався у кількох сценаріях: у кімнаті зі стабільним рівнем шуму, у приміщеннях із високим рівнем фонового шуму, а також на вулиці для імітації різних реальних умов експлуатації.

Перший етап тестування був присвячений перевірці базових функцій: чи правильно зчитуються значення з мікрофона, чи коректно визначається перевищення порогового рівня та чи своєчасно активується зумер. Для цього використовувалися джерела звуку різної інтенсивності: клацання пальцями, плескіт долонями, розмова на нормальному та підвищеному голосі. Результати тестування підтвердили, що при перевищенні встановленого порогу в 600 одиниць пристрій стабільно активував звукову сигналізацію.

Другий етап тестування стосувався роботи веб-інтерфейсу. Після підключення смартфона або ноутбука до створеної Wi-Fi точки доступу «LostFinder» користувач отримував можливість відкривати вебсторінку пристрою. Інтерфейс, раніше показаний, коректно відображав поточний рівень шуму, порогове значення та дозволяв змінювати його без необхідності перезавантаження нової прошивки в контролер. Особливо зручною виявилася функція зміни тривалості спрацювання зумера, яка дозволяла оперативно адаптувати роботу пристрою до конкретних умов.

У процесі тестування вебінтерфейсу перевірялась його стабільність, швидкість оновлення інформації та чутливість до змін. Встановлено, що навіть при частих зміненнях параметрів через вебсторінку пристрій стабільно реагував на команди без перезавантаження або підвисань. Середній час відгуку вебінтерфейсу складав приблизно 300–500 мс, що є прийнятним показником для подібних систем.

Для кращого розуміння ефективності роботи пристрою було складено узагальнену таблицю 3.1 з результатами основних етапів тестування, у якій наведено умови, сценарії використання, очікувані та фактичні результати.

Таблиця 3.1 – Результати тестування IoT-навігатора

№	Сценарій тестування	Очікуваний результат	Фактичний результат	Примітки
1	2	3	4	5
1	Тестування мікрофона в тихому приміщенні	Не спрацьовує при нормальному рівні шуму	Не спрацював	Відповідає вимогам

Продовження таблиці 3.1

1	2	3	4	5
2	Гучне плескання поруч з пристроєм	Спрацьовування зумера	Зумер активувався через ~200 мс	Реакція стабільна
3	Зміна порогового значення у вебінтерфейсі	Оновлення значення в реальному часі	Значення оновлено, пристрій адаптувався	Без перезавантаження
4	Зміна тривалості сигналу через інтерфейс	Зумер змінює тривалість звучання	Зміна тривалості активується одразу	Відповідає функціоналу
5	Час автономної роботи	Мінімум 8 годин на батареї 1000 мАг	9 годин 12 хвилин	У межах норми
6	Відгук вебінтерфейсу при частих запитах	Час відгуку не перевищує 1 секунди	300–500 мс	Стабільно
7	Перевірка помилкових спрацювань	Не реагує на телевізор або фонову розмову	Не реагує	Добра стійкість до шуму

Окрему увагу під час тестування приділили оцінці енергоспоживання пристрою. В умовах підключення акумулятора на 1000 мАг, пристрій показав автономність близько 8-10 годин при середньому рівні активності, що відповідає вимогам до портативних IoT-пристроїв. Для оптимізації енергоспоживання у майбутньому планується реалізація функцій енергозбереження на базі режимів сну ESP8266.

Ще одним важливим етапом стала перевірка стійкості до фонових шумів. У ході тестів було виявлено, що при правильному налаштуванні порогового значення система не спрацьовує на сторонні шуми середньої інтенсивності, такі як розмова або фоновий шум телевізора, проте ефективно реагує на різкі голосові команди або гучні звуки. Це свідчить про достатню стабільність алгоритму обробки даних із мікрофона.

Аналізуючи результати тестування, можна стверджувати, що розроблений IoT-навігатор для пошуку загублених речей повністю справляється із поставленими перед ним завданнями. Він стабільно працює в реальних умовах, забезпечує можливість налаштування ключових параметрів через зручний вебінтерфейс і має достатню автономність для використання без постійного підключення до джерела живлення.

В цілому, розроблений пристрій продемонстрував високу надійність, функціональність та відповідність технічним і експлуатаційним вимогам, що були сформульовані на початковому етапі проєкту. Додаткові можливості оптимізації, такі як впровадження енергозберігаючих режимів або розширення функціоналу вебінтерфейсу, залишають простір для подальшого розвитку пристрою.

ВИСНОВКИ

У процесі розробки було реалізовано прототип IoT-навігатора для пошуку загублених речей, що працює на базі контролера ESP8266. Була побудована компактна апаратна система, яка дозволяє ефективно виявляти гучні звуки та активувати сигнальний зумер, полегшуючи пошук загубленого предмета.

Під час роботи розроблено електричну схему пристрою, яка враховує специфіку роботи всіх компонентів: контролера, сенсора звуку та зумера. У схемі передбачено правильне підключення живлення, аналогових та цифрових сигналів, а також оптимальне використання системних пінів мікроконтролера для забезпечення стабільної роботи.

Було досліджено можливості використання контролера ESP8266, вивчено особливості його пінової розводки, вимоги до початкових станів контактів, режимів роботи Wi-Fi (STA, AP), а також вимоги до енергоспоживання для автономної роботи пристрою.

В ході виконання роботи спроектовано та реалізовано вебінтерфейс, що дозволяє налаштовувати параметри роботи пристрою у реальному часі без необхідності перепрошивки. Розроблений вебінтерфейс дозволяє змінювати порогове значення звуку для активації сигналу та тривалість звучання зумера.

Проведено тестування пристрою в реальних умовах використання. Результати показали, що пристрій стабільно реагує на задані умови активації, демонструє високу енергоефективність та стійкість до фонового шуму завдяки правильно обраному алгоритму обробки сигналу з мікрофона.

Проведено аналіз отриманих результатів: система повністю відповідає сформульованим у вступі вимогам. Були досягнуті всі поставлені цілі, а також запропоновані можливі шляхи для подальшого вдосконалення пристрою.

На основі виконаного дослідження можна зробити висновок, що розроблена система може бути ефективно використана для побутового або персонального використання, а також має перспективи подальшого розвитку та масштабування у рамках рішень для Інтернету речей (IoT).

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кочергін Б., Преварський В., Костючко С. IoT-Навігатор для пошуку загублених речей. *Програмне та апаратне забезпечення в інформаційних технологіях* : зб. тез доп. міжнар. наук.-практ. конф. молодих науковців та студентів, м. Луцьк, 6 травня 2025 р. / Луцький національний технічний університет. Луцьк, 2025. С. 92-93.

2. Як упорядкувати будинок: люди витрачають 6,5 місяців життя на пошуки втрачених речей – Преса Рівне. Преса Рівне. URL: https://pressa.rv.ua/news/yak-uporyadkuvaty-budynok-lyudy-vytrachayut-65-misyacziv-zhyttya-na-poshuky-vtrachenyh-rechej/ (дата звернення: 10.02.2025).

3. Статистика та факти про IoT на 2023 рік (+ прогноз на 10 років). Technique Tips. URL: https://koroglutech.com/uk/articles/31601-iot-statistics--facts-for-2023--10-year-forecast (дата звернення: 10.02.2025).

4. Обмеження AirTags у віддалених районах: труднощі відстеження. CodingMall.com. URL: https://codingmall.com/knowledge-base/25-global/231511-airtag (дата звернення: 13.02.2025).

5. Василишин В. В., Тимощук В. Д., Кітчак Н. Ю., Луцик Н. С. Аналіз характеристик та застосування мікроконтролерів ATTINY85, ATMEGA8, RP2040. *Актуальні задачі сучасних технологій* : зб. тез доп. XII міжнар. наук.-практ. конф. молодих учених та студентів, м. Тернопіль, 6-7 грудня 2023 р. / Тернопільський нац. Тех. Ун-т ім. Івана Пулюя. Тернопіль, 2023. С. 420.

6. Кращі пристрої для пошуку загублених речей: альтернативи Apple AirTag | CyberCalm. CyberCalm | Кіберзахист та кібербезпека простою мовою. URL: https://cybercalm.org/novyny/alternatyvy-apple-airtag/ (дата звернення: 13.02.2025).

7. Що таке мікропроцесор, мікроконтролер та програмований логічний контролер. URL: https://elprivod.nmu.org.ua/ua/interesting/what_is_mp_mc_plc.php (дата звернення: 15.02.2025).

8. Учасники проектів Вікімедіа. Intel 4004 – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Intel_4004 (дата звернення: 15.02.2025).

9. Contributors to Wikimedia projects. Texas Instruments TMS1000 - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Texas_Instruments_TMS1000 (дата звернення: 18.02.2025).

10. Contributors to Wikimedia projects. PIC microcontrollers - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/PIC_microcontrollers (дата звернення: 18.02.2025).

11. Учасники проектів Вікімедіа. Intel MCS-51 – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Intel_MCS-51 (дата звернення: 18.02.2025).

12. The History and Evolution of Arduino – Uniting Digital. Uniting Digital. URL: https://www.unitingdigital.com/articles/the-history-and-evolution-of-arduino (дата звернення: 20.02.2025).

13. Сенсорна електроніка і мікросистемні технології. Репозитарій ОНУ імені І.І.Мечникова elONUar Головна. URL: https://dspace.onu.edu.ua/collections/3f23852a-b61e-4619-861b-a7253dac0d8b (дата звернення: 21.02.2025).

14. Overview and evaluation of bluetooth low energy: an emerging low-power wireless technology. *MDPI*. URL: https://www.mdpi.com/1424-8220/12/9/11734 (дата звернення: 21.02.2025).

15. Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI. *MDPI*. URL: https://www.mdpi.com/2224-2708/2/3/589 (дата звернення: 23.02.2025).

16. Bluetooth low energy mesh networks: a survey. *MDPI*. URL: https://www.mdpi.com/1424-8220/17/7/1467 (дата звернення: 23.02.2025).

17. Bluetooth low energy mesh: applications, considerations and current state-of-the-art / I. Natgunanathan et al. Sensors. 2023. Vol. 23, no. 4. P. 1826. URL: https://doi.org/10.3390/s23041826 (дата звернення: 23.02.2025).

18. Low power wide area networks: an overview. *IEEE Xplore*. URL: https://ieeexplore.ieee.org/document/7815384 (дата звернення: 25.02.2025).

19. Koulouras G., Katsoulis S., Zantalis F. Evolution of bluetooth technology: BLE in the iot ecosystem. Sensors. 2025. Vol. 25, no. 4. P. 996. URL: https://doi.org/10.3390/s25040996 (дата звернення: 25.02.2025).

20. Connectivity technologies in iot: benefits & use cases. webbylab. URL: https://webbylab.com/blog/top-iot-connectivity-technologies/ (дата звернення: 28.02.2025).

21. IoT devices and sensors: types, functions, and use cases - e&ict academy, IIT kanpur. E&ICT Academy, IIT Kanpur - E&ICT Academy, IIT Kanpur. URL: https://eicta.iitk.ac.in/knowledge-hub/internet-of-things/iot-devices-and-sensors-types-functions-and-use-cases/ (дата звернення: 28.02.2025).

22. Prom – найбільший маркетплейс України. prom.ua. URL: https://prom.ua/ua/ (дата звернення: 10.03.2025).

23. Staff L. E. WeMos D1 mini pinout reference - last minute engineers. *Last Minute Engineers*. URL: https://lastminuteengineers.com/wemos-d1-mini-pinout-reference/ (дата звернення: 15.03.2025).

ДОДАТКИ

Додаток А

Код пристрою

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

#define SOUND_SENSOR_PIN A0
#define BUZZER_PIN D2

int soundThreshold = 600;
bool buzzerActive = false;
unsigned long buzzerStartTime = 0;
unsigned long activationDuration = 5000;
int currentSoundValue = 0;

ESP8266WebServer server(80);

// HTML-сторінка
String htmlPage() {
  String page = "<!DOCTYPE html><html><head>";
  page += "<meta charset='UTF-8'>";
  page += "<meta name='viewport' content='width=device-width, initial-scale=1'>";
  page += "<style>";
  page += "body { font-family: Arial, sans-serif; background-color: #1e1e1e;
color: #ffffff; margin: 20px; text-align: center; }";
  page += "h2 { color: #ffa500; }";
  page += ".button-row { display: flex; justify-content: center; gap: 20px;
margin: 20px 0; }";
  page += ".button-row form { display: inline; }";
  page += "button, input[type='submit'] { background-color: #444; color: #fff;
border: none; border-radius: 10px; padding: 10px 20px; font-size: 16px; cursor:
pointer; }";
  page += "button:hover, input[type='submit']:hover { background-color: #ffa500;
}";
  page += "input[type='number'] { border: 1px solid #ccc; border-radius: 5px;
padding: 8px; font-size: 14px; }";
  page += "</style>";
  page += "<script>";
  page += "function updateSound() {";
  page += "  fetch('/sound').then(r => r.text()).then(val => {";
  page += "    document.getElementById('soundVal').innerText = val;";
  page += "  });";
  page += "}";
  page += "setInterval(updateSound, 2000);";
  page += "</script>";
  page += "</head><body>";
  page += "<h2>LostFinder</h2>";
  page += "<p>Поточний поріг чутливості: <b>" + String(soundThreshold) +
"</b></p>";
  page += "<p>Поточний рівень шуму: <b id='soundVal'>" + String(currentSoundValue)
+ "</b></p>";
  page += "<form action='/set'>";
  page += "<input name='thresh' type='number' placeholder='Новий поріг'> ";
  page += "<input type='submit' value='Змінити'>";
  page += "</form>";
  page += "<p>Тривалість активації: <b>" + String(activationDuration / 1000) +
"</b> секунд</p>";
  page += "<form action='/set_duration'>";
  page += "<input name='duration' type='number' placeholder='Тривалість у
секундах'> ";
  page += "<input type='submit' value='Встановити'>";
  page += "</form>";
```

```
  page += "<div class='button-row'>";
  page += "<form action='/activate'><input type='submit'
value='Активувати'></form>";
  page += "<form action='/deactivate'><input type='submit'
value='Деактивувати'></form>";
  page += "</div>";
  page += "</body></html>";
  return page;
}

void handleSet() {
  if (server.hasArg("thresh")) {
    soundThreshold = server.arg("thresh").toInt();
  }
  server.send(200, "text/html", htmlPage());
}

void handleSetDuration() {
  if (server.hasArg("duration")) {
    activationDuration = server.arg("duration").toInt() * 1000;
  }
  server.send(200, "text/html", htmlPage());
}

void handleActivate() {
  buzzerActive = true;
  buzzerStartTime = millis();
  digitalWrite(BUZZER_PIN, HIGH);
  digitalWrite(LED_BUILTIN, LOW);
  server.sendHeader("Location", "/", true);
  server.send(302, "text/plain", "");
}

void handleDeactivate() {
  buzzerActive = false;
  digitalWrite(BUZZER_PIN, LOW);
  digitalWrite(LED_BUILTIN, HIGH);
  server.sendHeader("Location", "/", true);
  server.send(302, "text/plain", "");
}

void handleRoot() {
  server.send(200, "text/html", htmlPage());
}

void setup() {
  pinMode(SOUND_SENSOR_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(BUZZER_PIN, LOW);
  digitalWrite(LED_BUILTIN, HIGH);
  Serial.begin(9600);

  WiFi.softAP("LostFinder", "12345678");
  IPAddress myIP = WiFi.softAPIP();
  Serial.println("Точка доступу запущена. IP-адреса: " + myIP.toString());

  server.on("/", handleRoot);
  server.on("/set", handleSet);
  server.on("/set_duration", handleSetDuration);
  server.on("/activate", handleActivate);
  server.on("/deactivate", handleDeactivate);
  server.on("/sound", []() {
    server.send(200, "text/plain", String(currentSoundValue));
  });
```

```
  server.begin();
  Serial.println("Вебсервер запущено");
}

void loop() {
  server.handleClient();

  currentSoundValue = analogRead(SOUND_SENSOR_PIN);

  if (currentSoundValue > soundThreshold && !buzzerActive) {
    buzzerActive = true;
    buzzerStartTime = millis();
    digitalWrite(BUZZER_PIN, HIGH);
    digitalWrite(LED_BUILTIN, LOW);
    Serial.println("Гучний звук виявлено!");
  }

  if (buzzerActive && millis() - buzzerStartTime > activationDuration) {
    buzzerActive = false;
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(LED_BUILTIN, HIGH);
    Serial.println("Пристрій деактивовано автоматично.");
  }

  delay(100);
}
```