

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 охоронних систем

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**СИСТЕМА АВТОМАТИЗОВАНОГО ПІДРАХУНКУ РЕЗУЛЬТАТІВ
ГОЛОСУВАННЯ НА БАЗІ ESP32 З WEB-ІНТЕРФЕЙСОМ**

**AUTOMATED VOTING RESULTS COUNTING SYSTEM BASED ON
ESP32 WITH WEB INTERFACE**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-41
Пужик Павло Олександрович

(підпис)

Керівник:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Кваліфікаційну роботу
допущено до захисту
«_____» _____ 2026 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2026 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
« 23 » « 12 » 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Пужуку Павлу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Система автоматизованого підрахунку результатів голосування на базі ESP32 з web-інтерфейсом*

Керівник роботи *к.т.н., доцент Лавренчук Світлана Василівна*

затверджені наказом закладу вищої освіти від «20» грудня 2025 року № 536/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 28.05.2026 р.

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, різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Вступ

Аналіз предметної області та наявних рішень

Проектування системи автоматизованого голосування

Практична реалізація системи автоматизованого голосування

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Основні компоненти системи голосування

Алгоритм роботи системи голосування

Схема підключення компонентів системи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області та наявних рішень	Лавренчук С. В., доцент		
Проектування системи автоматизованого голосування	Лавренчук С. В., доцент		
Практична реалізація системи автоматизованого голосування	Лавренчук С. В., 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 23.12.2025 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 10.02.2026 р.	
2.	Аналіз проблеми за темою роботи та постановка завдань дослідження	до 02.03.2026 р.	
3.	Теоретичне дослідження та практична реалізація	до 02.04.2026 р.	
4.	Практична реалізація об'єкта проектування та формування додатків	до 10.04.2026 р.	
5.	Представлення остаточного варіанту кваліфікаційної роботи керівникові	до 01.05.2026 р.	
6.	Нормоконтроль	до 23.05.2026 р.	
7.	Інструментальна перевірка на академічний плагіат	до 26.05.2026 р.	
8.	Здача кваліфікаційної роботи та всіх супровідних документів на кафедру	до 28.05.2026 р.	

Здобувач вищої освіти

(підпис)

Павло ПУЖИК

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Світлана ЛАВРЕНЧУК

(прізвище, ініціали)

АНОТАЦІЯ

Пужик П. О. Система автоматизованого підрахунку результатів голосування на базі ESP32 з web-інтерфейсом. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2026.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

У першому розділі виконано огляд предметної області та обґрунтовано актуальність теми автоматизованого підрахунку результатів голосування. Проаналізовано сучасні системи електронного голосування, їх переваги, недоліки та особливості використання. Також розглянуто принципи роботи хмарних сервісів для зберігання та синхронізації даних.

У другому розділі здійснено вибір та обґрунтування апаратних і програмних засобів розробки системи. Розглянуто характеристики мікроконтролера ESP32, інфрачервоних датчиків HW-201 (YL-63), а також можливості Firebase Realtime Database та веб-технологій HTML, CSS і JavaScript. Виконано проектування структури системи та бази даних.

Третій розділ присвячено реалізації апаратної та програмної частини системи. Розроблено програмне забезпечення для ESP32, реалізовано веб-інтерфейс адміністративної панелі та інтеграцію з Firebase Realtime Database. Проведено тестування системи, перевірено стабільність роботи датчиків, передачу даних та точність підрахунку голосів.

Ключові слова: ESP32, Firebase, голосування, web-інтерфейс, ESP8266, Arduino IDE, автоматизований підрахунок голосів.

ANNOTATION

Puzhyk P. Automated Voting Results Counting System Based on ESP32 with web Interface. Manuscript.

Bachelor's qualification work in the educational program «Computer Engineering» of specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2026.

The qualification work consists of an introduction, three chapters, conclusions, a list of references, and appendices.

The first chapter provides an overview of the subject area and substantiates the relevance of the topic of automated voting results counting. Modern electronic voting systems, their advantages, disadvantages, and features of use are analyzed. The principles of operation of cloud services for data storage and synchronization are also considered.

The second chapter presents the selection and justification of hardware and software tools used for the development of the system. The characteristics of the ESP32 microcontroller, HW-201 (YL-63) infrared sensors, as well as the capabilities of Firebase Realtime Database and web technologies such as HTML, CSS, and JavaScript are described. The structure of the system and database design are also developed.

The third chapter is devoted to the implementation of the hardware and software parts of the system. Software for the ESP32 microcontroller was developed, a web-based administrative interface was implemented, and integration with Firebase Realtime Database was completed. The system was tested to verify sensor stability, data transmission reliability, and vote counting accuracy.

Keywords: ESP32, Firebase, voting, web interface, ESP8266, Arduino IDE, automated vote counting.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ	9
1.1 Поняття та види систем голосування.....	9
1.2 Аналіз існуючих систем онлайн голосування.....	10
1.3 Огляд сучасних технологій для реалізації систем голосування.....	12
РОЗДІЛ 2 ПРОЄКТУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ГОЛОСУВАННЯ.....	15
2.1 Архітектура системи автоматизованого підрахунку голосів	15
2.2 Вибір компонентів для реалізації системи	17
2.3 Принцип роботи та налаштування інфрачервоних датчиків HW-201 (YL-63)	20
2.4 Проєктування бази даних	24
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ.....	26
3.1 Реалізація апаратної частини системи	26
3.2 Реалізація програмної частини ESP	27
3.3 Розробка веб-інтерфейсу системи	31
3.4 Інтеграція з Firebase	35
3.5 Тестування системи	38
3.7 Інструкція з використання системи	42
ВИСНОВКИ.....	45
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	46

ВСТУП

У сучасному світі процеси цифровізації охоплюють практично всі сфери діяльності людини, включаючи системи голосування та обробки результатів. Традиційні методи підрахунку голосів, що базуються на ручній обробці, мають ряд суттєвих недоліків, серед яких низька швидкість обробки даних, можливість виникнення помилок через людський фактор та обмеженість у масштабуванні. У зв'язку з цим актуальним є впровадження автоматизованих систем, які дозволяють забезпечити точність, швидкість та прозорість підрахунку результатів голосування.

Особливого значення набуває використання технологій Інтернету речей (IoT), які дозволяють поєднувати апаратні пристрої з мережевими сервісами. Використання мікроконтролерів та у поєднанні з хмарними базами даних та веб-інтерфейсами, дозволяє створити ефективні системи збору, передачі та обробки інформації в реальному часі. Це забезпечує оперативний доступ до результатів голосування, можливість централізованого керування процесом та підвищує загальну надійність системи.

Крім того, автоматизовані системи голосування дозволяють мінімізувати вплив людського фактору, що є важливим фактором у забезпеченні об'єктивності результатів. Використання сучасних інформаційних технологій сприяє підвищенню рівня довіри до процесу голосування, а також відкриває можливості для подальшого розвитку подібних систем у різних галузях діяльності.

Таким чином, розробка системи автоматизованого підрахунку результатів голосування на базі сучасних технологій є актуальною задачею, що має як теоретичне, так і практичне значення.

Метою роботи є розробка системи автоматизованого підрахунку результатів голосування на базі ESP мікроконтролера з використанням веб-інтерфейсу та хмарної бази даних для забезпечення збору, обробки та відображення результатів у реальному часі.

Об’єкт дослідження – процеси електронного голосування та системи збору і обробки даних.

Предмет дослідження – система автоматизованого підрахунку результатів голосування на базі ESP8266 з використанням веб-інтерфейсу та хмарних технологій.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- спроектувати архітектуру системи, що включає апаратну та програмну частини;
- реалізувати передачу даних з мікроконтролера ESP8266 до хмарної бази даних Firebase;
- розробити веб-інтерфейс адміністратора для створення та управління голосуваннями;
- дослідити роботу системи в реальному часі та оцінити її ефективність;
- візуалізувати результати голосування у зручному для користувача вигляді;
- запропонувати механізм автоматичного визначення активного голосування.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Поняття та види систем голосування

Системи голосування є важливим інструментом прийняття рішень у різних сферах діяльності людини, включаючи політичну, освітню та корпоративну. Основною функцією таких систем є збір, обробка та аналіз інформації щодо вибору учасників з метою визначення кінцевого результату. Від ефективності функціонування систем голосування залежить точність отриманих даних, швидкість їх обробки та рівень довіри до результатів [1].

Традиційні системи голосування базуються на використанні паперових бюлетенів та ручного підрахунку голосів. Незважаючи на свою простоту, такі системи мають низку суттєвих недоліків. Зокрема, процес підрахунку є тривалим та потребує значних людських ресурсів. Крім того, існує ризик виникнення помилок, пов'язаних із людським фактором, що може впливати на достовірність результатів.

Суттєвим недоліком традиційних систем голосування є складність оперативного отримання результатів, особливо при великій кількості учасників голосування. Процес ручного підрахунку потребує значних часових витрат та участі великої кількості персоналу. Крім того, ручна обробка бюлетенів може супроводжуватись помилками під час сортування або підрахунку голосів.

Автоматизовані системи голосування дозволяють значно спростити процес збору та обробки результатів. Використання електронних засобів фіксації голосів забезпечує швидке оновлення даних та зменшує вплив людського фактору. Такі системи можуть використовуватись не лише у виборчих процесах, але й у навчальних закладах, корпоративному середовищі та під час проведення різноманітних опитувань [1].

З розвитком інформаційних технологій широкого застосування набули електронні системи голосування. Вони дозволяють автоматизувати процес збору та обробки даних, що значно підвищує швидкість отримання результатів.

Електронні системи можуть реалізовуватись як у вигляді локальних програмних рішень, так і у вигляді веб-сервісів, доступних через мережу Інтернет [2].

Окрему категорію становлять автоматизовані системи голосування, які поєднують апаратні та програмні компоненти. У таких системах процес фіксації голосу здійснюється за допомогою спеціальних пристроїв, а обробка та зберігання даних виконується автоматично. Це дозволяє мінімізувати участь людини в процесі підрахунку та зменшити ймовірність помилок [2].

Сучасні тенденції розвитку систем голосування пов'язані із застосуванням технологій Інтернету речей. Такі системи використовують мікроконтролери, сенсори та мережеві сервіси для збору та передачі даних у реальному часі. Це забезпечує високу швидкість обробки інформації та можливість віддаленого контролю за процесом голосування [3].

1.2 Аналіз існуючих систем онлайн голосування

На сучасному етапі розвитку інформаційних технологій існує значна кількість різноманітних систем голосування, які відрізняються за принципами функціонування, рівнем автоматизації та сферою застосування. Найбільш поширеними є онлайн-системи голосування, що реалізуються у вигляді веб-платформ та дозволяють користувачам брати участь у голосуванні дистанційно через мережу Інтернет.

Серед сучасних онлайн-платформ голосування варто виділити такі сервіси, як Google Forms, SurveyMonkey, Mentimeter, Slido, Poll Everywhere, StrawPoll та LimeSurvey. Дані платформи забезпечують можливість створення опитувань, автоматичного збору результатів та їх подальшої обробки у зручному вигляді.

Платформа Google Forms є одним із найпростіших інструментів для створення голосувань. Вона дозволяє швидко формувати анкети, збирати відповіді та зберігати результати у вигляді таблиць. Основною перевагою є безкоштовність та інтеграція з іншими сервісами, проте функціональні можливості обмежені у порівнянні з професійними системами [4].

Сервіс SurveyMonkey є більш розвиненою платформою, орієнтованою на проведення професійних досліджень. Він надає широкі можливості для аналізу результатів, налаштування опитувань та обробки даних. Водночас значна частина функціоналу доступна лише за умови платної підписки [5].

Для інтерактивного голосування у режимі реального часу використовуються платформи Mentimeter та Slido. Вони дозволяють проводити опитування під час заходів, миттєво відображати результати та взаємодіяти з аудиторією. Такі системи широко застосовуються у навчанні та під час презентацій [6].

Платформа Poll Everywhere також використовується для інтерактивного голосування та отримувати відповіді через мобільні пристрої або веб-інтерфейс. Вона підтримує різні типи запитань та забезпечує швидке відображення результатів [7].

Сервіс StrawPoll є прикладом простого інструменту для швидкого створення голосувань без реєстрації. Його основною перевагою є швидкість створення опитування, однак він не забезпечує достатнього рівня безпеки та контролю [8].

Окрему категорію становлять системи з відкритим вихідним кодом, такі як LimeSurvey. Вони дозволяють розгортати систему на власному сервері, що забезпечує більший контроль над даними та можливість гнучкого налаштування. Однак такі рішення потребують технічних знань для їх використання [9].

Незважаючи на значну кількість сучасних онлайн-платформ, більшість із них орієнтовані виключно на програмну реалізацію процесу голосування. Такі системи переважно не взаємодіють із фізичними пристроями або сенсорами, що обмежує можливість їх використання у системах автоматизованого підрахунку голосів на базі апаратних компонентів.

Крім того, значна частина існуючих платформ потребує постійного підключення до мережі Інтернет та залежить від роботи зовнішніх серверів. У випадку нестабільного мережевого з'єднання це може призводити до затримок оновлення результатів або тимчасової недоступності системи. Саме тому

актуальним є створення спеціалізованих систем, які поєднують апаратні та програмні засоби збору і передачі даних.

1.3 Огляд сучасних технологій для реалізації систем голосування

Сучасні системи автоматизованого голосування базуються на використанні комплексу технологій, які забезпечують збір, передачу, обробку та відображення даних. До основних компонентів таких систем належать мікроконтролери, мережеві технології, хмарні сервіси та веб-інтерфейси користувача. Важливою особливістю сучасних рішень є їх здатність працювати у режимі реального часу, що дозволяє оперативно отримувати результати голосування та аналізувати їх. Крім того, використання сучасних технологій дозволяє значно підвищити точність системи та зменшити вплив людського фактору на результати голосування.

Одним із основних елементів апаратної частини є мікроконтролери, які використовуються для обробки сигналів від сенсорів та передачі даних до серверів. У даній роботі доцільно використовувати мікроконтролер ESP8266, який має вбудований модуль Wi-Fi та підтримує роботу з мережевими протоколами. Це дозволяє реалізувати передачу даних без необхідності використання додаткового обладнання. Завдяки компактним розмірам та простоті програмування, ESP8266 широко застосовується у проєктах Інтернету речей, що робить його оптимальним вибором для систем голосування.

ESP8266 характеризується низьким енергоспоживанням, доступною вартістю та достатньою обчислювальною потужністю для реалізації задач збору даних. Завдяки підтримці протоколів HTTP та TCP/IP, він може безпосередньо взаємодіяти з веб-сервісами та хмарними платформами. Крім того, мікроконтролер підтримує роботу з різними бібліотеками, що значно спрощує процес розробки програмного забезпечення. Це дозволяє швидко створювати прототипи та тестувати систему у реальних умовах [10].

Для фіксації голосів у системі можуть використовуватись різні типи сенсорів, зокрема інфрачервоні (ІЧ) датчики. Вони дозволяють визначати наявність об'єкта у зоні дії датчика та формувати відповідний сигнал для мікроконтролера. Використання ІЧ-сенсорів є доцільним через їх простоту, надійність та невисоку вартість. Крім того, такі датчики мають високу швидкість реакції, що дозволяє точно фіксувати кожен голос без затримок.

Інфрачервоні сенсори працюють за принципом випромінювання та прийому інфрачервоного сигналу, який відбивається від об'єкта. При появі об'єкта сигнал змінюється, що фіксується контролером як подія. Такий принцип роботи забезпечує стабільність функціонування навіть у різних умовах освітлення. Це робить ІЧ-датчики універсальним рішенням для використання у системах автоматизованого підрахунку голосів [11].

Передача даних від мікроконтролера до серверної частини здійснюється за допомогою мережових технологій. Найчастіше використовується протокол HTTP або HTTPS, який дозволяє надсилати запити до віддалених серверів. Використання таких протоколів забезпечує стандартизований спосіб обміну даними між клієнтом і сервером. Це дозволяє легко інтегрувати різні компоненти системи між собою.

Захищене з'єднання HTTPS є особливо важливим у системах голосування, оскільки забезпечує шифрування переданих даних та підвищує рівень безпеки. Це дозволяє запобігти перехопленню інформації або її підміні під час передачі. Крім того, використання сучасних протоколів забезпечує сумісність системи з більшістю веб-сервісів та платформ [12].

Для зберігання та обробки результатів голосування доцільно використовувати хмарні бази даних. Одним із популярних рішень є Firebase Realtime Database, яка дозволяє зберігати дані у форматі JSON та забезпечує їх миттєву синхронізацію між клієнтами. Це дає можливість відображати результати голосування у режимі реального часу. Завдяки цьому користувачі можуть одразу бачити зміни результатів після кожного голосу.

Firebase Realtime Database має ряд переваг, серед яких простота інтеграції, відсутність необхідності створення власного серверу та підтримка масштабування. Крім того, платформа надає механізми автентифікації та контролю доступу, що є важливим для забезпечення безпеки системи. Використання хмарних технологій дозволяє значно спростити розробку та зменшити витрати на підтримку інфраструктури [13].

Для реалізації інтерфейсу використовуються веб-технології, такі як HTML, CSS та JavaScript. Вони дозволяють створювати зручні та функціональні веб-додатки, які забезпечують взаємодію користувача з системою. Використання цих технологій дозволяє створити кросплатформене рішення, яке працює на різних пристроях без необхідності встановлення додаткового програмного забезпечення [14].

РОЗДІЛ 2

ПРОЄКТУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ГОЛОСУВАННЯ

2.1 Архітектура системи автоматизованого підрахунку голосів

Архітектура системи автоматизованого підрахунку голосів визначає взаємодію її основних компонентів, які забезпечують збір, передачу, обробку та відображення результатів голосування. Розроблена система складається з апаратної та програмної частин, що функціонують як єдине ціле.

Апаратна частина системи включає мікроконтролер, сенсори для фіксації голосів та елементи підключення. У якості центрального керуючого елемента використовується мікроконтролер ESP8266, який виконує обробку сигналів від датчиків та передачу даних до віддаленого сервера. Для фіксації голосів застосовуються інфрачервоні датчики, встановлені в урнах для голосування, що дозволяє автоматично визначати факт голосування.

Програмна частина системи реалізована з використанням хмарної бази даних Firebase Realtime Database та веб-інтерфейсу користувача. Firebase використовується для зберігання результатів голосування та забезпечення їх синхронізації у режимі реального часу. Веб-інтерфейс виконує функції відображення результатів, створення нових голосувань та управління ними через адміністративну панель.

Взаємодія між компонентами системи здійснюється через бездротову мережу Wi-Fi. Після фіксації голосу датчиком відповідний сигнал передається на мікроконтролер, який обробляє подію та формує запит до бази даних. Далі інформація передається до Firebase, де зберігається та стає доступною для відображення на веб-сайті. Такий підхід дозволяє забезпечити безперервний обмін даними між апаратною та програмною частинами системи. Крім того, використання бездротового з'єднання спрощує розгортання системи та забезпечує її гнучкість у різних умовах експлуатації.

Для кращого розуміння структури системи доцільно розглянути основні її компоненти, наведені у таблиці 2.1. Представлена таблиця дозволяє

систематизувати інформацію про складові системи та їх функціональне призначення. Це сприяє більш чіткому уявленню про архітектуру розробленого рішення та взаємозв'язки між його елементами.

Таблиця – 2.1 Основні компоненти системи голосування

Компонент	Призначення
ESP8266	Обробка сигналів від датчиків та передача даних до сервера
Інфрачервоний датчик	Фіксація факту голосу
Wi-Fi модуль	Передача даних до мережі інтернет
Firebase Realtime Database	Зберігання та синхронізація результатів
Веб-сайт (адмін панель)	Управління голосуванням та відображення результатів

Функціонування системи базується на послідовному виконанні операцій, пов'язаних із обробкою сигналів та передачею даних. Для опису логіки роботи системи використовується алгоритм, представлений на рисунку 2.1.

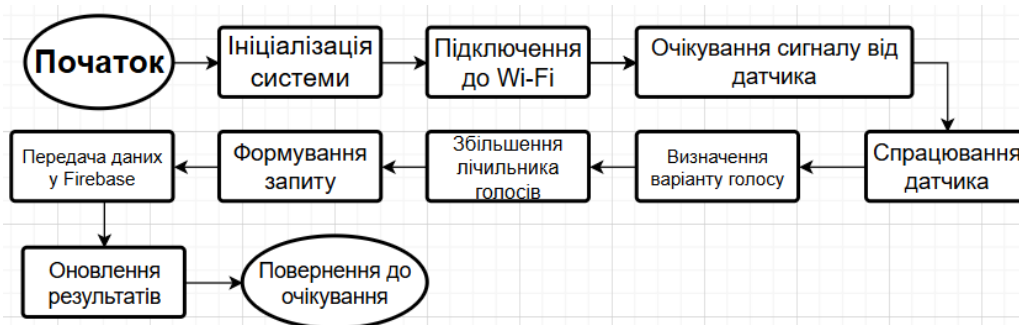


Рисунок 2.1 – Алгоритм роботи системи голосування

Цей алгоритм демонструє основні етапи роботи системи. Після запуску виконується ініціалізація та підключення до мережі Wi-Fi. Далі система переходить у режим очікування сигналу від датчиків. У разі спрацювання одного з датчиків визначається відповідний варіант голосу, після чого відбувається збільшення лічильника голосів та передача даних до бази.

2.2 Вибір компонентів для реалізації системи

Для реалізації системи автоматизованого підрахунку голосів було обрано ряд апаратних та програмних компонентів, які забезпечують стабільну роботу системи, простоту реалізації та доступність. Вибір кожного елемента здійснювався з урахуванням технічних характеристик, вартості та можливості інтеграції з іншими складовими системи.

Основним елементом апаратної частини є мікроконтролер ESP8266, який використовується для обробки сигналів від датчиків та передачі даних до хмарної бази. Даний мікроконтролер має вбудований модуль Wi-Fi, що дозволяє забезпечити бездротову передачу інформації без використання Wi-Fi додаткових пристроїв. Це значно спрощує конструкцію системи та зменшує її вартість.

До основних переваг ESP8266 належать низьке енергоспоживання, компактні розміри, доступна ціна та достатня обчислювальна потужність для виконання поставлених задач. Крім того, мікроконтролер підтримує роботу з протоколами HTTP та HTTPS, що дозволяє легко інтегрувати його. Завдяки широкій підтримці у вигляді бібліотек та документації, ESP8266 є одним із найпопулярніших рішень для проєктів Інтернету речей. Зовнішній вигляд мікроконтролера ESP8266 представлено на рисунку 2.2.



Рисунок 2.2 – EPS8266 ESP-12F [15]

Для фіксації голосів у системі обрано інфрачервоні датчики типу HW-201 (YL-63), які використовуються для виявлення об'єктів на невеликій відстані.

Даний датчик працює за принципом відбиття інфрачервоного сигналу від поверхні об'єкта, що дозволяє визначити його наявність у зоні дії сенсора.

Основними перевагами датчика HW-201 (YL-63) є простота використання, низька вартість та достатня точність для задач підрахунку голосів. Датчик формує цифровий сигнал, який може безпосередньо зчитуватися мікроконтролером, що спрощує процес обробки даних. Крім того, наявність регулятора чутливості дозволяє налаштовувати датчик під конкретні умови експлуатації.

Використання саме інфрачервоних датчиків є доцільним, оскільки вони забезпечують швидку реакцію на появу об'єкта та не потребують складної обробки сигналів. Це дозволяє мінімізувати затримки у роботі системи та забезпечити точний підрахунок голосів.

Зовнішній вигляд інфрачервоного датчика HW-201 (YL-63) представлено на рисунку 2.3.



Рисунок 2.3 – HW-201(YL-63) датчик перешкод [16]

Для зберігання та обробки результатів голосування обрано хмарну базу даних Firebase Realtime Database. Дане рішення дозволяє зберігати дані у вигляді структурованих записів та забезпечує їх миттєву синхронізацію між клієнтами.

У Firebase є можливість роботи у режимі реального часу, що дозволяє одразу відображати результати голосування на веб-інтерфейсі. Крім того, дана платформа не потребує розгортання власного серверу, що значно спрощує

розробку системи та зменшує витрати часу (детальніше про базу даних у підрозділі 2.3) [17].

Для реалізації користувацького інтерфейсі використано веб-технології, зокрема HTML, CSS та JavaScript. Це дозволяє створити адміністративну панель для управління голосуванням та перегляду результатів. Веб-інтерфейс забезпечує доступ до системи з будь-якого пристрою, що має підключення до мережі Інтернет [18].

Щоб розмістити веб-інтерфейс у системі буде використовуватись сервіс GitHub Pages, який дозволяє публікувати статичні веб-сайти без необхідності використання окремого серверного обладнання. Використання даного сервісу забезпечує постійний доступ до веб-інтерфейсу через мережу Інтернет та спрощує процес оновлення програмного коду. Крім того, GitHub забезпечує зручне зберігання проєкту та контроль версій, що є важливим під час розробки програмних систем [19].

Програмування мікроконтролера ESP8266 здійснювалось із використанням середовища розробки Arduino IDE, яке забезпечує зручний інтерфейс для написання, компіляції та завантаження програмного коду. Дане середовище є одним із найпоширеніших інструментів для розробки вбудованих систем та підтримує широкий спектр бібліотек і плат, що є перевагою для програмування ESP8266 [20].

Для роботи з мікроконтролером ESP8266 використовуються спеціалізовані бібліотеки, які дозволяють реалізувати підключення до Wi-Fi мережі та взаємодію з веб-сервісами. Це значно спрощує процес розробки та забезпечує стабільну роботу системи. Використання готових бібліотек також дозволяє зменшити кількість програмного коду та прискорити процес створення програмного забезпечення для мікроконтролера. Крім того, підтримка великої кількості відкритих бібліотек для ESP8266 забезпечує можливість швидкої інтеграції додаткових функцій, таких як обмін даними з хмарними сервісами або робота з сенсорами [21].

2.3 Принцип роботи та налаштування інфрачервоних датчиків HW-201 (YL-63)

Для реалізації системи автоматизованого підрахунку голосів важливим елементом є інфрачервоні датчики HW-201 (YL-63), які використовуються для фіксації факту проходження об'єкта через зону дії сенсора. Дані датчики дозволяють автоматично визначати появу бюлетеня або іншого об'єкта в урні для голосування та формувати відповідний сигнал для мікроконтролера ESP8266. Датчики даного типу характеризуються швидкою реакцією на зміну стану контрольованої зони та простотою інтеграції з мікроконтролерними системами. Завдяки цифровому виходу модуль може безпосередньо підключатися до входів ESP8266 без використання додаткових схем обробки сигналів.

Інфрачервоні датчики широко використовуються в автоматизованих системах завдяки простоті конструкції, низькій вартості та достатній точності роботи. Основною перевагою використання HW-201 (YL-63) є можливість швидкого визначення об'єкта без механічного контакту, що забезпечує надійність роботи системи та зменшує зношування компонентів. Крім того, використання безконтактного способу виявлення дозволяє підвищити довговічність системи та знизити ймовірність механічних пошкоджень. Невеликі розміри датчика також спрощують його встановлення всередині урн для голосування.

Принцип роботи датчика базується на використанні інфрачервоного випромінювання. Модуль містить два основних елементи: інфрачервоний світлодіод, який випромінює ІЧ-сигнал, та фотоприймач, який приймає відбитий сигнал (рис. 2.4). У нормальному стані випромінювання поширюється у навколишньому середовищі. При появі об'єкта частина інфрачервоного випромінювання відбивається від його поверхні та потрапляє на фотоприймач. Після цього формується електричний сигнал, який обробляється схемою датчика та передається на мікроконтролер. Час спрацювання сенсора є достатньо малим,

що дозволяє фіксувати об'єкти практично миттєво після їх появи у зоні дії датчика. Це особливо важливо для систем автоматизованого підрахунку голосів, де необхідно забезпечити точну та швидку реєстрацію кожного голосу.

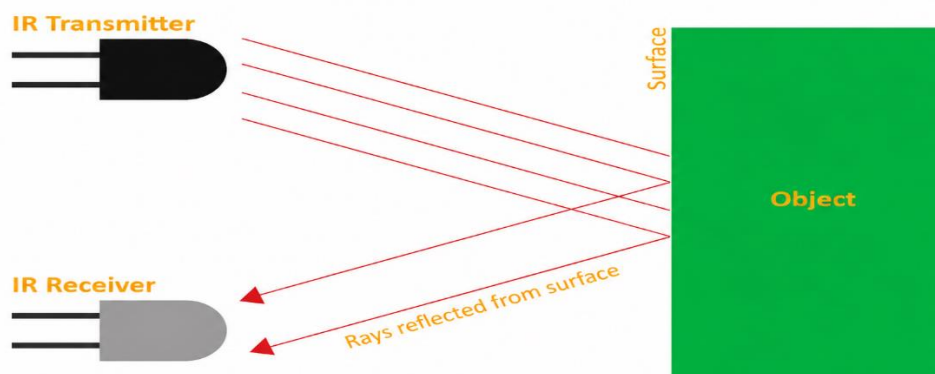


Рисунок 2.4 – Принцип роботи інфрачервоного датчика HW-201 (YL-63)
[22]

Важливу роль у роботі датчика відіграє компаратор LM393 (рис. 2.5), який входить до складу модуля. Його основним призначенням є порівняння рівня сигналу, отриманого від фотоприймача, із заданим пороговим значенням. Якщо рівень сигналу перевищує встановлений поріг, компаратор формує цифровий вихідний сигнал, який зчитується мікроконтролером як факт спрацювання датчика.

Використання компаратора дозволяє підвищити стабільність роботи сенсора та зменшити вплив електричних перешкод. Крім того, цифровий вихід значно спрощує програмну обробку сигналів у системі автоматизованого голосування.

Конструкція датчика HW-201 (YL-63) містить регулювальний резистор (потенціометр), який використовується для налаштування чутливості сенсора. За допомогою зміни положення резистора можна встановлювати відстань спрацювання датчика та рівень чутливості до відбитого інфрачервоного сигналу [23].

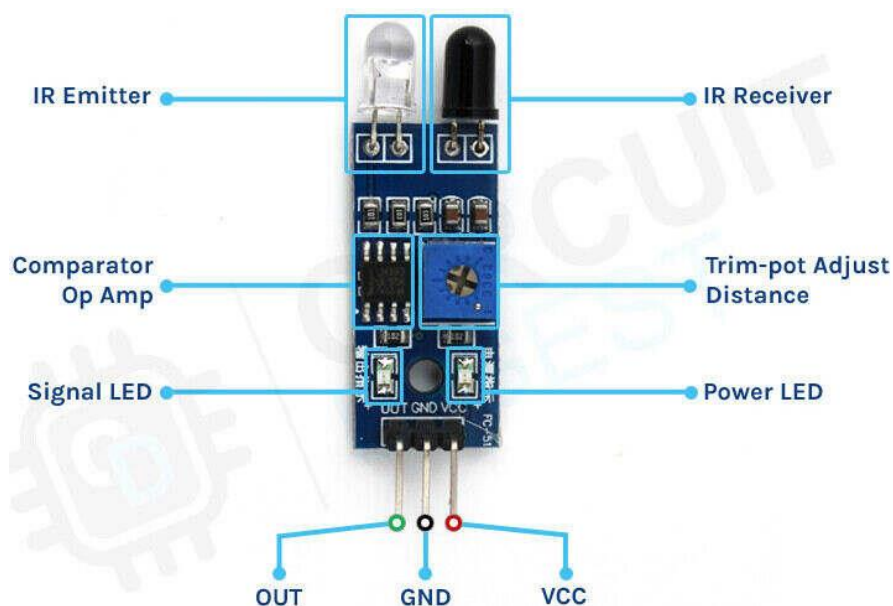


Рисунок 2.5 – Основні елементи датчика HW-201 (YL-63) [24]

Налаштування чутливості є важливим етапом під час підготовки системи до роботи. При надто високій чутливості датчик може реагувати на сторонні об'єкти, зміни освітлення або паразитні відбиття інфрачервоного випромінювання. Це може призводити до хибних спрацювань та неправильного підрахунку голосів. У випадку занадто низької чутливості можливе пропускання сигналів від об'єктів, що також негативно впливає на точність роботи системи.

Регулювання чутливості виконується шляхом обертання потенціометра викруткою. При обертанні за годинниковою стрілкою зазвичай збільшується дальність виявлення об'єкта, а при обертанні у протилежному напрямку – зменшується. Під час налаштування необхідно забезпечити стабільне спрацювання датчика лише при проходженні об'єкта через визначену зону контролю.

У процесі тестування системи було встановлено оптимальну чутливість датчиків для умов роботи урн голосування. Це дозволило мінімізувати кількість помилкових спрацювань та забезпечити стабільну реєстрацію голосів.

На точність роботи інфрачервоного датчика впливають також зовнішні фактори, серед яких освітлення, колір поверхні об'єкта та відстань до нього. Світлі поверхні краще відбивають інфрачервоне випромінювання, тоді як темні поверхні поглинають його значну частину. Крім того, пряме сонячне світло може створювати додаткові інфрачервоні перешкоди та впливати на стабільність роботи сенсора.

Для забезпечення стабільної роботи системи датчики були встановлені всередині урн таким чином, щоб мінімізувати вплив зовнішнього освітлення. Також було підібрано оптимальну відстань між датчиком та зоною проходження бюлетеня.

Основні технічні характеристики датчика HW-201 (YL-63) наведено у таблиці 2.2.

Таблиця – 2.2 Характеристики датчика HW-201 (YL-63)

Характеристика	Значення
Робоча напруга	3.3- 5 В
Тип сигналу	Цифрови
Дальність виявлення	2-30 см
Наявність регулювання чутливості	Так
Тип компаратора	LM393
Тип сенсора	Інфрачервоний

Використання інфрачервоних датчиків HW-201 (YL-63) у системі автоматизованого підрахунку голосів дозволило реалізувати простий та ефективний механізм фіксації голосування. Завдяки можливості налаштування чутливості та швидкій реакції сенсорів забезпечується стабільна робота системи та висока точність підрахунку результатів [23].

2.4 Проектування бази даних

Для зберігання та обробки результатів голосування у розробленій системі використовується хмарна база даних Firebase Realtime Database. Ця база даних забезпечує зберігання інформації у форматі JSON та підтримує синхронізацію даних у режимі реального часу. Це дозволяє миттєво оновлювати результати голосування на веб-інтерфейсі після надходження нових даних від мікроконтролера ESP8266.

База даних організована у вигляді ієрархічної структури, що забезпечує зручне зберігання інформації про голосування, варіанти відповідей та результати підрахунку голосів. Такий підхід дозволяє спростити доступ до даних та забезпечити високу швидкість їх обробки.

Структура бази даних (рис. 2.6) складається з двох основних вузлів верхнього рівня: polls та activePoll, polls містить інформацію про всі створені голосування, activePoll містить ідентифікатор активного голосування, яке використовується мікроконтролером для запису голосів.

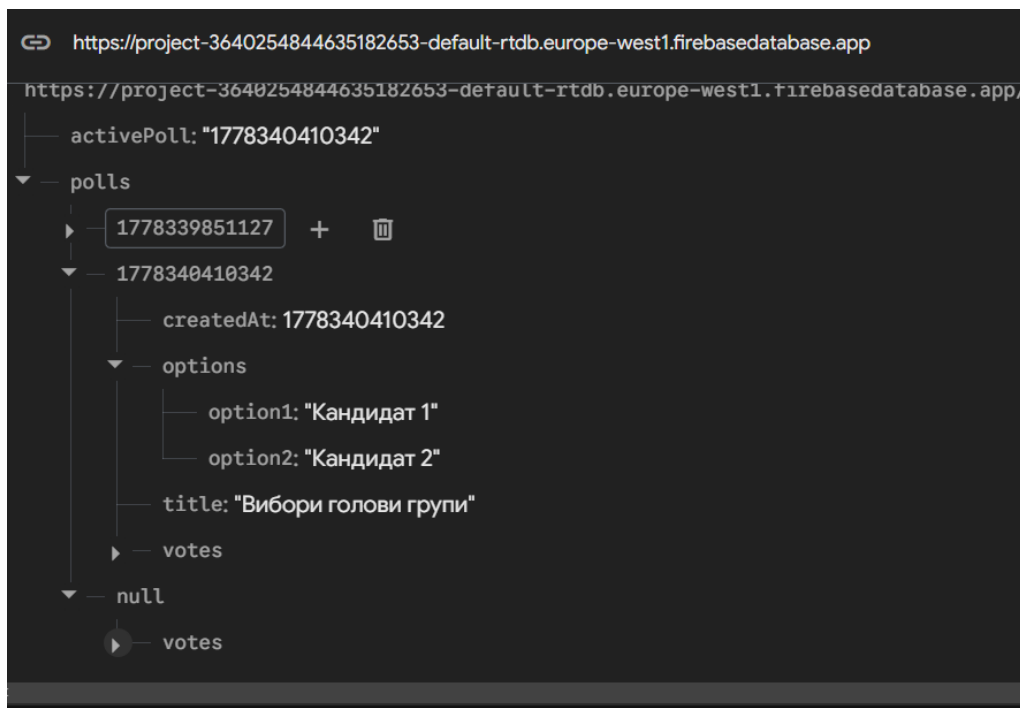


Рисунок 2.6 – Структура бази даних Firebase Realtime Database

Кожне голосування має власний унікальний ідентифікатор, який автоматично створюється під час додавання нового запису до бази даних. Це дозволяє зберігати необмежену кількість голосувань та забезпечує можливість їх подальшого перегляду через адміністративну панель.

Для автоматизації роботи системи використовується вузол activePoll, який містить ідентифікатор поточного активного голосування. Мікроконтролер ESP8266 отримує значення даного параметра перед відправкою голосів та автоматично записує результати у відповідний вузол бази даних. Це дозволяє змінювати активне голосування без необхідності внесення змін у програмний код мікроконтролера.

Зберігання результатів голосування реалізовано у вузлі votes, який містить кількість голосів для кожного варіанта відповіді, а також загальну кількість голосів. Для оновлення значень використовується механізм інкрементування, що забезпечує коректну зміну даних навіть при одночасному надходженні декількох запитів.

Організація бази даних у вигляді ієрархічної структури забезпечує простоту масштабування системи. За необхідності можна додавати нові голосування, додаткові параметри або нові функції без суттєвої зміни структури даних.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ

3.1 Реалізація апаратної частини системи

Для реалізації апаратної частини системи було використано мікроконтролер ESP8266 та два інфрачервоні датчики HW-201 (YL-63). Підключення компонентів виконано за допомогою макетної плати та з'єднувальних провідників.

Інфрачервоні датчики використовуються для фіксації голосів користувачів. Кожен датчик відповідає окремому варіанту голосування та підключений до цифрових входів мікроконтролера. Перший датчик підключено до контакту D1, другий до контакту D2.

Живлення датчиків здійснюється від мікроконтролера ESP8266 через контакти 3,3 V та GND. Для забезпечення стабільної роботи системи використовується бездротове підключення до мережі Wi-Fi, через яке виконується передача даних до Firebase Realtime Database.

Схему підключення компонентів представлено на рисунку 3.1.

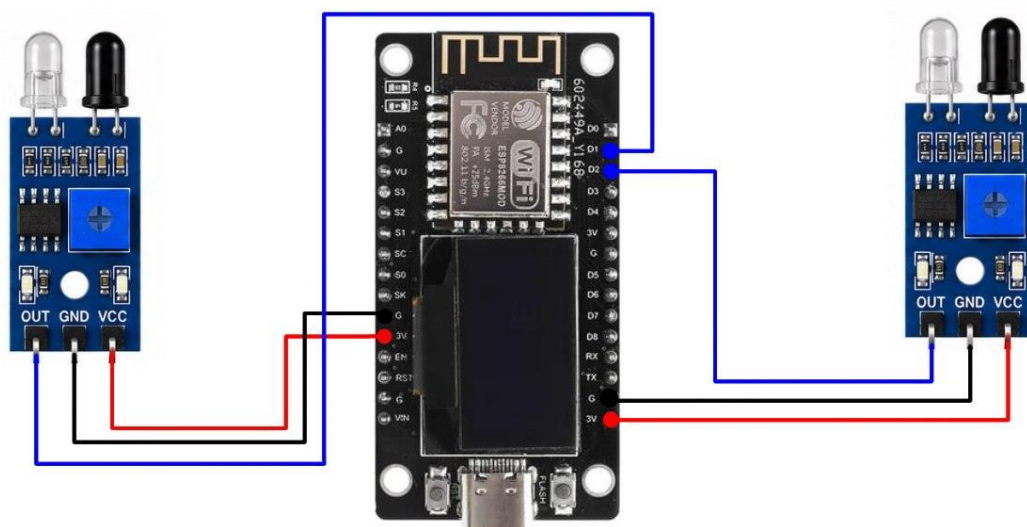


Рисунок 3.1 – Схема підключення компонентів системи

Основні піни для підключення компонентів наведено у таблиці 3.1.

Таблиця 3.1 – Підключення компонентів системи

Компонент	Пін підключення
Датчик HW-201 №1	D1
Датчик HW-201 №2	D2
Живлення датчиків	3,3 V
Загальний провід	GND

Після складання апаратної частини було проведено тестування роботи датчиків та мікроконтролера. Під час тестування перевірялась коректність спрацювання сенсорів, стабільність підключення до Wi-Fi та передача голосів до бази даних.

Для уникнення багаторазового спрацювання датчиків у програмному коді реалізовано механізм `debounce`, який забезпечує затримку між повторними сигналами. Це дозволяє підвищити точність підрахунку голосів та уникнути помилок у роботі системи.

3.2 Реалізація програмної частини ESP

Програмна частина системи реалізована у середовищі Arduino IDE із використанням мови програмування C++. Основним призначенням програмного забезпечення є обробка сигналів від інфрачервоних датчиків, підключення до мережі Wi-Fi та передача результатів голосування до Firebase Realtime Database.

Для роботи системи використовуються бібліотеки `ESP8266WiFi`, `ESP8266HTTPClient` та `WiFiClientSecureBearSSL`, які забезпечують підключення до бездротової мережі та виконання HTTPS-запитів до Firebase.

На початку програмного коду задаються параметри бездротової мережі Wi-Fi та адреса Firebase Realtime Database. Після запуску мікроконтролер автоматично виконує підключення до точки доступу та переходить у робочий режим.

У лістингу 3.1 представлено фрагмент програмного коду налаштування Wi-Fi та Firebase Realtime Database.

Лістинг 3.1 – фрагмент програмного коду налаштування Wi-Fi та Firebase Realtime Database.

```
const char* WIFI_SSID = «iPhone»;
const char* WIFI_PASSWORD = «00000000»;

const char* FIREBASE_HOST =
»project-3640254844635182653-default-rtdb.europe-
west1.firebaseio.com»;
```

кінець лістингу 3.1

Після успішного підключення до мережі мікроконтролер починає опитування інфрачервоних датчиків, підключених до контактів D1 та D2. Перший датчик використовується для реєстрації голосів за перший варіант відповіді, другий – за другий варіант.

Для уникнення помилкових спрацювань у програмі реалізовано механізм debounce. Після фіксації сигналу система ігнорує повторне спрацювання датчика протягом заданого проміжку часу. Це дозволяє уникнути багаторазового зарахування одного голосу та підвищує точність роботи системи.

Особливістю реалізованої системи є автоматичне визначення активного голосування. Для цього у Firebase Realtime Database використовується окремий параметр activePoll, який містить ідентифікатор поточного голосування.

У лістингу 3.2 представлено фрагмент програмного коду обробки сигналів датчиків.

Лістинг 3.2 – Фрагмент програмного коду обробки сигналів датчиків

```
int s1 = digitalRead(SENSOR_1);
int s2 = digitalRead(SENSOR_2);

unsigned long now = millis();

int vote1 = 0;
int vote2 = 0;
```

```

if (s1 == HIGH && lastState1 == LOW &&
    now - last1 > DEBOUNCE) {

    vote1 = 1;
    last1 = now;

    Serial.println(«Vote option1»);
}

if (s2 == HIGH && lastState2 == LOW &&
    now - last2 > DEBOUNCE) {

    vote2 = 1;
    last2 = now;

    Serial.println(«Vote option2»);
}

lastState1 = s1;
lastState2 = s2;

```

кінець лістингу 3.2

Мікроконтролер періодично надсилає GET-запит до Firebase та отримує значення параметра activePoll. Отриманий ідентифікатор використовується для автоматичного визначення вузла бази даних, до якого необхідно записувати результати голосування.

Такий підхід дозволяє змінювати активне голосування через адміністративну панель веб-сайту без необхідності повторного програмування ESP8266.

У лістингу 3.3 представлено фрагмент програмного коду отримання параметра activePoll.

Лістинг 3.3 – Фрагмент програмного коду отримання параметра activePoll

```

String getActivePoll() {

```

```

    std::unique_ptr<BearSSL::WiFiClientSecure>
    client(new BearSSL::WiFiClientSecure);

    client->setInsecure();

    HTTPClient http;

```

```

String url =
«https://» + String(FIREBASE_HOST) +
«/activePoll.json»;

if (!http.begin(*client, url)) {

    Serial.println(«GET failed»);

    return «»;
}

int code = http.GET();

String payload = http.getString();

http.end();

payload.replace(«\»«, «»);

return payload;
}

```

кінець лістингу 3.3

Після спрацювання одного з датчиків формується JSON-структура із даними для оновлення кількості голосів у Firebase Realtime Database. Передача інформації здійснюється за допомогою PATCH-запиту через HTTPS-з'єднання. Такий підхід забезпечує безпечну передачу даних між мікроконтролером ESP8266 та хмарною базою даних, а також дозволяє оперативно оновлювати результати голосування у режимі реального часу.

У програмі використовується механізм *increment*, який дозволяє автоматично збільшувати значення лічильників голосів без перезапису всієї структури даних. Одночасно оновлюється кількість голосів за відповідного кандидата та загальна кількість голосів. Використання даного механізму дозволяє зменшити навантаження на базу даних та підвищує стабільність роботи системи під час багаторазового надходження сигналів від датчиків.

У лістингу 3.4 представлено фрагмент програмного коду формування PATCH-запиту до Firebase.

Лістинг 3.4 – фрагмент програмного коду формування PATCH-запиту до Firebase.

```
String url =
«https://» + String(FIREBASE_HOST) +
«/polls/» + activePollID +
«/votes.json»;

String payload = «{»;

if (v1 > 0) {

    payload +=
    «\»option1\»: {\».sv\»: {\»increment\»:1}}»;
}

if (v2 > 0) {

    payload +=
    «\»option2\»: {\».sv\»: {\»increment\»:1}}»;
}

payload +=
«,\»total\»: {\».sv\»: {\»increment\»:»
+ String(v1 + v2) + «}}»;

payload += «}»;

int code =
http.sendRequest(«PATCH», payload);
```

кінець лістингу 3.4

Для контролю роботи системи використовувався послідовний монітор Arduino IDE. У ньому відображалась інформація про підключення до мережі Wi-Fi, отримання параметра activePoll, надсилання HTTP-запитів та результати передачі даних до Firebase. Використання послідовного монітора дозволило оперативно виявляти можливі помилки під час роботи системи, контролювати стабільність мережевого підключення та перевіряти коректність передачі голосів до бази даних.

3.3 Розробка веб-інтерфейсу системи

Для взаємодії користувача із системою автоматизованого підрахунку голосів було розроблено веб-інтерфейс, який забезпечує створення голосувань, перегляд результатів та керування активним голосуванням у режимі реального часу.

Веб-інтерфейс реалізовано з використанням HTML, CSS та JavaScript. Для синхронізації даних використовується Firebase Realtime Database, а розміщення веб-сайту виконано за допомогою сервісу GitHub Pages.

Структура веб-проєкту складається з декількох основних файлів, кожен з яких виконує окрему функцію у роботі системи. Основними файлами є index.html, create.html, active.html та style.css.

Файл index.html є головною сторінкою адміністративної панелі системи. На даній сторінці реалізовано навігацію між основними функціями веб-інтерфейсу, зокрема створенням голосувань та переглядом активних голосувань. Головна сторінка забезпечує швидкий доступ до всіх функцій системи та спрощує роботу адміністратора.

У лістингу 3.5 представлено фрагмент програмного коду головної сторінки.

Лістинг 3.5 – Фрагмент програмного коду головної сторінки

```
<!DOCTYPE html>
<html>
<head>
  <title>Адмін панель</title>
  <link rel=«stylesheet» href=«style.css»>
</head>

<body>

<div class=«container»>

  <h1>ПАНЕЛЬ АДМІНІСТРАТОРА</h1>
  <p>Ласкаво просимо, admin</p>

  <ul class=«menu»>

    <li><a href=«create.html»>Створити нове голосування</a></li>

    <li><a href=«active.html»>Переглянути активні голосування</a></li>
```

```

    <li><a href=«login.html»>Вийти</a></li>
  </ul>
</div>

```

кінець лістингу 3.5

Файл create.html використовується для створення нових голосувань. На сторінці реалізовано форму введення назви голосування та варіантів відповідей. Після створення нового голосування інформація автоматично записується до Firebase Realtime Database із формуванням унікального ідентифікатора.

У лістингу 3.6 представлено фрагмент програмного коду створення нового голосування.

Лістинг 3.6 – Фрагмент програмного коду створення нового голосування

```

const pollId = Date.now();

set(ref(db, «polls/» + pollId), {
  title: title,
  options: {
    option1: option1,
    option2: option2
  },
  votes: {
    option1: 0,
    option2: 0,
    total: 0
  },
  createdAt: pollId});

```

кінець лістингу 3.6

Після створення голосування інформація автоматично зберігається у Firebase Realtime Database та стає доступною для подальшого використання у системі. Для кожного голосування формується окремий вузол із власним унікальним ідентифікатором, що дозволяє зберігати необмежену кількість голосувань. Такий підхід забезпечує впорядковане зберігання даних та спрощує подальшу обробку результатів голосування. Крім того, використання унікальних ідентифікаторів дозволяє уникнути конфліктів між різними голосуваннями.

Файл `active.html` використовується для перегляду активних голосувань та результатів голосування у режимі реального часу. На сторінці реалізовано відображення назв голосувань, кількості голосів для кожного кандидата, а також загальної кількості голосів. Оновлення результатів відбувається автоматично без необхідності перезавантаження сторінки браузера. Це забезпечує зручний контроль за процесом голосування з боку адміністратора системи.

Крім перегляду результатів, сторінка `active.html` забезпечує можливість зміни активного голосування. Для цього використовується параметр `activePoll`, який зберігається у `Firebase Realtime Database`. Після вибору нового активного голосування значення параметра автоматично оновлюється, а `ESP8266` починає передавати голоси до нового вузла бази даних. Завдяки цьому адміністратор може швидко перемикає голосування без перепрограмування мікроконтролера. Такий механізм значно спрощує процес керування системою під час її експлуатації.

У лістингу 3.7 представлено фрагмент програмного коду встановлення активного голосування.

Лістинг 3.7 – Фрагмент коду встановлення активного голосування

```

window.setActivePoll = function(id) {

    set(ref(db, «activePoll»), id)
    .then(() => {
        alert(«Активне голосування змінено»);
    });
}

```

кінець лістингу 3.7

Для забезпечення автоматичного оновлення результатів голосування використовується функція `onValue()`, яка дозволяє отримувати актуальні дані з `Firebase` у режимі реального часу без необхідності перезавантаження сторінки.

У лістингу 3.8 представлено фрагмент програмного коду отримання даних із `Firebase Realtime Database`.

 Лістинг 3.8 – Фрагмент програмного коду отримання даних із Firebase

```
onValue(ref(db, «polls»), (snapshot) => {

  const data = snapshot.val();

  container.innerHTML = ««;

});
```

кінець лістингу 3.8

На сторінці `active.html` також реалізовано функцію видалення голосувань. Після натискання відповідної кнопки вибране голосування видаляється з Firebase Realtime Database та автоматично зникає з адміністративної панелі.

У лістингу 3.9 представлено фрагмент програмного коду видалення голосування.

 Лістинг 3.9 – Фрагмент програмного коду видалення голосування

```
window.deletePoll = function(id) {

  if (confirm(«Видалити голосування?»)) {
    remove(ref(db, «polls/» + id));
  }

};
```

кінець лістингу 3.9

Для оформлення веб-інтерфейсу використовується файл `style.css`, який забезпечує стилізацію сторінок адміністративної панелі (повний код `style.css` можна переглянути у додатку А). За допомогою CSS реалізовано оформлення кнопок, форм введення, навігаційного меню та блоків із результатами голосування. Використання окремого файлу стилів дозволило забезпечити єдиний дизайн усіх сторінок веб-інтерфейсу та спростило подальше редагування зовнішнього вигляду системи.

3.4 Інтеграція з Firebase

Для забезпечення зберігання та синхронізації даних у системі автоматизованого підрахунку голосів використовується Firebase Realtime Database. Дана хмарна база даних забезпечує обмін інформацією між мікроконтролером ESP8266 та веб-інтерфейсом у режимі реального часу. Використання Firebase дозволило реалізувати централізоване зберігання голосувань, автоматичне оновлення результатів та керування активним голосуванням через адміністративну панель.

Після створення нового голосування через веб-інтерфейс інформація автоматично записується до Firebase Realtime Database. Для кожного голосування формується окремий вузол із власним унікальним ідентифікатором, який створюється автоматично на основі поточного часу. У створеному вузлі зберігається назва голосування, варіанти відповідей, результати голосування та час створення запису.

У лістингу 3.10 представлено фрагмент програмного коду запису нового голосування до Firebase.

Лістинг 3.10 – Фрагмент програмного коду запису даних до Firebase

```
set(ref(db, «polls/» + pollId), {
  title: title,
  options: {
    option1: option1,
    option2: option2
  },
  votes: {
    option1: 0,
    option2: 0,
    total: 0
  },
  createdAt: pollId
});
```

кінець лістингу 3.10

Після спрацювання одного з інфрачервоних датчиків мікроконтролер ESP8266 формує PATCH-запит та передає результати голосування до Firebase Realtime Database. Як вже було зазначено, для оновлення результатів використовується механізм `increment`, який дозволяє автоматично збільшувати

значення лічильників голосів без необхідності повного перезапису вузла бази даних. Такий підхід забезпечує стабільну роботу системи навіть при швидкому надходженні декількох сигналів.

У лістингу 3.11 представлено фрагмент програмного коду оновлення результатів голосування.

Лістинг 3.11 – Фрагмент програмного коду оновлення результатів голосування

```
String url = «https://» + String(FIREBASE_HOST)
+ «/polls/» + activePollID + «/votes.json»;
```

```
int code = http.sendRequest(«PATCH», payload);
```

кінець лістингу 3.11

Для отримання актуальних результатів голосування веб-інтерфейс використовує функцію `onValue()`, яка забезпечує автоматичне оновлення даних у режимі реального часу. Після зміни інформації у Firebase результати голосування автоматично відображаються на веб-сторінці без необхідності ручного оновлення сторінки браузера. Це дозволяє забезпечити постійне відображення актуальних результатів голосування для адміністратора системи.

У лістингу 3.12 представлено фрагмент програмного коду читання даних із Firebase Realtime Database.

Лістинг 3.12 – Фрагмент програмного коду читання даних із Firebase

```
onValue(ref(db, «polls»), (snapshot) => {
```

```
    const data = snapshot.val();
```

```
    container.innerHTML = ««;
```

```
});
```

кінець лістингу 3.12

Для автоматизації роботи системи використовується параметр `activePoll`, який зберігається у Firebase Realtime Database. Даний параметр містить ідентифікатор активного голосування, до якого мікроконтролер ESP8266 записує

результати голосування. Під час роботи система періодично виконує GET-запит до Firebase та отримує актуальне значення параметра activePoll. Це дозволяє змінювати активне голосування через веб-інтерфейс без внесення змін у програмний код мікроконтролера.

У лістингу 3.13 представлено фрагмент програмного коду отримання значення параметра activePoll.

Лістинг 3.13 – Фрагмент програмного коду отримання activePoll

```
String url = «https://» + String(FIREBASE_HOST)
+ «/activePoll.json»;
```

```
int code = http.GET();
```

```
String payload = http.getString();
```

кінець лістингу 3.13

Інтеграція з Firebase забезпечує стабільний обмін даними між усіма компонентами системи. Використання Firebase Realtime Database дозволило реалізувати оновлення результатів голосування у режимі реального часу, автоматичне перемикавання активного голосування та централізоване зберігання інформації у межах єдиної хмарної бази даних.

3.5 Тестування системи

Після завершення розробки апаратної та програмної частини було проведено тестування системи автоматизованого підрахунку голосів. Основною метою тестування була перевірка коректності роботи інфрачервоних датчиків, стабільності передачі даних до Firebase Realtime Database та точності підрахунку голосів.

На першому етапі тестування виконувалась перевірка роботи інфрачервоних датчиків HW-201 (YL-63). Під час тестування перевірялась коректність спрацювання датчиків при появі об'єкта у зоні дії сенсора (рис. 3.6).

Також перевірялась швидкість реакції системи та стабільність формування цифрового сигналу для мікроконтролера ESP8266.

```

12:18:02.151 -> HTTP: 200
12:18:05.102 -> ActivePoll: 1778923072071
12:18:09.049 -> ActivePoll: 1778923072071
12:18:11.284 -> Vote option2
12:18:12.319 -> HTTP: 200
12:18:13.279 -> ActivePoll: 1778923072071
12:18:17.260 -> ActivePoll: 1778923072071
12:18:21.235 -> ActivePoll: 1778923072071
12:18:23.207 -> Vote option1
12:18:24.246 -> HTTP: 200

```

Рисунок 3.6 – Тестування роботи інфрачервоних датчиків HW-201 (YL-63)

У процесі тестування було встановлено, що датчики стабільно реагують на появу об'єкта та коректно передають сигнали до ESP8266. Для уникнення повторних спрацювань у програмному коді було реалізовано механізм *debounce*, який забезпечує затримку між повторними сигналами від одного датчика. Це дозволило підвищити точність роботи системи та уникнути помилкового зарахування декількох голосів під час одного спрацювання. Крім того, під час тестування перевірялась робота датчиків при різних умовах освітлення та різних відстані до об'єкта. Отримані результати підтвердили стабільну роботу сенсорів у межах робочих параметрів системи.

Наступним етапом було тестування стабільності передачі даних до Firebase Realtime Database через мережу Wi-Fi (рис. 3.7). Під час тестування перевірялась швидкість оновлення результатів голосування у веб-інтерфейсі та стабільність HTTPS-з'єднання між ESP8266 та Firebase. У ході перевірки було встановлено, що результати голосування оновлюються практично миттєво після спрацювання датчиків. Також система продемонструвала стабільну роботу при тривалому безперервному функціонуванні та багаторазовій передачі даних до Firebase.

У ході тестування система передавала результати голосування до Firebase у режимі реального часу. Після спрацювання датчиків значення лічильників голосів автоматично оновлювались у базі даних в реальному часі та відображались на веб-інтерфейсі без необхідності ручного оновлення сторінки.

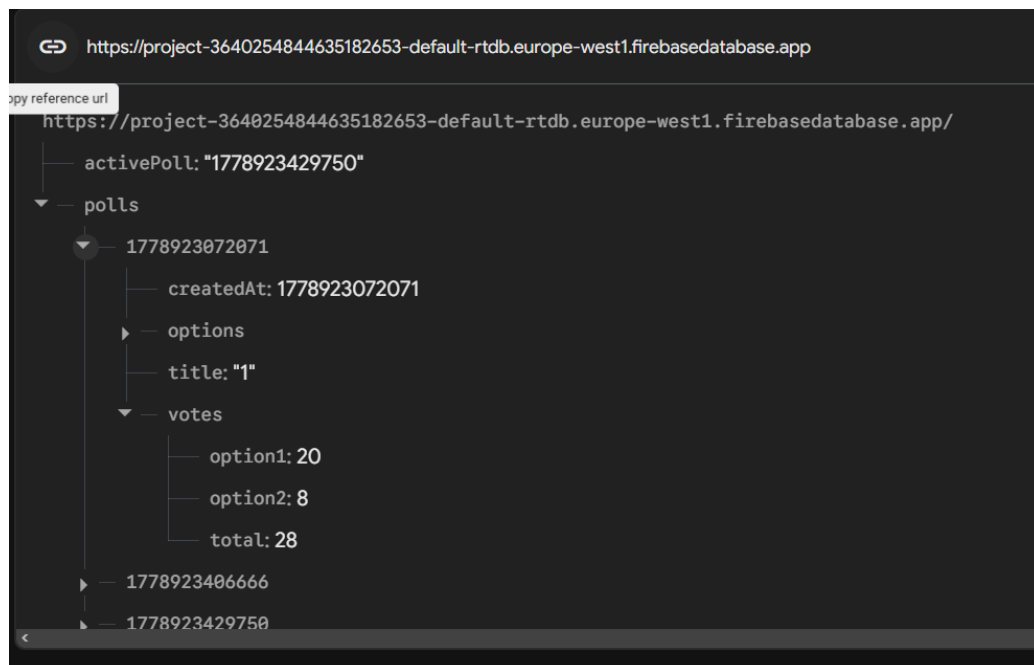


Рисунок 3.7 – Відображення результатів голосування у Firebase Realtime Database

Окрему увагу було приділено перевірці механізму автоматичного перемикання активного голосування (рис. 3.8).

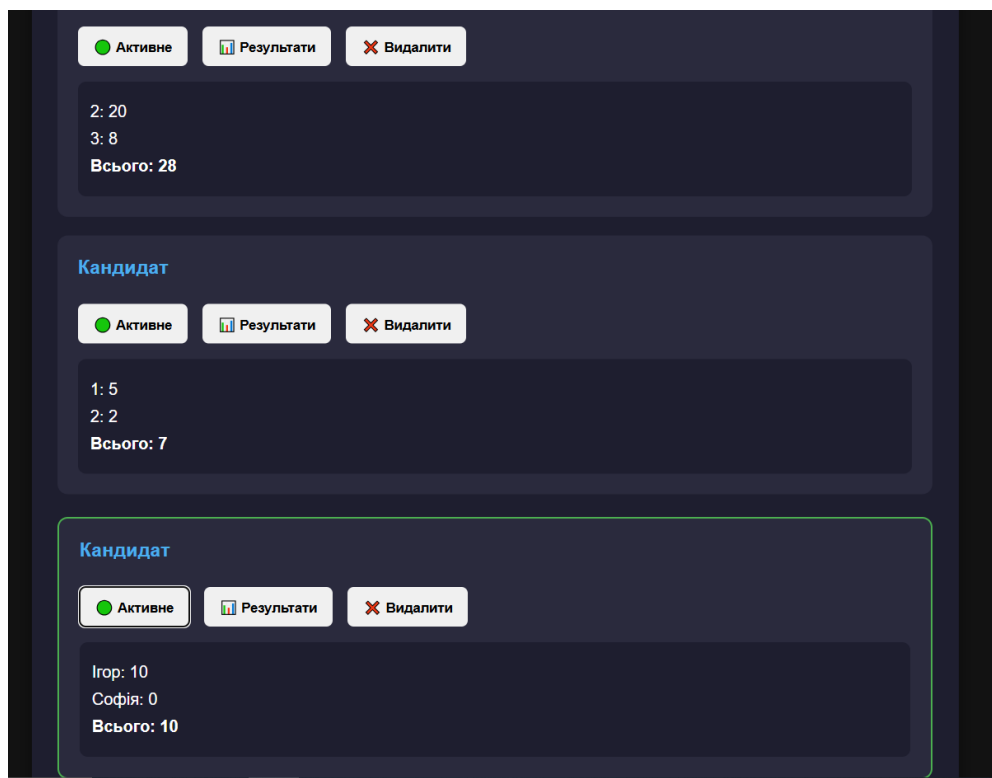


Рисунок 3.8 – Зміна активного голосування через адміністративну панель

Під час тестування через адміністративну панель змінювався параметр activePoll, після чого ESP8266 автоматично починав запис голосів до нового вузла бази даних. Проведена перевірка показала коректну роботу механізму без необхідності перепрограмування мікроконтролера.

Для перевірки точності підрахунку голосів було проведено серію тестових голосувань із різною кількістю спрацювань датчиків (рис. 3.9). Отримані результати порівнювались із фактичною кількістю поданих голосів. У результаті тестування встановлено, що система коректно підраховує голоси та забезпечує правильне оновлення результатів у Firebase Realtime Database. Під час проведення тестів не було виявлено втрати голосів або помилкового збільшення значень лічильників. Це підтвердило коректність роботи програмної логіки та механізму обробки сигналів від датчиків.

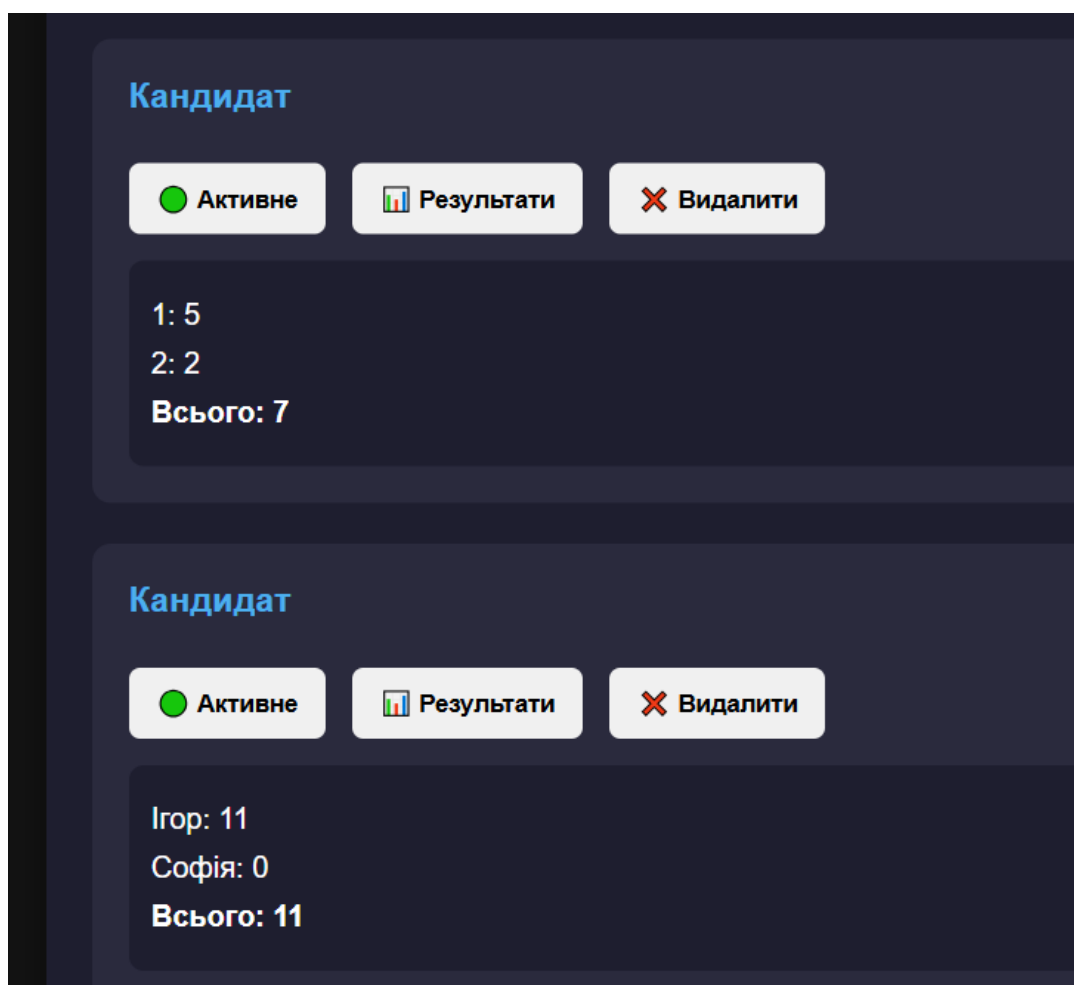


Рисунок 3.9 – Веб-інтерфейс системи під час тестування

Під час тестування також перевірялась робота веб-інтерфейсу системи. Було протестовано створення нових голосувань, перегляд результатів, видалення голосувань та зміну активного голосування через адміністративну панель. Усі функції веб-інтерфейсу працювали коректно та забезпечували стабільну взаємодію з Firebase Realtime Database. Крім того, перевірка показала зручність використання адміністративної панелі для керування процесом голосування.

Результати проведеного тестування підтвердили працездатність розробленої системи автоматизованого підрахунку голосів. Система обробляє сигнали від датчиків та передає інформацію до Firebase Realtime Database, забезпечує оновлення результатів голосування у режимі реального часу.

3.7 Інструкція з використання системи

Для роботи системи автоматизованого підрахунку голосів необхідно забезпечити підключення мікроконтролера ESP8266 до мережі Wi-Fi та доступ до Firebase Realtime Database. Перед початком роботи потрібно завантажити програмний код до мікроконтролера за допомогою середовища Arduino IDE. Під час завантаження прошивки необхідно перевірити правильність вибору COM-порту та моделі плати у середовищі Arduino IDE, оскільки неправильні налаштування можуть призвести до помилок компіляції або неможливості запису програми до пам'яті мікроконтролера.

Після запуску ESP8266 автоматично підключається до вказаної бездротової мережі та отримує значення активного голосування з Firebase Realtime Database. Якщо активне голосування відсутнє, передача голосів не здійснюється. У випадку неправильного введення назви мережі Wi-Fi або пароля мікроконтролер не зможе встановити підключення до мережі, тому перед запуском системи необхідно перевірити правильність параметрів підключення у програмному коді. Також необхідно забезпечити стабільний сигнал Wi-Fi, оскільки нестабільне підключення може спричинити затримки передачі даних або тимчасову втрату зв'язку з Firebase.

Для створення нового голосування необхідно відкрити веб-інтерфейс адміністративної панелі через браузер. На головній сторінці потрібно перейти до розділу створення голосування та ввести назву голосування і варіанти відповідей. Після підтвердження введені дані автоматично записуються до Firebase Realtime Database. Перед створенням нового голосування рекомендується перевіряти правильність введених даних, оскільки після початку голосування зміна параметрів може призвести до некоректного відображення результатів.

Для початку голосування необхідно перейти до сторінки активних голосувань та вибрати потрібне голосування. Після натискання кнопки встановлення активного голосування система автоматично оновлює параметр activePoll, а ESP8266 починає передавати результати голосування до відповідного вузла бази даних. Якщо активне голосування не змінюється,

необхідно перевірити наявність підключення до Firebase та коректність структури бази даних.

Під час голосування інфрачервоні датчики HW-201 (YL-63) фіксують проходження об'єкта та передають сигнал до ESP8266. Мікроконтролер обробляє отриманий сигнал і надсилає інформацію про голос до Firebase Realtime Database. Для уникнення помилкових спрацювань необхідно правильно налаштувати чутливість датчиків за допомогою регульовального резистора на модулі. Також слід уникати потрапляння прямого сонячного світла або сторонніх джерел інфрачервоного випромінювання, які можуть впливати на стабільність роботи сенсорів.

Результати голосування автоматично оновлюються у веб-інтерфейсі в режимі реального часу. Для перегляду результатів голосування необхідно відкрити сторінку активних голосувань, де відображається кількість голосів для кожного варіанта та загальна кількість поданих голосів. У випадку відсутності оновлення результатів необхідно перевірити підключення до мережі Інтернет та коректність роботи Firebase Realtime Database.

Після завершення голосування адміністратор може змінити активне голосування або видалити непотрібні голосування через веб-інтерфейс системи. Перед видаленням голосування рекомендується переконатися у необхідності виконання цієї дії, оскільки видалені дані не можуть бути автоматично відновлені.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено систему автоматизованого підрахунку результатів голосування на базі мікроконтролера ESP8266 із використанням веб-інтерфейсу та Firebase Realtime Database. Система забезпечує автоматичну фіксацію голосів за допомогою інфрачервоних датчиків, передачу даних через мережу Wi-Fi та відображення результатів голосування у режимі реального часу.

У процесі роботи було проведено аналіз сучасних систем електронного голосування та визначено основні вимоги до розроблюваної системи. Для реалізації апаратної частини використано ESP8266 та інфрачервоні датчики HW-201 (YL-63), а для програмної частини – HTML, CSS, JavaScript і Firebase Realtime Database.

Було спроектовано структуру бази даних та реалізовано програмне забезпечення для ESP8266, яке забезпечує обробку сигналів датчиків і передачу результатів голосування до Firebase за допомогою HTTPS-запитів. Для підвищення точності роботи системи реалізовано механізм debounce, який дозволяє уникнути повторного спрацювання датчиків.

Також було розроблено веб-інтерфейс адміністративної панелі, який дозволяє створювати голосування, переглядати результати, видаляти голосування та змінювати активне голосування у системі.

У ході тестування перевірено стабільність роботи системи, точність підрахунку голосів та коректність оновлення даних у режимі реального часу. Результати тестування підтвердили працездатність розробленої системи та можливість її використання для проведення локальних голосувань у навчальних закладах та організаціях.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сажина, М.Р. Електронне голосування як елемент електронної демократії: світовий досвід та Україна. Електронний репозитарій КРС ЧНУ ім. Петра Могили. 2025. URL: <https://krs.chmnu.edu.ua/jspui/handle/123456789/4499> (дата звернення: 05.04.2026).
2. Новіков, О.В., Литвинова Т.В. Впровадження дуальної системи голосування у виборчу систему України. DSpace : ELAKPI : Репозитарій КПІ ім. Ігоря Сікорського. 2022. URL: <https://ela.kpi.ua/handle/123456789/51475> (дата звернення: 05.04.2026).
3. Жураковський, Б.Ю., Зенів, І.О. ТЕХНОЛОГІЇ ІНТЕРНТУ РЕЧЕЙ. ПВНЗ «МЕГУ імені академіка Степана Дем'янчука»: 2021 URL: <https://library.megu.edu.ua:9443/jspui/handle/123456789/4212> (дата звернення: 05.04.2026).
4. Google | Google Workspace. Google Workspace. URL: <https://workspace.google.com/intl/uk/products/forms/> (date accessed: 05.04.2026).
5. SurveyMonkey: The World's Most Popular Survey Platform. SurveyMonkey. URL: <https://www.surveymonkey.com> (date accessed: 05.04.2026).
6. Interactive presentation & audience engagement tool – Mentimeter Interactive presentation & audience engagement tool – Mentimeter. URL: <https://www.mentimeter.com> (date accessed: 05.04.2026).
7. Engage your audience | Poll Everywhere. Engage your audience | Poll Everywhere. URL: <https://www.poll.everywhere.com> (date accessed: 05.04.2026).
8. StrawPoll – Create a Poll in Seconds. StrawPoll. URL: <https://strawpoll.com> (date accessed: 05.04.2026).
9. Створюйте опитування: LimeSurvey Безкоштовний інструмент для опитувань. LimeSurvey – Free Online Survey Tool. URL: <https://www.limesurvey.org/uk> (date accessed: 05.04.2026).
10. ESP8266 Wi-Fi SoC. Espressif Systems. URL: <https://www.espressif.com/en/products/socs/esp8266> (date accessed: 05.04.2026).

11. IR Sensor Module. Components101. URL: <https://components101.com/sensors/ir-sensor-module> (date accessed: 05.04.2026).
12. HTTP: Hypertext Transfer Protocol. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP> (date accessed: 05.04.2026).
13. Firebase Realtime Database. Firebase. URL: <https://firebase.google.com/docs/database> (date accessed: 05.04.2026).
14. W3Schools.com. W3Schools Online Web Tutorials. URL: <https://www.w3schools.com/html/> (date accessed: 05.04.2026).
15. Amazon.ae. URL: <https://www.amazon.ae/Development-Wireless-Micropython-Programming-Soldered/dp/B0BVM3H46B> (date accessed: 06.04.2026).
16. HW-201/YL-63 датчик перешкод Arduino. Мій Проект. URL: <https://myproject.com.ua/hw201-yl63-datchik-pereshkod-ua.html> (дата звернення: 06.04.2026).
17. Firebase Realtime Database. Firebase. URL: <https://firebase.google.com/docs/database> (date accessed: 06.04.2026).
18. W3Schools.com. W3Schools Online Web Tutorials. URL: <https://www.w3schools.com> (date accessed: 06.04.2026).
19. GitHub Pages documentation. GitHub Docs. URL: <https://docs.github.com/en/pages> (date accessed: 05.04.2026).
20. Arduino – Home. URL: <https://www.arduino.cc/en/software> (date accessed: 06.04.2026).
21. Welcome to ESP8266 Arduino Core's documentation! - ESP8266 Arduino Core documentation. URL: <https://arduino-esp8266.readthedocs.io/en/latest/> (date accessed: 06.04.2026).
22. HW-201/YL-63 датчик перешкод | 3,3-5V, 230 см Arduino. Мій Проект. URL: <https://myproject.com.ua/hw201-yl63-datchik-pereshkod-ua.html> (дата звернення: 06.04.2026).
23. HW201 Infrared (IR) Sensor Module. Circuits DIY. URL: <https://www.circuits-diy.com/hw201-infrared-ir-sensor-module/> (date accessed: 05.04.2026).

24. Assignment-11. Fab Academy. URL: <https://surl.li/hhlfqf> (date accessed: 05.04.2026).

