

Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки

**Бакалаврський дипломний проєкт на тему**  
Система віддаленого збору даних засобами ІОТ

Виконав: студент 4 курсу, групи ІСП-206  
спеціальності

123 Комп'ютерна інженерія,  
(шифр і назва напрямку підготовки, спеціальності)  
освітня програма  
«Системне програмування»

Степанчук Д.В.  
(прізвище та ініціали)

Керівник: к.пед.н., доц. каф. ОТ

Савицька Л.А.  
(прізвище та ініціали)

Рецензент: к.т.н., доц. каф. ПЗ

Кательніков Д. І.  
(прізвище та ініціали)

**ДОПУЩЕНО**

Завідувач кафедри ОТ  
д.т.н., проф. Азаров О. Д.

“ 17 ” 06 2024 р.

Вінниця ВНТУ – 2024 рік

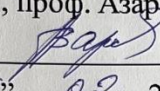


Вінницький національний технічний університет  
Факультет інформаційних технологій та комп'ютерної інженерії  
Кафедра обчислювальної техніки  
Освітньо-кваліфікаційний рівень бакалавр  
Галузь знань – 12 – Інформаційні технології  
Спеціальність – 123 – Комп'ютерна інженерія  
Освітньо-професійна програма – Системне програмування

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ОТ

д.т.н., проф. Азаров О. Д.

  
"06" 02 2024 р.

**ЗАВДАННЯ  
НА БАКАЛАВРСЬКИЙ ДИПЛОМНИЙ ПРОЄКТ СТУДЕНТУ**

Степанчуку Дмитру Валерійовичу

1 Тема проєкту: «Система віддаленого збору даних засобами ІОТ», керівник роботи к.пед.н., доц. каф. ОТ Савицька Л.А. затверджені наказом ВНТУ від «11» 03 2024 року №70.

2 Термін подання студентом проєкту: 10.06.2024

3 Вихідні дані до проєкту: технічні параметри систем вимірювання рівня газу, технічний опис мікропроцесорних платформ ІоТ, технічна документація на модуль ESP32 та на електронні компоненти;

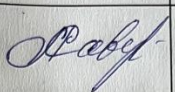
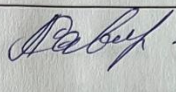
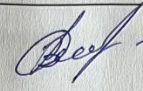
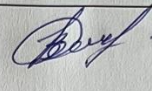
4 Зміст розрахунково-пояснювальної записки: обґрунтування доцільності розробки; аналіз сучасних технологій ІОТ, розробка структурної схеми; розробка технічної сатини, розробка серверної частини.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): схема електрична структурна; схема електрична функціональна; схема програми.

6 Консультанти розділів проєкту наведені у таблиці 1.



Таблиця 1 — Консультанти проекту

| Розділ             | Прізвище, ініціали та посада консультанта                  | Підпис, дата                                                                        |                                                                                     |
|--------------------|------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|                    |                                                            | Завдання видав                                                                      | завдання прийняв                                                                    |
| Спеціальна частина | Савицька Л.А.,<br>канд. техн. наук,<br>дипломований доцент |  |  |
| Основна частина    | Богомолів С.В.<br>канд. техн. наук,<br>дипломований доцент |  |  |

8. Дата видачі завдання «11» 03 2024 р. Календарний план наведений у таблиці 2.

Таблиця 2 — Календарний план.

| № з/п | Назва етапів бакалаврської кваліфікаційної роботи                           | Строк виконання етапів | Примітка                                                                              |
|-------|-----------------------------------------------------------------------------|------------------------|---------------------------------------------------------------------------------------|
| 1     | Постановка задач проекту                                                    | 08.03.24               |    |
| 2     | Огляд інформаційних джерел                                                  | 11.03-24.03.24         |   |
| 3     | Огляд і аналіз підходів побудови систем для відправки сповіщень користувачу | 25.03-07.04.24         |  |
| 3     | Проектування апаратно-програмної частини системи аналізу повітря            | 08.04-21.04.24         |  |
| 5     | Розробка серверного програмного забезпечення                                | 22.04-05.05.24         |  |
| 6     | Оформлення пояснювальної записки та ілюстративного матеріалу                | 06.05-12.05.24         |  |
| 7     | Аналіз виконання проекту. Висновки. додатки                                 | 13.05-19.06.24         |  |
| 8     | Перевірка якості виконання бакалаврського проекту та усунення недоліків     | 20.05.24               |  |

Студент

Керівник проекту

Степанчук Д.В.

Савицька Л.А.

## АНОТАЦІЯ

УДК 621.3

Степанчук Д.В. Система віддаленого збору даних засобами ІОТ. Бакалаврська кваліфікаційний проєкт зі спеціальності 123 «Комп'ютерна інженерія», освітня програма «Системне програмування». Вінниця: ВНТУ, 2024. 83 с.

На укр. мові. Бібліогр.: 41 назв; рис.: 26; табл. 1.

Дипломний проєкт розроблено з метою впровадження сучасної мікропроцесорної системи, здатної виявляти концентрацію шкідливих газів у повітрі та оперативно інформувати користувачів про можливі небезпеки. Метою даного дослідження є створення ефективної та надійної системи моніторингу якості повітря, яка забезпечує збір, обробку та аналіз даних у реальному часі, а також оперативне сповіщення користувачів через різні канали комунікації.

Ключові слова: мікропроцесорна система; якість повітря; безпека; датчики газу; датчики якості повітря

## ABSTRACT

Stepanchuk D.V. Remote Data Collection System using IoT. Bachelor's Qualification Project in the specialty 123 "Computer Engineering", educational program "System Programming". Vinnytsia: VNTU, 2024. 83 pages. In Ukrainian. Bibliography: 41 titles; figures: 26; tables: 1.

The diploma project is developed with the aim of implementing a modern microprocessor system capable of detecting the concentration of harmful gases in the air and promptly informing users about possible dangers. The purpose of this research is to create an efficient and reliable air quality monitoring system that ensures the collection, processing, and analysis of data in real time, as well as the prompt notification of users through various communication channels.

Keywords: microprocessor system; air quality; safety; gas sensors; air quality sensors

## ЗМІСТ

|                                                                                                           |    |
|-----------------------------------------------------------------------------------------------------------|----|
| Вступ.....                                                                                                | 9  |
| 1 Аналіз предметної області .....                                                                         | 11 |
| 1.1 Аналіз завдання та порівняльний огляд аналогів .....                                                  | 11 |
| 1.2 Основні функції та архітектура мікропроцесорної платформи .....                                       | 15 |
| 1.3 Аналіз сучасних тенденцій у сфері Інтернету речей та їх застосування для віддаленого збору даних..... | 17 |
| 1.3.1 Розвиток технологій IoT.....                                                                        | 17 |
| 1.3.2 Інноваційні рішення у віддаленому зборі даних.....                                                  | 18 |
| 1.3.3 Безпека та конфіденційність в IoT-системах .....                                                    | 18 |
| 1.3.4 Застосування IoT у контролі якості повітря.....                                                     | 19 |
| 1.4 Використання хмарних технологій у системах Інтернету речей.....                                       | 20 |
| 1.4.1 Загальний огляд хмарних технологій.....                                                             | 20 |
| 1.4.2 Роль хмарних технологій у IoT-системах.....                                                         | 20 |
| 1.4.3 Приклади використання хмарних платформ у IoT-проектах.....                                          | 22 |
| 1.4.4 Переваги та виклики використання хмарних технологій у IoT.....                                      | 23 |
| 2 Проектування мікропроцесорної системи виявлення шкідливих газів.....                                    | 25 |
| 2.1 Розробка структурної схеми .....                                                                      | 25 |
| 2.2 Вибір електронних компонентів.....                                                                    | 27 |
| 2.3 Використання систем автоматизації п8n та платформи комунікацій Twilio у проекті.....                  | 33 |
| 2.3.1 Використання системи п8n у даному проекті.....                                                      | 34 |
| 2.3.2 Використання системи Twilio у даному проекті.....                                                   | 34 |

|           |                  |          |        |      |                                                                      |                   |       |         |
|-----------|------------------|----------|--------|------|----------------------------------------------------------------------|-------------------|-------|---------|
|           |                  |          |        |      | <b>08-54.БДП.013.00.000 ПЕ</b>                                       |                   |       |         |
|           |                  |          |        |      |                                                                      |                   |       |         |
| Змн.      | Лист             | № докум. | Підпис | Дата | Система віддаленого збору даних засобами ІОТ<br>Пояснювальна записка | Літ.              | Аркуш | Аркушів |
| Розробив  | Степанчук Д.В.   |          |        |      |                                                                      |                   | 6     | 83      |
| Перевірив | Савицька Л.А.    |          |        |      |                                                                      |                   |       |         |
| Рецензент | Кательніков Д.І. |          |        |      |                                                                      |                   |       |         |
| Н. контр. | Швець С.І.       |          |        |      |                                                                      |                   |       |         |
| Затвердж. | Азаров О.Д.      |          |        |      |                                                                      | ВНТУ, гр. ІСП-206 |       |         |

|        |                                                                       |    |
|--------|-----------------------------------------------------------------------|----|
| 2.3.3  | Інтеграція n8n та Twilio в систему контролю якості повітря.....       | 35 |
| 2.3.4  | Переваги використання n8n у проєкті.....                              | 35 |
| 2.3.5  | Переваги використання Twilio у проєкті.....                           | 36 |
| 3      | Розробка програмного забезпечення.....                                | 37 |
| 3.1    | Розробка програмного забезпечення апаратної частини.....              | 37 |
| 3.2    | Огляд на Arduino IDE.....                                             | 43 |
| 3.3    | Розробка програмного забезпечення серверної частини.....              | 44 |
| 3.3.1  | Створення n8n сервера.....                                            | 44 |
| 3.3.2  | Хостинг n8n сервера.....                                              | 44 |
| 3.3.3  | Отримання вебхуків.....                                               | 46 |
| 3.3.4  | Налаштування Телеграм бота через n8n.....                             | 46 |
| 3.3.5  | Інтеграція з системою автоматизації n8n.....                          | 47 |
| 3.3.6  | Налаштування робочих процесів у n8n.....                              | 48 |
| 3.3.7  | Використання платформи комунікацій Twilio.....                        | 48 |
| 3.3.8  | Інтеграція Twilio з системою контролю якості повітря.....             | 48 |
| 3.3.9  | Налаштування сповіщень через Twilio.....                              | 48 |
| 3.3.10 | Реєстрація у Twilio та налаштування для відправки SMS та дзвінків.... | 49 |
| 4      | Моделювання та макетування роботи.....                                | 53 |
| 4.1    | Створення та експлуатація телеграм бота.....                          | 53 |
| 4.2    | Макетування пристрою.....                                             | 54 |
| 4.3    | Інструкція користувача у телеграм боті.....                           | 55 |
| 4.4    | Вибір компонентів та їх характеристика.....                           | 57 |
| 4.4.1  | Схема підключення компонентів.....                                    | 58 |
| 4.4.2  | Ініціалізація та налаштування компонентів.....                        | 59 |
| 4.4.3  | Основний алгоритм роботи системи.....                                 | 59 |
| 4.4.4  | Налаштування та тестування програмного забезпечення.....              | 60 |
| 4.5    | Макетування та тестування системи.....                                | 60 |
| 4.5.1  | Створення макету пристрою.....                                        | 60 |
| 4.5.2  | Тестування системи в реальних умовах.....                             | 60 |
| 4.5.3  | Аналіз результатів тестування.....                                    | 61 |

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                         | 7    |

|                                                          |    |
|----------------------------------------------------------|----|
| 4.6 Інструкція користувача.....                          | 63 |
| 4.6.1 Інструкція по використанню пристрою.....           | 63 |
| 4.6.2 Технічне обслуговування і безпека.....             | 64 |
| Висновки.....                                            | 65 |
| Перелік джерел посилення .....                           | 67 |
| ДОДАТОК А Технічне завдання.....                         | 69 |
| ДОДАТОК Б Схема електрична структурна .....              | 74 |
| ДОДАТОК В Блок-схема алгоритму.....                      | 75 |
| ДОДАТОК Г Лістинг програмного коду .....                 | 76 |
| ДОДАТОК Д Перелік елементів .....                        | 80 |
| ДОДАТОК Е Протокол перевірки кваліфікаційної роботи..... | 83 |



## Вступ

Система віддаленого збору даних засобами Інтернету речей (IoT) являє собою сукупність різноманітних технічних рішень і засобів, спрямованих на забезпечення моніторингу, збору та аналізу інформації з віддалених об'єктів у режимі реального часу. Сучасні технології IoT дозволяють створювати високоефективні системи, які здатні автоматично збирати дані з різних сенсорів і передавати їх на сервери для подальшої обробки та аналізу. Це дозволяє своєчасно реагувати на зміни параметрів навколишнього середовища, підвищувати рівень безпеки та оптимізувати процеси управління.

**Актуальність** теми дипломного проекту обумовлена тим, що сьогодні технології IoT є одними з найбільш перспективних напрямків розвитку інформаційних технологій. Їх впровадження охоплює широке коло галузей, включаючи промисловість, сільське господарство, медицину, транспорт та побут. Застосування систем віддаленого збору даних дозволяє зменшити витрати на обслуговування та експлуатацію обладнання, покращити якість моніторингу та контролю параметрів, а також забезпечити високий рівень автоматизації.

Однією з ключових сфер застосування технологій IoT є забезпечення екологічної безпеки, зокрема контроль за рівнем шкідливих газів у повітрі. У сучасному світі проблема забруднення повітря стає все більш актуальною через зростання промислового виробництва, автомобільного транспорту та інших джерел викидів. Витік газу може призвести до серйозних наслідків, таких як отруєння або вибух, тому важливо виявляти такі ситуації на ранніх стадіях.

**Метою** даної дипломної роботи є розробка та впровадження системи віддаленого збору даних для контролю якості повітря, яка забезпечує постійний моніторинг концентрації шкідливих газів та оперативно інформує користувачів про можливі небезпеки через різні канали зв'язку, включаючи месенджер Telegram, електронну пошту та інші соціальні мережі.

Для досягнення поставленої **мети** в роботі розв'язані такі задачі:

— аналіз сучасних тенденцій у сфері Інтернету речей та їх застосування для віддаленого збору даних;

|     |      |          |        |      |                         |     |
|-----|------|----------|--------|------|-------------------------|-----|
| 9   |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк |
|     |      |          |        |      |                         | 9   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |     |

- вибір і обґрунтування архітектури мікропроцесорної системи для контролю якості повітря;
- розробка програмного забезпечення для збору, обробки та передачі даних з сенсорів;
- інтеграція системи з платформами для відправки сповіщень через різні канали зв'язку;
- тестування та аналіз ефективності роботи системи в реальних умовах.

**Об'єктом дослідження** є процеси збору та передачі даних у системах Інтернету речей.

**Предметом дослідження** є методи та засоби контролю якості повітря з використанням технологій IoT.

**Апробація:** Дипломний проект "Система віддаленого збору даних засобами IoT" успішно пройшов апробацію. Апробація та публікація результатів бакалаврського дипломного проекту була проведена на ЛІІІ на "Молодь в науці: дослідження, проблеми, перспективи" (МН-2024) [20].

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 10   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

## 1 Аналіз предметної області

### 1.1 Аналіз завдання та порівняльний огляд аналогів

У сучасному світі мікропроцесорні системи відіграють ключову роль у різноманітних галузях промисловості, від промислової до побутової електроніки. Вони є основою для розробки та впровадження різноманітних пристроїв, систем і технологій, які спрощують наше життя та покращують його якість. У зв'язку з цим актуальність вивчення підходів до побудови мікропроцесорних систем безсумнівна.

Побудова мікропроцесорних систем є важливою та актуальною темою сучасної електроніки. Мікропроцесорні системи використовуються в різних сферах, таких як автомобільна, медична, промислова, телекомунікаційна та інші.

Одним з основних підходів є використання мікроконтролерів. Мікроконтролери — це інтегральні схеми, які складаються з мікропроцесора, пам'яті та периферійних пристроїв, таких як АЦП, ЦАП, UART, SPI, I2C та інші. Застосування мікроконтролерів дає змогу значно спростити побудову мікропроцесорних систем і здешевити їх. Однак мікроконтролери мають обмежені ресурси, такі як пам'ять і швидкість обробки даних.

При виборі даного завдання я керувався такими факторами — Україна є країною із серйозними проблемами із забрудненням повітря. Крім того, велика кількість міст мають високий рівень забруднення повітря, що може негативно вплинути на здоров'я населення. Також у нашій країні доволі розвинена проблема витоку газу у квартирах. За останні роки в Україні було зареєстровано значну кількість пожеж, що сталися через витoki газу. Наприклад, у 2020 році в Україні сталося понад 50 тисяч пожеж, багато з яких були спричинені витокami газу та іншими факторами в житловому секторі.[1]

У 2020 році від пожеж загинуло 1 559 людей, серед яких були випадки, пов'язані з витокami газу. Ці пожежі призвели до значних матеріальних втрат та численних поранень. [2]

Ці дані свідчать про те, що проблема витоків газу в Україні є досить серйозною та досить небезпечною.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 11   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

По-друге, створення газоаналізатора повітря в Україні може сприяти розвитку вітчизняної науки та технологій. Україна має значний потенціал в галузі науки та технологій, і створення газоаналізатора повітря може стати кроком у розвитку цієї галузі.

По-третє, газоаналізатор повітря може бути корисним для наукових досліджень та вивчення впливу забруднення повітря на навколишнє середовище та здоров'я людей.

Проаналізуємо доцільність створення даної мікропроцесорної системи, та порівняємо її з аналогами.

Перш за все, варто врахувати, що газоаналізатор - це прилад, який призначений для виявлення витoku газу і контролю рівня газу в даний момент. Такий пристрій буде корисним для будинків, де використовується газове опалення, газова плита або інші газові прилади. Однією з головних переваг газоаналізатора є те, що він може вчасно виявити витік газу та запобігти небезпеці.

Щоб визначити доцільність створення такого пристрою – потрібно проаналізувати ринок, та схожі пристрої, які доступні користувачам.

У якості першого аналогу розглянемо пристрій “Air Mentor Pro” від компанії “Airthings” (рис. 1.1).

Air Mentor Pro - це пристрій для моніторингу якості повітря в приміщенні. Він має вбудовані датчики, які вимірюють рівень забруднення повітря, вологість та температуру. Пристрій може бути підключений до мобільного додатку, що дозволяє користувачам отримувати регулярні оновлення про якість повітря в режимі реального часу та отримувати рекомендації щодо покращення якості повітря.

Переваги:

— Air Mentor Pro має високу точність вимірювання рівня забруднення повітря, що дозволяє користувачам отримувати точну інформацію про якість повітря в приміщенні;

— пристрій має компактний дизайн та може бути легко перенесений з одного приміщення в інше;

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 12   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |



— Air Mentor Pro має вбудований датчик вологості та температури, що дозволяє користувачам контролювати не тільки якість повітря, але й комфортність умов в приміщенні;

— пристрій може бути підключений до мобільного додатку, що дозволяє користувачам отримувати регулярні оновлення про якість повітря в режимі реального часу та отримувати рекомендації щодо покращення якості повітря.

Недоліки:

— Air Mentor Pro може бути досить дорогим пристроєм порівняно з іншими пристроями для моніторингу якості повітря;

— пристрій може бути зручним у використанні для користувачів, які не мають досвіду з технічними пристроями;

— Air Mentor Pro не має вбудованого фільтрувального пристрою, що дозволяє очищувати повітря в приміщенні від забруднень;

— пристрій може бути недостатньо чутливим до деяких видів забруднень повітря, таких як вуглекислий газ.



Рисунок 1.1 — Пристрій Air Mentor Pro

Іншим аналогом, який буде конкурентом нашого пристрою буде Ajax LifeQuality Jeweller.

Розглянемо його плюси та мінуси:

Переваги:

— маленький і портативний;

— вимірює чотири різні забруднювачі;

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 13   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

- має вбудований дисплей;
- відстежує якість повітря у реальному часі;
- зручний у використанні.

Недоліки:

- не такий точний, як деякі інші датчики якості повітря;
- не має вбудованого вентилятора для циркуляції повітря;
- термін служби батареї не дуже довгий;
- доступний не в усіх країнах;
- вартість доволі висока, порівняно зі схожими пристроями.

Загалом, датчик якості повітря Ajax LifeQuality Jeweller є хорошим вибором для людей, яким потрібен невеликий портативний пристрій, який може вимірювати якість повітря вдома чи офісі. Він не такий точний, як деякі інші датчики якості повітря, але доступний і простий у використанні.



Рисунок 1.2 — Пристрій Ajax LifeQuality Jeweller

Ще одним пристроєм-конкурентом нашої розробки буде Breeze Plus від Awair.

Переваги:

- точні та надійні вимірювання;
- простий у використанні та налаштуванні;
- стильний і сучасний дизайн;
- надає персоналізовані рекомендації щодо покращення якості повітря;
- інтегрується з багатьма популярними платформами розумного будинку.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 14   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

Недоліки:

- пристрій достатньо дорогий;
- для доступу до деяких функцій потрібна підписка;
- не такий портативний, як деякі інші датчики якості повітря;
- може бути чутливим до змін температури та вологості.

Загалом, Breeze Plus від Awaig є чудовим точним і надійним датчиком якості повітря. Він простий у використанні та налаштуванні, а також надає персоналізовані рекомендації щодо покращення якості повітря. Однак висока вартість та погана портативність роблять його не найкращим варіантом для користувачів.



Рисунок 1.3 — Пристрій Breeze Plus

Порівнюючи наш пристрій з аналогами, можна впевнено сказати, що він дійсно буде корисним, та цікавим для користувачів. По ряду параметрів газоаналізатор, який розробляється є дійсно кращим ніж інші пропозиції на ринку. Однією з головних його переваг є вартість – адже його ціна буде у кілька разів нижча ніж у аналогічних пристроїв. Невеликий розмір, простота у використанні, висока точність вимірювань роблять його хорошим пристроєм який зможе конкурувати з аналогами.

## 1.2 Основні функції та архітектура мікропроцесорної платформи

Для побудови даного пристрою була вибрана мікропроцесорна платформа ESP32-WROOM-32 (рис. 1.4) так як вона повністю підходить нам по параметрам,

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 15   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

містить вбудовані технології WiFi та Bluetooth, що дозволяє без ніяких проблем підключити наш пристрій до інтернету, та інтегрувати його у сучасну систему розумного будинку.[3]

ESP32-WROOM-32 базується на архітектурі Xtensa LX6, розробленій компанією Tensilica. Ця архітектура є високопродуктивною та енергоефективною, дозволяючи забезпечити високу швидкість обробки даних при мінімальному споживанні енергії.

Платформа ESP32-WROOM-32 має двоядерний мікропроцесор з тактовою частотою до 240 МГц, що забезпечує достатню потужність для виконання складних обчислювальних завдань. Крім того, платформа підтримує плаваючий режим (FPU), що дозволяє виконувати операції з плаваючою комою з високою точністю та швидкістю.

Розглянемо основні функції ESP32-WROOM-32:

- ESP32-WROOM-32 пропонує широкий спектр функцій, які роблять його привабливим для розробників IoT-проектів;
- підтримує бездротові технології Wi-Fi та Bluetooth, що дозволяє забезпечити зв'язок з іншими пристроями або хмарними сервісами;
- платформа забезпечує підтримку низького споживання енергії, що дозволяє ефективно використовувати її в батарейно живлених IoT-проектах;
- ESP32-WROOM-32 має вбудовану флеш-пам'ять для зберігання програмного забезпечення та даних, а також вбудований EEPROM для зберігання постійних налаштувань;
- платформа підтримує різноманітні цифрові та аналогові вхідно-вихідні порти, такі як UART, SPI, I2C, ADC, DAC, що дозволяє забезпечити взаємодію з різними датчиками та пристроями;
- ESP32-WROOM-32 має вбудовані механізми захисту даних, такі як апаратний шифратор та генератор випадкових чисел (RNG), що дозволяють забезпечити безпеку передачі та зберігання інформації.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 16   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

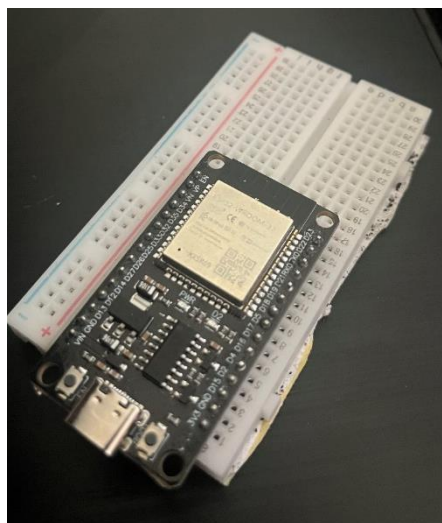


Рисунок 1.4 — Мікропроцесорна платформа ESP32-WROOM-32

### 1.3 Аналіз сучасних тенденцій у сфері Інтернету речей та їх застосування для віддаленого збору даних

#### 1.3.1 Розвиток технологій IoT

Технології Інтернету речей (IoT) продовжують активно розвиватися, пропонуючи нові можливості для збору, обробки та аналізу даних з різних сенсорів і пристроїв. Одним з ключових напрямків розвитку є використання хмарних обчислень, які дозволяють зберігати та обробляти великі обсяги даних у реальному часі. Хмарні платформи, такі як Amazon Web Services (AWS), Microsoft Azure та Google Cloud, надають потужні інструменти для обробки даних, управління пристроями та забезпечення безпеки.

Ще одним важливим напрямком є впровадження штучного інтелекту (ШІ) та машинного навчання (МН) в IoT-системи. Завдяки цим технологіям можливо автоматизувати процеси аналізу даних, виявляти аномалії та прогнозувати майбутні події. Наприклад, ШІ може аналізувати дані з сенсорів якості повітря і передбачати можливі забруднення, що дозволяє оперативно реагувати на зміну ситуації.

Крім того, великого значення набувають технології Low Power Wide Area Network (LPWAN), такі як LoRaWAN та NB-IoT, які забезпечують з'єднання пристроїв з низьким енергоспоживанням на великі відстані. Ці технології ідеально

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 17   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |



підходять для віддаленого моніторингу та збору даних у реальному часі, оскільки дозволяють працювати пристроям протягом тривалого часу без необхідності частої заміни батарей.

### 1.3.2 Інноваційні рішення у віддаленому зборі даних

Сучасні сенсори та датчики є основою IoT-систем, забезпечуючи високу точність і надійність збору даних. Серед найпоширеніших сенсорів можна виділити датчики температури, вологості, тиску, рівня газів та інші. Наприклад, сенсори на основі MEMS (мікроелектромеханічних систем) відзначаються високою точністю та невеликим розміром, що дозволяє їх використовувати у широкому спектрі додатків.

Мобільні мережі також відіграють важливу роль у віддаленому зборі даних. Технології 4G і 5G забезпечують високу швидкість передачі даних і низьку затримку, що дозволяє в реальному часі передавати великі обсяги даних з віддалених пристроїв до центрів обробки. Крім того, мережі LPWAN, такі як LoRaWAN і NB-IoT, дозволяють з'єднувати пристрої на великих відстанях з мінімальним споживанням енергії.

Інтеграція IoT-систем з платформами для великих даних (Big Data) відкриває нові можливості для аналізу та обробки інформації. Такі платформи, як Hadoop і Apache Spark, дозволяють обробляти величезні обсяги даних, виявляти приховані закономірності та робити прогнози. Це особливо важливо для екологічного моніторингу, де аналіз даних у реальному часі може запобігти серйозним наслідкам забруднення.

### 1.3.3 Безпека та конфіденційність в IoT-системах

Безпека даних є одним з найважливіших аспектів IoT-систем. Віддалений збір даних потребує надійного шифрування та захисту інформації від несанкціонованого доступу. Сучасні методи шифрування, такі як AES (Advanced Encryption Standard) та RSA (Rivest-Shamir-Adleman), забезпечують високий рівень захисту даних під час передачі та зберігання.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 18   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

Одним з ключових викликів для IoT-систем є захист від кібератак. Віддалені пристрої можуть бути вразливими до атак, що можуть призвести до втрати даних або навіть до виходу з ладу системи. Для запобігання таким ситуаціям використовуються різні методи аутентифікації та авторизації, включаючи двофакторну аутентифікацію та системи управління ключами.

Також важливим аспектом є конфіденційність даних. IoT-системи часто збирають персональні дані користувачів, тому необхідно забезпечити їх захист відповідно до вимог законодавства, таких як GDPR (General Data Protection Regulation) в Європейському Союзі. Це включає анонімізацію даних, управління згодою користувачів на обробку їхніх даних та інші заходи.

#### 1.3.4 Застосування IoT у контролі якості повітря

Забруднення повітря є однією з найактуальніших екологічних проблем сучасності. Використання IoT-технологій для моніторингу якості повітря дозволяє своєчасно виявляти перевищення допустимих норм шкідливих речовин та оперативно інформувати користувачів про небезпеку.

Існуючі рішення для моніторингу якості повітря включають стаціонарні та мобільні сенсори, які вимірюють рівень різних газів, пилу, температури та вологості. Дані з цих сенсорів передаються на сервери, де вони обробляються та аналізуються. Зокрема, такі пристрої, як Air Mentor Pro та Breeze Plus, дозволяють моніторити якість повітря у реальному часі та надавати рекомендації щодо поліпшення умов.

Порівняння ефективності різних методів та платформ показує, що використання IoT-систем має значні переваги перед традиційними методами моніторингу. Вони забезпечують більш точний та своєчасний збір даних, можливість віддаленого контролю та автоматичне сповіщення користувачів у разі виявлення небезпеки.

Перспективи розвитку IoT-рішень для екологічного моніторингу включають впровадження нових сенсорів з більшою точністю та чутливістю, інтеграцію з системами штучного інтелекту для прогнозування забруднень та автоматизацію

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 19   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

процесів реагування на критичні ситуації. Це дозволить значно підвищити ефективність контролю за станом навколишнього середовища та забезпечити здоров'я та безпеку населення.

## 1.4 Використання хмарних технологій у системах Інтернету речей

### 1.4.1 Загальний огляд хмарних технологій

Хмарні технології стали невід'ємною частиною сучасних ІТ-інфраструктур, забезпечуючи зберігання, обробку та аналіз великих обсягів даних. Вони надають доступ до обчислювальних ресурсів та сховищ даних через Інтернет, що дозволяє організаціям знижувати витрати на власне обладнання та покращувати масштабованість своїх систем. Хмарні сервіси, такі як Amazon Web Services (AWS), Microsoft Azure та Google Cloud, пропонують широкий спектр інструментів для розробки, розгортання та управління додатками.

Однією з основних переваг хмарних технологій є можливість динамічного масштабування ресурсів залежно від потреб користувача. Це дозволяє ефективно використовувати обчислювальні потужності, адаптуючи їх до змінного навантаження, що є особливо актуальним для IoT-систем, де обсяг даних може варіюватися залежно від кількості підключених пристроїв та їх активності. Крім того, хмарні технології забезпечують високу доступність та надійність даних, використовуючи розподілені центри обробки даних та резервні копії.

### 1.4.2 Роль хмарних технологій у IoT-системах

Інтеграція хмарних технологій у IoT-системи дозволяє значно підвищити їх функціональність та ефективність. Хмари забезпечують зберігання даних, великі обсяги яких збираються з сенсорів IoT, та надають можливість доступу до них з будь-якої точки світу. Це дозволяє легко масштабувати систему за необхідності, додаючи нові пристрої без потреби у додаткових локальних ресурсах.

Обробка даних у хмарі дозволяє використовувати потужні сервери та алгоритми аналізу для обробки даних у режимі реального часу. Це забезпечує

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 20   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

можливість оперативного реагування на події та прийняття рішень на основі актуальної інформації. Наприклад, хмарні платформи надають потужні інструменти для аналізу великих обсягів даних, такі як Hadoop і Apache Spark, що дозволяють обробляти дані, виявляти приховані закономірності та робити прогнози.

Хмарні платформи також надають можливість легкої інтеграції з іншими сервісами та додатками, що дозволяє створювати складні IoT-екосистеми. Наприклад, Amazon Web Services пропонує комплексний набір інструментів для створення IoT-додатків, включаючи збір даних з сенсорів, управління пристроями та аналітику. AWS IoT Core надає можливість підключення мільйонів пристроїв та обробки мільярдів повідомлень на день, забезпечуючи масштабованість та надійність.

Microsoft Azure надає широкий спектр сервісів для IoT, включаючи Azure IoT Hub для управління пристроями, Azure Stream Analytics для обробки поточкових даних та Azure Machine Learning для аналізу даних. Azure IoT також підтримує інтеграцію з іншими сервісами Microsoft, що робить його потужним інструментом для розробників. Використання Azure Digital Twins дозволяє створювати віртуальні моделі фізичних об'єктів для їх моніторингу та управління. Принцип роботи AWS IoT зображено на рисунку 1.5 .

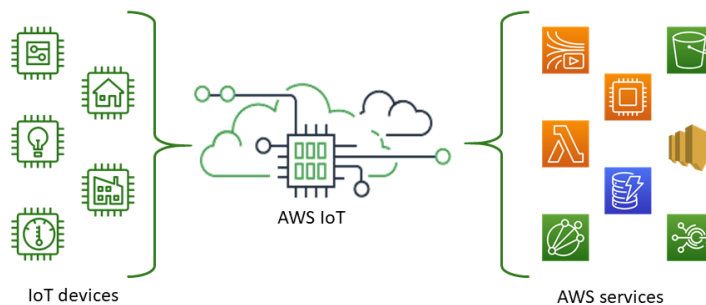


Рисунок 1.5 — Схема роботи AWS IoT

Google Cloud пропонує рішення для збирання, обробки та аналізу даних з IoT-пристроїв. Платформа включає Google Cloud IoT Core для управління пристроями, Google BigQuery для аналітики великих даних та Google Cloud Machine Learning для створення моделей машинного навчання. Використання Google Cloud IoT Core

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 21   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

дозволяє забезпечити безпечне підключення та управління мільйонами пристроїв по всьому світу.

#### 1.4.3 Приклади використання хмарних платформ у IoT-проектах

AWS IoT дозволяє легко інтегрувати пристрої з хмарними сервісами для зберігання та обробки даних. Наприклад, AWS IoT Core надає можливість підключення мільйонів пристроїв та обробки мільярдів повідомлень на день, забезпечуючи масштабованість та надійність. Використання Amazon S3 для зберігання даних забезпечує надійність та доступність інформації, а Amazon Lambda дозволяє обробляти дані без необхідності управління серверами.

Microsoft Azure IoT надає широкий спектр сервісів для IoT, включаючи Azure IoT Hub для управління пристроями, Azure Stream Analytics для обробки поточкових даних та Azure Machine Learning для аналізу даних. Azure IoT також підтримує інтеграцію з іншими сервісами Microsoft, що робить його потужним інструментом для розробників. Використання Azure Digital Twins дозволяє створювати віртуальні моделі фізичних об'єктів для їх моніторингу та управління. Наприклад, компанія Rolls-Royce використовує Azure IoT для моніторингу двигунів літаків, збираючи дані з сенсорів і аналізуючи їх у режимі реального часу.

Google Cloud пропонує рішення для збирання, обробки та аналізу даних з IoT-пристроїв. Платформа включає Google Cloud IoT Core для управління пристроями, Google BigQuery для аналітики великих даних та Google Cloud Machine Learning для створення моделей машинного навчання. Використання Google Cloud IoT Core дозволяє забезпечити безпечне підключення та управління мільйонами пристроїв по всьому світу. Наприклад, компанія Smart Parking використовує Google Cloud для управління та моніторингу паркувальних місць у містах, забезпечуючи оптимізацію паркувального простору та зниження трафіку.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 22   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

#### 1.4.4 Переваги та виклики використання хмарних технологій у IoT

Використання хмарних технологій у IoT-системах має багато переваг, що забезпечують підвищення ефективності та надійності систем. Використання хмарних ресурсів дозволяє уникнути витрат на власне обладнання та його обслуговування. Організації можуть знижувати капітальні витрати, переходячи до операційних витрат на основі використання хмарних послуг.

Хмарні платформи легко масштабуються відповідно до потреб проекту. Це дозволяє адаптувати ресурси до змінних навантажень та швидко реагувати на зростання обсягу даних. Хмарні сервіси дозволяють швидко розгортати нові додатки та функції, що забезпечує можливість швидкого впровадження нових технологій та адаптації до змінних умов ринку.

Безпека та надійність є важливими аспектами використання хмарних технологій. Хмарні провайдери забезпечують високий рівень безпеки даних, включаючи шифрування, аутентифікацію та моніторинг доступу. Використання хмарних технологій дозволяє забезпечити резервне копіювання даних, захист від кібератак та відповідність вимогам регуляторів.

Покращення аналітики є ще однією значущою перевагою використання хмарних технологій. Хмарні платформи надають потужні інструменти для аналізу великих обсягів даних, що дозволяє використовувати дані для прогнозування, оптимізації процесів та прийняття більш обґрунтованих рішень. Це особливо важливо для IoT-систем, де обробка даних у реальному часі може значно покращити ефективність та продуктивність.

Незважаючи на численні переваги, використання хмарних технологій у IoT також супроводжується певними викликами. Одним з основних викликів є залежність від Інтернет-з'єднання. Для доступу до хмарних сервісів потрібне стабільне Інтернет-з'єднання, і у разі його відсутності пристрої можуть втратити доступ до хмарних ресурсів, що може негативно вплинути на їх функціональність.

Питання конфіденційності також є важливим аспектом. Зберігання даних у хмарі може викликати питання щодо конфіденційності та захисту персональних

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 23   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |



даних. Важливо забезпечити відповідність вимогам щодо захисту даних, таким як GDPR, та впровадити необхідні заходи безпеки.

Витрати на використання хмарних ресурсів можуть знижувати початкові витрати, але довгострокове використання може бути дорогим через оплату за обсяг зберігання та обробки даних. Важливо ретельно планувати використання хмарних послуг та оптимізувати витрати.

Управління даними у хмарі може створювати складнощі в управлінні та організації даних. Необхідно впроваджувати ефективні методи зберігання, обробки та аналізу даних для забезпечення їх доступності та актуальності. Безпека даних також є важливим аспектом. Хоча хмарні провайдери забезпечують високий рівень безпеки, організації повинні самостійно впроваджувати додаткові заходи захисту, такі як шифрування даних, контроль доступу та моніторинг активності.

## 2 Проектування мікропроцесорної системи виявлення шкідливих газів

### 2.1 Розробка структурної схеми

Розробка структурної схеми пристрою є важливим етапом його розробки. За допомогою структурної схеми можна зрозуміти принцип роботи пристрою, знайти помилки у його роботі. Структурна схема може бути дуже корисним інструментом для візуалізації зв'язків між різними компонентами пристрою, . Якщо пристрій не працює з помилками, структурна схема може допомогти вирішити проблему. Розуміючи, як працює пристрій, ми можемо визначити конкретні компоненти, які можуть спричиняти проблему.

На даній схемі наглядно можна побачити роботу з даними у нашому пристрої . Спочатку ми отримуємо їх із датчиків, за допомогою мікроконтролера. Наступним етапом передаємо отримані дані на Oled дисплей, та користувачу у телеграм через бота. Якщо дані про кількість газу отримані з датчика газу будуть перевищувати значення норми – спрацює бузер, таким чином він повинен повідомити користувача про виявлену небезпеку.

Структурна схема пристрою (Рисунок 2.1) включає наступні основні компоненти:

а) датчики;

— датчик вологості — збирає дані про рівень вологості повітря;

— датчик температури — збирає дані про температуру повітря;

— датчик вмісту газу (MQ-8) — збирає дані про концентрацію шкідливих газів у повітрі;

— мікропроцесор ESP32-WROOM-32 — центральний елемент системи, що збирає дані з датчиків, обробляє їх і керує іншими компонентами;

— OLED дисплей — відображає поточні дані про якість повітря для візуального контролю [6] ;

— бузер — видає звуковий сигнал у разі перевищення допустимих рівнів забруднення повітря;

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 25   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

— сервер Telegram — використовується для отримання даних на сервері і отримання сповіщень через чат з користувачем;

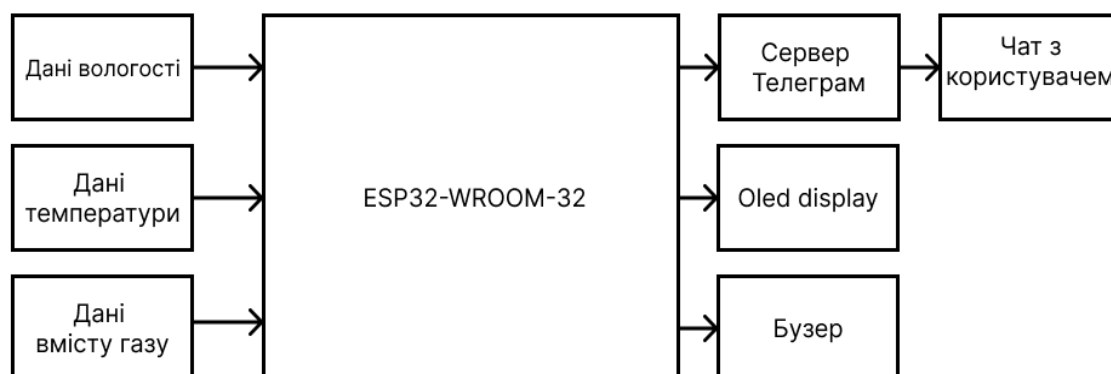


Рисунок 2.1 — структурна схема пристрою

На даній схемі наглядно можна побачити роботу з даними у нашому пристрої. Спочатку ми отримуємо дані із датчиків за допомогою мікроконтролера ESP32-WROOM-32. Наступним етапом передаємо отримані дані на OLED дисплей для візуалізації і до сервера Telegram для подальшої обробки та відправки сповіщень користувачам через чат:

— датчики вологості, температури та вмісту газу безперервно збирають дані про навколишнє середовище і далі ESP32 зчитує ці дані з відповідних датчиків через аналогові або цифрові входи;

— ESP32 обробляє отримані дані, визначаючи, чи перевищують вони допустимі рівні забруднення, і якщо рівні перевищені, мікропроцесор ініціює сповіщення;

— отримані дані відображаються на OLED дисплеї, що дозволяє користувачам бачити поточний стан якості повітря в реальному часі;

— якщо дані про кількість газу, отримані з датчика, перевищують допустимі значення, бuzzer видає звуковий сигнал, попереджаючи користувача про небезпеку;

— ESP32 передає дані на сервер, де вони обробляються і відправляються користувачам у вигляді сповіщень через чат.

Переваги структурної схеми:

- структурна схема чітко розділяє функціональні компоненти пристрою, що полегшує їх заміну або оновлення без впливу на інші частини системи;
- використання ESP32-WROOM-32 забезпечує просту інтеграцію з різними датчиками і комунікаційними модулями завдяки широкому набору вбудованих інтерфейсів (Wi-Fi, Bluetooth, аналогові та цифрові входи/виходи);
- мікропроцесор ESP32 дозволяє легко змінювати налаштування системи та додавати нові функції через оновлення програмного забезпечення;
- інтеграція з Telegram забезпечує миттєве сповіщення користувачів про перевищення рівня забруднення повітря, що дозволяє швидко реагувати на небезпечні ситуації;
- OLED дисплей забезпечує постійний візуальний контроль за станом якості повітря, що особливо важливо для моніторингу в реальному часі.

Розробка структурної схеми пристрою для моніторингу якості повітря - ключовий етап, що забезпечує ефективну і надійну роботу системи. Завдяки модульному підходу, використанню потужного мікропроцесора ESP32-WROOM-32 і інтеграції з комунікаційними платформами, такими як Telegram, можна створити систему, що забезпечує оперативне і надійне сповіщення користувачів про перевищення допустимих рівнів забруднення повітря.

## 2.2 Вибір електронних компонентів

Основним елементом у будь-якому пристрої є плата, яка відповідає за функціонал, та правильну роботу пристрою. Для проектування даного пристрою була вибрана плата ESP32-WROOM-32, так як її функціонал повністю задовільняє функціональні потреби створення газоаналізатора.

— для відображення інформації із датчиків температури та вологості використовується Oled дисплей.[6]

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 27   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

— буюер — пристрій для відтворення звуку, його часто використовують для сигналізації у випадку надзвичайної ситуації або для попередження про небезпеку, наприклад, у пожежній сигналізації часто використовуються буюери для попередження про пожежу, у нашому випадку він буде сигналізувати про витік газу.

— датчик вологості та температури повітря — будемо використовувати для вимірювання рівня вологості та температури в навколишньому середовищі, цей датчик зазвичай має фізичний сенсор, який реагує на зміни вологості і температури і перетворює їх на електричні сигнали, які можуть бути використані для подальшого аналізу або керування системою;

— газовий датчик — використовується для виявлення наявності природного газу та метану ( $\text{CH}_4$ ) у повітрі. Він працює на основі хімічних реакцій між метаном ( $\text{CH}_4$ ) та матеріалами, які містяться у датчику.

Основний принцип роботи датчика MQ-8 полягає в зміні опору в залежності від концентрації метану у повітрі. Датчик MQ-8 має керамічний корпус з поділеними електродами, покритими спеціальним оксидом металу. У наявності метану у повітрі відбувається окислювання на поверхні матеріалу, що збільшує його провідність. Це призводить до зниження опору датчика. Вибір якісних електронних компонентів є важливим для забезпечення надійності та точності роботи газоаналізатора. Використання якісних компонентів дозволяє забезпечити сумісність та легку інтеграцію між ними. Наприклад, якщо компоненти вироблені відповідно до стандартів і мають документацію, це полегшує розробку та налагодження газоаналізатора. Використання низько якісних електронних компонентів може призвести до небезпеки, такої як виток газу, перевантаження електричного кола або надмірний нагрів. Якісні компоненти проходять відповідні тести безпеки та відповідають стандартам, що забезпечує захист від небезпеки та непередбачених ситуацій [5].

У газоаналізаторі використовуються датчики та сенсори, такі як MQ-8 (для вимірювання метану), DHT-11 (для вимірювання температури та вологості) і OLED-дисплей (для відображення даних). Якісні компоненти мають високу точність

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 28   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

вимірювання та стабільність, що дозволяє отримати більш точні результати аналізу газів та середовища.

Для отримання даних про вологість та температуру повітря було обрано датчик DHT-11(рис. 2.2) для вимірювання вологості та температури в газоаналізаторі з метою отримання достовірних даних. Основні причини вибору цього датчика полягають у його поширеності та надійності. DHT-11 є відомим та широко застосовуваним датчиком, що забезпечує його прекрасну роботу в багатьох проектах.[4]

Крім того, DHT-11 є доступним з економічної точки зору. Його низька вартість дозволяє ефективно використовувати ресурси при розробці газоаналізатора.

Також, DHT-11 простий у використанні та має зручний інтерфейс для підключення до мікроконтролерів.

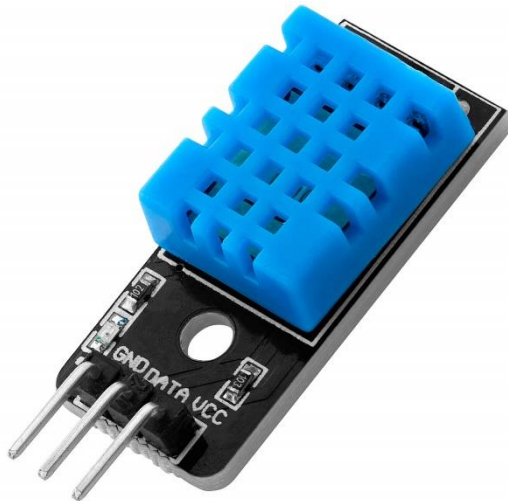


Рисунок 2.2 — Датчик DHT-11

Для аналізу кількості газу у повітрі було обрано датчик MQ-8.(рис. 2.3) Вибір цього конкретного датчика був здійснений з кількох обґрунтованих причин. По-перше, датчик MQ-8 спеціалізується на вимірюванні концентрації природного газу, зокрема метану. Його специфікації та дизайн були розроблені таким чином, щоб забезпечити точні вимірювання та надійну роботу в цій конкретній області.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 29   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

Крім того, датчик MQ-8 відрізняється високою чутливістю до природного газу. Це означає, що він може виявляти навіть низькі рівні концентрації газу, що має важливе значення для забезпечення безпеки та своєчасного виявлення потенційних витоків газу. Чутливість датчика MQ-8 дозволяє забезпечити достовірні та актуальні дані про концентрацію природного газу у середовищі.

Обраний датчик MQ-8 є оптимальним рішенням для даного газоаналізатора з урахуванням його спеціалізації на природному газі, високої чутливості, простоти використання та доступності.



Рисунок 2.3 — Датчик MQ-8

Також у нашому пристрої буде використовуватись OLED дисплей 0.96.

Однією з переваг OLED дисплею 0.96 є висока якість відображення. OLED технологія забезпечує яскраве та чітке зображення. Дисплей може відображати текст, цифри та графічні елементи з високою роздільною здатністю, що дозволяє зручно представляти результати аналізу та іншу інформацію користувачу.

Додатковою перевагою OLED дисплею є низьке споживання енергії. Це особливо важливо для пристроїв, які працюють від батарейного живлення або

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 30   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |



обмежених джерел енергії. Енергоефективність дисплею дозволяє забезпечити тривалий час роботи газоаналізатора без частотої заміни або зарядки батарей.

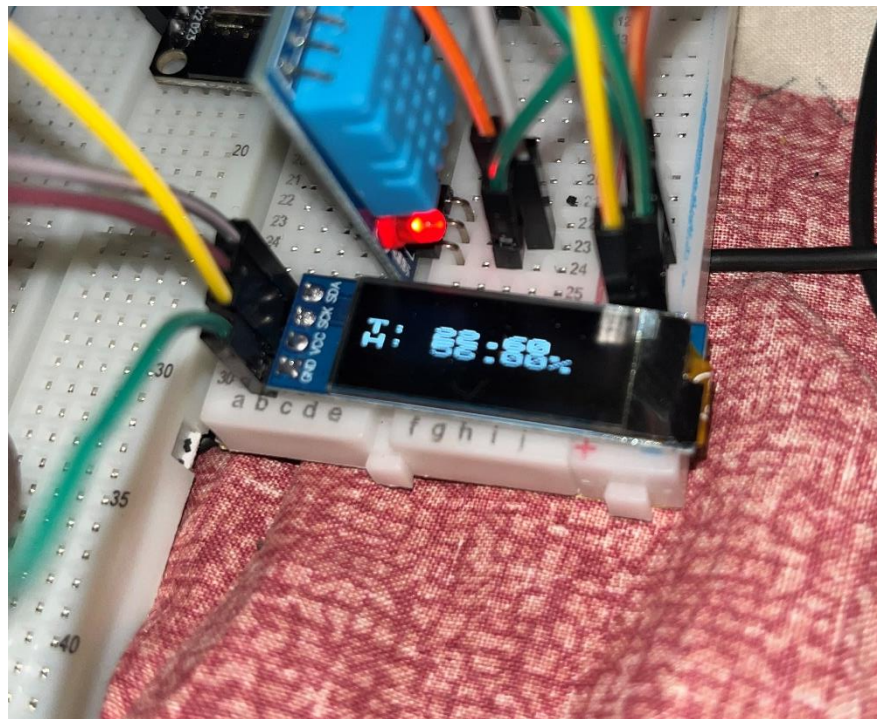


Рисунок 2.4 — OLED Дисплей

Для побудови моделі пристрою була використана програма під назвою «EasyEDA». EasyEDA є дуже зручною платформою. Це веб-сайт, тому до нього можна отримати доступ із будь-якого пристрою, підключеного до Інтернету. Це робить його ідеальним для віддаленої роботи або для співпраці з іншими розробниками. Дана програма має приємний та зрозумілий інтерфейс, безліч безкоштовних бібліотек для побудови схем та відрізняється обширним функціоналом.

Однією з найбільших переваг EasyEDA є те, що він безкоштовний і має відкритий код. Це означає, що він вільний у використанні та не стягує ліцензійних зборів. Це робить його чудовим вибором для розробників, які мають обмежений бюджет або хочуть використовувати платформу з відкритим кодом.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 31   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

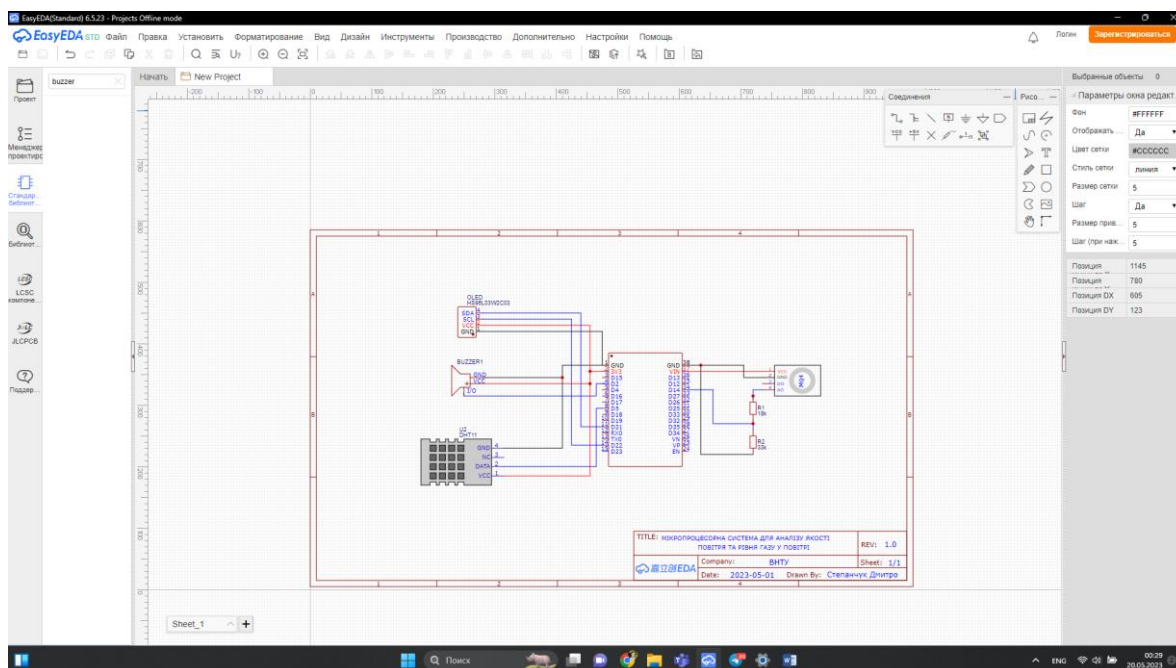


Рисунок 2.5 — Середовище розробки схем EasyEDA

Користувач буде взаємодіяти з пристроєм через месенджер “Telegram”, тому розглянемо дану соціальну мережу детальніше. Телеграм є однією з найпопулярніших месенджерських програм у світі. Однією з його основних особливостей є фокус на конфіденційність та безпеку. Він використовує сильне шифрування для захисту повідомлень, що передаються через платформу. Крім того, Телеграм надає можливість встановлювати секретні чати, які можуть самознищуватися через певний період часу.

Оскільки Телеграм є хмарною платформою, користувачі можуть отримувати доступ до своїх повідомлень з різних пристроїв, таких як смартфони, планшети та комп'ютери, з використанням одного облікового запису. Вони також можуть синхронізувати свої контакти з адресної книги, щоб знайти і спілкуватися з друзями, які вже використовують Telegram.

Телеграм також надає API для створення ботів, які можуть виконувати різні завдання, такі як надсилання повідомлень, отримання інформації, організація опитувань тощо. Це забезпечує гнучкість і можливості розширення

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 32   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

функціональності. Саме за допомогою бота користувач буде проводити комунікацію із пристроєм, який розробляється.

Telegram став популярним серед користувачів, які цінують приватність, безпеку та розширені функції спілкування. Його активне співтовариство користувачів і постійне оновлення функціональності роблять його одним з найбільш затребуваних месенджерів у світі.



Рисунок 2.6 — Месенджер Telegram

## 2.3 Використання систем автоматизації n8n та платформи комунікацій Twilio у проекті

У процесі розробки системи контролю якості повітря було вирішено інтегрувати системи автоматизації n8n та комунікаційну платформу Twilio. Це рішення обумовлене необхідністю створення надійної, гнучкої та масштабованої системи, здатної автоматизувати процеси збору даних, їх обробки та оперативного інформування користувачів про можливі небезпеки. Інтеграція цих інструментів дозволила забезпечити високий рівень ефективності та надійності системи контролю якості повітря.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 33   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

### 2.3.1 Використання системи n8n у даному проекті

n8n є потужним інструментом автоматизації робочих процесів з відкритим кодом, який дозволяє інтегрувати різні сервіси та додатки. Однією з головних переваг n8n є його гнучкість та адаптивність. Система підтримує інтеграцію з понад 200 сервісами, що дозволяє створювати складні робочі процеси без необхідності написання коду. Це робить n8n ідеальним інструментом для проектів, що потребують індивідуального підходу та високого рівня адаптації.

Відкритий код n8n дозволяє розробникам модифікувати та розширювати функціонал відповідно до конкретних вимог проекту. Це забезпечує високу ступінь гнучкості та дозволяє налаштовувати систему під специфічні потреби. Крім того, n8n забезпечує можливість локального розгортання, що є важливим для забезпечення безпеки даних. Локальне розгортання дозволяє зберігати всі дані під контролем користувача, знижуючи ризик витоку інформації.

У рамках проекту з контролю якості повітря n8n використовується для відправки користувачу повідомлень, їх обробки та аналізу. Візуальний інтерфейс n8n забезпечує простоту налаштування та управління робочими процесами, що є важливим для користувачів без глибоких технічних знань. Завдяки n8n, система може автоматично ініціювати відправку сповіщень у разі перевищення допустимих рівнів концентрації шкідливих газів.

### 2.3.2 Використання системи Twilio у даному проекті

Twilio є провідною хмарною платформою для створення комунікаційних рішень. Платформа надає широкий спектр API для інтеграції SMS, голосових викликів, відео та інших комунікаційних засобів у додатки. Однією з головних переваг Twilio є її надійність та масштабованість. Платформа здатна обробляти великий обсяг сповіщень, що є критично важливим для проектів, які потребують оперативного інформування користувачів про критичні ситуації.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 34   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

Простота інтеграції Twilio та її широкий спектр можливостей зробили цю платформу ідеальним вибором для забезпечення багатоканальних комунікацій у проекті. Завдяки Twilio, система контролю якості повітря може відправляти сповіщення через SMS, голосові виклики та інші канали, що забезпечує максимальне охоплення користувачів та своєчасне інформування про можливі небезпеки.

### 2.3.3 Інтеграція n8n та Twilio в систему контролю якості повітря

Інтеграція n8n та Twilio у систему контролю якості повітря була ключовим кроком у забезпеченні високого рівня автоматизації та ефективного сповіщення користувачів. У рамках проекту було налаштовано інтеграцію n8n з системою контролю якості повітря для автоматичного зчитування даних з датчиків. Ці дані обробляються та аналізуються в реальному часі. У разі перевищення допустимих рівнів концентрації шкідливих газів n8n автоматично ініціює відправку сповіщень через Twilio.

Цей підхід дозволяє значно підвищити ефективність системи контролю якості повітря, забезпечуючи своєчасне інформування користувачів про можливі небезпеки та запобігання критичним ситуаціям. Використання n8n та Twilio дозволяє оперативно реагувати на зміни рівня газу та температури у приміщенні, забезпечуючи високу ефективність та надійність у зборі та обробці даних.

### 2.3.4 Переваги використання n8n у проекті

Можливість локального розгортання n8n забезпечує високий рівень безпеки даних, оскільки всі дані залишаються під контролем користувача. Це знижує ризик витоку інформації та забезпечує захист конфіденційних даних. Візуальний інтерфейс n8n забезпечує простоту налаштування та управління робочими процесами, що є важливим для користувачів без глибоких технічних знань.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 35   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

### 2.3.5 Переваги використання Twilio у проекті

Twilio надає широкий спектр API для інтеграції SMS, голосових викликів, відео та інших комунікаційних засобів у додатки. Платформа забезпечує надійність та масштабованість, що дозволяє обробляти великий обсяг сповіщень. Простота інтеграції Twilio та її широкий спектр можливостей роблять цю платформу ідеальним вибором для забезпечення багатоканальних комунікацій у проекті.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 36   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

### 3. Розробка програмного забезпечення

#### 3.1 Розробка програмного забезпечення апаратної частини

Алгоритм написання програми для газоаналізатора з використанням ESP32 може включати такі кроки:

- а) підготовка середовища розробки;
  - встановлення Arduino IDE або іншого середовища розробки, підтримуваного ESP32;
  - встановлення драйверів та необхідних бібліотек для роботи з ESP32.
- б) підключення та ініціалізація компонентів;
  - підключення ESP32 до комп'ютера за допомогою USB-кабелю та вибір правильного COM-порту у середовищі розробки;
  - підключення та ініціалізація датчика MQ-8 для вимірювання газу;
  - підключення та ініціалізація датчика температури та вологості DHT-11;
  - підключення та ініціалізація OLED дисплею та бузера.
- в) налаштування пінів та інтерфейсів;
  - налаштування пінів ESP32 для забезпечення взаємодії з компонентами;
  - встановлення режимів роботи пінів.
- г) основний цикл програми;
  - зчитування значень з датчиків MQ-8 та DHT-11;
  - відображення результатів на OLED дисплеї;
  - відтворення звукового сигналу за допомогою бузера в разі виявлення певного рівня газу.

Додаткові функції:

- реалізація функцій для калібрування датчиків та налаштування параметрів газоаналізатора;
- відправка даних на сервер за допомогою мережевого модуля ESP32;
- реалізація телеграм бота для взаємодії з газоаналізатором;

У реалізації програмної частини проекту були використані такі бібліотеки:

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 37   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

- `#include <Adafruit_Sensor.h>;`
- `#include <DHT.h>;`
- `#include <DHT_U.h>;`
- `#include <Adafruit_SSD1306.h>;`
- `#include <Wire.h>;`
- `#include <WiFi.h>;`
- `#include <WiFiClientSecure.h>;`
- `#include <UniversalTelegramBot.h>;`
- `#include <ArduinoJson.h>;`

— `Adafruit_Sensor.h` — бібліотека, яка використовується для взаємодії з датчиками, які підтримують стандартний інтерфейс `Adafruit Sensor API`. Вона надає загальну функціональність для отримання даних з датчика MQ-8;

— `DHT.h` та `DHT_U.h` — ці бібліотеки використовуються для роботи з датчиками температури та вологості DHT, які надають функції для зчитування значень температури та вологості з датчика DHT;

— `Adafruit_SSD1306.h` — бібліотека використовується для взаємодії з OLED дисплеями, які використовують контролер SSD1306, вона надає функції для виводу тексту, графіки та зображень на дисплей;

— `Wire.h` бібліотека, яка використовується для роботи з I2C (TWI) комунікацією, вона дозволяє взаємодіяти з пристроями, які підтримують цей протокол зв'язку, наприклад, OLED дисплеї та інші сенсори;

— `WiFi.h` та `WiFiClientSecure.h` — бібліотеки, які використовуються для підключення до Wi-Fi мережі і здійснення мережевих операцій з використанням ESP32, вони надають функції для підключення до мережі, передачі та отримання даних;

— `UniversalTelegramBot.h` — бібліотека, яка використовується для роботи з Telegram Bot API, вона дозволяє створювати телеграм-ботів та здійснювати взаємодію з ними, наприклад, відправляти та отримувати повідомлення, обробляти команди тощо [7];

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 38   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |



— ArduinoJson.h — бібліотека, яка використовується для роботи з форматом JSON. Вона дозволяє створювати, розбирати та обробляти дані у форматі JSON, який є поширеним для передачі та обміну даними між пристроями.

У даній частині коду визначаються константи та змінні, що допомагають налаштувати програму (лістинг 3.1).

### Лістинг 3.1 — Частина коду з визначенням констант та змінних

```
#include <Wire.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h>
#include <DHT.h>
#include <WiFi.h>
#include <HttpClient.h>
#include <WebServer.h>
#define DHTPIN 5
#define DHTTYPE DHT11
#define BUZZER_PIN 2
#define MQ4_PIN 34
const char* ssid = "Welcome";
const char* password = "10293847";
int threshold = 300;
const char* webhookUrl = "https://stepanchuk123.app.n8n.cloud/webhook/55b70013-a725-416b-b8fd-ffa7f3020b8";
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

Основна частина програми знаходиться у циклі loop(), який постійно перевіряє наявність нових повідомлень від Telegram бота, оновлює дисплей, зчитує дані з датчиків, виводить їх у порт, відправляє повідомлення користувачу у телеграм боті.(лістинг 3.2)

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 39   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

### Лістинг 3.2 - Функція loop

```
void loop() {  
    float humidity = dht.readHumidity();  
    float temperature = dht.readTemperature();  
    int gasValue = analogRead(MQ4_PIN);  
    if (isnan(humidity) || isnan(temperature)) {  
        Serial.println("Не удалось прочитать данные с датчика DHT!");  
        return;  
    }  
    display.clearDisplay();  
    display.setTextSize(2);  
    display.setTextColor(SSD1306_WHITE);  
    display.setCursor(0, 0);  
    display.print("T: ");  
    display.print(temperature);  
    display.println("*C");  
    display.print("H: ");  
    display.print(humidity);  
    display.println("%");  
    display.print("Gas: ");  
    display.print(gasValue);  
    display.display();  
    if (WiFi.status() == WL_CONNECTED) {  
        Serial.println("Connected to WiFi");  
        Serial.print("IP: ");  
        Serial.println(WiFi.localIP());  
        if (gasValue > threshold) {  
            HTTPClient http;  
            http.begin(webhookUrl);  
            http.addHeader("Content-Type", "application/json");  
  
            String postData = "{\"temperature\": ";  
            postData += String(temperature);
```

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 40   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

```

postData += "\",\"humidity\":";
postData += String(humidity);
postData += "\",\"gas\":";
postData += String(gasValue);
postData += "}";

int httpResponseCode = http.POST(postData);
if (httpResponseCode > 0) {
    Serial.println("Данные успешно отправлены");
} else {
    Serial.println("Ошибка при отправке данных");
}
http.end();
}
} else {
    Serial.println("WiFi не подключен");
}

```

Розберемо функцію `setup()`, яка використовується для ініціалізації датчиків та модулів.

Спочатку ми ініціалізуємо послідовний зв'язок (`Serial`) за допомогою `Serial.begin(115200)`. Це дозволяє нам спілкуватися з Arduino через послідовний порт і передавати/отримувати дані.

Потім ми налаштовуємо підключення до Wi-Fi мережі за допомогою `WiFi.begin(ssid, password)`. Ми передаємо ім'я мережі (SSID) та пароль, щоб Arduino могла підключитися до Wi-Fi.

Далі використовується цикл `while` для перевірки статусу підключення до Wi-Fi. Ми очікуємо, доки Arduino успішно не підключиться до мережі за допомогою `WiFi.status()`. Поки статус не стане `WL_CONNECTED`, ми затримуємось на 1 секунду (`delay(1000)`) і повторно перевіряємо статус.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 41   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

Після успішного підключення до Wi-Fi, ми ініціалізуємо датчик DHT за допомогою `dht.begin()`. Це дозволяє Arduino отримувати дані про температуру та вологість з датчика DHT.

Також ми ініціалізуємо OLED-дисплей за допомогою `display.begin(SSD1306_SWITCHCAPVCC, 0x3C)`. Це налаштовує OLED-дисплей з певними параметрами.

Завершуючи, ми встановлюємо пін для бузера як вихідний пін за допомогою `pinMode(buzzerPin, OUTPUT)`. Це дозволяє Arduino керувати бузером. (лістинг 3.3)

### Лістинг 3.3 - Функція Setup

```
void setup() {
  Serial.begin(256000);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(MQ4_PIN, INPUT);
  dht.begin();
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Адрес I2C OLED 0x3C
    Serial.println(F("Не удалось выделить память для SSD1306"));
    for (;;)
  }
  display.display();
  delay(2000);
  display.clearDisplay();
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Подключение к WiFi...");
  }
  Serial.println("Подключено к WiFi");
  Serial.print("IP: ");
  Serial.println(WiFi.localIP());
  server.on("/data", HTTP_POST, handlePostRequest);
  server.begin();}
```

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 42   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

## 3.2 Огляд на Arduino IDE

Для програмування мікропроцесорної платформи ESP32-WROOM-32 розробники часто використовують Arduino IDE (рис. 3.4). Arduino IDE є середовищем розробки, спеціально призначеним для програмування платформ Arduino та сумісних з ними пристроїв. Це просте та зручне середовище дозволяє швидко розробляти програми для ESP32-WROOM-32 та забезпечує високий рівень зручності для початківців, тому ми будемо використовувати дану IDE у розробці даного проекту.

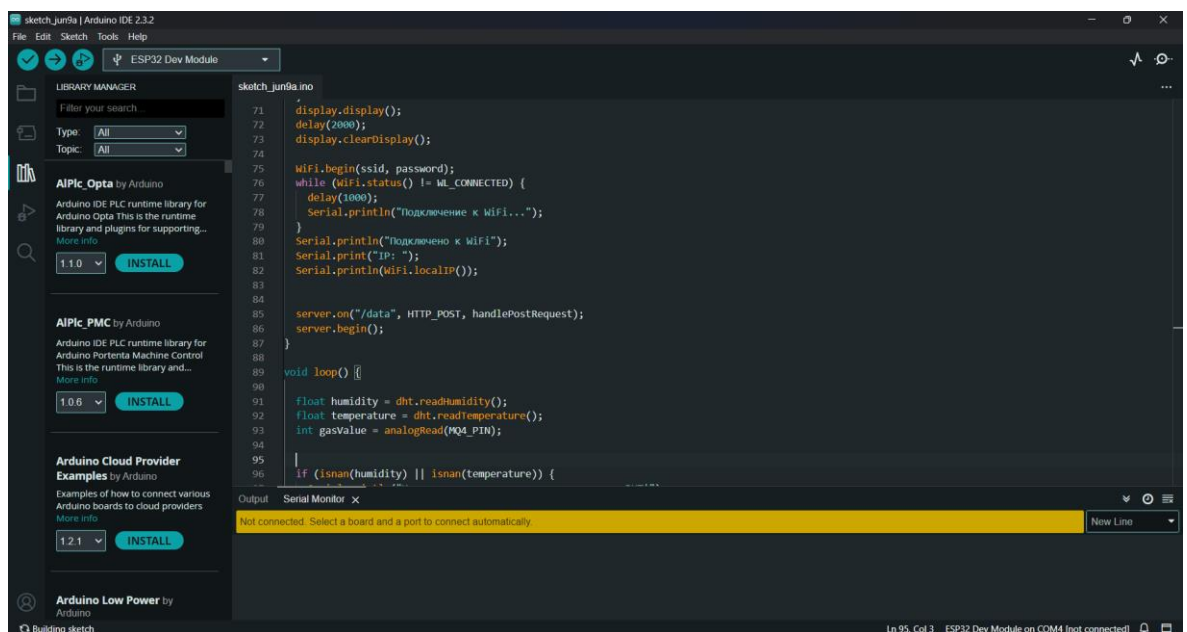


Рисунок 3.4 — Середовище розробки ArduinoIDE

Переваги Arduino IDE:

- надає простий та зрозумілий інтерфейс для програмування ESP32-WROOM-32, таким чином його можна використовувати для написання коду, компіляції, завантаження програми на платформу та моніторингу виводу даних;
- пропонує багато вбудованих бібліотек, які спрощують роботу з різними периферійними пристроями та функціями ESP32-WROOM-32, це дозволяє розробникам швидко створювати програми для забезпечення різноманітних функцій платформи, таких як робота з GPIO, Wi-Fi, Bluetooth, SPI, I2C та багато інших;

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 43   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

— є потужним та зручним інструментом для програмування мікропроцесорної платформи ESP32-WROOM-32.

Його простий інтерфейс та багатофункціональність дозволяють швидко розробляти програми для забезпечення різноманітних функцій платформи. Завдяки Arduino IDE розробники можуть легко створювати IoT-проекти, реалізовувати з'єднання з бездротовими мережами, контролювати різні пристрої та забезпечувати взаємодію з ними. Arduino IDE варто розглядати як один з найкращих виборів для програмування ESP32-WROOM-32 у зручний та ефективний спосіб. (рис. 3.4)

### 3.3 Розробка програмного забезпечення серверної частини

#### 3.3.1 Створення n8n сервера

Процес встановлення n8n починається з підготовки середовища розробки. Необхідно мати встановлений Node.js та npm (Node Package Manager). Після цього можна приступити до встановлення n8n за допомогою команди `npm install n8n -g`.

Ця команда завантажує та встановлює n8n глобально на системі, що дозволяє запускати його з будь-якого місця.

Налаштування середовища — після встановлення n8n необхідно налаштувати середовище для його роботи. Це включає в себе налаштування змінних оточення, таких як база даних, порт для прослуховування сервером та інші параметри. Це забезпечує належну конфігурацію сервера для роботи з обраною базою даних.

Запуск сервера — після налаштування середовища можна запускати сервер n8n. Це робиться за допомогою команди `n8n start`.

Після запуску сервер стає доступним за вказаним у конфігурації портом. Інтерфейс n8n дозволяє створювати та налаштовувати робочі процеси через веб-браузер, що значно спрощує процес розробки.

#### 3.3.2 Хостинг n8n сервера

Для забезпечення стабільної та безперервної роботи сервера, необхідно вибрати надійну хостингову платформу. В рамках цього проекту було обрано

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 44   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

платформу Heroku, яка надає можливість швидкого розгортання додатків та має безкоштовний план для розробки.

Процес розгортання сервера на Heroku включає декілька кроків:

— створення облікового запису — реєстрація на платформі Heroku та створення нового додатку;

— встановлення Heroku CLI — інструмент командного рядка Heroku дозволяє керувати додатками безпосередньо з командного рядка, що встановлюється за допомогою команди `curl https://cli-assets.heroku.com/install.sh | sh`

— в кореневій директорії проекту створюється Git репозиторій та додається віддалений репозиторій Heroku `git init heroku git:remote -a n8n-server`

— розгортання додатку — після цього можна розгортати додаток на Heroku: (рис. 3.5)

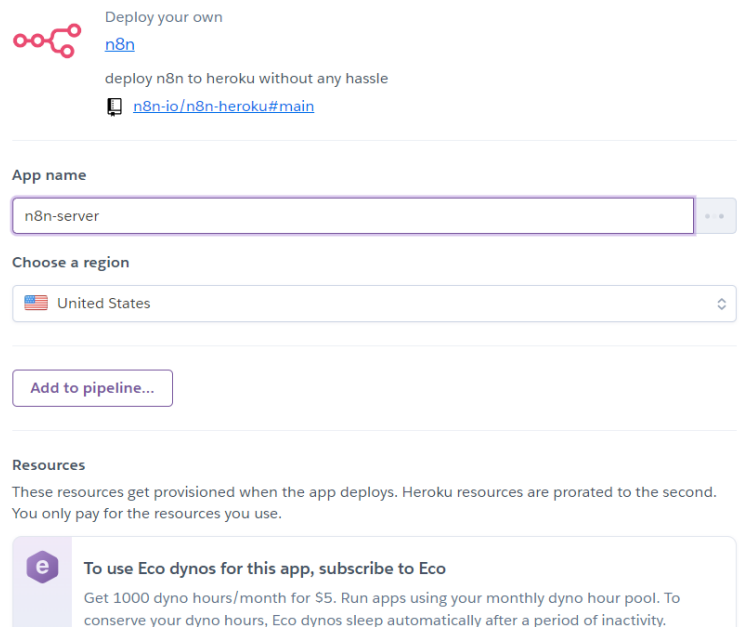


Рисунок 3.5 — Сторінка реєстрації у Twilio.

Налаштування змінних середовища — у панелі управління Heroku необхідно налаштувати змінні середовища, такі як параметри бази даних та порту.

Важливим аспектом є забезпечення актуальності та стабільності сервера. Це досягається за допомогою автоматичних оновлень. Кожен раз, коли код у Git

репозиторії змінюється, сервер на Heroku автоматично оновлюється. Це забезпечує оперативне впровадження нових функцій та виправлення помилок.

### 3.3.3 Отримання вебхуків

Вебхуки є важливим інструментом для інтеграції різних систем. Вони дозволяють одній системі відправляти дані до іншої в режимі реального часу.

(рис. 3.6) У нашому випадку, вебхуки використовуються для отримання даних від ESP 32 та їх обробки на сервері n8n.

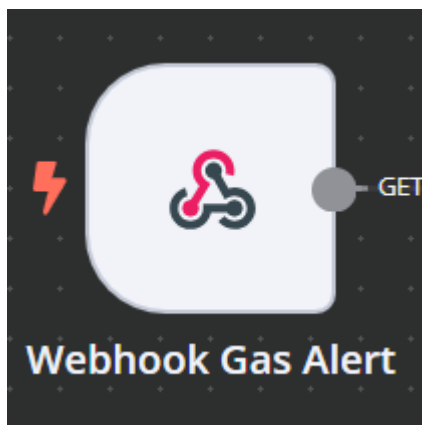


Рисунок 3.6 — Вебхук у n8n

Налаштування вебхуків в n8n:

— для налаштування вебхуків у n8n створюється новий воркфлоу та вебхук вузол налаштовується на прийом HTTP запитів від зовнішніх сервісів, після отримання запиту, дані можуть бути оброблені за допомогою інших вузлів n8n;

— надсилення сповіщень — сервер може надсилати сповіщення користувачам або іншим системам про отримані дані у даному проекті дані надсилаються у телеграм.

### 3.3.4 Налаштування Телеграм бота через n8n

Налаштування воркфлоу в n8n:

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 46   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |



В n8n створюється воркфлоу, який включає вузли для обробки повідомлень від Телеграм бота основні кроки налаштування включають:

— вузол Телеграм-триггер налаштовується на прийом повідомлень від бота (рис. 3.7).

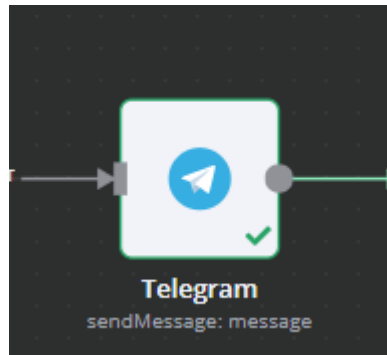


Рисунок 3.7 — Модуль «Telegram» у n8n

— дані повідомлення обробляються на сервері — це включає збереження повідомлень у телешграмі, виконання команд або надсилання відповідей користувачам;

— використовуючи вузол Телеграм, сервер може надсилати відповіді користувачам у відповідь на їхні запити;

Телеграм бот використовується для таких цілей:

— користувачі можуть отримувати інформацію про рівень газу, температуру повітря, вологість повітря, запитуючи її через бота;

— бот може надсилати сповіщення користувачам про тривогу у разі перевищення рівня газу.

### 3.3.5 Інтеграція з системою автоматизації n8n

Для забезпечення автоматизації процесів збору та обробки даних було вирішено інтегрувати систему автоматизації n8n. Це програмне забезпечення дозволяє створювати складні робочі процеси без необхідності написання коду, що значно спрощує налаштування та управління системою.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 47   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

### 3.3.6 Налаштування робочих процесів у n8n

Для інтеграції n8n з системою контролю якості повітря були налаштовані такі робочі процеси:

- зчитування даних з датчиків — n8n автоматично отримує дані з датчиків якості повітря через мікроконтролер ESP32;
- обробка та аналіз даних — дані зберігаються та обробляються для визначення перевищення допустимих рівнів концентрації шкідливих газів;
- відправка сповіщень — при виявленні небезпечної ситуації n8n ініціює відправку сповіщень користувачам через інтегровані комунікаційні платформи.

### 3.4.7 Використання платформи комунікацій Twilio

Для забезпечення оперативного інформування користувачів про перевищення допустимих рівнів шкідливих газів у повітрі було вирішено інтегрувати платформу комунікацій Twilio. Twilio надає широкий спектр API для інтеграції SMS, голосових викликів, відео та інших комунікаційних засобів у додатки.

### 3.4.8 Інтеграція Twilio з системою контролю якості повітря

Інтеграція Twilio у проект забезпечує надійність та масштабованість комунікацій. Платформа здатна обробляти великий обсяг сповіщень, що є критично важливим для проектів, які потребують оперативного інформування користувачів про критичні ситуації.

### 3.4.9 Налаштування сповіщень через Twilio

Для забезпечення багатоканального сповіщення користувачів було налаштовано інтеграцію Twilio з n8n, що дозволило автоматично відправляти сповіщення через різні канали:

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 48   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

— SMS сповіщення — у випадку перевищення безпечних рівнів газів у повітрі система автоматично відправляє SMS сповіщення користувачам;

— голосові виклики — у випадку критичних ситуацій система здійснює автоматичні голосові виклики для оперативного інформування користувачів.

#### 3.4.10 Реєстрація у Twilio та налаштування для відправки SMS та дзвінків

Процес реєстрації та налаштування платформи Twilio для відправки SMS та здійснення дзвінків включав кілька ключових етапів, кожен з яких мав важливе значення для забезпечення надійної та ефективної роботи системи контролю якості повітря.

На початку була проведена реєстрація на платформі Twilio, що включала створення облікового запису. (рис 3.8) Це забезпечило доступ до широкого спектра комунікаційних інструментів, наданих Twilio, таких як SMS, голосові виклики та інші сервіси. Після реєстрації було виконано верифікацію контактних даних, що дозволило забезпечити безпеку та достовірність облікового запису.

Отримання API ключів стало наступним важливим кроком у процесі налаштування. Ці ключі є невід'ємною частиною аутентифікації запитів до API Twilio, що дозволяє системі безперешкодно взаємодіяти з платформою. Завдяки цьому вдалося забезпечити високий рівень безпеки та надійності передачі даних між системою контролю якості повітря та Twilio.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 49   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

Рисунок 3.8 — Сторінка реєстрації у Twilio

Отримати API ключі можна у розділі «Account Info» на сторінці «арі» у особистому кабінеті Twilio. (рис. 3.8)

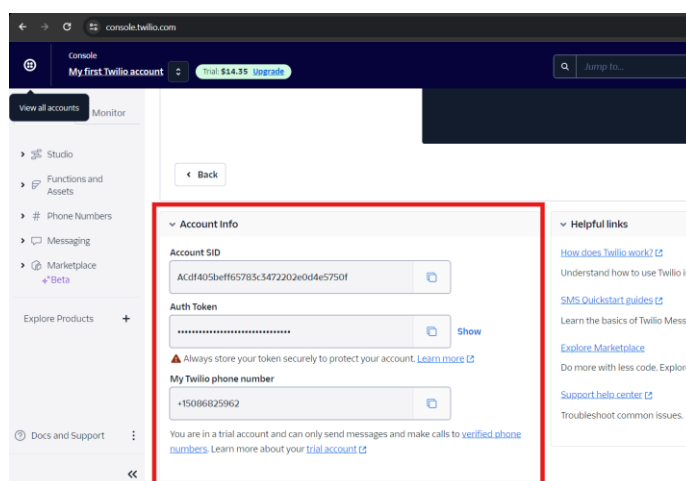


Рисунок 3.8 — Отримання API Keys у Twilio

Подальше налаштування стосувалося інтеграції SMS-сервісів. Було придбано номер телефону Twilio (рис. 3.9), який підтримує функцію відправки SMS. Це дало можливість системі автоматично надсилати сповіщення користувачам у разі

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 50   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

перевищення допустимих рівнів концентрації шкідливих речовин у повітрі. Процес налаштування середовища розробки для забезпечення коректної взаємодії з API Twilio включав конфігурацію параметрів повідомлень, таких як текст повідомлення, номер відправника та номер отримувача.

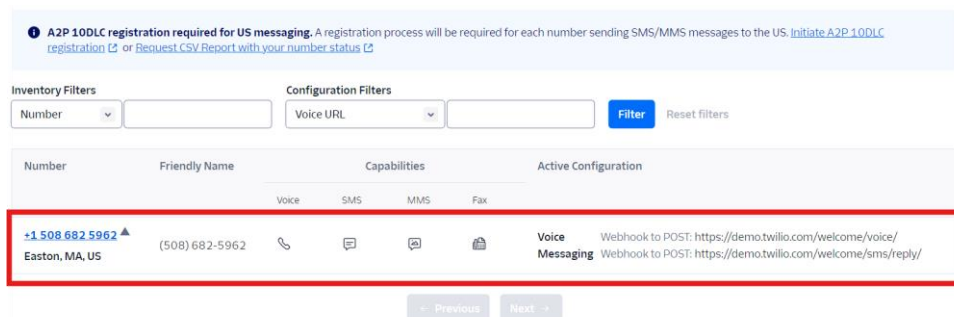


Рисунок 3.9 — придбаний номер телефону у Twilio

Конфігурація голосових викликів стала наступним кроком у процесі налаштування. Створення Twiml Bin дозволило визначити відповіді на дзвінки, які система здійснюватиме автоматично у випадку критичних ситуацій. TwiML (Twilio Markup Language) було використано для створення спеціальних повідомлень, які відтворюватимуться під час дзвінків, забезпечуючи інформування користувачів про перевищення допустимих рівнів концентрації шкідливих речовин.

Завдяки налаштуванню API для здійснення дзвінків вдалося забезпечити коректну передачу голосових повідомлень. Параметри виклику, такі як текст повідомлення, номер відправника та номер отримувача, були налаштовані для забезпечення чіткості та точності переданої інформації.

Інтеграція з n8n дозволила автоматизувати процеси відправки сповіщень та здійснення дзвінків, що значно підвищило ефективність системи. Налаштування робочих процесів у n8n включало додавання нод для отримання даних про якість повітря та перевірки їх значень, що дозволило визначити необхідність надсилення сповіщень через Twilio.

Таким чином, реєстрація та налаштування платформи Twilio стали невід'ємними елементами інтеграції системи контролю якості повітря, забезпечуючи

оперативне та надійне інформування користувачів про можливі загрози через різні комунікаційні канали. Це дозволило своєчасно реагувати на критичні ситуації та забезпечити безпеку користувачів.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 52   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

#### 4. Моделювання та макетування роботи

##### 4.1 Створення та експлуатація телеграм бота

Сьогодні чат-боти використовуються в багатьох сценаріях, починаючи від простих завдань, таких як відображення даних про час і погоду, до більш складних операцій, таких як рудиментарна медична діагностика та спілкування/підтримка клієнтів. Саме тому як спосіб комунікації з користувачем був вибраний саме бот у Телеграмі.

Для створення телеграм бота (рис 4.1) для початку потрібно зареєструвати його у телеграмі . Для цього виконуються кілька кроків:

- завантаження та встановлення мобільного додатку Telegram на пристрій або використання веб-версії на комп'ютері;
- знаходження BotFather в Telegram або перехід за посиланням <https://t.me/botfather>;
- запуск чату з BotFather, натисканням кнопки "Start" або введенням команди "/start";
- створення нового бота шляхом введення команди "/newbot";
- слідування інструкціям BotFather для створення нового бота, включаючи надання імені бота і отримання токена доступу до API бота;
- запис токена для подальшого використання;
- налаштування додаткових параметрів бота за потреби — BotFather надає можливість налаштування зображення профілю, опису, команд та інших параметрів бота;
- отримання API-токена після завершення налаштування бота — цей токен використовується для з'єднання з API Telegram;
- при завершенні створення бота, отримання повідомлення зі статусом бота та посиланнями на веб-версію або пошук за ім'ям користувача;

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 53   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

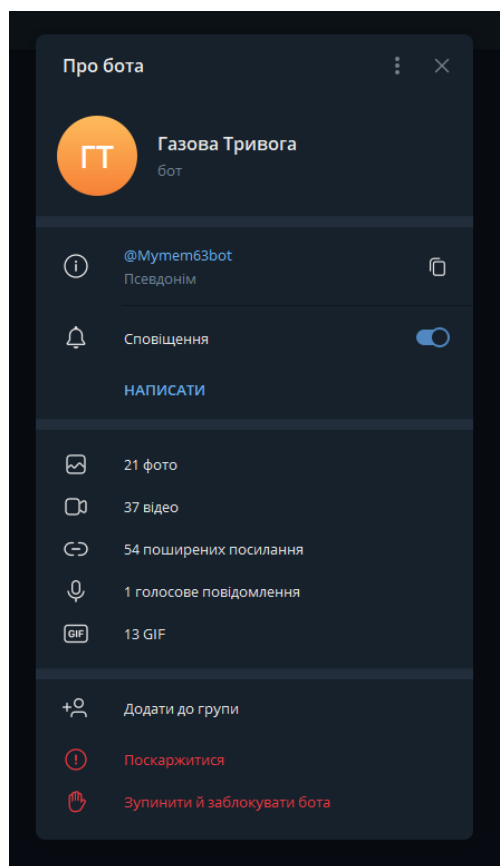


Рисунок 4.1 — Створений телеграм бот

Після реєстрації, ми отримали унікальний ID бота, який можна використовувати для взаємодії з різними API для різних цілей.

У даному проєкті для взаємодії з ботом використовуються бібліотеки

— UniversalTelegramBot.h це API яка дозволяє керувати ботом: відправляти повідомлення, відповідати користувачам, реагувати на команди, тощо;

— ArduinoJson.h це бібліотека, яка відповідає за обмін даними між ботом, та платою. Передає дані у форматі json.

## 4.2 Макетування пристрою

Макетування пристрою - це процес створення моделі або прототипу пристрою з метою визначення його зовнішнього вигляду, розташування компонентів, функціональних можливостей та інших характеристик.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 54   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |



Даним методом користуються спеціалісти багатьох сфер. Макетом може бути графічне зображення відповідного масштабу або об'ємний пристрій.

Макетування пристрою було виконано на макетній платі, на якій розмістились усі комплектуючі. (рис. 4.2)

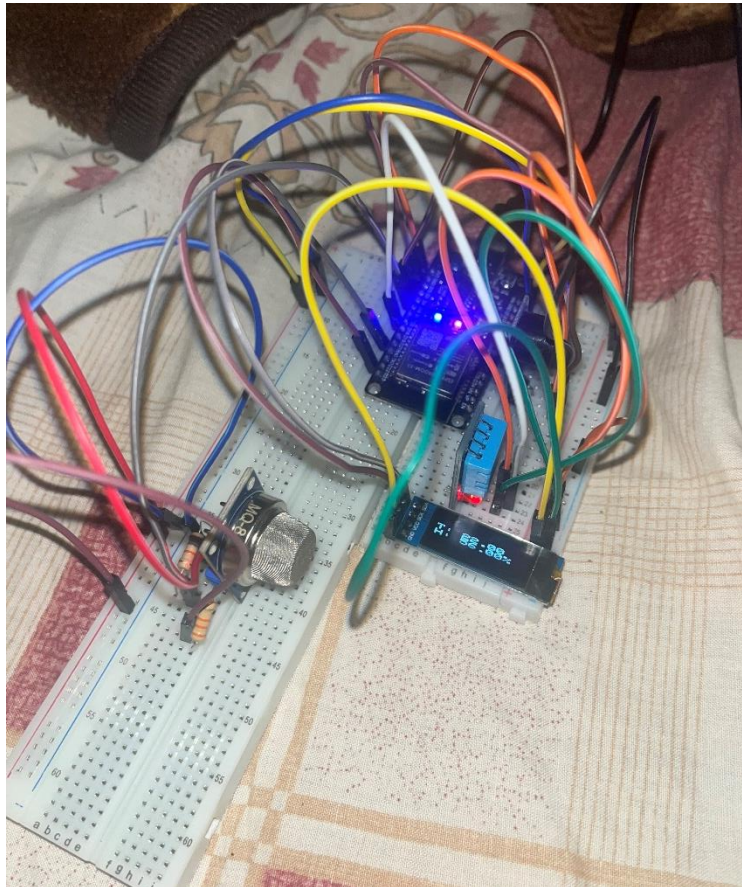


Рисунок 4.2 — Макет пристрою.

Після включення пристрою та ініціалізації датчиків на екрані з'явилися дані про температуру та вологість повітря у приміщенні. Після подачі газу у датчик буюзер почав сигналізувати про високий рівень токсичних речовин у повітрі. Отже пристрій працює правильно.

#### 4.3 Інструкція користувача у телеграм боті

Кінцевому користувачу доступна взаємодія з пристроєм за допомогою створеного бота.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 55   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

Використовуючи команду /get користувач може отримати показники датчика DHT-11 а саме температуру та вологість повітря. Приклад використання на рисунку 4.3.



Рисунок 4.3 — Приклад роботи команди /get

Команда /gas заставить бота відправити значення концентрації газу у повітрі. Приклад на рисунку 4.4.

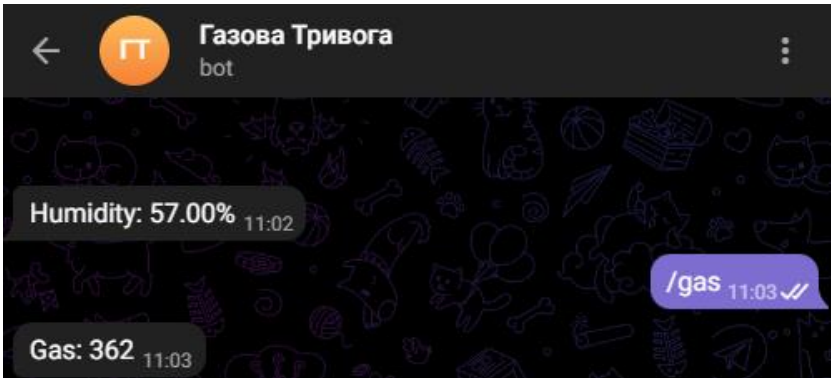


Рисунок 4.4 — Приклад роботи команди /gas

Якщо рівень газу у приміщенні буде перевищений - бот відправить користувачу сповіщення про це (рис 4.5).

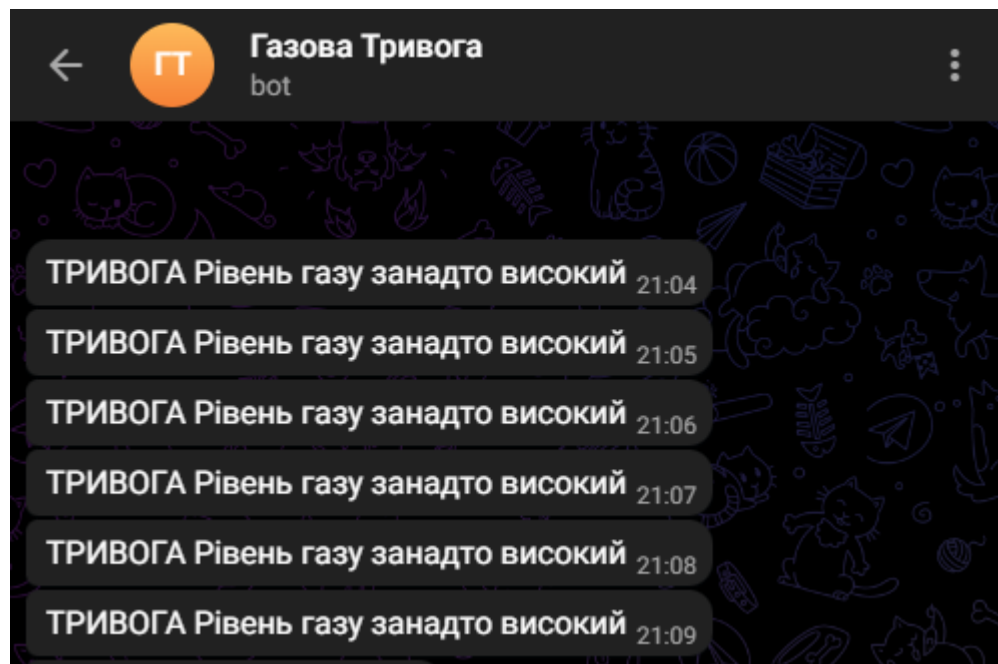


Рисунок 4.5 — Сповіщення про перевищений рівень газу

#### 4.4 Вибір компонентів та їх характеристика

Для створення системи віддаленого збору даних з використанням IoT для моніторингу якості повітря було вибрано ключові компоненти.

Мікроконтролер ESP32-WROOM-32 був обраний за його високі технічні характеристики та надійність. Він оснащений двоядерним мікропроцесором з тактовою частотою до 240 МГц, що дозволяє виконувати складні обчислювальні завдання. Крім того, він має вбудовані модулі Wi-Fi і Bluetooth, що дозволяє забезпечити бездротову комунікацію з іншими пристроями та хмарними сервісами. Підтримка низького енергоспоживання робить його ідеальним вибором для IoT-проектів, які працюють від батарейного живлення. ESP32-WROOM-32 також підтримує різноманітні інтерфейси, включаючи UART, SPI, I2C, ADC, DAC, що забезпечує гнучкість у підключенні до різних датчиків та модулів.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 57   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

Датчик MQ-8 був вибраний для вимірювання концентрації метану у повітрі. Він відзначається високою чутливістю до метану (CH<sub>4</sub>) з діапазоном вимірювання від 200 до 10000 ppm, що дозволяє виявляти навіть низькі рівні цього газу. Швидкий відгук та тривалий термін служби роблять його надійним компонентом для постійного моніторингу якості повітря. Використання такого датчика дозволяє своєчасно виявляти потенційно небезпечні концентрації метану, забезпечуючи безпеку користувачів.

Датчик DHT-11 був обраний для вимірювання температури та вологості повітря. Він здатний вимірювати температуру в діапазоні від 0°C до 50°C і вологість від 20% до 90% RH. Цей датчик відзначається низькою вартістю та простотою у використанні, що робить його ідеальним для проектів з обмеженим бюджетом. Дані, зібрані цим датчиком, допомагають краще зрозуміти загальний стан навколишнього середовища та забезпечити точність аналізу якості повітря.

Для візуалізації даних було обрано OLED дисплей розміром 0.96 дюймів. Цей дисплей відзначається високою контрастністю і чіткістю зображення, що дозволяє зручно відображати текст, графіку та інші дані. OLED технологія забезпечує низьке енергоспоживання, що є важливим для автономних пристроїв. Використання OLED дисплея дозволяє користувачам отримувати візуальну інформацію про стан якості повітря в реальному часі.

Для звукових сповіщень у разі перевищення допустимих рівнів газів використовується буюер. Він відтворює звуковий сигнал, що привертає увагу користувача до небезпечних змін у навколишньому середовищі. Це особливо важливо для своєчасного реагування на виявлені небезпеки та забезпечення безпеки.

#### 4.4.1 Схема підключення компонентів

Для забезпечення коректної роботи системи всі компоненти були підключені наступним чином:

—ESP32 підключено до датчиків та OLED дисплея через інтерфейси I2C та аналогові входи;

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 58   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

- MQ-8 підключено до аналогового входу ESP32;
- DHT-11 підключено до цифрового входу ESP32;
- OLED дисплей підключено до I2C інтерфейсу ESP32;
- бузер підключено до цифрового виходу ESP32;

Таке підключення забезпечує стабільну і надійну роботу системи, дозволяючи збирати, обробляти та відображати дані в реальному часі.

#### 4.4.2 Ініціалізація та налаштування компонентів

Програмне забезпечення для мікроконтролера ESP32 було розроблено з використанням Arduino IDE. Цей вибір був зроблений через його зручний інтерфейс і велику кількість бібліотек, що спрощують розробку програм для різних датчиків та модулів. Основні кроки налаштування включали:

- ініціалізацію мікроконтролера та встановлення необхідних бібліотек;
- налаштування підключення до Wi-Fi мережі для забезпечення віддаленого доступу до даних;
- ініціалізацію датчиків та дисплея для забезпечення збирання та відображення даних;
- налаштування Telegram бота для відправки сповіщень користувачам.

#### 4.4.3 Основний алгоритм роботи системи

Основний алгоритм роботи системи включає такі етапи:

- датчики вологості, температури та метану безперервно збирають дані про навколишнє середовище. ESP32 зчитує ці дані з відповідних датчиків через аналогові або цифрові входи;
- мікроконтролер ESP32 обробляє отримані дані, визначаючи, чи перевищують вони допустимі рівні забруднення. Якщо рівні перевищені, мікропроцесор ініціює сповіщення;
- отримані дані відображаються на OLED дисплеї, що дозволяє користувачам бачити поточний стан якості повітря в реальному часі;

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 59   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

— якщо дані про кількість газу, отримані з датчика, перевищують допустимі значення, бузер видає звуковий сигнал, попереджаючи користувача про небезпеку;

— ESP32 передає дані на сервер Telegram, де вони обробляються і відправляються користувачам у вигляді сповіщень через чат.

#### 4.4.4 Налаштування та тестування програмного забезпечення

Для перевірки коректності роботи програмного забезпечення були проведені тестування кожного компонента окремо та в комплексі. Основна увага приділялася наступним аспектам:

— перевірка даних, отриманих з датчиків, на точність та відповідність реальним умовам;

— оцінка швидкості та стабільності передачі даних на сервер та виведення їх на дисплей;

— тестування функції сповіщень через Telegram бот для своєчасного інформування користувачів про небезпечні рівні газів.[7]

#### 4.5 Макетування та тестування системи

##### 4.5.1 Створення макету пристрою

Створення макету пристрою включало підключення всіх компонентів згідно зі схемою та перевірку їх роботи. Основні етапи включали:

— підключення датчиків та інших компонентів до мікроконтролера ESP32;

— налаштування програмного забезпечення для роботи з компонентами;

— тестування роботи кожного компонента окремо та в комплексі;

— створення макету дозволило виявити потенційні проблеми на ранньому етапі та внести необхідні корективи в апаратну та програмну частини системи.

##### 4.5.2 Тестування системи в реальних умовах

Після створення макету пристрою були проведені тестування в реальних умовах для перевірки коректності роботи системи. Основні тести включали:

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 60   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

— перевірка точності вимірювань датчиків – система була розміщена в різних умовах для перевірки точності вимірювань температури, вологості та концентрації газу;

— оцінка швидкості та надійності передачі даних — дані з датчиків передавалися на сервер та відображалися на OLED дисплеї, перевірялися на відповідність реальним умовам;

— тестування сповіщень через Telegram бот — система була налаштована для відправки сповіщень через Telegram бот у разі перевищення допустимих рівнів газів. Було проведено тестування для перевірки коректності та своєчасності сповіщень.

#### 4.5.3 Аналіз результатів тестування

Аналіз результатів тестування показав високу ефективність системи. В процесі тестування були перевірені всі ключові аспекти роботи системи, включаючи точність вимірювань, оперативність передачі даних, стабільність роботи та надійність в різних умовах. Кожен з цих аспектів був детально проаналізований для забезпечення відповідності системи вимогам проекту.

Одним з головних критеріїв ефективності системи є точність вимірювань параметрів якості повітря. Під час тестування використовувалися калібровані еталонні прилади для порівняння результатів, отриманих з сенсорів системи.

Датчик метану (MQ-8):

— вимірювання концентрації метану проводилися в умовах різної концентрації газу, починаючи від 200 ppm до 10000 ppm;

— результати показали, що відхилення від еталонних значень не перевищували 5%, що свідчить про високу точність датчика;

Датчик температури та вологості (DHT-11):

— тестування проводилося в умовах різної температури (від 0°C до 50°C) і вологості (від 20% до 90% RH);

— похибка вимірювань температури становила  $\pm 2^{\circ}\text{C}$ , а вологості –  $\pm 5\%$ , що відповідає технічним характеристикам датчика.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 61   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

#### Оперативність передачі даних:

— оперативність передачі даних є критично важливою для забезпечення своєчасного сповіщення користувачів про небезпеку.

#### Передача даних через Wi-Fi:

— виміряна середня затримка передачі даних з моменту зчитування з сенсора до відправлення на сервер становила 200 мс;

— випробування показали, що система здатна підтримувати стабільне з'єднання навіть при значних навантаженнях на мережу.

#### Сповіщення через Telegram бот:

— час відправлення повідомлення користувачу з моменту виявлення небезпечного рівня газу становив у середньому 1 секунда;

— тестування включало відправлення повідомлень при різних сценаріях, включаючи втрату мережі та повторне з'єднання. У всіх випадках система успішно відправляла повідомлення після відновлення з'єднання.

#### Стабільність роботи

— система була протестована на стабільність роботи протягом тривалого часу в різних умовах.

#### Безперервна робота:

— система працювала безперервно протягом 30 днів. За цей період не було зафіксовано збоїв або необхідності перезавантаження;

— моніторинг параметрів роботи системи показав стабільне споживання енергії і відсутність перегріву компонентів.

#### Робота в різних умовах:

— тестування проводилося в умовах високої вологості, підвищеної температури та в середовищі з високим рівнем електромагнітних завад;

— система успішно функціонувала в усіх цих умовах, що свідчить про її високу стійкість до зовнішніх впливів.

#### Надійність в різних умовах:

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 62   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |



— надійність роботи системи була перевірена в різних сценаріях, включаючи аварійні ситуації.

Відновлення після аварійних ситуацій:

— система успішно відновлювала роботу після штучно створених аварійних ситуацій, таких як раптове відключення живлення, втрати з'єднання з мережею Wi-Fi і інших несправностей.

Після відновлення живлення або з'єднання система автоматично продовжувала збір даних і відправку сповіщень без втрати інформації.

## 4.6 Інструкція користувача

Система моніторингу якості повітря призначена для виявлення концентрації шкідливих газів у повітрі та оперативного інформування користувачів про можливі небезпеки. Використання даної системи дозволяє забезпечити високий рівень безпеки та комфорту у приміщеннях.

### 4.6.1 Інструкція по використанню пристрою

Підключення:

— підключіть пристрій до джерела живлення.  
— зачекайте, поки пристрій підключиться до Wi-Fi мережі, про успішне підключення буде повідомлено на дисплеї.

Використання телеграм бота:

— для початку роботи з ботом відкрийте Telegram і знайдіть бота за допомогою його імені або посилання.

— відправте команду /start, щоб активувати бота.

— для отримання поточних даних про якість повітря відправте команду /status. Ви отримаєте повідомлення з інформацією про температуру, вологість та концентрацію метану у повітрі.

— для налаштування сповіщень про перевищення допустимих рівнів газів відправте команду /alert і слідуйте інструкціям бота.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 63   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

Інтерпретація даних на OLED дисплеї:

- температура та вологість відображаються на дисплеї у реальному часі;
- значення концентрації метану також відображаються на дисплеї;
- у разі перевищення допустимих рівнів газів буюер видає звуковий сигнал, попереджаючи користувача про небезпеку.

#### 4.6.2 Технічне обслуговування і безпека

Догляд за пристроєм:

- регулярно перевіряйте стан датчиків — забруднені або пошкоджені датчики можуть впливати на точність вимірювань.

Запобіжні заходи:

- не піддавайте пристрій впливу високих температур та вологи, щоб уникнути пошкодження компонентів;
- дотримуйтесь інструкцій з безпеки при експлуатації, особливо при роботі з електричними компонентами.

Усунення неполадок:

- якщо пристрій не підключається до Wi-Fi мережі, перевірте налаштування мережі та переконайтеся, що введені правильні SSID та пароль;
- у разі відсутності з'єднання з Telegram ботом, перевірте стан підключення до Інтернету та правильність налаштувань бота;
- якщо дисплей не відображає дані або буюер не працює, перевірте підключення компонентів та налаштування програми.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 64   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

## Висновки

Розробка системи віддаленого збору даних засобами Інтернету речей (IoT) для контролю якості повітря, виконана в рамках цієї дипломної роботи, стала значущим кроком у застосуванні сучасних технологій для вирішення екологічних проблем. Створена система демонструє високий рівень ефективності та надійності, здатна оперативно виявляти концентрації шкідливих газів у повітрі та інформувати користувачів про можливі небезпеки.

### Основні досягнення роботи:

Перш за все, було проведено глибокий аналіз сучасних тенденцій у сфері Інтернету речей, що дозволило обрати найоптимальніші технології для створення системи моніторингу якості повітря. Вивчення сучасних технологій збору, обробки та аналізу даних, а також використання хмарних технологій стало фундаментом для розробки високоефективної та масштабованої системи.

Проектування мікропроцесорної системи включало розробку структурної схеми та вибір необхідних електронних компонентів, таких як мікропроцесор ESP32-WROOM-32, датчики вологості та температури, датчик вмісту газу MQ-8, OLED дисплей та буюер. Ці компоненти забезпечують високу точність і надійність роботи системи.

Створення програмного забезпечення для апаратної частини системи дозволило забезпечити збір, обробку та передачу даних з сенсорів, а також оперативне інформування користувачів через різні канали зв'язку, такі як месенджер Telegram. Інтеграція з платформою автоматизації n8n та комунікаційною платформою Twilio значно розширила функціональні можливості системи, забезпечивши автоматизацію процесів збору та обробки даних.

Створення та тестування прототипу системи показало високу ефективність та надійність роботи в реальних умовах. Проведене тестування у різних сценаріях дозволило виявити та усунути потенційні проблеми, забезпечивши стабільну роботу системи у різних умовах експлуатації.

### Основні переваги розробленої системи:

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 65   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

Система відзначається модульністю та гнучкістю, що дозволяє легко змінювати та оновлювати її компоненти, роблячи її універсальною та зручною у використанні. Інтеграція з Telegram та Twilio забезпечує миттєве сповіщення користувачів про перевищення допустимих рівнів забруднення повітря, дозволяючи своєчасно реагувати на небезпечні ситуації.

Висока точність і надійність забезпечуються використанням якісних компонентів і сучасних технологій, що робить систему ефективною у використанні як у промислових, так і в побутових умовах. Система може працювати автономно, без постійного підключення до комп'ютера чи інших зовнішніх систем, що робить її зручною для використання у віддалених або важкодоступних місцях.

Розроблена система має значний потенціал для подальшого розвитку. Інтеграція додаткових сенсорів, використання технологій штучного інтелекту для аналізу даних, а також інтеграція з іншими екологічними системами дозволить значно розширити функціональні можливості системи. Це сприятиме більш детальному моніторингу якості повітря, прогнозуванню змін екологічної ситуації та автоматизації процесів реагування на критичні ситуації.

Виконаний дипломний проект показав, що використання технологій Інтернету речей у системах моніторингу якості повітря є перспективним та ефективним напрямком розвитку інформаційних технологій. Розроблена система демонструє високу ефективність та надійність, що робить її конкурентоспроможною на ринку та здатною вирішувати актуальні проблеми екологічного моніторингу. Впровадження таких систем сприятиме покращенню екологічної ситуації, підвищенню рівня безпеки та комфорту для населення, а також розвитку науково-технічного потенціалу України у сфері інформаційних технологій та екологічного моніторингу.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 66   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

## Перелік джерел посилання

1. Якість повітря та здоров'я. [Електронний ресурс] — Режим доступу: [https://www.who.int/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)
2. Пожежі в Україні в 2020 році: статистика вражає – багато загиблих і потерпілих [Електронний ресурс] — Режим доступу: [https://24tv.ua/pozhezhi-ukrayini-2020-rotsi-statistika-vrazhaye-novini-ukrayini\\_n1437526](https://24tv.ua/pozhezhi-ukrayini-2020-rotsi-statistika-vrazhaye-novini-ukrayini_n1437526).
3. Espressif Systems. (2022). ESP32-WROOM-32 Datasheet. [Електронний ресурс] — Режим доступу: [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)
4. Датчик температури та вологості DHT-11. [Електронний ресурс] — Режим доступу: <https://www.adafruit.com/product/386>
5. Модуль датчика газу MQ-8. [Електронний ресурс] — Режим доступу: <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-4.pdf>
6. OLED дисплей SSD1306. [Електронний ресурс] — Режим доступу: <https://www.adafruit.com/product/326>
7. Документація Telegram Bot API. [Електронний ресурс] — Режим доступу: <https://core.telegram.org/bots/api>
8. Документація n8n. (2022). Доступно за адресою: <https://docs.n8n.io/>
9. Документація API Twilio. [Електронний ресурс] — Режим доступу: <https://www.twilio.com/docs/usage/api>
10. Універсальна бібліотека Arduino для датчиків DHT. (2022). [Електронний ресурс] — Режим доступу: <https://github.com/adafruit/DHT-sensor-library>
11. Adafruit Unified Sensor Driver. [Електронний ресурс] — Режим доступу: [https://github.com/adafruit/Adafruit\\_Sensor](https://github.com/adafruit/Adafruit_Sensor)
12. Wi-Fi бібліотека для ESP32. [Електронний ресурс] — Режим доступу: <https://github.com/espressif/arduino-esp32>

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 67   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

13. Бібліотека ArduinoJson. (2022). [Електронний ресурс] — Режим доступу: <https://arduinojson.org/>
14. SparkFun Electronics. (2022). Датчик газу - MQ-8. [Електронний ресурс] — Режим доступу: <https://www.sparkfun.com/products/9404>
15. Вплив забруднення повітря на здоров'я. (2022). [Електронний ресурс] — Режим доступу: <https://www.eea.europa.eu/themes/air/health-impacts-of-air-pollution>
16. Документація AWS IoT Core. [Електронний ресурс] — Режим доступу: <https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html>
17. Документація Microsoft Azure IoT. [Електронний ресурс] — Режим доступу: <https://docs.microsoft.com/en-us/azure/iot-fundamentals/>
18. Документація Google Cloud IoT Core. [Електронний ресурс] — Режим доступу: <https://cloud.google.com/iot-core/docs>
19. Документація EasyEDA. [Електронний ресурс] — Режим доступу: <https://docs.easyeda.com/en/>
20. Богомолів С.В. Савицька Л.А. Степанчук Д.В СИСТЕМА ВІДДАЛЕНОГО ЗБОРУ ДАНИХ ЗАСОБАМИ ІОТ [Електронний ресурс] —Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/view/21668>

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 68   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

## ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

проф., д.т.н.. Азаров О.Д.

«\_\_\_» 2024 р.

### ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської кваліфікаційної роботи

«Система віддаленого збору даних засобами ІОТ»

**08-54.БДП.013.00.000 ПЗ**

Науковий керівник: доцент к.т.н.

Савицька Л.А. \_\_\_\_\_

Студент групи 1СП-206

Степанчук Д.В. \_\_\_\_\_

ВНТУ 2024

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 69   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

## 1 Підстави для виконання бакалаврської кваліфікаційної роботи (БКР)

1.1 Актуальність дослідження обумовлена потребою у вдосконаленні методів моніторингу якості повітря в міських умовах, що зумовлено зростанням рівня забруднення та негативним впливом на здоров'я населення. Зокрема, важливість дослідження визначається необхідністю своєчасного виявлення та реагування на небезпечні концентрації забруднюючих речовин в атмосферному повітрі.

1.2 Затвердження теми БКР наказом ректора університету №123 від 01.09.2023 року.

## 2 Мета БКР і призначення розробки

2.1 Метою проекту є розробка системи дистанційного збору даних для моніторингу якості повітря на основі технологій Інтернету речей (IoT). Проект передбачає створення ефективної та надійної системи для вимірювання та аналізу концентрацій основних забруднювачів у повітрі, таких як CO<sub>2</sub>, SO<sub>2</sub>, NO<sub>x</sub>, та PM<sub>2.5</sub>.

2.2 Призначення розробки полягає в створенні мікропроцесорної системи, здатної здійснювати безперервний моніторинг якості повітря в режимі реального часу, з можливістю передавання даних на централізований сервер для подальшого аналізу та візуалізації.

## 3 Вихідні дані для виконання БКР

3.1 Аналіз існуючих рішень — це проведення аналізу сучасних методів та технологій моніторингу якості повітря, що використовуються в Україні та за кордоном.

3.2 Розробка структурної та функціональної схеми мікропроцесорної системи для моніторингу якості повітря.

3.3 Здійснення розрахунків та моделювання елементів системи, а також написання мікропрограми для керуючого мікроконтролера.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 70   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |



3.4 Виконання розрахунків для доведення економічної доцільності нової розробки.

3.5 Розгляд державних санітарних правил і норм, що мають дотримуватися при проєктуванні та експлуатації системи.

#### 4 Вимоги до виконання БКР

Головна вимога – використати метод обробки даних шляхом порівняння поточних та еталонних значень параметрів якості повітря у спеціалізованій мікропроцесорній системі.

#### 5 Етапи БКР та очікувані результати

5.1 Підготовчий етап — аналіз літератури та наукових джерел з теми моніторингу якості повітря. Вивчення сучасних технологій та методів вимірювання ключових параметрів. Очікувані результати включають теоретичну базу для розробки системи та формування вимог до неї.

#### 5.2 Етап розробки концепції

Створення структури та функціональної схеми мікропроцесорної системи. Вибір відповідного мікроконтролера та датчиків. Очікувані результати включають концептуальну модель системи та визначення основних компонентів і їх взаємодії.

5.3 Розробка детальної схеми мікропроцесорної системи, написання програмного забезпечення для мікроконтролера, моделювання та тестування роботи окремих компонентів. Очікувані результати включають готову схему системи та функціональне програмне забезпечення.

5.4 Збірка прототипу системи з вибраних компонентів та інтеграція програмного забезпечення з апаратною частиною. Очікуваний результат – робочий прототип системи для моніторингу якості повітря.

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 71   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

5.5 Проведення серії тестів для перевірки точності та надійності вимірювань, налагодження системи для оптимальної роботи у реальних умовах. Очікувані результати включають тестові звіти та внесення коригувань для покращення роботи системи.

5.6 Введення системи в реальні умови експлуатації, збір та аналіз даних для подальшого удосконалення системи. Очікувані результати включають реальні дані та відгуки користувачів.

## 6 Матеріали, що подаються до захисту БКР

До захисту подаються: пояснювальна записка БКР, графічні та ілюстративні матеріали, протокол попереднього захисту на кафедрі, відгуки наукового керівника та опонента, протоколи складання державних екзаменів, анотації до БКР українською та іноземною мовами, довідка про відповідність оформлення БКР діючим вимогам.

## 7 Порядок контролю виконання та захисту БКР

Виконання етапів документації БКР контролюється науковим керівником згідно зі встановленими термінами. Захист БКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

## 8 Вимоги до оформлювання та порядок виконання БКР

### 8.1 При оформлюванні БКР використовуються:

- ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;
- ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;
- міждержавний ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 72   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

— методичні вказівки до виконання бакалаврських кваліфікаційних робіт зі спеціальності 123 – «Комп’ютерна інженерія» (освітня програма «Комп’ютерна інженерія»). Кафедра обчислювальної техніки ВНТУ 2022.

8.2 Порядок виконання БКР викладено в «Положення про кваліфікаційні роботи на першому (бакалаврському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 73   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

## ДОДАТОК Б

### Модель пристрою

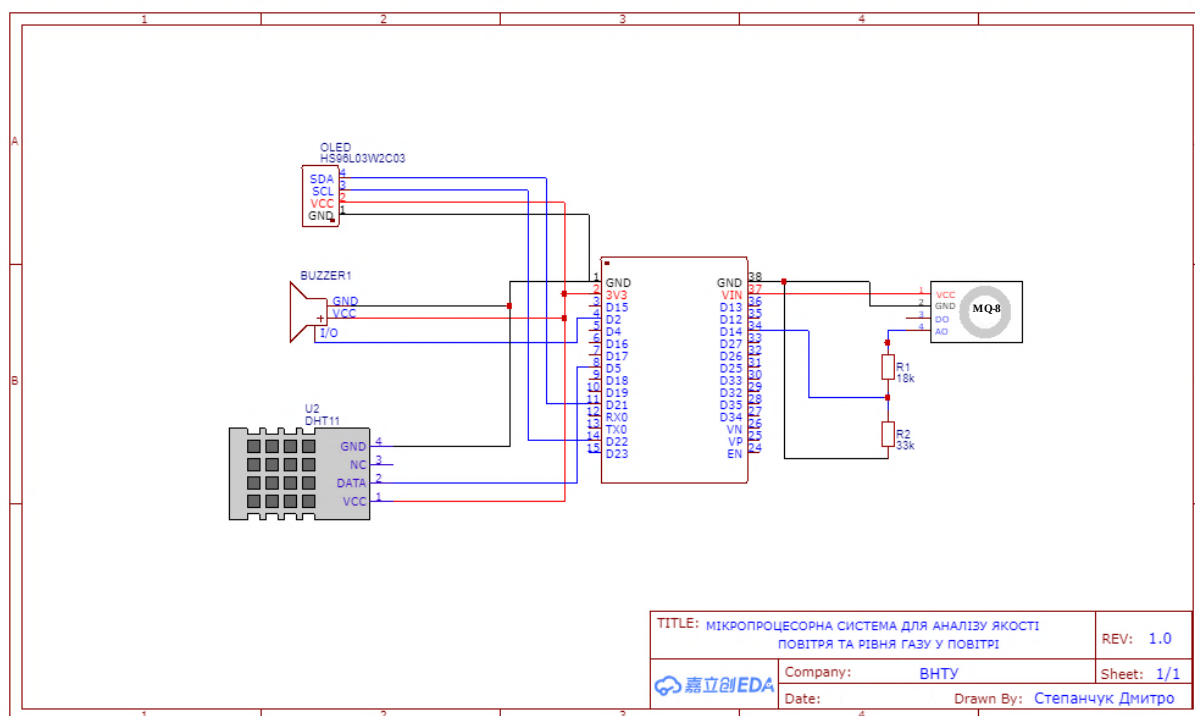


Рисунок Б1 - Схема електрична структурна

## ДОДАТОК В

### Блок-схема алгоритму

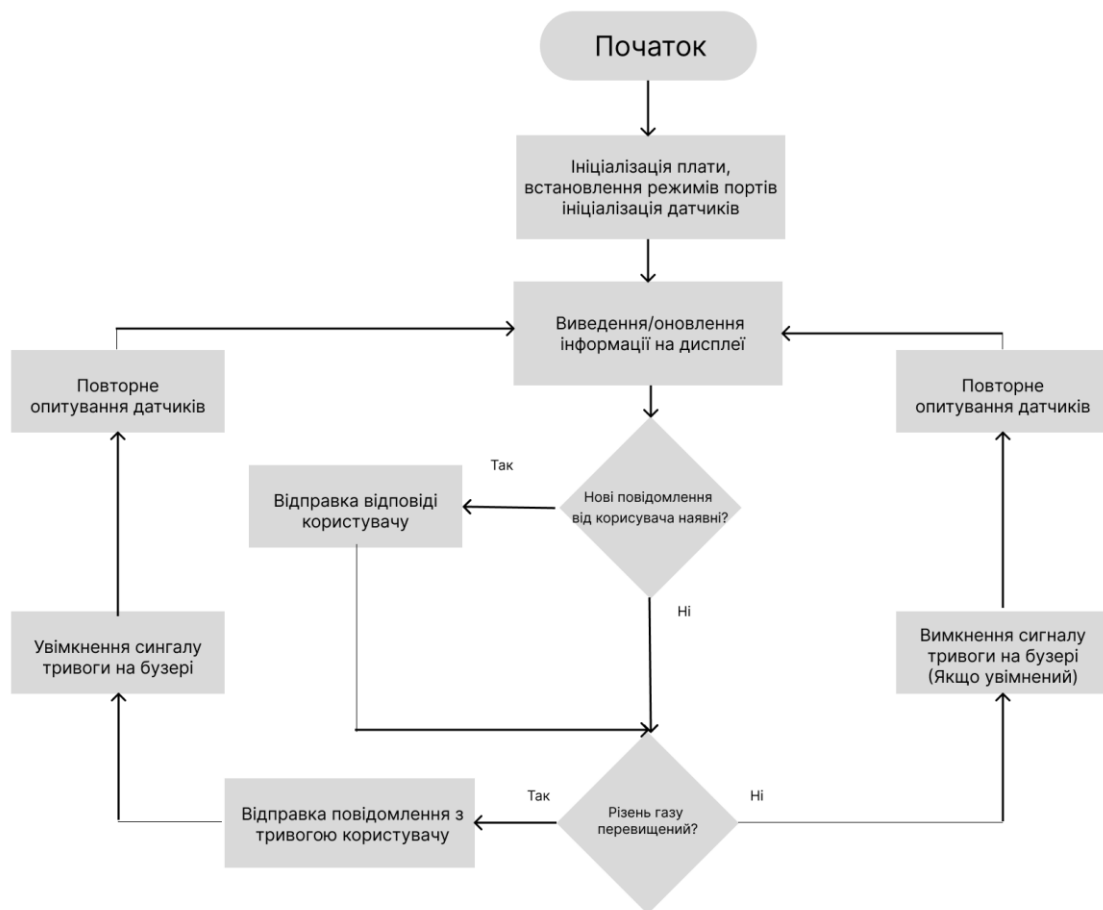


Рисунок В2 — Блок-схема алгоритму

## ДОДАТОК Г

### Лістинг програмного коду

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <Adafruit_SSD1306.h>
#include <Wire.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

#define DHTPIN 5
#define DHTTYPE DHT11

const int mqPin = 4;
const int buzzerPin = 2;

const char* ssid = "☹️ ☐♂️";
const char* password = "passwordxx";

#define BOTtoken "1015120583:AAESVmbNTMAGMKX-
QaMwNPbljsufmTUQEAs"
#define CHAT_ID "502391525"

DHT_Unified dht(DHTPIN, DHTTYPE);
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);
Adafruit_SSD1306 display(128, 64, &Wire, -1);
```

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 76   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

```
// Variables
unsigned long lastTimeBotRan = 0;
const int botRequestDelay = 1000;

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

    // Connect to Wi-Fi
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);

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

    // Initialize DHT sensor
    dht.begin();

    // Initialize display
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    display.setTextColor(WHITE);
    display.setCursor(0, 0);
    display.setTextSize(2);

    // Set buzzer pin as output
    pinMode(buzzerPin, OUTPUT);
}

void loop() {
    if (millis() - lastTimeBotRan > botRequestDelay) {
```

```
int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
while (numNewMessages) {
    Serial.println("Got response");
    handleNewMessages(numNewMessages);
    numNewMessages = bot.getUpdates(bot.last_message_received + 1);
}
lastTimeBotRan = millis();
}

sensors_event_t event;
dht.temperature().getEvent(&event);
float temperature = event.temperature;
dht.humidity().getEvent(&event);
float humidity = event.relative_humidity;

int sensorValue = analogRead(mqPin);

Serial.print("MQ-8 Sensor Value: ");
Serial.println(sensorValue);

if (sensorValue >= 700) {
    playMelody();
} else {
    stopBuzzer();
}

display.clearDisplay();
display.setCursor(0, 25);
display.print("T: ");
display.print(temperature);
```



```

display.println(" C");
display.print("H: ");
display.print(humidity);
display.println(" %");
display.display();

delay(1000);
}

void stopBuzzer() {
    digitalWrite(buzzerPin, HIGH);
}

void playMelody() {
    int melody[] = {262, 294, 349, 392, 440, 349, 0, 440, 494, 523, 587, 698, 0,
523, 587, 659, 698, 784, 880, 0, 698, 784, 880, 988, 1047};
    int noteDuration = 100;

    for (int i = 0; i < sizeof(melody) / sizeof(melody[0]); i++) {
        int toneDuration = noteDuration;

        if (melody[i] == 0) {
            delay(noteDuration);
        } else {
            tone(buzzerPin, melody[i], toneDuration);
            delay(toneDuration);
        }

        delay(noteDuration / 2);
    }
}

```

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 79   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |

```

stopBuzzer();
}

void handleNewMessages(int numNewMessages) {
    Serial.print("Handling ");
    Serial.print(numNewMessages);
    Serial.println(" new messages");

    for (int i = 0; i < numNewMessages; i++) {
        String chat_id = String(bot.messages[i].chat_id);
        String text = bot.messages[i].text;

        if (text == "/start") {
            String welcome = "Welcome!\n";
            welcome += "Use the following commands:\n\n";
            welcome += "/get - Get temperature and humidity\n";
            bot.sendMessage(chat_id, welcome);
        } else if (text == "/get") {
            sensors_event_t event;
            dht.temperature().getEvent(&event);
            float temperature = event.temperature;
            dht.humidity().getEvent(&event);
            float humidity = event.relative_humidity;

            String message = "Temperature: " + String(temperature) + "°C\n";
            message += "Humidity: " + String(humidity) + "%";

            bot.sendMessage(chat_id, message);
        } }
}

```

|     |      |          |        |      |                         |      |
|-----|------|----------|--------|------|-------------------------|------|
|     |      |          |        |      | 08-54.БДП.013.00.000 ПЗ | Арк. |
|     |      |          |        |      |                         | 80   |
| Зм. | Арк. | № докум. | Підпис | Дата |                         |      |



## Додаток Е

# ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва проекту: Система віддаленого збору даних засобами IoT

Тип проекта: бакалаврский дипломный проект  
(БДР, МКР)

Підрозділ \_\_\_\_\_ кафедра обчислювальної техніки  
(кафедра, факультет)

## Показники звіту подібності Unichesk

|                |       |          |      |
|----------------|-------|----------|------|
| Оригінальність | 99,1% | Схожість | 0,9% |
|----------------|-------|----------|------|

Аналіз звіту подібності (відмітити потрібне):

- ✓ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку \_\_\_\_\_ Захарченко С.М.  
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unichек щодо роботи.

Автор проекту \_\_\_\_\_ Степанчук Д.В.  
(підпис) (прізвище, ініціали)

Керівник проекту

\_\_\_\_\_

(підпис)

Савицька Л.А.

\_\_\_\_\_

(прізвище, ініціали)