

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І  
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ**

Завідувач кафедри

Комп'ютерних систем, мереж та кібербезпеки

Касаткін Д.Ю., к. пед.н., доц.

підпис ПБ, вчене звання і ступінь

«\_\_» \_\_\_\_\_ 2025 р.

**КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА**

На тему: «Розробка підсистеми сенсорів газів у системі моніторингу  
концентрації парникових газів»

Спеціальність 123 «Комп'ютерна інженерія»

Гарант освітньої програми: \_\_\_\_\_ / Нікітенко Є. В. /

підпис

ПБ

Керівник дипломного проекту: \_\_\_\_\_ / Смолій В.В. /

підпис

ПБ

Виконав: \_\_\_\_\_ / Білокур В. А. /

підпис

ПБ

**КИЇВ-2025**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**«ЗАТВЕРДЖУЮ»**

**завідувач кафедри**

комп'ютерних систем, мереж та кібербезпеки

/ Касаткін Д.Ю., к.п.н., доц. /

підпис

ПІБ, вчене звання і ступінь

«\_\_» \_\_\_\_\_ 20\_\_ р.

**З А В Д А Н Н Я**

**ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ БАКАЛАВРСЬКОЇ СТУДЕНТУ**

Білокур Владислав Андрійович

(прізвище, ім'я, по батькові)

Спеціальність (напрямок підготовки): 123 Комп'ютерна інженерія

Тема кваліфікаційної бакалаврської роботи: «Розробка підсистеми сенсорів газів у системі моніторингу концентрації парникових газів»

затверджена наказом ректора НУБіП України від “16” грудня 2024р. № 2251С

Термін подання завершеної роботи на кафедру 28.05.2025 року

Вихідні дані до кваліфікаційної бакалаврської роботи технічне завдання, наукові публікації, навчальна література

Перелік питань, що підлягають розробці:

1. Проаналізувати принцип роботи, застосування та класифікація існуючих сенсорів газу.

2. Описати розробку підсистеми на основі роботи сенсору МН-Z19С та її принцип роботи.

3. Створення програмного забезпечення для реалізації підсистеми сенсорів газів.

Перелік графічного матеріалу (за потреби) презентація, рисунки, таблиці.

Дата видачі завдання **“ 16 ” грудня 2024 р.**

Керівник бакалаврської роботи \_\_\_\_\_

( підпис )

Смолій В.В., к.т.н., професор

(прізвище та ініціали)

Завдання прийняв до виконання \_\_\_\_\_

( підпис )

Білокур В.А.

(прізвище та ініціали студента)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз предметної області	04.02.2025 р.	Виконано
2	Проектування системи	15.03.2025 р.	Виконано
3	Реалізація системи	10.04.2025 р.	Виконано
4	Тестування системи	01.05.2025 р.	Виконано
5	Оформлення пояснювальної записки	21.05.2025 р.	Виконано
6	Оформлення графічного матеріалу	25.05.2025 р.	Виконано

Студент \_\_\_\_\_ **Владислав БЛОКУР**  
(підпис) (ініціали та прізвище)

Керівник проекту (роботи) \_\_\_\_\_ **Віктор СМОЛІЙ**  
(підпис) (ініціали та прізвище)

## РЕФЕРАТ

Пояснювальна записка: 67 сторінок, 22 рисунків, 10 таблиць, 6 додатків, 19 джерел.

Ключові слова: СЕНСОР, ВУГЛЕКИСЛИЙ ГАЗ, МН-Z19С, МОНИТОРИНГ, POSTGRESQL

Об'єкт дослідження – сенсорна підсистема, інтегрована в систему моніторингу концентрації CO<sub>2</sub>.

Метою роботи є створення незалежного пристрою, основою якого є сенсор МН-Z19С, для фіксації, накопичення та буферизації даних про рівень CO<sub>2</sub>.

Проект складається з чотирьох розділів

Перший розділ присвячено аналізу предметної області, оцінці існуючих рішень та обґрунтуванню вибору сенсора й загальної структури системи. У другому розділі розкрито архітектуру, логіку обробки та взаємодію компонентів.

Третій розділ присвячено реалізації збору, буферизації та передачі даних, а також створенню веб-інтерфейсу.

У четвертому розділі проведено тестування системи, оцінено її стабільність, точність, буферизацію даних та відображення інформації у веб-інтерфейсі.

В результаті виконання дипломної роботи створено сенсорну підсистему для моніторингу CO<sub>2</sub>, реалізовано програмну частину, протестовано систему

Т

а

П

і

					<i>15.04 - БКР.2251 "С" 24.12.16.8.ПЗ</i>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розр.	Б.	Білокур В. А.			<i>«Розробка підсистеми сенсорів газів у системі моніторингу концентрації парникових газів»</i>	Літ.	Арк.	Акрушів
Перевір.	Є	Смолій В. В.					4	67
Н. Контр.	Р	Смолій В. В.				<i>КІ-23010бск</i>		
Зав. Каф.		Касаткін Д.Ю.						

П

## **Зміст**

### **ВСТУП8**

### **РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ10**

- 1.1 Актуальність теми10
- 1.2 Аналіз існуючих сенсорів газу11
- 1.3 Аналіз існуючих рішень16
- 1.4 Основні вимоги до сенсорної системи17
- 1.5 Вибір сенсора МН-Z19С для вимірювання СО<sub>2</sub>19
- 1.6 Загальна структура системи моніторингу22
- 1.7 Висновок до розділу 124

### **РОЗДІЛ 2. ПРОЄКТУВАННЯ СЕНСОРНОЇ ПІДСИСТЕМИ25**

- 2.1 Архітектура підсистеми25
- 2.2 Схема взаємодії компонентів28
- 2.3 Логіка обробки даних29
- 2.4 Висновок до розділу 232

### **РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ33**

- 3.1 Програмна реалізація збору даних33
- 3.2 Збереження даних у разі втрати зв'язку34
- 3.3 Передача даних до централізованої бази38
- 3.4 Тестування44
- 3.5 Результати тестування та їх аналіз47
- 3.6 Висновок до розділу 351

### **ЗАГАЛЬНІ ВИСНОВКИ53**

**ДОДАТОК А55**

**ДОДАТОК Б57**

**ДОДАТОК В60**

**ДОДАТОК Г62**

**ДОДАТОК Д63**

**ДОДАТОК Е65**

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ66**

					<b>15.04 - БКР.2251 "С" 24.12.16.8.ПЗ</b>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		6

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CO <sub>2</sub>	– Вуглекислий газ (карбон(IV) оксид)
ppm	– Parts per million — частин на мільйон
UART	– Universal Asynchronous Receiver-Transmitter — універсальний асинхронний приймач-передавач
RTC	– Real-Time Clock — годинник реального часу
JSON	– JavaScript Object Notation — формат обміну даними у вигляді тексту
REST	– Representational State Transfer — архітектурний стиль веб-служб
MQTT	– Message Queuing Telemetry Transport — легкий протокол обміну повідомленнями
HTTP	– HyperText Transfer Protocol — протокол передавання гіпертексту
SD	– Secure Digital — формат карти пам'яті
EEPROM	– Electrically Erasable Programmable Read-Only Memory — енергонезалежна пам'ять
БД	– База даних
СУБД	– Система управління базами даних
MH-Z19C	– Сенсор для вимірювання концентрації CO <sub>2</sub> на основі NDIR
NDIR	– Non-Dispersive Infrared — недисперсійна інфрачервона технологія

## ВСТУП

Зміна клімату, спричинена збільшенням концентрації парникових газів, є одним з найважливіших викликів сучасності. Серед цих газів ключову роль відіграє вуглекислий газ (CO<sub>2</sub>), адже його доля в антропогенних викидах є найбільшою. Постійне накопичення CO<sub>2</sub> в атмосфері призводить до парникового ефекту, що спричиняє підвищення середньої температури на планеті, зміни погодних умов, танення льодовиків, втрату біорізноманіття та інші негативні впливи на екосистему.

Контроль викидів CO<sub>2</sub> та постійне спостереження за його рівнем в атмосфері є важливим аспектом екологічної політики багатьох країн. Для ефективного моніторингу використовуються автоматизовані системи, які дозволяють фіксувати концентрацію вуглекислого газу в режимі реального часу. Ключовим елементом таких систем є сенсорна підсистема, що відповідає за збір первинних даних про якість повітря.

Підсистема газових сенсорів має відповідати певним критеріям: висока точність вимірювань, здатність до автономної роботи в польових умовах, надійна передача даних до центрального сховища, збереження даних у випадку втрати зв'язку, енергоефективність та можливість масштабування. У зв'язку з цим виникає потреба у розробці сучасних технічних рішень, які б відповідали цим вимогам.

Для створення такої підсистеми було обрано сенсор вуглекислого газу МН-Z19С. Цей сенсор використовує принцип недисперсійної інфрачервоної спектроскопії (NDIR) і забезпечує високу точність вимірювання рівня CO<sub>2</sub> в повітрі. Його основні переваги включають простоту підключення, підтримку цифрових інтерфейсів, низьке споживання енергії, компактність і надійність при використанні в різних умовах навколишнього середовища.

Основною ціллю даної дипломної роботи є створення сенсорної підсистеми на базі МН-Z19С для автоматизованого контролю за вмістом вуглекислого газу в навколишньому середовищі.

					<i>15.04 - БКР.2251 "С" 24.12.16.8.ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		8

Передбачається розробка функціонального рішення, здатного працювати у складі екологічних мереж, забезпечуючи збереження даних вимірювань, їх передачу до єдиної бази даних та інтеграцію з призначеним для користувача інтерфейсом.

Під час роботи буде здійснено аналіз предметної сфери та властивостей сенсора МН-Z19С, обґрунтовано підбір необхідних технічних складових, спроектовано структурну модель підсистеми, розроблено прототип та проведено його тестування. Передбачається, що результати даного дослідження знайдуть своє застосування в системах екологічного моніторингу, а також у науково-дослідних та освітніх цілях.

Загальна будова цієї дипломної праці виглядає наступним чином: вступ, аналітичний розділ з оглядом наявних рішень, проектний розділ з детальним описом архітектури підсистеми, розділ з тестуванням для перевірки її функціонування, розділ з охорони праці та техніки безпеки, висновки та перелік використаних джерел, а також додатки.

					<i>15.04 - БКР.2251 "С" 24.12.16.8.ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		9

## РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Актуальність теми

Останні десятиліття позначені для людства відчутними наслідками глобального потепління та змін клімату, що стають все більш вираженими. Першопричинами таких явищ виступає збільшення обсягу парникових газів в атмосфері, з яких вуглекислий газ (CO<sub>2</sub>) набуває особливо важливого значення. Цей газ генерується в результаті спалювання палива, виробничих процесів та роботи транспорту, і саме він визначає основну тенденцію до підвищення середньої температури на Землі.

Контроль над рівнем CO<sub>2</sub> має надзвичайне значення, як для глобальних ініціатив з охорони довкілля, так і для місцевого спостереження у міських агломераціях, виробничих комплексах, сільськогосподарських угіддях та інших місцях. Своєчасне виявлення збільшеної концентрації CO<sub>2</sub> не тільки дає можливість оцінювати вплив людської діяльності на екосистеми, а й сприяє прийняттю обґрунтованих стратегій для скорочення шкідливих викидів.

У контексті поступу до сталого розвитку та дедалі ширшого впровадження «зелених» технологій, набувають першочергового значення автоматизовані системи моніторингу якості повітря. Ці системи дають змогу оперативно отримувати чіткі дані про атмосферний стан, складати статистичні звіти, передбачати екологічні загрози та інформувати відповідні установи чи суспільство.

Важливе значення у таких системах має сенсорна підсистема, котра слугує джерелом початкової інформації. Без надійного збору даних подальший аналіз, винесення рішень або реагування на можливі небезпеки втрачають свою дієвість. У той же час, сенсори повинні відповідати низці критеріїв: бути стабільними, незалежними, пристосованими до різноманітних умов експлуатації, а також економічно обґрунтованими при розгортанні на значних площах [166].

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		10

Отже, створення підсистеми сенсорів для відстеження рівня вуглекислого газу – це справа не лише для ІТ-галузі. Вона має прямий практичний вплив на покращення екології та сприяє збереженню довкілля.

## 1.2 Аналіз існуючих сенсорів газу

Існує багато типів газових датчиків, для того, щоб вибрати найбільш підходящий газовий датчик, ми повинні розуміти різні характеристики датчика. Наприклад: Який датчик може вимірювати токсичні гази? Який датчик зручно носити з собою? Який датчик вимірює точніше? Якщо ви знаєте достатньо про датчик, ви точно можете зробити правильний вибір.

Відповідно до різних типів газу: його можна розділити на датчики *горючих газів* (часто використовують каталітичне згоряння, інфрачервоний, теплопровідність, напівпровідниковий тип), датчики *токсичних газів* (зазвичай використовують електрохімічні, метало-напівпровідникові, фотоіонізаційні, полум'яної іонізації), датчики *шкідливих газів* (часто інфрачервоні, ультрафіолетові тощо), *кисню* (часто парамагнітні, цирконієві) та інші типи.

Відповідно до різних методів використання: його можна розділити на *портативні* газові детектори та *стаціонарні* газові детектори.

Відповідно до різних методів відбору проб: їх можна розділити на *дифузійні датчики* (датчик безпосередньо встановлюється у вимірюваному середовищі, а вимірюваний газ безпосередньо контактує з блоком детектування через природну дифузію), *насосні датчики* (через всмоктувальний насос тощо газ всмоктується в блок детектування).

За різними функціями: його можна розділити на *датчик лише одного газу* та *композитний датчик* газу (може виявляти кілька газів одночасно).

Відповідно до різних принципів виявлення: його можна розділити на *напівпровідниковий датчик* газу, *електрохімічний датчик* газу, *NDIR-датчик* газу, *каталітичний датчик* газу, датчик *теплопровідності* газу, *магнітний датчик* газу і так далі [2].

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		11

Класифікація за різними принципами роботи наразі є найбільш поширеною. Розглянемо три вида датчиків газу:

1. Напівпровідникові датчики газу.

Напівпровідниковий датчик газу – це пристрій, який використовує напівпровідниковий елемент як вимірювальну одиницю. Принцип його роботи полягає в тому, що газ піддається окислювально-відновній реакції на напівпровіднику, що призводить до зміни значення опору. Коли газ проходить через вимірювальну комірку, він адсорбується на її поверхні і вступає в реакцію. Рівень газу вимірюється шляхом зміни провідності або потенціалу, що характеризується рухом носіїв (Рис 1.1).



Рисунок 1.1 – Зображення напівпровідникового датчику.

Наприклад, коли газ метан проходить через вимірювальну комірку, відбувається реакція, яка споживає кисень. Закріплені електрони вивільняються, а електрони повертаються до напівпровідника, збільшуючи провідність. Коли електрони фіксуються киснем, опір збільшується; коли метан реагує з киснем, і електрони повертаються в напівпровідник, опір зменшується [2]. Зміна опору пов'язана з наявністю, відсутністю та кількістю метану, а вміст метану можна дізнатися, вимірявши зміну опору.

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		12

- Переваги: проста конструкція, низька ціна, висока чутливість виявлення, швидка швидкість реакції тощо.
- Недоліки: Лінійний діапазон вимірювання невеликий. Йому сильно заважають інші гази в повітрі, і на нього легко впливає температура навколишнього середовища.

Це найпоширеніший газовий датчик, який широко використовується для виявлення витоків горючих газів у будинках і на заводах, і підходить для метану, скрапленого газу, водню тощо.

## 2. Електрохімічні датчики газу.

Електрохімічні датчики реагують з вимірюваним газом і генерують електричний сигнал, пропорційний концентрації газу. Більшість електрохімічних датчиків газу є датчиками струму, що виробляють струм, лінійно пропорційний концентрації газу (Рис 1.2).



Рисунок 1.2 – Зображення електрохімічного датчику.

Принцип роботи електрохімічних газових датчиків: газ проходить через діафрагму, яка запобігає конденсації, і ця діафрагма також має

пилонепроникний ефект. Потім молекули газу дифундують через фільтр і досягають робочого електрода. Тут газ окислюється або відновлюється, і ця електрохімічна реакція викликає зміну струму, що протікає через зовнішній ланцюг.

Серед датчиків, заснованих на електрохімічному принципі роботи, найпростішим є двоелектродна система. Її робочий електрод і протилежний електрод розділені електролітом і з'єднані із зовнішнім ланцюгом з невеликим опором. Коли газ дифундує в датчик, він вступає в реакцію і генерує електричний струм. Чим більше газу, тим більший струм [3].

- Переваги: швидка реакція, хороший лінійний вихід та інші переваги.
- Недоліки: Потрібно детектувати в середовищі з киснем (1-10% об'ємної частки кисню), а під час використання потрібно витратити рідкі електроліти. Легко піддається впливу температури, вологості, тиску і газів зі схожими хімічними властивостями. Якщо на місці присутній змішаний газ, це може призвести до помилок.

Ці газові датчики широко використовуються для виявлення газів у промислових і цивільних сферах. Він може виявляти озон, формальдегід, чадний газ, аміак, сірководень, діоксид сірки, діоксид азоту, кисень та інші гази. Її часто використовують у портативних приладах та приладах онлайн-моніторингу газу.

### 3. NDIR-датчики газу

NDIR (недисперсійні інфрачервоні) газові датчики випромінюють інфрачервоне світло, що змушує молекули газу вібрувати. Принцип роботи полягає в тому, що різні гази можуть поглинати різні довжини інфрачервоних хвиль [3]. Чим більше газу, тим менше інфрачервоного світла може пройти крізь нього. Коли інфрачервоні промені проходять через газ в зоні вимірювання, вони резонують з молекулами газу і поглинаються молекулами газу, коли молекули вібрують (Рис 1.3).

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		14

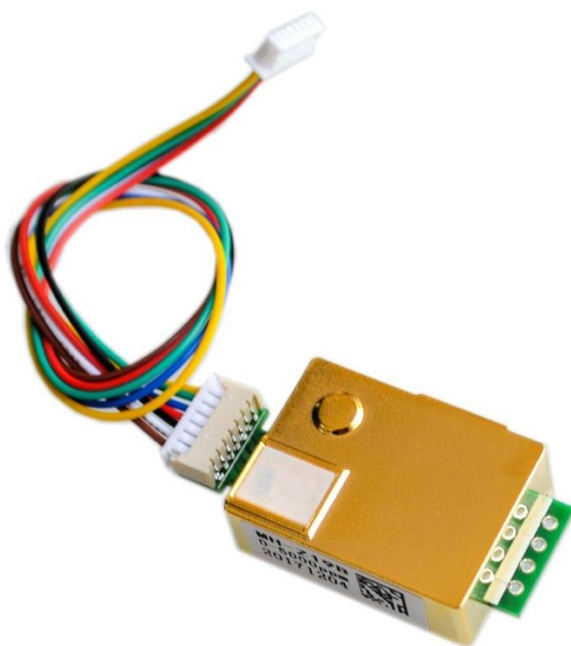


Рисунок 1.3 – Зображення NDIR-датчику.

Це послаблює інфрачервоні промені, що передаються. Молекули газу коливаються на різних частотах. Деякі атоми дуже малі і легкі, тому вони коливаються на високих частотах. Наприклад, частота коливань молекули метану становить 333кГц, а центральна довжина хвилі - 3,3мкм. Оскільки молекулярна маса відрізняється, довжина хвилі поглинання інфрачервоного випромінювання також відрізняється (Рис 1.4).

- Переваги: інші поширені на ринку датчики, не можуть вимірювати будь-який газ, як ось  $\text{CO}_2$ , але NDIR-датчики на це спроможний. Найбільш часто вимірювані гази для NDIR - це  $\text{CH}_4$  і  $\text{CO}_2$ . Також NDIR є оптичним датчиком і для його роботи не потрібен кисень. Стабільність датчика NDIR в основному залежить від джерела світла. Якщо вибрано хороше джерело світла, довгострокова стабільність буде відмінною, і її можна досягти без калібрування протягом 2 років. Також датчик має низьку вартість обслуговування.

- Недоліки: NDIR-датчики споживають велику потужність. Вимірювання концентрації газу на рівні ppm є дорогим. Структура, програмне та апаратне забезпечення є більш складними, а ціна відносно високою.

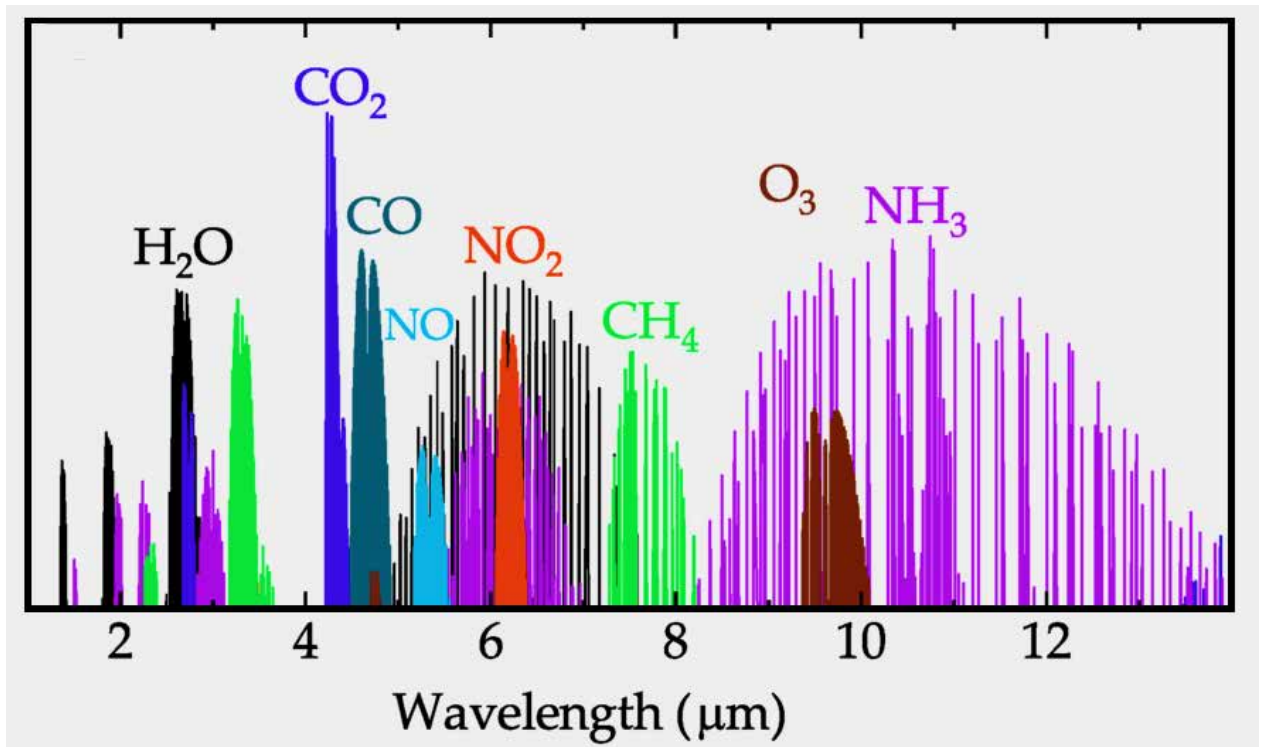


Рисунок 1.4 – Спектри поглинання деяких газів у інфрачервоній частині спектра.

### 1.3 Аналіз існуючих рішень

Системи спостереження за вуглекислим газом нині широко використовуються у світі — від масштабних державних проектів до місцевих екологічних рухів. Більшість з них побудовані за подібним принципом: сенсорні пристрої фіксують показники CO<sub>2</sub>, надсилаючи їх на централізовані сервери, де інформація зберігається, опрацьовується та представляється графічно для подальшого дослідження.

Найбільш поширеними є стаціонарні екологічні станції. Їх розгортають у містах, індустріальних зонах або неподалік природних об’єктів. Зазвичай, такі системи обладнані високоточними вимірювальними приладами. Проте вони потребують значних фінансових витрат на обслуговування, забезпечення електроенергією та зв’язком. Наприклад, у країнах Європейського Союзу функціонують національні мережі моніторингу атмосферного повітря (наприклад Air Quality e-Reporting). Вони використовують складні, багатокомпонентні станції, зокрема, для контролю за вмістом CO<sub>2</sub>.

Окремо варто виділити портативні чи модульні системи, що розгортаються на основі недорогих сенсорів, таких як, МН-Z19С. Їхня перевага в доступності, незалежності та здатності формувати розподілену мережу з багатьох точок. Ці рішення дедалі активніше застосовуються в сільському господарстві, для освітніх та наукових задач, а також в "розумних" міських системах. Прикладами подібних підходів слугують відкриті проекти, як-от AirGradient, котрі використовують прості мікроконтролери та датчики для отримання даних про якість повітря з можливістю інтеграції з хмарними платформами [4].

Недоліком багатьох наявних розв'язків рішення є труднощі з масштабуванням, обмежена автономність (особливо в умовах реального середовища) та брак функцій буферизації даних у разі втрати з'єднання. До того ж, не всі реалізації володіють ретельно продуманою системою контролю цілісності даних та резервного копіювання.

Аналітичні дані свідчать про стрімкий розвиток ринку екологічного контролю за рівнем CO<sub>2</sub>. Водночас, відчувається нагальна потреба в адаптованих рішеннях, що роблять акцент на зручності, автономності, стабільності роботи та можливості застосування у віддалених районах. Підсистема сенсорів, над якою ведеться робота в межах цього дипломного проекту, буде відповідати саме цим ключовим вимогам [3].

#### **1.4 Основні вимоги до сенсорної системи**

Підсистема сенсорів у системі моніторингу парникових газів відіграє ключову роль, гарантуючи отримання надійних та своєчасних даних щодо вмісту вуглекислого газу в атмосфері. З огляду на це, до подібної системи висуваються численні технічні, функціональні та експлуатаційні вимоги, дотримання яких є вкрай важливим для роботи всієї екологічної інфраструктури [5].

Насамперед, система має гарантувати точне вимірювання концентрації CO<sub>2</sub> з огляду на зміни температури, вологості та тиску, котрі здатні впливати

					<i>15.04 - БКР.2251 "С" 24.12.16.8.ПЗ</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		17

на точність даних. Це потребує застосування відповідних алгоритмів для компенсації похибок і надійну роботу сенсора в широкому спектрі умов.

Другим важливим критерієм є періодичність збору даних. Вимоги передбачають щонайменше зчитування показників кожну 1 годину. Це дозволяє виявляти короточасні зміни концентрації CO<sub>2</sub> та формувати більш точну динаміку процесів.

Оскільки систему розроблено для експлуатації в польових умовах, ключовою вимогою є автономність. Сенсорний модуль мусить функціонувати незалежно від стаціонарного джерела живлення. Для цього передбачається використання Li-Po акумуляторів, що працюватимуть спільно з сонячними панелями для підзарядки.

Необхідно зважати й на опір до зовнішніх впливів. Датчик має функціонувати у температурному режимі від -40°C до +125°C, а також бути захищеним від пилу, вологи та інших атмосферних умов (відповідно до класів IP). Всі складові частини повинні відповідати промисловому стандарту, а побутовому.

Ще одним важливим аспектом є збереження та передавання інформації. У випадку розриву з'єднання з основною базою даних, контролер повинен накопичувати зібрані дані у внутрішній пам'яті [6].

Після відновлення з'єднання дані автоматично передаються до централізованого сховища. До того ж, необхідно забезпечити перевірку цілісності інформації та уникнення її дублювання чи втрати (Рис 1.5).

Окремо слід виділити вимоги до масштабованості. Система мусить передбачати до давання нових сенсорних вузлів без кардинальних змін у структурі. Крім того, було б добре мати можливість початкового налаштування через Bluetooth, що значно спрощує процес встановлення та подальше обслуговування.

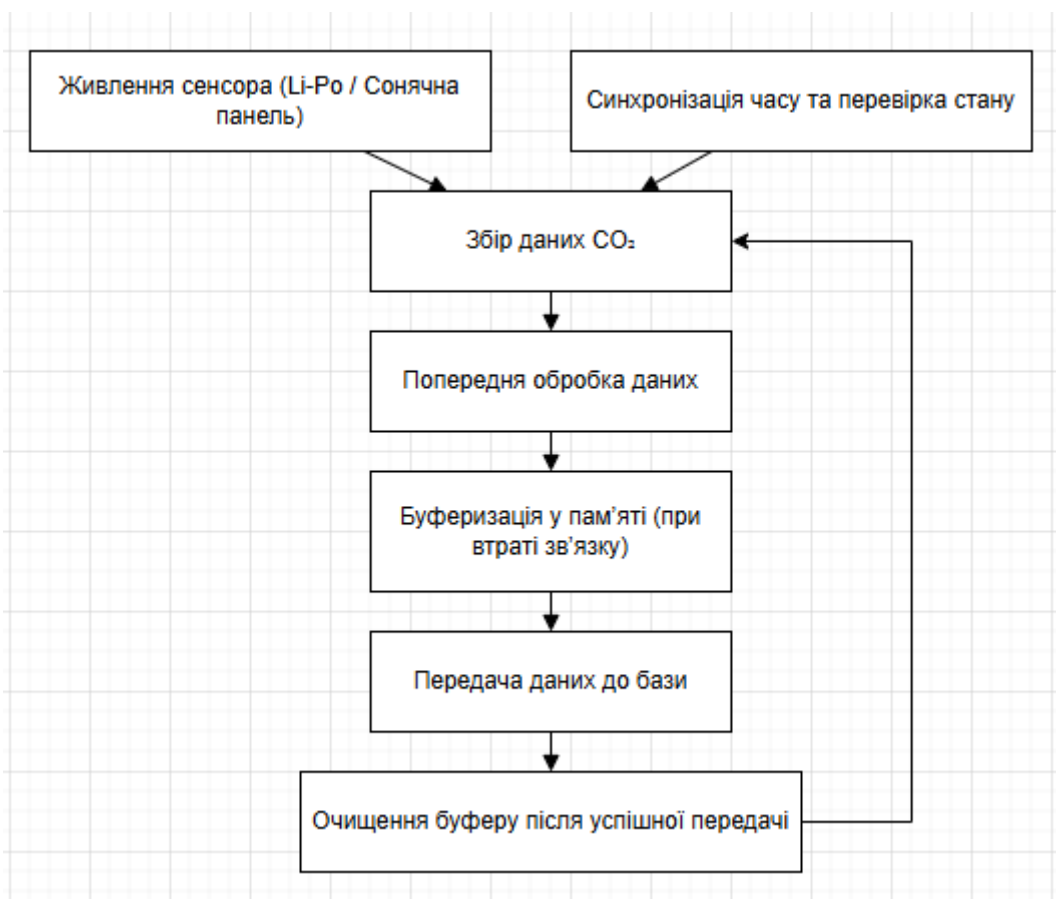


Рисунок 1.5 – Загальний цикл роботи сенсорного модуля в умовах автономного функціонування.

Отже, ключові параметри, що висуваються до сенсорної системи, зводяться до таких аспектів: акуратність, незалежність, безпека, надійність передачі інформації, можливість розширення та легкість впровадження. Саме ці чинники визначають працездатність майбутньої системи спостереження.

### 1.5 Вибір сенсора МН-Z19С для вимірювання CO<sub>2</sub>

Для створення дієвої підсистеми моніторингу парникових газів, надзвичайно важливим є вибір відповідного датчика. Цей датчик повинен забезпечувати точне, стабільне та довготривале вимірювання концентрації вуглекислого газу в атмосфері. У рамках даного проекту було вирішено використовувати сенсор МН-Z19С, як оптимальний вибір враховуючи критерії точності, надійності, енергоефективності та простоти інтеграції .

МН-Z19С - це недисперсійний інфрачервоний (NDIR) датчик, який спеціально призначений для визначення концентрації CO<sub>2</sub> в атмосферному

повітрі (Рис 1.6). Його функціонування ґрунтується на принципі поглинання інфрачервоного випромінювання молекулами вуглекислого газу [4]. Такий підхід гарантує високу вибірковість та точність навіть при тривалій експлуатації.

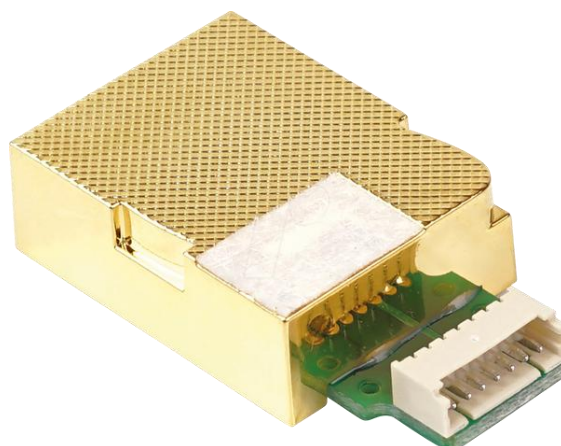


Рисунок 1.6 – Зовнішній вигляд та конструкція сенсора MH-Z19C

На рисунку нижче представлено конструктивне виконання корпусу модуля сенсора, який використовується в розробленій системі. Зображення містить вигляд зверху, збоку та ізометричну проекцію (Рис 1.7).

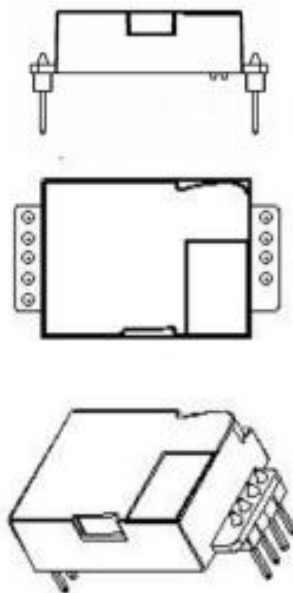


Рисунок 1.7 – Конструктивне виконання сенсорного модуля.

Однією з важливих переваг цього датчика є його здатність працювати автономно, не потребуючи частого калібрування. Він оснащений автоматичною базовою калібрувкою (ABC) для безперервної роботи. Сенсор має вбудовані аналоговий та цифровий інтерфейси (UART/PWM), що

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		20

полегшує його підключення до мікроконтролерів інтеграцію в складніші системи.

Ще одним важливим аспектом є мінімальне споживання енергії. Завдяки цьому МН-Z19С чудово підходить для автономних пристроїв, що працюють від батарей чи сонячних панелей. Крім того, сенсор відрізняється невеликими габаритами, що дозволяє інтегрувати його у компактні корпуси сенсорних модулів, включаючи герметичні варіанти для використання у польових умовах.

На відміну від дешевих аналогів, МН-Z19С гарантує справжній вимір CO<sub>2</sub>, а не тільки оцінку стану повітря за непрямими параметрами (що здебільшого трапляється з дешевими датчиками “CO<sub>2</sub>” для дому). Ось чому його все частіше використовують у наукових дослідженнях та виробництві, особливо у проектах, де потрібно слідкувати за мікрокліматом, вентиляцією, а також у сільському господарстві.

Таблиця 1.1 – Основні характеристики сенсора CO<sub>2</sub> МН-Z19С [4]

Параметр	Значення
Робоча напруга	5,0±0,1V
Середній струм	<40mA (при 5V живленні)
Піковий струм	125mA (при 5V живленні)
Дальність виявлення	0~5000ppm
Вихідний сигнал	Послідовний порт(UART) (TTL рівень 3.3V)
	PWM
Час попереднього розігріву	1 хв
Робоча температура	-10~50°C
Робоча вологість	0~95% RH (без конденсації)
Вага	5 грам
Тривалість життя	>5 років

Отже, обрання МН-Z19С як ключового датчика в цьому проекті являє собою обґрунтоване технічне рішення, що враховує як властивості газу, який вимірюється, так і вимоги до незалежності живлення, стійкості та можливості розширення системи.

## 1.6 Загальна структура системи моніторингу

Система моніторингу концентрації парникових газів, зокрема діоксиду вуглецю (CO<sub>2</sub>), представляє собою складну технічну структуру, що складається з взаємозалежних модулів (Рис 1.8). Її функціональність забезпечується злагодженою взаємодією апаратного та програмного забезпечення, що гарантує збір, збереження, передавання та аналіз екологічних даних у режимі реального часу [7]. До базових компонентів системи належать:

- Сенсорний модуль — це основний компонент, який безпосередньо вимірює концентрацію CO<sub>2</sub>. У цьому проекті як сенсор використовується МН-Z19С, що дозволяє фіксувати рівень газу з заданою частотою, навіть в умовах змінної температури та вологості [4].
- Контролер — відповідає за обробку даних з сенсора, здійснюючи попередню фільтрацію або перевірку на коректність, додає мітку часу (використовуючи RTC), зберігає показники у тимчасовій пам'яті, а також передає дані до бази даних.
- Модуль живлення — оскільки система працює автономно, живлення забезпечується літій-полімерним акумулятором з можливістю заряджання через сонячну панель.
- Модуль зв'язку — відповідає за передачу даних. Вибір технології (Wi-Fi, LoRa або мобільний зв'язок (LTE)) залежить від умов розгортання системи.
- Централізована база даних — сервер, котрий отримує, зберігає та впорядковує отримані показники. Працює на СУБД типу PostgreSQL.
- Веб-інтерфейс користувача — програмний інструмент для перегляду, аналізу та візуалізації даних про стан повітря. Доступ реалізується з браузера на ПК чи мобільному пристрої.

Система має бути здатною працювати періодично (збір даних раз на годину), забезпечуючи стабільність і надійність функціонування з мінімальною потребою у технічному обслуговуванні; до того ж, це дозволяє

					<i>15.04 - БКР.2251 "С" 24.12.16.8.ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		22

ефективно використовувати різні режими енергозбереження. Це гарантує виконання її завдань у широкому спектрі умов, що критично важливо для ефективності та довговічності.

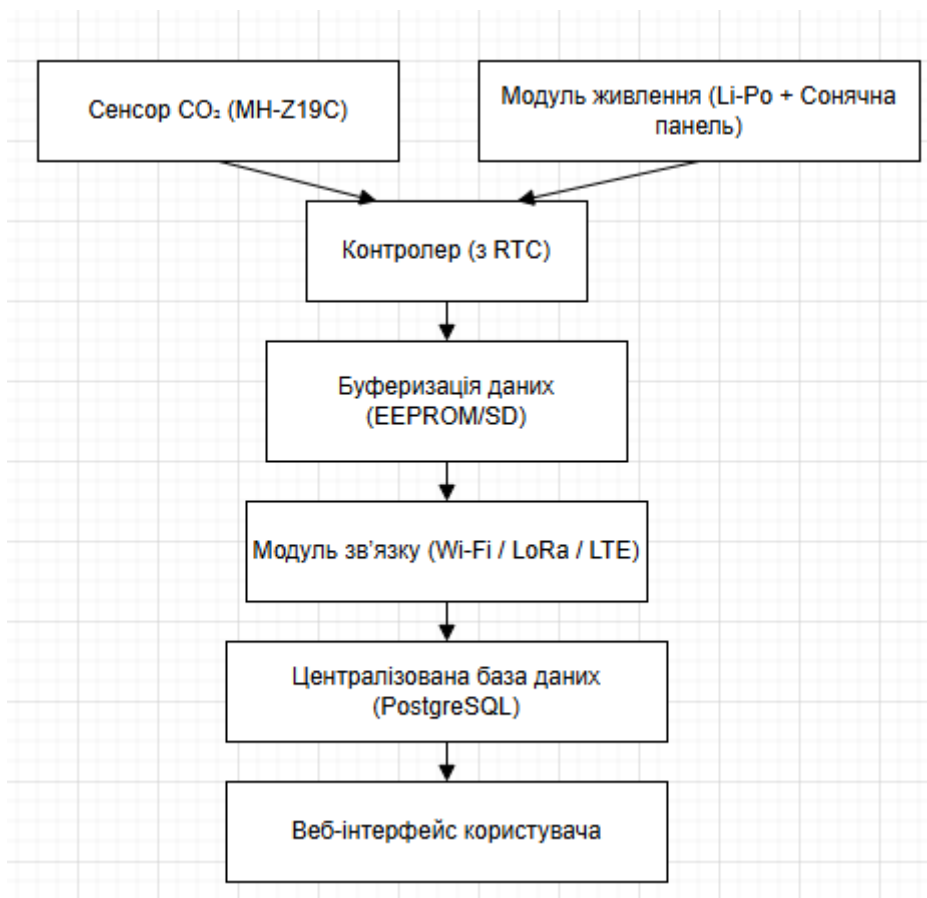


Рисунок 1.8 – Загальна структура системи моніторингу концентрації CO<sub>2</sub>

У разі втрати зв'язку сенсорний модуль не лише не припиняє свою роботу, а й продовжує виконувати всі необхідні процеси в автономному режимі. Він зберігає зібрану інформацію у власній інтегрованій пам'яті до моменту відновлення з'єднання, забезпечуючи цілісність та доступність даних без втрат або переривань.

Така архітектура дозволяє системі демонструвати високу гнучкість та адаптивність у будь-якому середовищі. Вона легко масштабується та налаштовується відповідно до умов конкретного середовища, незалежно від того, чи йдеться про складну міську інфраструктуру, чи про віддалені локації для польових досліджень у природному середовищі. Завдяки продуманій організації системи її можна ефективно використовувати для моніторингу,

збору даних, аналітики та інших важливих завдань, що потребують високої надійності.

### 1.7 Висновок до розділу 1

У першому розділі було здійснено аналітичний огляд галузі, що стосується моніторингу парникових газів, приділяючи особливу увагу контролю за концентрацією вуглекислого газу (CO<sub>2</sub>). Було представлено важливість цієї теми в контексті глобальної екологічної ситуації та потреби швидкого виявлення змін у складі повітря.

Аналіз наявних рішень показав, що, незважаючи на існування стаціонарних і комерційних систем, виникає потреба у розробці доступних, незалежних та гнучких рішень, придатних для розгортання в умовах обмеженої інфраструктури. Саме це визначає важливість створення власної сенсорної підсистеми, яка відповідала б сучасним критеріям точності, енергоефективності та стійкості.

Визначено ключові критерії до сенсорного вузла: надійне функціонування в реальних умовах, незалежне джерело живлення, періодичне оновлення інформації, зберігання даних при збоях у зв'язку, а також легке поєднання з центральною системою та призначенням для користувача інтерфейсом. Зважаючи на аналіз, обґрунтовано вибір сенсора МН-Z19С, який відповідає технічним вимогам системи та оптимально підходить для реалізації в рамках проекту.

Окрім того, розроблено загальну структурну модель системи спостереження, що складається з сенсорного модуля, контролера, модулі в комунікації та живлення, серверної частини й інструментів візуалізації. Ця модель послужить фундаментом для подальшого проектування та впровадження завершеного рішення в наступних частинах дослідження.

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		24

## РОЗДІЛ 2. ПРОЄКТУВАННЯ СЕНСОРНОЇ ПІДСИСТЕМИ

### 2.1 Архітектура підсистеми

Сенсорна підсистема становить собою визначальний функціональний компоненту структурі системи контролю рівня CO<sub>2</sub>, адже він безпосередньо відповідає за початковий збір інформації про поточний стан повітряного середовища. Його завданням є забезпечення автономної діяльності, досягнення високої точності вимірювань, а також надійне збереження й транспортування показників до централізованого сховища даних.

Архітектурний устрій підсистеми передбачає взаємодію апаратного та програмного компонентів, які спільно утворюють автономний елемент для збирання даних про навколишнє середовище [8].

Базова структура сенсорного модуля складається з таких основних елементів (Рис 2.1):

- Газовий сенсор МН-Z19С – головний вимірювальний елемент, що здійснює контроль за концентрацією вуглекислого газу в повітрі.
- Контролер – прилад, що керує опитуванням датчика, обробкою отриманої інформації, створенням структури пакету, узгодженням часу та підготовкою даних до передачі.
- RTC-модуль (Real-Time Clock) – гарантує надійне відстеження часу кожної фіксації вимірювань, незважаючи на наявність чи відсутність зв'язку з мережею.
- Буфер пам'яті – EEPROM чи SD-карта, що використовуються для тимчасового зберігання даних у випадку збою зв'язку із сервером.
- Модуль зв'язку – відповідає за пересилання інформації до централізованого сховища. Вид з'єднання здатен підлаштовуватися до навколишніх факторів — можливе використання Wi-Fi, LoRa чи LTE.
- Живлення – літій-полімерний акумулятор з функцією підзарядки від сонячного світла гарантує цілковиту незалежність модуля, навіть там, де доступ ускладнено.

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		25

Уся взаємодія між компонентами підсистеми відбувається в межах одного циклу: отримання значення → обробка → збереження → передача. У випадку, коли передача даних не може відбутися негайно, інформація зберігається у буфері пам'яті до відновлення стабільного з'єднання. Це забезпечує цілісність даних і їхню надійність.

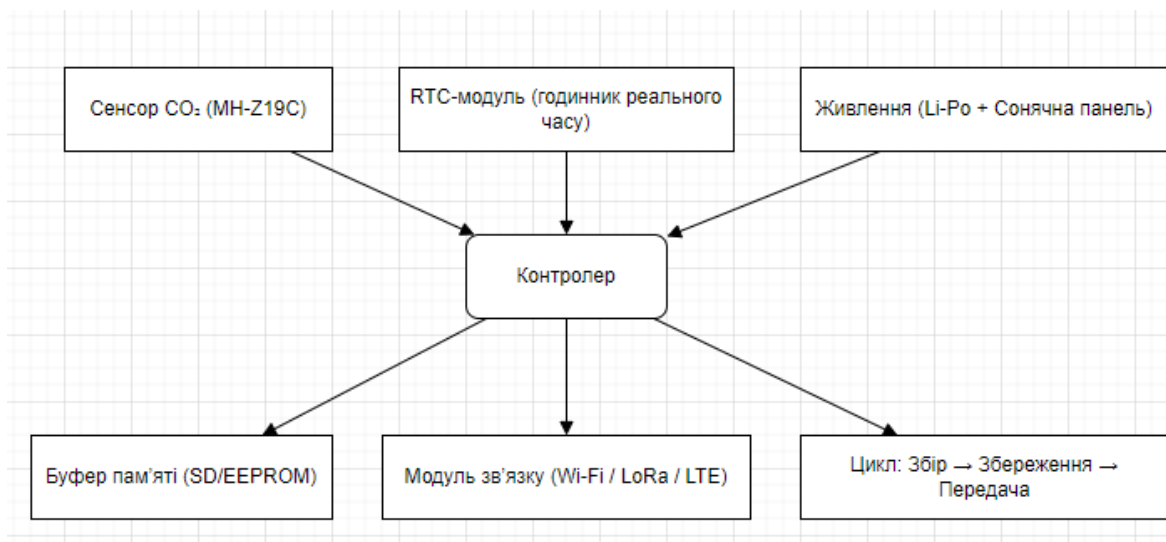


Рисунок 2.1 – Архітектура сенсорного вузла для вимірювання концентрації CO<sub>2</sub>

Архітектура створена з урахуванням можливості масштабування, що означає, що додавання нових сенсорних вузлів до мережі не вимагає змін у програмному забезпеченні чи структурі бази даних. Кожен модуль працює незалежно та має власний унікальний ідентифікатор, що дозволяє обробляти дані окремо для кожного з них.

Отже, спроектована архітектура сенсорної підсистеми задовольняє всі основні потреби, що стосуються надійності, можливості розширення, ефективності використання енергії та здатності функціонувати в умовах нестійкого з'єднання.

Для наочності нижче представлено з'єднання сенсора MH-Z19C з мікроконтролером Arduino Uno за допомогою UART-інтерфейсу. Дотримання схеми забезпечує стабільну передачу даних (Рис 2.2).

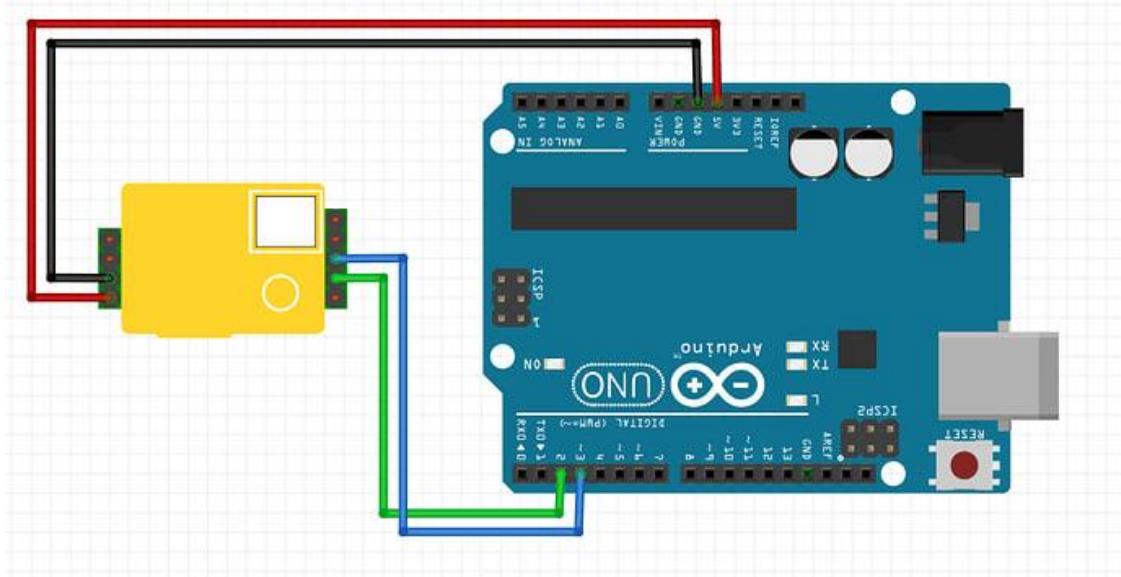


Рисунок 2.2 – Макетне з'єднання сенсора MH-Z19C із Arduino Uno»

Для наочного відображення зібраного апаратного блоку сенсорної підсистеми нижче представлено фото, що демонструє з'єднання сенсора MH-Z19C з Arduino Uno відповідно до схеми (Рис 2.3).

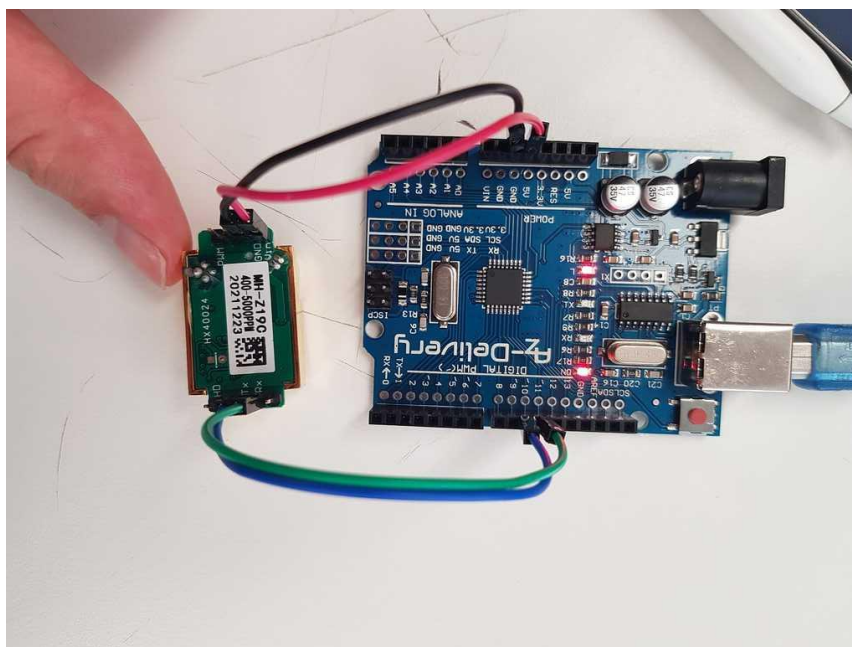


Рисунок 2.3 – З'єднання сенсора MH-Z19C з контролером Arduino Uno

Змін.	Арк.	№ докум.	Підпис	Дата

## 2.2 Схема взаємодії компонентів

Щоб система спостереження за рівнем CO<sub>2</sub> працювала без перебоїв та з високою точністю, всі її складники повинні функціонувати як цілісний ланцюг, гарантуючи скоординований рух даних від датчика до кінцевого інтерфейсу для користувача. В цьому розділі розглядається принцип взаємодії між фізичними й програмними модулями сенсорного блоку та центральною системою.

Взаємодія стартує з сенсора MH-Z19C, котрий періодично здійснює вимірювання концентрації CO<sub>2</sub> у навколишньому просторі. Контролер ініціює зчитування даних з сенсора через інтерфейс UART, фіксує отримані показники разом із часовою позначкою (від RTC-модуля) у внутрішню змінну.

Наступним етапом є опрацювання зібраної інформації: контрольна відповідність дозволеним значенням, округлення, приведення до вигляду JSON – пакета чи іншої структури, виходячи з протоколу обміну. У випадку наявності з'єднання дані передаються через комунікаційний модуль до центральної бази даних. Якщо ж з'єднання немає, інформація зберігається в локальній пам'яті (SD або EEPROM), і буде передана пізніше — після відновлення зв'язку.

В центральному вузлі системи дані отримує сервер (наприклад, використовуючи HTTP POST або MQTT). Після перевірки на відповідність, показники фіксуються в базі даних. Звідти вони можуть бути відображені у форматі графіків чи таблиць через веб-інтерфейс. Даний інтерфейс надає можливість проводити аналіз, переглядати історію показників та здійснювати моніторинг актуального стану сенсорної мережі.

Взаємодія елементів сенсорної системи вибудована за чітко визначеною послідовністю. У базовій конфігурації використовується мікроконтролер ESP32, що забезпечує бездротову передачу даних через Wi-Fi. Модуль годинника реального часу (RTC) визначає час активації, після чого контролер запускає прогрів сенсора MH-Z19C та здійснює зчитування показників CO<sub>2</sub>. При наявності підключення дані передаються на сервер бази даних

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		28

PostgreSQL. У випадку відсутності з'єднання інформація тимчасово зберігається в буфері пам'яті (EEPROM або SD-карті). Фінальний доступ до зібраних даних реалізовано через веб-інтерфейс користувача.

### 2.3 Логіка обробки даних

Сенсорна підсистема системи контролю CO<sub>2</sub> функціонує на основі циклічного процесу: отримання вимірювань, їх попередня обробка, збереження у буфері та відправка екологічних даних.

Цикл розпочинається з ініціації модулем реального часу (RTC), який будить мікроконтролер у заздалегідь визначений момент часу (скажімо, 00:00, 01:00, 02:00 або інші). Після того, як пристрій "прокидається", він активує процес прогріву сенсора МН-Z19С, адже точність його показів безпосередньо залежить від температурної стабільності внутрішніх елементів. Зчитування концентрації CO<sub>2</sub> здійснюється лише після закінчення цього прогріву.

Отримавши результат, контролер формує пакет даних, до якого додає мітку часу та ідентифікатор вузла. Якщо контролер підключено до мережі, пакет негайно надсилається на сервер. У випадку, коли з'єднання немає, дані тимчасово зберігаються у вбудованій пам'яті. Завершивши всі операції, контролер переходить у режим сну, що сприяє економії енергії до надходження наступного сигналу пробудження від RTC.

Кожен з цих кроків виконується повторно, відповідно до заданого часового інтервалу, який встановлено на одну годину. Це означає, що система щогодини проводить повний цикл операцій, зокрема збір і обробку даних, що дозволяє завжди мати актуальний стан вимірювань та оперативно реагувати на можливі зміни в навколишньому середовищі.

Продумана логіка обробки даних не лише забезпечує високу точність вимірювань, а й інтегрує механізми, що гарантують стійкість до численних потенційних проблем. Наприклад, за наявності переривань зв'язку, збою в роботі обладнання або навіть повного відключення живлення система розрахована на автоматичне відновлення робочого процесу. У таких випадках

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		29

вона може застосовувати резервні алгоритми, які, після відновлення з'єднання або електроживлення, безперешкодно продовжують роботу, званою логікою компенсуючи втрати або затримки.

Кожного разу, коли система проводить цикл вимірювань, здійснюється ретельний аналіз отриманих даних з подальшим збереженням результатів у відповідних контейнерах для логування. Це дозволяє не лише відслідкувати коректність роботи в режимі реального часу, а й забезпечити можливість подальшої діагностики або аналізу у разі виявлення неполадок. Завдяки цьому підходу будь-яке відхилення від норми, виявлене системою, може бути оперативно ідентифіковано, що сприяє своєчасному усуненню проблем.

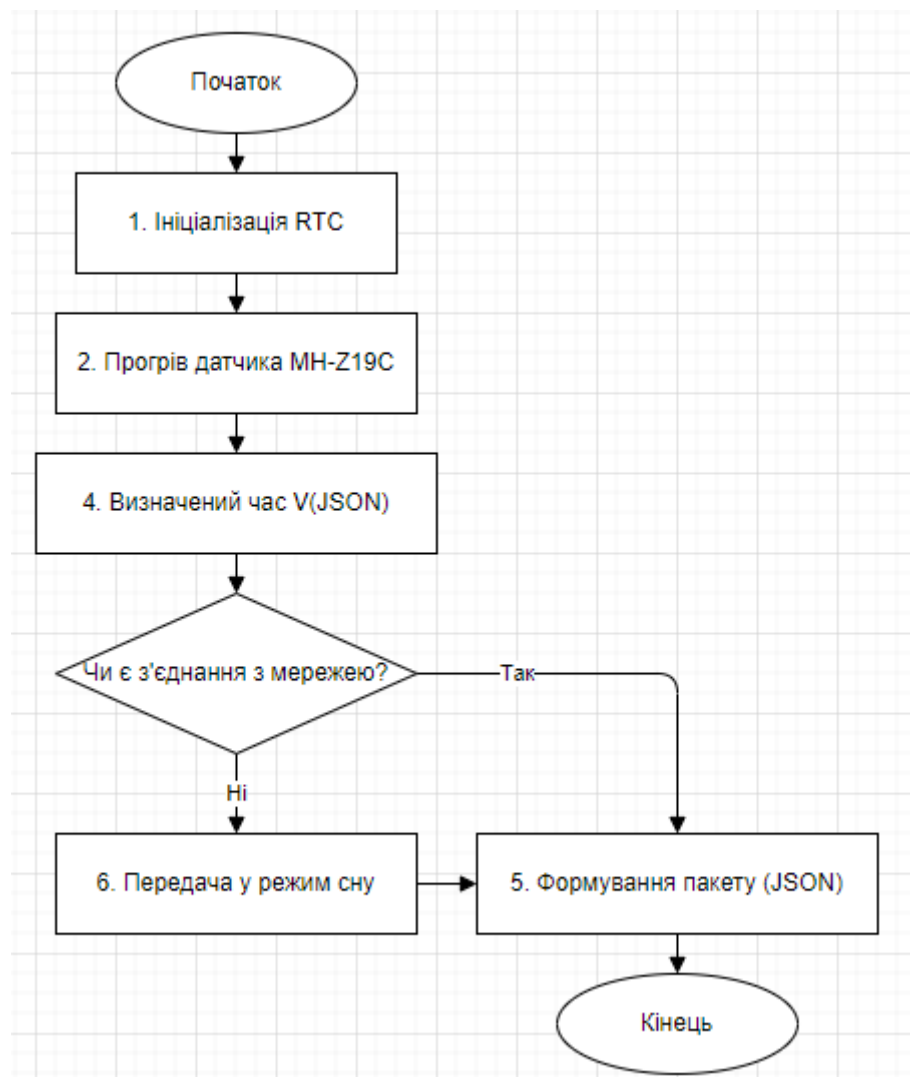


Рисунок 2.4 – Алгоритм роботи сенсорного вузла з ініціацією від RTC, прогрівом сенсора, обробкою та переходом у режим сну.

Крім того, завдяки інтеграції резервних алгоритмів і систем самоусунення, будь-які потенційні проблеми, пов'язані з збоями обладнання чи перериваннями зв'язку, не впливають на загальну якість вимірювань. При відновленні нормальної роботи, система автоматично перевіряє останні отримані дані і знову ініціює цикл вимірювань, завдяки чому відновлення обслуговування проходить максимально плавно і без втрати даних.

Алгоритм роботи сенсорного модуля, що враховує ініціацію від модуля реального часу (RTC), прогрів сенсора, логіку передачі або буферизації даних та перехід у режим сну, зображено на (Рис 2.4):

- *Зчитування показника CO<sub>2</sub>*. Датчик MH-Z19C, реагуючи на запит з контролера, здійснює вимірювання концентрації CO<sub>2</sub> в атмосфері. Далі контролер отримує цифровий показник, виражений в ppm (частинах на мільйон).

- *Додавання мітки часу*. Відразу після зчитування даних, контролер звертається до RTC-модуля. Отримавши інформацію, він долучає до зібраних показників точну позначку часу. Це дає змогу прив'язати отримані дані до конкретного моменту в часі[5].

- *Перевірка допустимості значень*. Алгоритм, що виконується контролером ,здійснює перевірку, чи відповідає показник CO<sub>2</sub> заданому діапазону допустимих значень (скажімо, 300–5000 ppm). Значення, які явно не відповідають фізично обґрунтованим обмеженням, відкидаються або отримують статус потенційно недостовірних [9].

- *Формування пакету даних*. Показник CO<sub>2</sub>, мітка часу та ідентифікатор сенсорного вузла групуються в структуру (наприклад, JSON), що придатна для передавання або зберігання. Прийняття рішення: передача чи буферизація.

- *Прийняття рішення: передача чи буферизація*. Якщо на момент обробки даних з'єднання з мережею є активним, пакет передається через модуль комунікації на сервер. У разі відсутності зв'язку дані зберігаються у спеціальному буфері (SD-карта або EEPROM).

– *Очистка локального буфера.* Після отримання від сервера підтвердження успішного прийому, буфер очищується автоматично. Усі виконані дії фіксуються в лог-контейнері.

## 2.4 Висновок до розділу 2

У другому розділі розглянуто структуру сенсорної системи для моніторингу концентрації CO<sub>2</sub>, встановлено її основні елементи та взаємодію між ними. З особливою увагою проаналізовано запуск циклу збору інформації, використовуючи модуль реального часу (RTC), що забезпечує синхронізацію по всій мережі. Наведено механізм попереднього прогріву сенсора МН-Z19С, зчитування даних, а також умовну передачу або буферизацію інформації, залежно від доступності з'єднання. Додатково розглянуто режим сну контролера як спосіб економії електроенергії. Схеми та текстові описи демонструють повний цикл функціонування сенсорного вузла в автономному режимі.

Окреслено способи взаємодії між складовими системи, що охоплюють весь шлях-від зчитування даних з датчика CO<sub>2</sub> до їх передавання у базу даних, а також відображення у веб-інтерфейсі. Логіка роботи передбачає процес обробки даних з реалізацією буферизації на випадок втрати зв'язку, що збільшує стійкість системи в реальних умовах [Помилка! Джерело посилання не знайдено.].

З огляду на розроблені структури та алгоритми опрацювання даних, стає очевидним, що спроектована сенсорна підсистема цілковито задовольняє визначені функціональні потреби. Її можна застосовувати як незалежно, так і в складі розгалуженої мережі екологічного спостереження. Запропонована архітектура послужить основою для подальших апаратних та програмних розробок, які буде розглянуто в наступному розділі.

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		32

## РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1 Програмна реалізація збору даних

Збір даних щодо концентрації вуглекислого газу у повітрі здійснюється з використанням датчика MH-Z19C, що базується на методі недисперсійної інфрачервоної спектроскопії (NDIR). Запуск збору даних контролюється модулем реального часу (RTC), який пробуджує контролер зі сну в заздалегідь визначений час (скажімо, кожну годину о 00:00, 01:00 і так далі). Після активації, контролер спочатку запускає прогрів сенсора, і лише після цього відбувається зчитування значення CO<sub>2</sub>. Зібрана інформація обробляється та передається на сервер, або ж буферизується у разі втрати з'єднання. Після виконання всіх операцій пристрій знову переходить у режим сну до наступного сигналу від RTC.

Програмний код, що працює на контролері, виконує наступну послідовність дій: зчитування показань сенсора → запис часу події → створення пакету даних → відправлення або збереження. Для фіксації часу вимірювання використовується вбудований модуль реального часу (RTC), який дає змогу синхронізувати дані між різними пристроями в мережі.

Нижче наведено фрагмент логіки збору даних, представлений у вигляді узагальненого коду (Дод. Д, стр. 0014 – 0023):

```
int co2ppm = readCO2();
DateTime now = rtc.now();
StaticJsonDocument<200> doc;
doc["device_id"] = "sensor_01";
doc["timestamp"] = getTimestamp(now);
doc["co2"] = co2ppm;
String output;
serializeJson(doc, output);
Serial.println(output);
delay(3600000); // 1 година
```

Представлений фрагмент містить функції зчитування даних з сенсора МН-Z19С через UART, отримання поточної дати та часу з модуля RTC, а також формування пакету у форматі JSON для подальшої передачі чи логування. Кожна з функцій реалізована згідно до особливостей обладнання, використаного в системі (контролер, сенсор, модуль часу), і забезпечує стабільну роботу сенсорного вузла в автономному режимі.

```
{ "co2": 417, "timestamp": "2025-05-07T14:00:00" }
```

Рисунок 3.1 – Вивід зчитаного рівня CO<sub>2</sub> з міткою часу.

Результат виконання збору даних із сенсора МН-Z19С (Рис 3.1):

```
{ "co2": 417, "timestamp": "2025-05-07T14:00:00" }
```

Отриманий пакет даних має змогу зберігатися на місці у сховищі модуля (SD-карта чи EEPROM), або відправлятися на віддалений сервер. Передача може відбуватися по бездротовому зв'язку: mesh мережі LoRa, чи через мобільний інтернет, використовуючи LTE-модем. Спосіб взаємодії системи збирається відповідно до інфраструктури, де цю систему розгорнуто [].

### 3.2 Збереження даних у разі втрати зв'язку

Надійність функціонування сенсорної системи моніторингу CO<sub>2</sub> значною мірою зумовлена її здатністю зберігати інформацію, навіть коли з'єднання з центральним сховищем відсутнє. У практичних умовах розгортання в полі зв'язок може перериватися через низку факторів: відсутність сигналу, проблеми на сервері або ж обмеження в енергозабезпеченні. Відтак, в архітектурі системи реалізовано механізм локального резервного зберігання даних — буферизацію [**Помилка! Джерело посилання не знайдено.**].

Цей процес відбувається на рівні контролера. Після зчитування та аналізу показника CO<sub>2</sub>, контролер перевіряє, чи встановлено з'єднання з мережею. Якщо зв'язок відсутній, сформований пакет даних, який містить показник CO<sub>2</sub>, позначку часу та ідентифікатор вузла, записується в локальну пам'ять, наприклад:

1. На SD-карту (для великих обсягів);
2. У вбудовану EEPROM (якщо доступна).

Кожен рядок у буфері даних зберігається у структурованому форматі (JSON чи CSV), супроводжуючись інформацією про статус передачі: передано / не передано [Помилка! Джерело посилання не знайдено.].

Логіку перевірки статусу кожного запису в буфері та умовної повторної передачі даних наведено на рисунку 3.2.

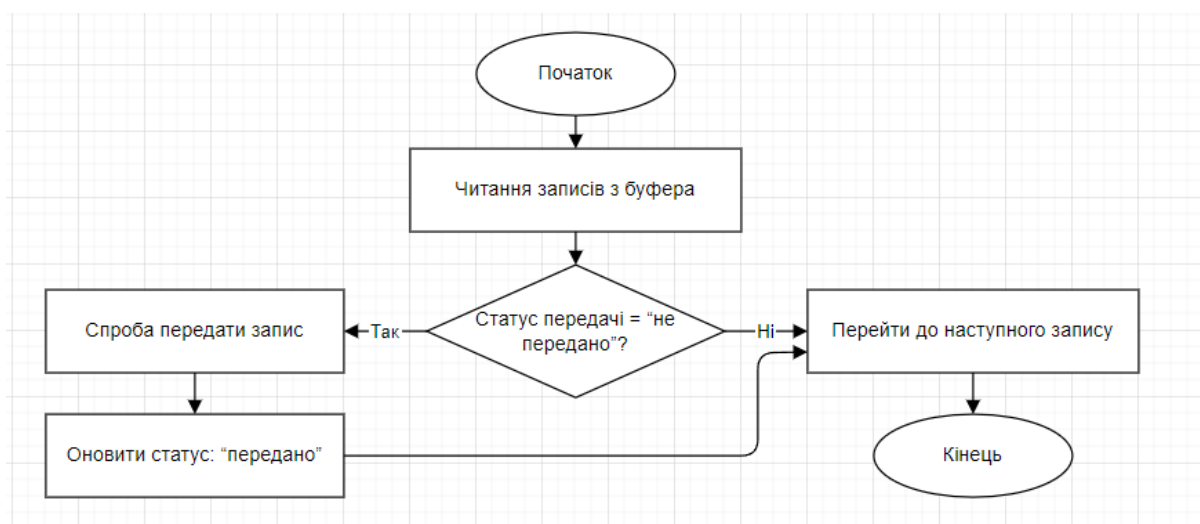


Рисунок 3.2 – Алгоритм перевірки буфера та повторної передачі даних у разі статусу “не передано”.

Фрагмент логіки буферизації (Дод. А, стр. 0018 – 0028):

```

int co2ppm = co2Sensor.getCO2();
String timestamp = rtc.getTimestamp();
String packet = formatPacket(co2ppm, timestamp);
if (network.isConnected()) {
    bool sent = network.send(packet);
    if (sent) {
        buffer.clear();
    }
} else {
    buffer.save(packet);
}
  
```

Після відновлення з'єднання система одразу переходить до активної фази синхронізації даних. Контролер, який виконує роль мозку всіх операцій, автоматично звертається до буфера — спеціального тимчасового сховища, де накопичувалися всі дані за період перебування системи в автономному режимі. Це дозволяє відновити інформацію, яка накопичилась під час втрати зв'язку, не дозволяючи жодному її фрагменту залишитись непереданим.

Контролер надсилає ці дані на сервер, дотримуючись суворої хронологічної послідовності. Такий порядок передачі гарантує, що всі події та вимірювання зберігають свою первісну послідовність, що є критично важливим для подальшої обробки та аналізу інформації. Завдяки цьому не виникає змішування даних, а кожна подія повертається до своєї правильної позиції у загальній картині інформаційного потоку.

Перш ніж контролер дозволить собі видалити оброблені дані з локального сховища, він чітко переконується, що сервер успішно прийняв кожен порцію інформації. Лише після отримання підтвердження від серверу, яке є маркером цілісності та точності доставки, дані звільняються з буфера. Такий методичний підхід практично усуває ризик втрати інформації, адже жодна частина даних не зникає без надійного підтвердження.

Завдяки цьому комплексному механізму збереження та передачі даних, система може безпечно функціонувати навіть у разі тривалих періодів відключення від мережі — навіть якщо таке відключення триває кілька годин або навіть цілий день. Такий дизайн не тільки мінімізує можливість виникнення помилок, але й забезпечує стабільну, надійну роботу системи незалежно від зовнішніх умов і непередбачуваних мережевих збоїв (Рис 3.2).

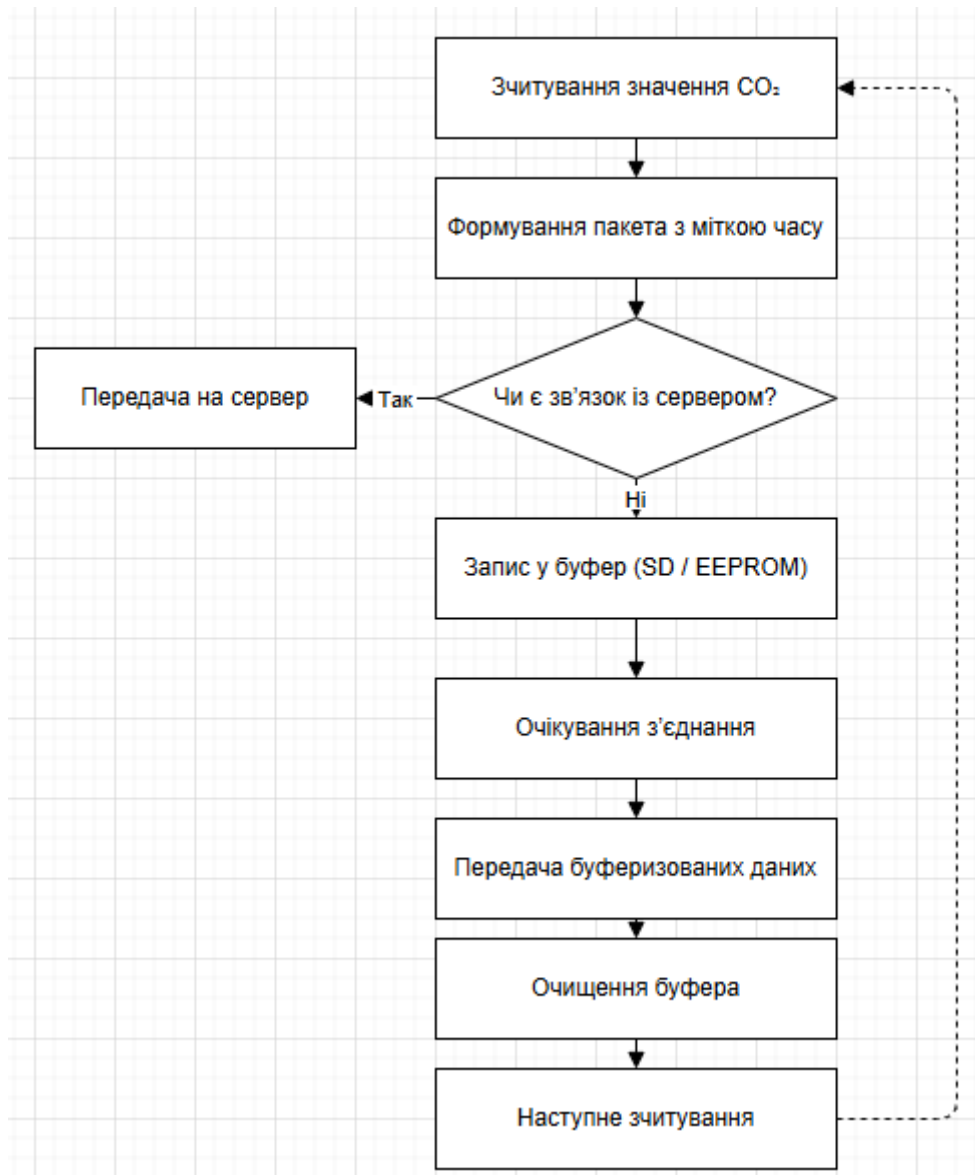


Рисунок 3.3 – Логіка збереження та синхронізації даних у разі втрати зв'язку.

Для гарантування стабільної роботи сенсорної системи в реальному середовищі, впроваджено механізми обробки непередбачуваних ситуацій. У разі переривання зв'язку, дані автоматично зберігаються у незалежній пам'яті до відновлення з'єднання. Коли пристрій перезавантажується — контролер зчитує буфер та відправляє збережені дані. При відключенні живлення інформація не зникає, оскільки вона записується у flash або EEPROM [Помилка! Джерело посилання не знайдено.]. Додатково передбачено повторне надсилення даних, якщо сервер не підтвердив отримання пакету (Табл. 3.1).

Таблиця 3.1 – Реакція сенсорної системи на аварійні ситуації та нештатні події.

Ситуація	Поведінка системи
Зникнення мобільного зв'язку	Дані буферизуються у EEPROM або SD-карті. Передача виконується при наступному з'єднанні
Перезапуск або перезавантаження контролера	RTC зберігає поточний час. Контролер після старту читає буфер і надсилає дані
Відключення живлення	Інформація не втрачається — вона записана у незалежній пам'яті
Пошкодження пакета	У разі помилки при передачі сервер не підтверджує прийом — ініціюється повторна відправка
Відсутність пробудження у визначений час	RTC модуль гарантує пробудження системи. У разі збою – контролер ініціалізує резервну перевірку часу
Завершення циклу	Після виконання всіх дій пристрій переходить у режим сну до наступного сигналу від RTC

Від так, система гарантує спостереження, навіть за наявності ускладнень, а усі екологічні параметри передаються до бази даних без втрати інформації, після поновлення з'єднання.

### 3.3 Передача даних до централізованої бази

Одним з ключових етапів функціонування сенсорної системи моніторингу CO<sub>2</sub> виступає передача отриманих показників до централізованого серверу. Тут дані зберігатимуться, оброблятимуться та

візуалізуватимуться. Цей крок завершує цикл збору даних, забезпечуючи сполучення сенсорної частини з усією інфраструктурою.

Дані, отримані з сенсора МН-Z19С, проходять попередню обробку на контролері (формування пакету, додавання мітки часу), після чого передаються до централізованої бази даних — PostgreSQL, як зазначено у вимогах. Передача може відбуватись через різні комунікаційні протоколи, зокрема:

- HTTP POST – якщо використовується інтернет-з'єднання (LTE або 5G);
  - MQTT – якщо система реалізована на основі mesh-мережі (LoRa)
- [Помилка! Джерело посилання не знайдено.]

Передача виконується тільки при наявності стабільного з'єднання. У разі відсутності зв'язку дані тимчасово буферизуються та надсилаються після його відновлення (Рис 3.4).

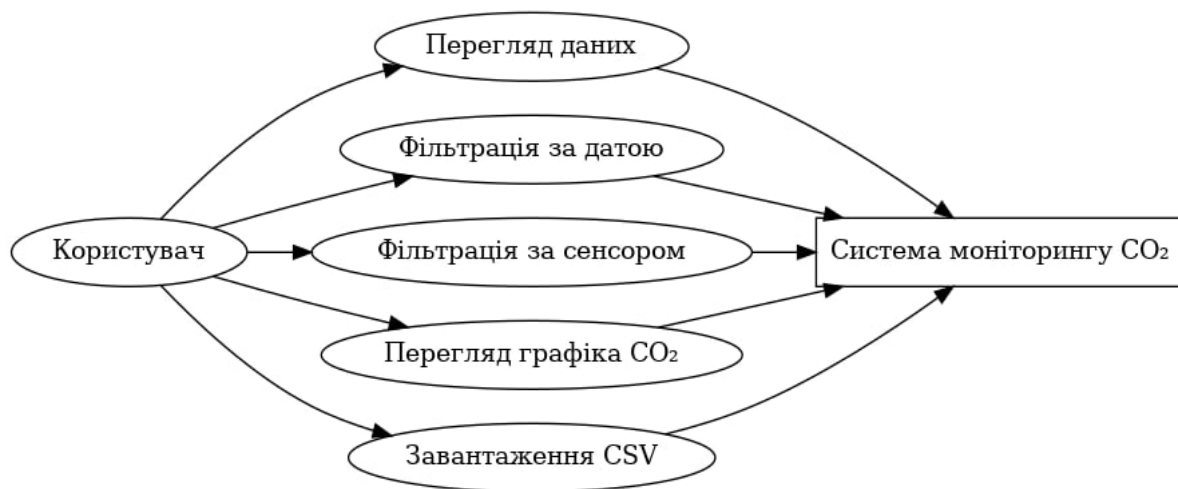


Рисунок 3.4 – Діаграма варіантів використання: взаємодія користувача з системою моніторингу CO<sub>2</sub>.

Фрагмент коду (Дод. А , стр. 0039 – 0045):

```
void sendToServer(String data) {  
    HTTPClient http;  
    http.begin("http://192.168.0.100:5000/data");  
    http.addHeader("Content-Type", "application/json");  
    int httpResponseCode = http.POST(data);  
    http.end();  
}
```

}

На серверному боці працює спеціалізований приймальний модуль, який є центральним елементом обробки вхідних даних. Він займається реєстрацією кожного отриманого пакета, фіксуючи детальну інформацію, таку як час надходження, ідентифікація пакета та інші метадані, що є критично важливими для подальшого аналізу та аудиту. Цей процес дозволяє відслідковувати всі події в системі в реальному часі, забезпечуючи точність і послідовність даних [Помилка! Джерело посилання не знайдено.].

Після реєстрації кожного пакета модуль переходить до етапу перевірки, який включає кілька важливих кроків [Помилка! Джерело посилання не знайдено.]:

- *Оцінка структури*: модуль ретельно аналізує внутрішню організацію пакета, перевіряючи, чи відповідає розташування і формат даних встановленим стандартам і протоколам. Це знижує ризик обробки пошкоджених або некоректних даних.

- *Перевірка дати*: важливим аспектом є валідність дати або часової мітки, що міститься в пакеті. Це гарантує, що дані є актуальними і мають правильний часовий контекст, виключаючи можливість помилкового аналізу або обробки застарілої інформації.

- *Перевірка формату*: модуль проводить аналіз відповідності формату даних: перевіряється, чи використовуються правильні типи даних (наприклад, числові, текстові, формати дати) та збережено правильне кодування інформації.

Виконання цих перевірок дозволяє забезпечити високу якість даних перед їх подальшим збереженням.

Після успішної валідації даних приймальний модуль переходить до збереження інформації. Отримані дані автоматично направляються для запису в конкретну таблицю бази даних PostgreSQL [17]. Структура таблиці ретельно продумана для збереження великого обсягу інформації, з урахуванням її різноманітності та потреби в подальшій обробці (Табл. 3.2).

Таким чином, серверний приймальний модуль забезпечує не лише реєстрацію і первинну перевірку кожного надходячого пакета, а й гарантує систематизоване збереження даних за стандартами PostgreSQL [18]. Це дозволяє системі підтримувати високу надійність, ефективну масштабованість та легкість подальшої обробки, що є надзвичайно важливими для стабільної роботи в умовах великого обсягу інформації та високих вимог до якості даних(Рис 3.4).

Таблиця 3.2 – Структура таблиці sensor\_data у базі даних PostgreSQL.

Поле	Тип даних	Опис
id	SERIAL	Первинний ключ, унікальний ідентифікатор запису
device id	TEXT	Ідентифікатор сенсорного вузла
Timestamp	TIMESTAMP	Час проведення вимірювання
co2	INTEGER	Зафіксоване значення CO <sub>2</sub> у ppm

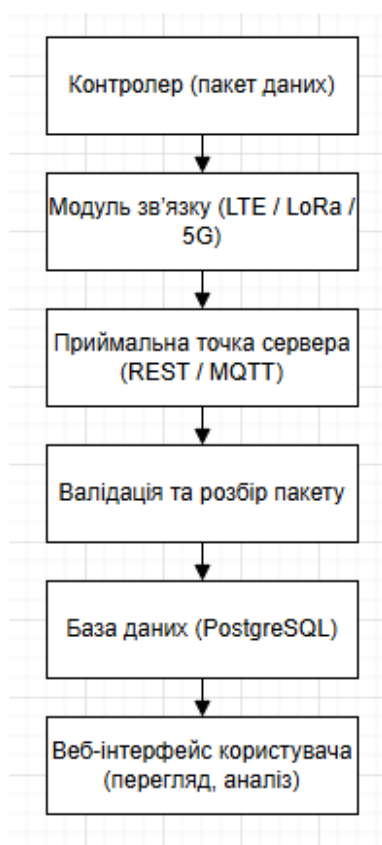


Рисунок 3.5 – Процес передачі екологічних даних до централізованої бази даних PostgreSQL.

Виконав ініціалізацію Flask-додатку `app = Flask(__name__)`. Цей рядок створює об'єкт Flask, який є основою веб-програми. Через цей об'єкт у подальшому задаються маршрути (ендпоінти) та обробники запитів.

Фрагмент коду (Функція збереження даних у PostgreSQL) (Дод Б , стр 0008 -0026)

```
def save_to_db(data):
    conn = psycopg2.connect(
        dbname="co2_monitoring",
        user="postgres",
        password="your_password",
        host="localhost"
    )
    cur = conn.cursor()
    cur.execute("""
INSERT INTO sensor_data (device_id, timestamp, co2, received_at)
VALUES (%s, %s, %s, %s)
""", (data["device_id"], data["timestamp"], data["co2"],
datetime.datetime.now()))
    conn.commit()
    conn.close()
```

Функція `save_to_db` відкриває з'єднання з базою даних PostgreSQL, готує SQL-запит на вставку даних у таблицю `sensor_data`, виконує його та зберігає новий запис, що містить [9]:

- ідентифікатор пристрою (`device_id`);
- час вимірювання (`timestamp`);
- значення CO<sub>2</sub> (`co2`);
- фактичний час отримання пакета сервером (`received_at`).

Фрагмент коду (Маршрут API для приймання даних від сенсорів) (Дод. Б , стр. 0028 – 0032):

```
@app.route("/api/data", methods=["POST"])
```

```
def api_data():
    data = request.get_json()
    save_to_db(data)
    return {"status": "success"}
```

Це визначення маршруту /api/data, який обробляє POST-запити. Зазвичай сюди надсилає JSON-серіалізований пакет даних контролер з сенсором. Функція приймає запит, витягує з нього дані, зберігає їх у БД через save\_to\_db() і повертає JSON-відповідь про успішну операцію [19].

Фрагмент коду (Маршрут веб-інтерфейсу для перегляду даних) (Дод. Б, стр. 0034 - 0045):

```
@app.route("/", methods=["GET"])
def index():
    device_id = request.args.get("device_id")
    date_filter = request.args.get("date")

    conn = psycopg2.connect(
        dbname="co2_monitoring",
        user="postgres",
        password="your_password",
        host="localhost"
    )
    cur = conn.cursor()

    query = "SELECT device_id, timestamp, co2 FROM sensor_data WHERE
1=1"
```

Цей маршрут виконується при зверненні до головної сторінки веб-інтерфейсу (/). Він підключається до БД, витягує 100 останніх записів з таблиці sensor\_data, зберігає результат у rows і передає його до шаблону index.html для відображення.

Фрагмент коду (HTML-шаблон для виводу даних) (Дод. В, стр. 0027 - 0044):

```

<table class="table table-bordered">
  <thead>
    <tr>
      <th>Сенсор</th>
      <th>Час вимірювання</th>
      <th>CO2 (ppm)</th>
      <th>Час отримання</th>
    </tr>
  </thead>
  <tbody>
    {% for row in data %}
      <tr>
        <td>{{ row[0] }}</td>
        <td>{{ row[1] }}</td>
        <td>{{ row[2] }}</td>
        <td>{{ row[3] }}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>

```

HTML-шаблон для веб-інтерфейсу, який відображає таблицю з даними. Усі значення заповнюються динамічно за допомогою Jinja2: `{% for row in data %}` перебирає список записів із Python-коду, а `{{ row[i] }}` вставляє відповідні значення в HTML.

### 3.4 Тестування

Метою тестування сенсорної підсистеми є верифікація її функціонування, оцінка стійкості роботи в різних конфігураціях, перевірка можливості збереження інформації при розриві з'єднання, а також відповідність оголошеним функціональним специфікаціям. Тестування виконувалось як в

умовах моделювання, так і в реальних умовах, що максимально нагадують умови експлуатації.

Під час тестування функціоналу системи провели перевірку основних компонентів архітектури, що базується на мікроконтролері ESP32. Зокрема, було досліджено правильність запуску циклу збирання даних від модуля годинника реального часу (RTC) в обумовлені часові інтервали, відповідність температурного режиму сенсора МН-Z19С технічним параметрам, а також точність показників після його прогріву. З'ясовано, що передача даних через Wi-Fi функціонує без збоїв за умови стабільного з'єднання. У разі втрати зв'язку дані зберігаються в EEPROM (або наSD-карті) та автоматично надсилаються після відновлення підключення. Додатково було відтворено ситуацію з навмисним припиненням з'єднання з метою перевірки механізму буферизації, який показав свою ефективність.

Тестування поділялось на кілька етапів:

1. Перевірка зчитування значень CO<sub>2</sub>.

- i. *Мета:* переконатись, що сенсор МН-Z19С стабільно зчитує дані через UART.
- ii. *Критерій успішності:* отримання числових значень у діапазоні 0–5000 ppm.
- iii. *Інструменти:* серійний монітор, журнал виводу даних.

2. Перевірка точності часової мітки.

- i. *Мета:* перевірити роботу RTC-модуля та правильність фіксації часу вимірювання.
- ii. *Критерій успішності:* відповідність часу показанням системного або реального годинника.
- iii. *Метод:* порівняння виводу з системним логом.

3. Буферизація даних при втраті зв'язку.

- i. *Мета:* оцінити роботу алгоритму локального збереження.
- ii. *Умови:* примусове вимкнення модуля зв'язку.
- iii. *Критерій успішності:* збереження даних у буфері без втрат.

4. Синхронізація даних після відновлення зв'язку.

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		45

- i. *Мета:* перевірити, чи надсилаються буферизовані дані після повернення з'єднання.
  - ii. *Критерій успішності:* відправка всіх накопичених записів без дублювання.
5. Передача даних на сервер і запис у базу.
- i. *Мета:* перевірити стабільність API або брокера (REST/MQTT), правильність формату запиту.
  - ii. *Критерій:* наявність записів у таблиці sensor\_data з актуальними значеннями CO<sub>2</sub>.
6. Тестування інтерфейсу користувача.
- i. *Мета:* перевірити, чи дані з бази відображаються у веб-інтерфейсі.
  - ii. *Критерій:* візуалізація таблиці або графіку в браузері без затримки чи помилок.
7. Тестування функції експорту даних.
- i. *Мета:* перевірити працездатність механізму збереження показників CO<sub>2</sub> з веб-інтерфейсу у форматі CSV.
  - ii. *Критерій:* коректне формування та збереження CSV-файлу з актуальними даними з бази, правильне відображення вмісту файлу при відкритті в табличному редакторі.

Тестування здійснювалося із застосуванням звичних інструментів розробки (послідовний монітор, ведення журналів, перевірка через веб).

На рисунку зображено веб-інтерфейс, який дозволяє користувачеві обрати сенсор, встановити дату фільтрації та завантажити дані у форматі CSV для подальшої обробки. (Рис. 3.6)

Останні показники CO<sub>2</sub>

The image shows a web interface with the following elements:
 

- A text input field labeled "Сенсор (device\_id)".
- A date input field labeled "дд.мм.рррр" with a calendar icon.
- A blue button labeled "Фільтрувати".
- A green button labeled "Завантажити CSV".

Рисунок 3.6 – Елементи інтерфейсу для фільтрації даних та експорту CSV.

Для перевірки функціональності інтерфейсу користувача було протестовано механізм фільтрації за датою. Він дозволяє обрати потрібний день у календарі та переглянути відповідні значення CO<sub>2</sub>, що значно полегшує роботу з великим обсягом даних. (Рис. 3.7)

### Останні показники CO<sub>2</sub>

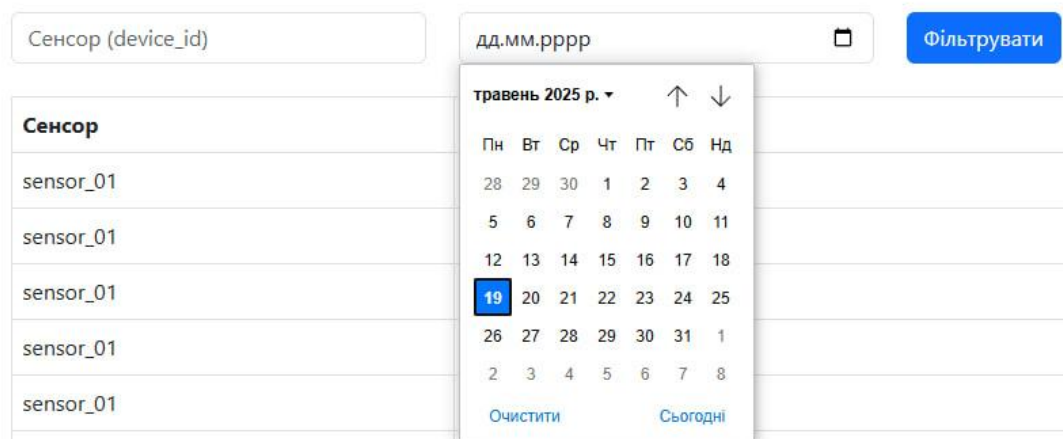


Рисунок 3.7 – Вибір дати у веб-інтерфейсі для фільтрації показників CO<sub>2</sub>.

Реалізація цього елемента інтерфейсу забезпечує зручність користування системою, підвищує швидкість доступу до потрібної інформації та підтверджує відповідність веб-інтерфейсу сучасним вимогам до UI/UX при роботі з даними моніторингу.

Результати оцінювалися за допомогою елементарних критеріїв — відповідність реальної поведінки системи передбачуваним функціональним випадкам. (Рис 3.8)

```
[14:00] CO2: 417 ppm | Статус: буферизовано (зв'язку немає)
[15:00] CO2: 420 ppm | Статус: буферизовано (зв'язку немає)
[16:00] Мережа відновлена
[16:00] Передано 2 пакети з буфера
[16:00] CO2: 422 ppm | Статус: передано
```

Рисунок 3.8 – Результат перевірки буферизації даних при втраті зв'язку.

### 3.5 Результати тестування та їх аналіз

На основі аналізу тестів, що розглядалися вище, було виявлено, що сенсорна підсистема, яка відповідає за збирання, обробку та передачу інформації про рівень вуглекислого газу, функціонує стабільно і без проблем (Табл. 3.3).

Тестування відбувалося двома фазами: спочатку з емульованими умовами (для перевірки логіки алгоритмів), а потім – у реальних умовах, з періодичним штучним розривом мережевого з'єднання. Результати фіксувалися у логах та додатково верифікувалися на стороні серверу у базі даних PostgreSQL.

Таблиця 3.3 – Основні результати тестування сенсорної підсистеми.

№	Етап тестування	Результат	Оцінка
1	Зчитування даних з МН-Z19С	Дані стабільно отримувались кожен годину	Успішно
2	Визначення часу	Мітки часу відповідали фактичному часу вимірювання	Успішно
3	Втрата зв'язку	Дані буферизувались у пам'яті без втрат	Успішно
4	Передача після відновлення зв'язку	Усі накопичені записи були передані без дублювання	Успішно
5	Запис у базу PostgreSQL	Дані коректно зберігались у таблиці sensor_data	Успішно
6	Відображення у веб-інтерфейсі	Дані відображались в реальному часі у вигляді таблиці	Успішно

Додатковою функціональністю веб-інтерфейсу є можливість експорту даних у форматі CSV. Після натискання відповідної кнопки дані автоматично формуються у файл, який зберігається на пристрої користувача. (Рис. 3.9)



Рисунок 3.9 – Завантаження файлу sensor\_data.csv із веб-інтерфейсу системи моніторингу.

Завантажений CSV-файл містить ідентифікатор сенсора, час вимірювання та значення CO<sub>2</sub>, що дозволяє швидко інтегрувати отримані дані в інші системи або обробляти їх у табличному редакторі.

	A	B	C
1	device_id	timestamp	co2
2	sensor_01	12.05.2025 0:00	444
3	sensor_01	12.05.2025 1:00	437
4	sensor_01	12.05.2025 2:00	426
5	sensor_01	12.05.2025 3:00	408
6	sensor_01	12.05.2025 4:00	435
7	sensor_01	12.05.2025 5:00	424
8	sensor_01	12.05.2025 6:00	429
9	sensor_01	12.05.2025 7:00	405
10	sensor_01	12.05.2025 8:00	424
11	sensor_01	12.05.2025 9:00	410
12	sensor_01	12.05.2025 10:00	447
13	sensor_01	12.05.2025 11:00	429
14	sensor_01	12.05.2025 12:00	427
15	sensor_01	12.05.2025 13:00	438
16	sensor_01	12.05.2025 14:00	402
17	sensor_01	12.05.2025 15:00	426
18	sensor_01	12.05.2025 16:00	438
19	sensor_01	12.05.2025 17:00	407
20	sensor_01	12.05.2025 18:00	439
21	sensor_01	12.05.2025 19:00	401
22	sensor_01	12.05.2025 20:00	419
23	sensor_01	12.05.2025 21:00	427
24	sensor_01	12.05.2025 22:00	434
25	sensor_01	12.05.2025 23:00	421
26			

Рисунок 3.10 – Вміст CSV-файлу, збереженого з веб-інтерфейсу системи моніторингу CO<sub>2</sub>.

Після натискання кнопки «Завантажити CSV» формується файл формату CSV, який може бути відкритий у табличних редакторах (наприклад, Excel). Файл містить колонки з ідентифікатором сенсора, міткою часу та значенням CO<sub>2</sub>, що повністю відповідає структурі таблиці в базі даних. (Рис. 3.10).

Наявність графічної візуалізації є важливою складовою інтерфейсу, яка підвищує зручність користування системою та дає змогу виявляти закономірності у змінах показників без потреби аналізу таблиць або зовнішніх інструментів. (Рис. 3.11)

Графік CO<sub>2</sub>



Рисунок 3.11 – Динаміка зміни концентрації CO<sub>2</sub> протягом доби за результатами роботи сенсорної підсистеми

Отримані результати свідчать про відповідність реалізованого рішення всім функціональним вимогам:

- надійність — жодного втрачено значення навіть при втраті мережі;
- стабільність — жодного зависання або помилок при передачі;
- масштабованість — логіка підтримує додавання нових вузлів без зміни серверної частини;
- інтерфейсність — усі дані доступні для візуального перегляду в реальному часі.

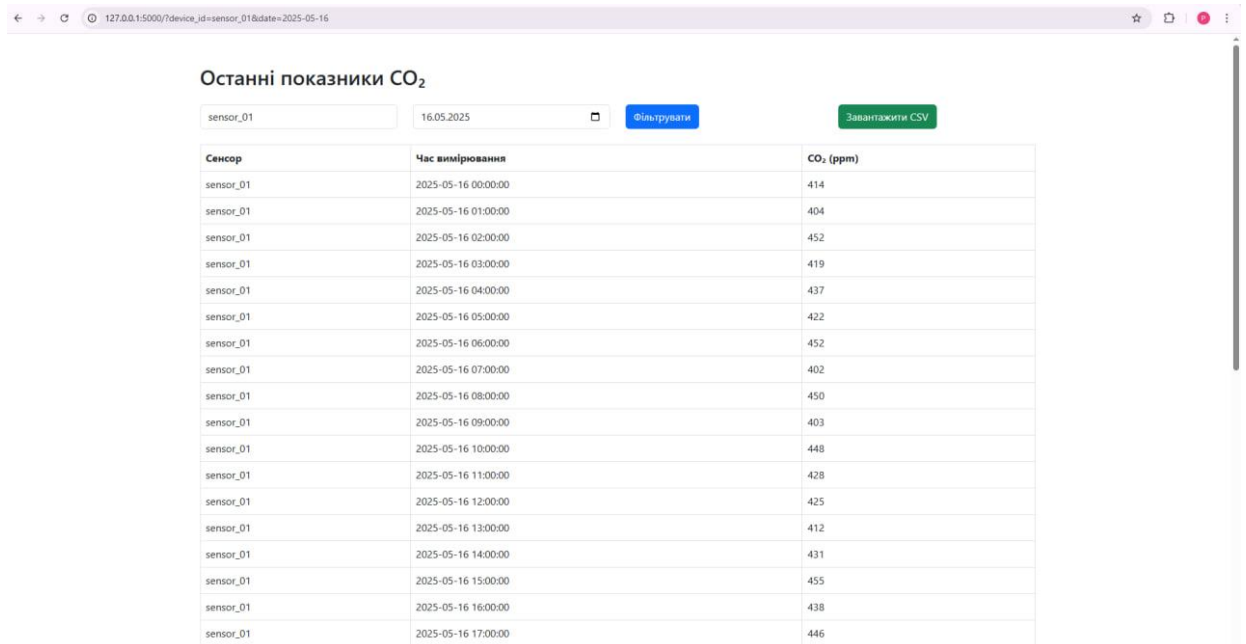


Рисунок 4.7 – Відображення даних CO<sub>2</sub> у веб-інтерфейсі користувача.

Отже, система продемонструвала відмінну роботу, функціонуючи в умовах невизначеності, і повністю відповідає тим критеріям, що були визначені у початкових вимогах проекту.

### 3.6 Висновок до розділу 3

У третьому розділі дипломної роботи розроблено повно функціональне програмне забезпечення сенсорної підсистеми, призначене для автоматичного моніторингу концентрації CO<sub>2</sub> у повітрі. Розділ детально розглядає всі необхідні етапи: збір даних, буферизацію, передачу інформації, її збереження та наочне представлення екологічних показників.

Розроблено процедуру збору інформації. Запуск відбувається за сигналом від модуля годинника реального часу (RTC). Після цього відбувається прогрів сенсора MH-Z19C. Далі відбувається зчитування показників концентрації CO<sub>2</sub>. Формується структурований пакет у форматі JSON, до якого додається мітка часу. Дані, залежно від наявності зв'язку, або відправляються до централізованої бази, або зберігаються у внутрішній пам'яті при відсутності з'єднання.

Окрему увагу зосереджено на впровадженні механізму буферизації: у випадку обриву з'єднання, пакети зберігаються в EEPROM або на SD-карті зі статусом "не передано". Після відновлення підключення автоматично виконується синхронізація даних у вірній часовій послідовності, з обов'язковим підтвердженням доставки. Це гарантує цілісність, повноту і достовірність даних, навіть у випадках аварійного функціонування мережі.

На серверній частині було створено веб-сервер, використовуючи Flask, з REST-кінцевими точками для отримання пакетів та їх подальшого запису до бази даних PostgreSQL. Перед самим збереженням, кожен дані проходять обов'язкову валідацію формату, а також перевірку часу вимірювання і структури пакету.

Сформовано таблицю `sensor_data`, де зберігаються: ідентифікатор пристрою, момент вимірювання, рівень CO<sub>2</sub> та час надходження пакету. Це дає можливість моніторити дані з багатьох вузлів в режимі реального часу.

Окрім бекенду, було розроблено веб-інтерфейс для користувачів, через який можна переглядати актуальні вимірювання в табличному форматі, на графіках та завантажувати дані у форматі CSV. Система пропонує фільтрацію за сенсором та датою, що полегшує навігацію серед великих обсягів інформації. Для забезпечення зручності, було застосовано елементи UI-фреймворку Bootstrap та інтегровано Chart.js.

Окремо було здійснено функціональне тестування всіх компонентів системи в умовах моделювання та часткового відключення мережі.

Перевірено:

- стабільність зчитування сенсором MH-Z19C;
- точність фіксації часу вимірювань;
- роботу буферизації в умовах втрати зв'язку;
- повторну передачу та підтвердження даних;
- збереження записів у базі PostgreSQL;
- відображення і фільтрацію у веб-інтерфейсі;
- експорт даних у CSV;
- візуалізацію графіків у браузері.

Результати проведених випробувань підтвердили стійку роботу системи, її відповідність технічним параметрам, а також продемонстрували високу надійність, здатність до масштабування та придатність для практичного використання у реальних умовах.

Отже, у цьому розділі було розроблено закінчене та функціональне програмне рішення, яке гарантує збір, обробку та наочну візуалізацію даних про концентрацію вуглекислого газу, враховуючи потреби в автономності, безвідмовності та адаптивності при застосуванні у сфері екологічного моніторингу.

## ЗАГАЛЬНІ ВИСНОВКИ

В ході виконання дипломної роботи на тему "Розробка підсистеми сенсорів газів у системі моніторингу концентрації парникових газів" було створено повнофункціональну сенсорну підсистему для автоматизованого збирання, зберігання та візуалізації екологічних даних стосовно концентрації вуглекислого газу (CO<sub>2</sub>). Ця задача набуває особливої ваги в контексті глобального потепління та зростаючого значення цифрових технологій в екологічному моніторингу.

На аналітичному етапі було проведено огляд сучасних технологій та рішень у сфері моніторингу повітря, зокрема газових сенсорів, їх принципів роботи, переваг і недоліків. Здійснено порівняльний аналіз існуючих типів сенсорів — напівпровідникових, електрохімічних і NDIR, що дозволило обґрунтовано вибрати сенсор МН-Z19С як основний компонент розроблюваної системи. Сенсор відзначається високою точністю, стабільністю та енергоефективністю, що робить його придатним для використання в автономних умовах.

У проєктному розділі було розроблено архітектуру сенсорного вузла, що включає апаратну частину (сенсор МН-Z19С, мікроконтролер, RTC, модулі живлення та зв'язку) та логіку роботи, адаптовану до автономного функціонування. Детально спроектовано цикл вимірювання: ініціалізація пробудження від RTC, прогрів сенсора, зчитування значень CO<sub>2</sub>, формування пакету даних з позначкою часу, перевірка з'єднання та подальша передача або буферизація. У разі відсутності зв'язку дані зберігаються у внутрішній пам'яті (EEPROM або SD-карта) до моменту відновлення підключення.

Третій розділ був присвячений реалізації програмного забезпечення. Було застосовано мову Python та фреймворк Flask для створення серверної частини системи. Розроблено REST-інтерфейс для прийому даних, здійснено збереження у централізовану базу даних PostgreSQL, реалізовано перевірку цілісності пакету, структури, формату та валідності мітки часу. Таблиця

					<i>15.04 - БКР.2251 "С" 24.12.16.8.ПЗ</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		53

sensor\_data дозволяє фіксувати показники з великої кількості пристроїв, що забезпечує масштабованість рішення.

Окремо розроблено веб-інтерфейс для користувачів, що дозволяє переглядати останні вимірювання у таблиці або на графіках, фільтрувати дані за сенсором або датою, а також експортувати у форматі CSV. Для реалізації інтерфейсу використано фреймворк Bootstrap і бібліотеку Chart.js. Це забезпечило не тільки зручність користування, але й можливість інтеграції системи у більш складні аналітичні екосистеми.

На завершальному етапі роботи проведено всебічне тестування системи — як у стабільному середовищі, так і в умовах моделювання аварійних ситуацій. Перевірено стабільність зчитування сенсором, точність фіксації часу, працездатність буферизації при втраті зв'язку, автоматичну синхронізацію даних, запис у базу та коректну візуалізацію інформації. Всі функції системи спрацювали відповідно до очікувань.

У результаті виконаної роботи вдалося досягти поставленої мети — створити надійну, автономну, масштабовану сенсорну підсистему, що забезпечує збір, обробку та наочне представлення даних про концентрацію CO<sub>2</sub> в атмосфері. Всі заявлені завдання — від технічного проєктування до впровадження серверної інфраструктури та інтерфейсу — були виконані в повному обсязі.

Розроблене рішення може бути використане як окрема екологічна система для спостереження за атмосферними показниками або як складова більшої мережі моніторингу, наприклад, у сільському господарстві, міському управлінні, виробничих комплексах чи науково-дослідних майданчиках. Воно демонструє високу надійність та потенціал для подальшого розвитку, зокрема інтеграції з хмарними платформами, мобільними додатками або інтелектуальними системами прогнозування.

Отже, дипломна робота не тільки підтверджує теоретичні та практичні знання автора, а й має цінність як готове технічне рішення, яке може знайти реальне застосування у сфері екологічного моніторингу.

					<i>15.04 - БКР.2251 "С" 24.12.16.8.ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		54

Код програмного забезпечення контролера системи моніторингу CO<sub>2</sub>:

```

0001 #include "MHZ19.h" // Бібліотека для сенсора MH-Z19C
0002 #include "RTC_Module.h" // Модуль реального часу (RTC)
0003 #include "NetworkModule.h" // Інтерфейс зв'язку: LTE / LoRa / інше
0004 #include "StorageModule.h" // Модуль буферизації: EEPROM або SD
0005 h> // Для передачі даних через HTTP
0006 MHZ19 co2Sensor;
0007 RTC_Module rtc;
0008 NetworkModule network;
0009 StorageModule buffer;
0010 void setup() {
0011     Serial.begin(9600); // Ініціалізація UART
0012     co2Sensor.begin(); // Ініціалізація сенсора
0013     // Ініціалізація модуля реального часу
0014     network.init(); // Ініціалізація зв'язку
0015     buffer.init(); // Ініціалізація буфера
0016 }
0017 void loop() {
0018     int co2ppm = co2Sensor.getCO2(); // Зчитування CO2
0019     String timestamp = rtc.getTimestamp(); // Мітка часу
0020     String packet = formatPacket(co2ppm, timestamp);
0021     if (network.isConnected()) {
0022         bool sent = network.send(packet); // Надсилання пакету
0023         if (sent) {
0024             buffer.clear(); // Очищення буфера після передачі
0025         }
0026     } else {
0027         ; // Буферизація у разі втрати зв'язку
0028     }
0029     (3600000); // Очікування 1 год перед наступним циклом
0030 }
0031 String formatPacket(int co2, String time) {
0032     String data = "{}";
0033     data += "\"co2\":" + String(co2) + ",";
0034     data += "\"timestamp\":" + time + "\"";
0035     data += "}";
0036     return data;
0037 }

```

```

0038 // Реалізація функції відправки даних на сервер через HTTP POST
0039 void sendToServer(String data) {
0040     HTTPClient http;
0041     // Адреса API сервера
0042     http.addHeader("Content-Type", "application/json"); // Заголовок для JSON
0043     // Надсилання POST-запиту
0044     http.end(); // Завершення з'єднання
0045 }

```

Змін.	Арк.	№ докум.	Підпис	Дата

## ДОДАТОК Б

Код серверної частини (Flask):

```
0001 from flask import Flask, request, render_template, send_file
0002 import psycopg2
0003 import datetime
0004 import csv
0005 import io
0006 app = Flask(name)
0007 # Функція збереження даних у базу PostgreSQL
0008 def save_to_db(data):
0009     conn = psycopg2.connect(
0010         dbname="co2_monitoring",
0011         user="postgres",
0012         password="your_password",
0013         host="localhost"
0014     )
0015     cur = conn.cursor()
0016     cur.execute("""
0017         INSERT INTO sensor_data (device_id, timestamp, co2, received_at)
0018         VALUES (%s, %s, %s, %s)
0019         """, (
0020         data["device_id"],
0021         data["timestamp"],
0022         data["co2"],
0023         datetime.datetime.now()
0024     ))
0025     conn.commit()
0026     conn.close()
0027 # Маршрут для прийому даних від сенсорів
0028 @app.route("/api/data", methods=["POST"])
0029 def api_data():
0030     data = request.get_json()
0031     save_to_db(data)
0032     return {"status": "success"}
0033 # Головна сторінка для перегляду даних
0034 @app.route("/", methods=["GET"])
0035 def index():
0036     device_id = request.args.get("device_id")
0037     date_filter = request.args.get("date")
```

					15.04 - БКР.2251 "С" 24.12.16.8.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		1

```

0038     conn = psycopg2.connect(
0039         dbname="co2_monitoring",
0040         user="postgres",
0041         password="your_password",
0042         host="localhost"
0043     )
0044     cur = conn.cursor()
0045     query = "SELECT device_id, timestamp, co2 FROM sensor_data WHERE 1=1"
0046     params = []
0047     if device_id:
0048         query += " AND device_id = %s"
0049         params.append(device_id)
0050     if date_filter:
0051         query += " AND DATE(timestamp) = %s"
0052         params.append(date_filter)
0053     query += " ORDER BY timestamp ASC LIMIT 100"
0054     cur.execute(query, tuple(params))
0055     data = cur.fetchall()
0056     conn.close()
0057     return render_template("index.html", data=data)
0058 # Завантаження CSV-файлу з бази даних
0059 @app.route("/download", methods=["GET"])
0060 def download():
0061     conn = psycopg2.connect(
0062         dbname="co2_monitoring",
0063         user="postgres",
0064         password="your_password",
0065         host="localhost"
0066     )
0067     cur = conn.cursor()
0068     cur.execute("SELECT device_id, timestamp, co2 FROM sensor_data ORDER BY
timestamp ASC")
0069     rows = cur.fetchall()
0070     conn.close()
0071     si = io.StringIO()
0072     cw = csv.writer(si)
0073     cw.writerow(["device_id", "timestamp", "co2"])
0074     cw.writerows(rows)
0075     output = io.BytesIO()
0076     output.write(si.getvalue().encode('utf-8'))
0077     output.seek(0)

```

Змін.	Арк.	№ докум.	Підпис	Дата

0078 return send\_file(output, mimetype="text/csv", as\_attachment=True,  
download\_name="co2\_data.csv")

Змін.	Арк.	№ докум.	Підпис	Дата	15.04 - БКР.2251 "С" 24.12.16.8.ПЗ		3 59

Код створення веб-інтерфейсу:

```

0001 <!DOCTYPE html>
0002 <html lang="uk">
0003 <head>
0004   <meta charset="UTF-8">
0005   <title>Моніторинг CO2</title>
0006   <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
0007   <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
0008 </head>
0009 <body class="container mt-5">
0010   <h2 class="mb-4">Останні показники CO2</h2>
0011
0012   <form method="get" class="row mb-4">
0013     <div class="col">
0014       <input type="text" name="device_id" class="form-control"
placeholder="Сенсор (device_id)">
0015     </div>
0016     <div class="col">
0017       <input type="date" name="date" class="form-control">
0018     </div>
0019     <div class="col">
0020       <button type="submit" class="btn btn-primary">Фільтрувати</button>
0021     </div>
0022     <div class="col">
0023       <a href="/download" class="btn btn-success">Завантажити CSV</a>
0024     </div>
0025   </form>
0026
0027   <table class="table table-bordered">
0028     <thead>
0029       <tr>
0030         <th>Сенсор</th>
0031         <th>Час вимірювання</th>
0032         <th>CO2 (ppm)</th>
0033       </tr>
0034     </thead>

```

```

0035 <tbody>
0036   {% for row in data %}
0037   <tr>
0038     <td>{{ row[0] }}</td>
0039     <td>{{ row[1] }}</td>
0040     <td>{{ row[2] }}</td>
0041   </tr>
0042   {% endfor %}
0043 </tbody>
0044 </table>
0045
0046 <h4 class="mt-5">Графік CO2</h4>
0047 <canvas id="co2Chart" width="400" height="150"></canvas>
0048 <script>
0049   const ctx = document.getElementById('co2Chart').getContext('2d');
0050   const co2Chart = new Chart(ctx, {
0051     type: 'line',
0052     data: {
0053       labels: [{% for row in data reversed %}"{{ row[1] }}",{% endfor %}],
0054       datasets: [{
0055         label: 'CO2 (ppm)',
0056         data: [{% for row in data reversed %}{{ row[2] }},{% endfor %}],
0057         fill: false,
0058         borderColor: 'rgb(75, 192, 192)',
0059         tension: 0.1
0060       }]
0061     }
0062   });
0063 </script>
0064 </body>
0065 </html>

```

SQL запит для створення централізованого сховища отриманих даних:

- SQL-структура таблиці sensor\_data:

```
0001 CREATE TABLE sensor_data (  
0002     id SERIAL PRIMARY KEY,  
0003     device_id TEXT,  
0004     timestamp TIMESTAMP,  
0005     co2 INTEGER,  
0006 );
```

					<i>15.04 - БКР.2251 "С" 24.12.16.8.ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		1

Код програмного забезпечення контролера для збору даних CO<sub>2</sub> та формування JSON-пакету:

```

0001 #include <SoftwareSerial.h>
0002 #include <Wire.h>
0003 #include <RTCLib.h>
0004 #include <ArduinoJson.h>
0005 SoftwareSerial co2Serial(10, 11); // RX, TX
0006 RTC_DS3231 rtc;
0007 void setup() {
0008   Serial.begin(9600);
0009   co2Serial.begin(9600);
0010   Wire.begin();
0011   rtc.begin();
0012 }
0013 void loop() {
0014   int co2ppm = readCO2();
0015   DateTime now = rtc.now();
0016   StaticJsonDocument<200> doc;
0017   doc["device_id"] = "sensor_01";
0018   doc["timestamp"] = getTimestamp(now);
0019   doc["co2"] = co2ppm;
0020   String output;
0021   serializeJson(doc, output);
0022   Serial.println(output);
0023   delay(3600000); // 1 година
0024 }
0025 int readCO2() {
0026   byte cmd[] = {0xFF, 0x01, 0x86, 0, 0, 0, 0, 0, 0x79};
0027   byte response[9];
0028   co2Serial.write(cmd, 9);
0029   delay(100);
0030   if (co2Serial.available() >= 9) {
0031     co2Serial.readBytes(response, 9);
0032     if (response[0] == 0xFF && response[1] == 0x86) {
0033       return response[2] * 256 + response[3];
0034     }
0035     return -1; // помилка
0036   }

```

```
0037 String getTimestamp(DateTime dt) {
0038     char buf[25];
0039     sprintf(buf, "%04d-%02d-%02dT%02d:%02d:%02d", dt.year(), dt.month(),
0040             dt.day(), dt.hour(), dt.minute(), dt.second());
0041     return String(buf);
0042 }
```

					<i>15.04 - БКР.2251 "С" 24.12.16.8.ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		2

Щогодинні вимірювання концентрації CO<sub>2</sub> одним сенсором:

Сенсор	Час вимірювання	CO <sub>2</sub> (ppm)
sensor_01	2025-05-12 00:00:00	444
sensor_01	2025-05-12 01:00:00	437
sensor_01	2025-05-12 02:00:00	426
sensor_01	2025-05-12 03:00:00	408
sensor_01	2025-05-12 04:00:00	435
sensor_01	2025-05-12 05:00:00	424
sensor_01	2025-05-12 06:00:00	429
sensor_01	2025-05-12 07:00:00	405
sensor_01	2025-05-12 08:00:00	424
sensor_01	2025-05-12 09:00:00	410
sensor_01	2025-05-12 10:00:00	447
sensor_01	2025-05-12 11:00:00	429
sensor_01	2025-05-12 12:00:00	427
sensor_01	2025-05-12 13:00:00	438
sensor_01	2025-05-12 14:00:00	402
sensor_01	2025-05-12 15:00:00	426
sensor_01	2025-05-12 16:00:00	438
sensor_01	2025-05-12 17:00:00	407
sensor_01	2025-05-12 18:00:00	439
sensor_01	2025-05-12 19:00:00	401
sensor_01	2025-05-12 20:00:00	419
sensor_01	2025-05-12 21:00:00	427
sensor_01	2025-05-12 22:00:00	441
sensor_01	2025-05-12 23:00:00	400

					<i>15.04 - БКР.2251 "С" 24.12.16.8.ПЗ</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		1

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кузнецов О.О., Семенов С.І. Автоматизовані системи контролю середовища. – К.: Ліра-К, 2020. – 204 с. Вимоги до розробки сенсорної системи моніторингу парникових газів [Електронний ресурс] / НУБіП України – 2025. – 15 с.
2. Газові сенсори та застосування у сенсоричі: навч. посіб./ Л. П. Олесенко, Н. П. Максимович, І. П. Матушко, Г. В. Федоренко. – К.: ВПЦ «Київський університет», 2019,. – 143 с.
3. Il-Doо Kim, Rothschild A. and Tuller H. L. (2013) “Advances and new directions in gas-sensing devices”, Acta Mat., vol. 51, pp. 974 – 1000.
4. MH-Z19C CO2 Sensor Datasheet. [Електронний ресурс]. – [Режим доступу]: <https://www.winsen-sensor.com/d/files/MH-Z19C.pdf>
5. Литвин В. В., Кінащук М. М. Основи побудови мікропроцесорних систем. – К.: Видавництво НУБіП, 2021. – 168 с
6. .Varonti P., Pillai P., Chook V.C. Wireless sensor networks: a survey. – Computer Networks, 2007.
7. Kurniawan A. Internet of Things Projects with ESP32. – Apress, 2019. – 285 с.
8. Системи автоматизованого моніторингу навколишнього середовища: навчальний посібник / Під ред. І.Ю. Сидоренка. – К.: Політехніка, 2021. – 212 с.
9. PostgreSQL documentation. [Електронний ресурс]. – [Режим доступу]: <https://www.postgresql.org/docs/>
10. Seinfeld, J.H.; Pandis, S.N. Atmospheric Chemistry and Physics: From Air Pollution to Climate Change, 2nd ed.; Wiley: Hoboken, NJ, USA, 2012.
11. Antonio, P.; Grimaccia, F.; Mussetta, M. Architecture and methods for innovative heterogeneous wireless sensor network applications. Remote Sens. 2012, 4, 1146–1161

12. Протокол HTTP. RFC 2616. [Електронний ресурс]. – [Режим доступу]: <https://datatracker.ietf.org/doc/html/rfc2616>
13. Архітектура мікропроцесорних систем / М. І. Мельник. – Львів: Видавництво Львівської політехніки, 2019.
14. MQTT v3.1.1 Specification. [Електронний ресурс]. – [Режим доступу]: <http://mqtt.org/>
15. Javier Diz-Bugarín, José L. Rodríguez-Fernández, "Digital Radiocommunications Module for Learning Applications and Remote Sensing", 2024 XVI Congreso de Tecnología, Aprendizaje y Enseñanza de la Electrónica (TAEE), pp.1-7, 2024.
16. FLUXNET: a global network. Integrating worldwide CO2, water and energy flux measurements. Historical Site Status. Web Page from ORNL DAAC, Oak Ridge, Tennessee, U.S.A. 2013. Електронний доступ: <http://fluxnet.ornl.gov/site-status>
17. Pritpal Singh. Embedded System Design with ARM Cortex-M Microcontrollers. – Springer, 2020.
18. OpenWeather API. – [Режим доступу]: <https://openweathermap.org/api>
19. Node.js HTTP Module Documentation. – [Режим доступу]: <https://nodejs.org/api/http.html>