

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра спеціалізованих комп'ютерних систем

ПСУЙ ВОЛОДИМИР БОГДАНОВИЧ

Сенсорна мережа контролю радіаційних і хімічних
загроз із застосуванням LoRa та мобільних клієнтів/ Sensor
network for monitoring radiation and chemical threats using
LoRa and mobile clients

спеціальність 152 – Метрологія та інформаційно-вимірвальна техніка
освітньо професійна програма – Технології інтернету речей

Кваліфікаційна робота

Виконав студент групи
МТІР – 41 В.Б.Псуй

Науковий керівник :
к.т.н. доц. Б.О.Масляк

Кваліфікаційну роботу
допущено до захисту

«___» _____ 2026р.

Завідувач кафедри

_____ А.І.Сегін

Тернопіль - 2026

Факультет комп'ютерних інформаційних технологій
Кафедра спеціалізованих комп'ютерних систем
Освітній ступінь "бакалавр"

Спеціальність: 152 – Метрологія та інформаційно-вимірювальна
техніка

Освітньо-професійна програма – Технології інтернету речей

“ЗАТВЕРДЖУЮ”
Завідувач кафедри СКС

_____ А.І. Сегін
“ ” _____ 2025р.

ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Псуй Володимиру Богдановичу
(прізвище, ім'я по-батькові)

1. Тема випускної кваліфікаційної роботи

Сенсорна мережа контролю радіаційних і хімічних загроз із застосуванням LoRa та мобільних клієнтів / Sensor network for monitoring radiation and chemical threats using LoRa and mobile clients

Керівник роботи _____ к.т.н., доцент Б.О.Масляк
затверджені наказом по університету від «1» грудня 2025р. № 894

2. Строк подання студентом закінченої випускної кваліфікаційної роботи: 25 травня 2026р.

3. Вихідні дані до випускної кваліфікаційної роботи:

1. Технічні характеристики та принципи роботи сенсорів контролю радіації та хімічних загроз.
2. Особливості вимірювання рівню радіації та хімічного забруднення з використанням мікроконтролерів.
3. Характеристики сучасних мікроконтролерів.
4. Системи передачі інформації в складі територіально розподілених мереж .

4. Основні питання, які потрібно розробити:

1. Розглянути особливості контролю радіаційних та хімічних параметрів середовища.

2. Проектування компонентів пристрою їх вимірювання та контролю.

3. Програмно-апаратна реалізація багатоканального пристрою вимірювання та контролю радіаційних та хімічних параметрів на предмет загроз на основі використання мікроконтролерів.

4. Реалізувати процес передачі інформації (інтерфейс LoraWan) на сервер з метою запобігання забрудненням.

5. Перелік графічного матеріалу у роботі:

1. Структурна схема мікропроцесорного вузла вимірювання та контролю рівню радіації та хімічного забруднення.

2. Схема підключення первинних сенсорів.

4. Структура LoraWan мереж.

6. Консультанти розділів випускної кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Масляк Б.О.		
2	Масляк Б.О.		
3	Масляк Б.О.		

7. **Дата видачі завдання 28 листопада 2025 р.**

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назви етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Особливості побудови систем вимірювання та контролю радіаційних та хімічних параметрів	01.12.2025р. – 15.01.2026р.	
2	Обґрунтування вибору технічних засобів та мікропроцесорної техніки	16.01.2026р. – 15.03.2026р.	
3	Апаратно-програмна реалізація сенсорної мережі	16.03.2026р. – 30.04.2026р.	

Студент

(підпис)

Псуй В.Б.

Керівник роботи

(підпис)

к.т.н., доц. Масляк Б.О.

РЕФЕРАТ

В результаті виконання кваліфікаційної роботи на тему «Сенсорна мережа контролю радіаційних і хімічних загроз із застосуванням LoRa та мобільних клієнтів» проведено аналіз існуючих систем радіаційно-хімічного контролю та виявлено їх ключовий недолік - оцінка кожного фактора здійснюється незалежно, що не враховує синергетичний ефект — ситуацію, коли жоден із параметрів не перевищує граничнодопустимого рівня, але їх спільний вплив є небезпечним.

Об'єкт дослідження — процеси моніторингу радіаційних і хімічних загроз у розподілених середовищах із застосуванням сенсорних мереж.

Предмет дослідження — архітектура сенсорної мережі, апаратно-програмні засоби вузлів та методи оцінки комплексної небезпеки при одночасному впливі кількох факторів.

Мета роботи — розробка сенсорної мережі для моніторингу радіаційних і хімічних загроз на основі мультисенсорної обробки даних із передачею даних за технологією LoRa та відображенням інформації на мобільних клієнтах.

Розроблено архітектуру розподіленої мережі з автономних вузлів на базі ESP32 з радіомодулем LoRa SX1276 (868/915 МГц), шлюзу та хмарного сервера на основі брокера MQTT. Кожен вузол вимірює рівень іонізуючого випромінювання та концентрацію токсичних речовин, обчислює інтегральний індекс ризику з урахуванням синергетичного ефекту та передає результати до шлюзу. Мобільний клієнт відображає стан вузлів на інтерактивній карті та надсилає push-сповіщення при виявленні небезпечної комбінації факторів. Тестування підтвердило дальність зв'язку до 8 км у відкритому просторі, час відгуку клієнта до 3 с та автономність вузла понад 30 діб.

Розроблена система може застосовуватись для моніторингу промислових об'єктів підвищеної небезпеки, контролю зон забруднення та в інтересах служб цивільного захисту.

Ключові слова: СЕНСОРНА МЕРЕЖА, МОНІТОРИНГ, ХІМІЧНА ЗАГРОЗА, ESP32, МОБІЛЬНИЙ КЛІЄНТ.

ABSTRACT

As a result of the qualification work on the topic "Sensor network for monitoring radiation and chemical threats using LoRa and mobile clients", an analysis of existing radiation and chemical control systems was conducted and their key drawback was identified - the assessment of each factor is carried out independently, which does not take into account the synergistic effect - a situation when none of the parameters exceeds the maximum permissible level, but their joint impact is dangerous. The object of the study is the processes of monitoring radiation and chemical threats in distributed environments using sensor networks. The subject of the study is the architecture of the sensor network, hardware and software of the nodes and methods for assessing complex danger with the simultaneous impact of several factors. The purpose of the work is to develop a sensor network for monitoring radiation and chemical threats based on multi-sensor data processing with data transmission using LoRa technology and displaying information on mobile clients. A distributed network architecture has been developed from autonomous nodes based on ESP32 with a LoRa SX1276 radio module (868/915 MHz), a gateway and a cloud server based on an MQTT broker. Each node measures the level of ionizing radiation and the concentration of toxic substances, calculates the integral risk index taking into account the synergistic effect and transmits the results to the gateway. The mobile client displays the status of the nodes on an interactive map and sends push notifications when a dangerous combination of factors is detected. Testing confirmed a communication range of up to 8 km in open space, a client response time of up to 3 s and node autonomy of more than 30 days. The developed system can be used for monitoring high-risk industrial facilities, controlling pollution zones and in the interests of civil protection services.

Keywords: SENSOR NETWORK, MONITORING, CHEMICAL THREAT, ESP32, MOBILE CLIENT.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	7
Вступ	8
Розділ 1. Аналіз методів та засобів побудови систем екологічного моніторингу	12
1.1. Класифікація радіаційних і хімічних загроз	12
1.2. Принципи роботи сенсорів радіаційних та хімічних забруднень	15
1.3 Огляд систем бездротового моніторингу	19
1.4. Постановка задачі	23
Розділ 2. Проєктування та розробка апаратно-програмних компонентів мережі	25
2.1 Розробка структурної схеми мережі	25
2.2 Вибір та обґрунтування апаратної бази	28
2.3 Алгоритми збору та первинної обробки даних на автономних вузлах	34
Розділ 3. Програмна реалізація мережевої взаємодії та мобільного клієнта	40
3.1 Організація передачі даних за протоколом LoRa/LoRaWAN	40
3.2 Розробка програмного додатку вимірювання хімічних забруднень та радіації	42
3.3 Розробка мобільного додатка для візуалізації загроз	47
Висновки	54
Список використаних джерел	55
ДОДАТКИ	57

					ДП.МТІР.97022469.00.00.00.000 ПЗ			
		№ докум.						
Розроб.	Псуй В.Б.				Сенсорна мережа контролю радіаційних і хімічних загроз із застосуванням LoRa та мобільних клієнтів Sensor network for monitoring radiation and chemical threats using LoRa and mobile clients			Акру
Перевірів.	Масляк Б.О.							68
Н. Контр.	Масляк Б.О.					ЗУНУ.ФКІТ.МТ		
Затверд.	Сегін А.І.							

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

IoT Internet of Things — Інтернет речей

Range — технологія бездротової передачі даних на великі відстані

LoRaWAN Long Range Wide Area Network — мережа широкого охоплення на основі LoRa

Wi-Fi Wireless Fidelity — стандарт бездротової локальної мережі

MQTT Message Queuing Telemetry Transport — протокол обміну повідомленнями для IoT

MCU Microcontroller Unit — мікроконтролерний пристрій

PCB Printed Circuit Board — друкована плата

ADC Analog-to-Digital Converter — аналого-цифровий перетворювач

UART Universal Asynchronous Receiver-Transmitter — універсальний асинхронний приймач-передавач

SPI Serial Peripheral Interface — послідовний периферійний інтерфейс

I²C Inter-Integrated Circuit — міжмікросхемний інтерфейс

ПЗ Програмне забезпечення

АЦП Аналого-цифровий перетворювач

МК Мікроконтролер

ПК Персональний комп'ютер

ВСТУП

Актуальність теми. Стрімкий розвиток промислової інфраструктури, розширення мережі підприємств хімічної галузі, атомної енергетики, а також загострення безпекової ситуації в Україні спричиняють суттєве зростання ризиків виникнення хімічних, ядерних, біологічних та радіологічних (ХНБР) надзвичайних ситуацій. За даними Державної служби України з надзвичайних ситуацій, тільки впродовж 2022–2024 років кількість зареєстрованих хімічно небезпечних подій зросла на 34%, а кількість об'єктів, що потребують постійного радіаційного контролю, перевищила 3 500 по всій країні.

Традиційні стаціонарні системи радіаційно-хімічного моніторингу, побудовані на провідних технологіях або стільниковому зв'язку (GSM/LTE), мають критичні вади в умовах надзвичайних ситуацій: залежність від збереження кабельної мережі та базових станцій, висока вартість розгортання, значне енергоспоживання та відсутність масштабованості. Руйнування інфраструктури під час збройного конфлікту або техногенної катастрофи фактично паралізує такі системи саме тоді, коли потреба в даних найвища.

Перспективним рішенням є застосування технології LoRa (Long Range) у стандарті LoRaWAN — маломіцного безліцензійного радіозв'язку, що забезпечує дальність передачі від 2 до 15 км у відкритому просторі за надзвичайно низького споживання енергії (десятки мкА в режимі передачі). Інтеграція сенсорних вузлів на основі LoRa з мобільними клієнтськими застосунками дозволяє створити повністю автономну, децентралізовану та масштабовану систему контролю ХНБР-загроз, не залежну від стаціонарної інфраструктури.[2]

Таким чином, розробка сенсорної мережі контролю радіаційних і хімічних загроз із застосуванням LoRa та мобільних клієнтів є науково та практично актуальним завданням, що відповідає пріоритетам державної безпеки,

цивільного захисту та концепції Інтернету речей (IoT) у критичній інфраструктурі.

Зв'язок роботи з науковими програмами, планами, темами. Бакалаврська робота виконана відповідно до навчального плану підготовки здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю у рамках дисципліни «Інформаційно-вимірювальні технології» та узгоджується з науковим напрямом кафедри у галузі систем збору та обробки даних в умовах надзвичайних ситуацій.

Мета і завдання роботи. Метою бакалаврської роботи є розробка архітектури розподіленої сенсорної мережі для автономного контролю радіаційних і хімічних загроз із використанням технології LoRa/LoRaWAN та мобільних клієнтів, що забезпечує надійну передачу вимірювальних даних за умов відсутності стаціонарної телекомунікаційної інфраструктури.

Для досягнення поставленої мети визначено такі завдання:

- здійснити аналіз загроз ХНБР-характеру та діючої нормативно-правової бази щодо допустимих рівнів радіаційного і хімічного забруднення (НРБУ-97, ДР-2006, ДСАНПІН 8.8.1.2.3.4-000-2001);
- провести порівняльний аналіз сучасних бездротових технологій для IoT-систем (LoRa, Zigbee, NB-IoT, Wi-Fi) та обґрунтувати вибір LoRaWAN для ХНБР-моніторингу;
- розробити апаратну схему сенсорного вузла на основі мікроконтролера STM32/ESP32 з трансивером SX1276 та відповідними детекторами (лічильник Гейгера-Мюллера, електрохімічний та PID-сенсори);
- спроектувати протокол передачі даних, алгоритм порогової сигналізації та ескалації тривоги (рівні «увага» / «небезпека» / «евакуація»);
- розробити серверну частину системи (Chirpstack / TTN + MQTT-брокер + Node-RED) та мобільний клієнтський застосунок з відображенням даних на карті, графіками показань і Push-сповіщеннями.

Об'єкт дослідження — процеси збору, передачі та відображення вимірювальної інформації про рівні радіаційного та хімічного забруднення навколишнього середовища в умовах автономного функціонування телекомунікаційної інфраструктури.

Предмет дослідження — архітектура розподіленої сенсорної мережі на основі технології LoRaWAN з мобільними клієнтами, методи обробки та передачі вимірювальних даних від детекторів радіаційного та хімічного забруднення.

Методи дослідження. У роботі застосовано комплекс методів: системного аналізу — для дослідження предметної галузі та порівняння існуючих рішень; математичного моделювання — для розрахунку лінк-бюджету радіоканалу LoRa (модель поширення Okumura-Hata); методи цифрової обробки сигналів — для фільтрації та усереднення показань сенсорів; об'єктно-орієнтованого програмування — для реалізації прошивки вузла та мобільного застосунку; експериментальні методи — для стендового тестування точності вимірювань і верифікації протоколу передачі даних.

Наукова новизна одержаних результатів. В результаті виконання роботи:

- запропоновано інтегровану архітектуру моніторингу забруднень на основі LoRaWAN, що поєднує різнотипні детектори (іонізуючого випромінювання та токсичних хімічних речовин) в єдиній мережі з уніфікованим форматом пакету даних та триступеневою системою ескалації тривоги;
- удосконалено алгоритм адаптивного керування частотою опитування сенсорних вузлів, що дозволяє підвищити автономність системи живлення до 30% порівняно з фіксованим інтервалом опитування;
- набула подальшого розвитку методика розрахунку зони покриття одного LoRa-шлюзу в умовах забудованої місцевості з урахуванням рельєфу та типу перешкод для задач ХНБР-моніторингу.

Практичне значення одержаних результатів. Розроблена система може бути безпосередньо застосована для:

- автономного моніторингу радіаційного фону та концентрацій хімічних забруднень на промислових підприємствах (хімічні заводи, атомні електростанції, склади небезпечних речовин);
- розгортання мобільних пунктів контролю забруднення в зонах надзвичайних ситуацій і бойових дій, де стаціонарна телекомунікаційна інфраструктура зруйнована або недоступна;
- інтеграції в системи цивільного захисту та оперативного реагування ДСНС України як компонент ситуаційної обізнаності рятувальних підрозділів.

Результати роботи можуть бути використані в навчальному процесі при викладанні дисциплін «Інформаційно-вимірювальні системи», «Технології інтернету речей».

Структура та обсяг роботи. Бакалаврська робота складається зі вступу, трьох розділів, висновку, списку використаних джерел. Загальний обсяг роботи — _____ сторінок основного тексту, з яких _____ містять _____ рисунків та _____ таблиць. Список використаних джерел нараховує _____ найменувань.

1. АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ПОБУДОВИ СИСТЕМ ЕКОЛОГІЧНОГО МОНІТОРИНГУ

1.1. Класифікація радіаційних і хімічних загроз

Радіаційна та хімічна безпека є ключовими компонентами екологічного моніторингу. Перш ніж проектувати вимірювальну мережу, необхідно чітко визначити класи загроз, фізичні механізми їхнього впливу на живі організми та чинні нормативні вимоги. В цьому параграфі систематизуємо зазначені відомості про забруднення як основу для подальшого вибору сенсорів і алгоритмів обробки даних [2, 5]

Типи іонізуючих випромінювань. Іонізуюче випромінювання поділяють на корпускулярне (*alpha*-, *beta*-частинки, нейтрони) та електромагнітне (*gamma*-кванти, рентгенівське). Кожний тип характеризується проникною здатністю, питомою іонізацією та біологічним впливом.

Alpha-випромінювання (α) — потік ядер гелію. Пробіг у повітрі становить 3–10 см, у біологічній тканині — 0,04–0,08 мм. Незважаючи на мінімальну проникну здатність внутрішнє опромінення через інгаляцію або їжу є критичним сценарієм.

Beta-випромінювання (β) — потік електронів або позитронів з енергією 0,02–3,5 МеВ (мегаелектронвольт). Пробіг у тканині — до кількох сантиметрів. Зовнішній вплив небезпечний для шкіри та ока; захист — кілька міліметрів алюмінію або оргскла.

Gamma-випромінювання (γ) — електромагнітні кванти енергіями 0,01–10 МеВ. Має найбільшу проникну здатність: напівшар послаблення у воді — 10–30 см. Для захисту потрібні важкі матеріали — свинець, бетон.

Нейтрони не мають заряду та характеризуються великим пробігом. Джерела — ядерні реактори, критичні аварії, термоядерні реакції.

Ефективна доза опромінення розраховується за формулою:

$$E = \sum T(w_T * H_T), \quad H_T = \sum R(w_R * D_{\{T,R\}})$$

					ДП.МТІР.9702469.00.00.00.000 ПЗ	
		№				12

де D — поглинена доза;
 w_R — ваговий коефіцієнт випромінювання;
 w_T — ваговий коефіцієнт тканини;
 H_T — еквівалентна доза у тканині,.

Таблиця 1.1 — Порівняльна характеристика типів іонізуючого випромінювання

Тип	Частинка/квант	Діапазон енергій	Пробіг у тканині	ВБЕ	Зовн. небезпека
Alpha (α)	Ядро He-4	4–9 MeV	0,04–0,08 мм	20	Мала
Beta (β)	e^- / e^+	0,02–3,5 MeV	до кількох см	1	Помірна
Gamma (γ)	Фотон	0,01–10 MeV	десятки см	1	Висока
Нейтрони (n)	Нейтрон	0,001 eВ–20 MeV	десятки см	5–20	Висока

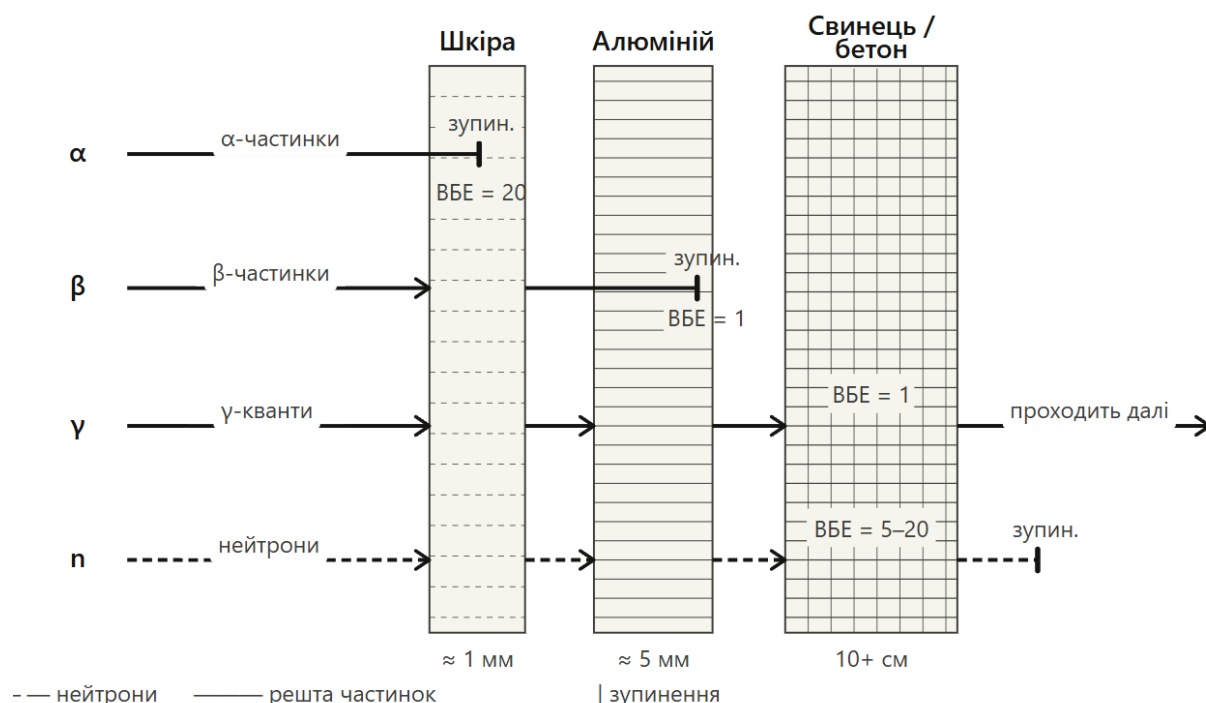


Рисунок 1.1 — Проникна здатність різних видів випромінювання

Класи аварійно небезпечних хімічних речовин (АХНР). Аварійно хімічно небезпечні речовини (АХНР) — токсичні сполуки, викид яких в атмосферу може

призводити до масових уражень. Відповідно до ДСТУ 3273-95 та ГН 2.1.6.1338-03 їх класифікують за характером токсичної дії та агрегатним станом таблиця 1.2.

Таблиця 1.2 — Основні класи АХНР і типові представники

Клас	Характер дії	Типові речовини	Першочергові сенсори
Задушливі	Набряк легень	Хлор, фосген	Електрохімічні
Загальноотруйні	Блокування дихання	HCN, CO, H ₂ S	Електрохімічні, PID
Нейротоксичні	Пригнічення ХЕ	Зарин, VX, ФОС	PID, імуносенсиори
Подразнюючі	Подразнення слизових	NH ₃ , SO ₂ , акролеїн	Електрохімічні
Шкірно-нарівні	Ушкодження шкіри/очей	Іприт, люїзит	PID, Surface Acoustic Wave
Радіоактивні аерозолі	Внутрішнє опромінення	Cs-137, I-131, Pu-239	Сцинтиляційні

В Україні основним нормативним документом у сфері радіаційної безпеки є «Допустимі рівні вмісту радіонуклідів Cs-137 та Sr-90 у продуктах харчування» (ДР-2006), затверджені МОЗ України. Паралельно діє НРБУ-97/Д-2000, що встановлює граничні дози опромінення населення.

Ключові показники НРБУ-97:

- ефективна доза опромінення населення — не більше 1 мЗв/рік (крім природного фону та медичного)
- еля персоналу категорії А — не більше 20 мЗв/рік
- еквівалентна доза для шкіри — не більше 500 мЗв/рік

Для хімічних забруднень в Україні діють гігієнічні нормативи ГДК (гранично допустимі концентрації) у повітрі робочої зони (ГОСТ 12.1.005-88, адаптований в Україні) та ГДК в атмосферному повітрі населених місць (ДСП 201-97). Відповідно до ДСП 201-97 для хлору ГДК_{мр} = 0,1 мг/м³, для аміаку — 0,2 мг/м³, для сірководню — 0,008 мг/м³ (максимально разова).

Комбінований вплив радіаційного та хімічного забруднення. Окремою науковою і практичною проблемою є оцінка комбінованого впливу радіаційного та хімічного чинників. Взаємодія між ними може мати адитивний, синергічний

або антагоністичний характер. Синергізм зафіксовано у випадку поєднання гамма-опромінення та важких металів (Pb, Cd): експерименти на лабораторних тваринах показують зростання частоти хромосомних аберацій (зміна структури або кількості хромосом у клітині) у 1,5–3. Нейротоксичні ефекти органофосфатів посилюються при фоновому хронічному бета-опроміненні внаслідок окисного стресу та порушень репарації ДНК.

Приклад розрахунку сумарного ризику за методом коефіцієнтів небезпеки (Hazard Index, HI) для хімічних речовин:

$$HI = \sum_i(HQ_i) = \sum_i(C_i / RfC_i)$$

де C_i — виміряна концентрація речовини i , мг/м³;

RfC_i — референтна концентрація (безпечний рівень), мг/м³.

При $HI > 1$ існує неприйнятний нехімічний ризик; при комбінуванні з радіаційними дозами $D > 0,1$ мЗв/рік ризик рекомендується множити на коефіцієнт синергізму $k_s \approx 1,2$ – $2,5$.

Для тваринного світу порогові концентрації часто є нижчими, ніж для людини: наприклад, для комах-запилювачів ГДК пестицидів з фосфорорганічною основою у 5–10 разів нижчі, ніж для людини. Тому системи моніторингу доцільно налаштовувати на найбільш чутливий вид у екосистемі.

1.2. Принципи роботи сенсорів радіаційних та хімічних забруднень

Вибір типу сенсора визначає точність, енергоспоживання та вартість всієї вимірювальної системи. У цьому параграфі розглянуто чотири ключові класи детекторів: лічильники Гейгера-Мюллера (ГМ), сцинтиляційні, електрохімічні та фотоіонізаційні (PID) сенсори.

Детектори Гейгера-Мюллера (рисунок 1.2) це газонаповнений детектор з двома електродами, між якими прикладена напруга 400–1200 В.

					ДП.МТІР.9702469.00.00.00.000 ПЗ	
		№	1			15



Рисунок 1.2 – Лічильник Гейгера-Мюллера

При проходженні іонізуючої частинки виникає лавинний розряд, що реєструється як імпульс. Принцип роботи: частинка іонізує атоми газу (зазвичай $\text{Ar} + 10\% \text{ ізобутану}$) $\rightarrow$ первинні електрони прискорюються $\rightarrow$ вторинна іонізація $\rightarrow$ лавина $\rightarrow$ імпульс струму $\approx 1\text{--}10 \text{ мА}$ тривалістю $\approx 100 \text{ мкс}$. Мертвий час ГМ-трубки:

$$\tau = 100 \dots 300 \text{ мкс},$$

що обмежує максимальну реєстровану активність. Виміряна активність $N_{\text{вим}}$ пов'язана з істинною $N_{\text{іст}}$ формулою корекції мертвого часу:

$$N_{\text{іст}} = N_{\text{вим}} / (1 - N_{\text{вим}} * \tau).$$

Чутливість ГМ-трубок до різних видів випромінювання неоднакова: для гамма- та бета-частинок — висока, для альфа-частинок — низька через товсту стінку (виключення — тонковіконні трубки типу LND 712). Типові параметри детектора Д-3615М: напруга плато 400–475 В, чутливість 1–3 імп/(мкР/год).

Сцинтиляційні детектори детектор перетворюють енергію іонізуючого випромінювання у фотони видимого світла, які далі реєструються фотопомножувачем або кремнієвим фотодіодом (SiPM). Процес: частинка або фотон $\rightarrow$ збудження молекул сцинтилятора $\rightarrow$ люмінесцентний спалах (спектр

300–550 нм) → ФЕП → електричний сигнал. Матеріали сцинтиляторів: NaI(Tl) — для гамма, BGO (Bi₄Ge₃O₁₂) — для гамма/нейтронів, LaBr₃(Ce) — для гамма з кращою енергетичною роздільною здатністю, органічні (антрацен, пластикові) — для бета й нейтронів. Енергетична роздільна здатність:

$$R = \Delta E / E = 2,35 * (F * N_{ph})^{(-1/2)}$$

де F — коефіцієнт Фано;

N_{ph} — кількість фотонів.

Перевагою сцинтиляційних детекторів є можливість ідентифікації радіонуклідів за гамма-спектром — це критично важливо для розрізнення природних (Ra-226, K-40) і техногенних (Cs-137, Co-60) джерел.

Електрохімічні сенсори газів [3, 4]. Електрохімічний сенсор (ECS) перетворює концентрацію газу на електричний струм через реакцію на робочому електроді. Конструкція: три електроди (робочий — WE, протиелектрод — CE, референтний — RE), розділені протонообмінною мембраною, залитою електролітом. Сигнал (струм) пропорційний концентрації газу (закон Фарадея–Баттлера–Фолмера):

$$I = n * F * A * k_m * C$$

де n — кількість електронів у реакції;

$F = 96\,485$ Кл/моль (стала Фарадея);

A — площа електрода;

k_m — коефіцієнт масопереносу;

C — концентрація газу.

Типові параметри: діапазон вимірювання Cl₂ 0–20 ppm, роздільна здатність 0,01 ppm, час відгуку T₉₀ < 30 с (сенсор 4-CKL від Alphasense). Основний недолік — деградація електроліту та перехресна чутливість до сторонніх газів.

Фотоіонізаційні детектори (PID). Фотоіонізаційний детектор (PID) використовує ультрафіолетову лампу (енергія фотонів 10,6 або 11,7 еВ) для іонізації органічних газів і парів, після чого вимірює іонний струм. Принцип дії:

органічна молекула AX поглинає фотон з енергією $h\nu \rightarrow AX + h\nu \rightarrow AX^+ + e^-$.

Струм між електродами:

$$I = Q * e * \mu * E / d$$

де Q — кількість іонізованих молекул;

e — заряд електрона; μ — рухливість;

E — напруженість поля;

d — відстань між електродами.

PID придатний для виявлення ароматичних сполук, ФОС (зарин: ІЕ = 10,1 еВ), альдегідів, меркаптанів. Нижня межа виявлення — 1 ppb для бензолу. Результати порівняння сенсорів для системи моніторингу приведені в таблиці 1.3.

Таблиця 1.3 — Порівняння сенсорів для системи моніторингу

Параметр	ГМ-трубка	Сцинтилятор NaI	ECS	PID
Чутливість	1–5 мкЗв/год	0,01 мкЗв/год	0,01 ppm	1 ppb
Цільовий аналіт	γ , β	γ , ідентифікація	Cl ₂ , CO, H ₂ S	ФОС, VOC, аромати
Живлення, В	5 (DC-DC до 500 В)	5 + ВН	3,3–5	5
Струм спокою, мА	5–20	50–200	2–5	20–50
T ₅₀ відгуку	Миттєво (імп.)	Миттєво	<30 с	<5 с
Вартість (USD)	5–25	150–500	20–60	100–300
Строк служби	5–10 років	10+ років	1–3 роки	2–4 роки

Таким чином, в даному розділі розглянуто основні типи сенсорів радіаційного та хімічного забруднень.

1.3. Огляд систем бездротового моніторингу та аналіз аналогів

Вибір технології бездротового зв'язку визначає енергетичний бюджет, дальність, пропускну здатність і вартість інфраструктури моніторингової мережі. У підрозділі порівнюються технології Zigbee, Wi-Fi, NB-IoT та LoRa, аналізуються існуючі комерційні системи, а також обґрунтовується вибір LoRa/LoRaWAN [9-11]. В таблиці 1.4 приведені основні характеристики існуючих технологій створення моніторингових мереж.

Таблиця 1.4 — Порівняння стандартів бездротового зв'язку для IoT-моніторингу

Параметр	Zigbee (IEEE 802.15.4)	Wi-Fi (IEEE 802.11n)	NB-IoT (3GPP R13)	LoRa/LoRaWAN
Частота, ГГц	2,4 / 0,868	2,4 / 5	0,7–0,9 (LTE)	0,433 / 0,868 / 0,915
Дальність, м	10–100 (mesh)	50–200	до 15 000	до 15 000 (під.міст.)
Швидкість, кбіт/с	250	до 150 000	до 250	0,3–50
Потужність TX	1–100 мВт	до 100 мВт	23 дБм	до 25 мВт (+14 дБм)
Струм сну, мкА	5–10	200–500	3–5	1–2
Вартість модуля	3–15 USD	2–10 USD	10–25 USD	5–20 USD
Автономність, доба	30–180	1–5	365+	365+

Zigbee добре підходить для мереж з mesh-топологією в закритих приміщеннях (підприємства, бункери), однак мала дальність і необхідність ретрансляторів ускладнюють розгортання у відкритому полі. Wi-Fi забезпечує велику пропускну здатність, але критичне енергоспоживання унеможливорює тривалу автономну роботу від батареї. NB-IoT вирізняється найвищою проникністю сигналу, але потребує оплаченого LTE-покриття, що в поставарійних зонах може бути відсутнім.

Технологія LoRa та протокол LoRaWAN. LoRa (Long Range) — модуляція з розширеним спектром на основі Chirp Spread Spectrum (CSS), розроблена компанією Cycleo (придбана Semtech у 2012 р.) та стандартизована через LoRa Alliance. Фізичний рівень. Модуляція CSS генерує chirp-сигнал, частота якого лінійно змінюється в межах смуги BW протягом символного інтервалу T_s . Параметри, що визначають компроміс між дальністю, швидкістю та споживанням:

- коефіцієнт розширення спектру (SF, Spreading Factor) — SF7 до SF12; збільшення SF на 1 подвоює час на символ і дальність, але вдвічі зменшує швидкість.
- смуга пропускання BW — 125, 250 або 500 кГц; ширша BW → коротший символ → вища швидкість, але менша чутливість.
- кодова відстань CR (Code Rate) — 4/5, 4/6, 4/7, 4/8; вища CR → краща корекція помилок, але менша ефективна швидкість.

Тривалість символу та бітова швидкість:

$$T_s = 2^{SF} / BW$$

$$R_b = SF * CR * BW / 2^{SF} \text{ [біт/с]}$$

Чутливість приймача визначається рівнем шуму прийому та мінімально необхідним відношенням сигнал/шум (SNR):

$$S_{min} = -174 + 10 * \log_{10}(BW) + NF + SNR_{min} \text{ [дБм]}$$

де NF — коефіцієнт шуму підсилювача (5–7 дБ);

SNR_{min} для SF12 ≈ -20 дБ.

Протокол LoRaWAN визначає мережевий рівень: топологію «зірка», три класи пристроїв (A, B, C), шифрування AES-128, адресацію DevEUI/AppKey. Клас A (найнижче споживання): два вікна прийому (RX1, RX2) після кожної передачі.

На ринку існує ряд спеціалізованих систем екологічного радіаційно-хімічного моніторингу. Серед них можна виділити *Intellicheck CBRN Monitoring System* (Інтеллічек, США) — стаціонарна мережа детекторів для аеропортів і

портів, що виявляє CBRN-загрози (хімічні, біологічні, радіаційні, ядерні). Використовує провідний зв'язок Ethernet/WiFi і 4G-резервування. Недоліки: висока вартість (>50 000 USD за вузол), відсутність автономності, залежність від інфраструктури.

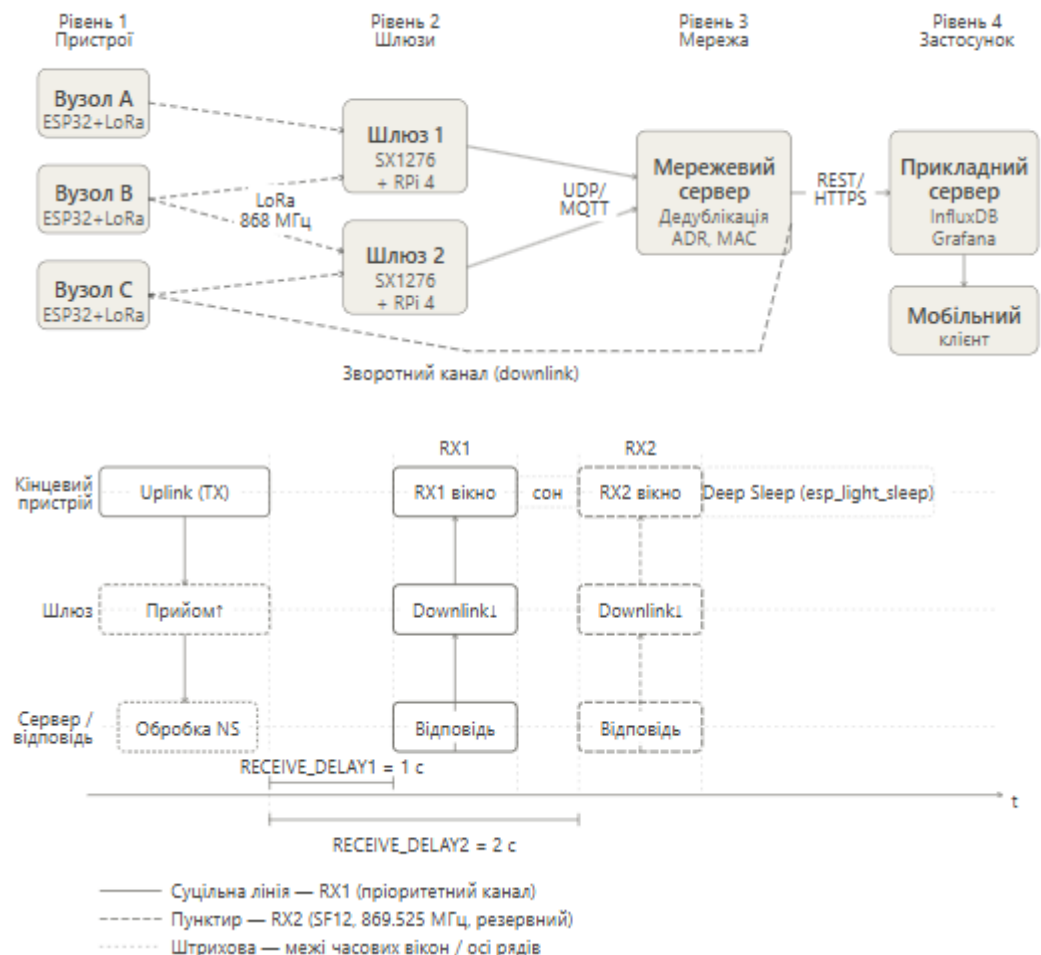


Рисунок 1.4 — Архітектура LoRaWAN та часова діаграма Класу А

SafePort Radiation Monitoring Network (Ludlum Measurements, США) — розподілена мережа гамма-детекторів на базі NaI-сцинтиляторів з RS-485 або Ethernet. Програмний пакет дозволяє real-time картографування доз. Обмеження: немає хімічного моніторингу, потрібне живлення від мережі.

ARGOS Environmental Radiation Monitoring (Rodas Group, Данія) — комплексна система моніторингу для НПС (атомних станцій) з підтримкою метеоданих і прогнозуванням поширення хмари. Використовує GPRS/4G та

дротові канали. Недоліки: висока вартість впровадження, складність розгортання в польових умовах.

Таблиця 1.5 — Порівняльний аналіз систем-аналогів

Параметр	Intellicheck	SafePort	ARGOS	Пропонована система
Радіац. моніторинг	Так	Так (NaI)	Так (NaI, ГМ)	Так (ГМ + сцинт.)
Хім. моніторинг	Так (CBRN)	Ні	Ні	Так (ECS + PID)
Зв'язок	Ethernet/WiFi/4G	RS-485/Ethernet	GPRS/4G	LoRaWAN
Автономність	Ні (мережа)	Ні	Ні	Так (сонячна)
Дальність зв'язку	<100 м WiFi	Провід	GPRS-зона	До 15 км
Мобільний клієнт	Обмежено	Ні	Веб-браузер	Android/iOS

Аналіз технологій та аналогів показує, що для автономних вузлів у відкритих зонах (зони відчуження, промислові майданчики, сільськогосподарські угіддя) LoRa/LoRaWAN є оптимальним рішенням з наступних причин:

- дальність до 15 км у сільській місцевості та до 5 км у міській — забезпечує покриття великих ділянок одним шлюзом (gateway);
- струм у режимі сну 1–2 мкА, у режимі передачі 30–120 мА протягом 50–200 мс: типовий середній струм при передачі раз на 15 хв — 0,05–0,5 мА, що відповідає рокам автономної роботи від Li-SOCl₂ або сонячної панелі;
- робота у вільній ISM-смузі (868 МГц в Україні) без щомісячної оплати за SIM-карти;
- вбудоване шифрування AES-128 на рівні мережі та додатку;
- бюджет лінії зв'язку: при потужності TX = +14 дБм (25 мВт) і чутливості -137 дБм: Link Budget = 151 дБ — на 20–30 дБ більше, ніж у WiFi або Zigbee.

1.4. Постановка задачі

Аналіз існуючих рішень та виявлені недоліки у параграфах 1.1–1.3 показав, що:

- наявні комерційні системи (Intellicheck, SafePort, ARGOS) не поєднують радіаційний та хімічний моніторинг в одному автономному вузлі.
- системи на базі WiFi та Ethernet вимагають постійного електроживлення та наявної інфраструктури, яка може бути зруйнована під час надзвичайної ситуації.
- жодна з проаналізованих систем не забезпечує повноцінний мобільний клієнт з відображенням карти та сповіщеннями у реальному часі.
- вартість готових рішень (5 000–50 000 USD за вузол) унеможлиблює щільне покриття великих площ.

Метою бакалаврської роботи є розроблення прототипу сенсорної мережі для моніторингу радіаційних і хімічних загроз на основі технології LoRa з мобільними клієнтами, яка відповідає таким вимогам: автономність, широка зона покриття, низька вартість, відповідність нормативам НРБУ-97 та ДР-2006.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- визначити перелік вимірюваних параметрів: потужність дози гамма-випромінювання (мкЗв/год), об'ємна активність радіонуклідів (Бк/м³), концентрації Cl₂, CO, H₂S, NH₃ (ppm), VOC (ppb);
- розробити апаратну платформу сенсорного вузла на базі мікроконтролера з підтримкою LoRa-трансивера (SX1276/SX1278), сенсорного масиву та системи живлення з сонячного елемента;
- розробити прошивку вузла (алгоритм та скетч): алгоритм зчитування сенсорів, пакетизації та передачі даних LoRaWAN, режим сну для мінімізації споживання.
- розгорнути LoRaWAN-шлюз та серверну частину (The Things Network або власний сервер ChirpStack) для збору та зберігання вимірювань.

– розробити мобільний застосунок для push-сповіщень при перевищенні порогів.

Технічні вимоги та критерії успішності розроблюваної системи приведені у таблиці 1.6.

Таблиця 1.6 — Технічні вимоги до системи

Параметр	Вимога	Підстава
Дальність зв'язку	> 3 км у міській, > 10 км у польовій зоні	Аналіз LoRa, розд. 1.3
Автономність вузла	> 6 місяців при передачі 1 раз/15 хв	Польові умови
Поріг гамма	0,3 мкЗв/год (0,3 * природний фон)	НРБУ-97
Поріг С _{l2}	0,5 ppm (5 x ГДК _{рз})	ГН 2.1.6.1338
Поріг СО	50 ppm (NIOSH IDLH / 100)	NIOSH
Затримка сповіщення	< 30 с від перевищення порогу	Оперативність реагування
Температурний діапазон	-30 ... +60 °С	Кліматичні умови України
Вартість вузла	< 200 USD	Масштабованість

Таким чином у розділі 1 систематизовано клас загроз, що підлягають моніторингу: чотири типи іонізуючого випромінювання та шість класів АХНР; показано, що їх комбінований вплив може синергічно підвищувати ризик для живих організмів при значеннях, що відповідають допустимим нормам окремо взятого фактора. Розглянуто фізичні принципи роботи сенсорів. Огляд технологій бездротового зв'язку та аналогів підтвердив переваги LoRa/LoRaWAN для побудови автономних польових вузлів. На основі виявлених недоліків існуючих рішень сформовано мету, конкретні задачі і вимоги до системи, що розробляється.

2. ПРОЄКТУВАННЯ ТА РОЗРОБКА АПАРАТНО-ПРОГРАМНИХ КОМПОНЕНТІВ МЕРЕЖІ

2.1 Розробка структурної схеми мережі

Побудова розподіленої вимірювальної мережі розпочинається з визначення її загальної архітектури — способу, яким окремі пристрої організовані в єдину мережу, обмінюються даними і взаємодіють між собою. Для системи контролю радіаційних і хімічних загроз обрано чотирирівневу ієрархічну архітектуру, що відповідає сучасній концепції Internet of Things і дозволяє масштабувати мережу від кількох вузлів до кількох сотень без жодної зміни у базовому підході.

Перший, найнижчий рівень складають автономні сенсорні вузли, розміщені безпосередньо у зоні моніторингу. Кожен вузол самостійно вимірює фізичні величини, виконує первинну цифрову обробку отриманих відліків і передає стиснений пакет даних у бездротовий ефір. Другий рівень — шлюз (gateway): спеціалізований пристрій, який одночасно приймає повідомлення від усіх вузлів у радіусі покриття і перенаправляє їх на сервер через провідне з'єднання Ethernet або мережу мобільного зв'язку. Третій рівень — серверна платформа, де дані накопичуються у часовій базі даних, аналізуються і порівнюються із встановленими пороговими значеннями. Четвертий, найвищий рівень — мобільний застосунок, через який черговий оператор у режимі реального часу спостерігає стан усіх точок контролю, отримує сигнали тривоги і може переглядати архів вимірювань за будь-який минулий період.

Така ієрархія дозволяє ізолювати функції кожного рівня: відмова одного вузла не впливає на роботу решти мережі, шлюз може тимчасово накопичувати дані у внутрішньому буфері в разі перерви з'єднання з сервером, а сервер залишається незалежним від кількості підключених мобільних клієнтів. Загальна структурна схема системи показана на рисунку 2.1.

Вибір переліку небезпечних факторів, що підлягають моніторингу,

					ДП.МТІР.9702469.00.00.00.000 ПЗ	
		№	1			25

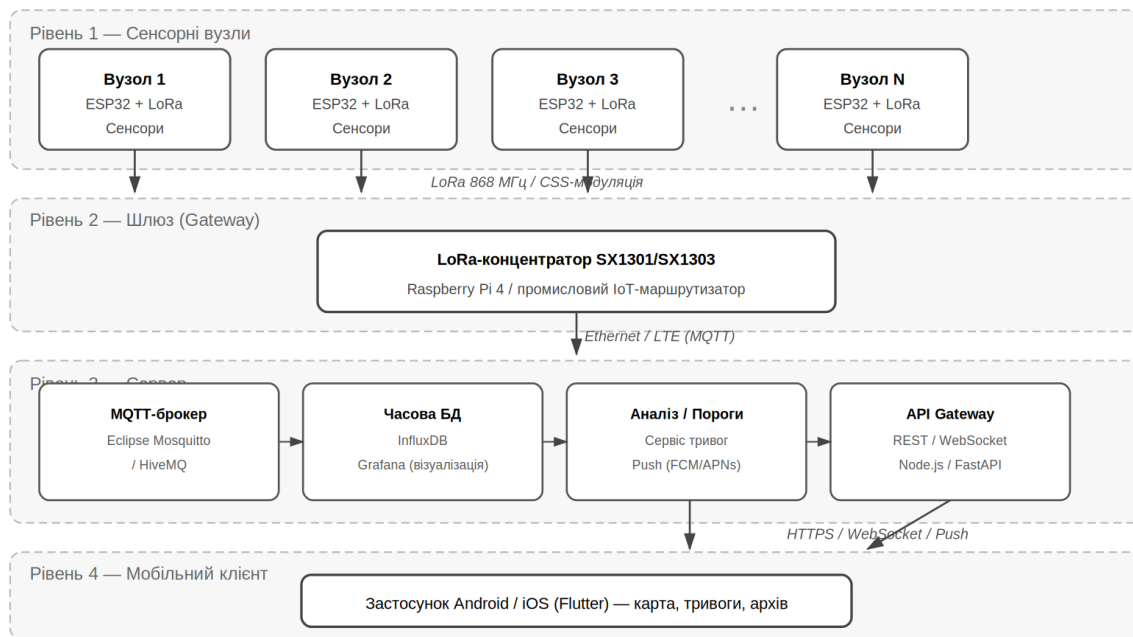


Рисунок 2.1 — Чотирирівнева структурна схема системи моніторингу

спирається на аналіз нормативних документів: вітчизняних Норм радіаційної безпеки НРБУ-97, Директиви ЄС 2013/59/Euratom з радіаційного захисту, стандартів ASHRAE 62.1 щодо якості повітря у приміщеннях, гранично допустимих концентрацій та порогів. Додатково враховано практику реальних техногенних аварій — Чорнобиль 1986 р., Фукусіма 2011 р., численні вибухи на хімічних підприємствах, — у яких найчастіше фіксувалося поєднання радіаційного забруднення з викидом токсичних газів.

Гамма-випромінювання є основним чинником зовнішнього радіаційного опромінення при ядерних аваріях і терористичних інцидентах. Монооксид вуглецю (CO) виникає при горінні, вибухах та неповному згорянні будь-якого органічного палива і є найпоширенішою причиною отруєнь у надзвичайних ситуаціях. Сірководень (H₂S) характерний для нафтопереробної та хімічної промисловості і є надзвичайно небезпечним вже при концентраціях порядку 1 ppm за тривалого впливу. Аміак (NH₃) вивільняється при аваріях на сільськогосподарських, харчових та хімічних підприємствах. Загальна концентрація летких органічних сполук (TVOC) слугує інтегральним

показником хімічного забруднення повітря при промислових аваріях. Діоксид вуглецю (CO₂) відображає загальний стан вентиляції і може бути індикатором пожежі або теракту у замкнених приміщення.

Таблиця 2.1 узагальнює повний перелік контрольованих факторів із зазначенням відповідних сенсорів та нормативних граничних значень. Обраний перелік охоплює найбільш ймовірні сценарії надзвичайних ситуацій в умовах України і при цьому є реалізованим на доступній апаратній базі без залучення дорогого лабораторного обладнання.

Таблиця 2.1 — Перелік контрольованих загроз та відповідних сенсорів

Загроза	Сенсор	Граничне значення	Стандарт
Гамма-випромінювання	ГМ-трубка SBM-20 / J305	0,3 мкЗв/год	НРБУ-97
Монооксид вуглецю (CO)	MQ-7 / MICS-4514	20 ppm (8 год)	ДСТУ EN 50271
Діоксид вуглецю (CO ₂)	MH-Z19B (NDIR)	1000 ppm	ASHRAE 62.1
Леткі орг. сполуки (TVOC)	SGP30 / CCS811	0,5 мг/м ³	WHO 2010
Аміак (NH ₃)	MICS-6814 / MQ-137	25 ppm (8 год)	ACGIH TLV
Сірководень (H ₂ S)	MICS-6814 / MQ-136	1 ppm (стелева)	OSHA PEL

Автономний сенсорний вузол є ключовим елементом мережі — саме він виконує всю роботу з безпосереднього вимірювання небезпечних факторів та прийняття первинних рішень. Вузол складається з мікроконтролера ESP32 як центрального процесора, модуля бездротового зв'язку LoRa, набору сенсорів і блоку живлення на базі літій-полімерного акумулятора із сонячною панеллю. Всі компоненти розміщуються у захисному корпусі класу IP67, стійкому до пилу і тривалого занурення у воду. Рисунок 2.2 наочно ілюструє внутрішню структуру вузла та зв'язки між його блоками.

Мікроконтролер керує всіма блоками вузла: ініціює вимірювання, зчитує дані з кожного сенсора через відповідний інтерфейс, виконує фільтрацію та агрегування результатів і відправляє готовий пакет через модуль LoRa.

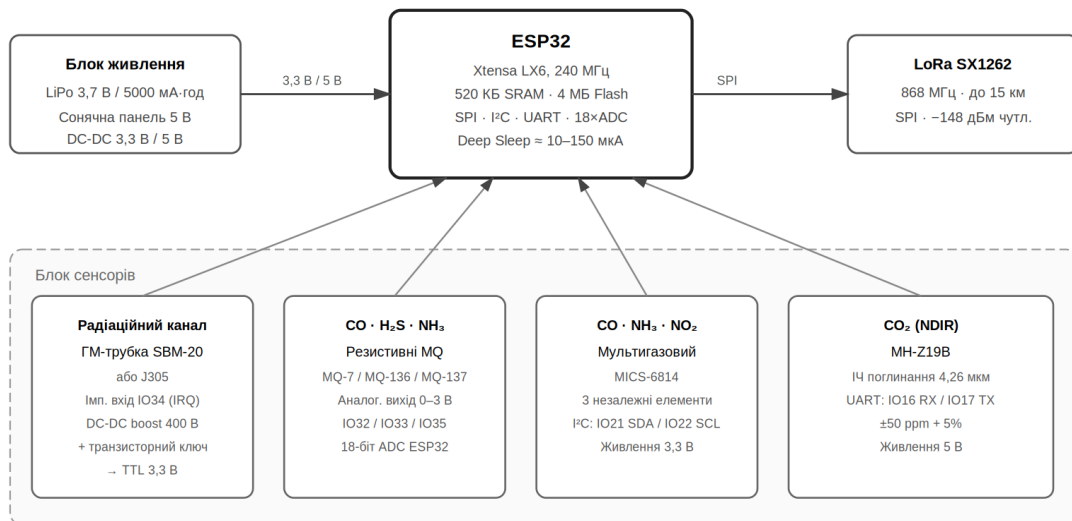


Рисунок 2.2 — Структурна схема автономного сенсорного вузла

В інтервалах між вимірюваннями ESP32 переходить у режим глибокого сну, що знижує споживання до 10–150 мкА і дозволяє забезпечити автономну роботу від акумулятора ємністю 5000 мА·год протягом 6–12 місяців залежно від частоти опитування.

3.2 Вибір та обґрунтування апаратної бази

Вибір конкретних електронних компонентів здійснювався за чотирма критеріями: відповідність технічним вимогам системи, мінімальне енергоспоживання в режимі очікування, доступність і вартість на ринку України, а також наявність відкритих бібліотек і документації для розробки програмного забезпечення. Нижче послідовно розглянуто кожену групу компонентів.

Для керування вузлом обрано мікроконтролер ESP32 виробництва компанії Espressif Systems. Він побудований на двоядерному процесорі Xtensa LX6 з тактовою частотою до 240 МГц і містить 520 КБ оперативної пам'яті SRAM та підтримку зовнішнього Flash-сховища. Периферія мікроконтролера є

виключно різноманітною: 34 програмовані виводи GPIO, 18-канальний АЦП з роздільною здатністю 12 бітів, три незалежні апаратні канали SPI, два канали I²C, три канали UART, а також вбудований Wi-Fi стандарту 802.11 b/g/n і Bluetooth 4.2/BLE. Ключовою перевагою ESP32 для даного застосування є гнучка система керування живленням. Мікроконтролер підтримує кілька режимів зниженого споживання: Light Sleep із споживанням близько 0,8 мА; Deep Sleep від 10 до 150 мкА залежно від активних джерел пробудження; та режим ULP-сопроцесора з потужністю лише 150 мкВт, у якому можна виконувати найпростіші завдання без увімкнення основних ядер. Вбудований Wi-Fi використовується у системі як резервний канал зв'язку при наявності точки доступу поблизу вузла та як засіб оновлення програмного забезпечення за технологією OTA (Over-the-Air update) без фізичного доступу до пристрою.

Порівняно з популярними альтернативами — STM32L476 та nRF52840 — ESP32 поступається за абсолютним мінімумом струму сну (STM32L476 споживає 1–5 мкА у режимі Standby), однак суттєво перевершує обидва за обчислювальною потужністю, вартістю (3–5 USD за модуль WROOM-32) і доступністю готових бібліотек у екосистемах Arduino та ESP-IDF. Оскільки при 60-секундному циклі вимірювань різниця між 150 мкА (ESP32 Deep Sleep) та 5 мкА (STM32 Standby) за добу складає лише близько 8 мА·год, що є прийнятним для акумулятора ємністю 5000 мА·год, перевага STM32 за енергоефективністю не компенсує значно більших витрат часу на розробку.

Технологія LoRa (Long Range) ґрунтується на модуляції Chirp Spread Spectrum (CSS), розробленій компанією Semtech [9-12]. Суть методу — у розподілі сигналу по широкій смузі частот за допомогою лінійно частотно-модульованих чирп-імпульсів, що робить прийом стійким до завад і дозволяє виявляти сигнал навіть на рівні нижче шуму — до -20 дБ відносно шумового підлогу. Завдяки цьому LoRa забезпечує дальність зв'язку до 15 км у відкритій місцевості та до 2 км у міській забудові навіть за потужності передавача 20–22 дБм. Для вузлів мережі розглядалися два чіпи Semtech: SX1276 і більш сучасний

SX1262. Обидва підтримують коефіцієнти розкиду від SF6 до SF12 та смуги пропускання від 7,8 до 500 кГц. Основні відмінності між ними наведені у таблиці 2.2.

Таблиця 2.2 — Порівняльні характеристики модулів LoRa

Параметр	SX1276	SX1262
Діапазон частот	137–1020 МГц	150–960 МГц
Макс. потужність передавача	+20 дБм	+22 дБм
Чутливість прийому (SF12)	–148 дБм	–148 дБм
Струм передачі (макс.)	100 мА	87 мА
Струм прийому	10,8 мА	4,2 мА
Струм режиму сну	0,2 мкА	0,9 мкА
Інтерфейс MCU	SPI	SPI

SX1262 має дещо вищу максимальну потужність передавача і, що важливіше, більш ніж удвічі менший струм у режимі прийому (4,2 мА проти 10,8 мА для SX1276), що при частому прийомі підтверджень суттєво зменшує добовий баланс енергоспоживання. Для сенсорних вузлів обрано SX1262 як більш енергоефективне рішення. Для шлюзу використовується концентратор SX1301 або SX1303, здатний одночасно приймати повідомлення на восьми незалежних каналах LoRa і обслуговувати до 1000 вузлів в одній комірці покриття.

Обмін між вузлами і шлюзом відбувається за протоколом LoRaWAN 1.0.3 організації LoRa Alliance. Вузли працюють у класі А — найбільш енергоефективному: після кожного акту передачі вузол відкриває два короткі вікна прийому для підтвердження або команд від мережі, а решту часу перебуває у режимі глибокого сну. У діапазоні 868 МГц норма Duty Cycle становить 1%, що обмежує тривалість передачі до 36 секунд на годину; пакет розміром 18 байт

при SF10 і смузі 125 кГц займає ефір лише близько 330 мс, залишаючи достатній ресурс для мережі з сотнями вузлів.

Для вимірювання гамма-випромінювання використовуються лічильники Гейгера-Мюллера (ГМ) — найбільш відпрацьований і доступний інструмент іонізаційної дозиметрії, надійність якого підтверджена десятиліттями промислового та наукового застосування. Принцип роботи ґрунтується на лавинній іонізації газу у розрядному проміжку між анодом і катодом трубки: кожна часточка іонізуючого випромінювання, що потрапляє до робочого об'єму, ініціює короткий лавинний розряд, котрий реєструється електронною схемою як один імпульс. Кількість таких імпульсів за хвилину — CPM (Counts Per Minute) — є прямою мірою інтенсивності опромінення.

Трубка SBM-20 (рисунок 1.2) є класичним рішенням, що широко використовується у сучасних дозиметрах завдяки низькій вартості та відпрацьованій технологічній надійності. Вона чутлива до гамма-випромінювання в енергетичному діапазоні 70–1200 кеВ і до бета-частинок. Робоча напруга становить 400 В, мертвий час — 90 мкс. Конверсійний коефіцієнт для перетворення CPM у мкЗв/год дорівнює 153,8 CPM/(мкЗв/год): природний радіаційний фон близько 0,12 мкЗв/год відповідає приблизно 18–20 імпульсам на хвилину. Сучасний аналог J305 має нижчий поріг чутливості — 0,05 мкЗв/год — і конверсійний коефіцієнт 123,1 CPM/(мкЗв/год), що дещо підвищує точність вимірювань при малих рівнях забруднення.

Підключення ГМ-трубки до ESP32 потребує додаткової схемотехніки. По-перше, необхідний підвищувальний DC-DC перетворювач, що формує стабільну напругу 400 В із напруги живлення 3,3–5 В. По-друге, кожен імпульс високої напруги необхідно перетворити на TTL-сигнал рівнем 3,3 В, придатний для входу GPIO мікроконтролера — для цього використовується транзисторний ключ або компаратор. Сформований таким чином імпульс подається на вхід Ю34 мікроконтролера, налаштований на зовнішнє переривання: кожне переривання

збільшує програмний лічильник на одиницю, і за заданий інтервал накопичення (30–60 секунд) формується значення СРМ.

Сенсори серії MQ (MQ-7, MQ-135, MQ-136, MQ-137) – рисунок 2.3, є резистивними напівпровідниковими перетворювачами на основі оксидів металів, переважно діоксиду олова SnO_2 .



Рисунок 2.3 - Модуль серії MQ

Механізм роботи такий: за відсутності цільового газу поверхня матеріалу адсорбує молекули кисню, що підвищує потенційний бар'єр і знижує електропровідність. При контакті з молекулами газу-відновника (CO , H_2S , NH_3 , різні вуглеводні) вони реагують з адсорбованим киснем, знижуючи потенційний бар'єр і збільшуючи провідність — відповідно, опір чутливого елемента зменшується. Ця зміна опору реєструється мікроконтролером через аналого-цифровий перетворювач. Залежність концентрації газу від відношення опорів описується степеневою функцією :

$$C = a \cdot (R_s / R_0)^b,$$

де C — концентрація газу, ppm;

R_s — поточний опір чутливого елемента, Ом;

R_0 — опір у чистому повітрі, Ом;

a і b — константи, що визначаються з технічної документації окремо для кожного газу і типу сенсора.

Перед початком вимірювань сенсор потребує прогріву нагрівального елемента протягом щонайменше 24–48 годин та первинного калібрування в умовах відомої концентрації еталонного газу або у чистому повітрі.

MICS-6814 є мультигазовим сенсором компанії SGX Sensortech, що виявляє одночасно CO у діапазоні 10–500 ppm, NH₃ у діапазоні 1–500 ppm та NO₂ у діапазоні 0,05–10 ppm завдяки трьом незалежним чутливим елементам, інтегрованим на одному чіпі. Сенсор підключається до ESP32 через шину I²C, що спрощує монтаж і зменшує кількість задіяних виводів GPIO порівняно з трьома окремими MQ-сенсорами.

Для вимірювання CO₂ обрано недисперсійний інфрачервоний сенсор MH-Z19B. Його принцип роботи кардинально відрізняється від резистивних сенсорів: молекули CO₂ поглинають інфрачервоне випромінювання на довжині хвилі 4,26 мкм, і за ступенем цього поглинання визначається концентрація газу. Завдяки цьому MH-Z19B не реагує на зміну температури та вологості так різко, як MQ-сенсори, має вбудовану температурну компенсацію і не потребує первинного калібрування за R₀. Точність вимірювань становить ± 50 ppm плюс 5% від показника. Сенсор передає дані до ESP32 через апаратний UART у текстовому форматі, що унеможливорює помилки при перетворенні аналогового сигналу.

На рисунку 2.4 наведена зведена принципова схема підключення всіх компонентів до мікроконтролера ESP32.

Розподіл інтерфейсів спроектовано з урахуванням апаратних обмежень конкретної мікросхеми: виводи IO34 та IO35 є виключно вхідними і використані для аналогових та імпульсних сигналів; виводи IO16 і IO17 виділені під апаратний UART2 для MH-Z19B; шина SPI з виводами IO5, IO18, IO19, IO23 з'єднує модуль LoRa; шина I²C з виводами IO21 і IO22 обслуговує MICS-6814. Такий розподіл повністю виключає конфлікти апаратних ресурсів і дозволяє одночасно обслуговувати всі периферійні пристрої.

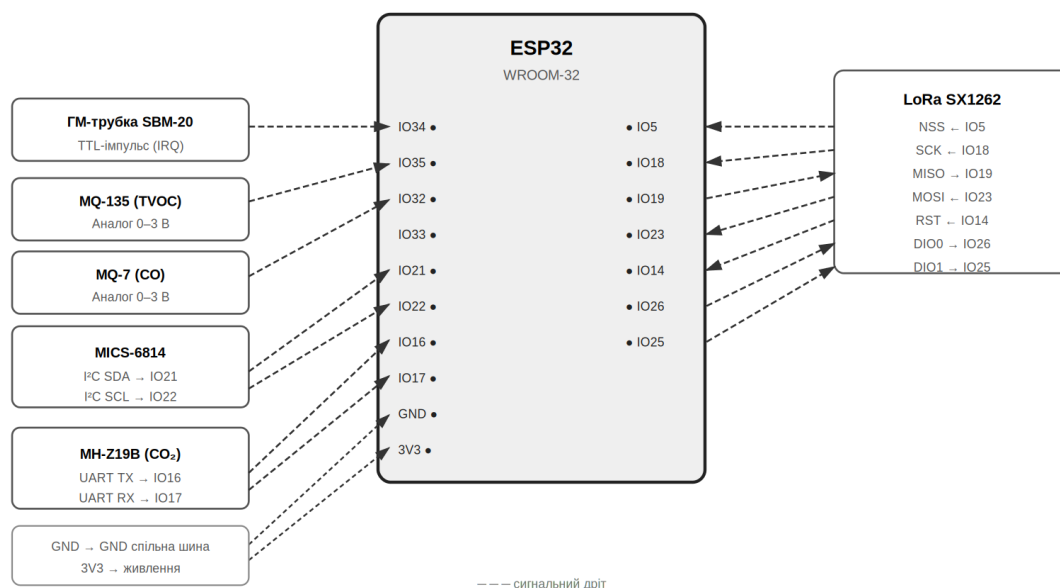


Рисунок 2.4 — Зведена схема підключення сенсорів і модуля LoRa до ESP32

Живлення більшості компонентів здійснюється від виводу 3,3 В ESP32. Виняток становлять сенсори MQ-серії та MH-Z19B, які потребують 5 В і підключаються через DC-DC перетворювач. Загальне споживання вузла у момент активного вимірювання складає близько 150 мА, а у режимі deep sleep — менше 1 мА.

2.3 Алгоритми збору та первинної обробки даних на автономних вузлах

Алгоритмічне забезпечення сенсорного вузла вирішує три взаємопов'язані завдання. Перше — організація циклічного опитування сенсорів з контролем часу прогріву і якості вимірювань. Друге — первинна цифрова фільтрація сигналів кожного каналу для усунення шумових флуктуацій. Третє і найважливіше з точки зору прийняття рішень — об'єднання даних від усіх сенсорів методом Data Fusion для формування єдиного інтегрального показника рівня небезпеки, на підставі якого вузол самостійно визначає частоту і терміновість відправлення пакета. Логіка реалізована у вигляді кінцевого

автомата (Finite State Machine), що послідовно проходить стани: пробудження — ініціалізація — прогрів — вимірювання — обробка — передача — сон.

Після пробудження мікроконтролер ініціалізує всі периферійні пристрої та вмикає живлення сенсорів. Для газових MQ-сенсорів обов'язковий прогрів нагрівального елемента протягом щонайменше 30 секунд — лише після цього їхні покази стають стабільними і придатними для вимірювань. Для МН-Z19В достатньо 20–30 секунд між циклами при регулярному опитуванні. Радіаційний канал є особливим: накопичення імпульсів ГМ-трубки починається одразу при пробудженні і відбувається паралельно з прогрівом хімічних сенсорів, оскільки мінімальний статистично значущий інтервал підрахунку СРМ становить 30–60 секунд.

Протягом фази вимірювань мікроконтролер опитує кожен канал N разів (рекомендована кількість ітерацій — від 5 до 10) і зберігає масив відліків для наступного етапу фільтрації. Статистична похибка підрахунку імпульсів ГМ-трубки визначається розподілом Пуассона і становить відносно $1/\sqrt{N}$, де N — кількість зареєстрованих імпульсів. При фоновому рівні близько 20 СРМ за 60 секунд накопичується близько 20 імпульсів, що дає відносну похибку $\pm 22\%$. При підвищеному радіаційному фоні кількість імпульсів зростає і похибка зменшується. Частота опитування визначається балансом між вимогами до своєчасності виявлення загрози та ресурсом акумулятора. Добовий баланс споживання описується формулою:

$$E_{\text{доба}} = (I_{\text{акт}} \cdot T_{\text{акт}} + I_{\text{сон}} \cdot T_{\text{сон}}) \cdot U \cdot 24,$$

де $I_{\text{акт}}$ — струм у активному стані, приблизно 150 мА;

$T_{\text{акт}}$ — тривалість активної фази, приблизно 60 с за цикл;

$I_{\text{сон}}$ — струм у режимі сну, приблизно 0,15 мА;

$T_{\text{сон}}$ — тривалість сну, наприклад 240 с при 5-хвилинному циклі;

U — напруга живлення, 3,7 В.

За цих умов добове споживання становить близько 120 мА/год, що при акумуляторі 5000 мА/год забезпечує більше 40 діб автономної роботи без підзарядки.

Сирі відліки кожного сенсорного каналу містять різні за природою джерела похибок: власний шум АЦП мікроконтролера, температурний дрейф чутливого елемента, флуктуації концентрації газу у точці вимірювання, а для MQ-серії — ще й перехресна чутливість до кількох газів одночасно. Для зменшення впливу цих факторів застосовується рекурсивний фільтр Калмана першого порядку — один з найбільш поширених і добре вивчених інструментів цифрової фільтрації у вбудованих системах реального часу.

Фільтр Калмана поєднує апіорну оцінку стану системи (засновану на попередньому відфільтрованому значенні) з новим виміром, зважуючи їх відповідно до відносної довіри до кожного джерела. Математичні рівняння дискретного фільтра першого порядку для скалярного випадку мають вигляд:

$$\begin{aligned}\hat{x}_k &= \hat{x}_{k-1} + K_k \cdot (z_k - \hat{x}_{k-1}), \\ K_k &= P_{k-1} / (P_{k-1} + R), \\ P_k &= (1 - K_k) \cdot P_{k-1} + Q,\end{aligned}$$

де $\hat{x}_k$ — відфільтрована оцінка концентрації на кроці k ;

z_k — поточний вимір сенсора;

K_k — посилення Калмана, коефіцієнт від 0 до 1, що визначає вагу нового виміру у результируючій оцінці;

P_k — дисперсія похибки оцінки;

R — дисперсія шуму вимірювання, що визначається з технічної документації або експериментально;

Q — дисперсія шуму процесу, що визначається емпірично.

Вибір параметрів Q і R потребує балансу між двома крайніми випадками. При великому значенні Q фільтр швидко реагує на реальні зміни концентрації, але гірше придушує шум. При малому Q відповідь на зміни повільна, зате

показники є стабільними. Для хімічних газових сенсорів рекомендованими початковими значеннями є $Q = 0,01$ і $R = 0,5$, для радіаційного каналу — $Q = 0,1$ і $R = 0,3$, оскільки там флуктуації мають природну пуасонівську природу і є ширшими за шум АЦП.

Найважливішою алгоритмічною задачею є прийняття рішення про рівень загрози у ситуації, коли одночасно фіксуються відхилення за кількома каналами. Проста логіка — «тривога, якщо хоча б один показник перевищує поріг» — є надмірно консервативною і призводить до великої кількості хибних спрацьовувань через шуми окремих сенсорів. Підхід «тривога лише при перевищенні більшості порогів» не враховує різницю у ступені небезпечності різних факторів. Тому реалізовано методику зваженого агрегування нормованих показників, відому у системах підтримки прийняття рішень як Composite Hazard Index (CHI) і що відповідає методологічній базі MCDM (Multi-Criteria Decision Making). Методика складається з трьох послідовних кроків. На першому кроці кожен відфільтрований показник нормується відносно відповідного граничного значення:

$$n_i = c_i / c_i^{\text{threshold}},$$

де n_i — нормований показник i -го фактора;

c_i — поточна відфільтрована концентрація або потужність дози;

$c_i^{\text{threshold}}$ — граничне значення з таблиці 3.1.

Значення $n_i = 1,0$ відповідає точно пороговому рівню, $n_i > 1,0$ означає перевищення норми.

На другому кроці виконується зважене підсумовування:

$$CHI = \sum_i \omega_i \cdot n_i, \quad \sum_i \omega_i = 1.$$

Вагові коефіцієнти ω_i відображають відносну небезпечність кожного фактора і встановлюються з урахуванням токсикологічних характеристик,

нормативних пріоритетів та аналізу реальних аварійних сценаріїв. Їх значення наведено у таблиці 2.3.

Таблиця 2.3 — Вагові коефіцієнти для розрахунку Composite Hazard Index

Фактор	ω_i	Обґрунтування пріоритету
Гамма-випромінювання	0,35	Найвища гострота ризику; нормативний пріоритет НРБУ-97 та ІАЕА
CO	0,20	Швидкодіюча токсична дія при аваріях і пожежах
H ₂ S	0,20	Висока летальність вже при 1 ppm тривалого впливу
NH ₃	0,10	Характерний для хімічних та сільськогосподарських аварій
TVOC	0,10	Хронічна токсичність, індикатор хімічного забруднення
CO ₂	0,05	Індикатор якості повітря, відносно менший гострий ризик

На третьому кроці вузол використовує розраховане значення СНІ для адаптивного вибору режиму роботи. При СНІ менше 0,5 обстановка є нормальною — дані відправляються кожні 5 хвилин. При значенні СНІ від 0,5 до 1,0 фіксується підвищений рівень небезпеки — інтервал між передачами скорочується до 60 секунд. При СНІ рівному або більшому за 1,0 вузол негайно формує та надсилає пакет із встановленим прапорцем тривоги, не очікуючи наступного планового циклу. Таким чином, поведінка вузла автоматично адаптується до реальної обстановки.

Алгоритм роботи вузла показано на рисунку 2.5 у вигляді блок-схеми. Схема охоплює етапи: ініціалізацію периферії, паралельний прогрів сенсорів і накопичення СРМ, N-кратне опитування кожного каналу, фільтрацію, нормування показників, розрахунок СНІ, прийняття рішення про передачу/сон.

Пакет, що передається модулем LoRa, містить 18 байт корисного навантаження. Мінімальний розмір є принциповою конструктивною вимогою: час передачі у ефірі (Time on Air) для LoRa прямо пропорційний кількості байтів, а загальна пропускна здатність каналу обмежена нормою Duty Cycle у 1% для діапазону 868 МГц.

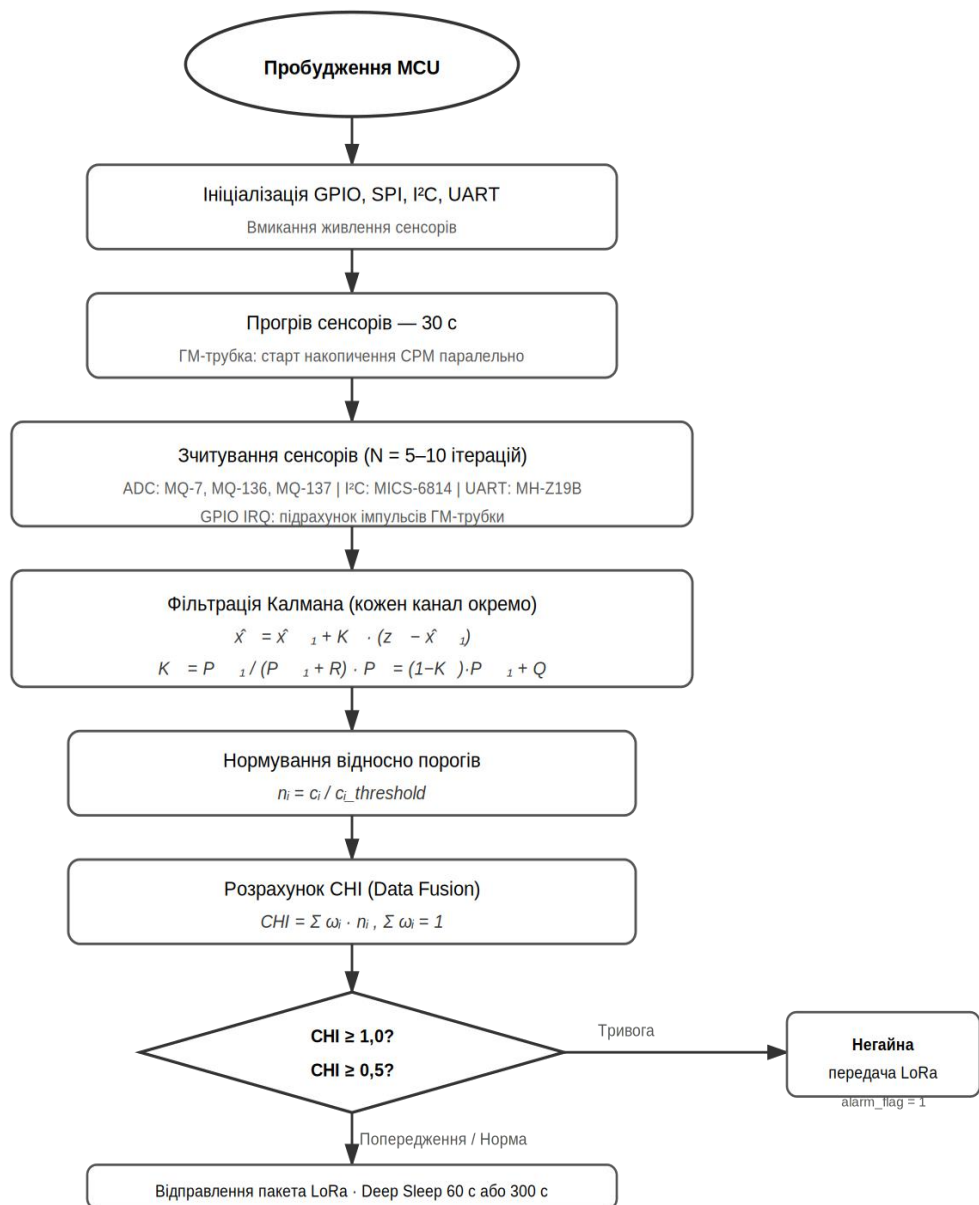


Рисунок 2.5 — Блок-схема алгоритму роботи автономного сенсорного вузла

Структура пакета показана на рисунку 2.6.

node_id 0–1	timestamp 2–3	radiation 4–5	co_ppm 6–7	co2_ppm 8–9	nh3 / h2s 10–13	tvoc_ppb 14–15	chi/flags 16–17
uint16	uint16	uint16	uint16	uint16	uint16×2	uint16	uint8×2

Рисунок 2.6 — Структура пакета даних LoRa (18 байт корисного навантаження)

Пакет включає ідентифікатор вузла (поле `node_id`), мітку часу як кількість секунд від опорної епохи за модулем 65536 (поле `timestamp`), виміряні значення всіх шести каналів у стиснутому цілочисловому форматі, значення СНІ помножене на 100 і записане як ціле 8-бітове число, та байт бітових прапорців тривоги — по одному біту на кожен фактор (`b0` — гамма-випромінювання, `b1` — CO, `b2` — H₂S і так далі). Такий формат дозволяє шлюзу і серверу однозначно ідентифікувати вузол-джерело, відновити мітку часу і миттєво, ще до повного розбирання пакета, виявити активні тривоги за значенням байта `flags`. Детальна реалізація обробки пакетів на стороні шлюзу і сервера розглядається у наступному розділі.

Таким чином, в даному розділі обґрунтовано архітектуру та апаратний склад мережі сенсорного моніторингу радіаційних і хімічних загроз. Запропоновано чотирирівневу структуру (вузол — шлюз — сервер — мобільний клієнт), що забезпечує масштабованість від кількох до сотень вузлів, відмовостійкість і функціональну ізоляцію кожного рівня. Перелік шести контрольованих факторів — гамма-випромінювання, CO, CO₂, TVOC, NH₃, H₂S. Для кожного фактора обрано відповідний сенсор з урахуванням вимог до точності, доступності та можливості підключення до ESP32. Описано алгоритм циклічного опитування з адаптивним інтервалом сну, що забезпечує автономну роботу протягом 40 і більше діб. Розроблено методику Data Fusion на основі Composite Hazard Index для інтегральної оцінки одночасно присутніх загроз. Визначено структуру 18-байтового пакета LoRa, що мінімізує час передачі і забезпечує достатній ефірний ресурс для мережі з сотнями вузлів.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ МЕРЕЖЕВОЇ ВЗАЄМОДІЇ ТА МОБІЛЬНОГО КЛІЄНТА

3.1 Організація передачі даних за протоколом LoRa/LoRaWAN

					ДП.МТІР.9702469.00.00.00.000 ПЗ	
		№				40

Передача вимірювальних даних від автономних сенсорних вузлів до серверного рівня є ключовим комунікаційним завданням усієї системи. Для його вирішення застосовано протокол LoRaWAN — відкритий медіадоступний стандарт організації LoRa Alliance, що визначає правила адресації, шифрування, підтвердження доставки та керування потужністю поверх фізичного рівня LoRa. LoRaWAN використовує топологію зіркоподібної мережі зірок (star-of-stars): сенсорні вузли передають дані напряму на шлюзи, а шлюзи пересилають їх до централізованого мережевого сервера. Така топологія є асиметричною — вузли не взаємодіють між собою і не виконують ретрансляції, що суттєво спрощує протокол і знижує споживання. Один шлюз може одночасно обслуговувати до восьми каналів LoRa завдяки концентратору SX1301/SX1303, що дозволяє приймати повідомлення від кількох сотень вузлів в одній зоні покриття.

Загальну топологію системи показано на рисунку 3.1. Два шлюзи разом з'єднані з єдиним мережевим сервером, що виключає втрату даних при відмові одного зі шлюзів: повідомлення, прийняте обома шлюзами, буде дедубліковане на рівні мережевого сервера. Мережевий сервер, у свою чергу, публікує дані у MQTT-брокер, звідки їх отримують сервер зберігання та мобільний застосунок.

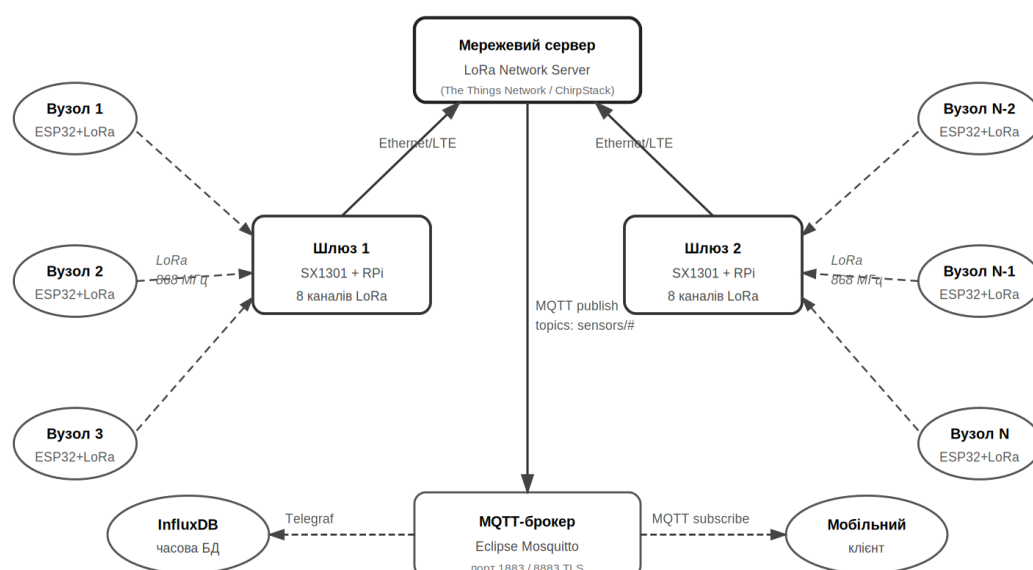


Рисунок 3.1 — Топологія мережі LoRaWAN з двома шлюзами, MQTT-брокером і мобільним клієнтом

Стандарт LoRaWAN визначає три класи пристроїв відповідно до режиму роботи вікон прийому. Їх порівняння наведено у таблиці 3.1.

Таблиця 3.1 — Порівняння класів пристроїв LoRaWAN

Клас	Вікна прийому	Сфера застосування
A (All)	2 вікна після кожної передачі; решта часу — сон. Мінімум споживання	Сенсорні вузли на батарейному живленні — основний клас проєкту
B (Beacon)	Синхронізовані вікна за маяком шлюзу, додатково до Class A	Застосунки з вимогами до затримки відповіді від сервера
C (Continuous)	Постійно відкритий прийом, відкривається лише під час передачі	Мережеві пристрої з постійним живленням (шлюзи, реле)

Усі сенсорні вузли проєкту працюють у класі A, оскільки дворазовий діалог (передача + два вікна підтвердження) повністю задовольняє вимоги системи і забезпечує максимальну автономність.

Мережевий сервер LoRaWAN публікує розшифровані дані сенсорів у MQTT-брокер Eclipse Mosquitto. MQTT (Message Queuing Telemetry Transport) є легковаговим протоколом публікації/підписки, розробленим спеціально для IoT-застосунків з обмеженою пропускною здатністю каналу [9, 10]. Протокол забезпечує три рівні якості сервісу, що детально наведені у таблиці 3.2.

Таблиця 3.2 — Рівні якості обслуговування MQTT (QoS)

Рівень QoS	Гарантія доставки	Кількість передач	Використання в системі
QoS 0	Не більше одного разу (fire and forget)	1	Потокові дані сенсорів у нормальному режимі
QoS 1	Щонайменше один раз (з підтвердженням)	1+	Дані при підвищеному СНІ (0.5–1.0)
QoS 2	Рівно один раз (4-кроковий handshake)	4	Пакети тривоги (СНІ $\geq$ 1.0)

Таким чином, в даному розділі розглянуті питання структури мережі сенсорних вузлів та механізмів передачі повідомлень.

3.2 Розробка програмного додатку вимірювання хімічних забруднень та радіації

Програмне забезпечення сенсорного вузла реалізовано у середовищі Arduino IDE 2.x у вигляді скетчу (sketch) — самодостатньої програми для мікроконтролера ESP32. Arduino IDE є де-факто стандартним середовищем для прототипування IoT-вузлів завдяки відкритості, багатій екосистемі бібліотек та підтримці ESP32 через офіційний пакет плат Espressif. Налаштування Arduino IDE для роботи з ESP32 передбачає використання пакету Arduino IDE. Для роботи з ESP32 у Arduino IDE необхідно додати URL менеджера плат Espressif: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json. Після встановлення пакету esp32 версії 3.x у меню "Інструменти → Плата" обирається "ESP32 Dev Module". Налаштування завантаження: Flash Mode — QIO, Flash Size — 4 MB, Upload Speed — 921600 бод, Partition Scheme — Default 4MB. Порт відповідає COM-порту, на якому з'явився пристрій після підключення через USB-кабель. У монітор порту встановлюється швидкість 115200 бод, що відповідає параметрам Serial.begin() у скетчі.

Необхідні бібліотеки встановлюються через менеджер бібліотек Arduino IDE (Ctrl+Shift+I): LoRa від Sandeep Mistry (версія 0.8.x) — для роботи з модулями SX1276/SX1262 через SPI; MHZ19 від WifWaf (версія 3.x) — для MH-Z19B через UART; SparkFun MICS6814 або сумісна — для MICS-6814 через I²C. Бібліотека Wire (I²C) і HardwareSerial входять до стандартного пакету ESP32.

Проект організовано як багатофайловий скетч: головний файл sensor_node.ino з функціями setup() та loop() та кілька заголовних файлів (.h) і файлів реалізації (.cpp), кожен з яких відповідає за окремий функціональний блок. Така модульна структура спрощує налагодження і подальшу підтримку коду. Схема залежностей між модулями наведена на рисунку 3.2.

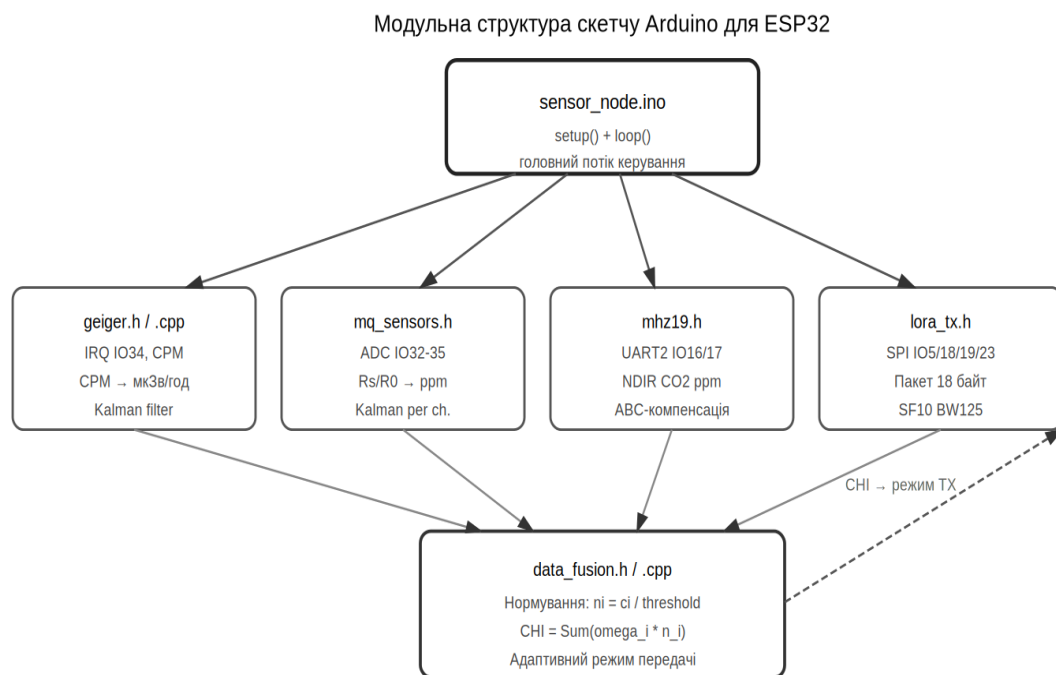


Рисунок 3.2 — Модульна структура скетчу Arduino для ESP32

Модуль `geiger.h` реалізує підрахунок імпульсів ГМ-трубки через механізм апаратних переривань ESP32. Функція-обробник переривання (ISR — Interrupt Service Routine) позначається атрибутом `IRAM_ATTR` для розміщення у швидкодіючій пам'яті IRAM і просто інкрементує лічильник. Основний потік читає і скидає лічильник кожні 60 секунд, отримуючи значення CPM. Оскільки ISR і основний потік звертаються до спільної змінної, вона оголошується з кваліфікатором `volatile` для запобігання оптимізацій компілятора.

Модуль `mq_sensors.h` реалізує зчитування резистивних сенсорів серії MQ через вбудований АЦП ESP32. Функція `analogRead()` повертає ціле значення від 0 до 4095 (12-бітний АЦП), що перетворюється у вихідну напругу за формулою:

$$V_{out} = ADC_value \cdot V_{cc} / 4095,$$

де $V_{cc} = 3,3 \text{ В}$ — напруга живлення.

Опір чутливого елемента розраховується як:

$$R_s = R_L \cdot (V_{cc} - V_{out}) / V_{out},$$

де R_L — навантажувальний резистор (зазвичай 10 кОм для MQ-серії).

Далі за відношенням R_s/R_0 та параметрами кривої з технічної документації обчислюється концентрація газу у ppm за формулою попереднього розділу. Фільтрація виконується рекурсивним фільтром Калмана, реалізованим окремою структурою `KalmanState` для кожного каналу.

Модуль `mhz19.h` обмінюється даними з CO2-сенсором MH-Z19B за протоколом UART. Кожні 5 секунд надсилається 9-байтовий запит зчитування (команда 0x86), у відповідь сенсор повертає 9 байт, де байти 2 і 3 містять значення CO2 у ppm у форматі big-endian. Перевірка контрольної суми (XOR від байт 1–8, взятий по модулю 256, має дорівнювати 0xFF мінус байт 8) захищає від помилок передачі.

Модуль `data_fusion.h` реалізує функцію розрахунку Composite Hazard Index. Функція приймає структуру `SensorData` з відфільтрованими значеннями всіх каналів, нормує кожен показник відносно табличного порогу (таблиця 3.1), множить на ваговий коефіцієнт (таблиця 3.3) і підсумовує. Функція `adaptSleep()` повертає інтервал сну у мікросекундах залежно від СНІ: $300 \cdot 10^6$ мкс при $\text{СНІ} < 0,5$; $60 \cdot 10^6$ мкс при $0,5 \leq \text{СНІ} < 1,0$; 0 (негайна повторна передача) при $\text{СНІ} \geq 1,0$.

Модуль `lora_tx.h` формує 18-байтовий пакет, серіалізуючи поля структури `SensorData` у бінарний буфер. Передача виконується функціями бібліотеки: `LoRa.beginPacket()` → запис байтів → `LoRa.endPacket()`. Параметри LoRa встановлюються один раз у `setup()`: частота 868 МГц, SF10, BW125 кГц, CR 4/5, потужність 20 дБм, преамбула 8 символів.

Рисунок 3.3 ілюструє середовище Arduino IDE під час роботи зі скетчем. У верхньому рядку відображається назва поточного файлу, у панелі інструментів — кнопки верифікації (компіляції) та завантаження у мікроконтролер. Редактор коду підтримує підсвічування синтаксису, автодоповнення та перехід до оголошення функції (F12).

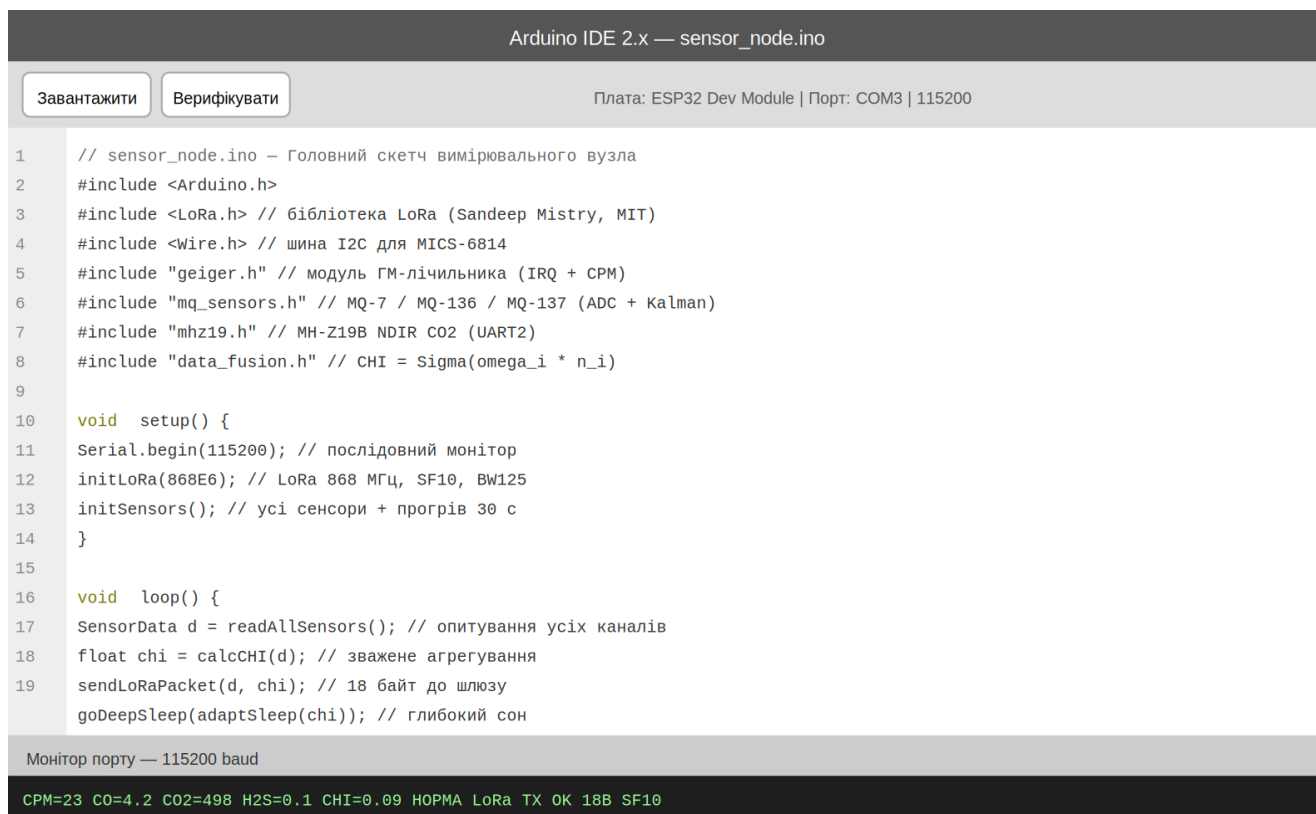


Рисунок 3.3 — Вигляд Arduino IDE під час роботи зі скетчем вузла

Нижня панель "Монітор порту" є основним інструментом налагодження: функція `Serial.println()` виводить поточні значення всіх каналів після кожного циклу вимірювання. Виводяться: значення CPM та потужність дози у мкЗв/год, концентрації CO, CO₂, H₂S, NH₃ у ppm, значення CHI і поточний режим роботи (НОРМА / ПОПЕРЕДЖЕННЯ / ТРИВОГА), а також статус передачі LoRa. Завдяки цьому розробник може верифікувати правильність перетворень і фільтрації без підключення до реального шлюзу.

Типовий сеанс налагодження починається з перевірки ініціалізації: якщо `LoRa.begin()` повертає 0 — не вдалося знайти модуль SPI, що свідчить про помилку у підключенні або у номерах виводів. Далі перевіряється прогрів сенсорів: протягом 30 секунд значення Rs/R0 повинні стабілізуватися. Після цього функція `calcCHI()` перевіряється у відомих умовах — у чистому повітрі CHI має бути меншим за 0,1 при нормальному радіаційному фоні.

Таблиця 3.3 систематизує основні функції та оператори скетчу, пояснюючи їх призначення у контексті системи моніторингу.

					ДП.МТІР.9702469.00.00.00.000 ПЗ	
		№	1			46

Таблиця 3.3 — Основні функції та оператори скетчу і їх призначення

Функція / оператор	Призначення у скетчі
#include <LoRa.h>	Підключення бібліотеки LoRa Sandeep Mistry для роботи з SX1276/SX1262 через SPI
LoRa.begin(868E6)	Ініціалізація радіомодуля на частоті 868 МГц ISM-діапазону
LoRa.setSpreadingFactor(10)	Встановлення коефіцієнту розкиду SF10 (баланс дальність/швидкість)
attachInterrupt(IO34, isr, RISING)	Підключення ISR-обробника переривань ГМ-трубки на зростаючий фронт
analogRead(IO32)	Зчитування 12-бітного АЦП для MQ-сенсорів (0–4095 → 0–3.3 В)
Serial2.begin(9600,SERIAL_8N1,16,17)	Ініціалізація UART2 для MH-Z19B: RX=IO16, TX=IO17
esp_deep_sleep(us)	Переведення ESP32 у режим глибокого сну на задану кількість мікросекунд
Wire.begin(IO21, IO22)	Ініціалізація I2C-шини для MICS-6814: SDA=IO21, SCL=IO22
calcCHI(SensorData&)	Власна функція: нормування, зважування $CHI = \sum \omega \cdot p_i$, повертає float

Реалізація методики Data Fusion у вбудованому програмному забезпеченні враховує обмеження мікроконтролера: відсутність операційної системи, обмежена пам'ять (520 КБ SRAM для ESP32) та необхідність завершити всі обчислення до виклику функції відправлення за кілька мілісекунд. Алгоритм реалізовано як пряму послідовність арифметичних операцій з числами з рухомою комою (float, 32 біти IEEE 754), що ESP32 виконує апаратно завдяки блоку FPU. Обчислення CHI на мікроконтролері виконується у два проходи. На першому проході застосовується фільтр Калмана до кожного каналу: нове значення z_k зчитується з відповідного сенсора, оновлюється дисперсія P і обчислюється відфільтроване $\hat{x}_k$ за формулами (3.3)–(3.5). На другому проході обчислюються нормовані показники p_i і виконується зважена сума за формулою (3.7). Сумарна кількість операцій з рухомою комою на один цикл не перевищує 80, що при тактовій частоті 240 МГц займає менше 1 мс.

Практично важливим аспектом є захист від ділення на нуль у формулі (3.4): якщо $P_{k-1} = 0$ і $R = 0$, посилення Калмана стає невизначеним. У реалізації R завжди ненульовий (мінімум 0,01), що гарантує стабільність алгоритму. Аналогічно, у формулі (4.3) обробляється випадок $V_{out} = 0$ (сенсор не підключений або коротке замикання): у цьому разі R_s встановлюється у максимальне значення, а концентрація — у нуль.

3.3 Розробка мобільного додатка для візуалізації загроз

Мобільний застосунок є кінцевою ланкою системи — інтерфейсом, через який оператор у режимі реального часу отримує повну картину стану мережі моніторингу. Основні вимоги до застосунку: крос-платформеність (Android та iOS з єдиної кодової бази), відображення поточних значень усіх сенсорів і розрахованого СНІ, карта з позначками вузлів, автоматичне push-сповіщення при тривозі, доступ до архіву вимірювань [13-15].

Для розробки мобільного застосунку доцільно використати середовище (фреймворк) Flutter (Google, відкрита ліцензія BSD) з мовою програмування Dart. Flutter компілює нативний ARM-код для Android та iOS з однієї кодової бази, забезпечуючи продуктивність, близьку до нативних застосунків, і повний контроль над пікселями. Альтернативами є React Native (Meta) та розробка окремо для Android (Kotlin) і iOS (Swift). React Native було відхилено через проміжний JavaScript-міст, що вносить додаткові затримки при обробці потоку MQTT-повідомлень. Окрема розробка вдвічі збільшила б обсяг коду і час підтримки. Flutter вирізняється найшвидшим часом запуску і найменшим споживанням батареї серед крос-платформених рішень згідно з порівняльним аналізом.

Застосунок розподілено на три шари, що наочно показано на рисунку 3.4. Шар UI містить головний екран з показниками, екран тривоги, карту вузлів і екран

архівних графіків. Шар BLoC обробляє події від UI, звертається до репозиторію і генерує нові стани, на які реагують відповідні widget-и. Шар даних встановлює з'єднання з брокером та декодує JSON-повідомлення), `SensorRepository` зберігає останні значення кожного вузла) та `NotificationService` (відповідає за push-сповіщення) .

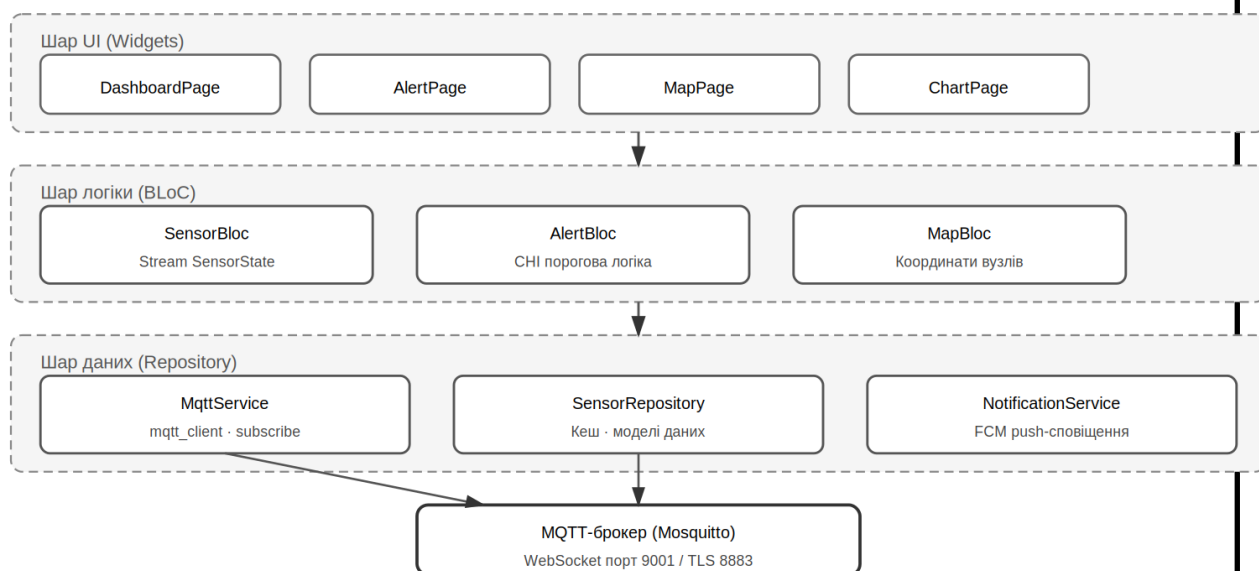


Рисунок 3.4 — Архітектура мобільного застосунку на базі Flutter та патерну BLoC

Список пакетів pub.dev, що доступні для системи моніторингу, наведено у таблиці 4.4.

Таблиця 3.4 — Пакети Flutter та їх призначення у застосунку

Пакет	Версія	Призначення
mqtt_client	2.9.x	MQTT over WebSocket, підписка на топіки сенсорів
flutter_bloc	8.x	State management (BLoC-патерн), реактивні потоки
fl_chart	0.66.x	Побудова лінійних графіків CHI та концентрацій
flutter_map	6.x	Карта OpenStreetMap, маркери вузлів

Пакет	Версія	Призначення
flutter_local_notifications	17.x	Локальні push-сповіщення при $СНІ \geq 1.0$
firebase_messaging	15.x	FCM push при закритому застосунку
shared_preferences	2.x	Збереження параметрів (адреса брокера, пороги)

Рисунок 3.5 ілюструє три можливі екрани застосунку. Головний екран відображає зведений стан мережі: загальний рівень небезпеки СНІ, плитки (tile) з поточними значеннями кожного сенсорного каналу і мініатюру карти з позначками вузлів. Екран тривоги з'являється автоматично при надходженні alert-повідомлення: він виводить значення СНІ, перелік перевищених норм і кнопку переходу до детального архіву. Третій екран показує лінійний графік зміни СНІ у часі для обраного вузла, побудований пакетом `fl_chart` на основі даних кільцевого буфера.

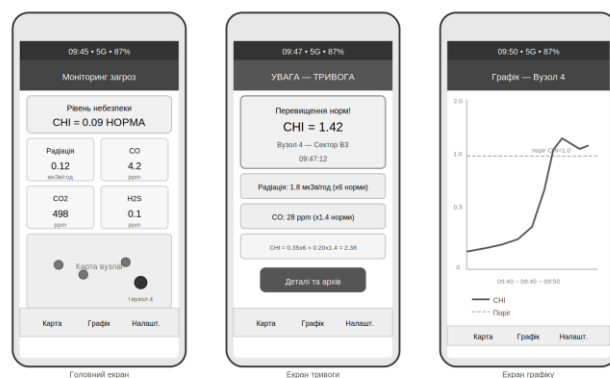


Рисунок 3.5 — Екрани мобільного застосунку: головний, тривога, графік СНІ

Поки значення не перевищує встановлений поріг, система лише продовжує моніторинг середовища. Якщо концентрація газу стає небезпечною, програма в ESP32 автоматично формує тривожне повідомлення. У повідомленні міститься інформація про небезпечну ситуацію та поточне значення концентрації.

Після цього ESP32 через Wi-Fi передає дані на сервер. Сервер виконує основну обробку інформації. Він приймає дані від мікроконтролера, аналізує їх, зберігає у базі даних та формує push-повідомлення для мобільного застосунку.

Мобільний застосунок створюється у середовищі Flutter. Flutter дозволяє розробити одну програму одночасно для Android та iOS. Застосунок постійно підтримує зв'язок із сервером через Інтернет. Коли сервер формує тривожне повідомлення, телефон отримує його та показує користувачу попередження про перевищення рівня хімічного забруднення. Навіть якщо програма не відкрита, операційна система смартфона може показати push-повідомлення на екрані. Загальний процес роботи системи можна описати так:

Сенсор → ESP32 → Wi-Fi → Сервер → Flutter → Телефон

Нижче наведено приклад скетчу для ESP32 у середовищі Arduino IDE.

```
#include <WiFi.h>
#include <HTTPClient.h>

const char* ssid = "YOUR_WIFI";
const char* password = "YOUR_PASSWORD";
// Адреса сервера
String serverName =
"http://192.168.1.100:5000/alert";

int gasSensorPin = 34;
float limitPPM = 50.0;
void setup() {
    Serial.begin(115200);
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Підключення до Wi-Fi...");
    }
    Serial.println("Wi-Fi підключено");
}
void loop() {
    int sensorValue = analogRead(gasSensorPin);

    // Умовне перетворення в ppm
    float ppm = sensorValue * 0.1;
```

```

Serial.print("Рівень газу: ");
Serial.println(ppm);
// Перевірка перевищення норми
if (ppm > limitPPM) {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(serverName);
        http.addHeader(
            "Content-Type",
            "application/json");
        String jsonData =
            "{\"gas_level\": "
            + String(ppm) + "}";
        int httpResponseCode =
            http.POST(jsonData);
        Serial.print("Код відповіді: ");
        Serial.println(httpResponseCode);
        http.end();
    }
}
delay(10000);
}

```

У цьому коді ESP32 зчитує аналоговий сигнал із газового сенсора, умовно переводить його у значення ppm та перевіряє, чи перевищує концентрація допустиму норму. Якщо рівень газу стає небезпечним, ESP32 формує JSON-повідомлення та передає його на сервер через HTTP POST-запит. Сервер може бути створений, наприклад, мовою Python із використанням бібліотеки Flask, однак це виходить за межі нашого проекту. Сервер відповідає за прийом даних від ESP32, їх аналіз та визначення необхідності створення аварійного повідомлення.

У Flutter застосунок (смартфон) може отримувати повідомлення від сервера через WebSocket, HTTP або спеціальні push-служби. Нижче наведено спрощений приклад Flutter-коду для отримання повідомлення від сервера. У реальній системі Flutter-застосунок не просто змінює текст на екрані, а підтримує зв'язок із сервером та автоматично приймає аварійні повідомлення через Інтернет. Таким чином ESP32 виконує вимірювання та передавання даних, сервер забезпечує обробку, аналіз і формування повідомлень, а Flutter використовується для створення мобільного застосунку, який показує користувачу попередження про небезпечний рівень хімічного забруднення у режимі реального часу.

```

import 'package:flutter/material.dart';
void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: HomePage(),
    );
  }
}
class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() =>
    _HomePageState();
}
class _HomePageState
  extends State<HomePage> {
  String message =
    "Очікування повідомлень";
  void receiveAlert() {
    setState(() {
      message =
        "⚠️ Небезпечний рівень газу!";
    });
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Моніторинг газу"),
      ),
      body: Center(
        child: Text(
          message,
          style: TextStyle(fontSize: 24),
        ),
      ),
      floatingActionButton:
        FloatingActionButton(
          onPressed: receiveAlert,
          child: Icon(Icons.warning),
        ),
    );
  }
}

```

Таким чином, в даному розділі описано програмну реалізацію всіх рівнів системи моніторингу.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи розроблено сенсорну мережу для моніторингу радіаційних і хімічних загроз із застосуванням технології LoRa та мобільних клієнтів. Головною проблемою, яку вирішує система, є виявлення комбінованої небезпеки: ситуацій, коли жоден із контрольованих параметрів не перевищує встановлених граничнодопустимих норм, але їх спільний вплив на організм людини є потенційно шкідливим.

Для усунення цього недоліку в роботі розроблено алгоритм комплексної оцінки ризику, який враховує одночасний вплив кількох факторів і формує інтегральний індекс небезпеки. Це дозволяє системі генерувати попередження навіть у випадках, коли кожен із параметрів перебуває в межах норми, але їх поєднання створює реальну загрозу для здоров'я людини.

Апаратна частина кожного сенсорного вузла включає датчики радіаційного фону та хімічних забруднювачів, мікроконтролер для первинної обробки даних і радіомодуль LoRa для передачі результатів на великі відстані при низькому енергоспоживанні. Мережа вузлів охоплює контрольовану територію та забезпечує безперервний збір просторово розподілених даних.

Розроблений мобільний клієнт надає операторам доступ до поточної карти рівнів забруднення, журналу подій та сповіщень у режимі реального часу. Це скорочує час реакції персоналу на виникнення небезпечної ситуації та дозволяє приймати обґрунтовані рішення щодо евакуації або локалізації джерела загрози.

Практична значущість роботи полягає у тому, що запропонований підхід до комплексної оцінки ризику може бути застосований на промислових підприємствах, у зонах екологічного контролю, а також у системах цивільного захисту, де традиційні порогові методи виявлення є недостатніми для забезпечення реальної безпеки персоналу

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Псуй
2. Щудро С., Єфанов А. Небезпека радіаційного забруднення під час повномасштабного вторгнення та його вплив на екологію України. Scientific Collection «InterConf». 2025. № 214. С. 142–148.
3. Хилько М. І. Екологічна безпека України : навчальний посібник. Київ : Знання, 2017. 324 с.
4. Медведєв В. В., Лактіонова Т. М. Моніторинг хімічного забруднення та деградаційних процесів у ґрунтах України : монографія. Харків : Смугаста типографія, 2021. 210 с.
5. Ковальов О. В., Сидоренко М. П. Особливості моделювання радіаційного забруднення техногенним тритієм водних екосистем України. Збірник наукових праць Інституту проблем моделювання в енергетиці ім. Г. Є. Пухова НАН України. 2024. Вип. 102. С. 45–58.
6. Кочан В. В., Саченко А. О. Особливості інтеграції вимірювальних даних у розподілених інтелектуальних сенсорних мережах. Вісник Тернопільського національного технічного університету. 2023. Т. 109, № 1. С. 84–95.
7. Основи метрології та вимірювальної техніки / Лис, О.М., Якименко, М.В., Шинкаренко [та ін.]. - Львів: Видавництво Львівської політехніки, 2021. – 424 с.
8. Цюцюра В. Д., Цюцюра С. В. Метрологія та основи вимірювань: К.:Знання-Прес, 2018. – 180с.
9. LoRa Alliance Technical Committee. LoRaWAN Specification v1.0.3. — 2018. URL: https://lora-alliance.org/resource_hub/lorawantm-specification-v1-0-3/ (дата звернення: 10.04.2026).
10. Semtech Corporation. SX1301 Multi-channel Concentrator Datasheet. — 2017. URL: <https://semtech.com/products/wireless-rf/lora-connect/sx1301> (дата звернення: 12.04.2025).

11. Sinha R.S., Wei Y., Hwang S.H. A survey on LPWA technology: LoRa and NB-IoT // ICT Express. — 2017. — Vol. 3, No. 1. — P. 14–21. DOI: 10.1016/j.icte.2017.03.004.
12. OASIS Standard. MQTT Version 3.1.1. — 2014. URL: <https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html> (дата звернення: 15.04.2025).
13. Arduino LLC. Arduino IDE 2 Documentation. — 2024. URL: <https://docs.arduino.cc/software/ide-v2> (дата звернення: 18.04.2025).
14. Espressif Systems. Arduino-ESP32 Getting Started Guide. — 2024. URL: https://docs.espressif.com/projects/arduino-esp32/en/latest/getting_started.html (дата звернення: 18.04.2025).
15. Biørn-Hansen A. et al. Progressive Web Apps vs. Native Apps: Cross-Platform Mobile Development Comparison // Proceedings of the 14th Int. Conference on Web Information Systems and Technologies. — 2020. — P. 35–43.

ДОДАТКИ

ДОДАТОК А

Програмний скетч роботи автономного вузла (sensor_node.ino

```
// sensor_node.ino – скетч сенсорного вузла системи
// моніторингу радіаційних і хімічних загроз
// --- Підключення бібліотек
#include <LoRa.h>          // Бібліотека LoRa (Sandeep Mistry). Керує чіпом
SX1276/SX1262
#include <Wire.h>          // Шина I2C (вбудована). Потрібна для MICS-6814
#include <HardwareSerial.h> // UART2. Потрібна для CO2-сенсора MH-Z19B
#include <driver/adc.h>     // Калібрований АЦП ESP32 (Espressif IDF)
#include <esp_adc_cal.h>    // Характеристики АЦП для точнішого
перетворення
#include <esp_sleep.h>     // API режимів сну ESP32 (deep sleep / light
sleep)
// --- Призначення виводів GPIO -
// Модуль LoRa SX1262 (SPI)
#define LORA_SS    5  // NSS (Chip Select) – вибір чіпа LoRa
#define LORA_RST   14 // RESET – скидання модуля LoRa
#define LORA_DIO0  26 // DIO0 – переривання від LoRa (TxDone/RxDone)
#define LORA_SCK   18 // SCK – тактовий сигнал SPI
#define LORA_MISO  19 // MISO – дані від LoRa до ESP32
#define LORA_MOSI  23 // MOSI – дані від ESP32 до LoRa

// ГМ-трубка SBM-20 (імпульсний вхід)
#define GEIGER_PIN 34 // GPIO34 – тільки вхід; TTL-імпульс від ГМ-трубки

// MQ-сенсори (аналогові входи АЦП)
#define MQ7_PIN    32 // ADC1_CH4 – CO (монооксид вуглецю)
#define MQ136_PIN  33 // ADC1_CH5 – H2S (сірководень)
#define MQ137_PIN  35 // ADC1_CH7 (тільки вхід) – NH3 (аміак)

// MH-Z19B CO2 (UART2)
#define MHZ_RX     16 // UART2 RX ← TX сенсора MH-Z19B
#define MHZ_TX     17 // UART2 TX → RX сенсора MH-Z19B

// MICS-6814 (I2C)
#define I2C_SDA    21 // SDA шини I2C
```

```

#define I2C_SCL    22  // SCL шини I2C
#define MICS_ADDR 0x75// I2C-адреса модуля MICS-6814

// --- Константи мережі та сенсорів -----
#define NODE_ID      7    // Унікальний номер цього вузла (0-65535)
#define LORA_FREQUENCY 866E6// Несуча 866 МГц (ISM-діапазон 863-870 МГц)
#define LORA_SF      10    // Spreading Factor SF10 → чутл. -137 дБм
#define LORA_BW     125000  // Smuga пропускання 125 кГц
#define LORA_CR      5     // Coding Rate 4/5 (мінімальний overhead)

// Інтервали роботи (мікросекунди для esp_sleep)
#define SLEEP_NORMAL 300000000ULL // 300 с – норма (CHI < 0.5)
#define SLEEP_WARN   600000000ULL // 60 с – підвищений (CHI 0.5-1.0)
#define WARMUP_MS     30000    // 30 с прогрів MQ-сенсорів
#define GEIGER_WINDOW 30000    // 30 с вікно накопичення CPM
#define N_SAMPLES     8       // Кількість ітерацій вимірювання

// Конверсійний коефіцієнт ГМ-трубки SBM-20
#define SBM20_CONV_FACTOR 153.8f // CPM → мкЗв/год: доза = CPM / 153.8

// Граничні (порогові) значення для нормування  $n_i = c_i / \text{threshold}_i$ 
#define THR_RADIATION 0.30f // мкЗв/год (НРБУ-97)
#define THR_CO        20.0f // ppm (ДСТУ EN 50271, 8-год. вплив)
#define THR_CO2       1000.0f // ppm (ASHRAE 62.1)
#define THR_TVOC      500.0f // ppb (WHO 2010)
#define THR_NH3       25.0f // ppm (ACGIH TLV)
#define THR_H2S       1.0f // ppm (OSHA PEL – стелева концентрація)

// Вагові коефіцієнти  $\omega_i$  для розрахунку CHI ( $\Sigma \omega_i = 1.0$ )
#define W_RADIATION 0.35f // Гамма – найвищий пріоритет
#define W_CO        0.20f // CO
#define W_CO2       0.05f // CO2
#define W_TVOC      0.10f // TVOC
#define W_NH3       0.10f // NH3
#define W_H2S       0.20f // H2S

// MQ-сенсори: опір у чистому повітрі R0 (Ом) та навантаж. резистор RL
#define MQ7_R0      27000.0f // Визначається при калібруванні в чистому
// повітрі
#define MQ136_R0    10000.0f
#define MQ137_R0    14000.0f

```

```

#define RL                10000.0f    // Навантажувальний резистор 10 кОм
(однаковий для всіх MQ)

#define VCC_MQ            5.0f    // Напруга живлення MQ-сенсорів (В)

// --- Структури даних -----

// Сипі / відфільтровані значення всіх сенсорних каналів
struct SensorData {
    float radiation_uSvh; // Потужність дози гамма (мкЗв/год)
    float co_ppm;         // CO (ppm)
    float co2_ppm;        // CO2 (ppm)
    float tvoc_ppb;       // TVOC (ppb), з MQ-135 або SGP30
    float nh3_ppm;        // NH3 (ppm)
    float h2s_ppm;        // H2S (ppm)
};

// Стан фільтра Калмана для одного вимірювального каналу
struct KalmanState {
    float x; // Поточна відфільтрована оцінка (апостеріорна)
    float p; // Дисперсія похибки оцінки P
    float q; // Шум процесу Q (обирається емпірично)
    float r; // Шум вимірювання R (з datasheet сенсора)
};

// --- Глобальні змінні -----

// UART2 для MH-Z19B. Параметри: 9600 бод, 8N1
HardwareSerial mhzSerial(2); // Об'єкт UART2

// volatile – змінна змінюється в обробнику переривання (ISR),
// тому компілятор не може кешувати її значення у регістрі
volatile uint32_t pulseCount = 0;

// Стани фільтрів Калмана для кожного з 6 каналів.
// Початковий стан: x=0, p=1, q і r – рекомендовані початкові значення.
KalmanState kRadiation = {0, 1, 0.10f, 0.30f};
KalmanState kCO         = {0, 1, 0.01f, 0.50f};
KalmanState kCO2        = {0, 1, 0.01f, 0.50f};
KalmanState kTVOC       = {0, 1, 0.01f, 0.50f};
KalmanState kNH3        = {0, 1, 0.01f, 0.50f};

```

```

KalmanState kH2S          = {0, 1, 0.01f, 0.50f};

// Характеристики АЦП (заповнюються при ініціалізації)
esp_adc_cal_characteristics_t adcChars;

// --- Обробник переривання ГМ-трубки -----
// IRAM_ATTR – розміщення функції у внутрішній RAM (IRAM).
// Це критично: при виконанні коду з Flash (SPI) є latency ~1 мкс,
// тоді як IRAM виконується миттєво. Для ISR – обов'язкова практика.
void IRAM_ATTR geigerISR() {
    pulseCount++; // Атомарний інкремент лічильника імпульсів
    // Примітка: на ESP32 операції з 32-бітними змінними є атомарними,
    // тому окремий mutex тут не потрібен
}

// --- Рекурсивний фільтр Калмана (скалярний випадок) -----
// Рівняння:
//  $P_k^- = P_{k-1} + Q$  (апріорна дисперсія)
//  $K_k = P_k^- / (P_k^- + R)$  (посилення Калмана)
//  $\hat{x}_k = \hat{x}_{k-1} + K_k \cdot (z_k - \hat{x}_{k-1})$  (оновлення оцінки)
//  $P_k = (1 - K_k) \cdot P_k^-$  (апостеріорна дисперсія)
float kalmanUpdate(KalmanState &s, float z) {
    s.p += s.q; // Крок передбачення: збільшуємо
    невизначеність
    float K = s.p / (s.p + s.r); // Посилення: чим менше R, тим більше
    довіра виміру
    s.x += K * (z - s.x); // Оновлення оцінки: рухаємось до
    виміру z
    s.p *= (1.0f - K); // Зменшуємо невизначеність після
    оновлення
    return s.x; // Повертаємо відфільтроване значення
}

// --- Допоміжна: зчитування АЦП і перетворення в опір Rs -----
// adc1_channel_t channel – канал АЦП1 (GPIO32=CH4, GPIO33=CH5,
GPIO35=CH7)
// float r0 – опір у чистому повітрі для даного сенсора
float readMqRs(adc1_channel_t channel, float r0) {
    // Зчитуємо сирий код АЦП (0-4095 при 12-бітній роздільній
    здатності)
    uint32_t raw = adc1_get_raw(channel);

```

```

// Перетворюємо код у напругу з калібруванням (мВ)
// esp_adc_cal_raw_to_voltage враховує нелінійність АЦП ESP32
uint32_t voltage_mV = esp_adc_cal_raw_to_voltage(raw, &adcChars);
float Vout = voltage_mV / 1000.0f; // Переводимо мВ → В

// Захист від ділення на нуль (при Vout ≈ VCC вся напруга на сенсорі)
if (Vout < 0.01f) return r0;

// Формула дільника напруги:  $R_s = (V_{CC} * R_L / V_{out}) - R_L$ 
// де VCC=5В – живлення MQ, RL=10кОм – навантажувальний резистор
float Rs = (VCC_MQ * RL / Vout) - RL;

// Повертаємо нормоване відношення Rs/R0 – аргумент для
характеристики ppm
return Rs / r0;
}

// Перетворення Rs/R0 у ppm за степеневою характеристикою:  $C = a * (R_s/R_0)^b$ 
// Константи a та b беруться з технічної документації сенсора
float mqToPpm(float ratio, float a, float b) {
    return a * powf(ratio, b);
}

// --- Зчитування всіх сенсорних каналів -----
SensorData readSensors() {
    SensorData d;

    // 1. Радіаційний канал: підрахунок імпульсів ГМ-трубки
    // Скидаємо лічильник, чекаємо GEIGER_WINDOW мс, зчитуємо CPM
    pulseCount = 0;
    uint32_t t0 = millis();
    while (millis() - t0 < GEIGER_WINDOW) { // Чекаємо 30 секунд
        yield(); // Дозволяємо watchdog і background tasks ESP32
    }
    // CPM = pulseCount * (60000 / GEIGER_WINDOW) – нормуємо до 1
    // хвилини
    float cpm = pulseCount * (60000.0f / GEIGER_WINDOW);
    // Перетворення CPM → мкЗв/год за коефіцієнтом SBM-20
    d.radiation_uSvh = cpm / SBM20_CONV_FACTOR;
}

```

```

// 2. CO (MQ-7): константи a=100.0, b=-1.513 (з datasheet MQ-7)
float ratioMQ7 = readMqRs(ADC1_CHANNEL_4, MQ7_R0);
d.co_ppm = mqToPpm(ratioMQ7, 100.0f, -1.513f);

// 3. CO2 (MH-Z19B): запит через UART-команду 0x86
// Протокол: надсилаємо 9 байт команди, отримуємо 9 байт відповіді
uint8_t cmd[9] = {0xFF, 0x01, 0x86, 0, 0, 0, 0, 0, 0x79};
mhzSerial.write(cmd, 9); // Надсилаємо команду "отримати CO2"
uint8_t resp[9]; memset(resp, 0, 9);
if (mhzSerial.readBytes(resp, 9) == 9 && resp[0]==0xFF &&
resp[1]==0x86) {
    // Байти 2 і 3 містять значення CO2 у ppm (big-endian uint16)
    d.co2_ppm = resp[2] * 256.0f + resp[3];
} else {
    d.co2_ppm = -1.0f; // -1 сигналізує про помилку читання
}

// 4. NH3 (MQ-137): константи a=102.6, b=-2.473 (з datasheet MQ-
137)
float ratioMQ137 = readMqRs(ADC1_CHANNEL_7, MQ137_R0);
d.nh3_ppm = mqToPpm(ratioMQ137, 102.6f, -2.473f);

// 5. H2S (MQ-136): константи a=36.7, b=-3.536 (з datasheet MQ-136)
float ratioMQ136 = readMqRs(ADC1_CHANNEL_5, MQ136_R0);
d.h2s_ppm = mqToPpm(ratioMQ136, 36.7f, -3.536f);

// 6. TVOC (MICS-6814 через I2C): читаємо канал CO як проху TVOC
// (для повноцінного TVOC потрібен SGP30; тут спрощена реалізація)
Wire.beginTransmission(MICS_ADDR);
Wire.write(0x04); // Регістр 0x04 – дані каналу TVOC
Wire.endTransmission(false);
Wire.requestFrom(MICS_ADDR, 2);
if (Wire.available() >= 2) {
    uint16_t raw16 = Wire.read() << 8 | Wire.read();
    // Масштабуємо сирий код у ppb (лінійна апроксимація для TVOC)
    d.tvoc_ppb = raw16 * 0.5f;
} else {
    d.tvoc_ppb = -1.0f;
}

```

```

        return d;
    }

    // --- Застосування фільтра Калмана до всіх каналів -----
    // Приймає сирі дані, повертає відфільтровану структуру SensorData
    SensorData applyKalman(SensorData raw) {
        SensorData f;
        // Кожен канал фільтрується незалежно зі своїм станом KalmanState
        f.radiation_uSvh = kalmanUpdate(kRadiation, raw.radiation_uSvh);
        f.co_ppm          = kalmanUpdate(kCO,          raw.co_ppm);
        f.co2_ppm         = kalmanUpdate(kCO2,         raw.co2_ppm);
        f.tvoc_ppb        = kalmanUpdate(kTVOC,        raw.tvoc_ppb);
        f.nh3_ppm         = kalmanUpdate(kNH3,         raw.nh3_ppm);
        f.h2s_ppm         = kalmanUpdate(kH2S,         raw.h2s_ppm);
        return f;
    }

    // --- Розрахунок Composite Hazard Index (Data Fusion) -----
    // CHI =  $\sum (\omega_i * n_i)$ , де  $n_i = c_i / \text{threshold}_i$ 
    // CHI < 0.5 → норма (green)
    // 0.5 ≤ CHI < 1.0 → попередження (yellow)
    // CHI ≥ 1.0 → тривога (red – перевищення хоча б одного порогу)
    float calcCHI(SensorData &d) {
        //  $n_i$  – нормований показник. Якщо значення недійсне (-1),  $n_i = 0$ 
        float n_rad = d.radiation_uSvh / THR_RADIATION;
        float n_co  = d.co_ppm / THR_CO;
        float n_co2 = (d.co2_ppm > 0) ? d.co2_ppm / THR_CO2 : 0.0f;
        float n_tvoc = (d.tvoc_ppb > 0) ? d.tvoc_ppb / THR_TVOC : 0.0f;
        float n_nh3  = d.nh3_ppm / THR_NH3;
        float n_h2s  = d.h2s_ppm / THR_H2S;

        // Зважене підсумовування ( $\sum \omega_i = 0.35+0.20+0.05+0.10+0.10+0.20 = 1.00$ )
        return W_RADIATION*n_rad + W_CO*n_co + W_CO2*n_co2
            + W_TVOC*n_tvoc + W_NH3*n_nh3 + W_H2S*n_h2s;
    }

    // --- Формування байта бітових прапорців тривоги -----
    // bit 0 (0x01) – гамма понад поріг
    // bit 1 (0x02) – CO понад поріг

```

```

// bit 2 (0x04) – CO2 понад поріг
// bit 3 (0x08) – TVOC понад поріг
// bit 4 (0x10) – NH3 понад поріг
// bit 5 (0x20) – H2S понад поріг
uint8_t buildAlarmFlags(SensorData &d) {
    uint8_t flags = 0;
    if (d.radiation_uSvh >= THR_RADIATION) flags |= 0x01;
    if (d.co_ppm >= THR_CO) flags |= 0x02;
    if (d.co2_ppm >= THR_CO2) flags |= 0x04;
    if (d.tvoc_ppb >= THR_TVOC) flags |= 0x08;
    if (d.nh3_ppm >= THR_NH3) flags |= 0x10;
    if (d.h2s_ppm >= THR_H2S) flags |= 0x20;
    return flags;
}

// --- Формування та відправлення пакета LoRa -----
// Структура пакета (18 байт):
// [0-1] node_id      uint16 – ідентифікатор вузла
// [2-3] timestamp    uint16 – секунди від епохи (mod 65536)
// [4-5] radiation     uint16 – СРМ * 10 (дробова частина збережена)
// [6-7] co_ppm        uint16 – CO * 100 (2 знаки після коми)
// [8-9] co2_ppm       uint16 – CO2 (ціле число ppm)
// [10-11] nh3_ppm     uint16 – NH3 * 100
// [12-13] h2s_ppm     uint16 – H2S * 1000
// [14-15] tvoc_ppb    uint16 – TVOC ppb (ціле)
// [16] chi            uint8  – CHI * 100 (0-255)
// [17] flags          uint8  – бітові прапорці тривоги
bool sendLoRaPacket(SensorData &d, float chi) {
    // Отримуємо поточну мітку часу. millis() / 1000 → секунди від
запуску
    uint16_t ts = (uint16_t)(millis() / 1000);
    uint8_t flags = buildAlarmFlags(d);

    LoRa.beginPacket(); // Починаємо формування пакета

    // Допоміжний лямбда-вираз: запис uint16 у LoRa (big-endian)
    auto writeU16 = [](uint16_t v){
        LoRa.write((v>>8)&0xFF);
        LoRa.write(v&0xFF);
    };

```

```

writeU16(NODE_ID); // байти 0-1
writeU16(ts); // байти 2-3
writeU16((uint16_t)(d.radiation_uSvh*153.8f*10)); // байти 4-5:
CPM*10
writeU16((uint16_t)(d.co_ppm * 100)); // байти 6-7
writeU16((uint16_t)(d.co2_ppm)); // байти 8-9
writeU16((uint16_t)(d.nh3_ppm * 100)); // байти 10-11
writeU16((uint16_t)(d.h2s_ppm * 1000)); // байти 12-13
writeU16((uint16_t)(d.tvoc_ppb)); // байти 14-15
LoRa.write((uint8_t)(chi * 100)); // байт 16: CHI*100
LoRa.write(flags); // байт 17

// endPacket(true) – синхронний режим: чекаємо завершення TX
// endPacket(false) – асинхронний (повертає керування одразу)
return LoRa.endPacket(true); // true = sync TX
}

// --- Вивід налагоджувальних даних у Serial Monitor -----
void printDebug(SensorData &raw, SensorData &flt, float chi) {
    Serial.println("=== Sensor Reading ===");
    // Serial.printf – форматований вивід (аналог printf зі stdlib C)
    Serial.printf("Radiation: raw=%.3f filt=%.3f uSv/h\n",
        raw.radiation_uSvh, flt.radiation_uSvh);
    Serial.printf("CO:          raw=%.1f filt=%.1f ppm\n",
        raw.co_ppm, flt.co_ppm);
    Serial.printf("CO2:          raw=%.0f filt=%.0f ppm\n",
        raw.co2_ppm, flt.co2_ppm);
    Serial.printf("NH3:          raw=%.2f filt=%.2f ppm\n",
        raw.nh3_ppm, flt.nh3_ppm);
    Serial.printf("H2S:          raw=%.3f filt=%.3f ppm\n",
        raw.h2s_ppm, flt.h2s_ppm);
    Serial.printf("TVOC:          raw=%.0f filt=%.0f ppb\n",
        raw.tvoc_ppb, flt.tvoc_ppb);
    // Вивід CHI для Плотера – значення через кому = кілька ліній на
    графіку
    Serial.printf("CHI: %.3f\n", chi);
}

// =====
// setup() – виконується ОДИН РАЗ після увімкнення або reset
// =====

```

```

void setup() {
    // Послідовний порт для налагодження: 115200 бод
    // Монітор (Ctrl+Shift+M) або Плотер (Ctrl+Shift+L) в Arduino IDE
    Serial.begin(115200);
    delay(500); // Невелика пауза для стабілізації USB-UART
    Serial.println("Node starting...");

    // Калібрування АЦП: MODE_11dB – затухання 11 дБ,
    // дозволяє вимірювати напругу до ~3.3 В (без затухання – до ~1.1
В)

    adc1_config_width(ADC_WIDTH_BIT_12);
    adc1_config_channel_atten(ADC1_CHANNEL_4, ADC_ATTEN_DB_11); // MQ-
7
    adc1_config_channel_atten(ADC1_CHANNEL_5, ADC_ATTEN_DB_11); // MQ-
136
    adc1_config_channel_atten(ADC1_CHANNEL_7, ADC_ATTEN_DB_11); // MQ-
137

    // Характеризуємо АЦП: VREF=3300 мВ, схема двоточкового
калібрування
    esp_adc_cal_characterize(ADC_UNIT_1, ADC_ATTEN_DB_11,
        ADC_WIDTH_BIT_12, 3300, &adcChars);

    // I²C для MICS-6814
    Wire.begin(I2C_SDA, I2C_SCL);

    // UART2 для MH-Z19B: 9600 бод, формат 8N1
    mhzSerial.begin(9600, SERIAL_8N1, MHZ_RX, MHZ_TX);

    // Переривання від ГМ-трубки: реагуємо на спадний фронт
    // FALLING – момент коли TTL-сигнал переходить з HIGH в LOW
(імпульс)
    pinMode(GEIGER_PIN, INPUT);
    attachInterrupt(digitalPinToInterrupt(GEIGER_PIN), geigerISR,
FALLING);

    // Ініціалізація LoRa SX1262
    SPI.begin(LORA_SCK, LORA_MISO, LORA_MOSI, LORA_SS);
    LoRa.setPins(LORA_SS, LORA_RST, LORA_DIO0);
    if (!LoRa.begin(LORA_FREQUENCY)) {
        Serial.println("LoRa init FAILED"); while(1); // Зупинка
    }
}

```

```

// Встановлюємо параметри радіозв'язку
LoRa.setSpreadingFactor(LORA_SF); // SF10 → чутл. -137 дБм
LoRa.setSignalBandwidth(LORA_BW); // 125 кГц
LoRa.setCodingRate4(LORA_CR); // CR 4/5
LoRa.setTxPower(20); // 20 дБм — максимум для SX1262
Serial.println("LoRa OK");

// Прогрів хімічних MQ-сенсорів (30 секунд).
// За цей час нагрівальний елемент виходить на робочу температуру
(~300°C)
// Паралельно відбувається накопичення CPM у geigerISR
Serial.println("Warming up sensors...");
delay(WARMUP_MS);
Serial.println("Warmup done.");
}

// =====
// loop() — основний цикл. Виконується ОДИН РАЗ,
// завершується командою deep sleep. "Цикл" реалізований через
// пробудження таймера esp_sleep_enable_timer_wakeup()
// =====
void loop() {
    // Зчитуємо сирі дані з усіх сенсорних каналів
    SensorData raw = readSensors();

    // Фільтруємо кожен канал фільтром Калмана
    SensorData filtered = applyKalman(raw);

    // Розраховуємо Composite Hazard Index
    float chi = calcCHI(filtered);

    // Виводимо дані у Serial Monitor / Plotter для налагодження
    printDebug(raw, filtered, chi);

    // Надсилаємо пакет через LoRa
    bool sent = sendLoRaPacket(filtered, chi);
    Serial.printf("LoRa TX: %s\n", sent ? "OK" : "FAIL");

    // Визначаємо тривалість сну на основі CHI
    uint64_t sleepUs;

```

```

        if      (chi >= 1.0f) sleepUs = 0;                // Тривога: негайне
повторення
        else if (chi >= 0.5f) sleepUs = SLEEP_WARN;      // Попередження: 60 с
        else          sleepUs = SLEEP_NORMAL; // Норма: 300 с

        if (sleepUs > 0) {
            Serial.printf("Deep sleep %llu s\n", sleepUs/1000000ULL);
            Serial.flush(); // Чекаємо завершення виводу перед сном
            // Налаштовуємо джерело пробудження: таймер на sleepUs
мікросекунд
            esp_sleep_enable_timer_wakeup(sleepUs);
            // Переводимо ESP32 у Deep Sleep: GPIO не активні, споживання
~10-150 мкА
            // Після пробудження ESP32 стартує з початку – знову
виконується setup()
            esp_deep_sleep_start();
        }
        // Якщо sleepUs == 0 (тривога) – loop() виконається одразу ще раз
    }

    // =====
    // Кінець файлу sensor_node.ino

```