Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Галузь знань _____12 – Інформаційні технології_____

Спеціальність _____123 –Комп'ютерна інженерія _____

на тему «Кіберфізична система контролю параметрів життєзабезпечення із забезпеченням приватності»

КвРКІП. 2301152.23.01.33 ПЗ

Виконав: студент 2 курсу, група КІ2м-23-1 Олександр ЗАБАВСЬКИЙ
Підпис Ім'я, прізвище

Керівник д.т.н., професор Василь ЯЦКІВ
Науковий ступінь, вчене звання Підпис Ім'я, прізвище

До захисту допускаю:
Зав. кафедри КІІС, доктор філософії, доцент
Ольга ПАВЛОВА
19 05 2025 р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Комп'ютерної інженерії та інформаційних систем
Освітній рівень магістр
Галузь знань 12 Інформаційні технології
Спеціальність 123 Комп'ютерна інженерія
Освітня програма освітньо-наукова програма «комп'ютерна інженерія та програмування»

ЗАТВЕРДЖУЮ
Зав. кафедри Ольга ПАВЛОВА
" 01 " 09 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Олександр ЗАБАВСЬКИЙ
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система контролю параметрів життєзабезпечення із забезпеченням приватності

Керівник проекту (роботи) Василь ЯЦКІВ, д.т.н., професор
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2025 №8

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

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

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КІІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІІС		

7. Дата видачі завдання «_01_»_____09_____2024р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	01.09.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2024	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2024	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2024	виконано
5	Робота над науковою статтею	01.02.2025	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2025	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2025	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	виконано
9	Попередній захист ДРМ	29.04.2025	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2025	

Студент _____ Олександр ЗАБАВСЬКИЙ
Підпис Ім'я, прізвище

Керівник роботи _____ Василь ЯЦКІВ
Підпис Ім'я, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Кіберфізична система контролю параметрів життєзабезпечення із забезпеченням приватності.

Автор роботи: Олександр ЗАБАВСЬКИЙ

Керівник роботи: Василь ЯЦКІВ

Пояснювальна записка: 75 с., 12 рис., 12 табл., 6 дод., 38 джерел.

КІБЕРФІЗИЧНА СИСТЕМА, КОНТРОЛЬ ПАРАМЕТРІВ ЖИТТЄЗАБЕЗПЕЧЕННЯ, ЗАБЕЗПЕЧЕННЯ ПРИВАТНОСТІ, СЕНСОРНІ ТЕХНОЛОГІЇ, ШИФРУВАННЯ ДАНИХ, БЕЗПЕКА ІНФОРМАЦІЇ, IOT, МОНІТОРИНГ ДАНИХ.

Об’єктом дослідження є процес моніторингу та контролю показників життєзабезпечення (температури, вологості, рівня газів та інших) за допомогою кіберфізичної системи, що об'єднує датчики й програмну частину в єдину керуючу інфраструктуру.

Предметом дослідження є методи проєктування, розробки та впровадження кіберфізичних систем, що забезпечують постійний моніторинг параметрів життєзабезпечення з урахуванням вимог до конфіденційності та захисту даних під час їх збору, обробки та передавання.

Метою кваліфікаційної роботи магістра є розробка кіберфізичної системи контролю параметрів життєзабезпечення із вбудованими механізмами захисту приватності, що дозволяє забезпечити безперервний моніторинг життєво важливих фізичних показників, захист зібраних даних та підвищити безпеку їх обробки й передавання в умовах реального або змодельованого середовища.

Для розв’язання поставлених задач використовувалися методи: Аналіз і синтез застосовувалися для вивчення існуючих підходів до побудови кіберфізичних систем і визначення найбільш ефективних методів забезпечення приватності. Порівняльний аналіз дав змогу оцінити ефективність алгоритмів шифрування та обґрунтувати вибір оптимальної архітектури системи. Методи математичного моделювання були використані для формалізації процесів збору,

обробки та передавання даних. Агентне моделювання дозволило описати взаємодію компонентів системи в умовах змінного середовища. Для перевірки працездатності та ефективності розробленої системи застосовувалися експериментальні методи, які включали тестування у лабораторних умовах, аналіз точності, стійкості та захищеності даних. Крім того, для реалізації програмної частини та візуалізації результатів використовувалися методи програмної реалізації та симуляції із залученням мови Python, відповідних бібліотек і засобів візуалізації. Комплексне використання цих методів забезпечило системний і обґрунтований підхід до створення та аналізу кіберфізичної системи контролю параметрів життєзабезпечення.

Наукова новизна отриманих результатів полягає в тому, що в рамках дослідження було вперше інтегровано механізми забезпечення приватності безпосередньо в архітектуру кіберфізичної системи контролю параметрів життєзабезпечення. Це дало змогу мінімізувати ризики витоку даних на всіх етапах — від збору та обробки до передавання. Особливістю розробленої системи стало поєднання класичних IoT-засобів із сучасними криптографічними методами, такими як AES та RSA, що дозволяє забезпечити конфіденційність у розподіленому середовищі. Крім того, набули подальшого розвитку методи локального зберігання й аналізу даних з метою захисту персональної інформації, а також інформаційна технологія інтеграції систем захисту в програмну логіку КФС. Реалізовані підходи суттєво підвищили рівень безпеки, автономності та гнучкості системи, що відкриває перспективи для її застосування у сферах медичного моніторингу, екологічного контролю та промислових IoT-рішень. набув подальшого розвитку метод;

Набула подальшого розвитку інформаційна технологія захисту персональних даних у складі кіберфізичних систем моніторингу. Вперше було інтегровано прості, але надійні засоби криптографії безпосередньо на рівні прототипу для локального збору, обробки та передачі даних від сенсорних пристроїв. Це дозволило не лише захистити інформацію на всіх етапах її обробки, а й забезпечити захищений доступ користувачів до даних. Запропонована технологія дозволяє ефективно поєднати

класичні IoT-рішення з сучасними засобами криптографії, що істотно підвищує рівень безпеки і відкриває можливості для подальшого застосування системи в медичних, екологічних та промислових умовах.

На основі проведених досліджень була створена архітектура кіберфізичної системи контролю параметрів життєзабезпечення, яка враховує сучасні вимоги до модульності, інтероперабельності та захисту приватності даних. У межах розробленої архітектури визначено ключові компоненти: сенсорний модуль для збору параметрів, мікроконтролерний модуль для попередньої обробки даних, модуль обробки та аналізу інформації, система локального збереження на основі бази даних SQLite, графічний інтерфейс користувача для візуалізації параметрів і оповіщення про критичні ситуації, а також модуль захисту інформації з вбудованими засобами шифрування й автентифікації. Запропонована архітектура дозволяє ефективно масштабувати систему, інтегрувати нові сенсори та підвищувати рівень безпеки, що підтверджено під час реалізації функціонального прототипу та проведення експериментальних досліджень.

Практична значимість отриманих результатів полягає у створенні функціонального прототипу кіберфізичної системи контролю параметрів життєзабезпечення з вбудованими засобами захисту даних. Реалізація системи із застосуванням сенсорних модулів, мови програмування Python, протоколу MQTT та алгоритмів шифрування (AES, RSA) дозволила забезпечити конфіденційність і надійність обробки та передавання інформації. Розроблене рішення може бути використане в медичних, побутових і промислових умовах для безпечного моніторингу важливих фізичних параметрів. Крім того, система має освітню та наукову цінність — вона може слугувати прикладом для вивчення методів побудови кіберфізичних систем, організації захисту інформації та роботи з IoT-компонентами. Написана програма, створена база даних і реалізовані засоби візуалізації можуть стати основою для подальших досліджень і розробок у галузі захищених систем моніторингу.

У першому розділі було проведено детальний аналіз існуючих підходів до побудови кіберфізичних систем для контролю параметрів життєзабезпечення. Визначено основні сфери застосування КФС, розглянуто сучасні сенсорні системи та охарактеризовано їхні технічні особливості. Окрему увагу приділено питанням забезпечення приватності в середовищах IoT та CPS, де проаналізовано загрози, визначено рівні захисту та запропоновано підходи до впровадження безпеки на етапі проєктування.

У другому розділі здійснено проєктування кіберфізичної системи контролю параметрів життєзабезпечення. Визначено функціональні й нефункціональні вимоги до системи, розроблено її архітектуру, а також обґрунтовано вибір технічних і програмних засобів. У цьому ж розділі реалізовано механізми захисту приватності, такі як шифрування та автентифікація, що дозволило забезпечити конфіденційність даних на всіх етапах обробки.

У третьому розділі проведено реалізацію програмної системи. Створено багаторівневу структуру програмного забезпечення, включаючи модулі збору, обробки, візуалізації та збереження даних. Реалізовано інтеграцію засобів криптографії, графічного інтерфейсу та протоколу MQTT для передачі даних. Розроблена система підтримує виявлення відхилень та реагування на критичні ситуації.

У четвертому розділі здійснено експериментальне дослідження функціонування системи. Проведено тести на точність, стабільність, затримки в обробці даних та ефективність механізмів захисту. Підтверджено працездатність розробленої системи та визначено перспективи її вдосконалення й масштабування.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ 5

ВСТУП 6

1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ПОБУДОВИ КІБЕРФІЗИЧНИХ СИСТЕМ ЗАХИСТУ ТА КОНТРОЛЮ 10

1.1 Сутність, структура, сфери застосування кіберфізичних систем 10

1.2 Огляд сучасних сенсорних систем для моніторингу параметрів життєзабезпечення 16

1.3 Забезпечення приватності в IoT- та CPS-середовищах 24

1.4 Постановка задачі 29

2 ПРОЄКТУВАННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ КОНТРОЛЮ ПАРАМЕТРІВ ЖИТТЄЗАБЕЗПЕЧЕННЯ 33

2.1 Визначення функціональних вимог до системи 33

2.2 Визначення нефункціональних вимог до системи 35

2.3 Архітектура системи та взаємодія компонентів 37

2.4 Обґрунтування вибору засобів реалізації 47

2.5 Механізми захисту приватності в обраній архітектурі 50

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ 54

3.1 Налаштування середовища програмування та бібліотек 54

3.2 Реалізація програмного коду 60

3.3 Вбудовані засоби захисту даних у програмному рішенні 64

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ І ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ 69

4.1 Організація тестового середовища 69

4.2 Методика вимірювання точності та стабільності 71

4.3 Оцінка ефективності захисту приватності 74

4.4 Аналіз результатів та можливості вдосконалення 75

ВИСНОВКИ 79

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 81

ДОДАТОК А Лістинг програмного забезпечення 85

ДОДАТОК Б Компонентна діаграма системи моніторингу життєвих параметрів 86

ДОДАТОК В Логування показників датчиків 87

ДОДАТОК Г Інтерфейс додатку 88

ДОДАТОК Д Інтерфейс додатку (графіки) 89

ДОДАТОК Є Сертифікат "Перспективні мережні та комп'ютерні технології" (ПерСиК 2025) 88

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

КФС – кіберфізична система

IoT (IoT) – Інтернет речей (Internet of Things)

ЦП – центральний процесор

RAM – оперативна пам’ять (Random Access Memory)

AES – алгоритм симетричного шифрування Advanced Encryption Standard

RSA – алгоритм асиметричного шифрування, названий за іменами авторів (Rivest–Shamir–Adleman)

GPIO – інтерфейс загального призначення для введення/виведення (General Purpose Input/Output)

DHT – цифровий датчик температури і вологості (Digital Humidity and Temperature sensor)

UART – універсальний асинхронний приймач-передавач (Universal Asynchronous Receiver-Transmitter)

JSON – формат обміну даними JavaScript Object Notation

API – програмний інтерфейс взаємодії (Application Programming Interface)

GUI – графічний інтерфейс користувача (Graphical User Interface)

Python – високорівнева мова програмування, що використовується для реалізації логіки системи

Tkinter – стандартна бібліотека Python для створення графічного інтерфейсу

SQLite – вбудована реляційна база даних

ВСТУП

У сучасному світі стрімкий розвиток інформаційних технологій та мікроелектроніки зумовив широке впровадження кіберфізичних систем (КФС) у різні сфери людської діяльності. Ці системи, що об'єднують обчислювальні алгоритми з фізичними процесами, знаходять застосування в:

- охороні здоров'я;
- промисловості;
- енергетиці;
- транспорті;
- будівництві;
- сільському господарстві;
- системах "розумного міста";
- обороні.

Зокрема, в медичній сфері КФС сприяють покращенню моніторингу пацієнтів, ефективному управлінню медикаментами та зниженню витрат шляхом автоматизації рутинних завдань [1]. У транспорті ці системи використовуються для управління автономними транспортними засобами, оптимізації дорожнього руху та підвищення безпеки. У сільському господарстві КФС забезпечують інтелектуальний контроль вологості ґрунту, температурного режиму та рівня освітлення в теплицях, сприяючи підвищенню врожайності [2]. У межах концепції «розумного міста» вони інтегруються в системи освітлення, відеоспостереження, управління трафіком і споживанням енергії, створюючи комфортне та безпечне середовище для мешканців.

Однак інтеграція фізичних об'єктів з цифровими технологіями створює нові виклики у сфері безпеки та приватності. Передача чутливих даних через відкриті канали зв'язку або використання публічних хмарних сервісів підвищує ризик несанкціонованого доступу та витоку інформації. Тому забезпечення конфіденційності та цілісності даних є критично важливим аспектом при розробці та впровадженні КФС, особливо в контексті життєзабезпечення [2].

Отже, актуальність роботи полягає у створенні кіберфізичної системи для контролю параметрів життєзабезпечення з вбудованими механізмами захисту приватності, що дозволить підвищити надійність і безпечність таких рішень.

Метою кваліфікаційної роботи є розробка кіберфізичної системи контролю параметрів життєзабезпечення із забезпеченням приватності. Поставлена мета досягається шляхом вирішення наступних завдань дипломної роботи:

- проаналізувати існуючі технічні рішення в галузі КФС для моніторингу параметрів життєзабезпечення;
- обґрунтувати вибір архітектури системи, засобів збору, обробки та захисту даних;
- реалізувати програмну модель КФС з елементами забезпечення приватності;
- провести експериментальне дослідження функціонування системи в умовах реального або змодельованого середовища.

Об’єктом дослідження є процес моніторингу та контролю параметрів життєзабезпечення, таких як температура, вологість, вміст газів та інші фізичні показники, за допомогою кіберфізичних систем, які інтегрують апаратне забезпечення з програмними модулями в єдину інформаційно-керуючу інфраструктуру.

Предметом дослідження є методи проєктування, розробки та реалізації кіберфізичних систем, що забезпечують безперервний контроль параметрів життєзабезпечення з урахуванням вимог до приватності та безпеки даних, які передаються або обробляються в системі.

Наукова новизна отриманих результатів полягає у впровадженні засобів забезпечення приватності безпосередньо в архітектуру КФС, що дозволяє мінімізувати ризики витоку даних під час зчитування, обробки та передавання параметрів життєзабезпечення. Головною особливістю розробленої системи є поєднання класичних IoT-засобів із сучасними криптографічними механізмами для забезпечення конфіденційності даних у розподіленому середовищі.

Практична значимість роботи полягає у створенні функціонального прототипу кіберфізичної системи з елементами захисту даних, реалізованого засобами мови програмування Python із використанням сенсорних модулів (наприклад, температури та вологості) і алгоритмів шифрування (AES, RSA або аналогічних) для збереження конфіденційності зібраної інформації. Запропонована система може бути використана в медичних, побутових або промислових умовах для забезпечення безпечного моніторингу життєво важливих показників, а також використовуватися як демонстраційний приклад у сфері освітніх чи наукових досліджень. Написання подібної роботи може слугувати елементом портфоліо, оскільки демонструє рівень володіння сучасними мовами програмування, знання основ побудови кіберфізичних систем, навички роботи з апаратним забезпеченням та реалізації криптографічного захисту даних.

У процесі дослідження та реалізації КФС контролю параметрів життєзабезпечення було застосовано комплекс наукових методів, що забезпечили системний підхід до вирішення поставлених завдань. Використання відповідних методів дозволило обґрунтувати вибір архітектурних рішень, алгоритмів обробки даних і засобів забезпечення приватності, а також провести моделювання функціонування системи в умовах, наближених до реальних. Для досягнення цілей дослідження були використані такі наукові методи [3]:

- аналіз та синтез — для вивчення існуючих підходів до побудови кіберфізичних систем та засобів забезпечення приватності;
- порівняльний аналіз — для оцінки ефективності алгоритмів шифрування та архітектур КФС;
- методи математичного моделювання — для формалізації процесів збору, обробки та передавання даних у кіберфізичній системі;
- агентне моделювання — з метою опису взаємодії компонентів системи у змінних середовищах.

Для реалізації прототипу КФС контролю параметрів життєзабезпечення з вбудованими механізмами забезпечення приватності було застосовано набір

сучасних програмних і апаратних засобів. Вибір інструментів обумовлювався вимогами до:

- гнучкості;
- надійності;
- масштабованості системи;
- необхідності реалізації криптографічного захисту переданих даних.

Технічна реалізація передбачала створення програмної логіки збору, обробки, візуалізації й зберігання інформації, а також організацію ефективної взаємодії між апаратними та програмними компонентами. Для цього були використані такі технічні засоби та методи:

- мова програмування Python — як основний інструмент розробки логіки системи;
- бібліотеки для роботи з сенсорними модулями та криптографічні бібліотеки;
- тестування взаємодії апаратних і програмних модулів;
- засоби візуалізації даних (matplotlib, Tkinter);
- модулі логування та збереження даних;
- тестування з навмисним внесенням збоїв - для перевірки поведінки системи у граничних умовах.

За темою кваліфікаційної роботи підготовлено публікацію у фаховому виданні.

Результати дослідження апробовано на 16-й міжнародній студентській науково-технічній конференції "Перспективні мережні та комп'ютерні технології" (ПерСиК 2025), що проходила у Харкові. За підсумками підготовлено та опубліковано тези на тему «Кіберфізична система контролю параметрів життєзабезпечення із забезпеченням приватності», в яких висвітлено архітектуру системи, засоби захисту даних і результати тестування. Сертифікат, що підтверджує апробацію роботи на конференції, наведено в (Додатку Є)

1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ПОБУДОВИ КІБЕРФІЗИЧНИХ СИСТЕМ ЗАХИСТУ ТА КОНТРОЛЮ

1.1 Сутність, структура, сфери застосування кіберфізичних систем

Кіберфізичні системи (КФС, англ. Cyber-Physical Systems, CPS) — це високотехнологічні комплекси, які інтегрують фізичні компоненти, обчислювальні засоби та мережеву інфраструктуру для моніторингу та керування об’єктами у фізичному середовищі в режимі реального часу. У таких системах відбувається безперервний обмін даними між сенсорними пристроями, виконавчими механізмами та програмною логікою, що забезпечує адаптивну поведінку системи [4].

Приклади КФС наведено в таблиці 1.1 з зазначенням сфери застосування та переліком основних модулів.

Таблиця 1.1 - Приклади застосування КФС у різних сферах

Сфера застосування	Приклад КФС	Сенсорні модулі	Керуючі модулі / дії	Призначення
Побут	Розумний будинок	Температури, вологості, руху, освітлення	Регулювання опалення, освітлення, охорони	Комфорт, енергоефективність, безпека
Медицина	Система дистанційного моніторингу пацієнтів	Серцевого ритму, тиску, рівня кисню	Передача даних лікарю, сповіщення про критичні зміни	Підвищення якості догляду, оперативне реагування

Продовження таблиці 1.1

Сфера застосування	Приклад КФС	Сенсорні модулі	Керуючі модулі / дії	Призначення
Транспорт	Автономні транспортні засоби	Камери, радар, GPS, акселерометри	Рух за маршрутом, уникнення перешкод, адаптивне керування	Безпека, автономність, оптимізація маршрутів
Сільське господарство	Розумна теплиця	Температури, вологості, освітлення, вологість ґрунту	Автоматичний полив, вентиляція, підсвітка	Збільшення врожайності, оптимізація ресурсів
Промисловість	Система контролю виробництва	Температури, тиску, рівня вібрацій	Регулювання технологічних процесів, аварійне зупинення	Якість продукції, безпека персоналу
Транспорт	Автономні транспортні засоби	Камери, радар, GPS, акселерометри	Рух за маршрутом, уникнення перешкод, адаптивне керування	Безпека, автономність, оптимізація маршрутів
Енергетика	Інтелектуальні електромережі (Smart Grid)	Споживання енергії, напруга, струм	Балансування навантаження, виявлення несправностей	Ефективне управління енергією, зниження втрат

Кінець таблиці 1.1

Сфера застосування	Приклад КФС	Сенсорні модулі	Керуючі модулі / дії	Призначення
Міська інфраструктура	«Розумне місто»	Рівень шуму, забруднення повітря, трафік, освітлення	Керування світлофорами, освітленням, сміттєзбиранням	Підвищення якості життя мешканців, зниження витрат

Типова архітектура КФС включає чотири ключові компоненти:

- сенсори, які здійснюють вимірювання фізичних параметрів;
- виконавчі пристрої, що забезпечують зворотний зв'язок із середовищем;
- обчислювальні модулі, які аналізують інформацію та приймають рішення;
- мережеві протоколи та інтерфейси, що здійснюють передачу даних.

Сенсори є первинним компонентом КФС, відповідальним за вимірювання фізичних параметрів навколишнього середовища або об'єкта контролю. Вони можуть фіксувати температуру, вологість, тиск, рівень шуму, серцевий ритм та інші показники. Зібрані дані передаються до обчислювальних модулів для подальшої обробки. Вибір сенсорів залежить від конкретної галузі застосування системи, її точності та швидкості реакції.

Виконавчі пристрої (актуатори) реалізують зворотний зв’язок між системою та фізичним світом. Вони виконують команди, сформовані обчислювальними модулями, наприклад: вмикають обігрівачі, відкривають клапани, запускають двигуни або активують світлову та звукову сигналізацію. Їх функція — реалізувати реакцію системи на зміну середовища або задані сценарії роботи.

Обчислювальні модулі виступають центральним елементом системи, який здійснює збір, обробку та аналіз даних, отриманих від сенсорів. Вони реалізують алгоритми прийняття рішень, штучного інтелекту, шифрування, фільтрації або статистичної обробки. Саме тут формується логіка дій, що визначає подальшу поведінку системи у відповідь на зміну параметрів.

Мережеві протоколи та інтерфейси забезпечують ефективну передачу даних між компонентами системи. Це можуть бути як дротові (Ethernet, RS-485), так і бездротові (Wi-Fi, Bluetooth, Zigbee, LoRa, MQTT) рішення. Протоколи визначають формат, частоту та спосіб комунікації, а також часто включають засоби забезпечення безпеки передавання — автентифікацію, шифрування та контроль цілісності.

Отже, структурно КФС є багаторівневими системами, в яких фізичне середовище (сенсори, об’єкти контролю) взаємодіє з кібернетичним середовищем (обчислювальні модулі, бази даних, засоби захисту), а комунікаційний рівень забезпечує безперервний обмін інформацією. Таке представлення дозволяє наочно відобразити архітектуру системи та виділити окремі шари її функціонування.

Для наочності на рисунку 1.1 подано адаптовану схему трирівневої структури КФС відповідно до концептуальної моделі з [5], де виділено фізичний, комунікаційний і кібернетичний рівні взаємодії.

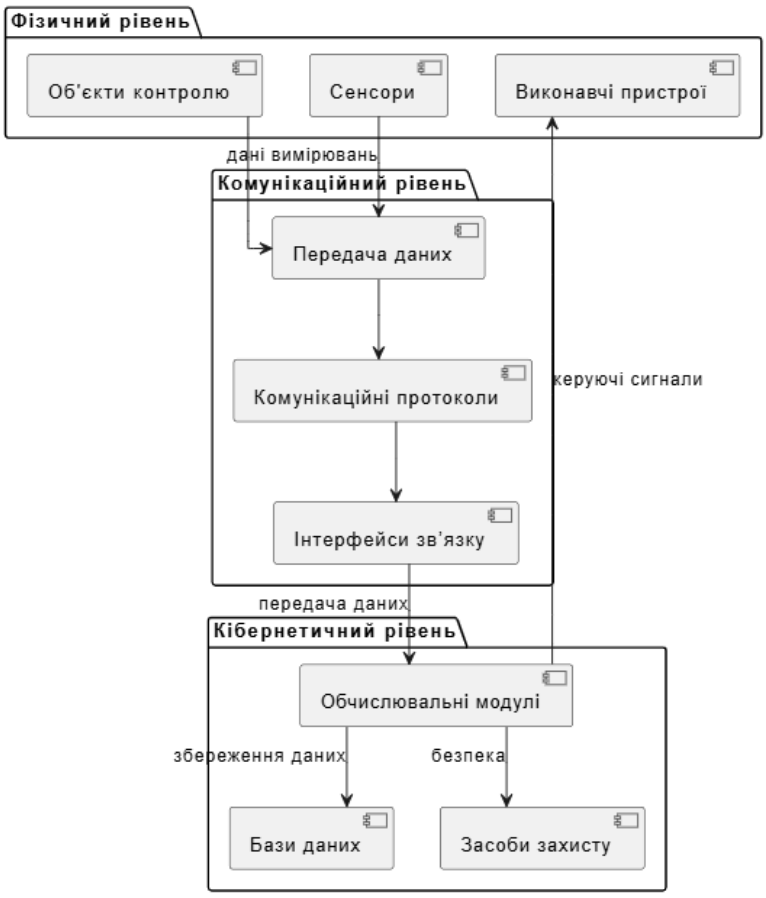


Рисунок 1.1 – Трирівнева структура КФС з урахуванням кібернетичної безпеки

Дана схема ілюструє розподіл компонентів КФС на:

- фізичне середовище, що включає сенсорні пристрої для збору інформації;
- комунікаційне середовище, де відбувається передача даних через бездротові протоколи (Wi-Fi, Bluetooth, MQTT);
- кібернетичне середовище, яке відповідає за зберігання, обробку інформації та забезпечення приватності (включаючи сервери, бази даних і засоби шифрування).

КФС мають високу гнучкість, масштабованість та здатність до самостійного прийняття рішень, що робить їх важливим елементом у реалізації концепції Інтернету речей (IoT), Індустрії 4.0, Smart Grid, цифрового сільського господарства тощо.

Завдяки своїй гнучкості, масштабованості та здатності до автономного функціонування, КФС застосовуються у найрізноманітніших сферах — від охорони здоров’я до промислового виробництва. Зведену характеристику сфер застосування представлено в таблиці 1.2.

Таблиця 1.2 – Основні сфери застосування кіберфізичних систем

Сфера застосування	Тип задач	Приклади рішень
Охорона здоров’я	Моніторинг стану пацієнтів, діагностика, автоматичне сповіщення	Системи віддаленого ЕКГ, трекери глюкози, розумні шприци
Розумне житло	Автоматизація освітлення, опалення, охорона, енергозбереження	Системи SmartHome, керування через смартфон, голосові помічники
Агропромисловий сектор	Контроль мікроклімату, моніторинг ґрунту, управління поливом	Системи SmartAgro, автоматизовані теплиці, датчики вологості

Кінець таблиці 1.2

Промисловість (Індустрія 4.0)	Керування виробничими лініями, контроль якості, цифрові двійники	CPS у виробництві, розумні роботи, автоматичний контроль
Транспорт	Автономне керування, навігація, безпека руху	Автопілот Tesla, роботизовані платформи, V2X-комунікації

На відміну від класичних вбудованих систем, КФС функціонують у постійній взаємодії з навколишнім середовищем, а тому особливо важливо забезпечити їх надійність, безпечність та захист даних, які часто містять конфіденційну або персональну інформацію. Розробка таких систем потребує впровадження сучасних засобів криптографічного захисту, багаторівневої автентифікації та безпечних протоколів обміну [5].

Актуальність цієї теми підтверджується зростаючим попитом на фахівців у галузі безпеки КФС. Наприклад, компанія Kudu Dynamics у США шукає інженера з реверс-інжинірингу кіберфізичних систем з оплатою $175,000–$200,000 на рік. Ця позиція передбачає аналіз апаратного та програмного забезпечення, розробку автоматизованих інструментів тестування та забезпечення безпеки систем [6].

Крім того, Університет Джорджії відкрив кілька вакансій для викладачів у сфері КФС, зосереджених на штучному інтелекті, обробці сигналів та безпеці систем, що підкреслює важливість досліджень у цій галузі [7].

Загалом, за даними ZipRecruiter, у США доступно понад 400 вакансій, пов'язаних із кіберфізичними системами, із середньою заробітною платою від $114,000 до $175,000 на рік [8].

Додатковий аналіз статистичних даних показав прогноз зростання ринку КФС [37]:

- 2024: $124,1 млрд;
- 2029: $255,3 млрд;
- середньорічний темп зростання (CAGR): 15,5%;

- основні галузі: виробництво, енергетика, охорона здоров’я, транспорт, розумні міста;
- ключові технології: IoT, цифрові двійники, штучний інтелект, edge computing;
- регіони-лідери: Азійсько-Тихоокеанський регіон, Північна Америка;
- основні гравці: Siemens, Honeywell, ABB, Schneider Electric, Rockwell Automation.

Найбільш затребувані навички [37]:

- Кібербезпека (захист критичної інфраструктури);
- штучний інтелект (аналіз даних, машинне навчання);
- IoT (розробка та інтеграція сенсорних мереж);
- обробка сигналів (аналіз та інтерпретація даних з датчиків);
- Програмування (Python, C++, Java для розробки КФС).

Таким чином, забезпечення безпеки та приватності в КФС є не лише технічною необхідністю, але й актуальним напрямом з високим попитом на ринку праці.

1.2 Огляд сучасних сенсорних систем для моніторингу параметрів життєзабезпечення

Сенсорні системи є ключовими компонентами кіберфізичних систем, що виконують функції збору даних із навколишнього середовища. У контексті життєзабезпечення ці системи призначені для постійного моніторингу критично важливих фізичних параметрів — температури, вологості, рівня вуглекислого газу, тиску, серцевого ритму, рівня глюкози тощо.

Історично розвиток сенсорних технологій розпочався з простих аналогових пристроїв, що реагували на зміни фізичних величин, як-от термометри чи барометри. З появою напівпровідникової електроніки в середині XX століття стало можливим створення мініатюрних і надійних електронних сенсорів. Подальший прорив відбувся завдяки мікроелектромеханічним системам (MEMS) у 1980–1990-

х роках, що дозволило поєднати сенсорні елементи з цифровою обробкою сигналу. Сьогодні сенсори інтегруються з бездротовими модулями зв’язку, мікроконтролерами та системами штучного інтелекту, що відкриває нові горизонти в моніторингу життєвих параметрів у режимі реального часу [9].

За способом взаємодії з обчислювальними пристроями сенсори поділяються на:

- провідні (wired) — використовуються в стаціонарних рішеннях із високим рівнем надійності;
- бездротові (wireless) — інтегруються в IoT-пристрої, мають низьке енергоспоживання, але вимагають захисту при передаванні даних [10].

Найпоширеніші типи сенсорів у системах життєзабезпечення:

- температурні сенсори (DS18B20, DHT11, MLX90614) — для вимірювання температури повітря, рідин або тіла (рис. 1.2);
- сенсори вологості (DHT22, HIH6130) — використовуються в кліматичних системах, інкубаторах (рис. 1.3);
- газові сенсори (MQ-2, MQ-135) — для виявлення вуглекислого газу, диму, метану тощо;
- біомедичні сенсори (MAX30100, AD8232, E-Health Kit) — для зчитування ЕКГ, пульсу, сатурації [11];
- сенсори тиску (BMP280, MPX5700) — в системах вентиляції, реанімації або промислового моніторингу.
- Інтеграція таких сенсорів у загальну архітектуру КФС здійснюється через мікроконтролери (наприклад, Arduino, ESP32, Raspberry Pi), які обробляють сигнал, формують цифрові пакети даних і передають їх до центрального вузла або хмарного сервісу для подальшого аналізу [12].

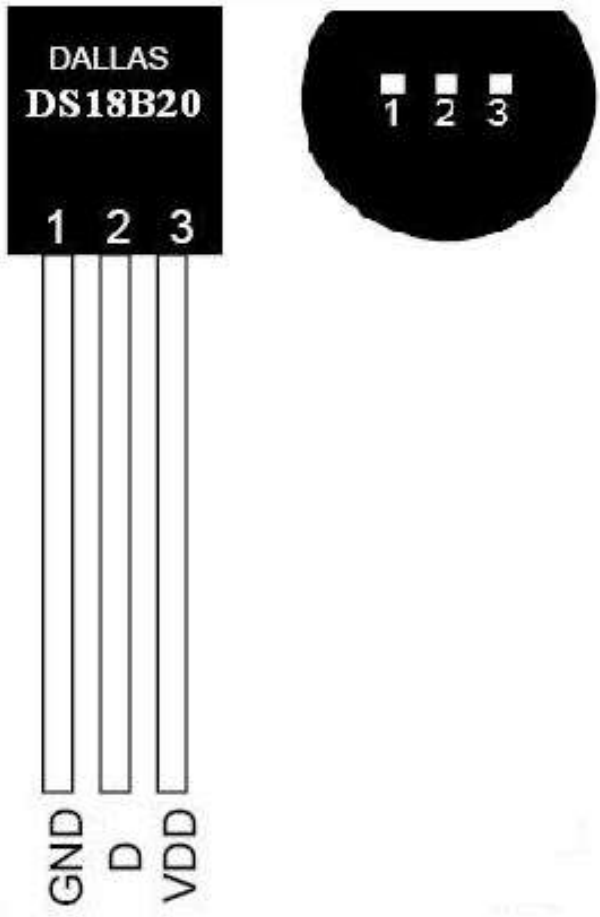
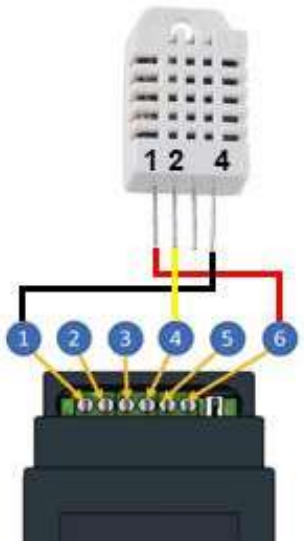


Рисунок 1.2 - Температурний сенсор DS18B20



DHT22 pins	
1	VCC
2	DATA
3	
4	GND

Верхня група контактів	
1	GND
2	імпульсний вхід 1
3	імпульсний вхід 2
4	вхід температурних датчиків
5	аналоговий вхід
6	+5 В

Рисунок 1.3 - Сенсор вологості DHT22

Сенсорні системи, що застосовуються в КФС життєзабезпечення, повинні відповідати низці технічних та безпекових вимог, які регламентуються міжнародними стандартами, зокрема:

- висока точність вимірювання визначається специфікаціями, що встановлюють допустимі похибки та методи калібрування сенсорів. Наприклад, стандарт IEC 60770-1 описує методи оцінки точності аналогових сенсорів;
- мінімальний час реакції визначає швидкість, з якою сенсор реагує на зміну вимірюваного параметра. Це критично для систем, де необхідна оперативна реакція, наприклад, у медичних пристроях або системах безпеки;
- низьке енергоспоживання особливо важливо для бездротових сенсорних мереж, де енергоефективність продовжує термін служби пристроїв. Стандарти, такі як IEEE 802.15.4, регламентують енергоспоживання для бездротових сенсорних мереж;
- захист від зовнішніх впливів заключається в тому, що сенсори повинні бути стійкими до впливу води, пилу та електромагнітних перешкод. Стандарти IP-класифікації (IEC 60529) визначають рівні захисту від проникнення твердих часток та вологи;
- безпека переданих даних забезпечується через впровадження криптографічних протоколів та автентифікації. Стандарти, такі як IEC 62443, встановлюють вимоги до кібербезпеки для промислових автоматизованих систем, включаючи сенсорні мережі.

Питання безпеки передачі даних є особливо важливим, оскільки дані, які зчитуються сенсорами, можуть бути персональними або чутливими (особливо в медичних і побутових рішеннях), тому в проєктуванні КФС необхідно реалізовувати зашифровану передачу даних, вбудовані механізми автентифікації сенсорів, а також контроль доступу до інформації [5].

Для кращого розуміння загальної структури сенсорних систем КФС на рисунку 1.4 наведено їх класифікацію за основними критеріями.

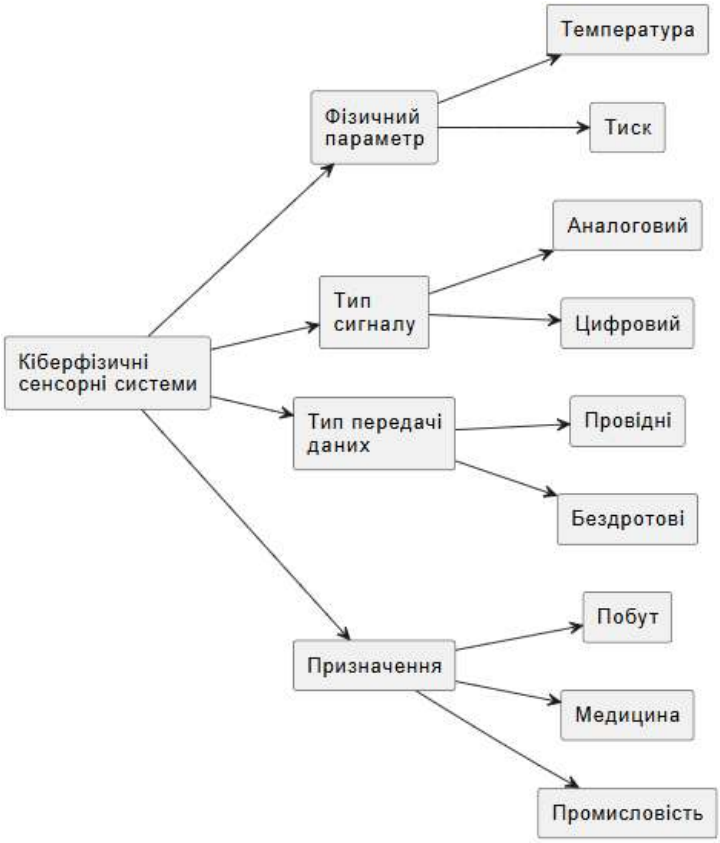


Рисунок 1.4 – Класифікація сенсорних систем у КФС за критеріями

Схема на рисунку 1.4 узагальнено відображає основні підходи до класифікації сенсорних систем, які використовуються у складі кіберфізичних систем. Класифікація побудована за кількома ознаками: типом фізичної величини, типом сигналу, способом передавання даних і сферою призначення. Зокрема, у групі фізичних параметрів на рисунку представлено базові типи, як-от температура і тиск, однак у реальних системах також активно використовуються сенсори вологості, газові аналізатори, біомедичні сенсори (пульс, сатурація, ЕКГ), які були детально описані в основному тексті.

Така структуризація дає змогу ефективно підбирати сенсорні компоненти для конкретних завдань, враховуючи як вимоги до точності, швидкодії та енергоспоживання, так і умови експлуатації (медичні, побутові, промислові системи тощо). Узгодження класифікаційних підходів із практичними прикладами дозволяє гнучко проєктувати сенсорні підсистеми кіберфізичних систем життєзабезпечення.

Сучасний розвиток сенсорних технологій у сфері життєзабезпечення характеризується низкою інноваційних розробок та патентів. Наприклад, компанія Masimo впровадила систему Patient SafetyNet, яка забезпечує дистанційний моніторинг пацієнтів і дозволяє значно знизити кількість критичних випадків у медичних закладах. Ця система використовує бездротові сенсори для постійного відстеження життєвих показників, таких як частота серцевих скорочень та рівень кисню в крові .

У галузі носимих біометричних пристроїв компанія Valencell розробила технологію PerformTek, яка дозволяє безперервно вимірювати такі параметри, як частота серцевих скорочень, рівень кисню в крові та інші біометричні показники. Ця технологія інтегрована в продукцію провідних виробників електроніки, включаючи Bose, Suunto та LG .

Щодо патентів, варто відзначити розробку системи для моніторингу життєвих показників, яка дозволяє відстежувати використання сенсорів, їх калібрування та інші параметри експлуатації. Така система забезпечує підвищену надійність та точність вимірювань, що є критично важливим у медичних застосуваннях .

Огляд ключових досліджень, компаній та інноваційних розробок у сфері сенсорних технологій та кіберфізичних систем (КФС) для моніторингу життєвих параметрів показав, що даний напрям є дуже актуальним.

У сфері конференцій та виставок, присвячених сенсорним технологіям, виділяються такі події:

SENSOR+TEST у Нюрнберзі — провідна міжнародна виставка, що охоплює широкий спектр технологій вимірювання та моніторингу. Цей захід є платформою для презентації новітніх розробок у галузі сенсорики та обміну досвідом між фахівцями з усього світу .

11th International Symposium on Sensor Science (I3S 2025), який відбудеться в Барселоні, Іспанія, з 17 по 19 листопада 2025 року. Цей симпозіум збирає провідних дослідників та експертів для обговорення останніх досягнень у сфері сенсорних технологій .

SENSORCOMM 2024, що пройшов з 3 по 7 листопада 2024 року в Ніцці, Франція. Ця конференція зосереджена на комунікаційних аспектах сенсорних систем та їх інтеграції в сучасні мережеві інфраструктури .

Участь у таких заходах сприяє обміну знаннями, встановленню нових партнерств та впровадженню передових технологій у практичні рішення для моніторингу параметрів життєзабезпечення.

Дослідницькі роботи та академічні проєкти присвячені даній темі:

IoT-орієнтовані медичні системи моніторингу.

У роботі на PubMed Central розглядаються різні підходи до проєктування й впровадження IoT-систем для поліпшення якості життя пацієнтів, де сенсорні вузли забезпечують безперервне збирання сигналів та їхню передачу лікарям у реальному часі

Безпровідні сенсорні мережі для охорони здоров’я та «розумного дому»

Дослідники пропонують архітектуру WSN із використанням глибинних нейромереж для обробки та класифікації даних сенсорів у реальному часі, що особливо корисно для віддаленого догляду за людьми похилого віку

Розвиток MEMS-сенсорів для моніторингу повітря

Огляд у Science of The Total Environment підкреслює, що MEMS-пристрої значно покращили точність та швидкодію при виявленні забруднювачів повітря, а також зменшили енергоспоживання та габарити сенсорів

Огляд кіберфізичних систем у промисловості та транспорті демонструє широкий спектр їхнього застосування. Зокрема, Claroty описує десять прикладів таких систем для управління трафіком та автономними транспортними засобами. Це свідчить про глобальну інтеграцію сенсорних мереж у промислові процеси.

У сфері медичних кіберфізичних систем (Medical Cyber-Physical Systems, MCPS) в Sci-Direct представлено концепцію, яка передбачає інтеграцію аналізу фізіологічних даних із мережевими сервісами. Це створює основу для систем моніторингу пацієнтів у режимі 24/7, що є важливим кроком у розвитку медицини.

Огляд IoT у медицині представлено в публікації Journal of Biosensors & Bioelectronics. Автори аналізують практичні приклади використання бездротових

сенсорних мереж у медичних установах і зазначають, що значна частина рішень поки що не перейшла у масове застосування через проблеми масштабованості та забезпечення безпеки.

Серед провідних компаній виділяють Bio-Rad Laboratories, яка є лідером ринку біосенсорів. У 2024 році вона продемонструвала зростання сегмента на 7,2 % (до ~$1.7 млрд), що стало можливим завдяки інноваціям у прецизійній діагностиці та ефективним M&A-стратегіям.

Phosphorus Cybersecurity пропонує платформу захисту кіберфізичних систем (КФС), що охоплює IoT, OT, IIoT та IoMT. Їх рішення особливо орієнтоване на превентивне виявлення вразливостей у сенсорних мережах.

LiXiA (Австралія) розробила бюджетні безпровідні сенсори для моніторингу рівня повеней. Їх вартість становить лише $1–2 тис. у порівнянні з традиційними сенсорами за $50 тис. Пристрої вже відповідають національним стандартам попередження про повінь.

Університетські дослідники з Northwestern University створили безконтактний носимий сенсор, який аналізує гази шкіри. Це дозволяє безпечно здійснювати моніторинг новонароджених та осіб з чутливою шкірою.

Plant-e впроваджує концепцію біогенерації електроенергії від рослин для автономного живлення сенсорів у віддалених екосистемах. Такий підхід активно розвивається в рамках досліджень у галузі environmental IoT.

Інноваційні продукти та ринки демонструють помітне зростання. Глобальний ринок біосенсорів, за прогнозами, сягне $75.84 млрд до 2033 року зі щорічним темпом зростання близько 8 %, що стимулюється попитом на носимі пристрої та неінвазивні датчики для охорони здоров’я.

Environmental Sensing & Monitoring активно розширюється. Звіт GlobeNewswire підкреслює тренд мініатюризації сенсорів та інтеграції AI для швидкої обробки екологічних даних, що особливо корисно для застосування на дронах та супутниках.

У сфері глибинного навчання в бездротових сенсорних мережах (WSN) стаття Elsevier висвітлює використання згорткових нейронних мереж для аналізу великих потоків даних. Це значно підвищує точність виявлення аномалій у критичних інфраструктурах.

Розвиток технологій Lab-on-a-Chip та гнучких сенсорів є ще одним ключовим напрямом. Очікується, що на I3S 2025 у Барселоні будуть представлені нові носимі сенсори та міні-лабораторії на чіпі, здатні проводити молекулярні аналізи прямо на місці або на тілі пацієнта.

Загалом сфера сенсорних технологій для моніторингу життєвих параметрів стрімко еволюціонує. Цьому сприяє симбіоз академічних досліджень, інновацій стартапів та рішень лідерів ринку. Платформи захисту КФС, біосенсори для медицини, безконтактні носимі пристрої та AI-інтегровані мережі формують майбутнє систем моніторингу.

1.3 Забезпечення приватності в IoT- та CPS-середовищах

КФС у сфері моніторингу параметрів життєзабезпечення дедалі частіше взаємодіють з персональними, біомедичними та екологічними даними, що робить питання захисту приватності однією з ключових складових у їхньому проєктуванні. Зокрема, такі системи часто є частиною архітектури Інтернету речей (IoT), де характерною рисою є розподіленість, динамічність підключень та відкритість каналів передачі даних.

У контексті життєзабезпечення КФС збирають дані з великої кількості сенсорів: температури тіла, вологості повітря, рівня шкідливих газів, пульсу, тиску тощо. Ці дані можуть бути віднесені до категорії конфіденційних або критично важливих, що вимагає застосування відповідних механізмів захисту на всіх рівнях обробки — від джерела збору інформації до її збереження та візуалізації [13].

Основні загрози приватності в CPS-середовищах включають:

− перехоплення даних у каналах зв’язку, особливо при використанні незашифрованих або нестійких протоколів передачі;

− несанкціонований доступ до пристроїв (сенсорів, шлюзів, серверів обробки);

− підміна або модифікація даних, яка може спотворити результати моніторингу;

− відсутність автентифікації між вузлами або підключеними пристроями;

− витоки інформації у хмарних або API-рішеннях через неналежне управління ключами або правами доступу [14].

Для забезпечення надійного захисту персональних даних у КФС застосовуються такі технічні та програмні рішення:

− шифрування даних (AES, ECC) на рівні сенсорів або мікроконтролера для унеможливлення перехоплення змістовної інформації;

− протоколи захищеної передачі - HTTPS, TLS, MQTTs, які використовують сертифікати та ключі сесій;

− системи автентифікації пристроїв і користувачів, зокрема багатофакторна (2FA), криптографічна (PKI);

− хешування та цифрові підписи, що гарантують цілісність даних;

− контроль доступу до даних через політики типу RBAC (role-based access control) або ABAC (attribute-based access control);

− системи журналювання та моніторингу дій (audit logs), які дозволяють виявити аномалії або несанкціоновані спроби доступу.

Крім технічних засобів, важливо враховувати також правові та етичні аспекти. Наприклад, відповідно до вимог Загального регламенту ЄС про захист персональних даних (GDPR), збір і обробка даних повинні здійснюватися лише з інформованої згоди користувача, із зазначенням мети обробки та способів збереження. Аналогічні вимоги містяться в Законі України «Про захист персональних даних».

Інтегрований підхід до безпеки КФС передбачає врахування принципу Security by Design - тобто вбудовування механізмів безпеки вже на етапі проєктування системи, що дає змогу не лише уникати значних втрат через витік або

спотворення інформації, але й підвищує рівень довіри користувачів до таких систем.

На рисунку 1.5 представлено багаторівневу модель забезпечення приватності у кіберфізичних системах, де захист реалізується на кожному етапі проходження даних - від сенсора до кінцевого користувача.

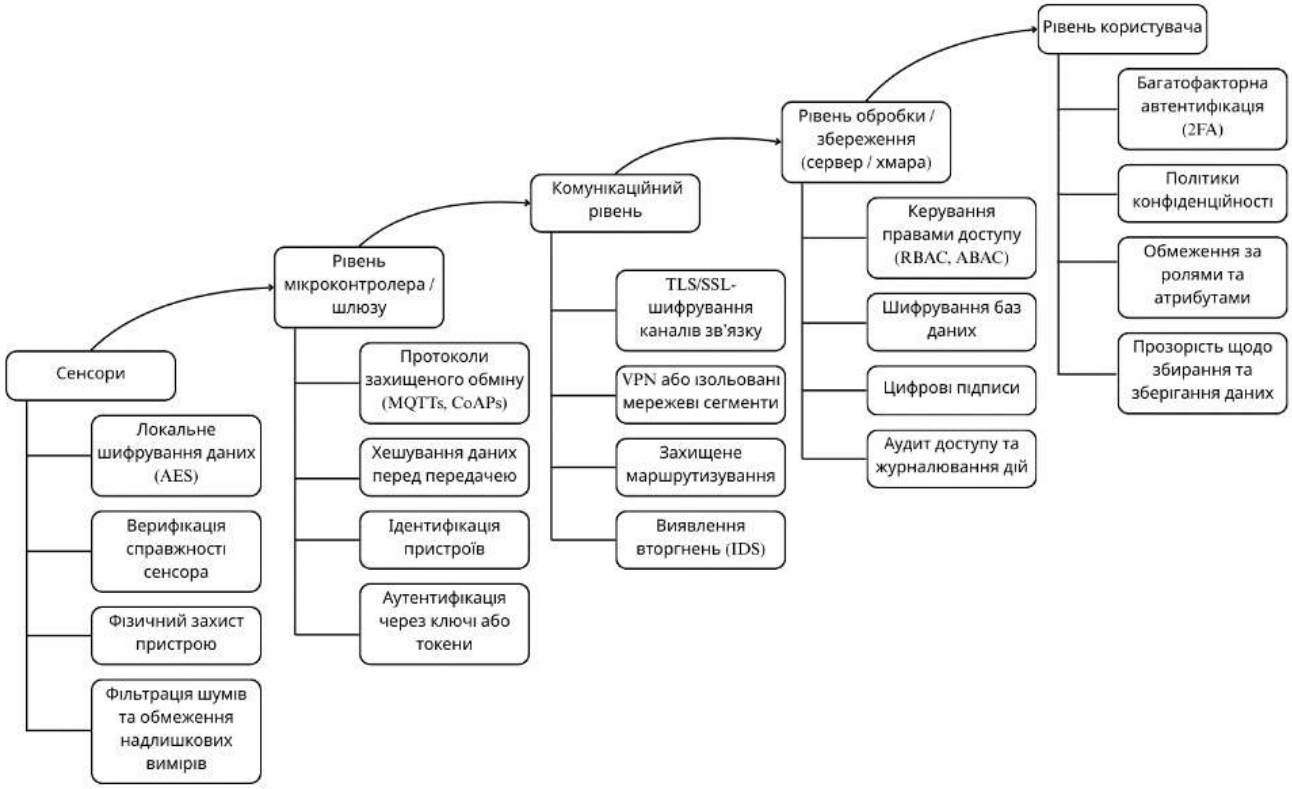


Рисунок 1.5 – Засоби забезпечення приватності у кіберфізичних системах

На рівні сенсорів застосовуються базові методи шифрування (наприклад, AES), фізичний захист пристрою, попередня фільтрація даних та верифікація джерела. Це дозволяє мінімізувати ризик перехоплення або спотворення інформації ще на стадії збору.

На рівні мікроконтролера або шлюзу дані проходять через механізми хешування, автентифікації пристрою та передаються через захищені протоколи типу MQTTs або CoAPs.

Комунікаційний рівень забезпечується через використання протоколів TLS/SSL, ізольованих мережевих сегментів, віртуальних приватних мереж (VPN) і систем виявлення вторгнень (IDS), що запобігає атакам на мережу.

На рівні обробки та збереження (локально або в хмарі) дані шифруються, контроль доступу здійснюється через політики RBAC або ABAC, застосовуються цифрові підписи та ведеться аудит дій.

На рівні користувача реалізується багатофакторна автентифікація, обмеження прав доступу, політики конфіденційності, а також прозорість щодо мети та способів зберігання інформації.

Така ієрархічна структура дозволяє забезпечити комплексний підхід до захисту приватності, що відповідає сучасним вимогам як технічного, так і правового характеру.

Крім вищезгаданих технічних рішень, усе більшої популярності набуває впровадження концепції Zero Trust Architecture (ZTA), яка базується на принципі «не довіряй нікому за замовчуванням». У межах КФС це означає, що кожен пристрій, користувач або мікросервіс проходить перевірку автентичності та авторизації незалежно від його розташування у мережі. Такий підхід дає змогу мінімізувати ризики внутрішніх загроз і витоків у складних розподілених середовищах [15, 16].

Іншим перспективним напрямом є застосування технологій блокчейн у кіберфізичних системах для забезпечення цілісності даних та достовірності записів. Розподілена структура зберігання інформації дає змогу уникнути єдиної точки відмови, підвищує стійкість системи до атак та забезпечує прозорість журналювання подій.

Крім технічних і нормативних заходів, актуальним є впровадження інтелектуальних систем виявлення аномалій (AI-based IDS), які використовують методи машинного навчання для аналізу телеметрії сенсорів у реальному часі. Це дозволяє не лише виявляти потенційні порушення політик безпеки, але й оперативно реагувати на нові типи атак, зокрема zero-day.

Заг У контексті стрімкого розвитку Інтернету речей (IoT) та кіберфізичних систем (CPS) питання забезпечення приватності й інформаційної безпеки набувають особливої актуальності. Збільшення кількості підключених пристроїв, обробка значних обсягів персональних та чутливих даних, а також необхідність

цілодобового функціонування таких систем зумовлюють потребу у впровадженні сучасних підходів до захисту інформації.

На сучасному етапі стандартом формування безпечних середовищ виступає інтеграція концепцій Security by Design і Zero Trust. Перша передбачає врахування вимог безпеки вже на етапі проєктування систем, що дозволяє мінімізувати вразливості та забезпечити більш високий рівень захисту в процесі їх подальшої експлуатації. Друга базується на принципі недовіри до будь-яких суб'єктів і об'єктів усередині мережі, вимагаючи постійної перевірки їхньої автентичності та повноважень.

Не менш важливою є концепція розподіленого зберігання даних. Вона дозволяє підвищити стійкість систем до збоїв та атак, а також забезпечити доступність інформації навіть в умовах порушення роботи окремих елементів інфраструктури. Це особливо актуально для обробки біомедичних і екологічних даних, де критично важливою є не лише конфіденційність, а й цілісність і доступність інформації.

Водночас дедалі ширше впроваджується інтелектуальний моніторинг, що базується на технологіях штучного інтелекту та машинного навчання. Використання таких засобів дозволяє здійснювати виявлення аномальної активності та потенційних загроз у режимі реального часу, що значно підвищує ефективність систем захисту.

Таким чином, сучасний підхід до забезпечення приватності та безпеки в IoT- і CPS-середовищах формується на основі комплексної взаємодії зазначених технологій. Це підтверджується численними науковими дослідженнями та аналітичними оглядами, які наголошують на необхідності системного підходу до організації захисту на всіх етапах життєвого циклу систем — від проєктування до їхньої практичної експлуатації.

Такий підхід є запорукою надійності, стійкості та відповідності сучасним вимогам кібербезпеки.

1.4 Постановка задачі

За результатами аналізу наукових джерел, існуючих архітектурних підходів та технологічних рішень у сфері кіберфізичних систем контролю параметрів життєзабезпечення, було виявлено низку невирішених проблем, пов’язаних із недостатньою інтеграцією засобів захисту приватності, низькою гнучкістю архітектури для масштабування та обмеженими можливостями адаптації до змін середовища. У зв’язку з цим виникла необхідність створення прототипу системи, яка поєднувала б надійний збір критичних фізичних параметрів із сучасними механізмами кібербезпеки, що особливо актуально в умовах розвитку Інтернету речей та зростаючих ризиків витоку чутливих біомедичних або екологічних даних.

1.5 Висновок до першого розділу

У подальших розділах кваліфікаційної роботи детально описано всі етапи реалізації запропонованого рішення — від формування технічних вимог до створення прототипу, його тестування та аналізу результатів. Послідовна структура роботи дозволяє розглянути кожен із компонентів системи не лише з точки зору технічної реалізації, але й з урахуванням вимог до надійності, безпеки, масштабованості та енергоефективності.

Відповідно до цього, основними задачами, що були вирішені й описані в межах розділів 2–4, є:

1. Проєктування архітектури КФС, яка включає сенсорний рівень, модуль обробки даних, механізми захищеного зв’язку та підсистему керування доступом (розділ 2). Особлива увага приділяється розподіленій структурі, що дозволяє адаптувати систему до змін кількості сенсорів або точок збирання даних без необхідності повної перебудови інфраструктури.

2. Обґрунтування вибору апаратної та програмної платформи для реалізації системи з урахуванням ключових критеріїв — точності вимірювання, енергоспоживання, швидкодії, зручності в розгортанні та підтримки сучасних

стандартів безпеки (розділ 2). Було проведено порівняльний аналіз альтернативних рішень із наведенням їхніх переваг і недоліків.

3. Реалізація функціонального прототипу системи на основі мови програмування Python, що забезпечила (розділ 3):

- високу швидкість розробки;
- наявність великої кількості бібліотек (gpiozero, Adafruit_DHT — для сенсорів; cryptography, PyCrypto — для шифрування; paho-mqtt, socket — для передавання даних);
- активну спільноту підтримки .

4. Інтеграція базових засобів забезпечення приватності, включаючи шифрування переданих даних, автентифікацію пристроїв, розмежування прав доступу до інформації та базову систему логування подій безпеки. Такий підхід дозволяє вбудовувати принципи Security by Design вже на початкових етапах розробки (розділ 3).

5. Проведення експериментальної оцінки роботи системи в лабораторних умовах. Тестування охоплювало аналіз точності зчитування параметрів, стійкості до втрати зв’язку, затримок при передаванні, реакції на збої та ефективності реалізованих криптографічних рішень (розділ 4).

6. Аналіз результатів моделювання та тестування, а також формулювання висновків щодо подальшого розвитку системи, її можливостей масштабування, інтеграції в більші інфраструктури (наприклад, системи розумного будинку або клінічні інформаційні системи), а також рекомендацій щодо її вдосконалення з урахуванням новітніх технологій, зокрема edge-computing, AI-модулів і блокчейн-технологій (розділ 4).

Таким чином, сформульовані задачі дослідження відображають цілісний життєвий цикл створення кіберфізичної системи: від концептуального проєктування до практичної реалізації функціонального прототипу з урахуванням сучасних вимог до захисту інформації, що значною мірою підтверджує актуальність і прикладну цінність даної роботи.

Отже, в першому розділі було проведено комплексний аналіз теоретичних та прикладних підходів до побудови кіберфізичних систем для контролю параметрів життєзабезпечення з урахуванням вимог до приватності:

- узагальнено поняття кіберфізичних систем, їх структуру та основні сфери застосування;
- встановлено, що КФС інтегрують фізичні сенсори, обчислювальні модулі та мережеві інтерфейси в єдине адаптивне середовище, здатне до автономного прийняття рішень;
- проаналізовано сучасні сенсорні системи, що використовуються для моніторингу життєво важливих параметрів;
- виокремлено типи сенсорів, актуальні моделі та вимоги до точності, енергоспоживання й безпеки;
- запропоновано класифікацію сенсорів за фізичними параметрами, сигналами, способами передавання даних і галузями застосування;
- розглянуто загрози приватності в IoT- та CPS-середовищах та визначено ключові рівні захисту персональних даних — від сенсорного до користувацького;
- узагальнено підходи до шифрування, автентифікації, контролю доступу та аудиту дій;
- наголошено на важливості принципу Security by Design при побудові таких систем;
- сформульовано задачі, що будуть реалізовані в межах практичної частини роботи. Задачі охоплюють проєктування архітектури системи, вибір технічних засобів, реалізацію програмної складової та проведення експериментального тестування.

Таким чином, розділ 1 заклав теоретичну та методологічну основу для подальшого проєктування, розробки та аналізу кіберфізичної системи контролю параметрів життєзабезпечення з інтегрованими засобами захисту приватності. У цьому розділі було систематизовано ключові поняття, пов’язані з архітектурою КФС, їх функціональними компонентами, а також принципами побудови безпечних розподілених систем у середовищі IoT. Особливу увагу приділено

аналізу сенсорних технологій, методів збирання та передавання даних, вимог до точності, надійності й захисту інформації.

Розглянуто сучасні підходи до забезпечення приватності, включаючи концепції Security by Design та Zero Trust, які стають обов’язковими елементами при розробці систем, що працюють з критичними біомедичними або екологічними даними. Крім того, на основі огляду наукової літератури та практичних кейсів було виявлено поточні виклики у сфері безпеки КФС, серед яких — складність забезпечення захисту в умовах обмежених ресурсів, висока динаміка змін середовища та потреба в гнучких, масштабованих рішеннях.

У підсумку, теоретичні положення, розглянуті у першому розділі, не лише окреслили рамки дослідження, а й дозволили сформулювати вимоги до функціональності та безпеки майбутньої системи. Це створило підґрунтя для практичної реалізації прототипу, опис якої наведено у наступних розділах, починаючи з визначення архітектурних принципів та вибору технічних рішень, що максимально відповідають поставленим завданням.

2 ПРОЄКТУВАННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ КОНТРОЛЮ ПАРАМЕТРІВ ЖИТТЄЗАБЕЗПЕЧЕННЯ

2.1 Визначення функціональних вимог до системи

Аналіз вимог до програмного забезпечення кіберфізичної системи контролю параметрів життєзабезпечення з урахуванням приватності є ключовим етапом у розробці такої системи. Цей аналіз має враховувати як функціональні, так і нефункціональні вимоги, з особливим акцентом на забезпечення інформаційної безпеки та приватності користувачів.

Функціональні вимоги — це специфікації того, що повинна робити система або програмне забезпечення для задоволення потреб користувачів та досягнення бізнес-цілей. Вони описують функції, поведінку та операції, які система має виконувати, такі як обробка даних, взаємодія з користувачем, обчислення та інтеграція з іншими системами [17].

Якісні функціональні вимоги повинні бути [18]:

- Конкретними - чітко описувати функцію без двозначностей;
- Вимірюваними - мати критерії для перевірки виконання;
- Досяжними - реалістичними для реалізації в межах проекту;
- Релевантними - відповідати цілям та потребам користувачів;
- Обмеженими у часі - мати визначені терміни реалізації;

Перераховані характеристики часто узагальнюються акронімом SMART (Specific, Measurable, Achievable, Relevant, Time-bound).

Кіберфізична система контролю параметрів життєзабезпечення з підтримкою конфіденційності призначена для постійного моніторингу фізичних параметрів навколишнього середовища або організму, виявлення відхилень від нормальних значень, реагування на критичні події та забезпечення безпечного зберігання і передачі зібраної інформації.

Функціональні вимоги системи сформульовано відповідно до ключових компонентів і сценаріїв її використання подані у таблиці 2.1, що структурує основні напрями реалізації системи та короткий опис кожної вимоги.

Таблиця 2.1 – Основні функціональні вимоги до системи

№	Категорія	Вимога	Опис
1	Збір даних	Підключення до сенсорів	Температура, вологість, CO_2, пульс, рух (залежно від сценарію)
2	Обробка	Пороговий аналіз	Виявлення критичних значень, оцінка небезпеки
3	Візуалізація	Інтерфейс	Відображення значень, графіки, попередження
4	Збереження	Локальна база даних	SQLite або аналогічна вбудована СКБД
5	Безпека	Шифрування та автентифікація	Захист даних при зберіганні та доступі
6	Масштабованість	Підтримка нових сенсорів	Гнучке розширення системи

Основні функціональні вимоги до системи сформульовані з урахуванням її призначення та необхідності забезпечення комплексного контролю параметрів життєзабезпечення. Передусім система повинна реалізовувати збір фізичних параметрів. Це передбачає зчитування показників з різноманітних сенсорів, зокрема температури повітря, вологості, рівня вуглекислого газу, пульсу та руху. Збір даних має здійснюватися з визначеною періодичністю, що задається на етапі налаштування системи.

Наступним важливим етапом є обробка та аналіз отриманої інформації. У межах цієї функції система повинна виконувати фільтрацію шумів і усереднення значень для підвищення достовірності результатів. Крім того, необхідно забезпечити виявлення відхилень від встановлених нормативних меж і здійснювати оцінку рівня небезпеки ситуацій відповідно до закладеної логіки прийняття рішень.

Зберігання даних є ще однією критичною вимогою. Усі параметри мають фіксуватися у локальній базі даних (наприклад, SQLite або іншій вбудованій системі керування базами даних), що дозволить уникнути залежності від зовнішніх

серверів. Також важливою є можливість формування історії показників за обраний часовий проміжок.

Необхідно передбачити засоби візуалізації та взаємодії з користувачем. До таких відносяться графічний інтерфейс для перегляду поточних значень, система сповіщень про тривожні ситуації та інструменти для налаштування порогових значень параметрів. Це забезпечить зручність у користуванні системою та оперативність реагування на можливі загрози.

Для своєчасного інформування користувачів і відповідного реагування на небезпечні стани передбачається реалізація системи оповіщення та активації виконавчих пристроїв. Вона включатиме генерацію сповіщень у різних формах (звукових, візуальних, повідомлень на телефон), а також можливість автоматичного включення відповідних пристроїв, наприклад, вентиляторів чи сирен.

Особлива увага приділяється забезпеченню безпеки та конфіденційності даних. Передбачено шифрування інформації під час зберігання і передачі, впровадження механізмів автентифікації користувачів для обмеження доступу, а також встановлення ролей для чіткої диференціації прав доступу.

Нарешті, система повинна бути масштабованою та модульною. Це означає можливість підключення додаткових сенсорів і адаптацію функціоналу до різних сценаріїв використання, включаючи домашній, медичний чи аграрний моніторинг.

З урахуванням апаратних обмежень, у межах дипломної роботи буде реалізовано спрощений прототип системи. Він імітуватиме роботу сенсорів та продемонструє ключові функції, такі як збір, обробка, збереження, візуалізація та захист даних. Це дозволить перевірити працездатність основних механізмів без необхідності використання дорогого спеціалізованого обладнання.

2.2 Визначення нефункціональних вимог до системи

Нефункціональні вимоги визначають обмеження та характеристики, які не стосуються безпосередньо функціональної поведінки системи, але мають критичне

значення для її ефективного функціонування [18]. У контексті кіберфізичної системи контролю параметрів життєзабезпечення з підтримкою приватності, такі вимоги охоплюють аспекти надійності, продуктивності, масштабованості, безпеки, зручності використання, сумісності та нормативної відповідності.

Дані вимоги забезпечують високий рівень якості програмно-апаратного комплексу, визначають очікування користувачів щодо поведінки системи в реальному часі та гарантують відповідність сучасним стандартам кібербезпеки й обробки даних. Їх формалізацію здійснено на основі загальноприйнятої класифікації, що наведена в таблиці 2.2, та відповідно до практичних сценаріїв експлуатації системи.

У межах розробки кіберфізичної системи контролю параметрів життєзабезпечення, що функціонує з урахуванням вимог конфіденційності та безпеки, особливу увагу слід приділити нефункціональним вимогам. Вони не визначають, що саме має робити система, але описують як вона повинна це робити — тобто встановлюють критерії якості функціонування, надійності, зручності, масштабованості, безпеки та відповідності нормативним вимогам. Формалізація цих вимог є критично важливою, оскільки система має працювати в умовах підвищеної відповідальності, зокрема при моніторингу фізіологічних або екологічних параметрів, що мають безпосередній вплив на безпеку життєдіяльності користувачів.

Таблиця 2.2 – Класифікація та опис нефункціональних вимог до системи

№	Категорія	Нефункціональна вимога	Опис
1	Надійність (Reliability)	Забезпечення безперервної роботи системи	Система має стабільно функціонувати протягом заданого часу без збоїв або втрати даних.

Кінець таблиці 2.2

№	Категорія	Нефункціональна вимога	Опис
2	Продуктивність (Performance)	Мінімальна затримка в обробці та передачі даних	Реакція на зміни параметрів має здійснюватися в реальному часі або з мінімальною затримкою.
3	Масштабованість (Scalability)	Можливість розширення системи	Система повинна підтримувати додавання нових сенсорів, користувачів або функцій без втрати ефективності.
4	Безпека (Security)	Захист даних від несанкціонованого доступу та модифікації	Реалізація механізмів шифрування, автентифікації та авторизації користувачів.
5	Зручність використання (Usability)	Дружній інтерфейс для користувача	Інтерфейс має бути інтуїтивно зрозумілим і адаптованим до потреб користувачів з різним досвідом.
6	Сумісність (Compatibility)	Інтеграція з іншими пристроями та програмними рішеннями	Система повинна підтримувати стандартні протоколи обміну даними (наприклад, MQTT, REST API).
7	Законодавча відповідність (Compliance)	Відповідність нормам захисту персональних даних	Дотримання стандартів, таких як GDPR, ISO/IEC 27001.

По-перше, для кожної категорії необхідно визначити конкретні метрики й порогові значення. Наприклад, надійність (Reliability) можна вимірювати середнім часом безвідмовної роботи (MTBF) не менше 1 000 годин, а рівень втрачених або пошкоджених повідомлень — не більше 0,01 %.

Продуктивність (Performance) слід контролювати через максимальну латентність реакції на зміну датчиків – не більше 100 мс, а пропускну здатність – не менше 100 записів у секунду.

Масштабованість (Scalability) перевіряється через навантажувальні тести, у яких кількість підключених сенсорів послідовно збільшується до 1 000 пристроїв без погіршення затримки та втрат даних.

По-друге, кожне рішення має супроводжуватися планом валідації та верифікації. Для безпеки (Security) це означає проведення аудиту кодової бази та тестування на проникнення не рідше разу на квартал, а також використання сертифікованих бібліотек шифрування (TLS 1.3, AES-256). Зручність використання (Usability) перевіряють через проведення інтерв'ю з кінцевими користувачами й аналіз показників успішного виконання типових сценаріїв (task success rate ≥ 90 %). Сумісність (Compatibility) підтверджується результати інтеграційних тестів із різними платформами (Linux, Windows, Android) та протоколами обміну даними (MQTT v3.1.1, REST API v2).

По-третє, важливо закласти механізми моніторингу та зворотного зв'язку в експлуатаційному середовищі. Усі ключові показники (надійність, продуктивність, безпека) мають автоматично логуватися та виводитися в єдину панель метрик, яка дозволить оперативно реагувати на відхилення.

Наприклад, у разі зростання середнього часу відповіді вище 150 мс або виявлення понад 10 невдалих спроб автентифікації на день система мусить надсилати тривожне повідомлення адміністраторам.

Нарешті, необхідно передбачити механізм безперервного поліпшення платформи. Після кожного виходу нової версії чи значного оновлення компонентів проводяться повторні тести за вищевказаними метриками, а результати аналізуються з точки зору дотримання нефункціональних вимог.

Такий підхід гарантує, що з часом система зберігатиме необхідний рівень якості, масштабованості та безпеки, а також буде гнучко адаптуватися до змін як у середовищі експлуатації, так і в нормативних вимогах.

2.3 Архітектура системи та взаємодія компонентів

Арітектура системи визначає структурні компоненти програмно-апаратного комплексу та зв’язки між ними з погляду реалізації вимог і виконання функцій. Зазвичай виділяють кілька рівнів:

- фізичний (синтез сенсорів і виконавчих пристроїв), мережевий (транспорт і протоколи обміну даними);
- обчислювальний (модулі обробки й збереження інформації);
- прикладний (інтерфейси та сервіси для користувача).

Ключовими принципами побудови архітектури є модульність — коли кожен компонент виконує окрему чітко визначену функцію, та інтероперабельність — здатність різних модулів обмінюватися повідомленнями за єдиними контрактами інтерфейсів.

Взаємодія компонентів у такій архітектурі забезпечується через формалізовані канали передачі даних:

- REST- або MQTT-запити;
- веб-сокети;
- шину повідомлень.

Дані канали і протоколи гарантують узгодженість форматів і часових характеристик обміну.

Для мінімізації зв’язності (coupling) і підвищення внутрішньої згуртованості (cohesion) застосовуються шаблони «продюсер-споживач» або «видавець-підписник». Кожен компонент декларує свої публічні API, а всі побічні комунікації здійснюються через проміжне програмне забезпечення (middleware), яке також відповідає за маршрутизацію, масштабування та безпеку повідомлень.

Архітектура кіберфізичної системи контролю параметрів життєзабезпечення передбачає взаємодію між фізичними сенсорами, модулем збору та обробки даних, системою збереження та користувацьким інтерфейсом. В умовах обмеженого середовища реалізації, в межах дипломної роботи проєктується повноцінна архітектура системи, а реалізується — прототип із симуляцією сенсорних даних.

Загальна архітектура системи включає такі основні компоненти:

Сенсорний модуль призначений для збору ключових параметрів життєзабезпечення, таких як температура повітря, вологість, рівень вуглекислого газу тощо. У повноцінній системі для цього використовуються відповідні фізичні сенсори, зокрема DHT22 і MQ135. У прототипі ж ці дані формуються програмно для імітації роботи реальних пристроїв.

Мікроконтролерний модуль виконує функції прийому інформації з сенсорів, її попередньої обробки та тимчасового зберігання в буфері. У реальній системі цю роль може виконувати апаратна платформа на кшталт Raspberry Pi або ESP32. У межах прототипу зазначений модуль реалізовано у вигляді Python-скрипту, що відповідає за логіку отримання та обробки вхідних даних.

Модуль обробки даних відповідає за глибший аналіз отриманих параметрів. Його завдання — виявлення відхилень від визначених порогових значень та формування відповідних попереджень. Важливо, що обробка інформації здійснюється локально, що сприяє підвищенню рівня приватності.

Система збереження даних забезпечує ведення історії вимірювань. У прототипі використовується локальна база даних SQLite, яка не потребує постійного мережевого підключення. У більш розширеній версії системи можлива інтеграція з хмарними сховищами для додаткової зручності та масштабованості.

Інтерфейс користувача реалізує можливість доступу до даних системи. Він дозволяє переглядати актуальні значення параметрів, аналізувати їхню динаміку та отримувати повідомлення про потенційно небезпечні ситуації. У прототипі інтерфейс побудовано за допомогою бібліотек tkinter або ttkbootstrap.

Модуль захисту інформації відіграє важливу роль у забезпеченні конфіденційності даних. Для цього використовуються базові механізми шифрування під час зберігання та доступу до інформації. У прототипі дана функція реалізована з використанням бібліотек cryptography або hashlib.

Таким чином, розроблена архітектура демонструє можливість створення як повноцінної функціональної системи, так і спрощеного прототипу, адаптованого

для цілей дипломної роботи. Це дозволяє наочно продемонструвати принципи роботи системи навіть без залучення реальних сенсорних пристроїв.

На рисунку 2.1 представлено логічну структуру взаємодії основних компонентів.

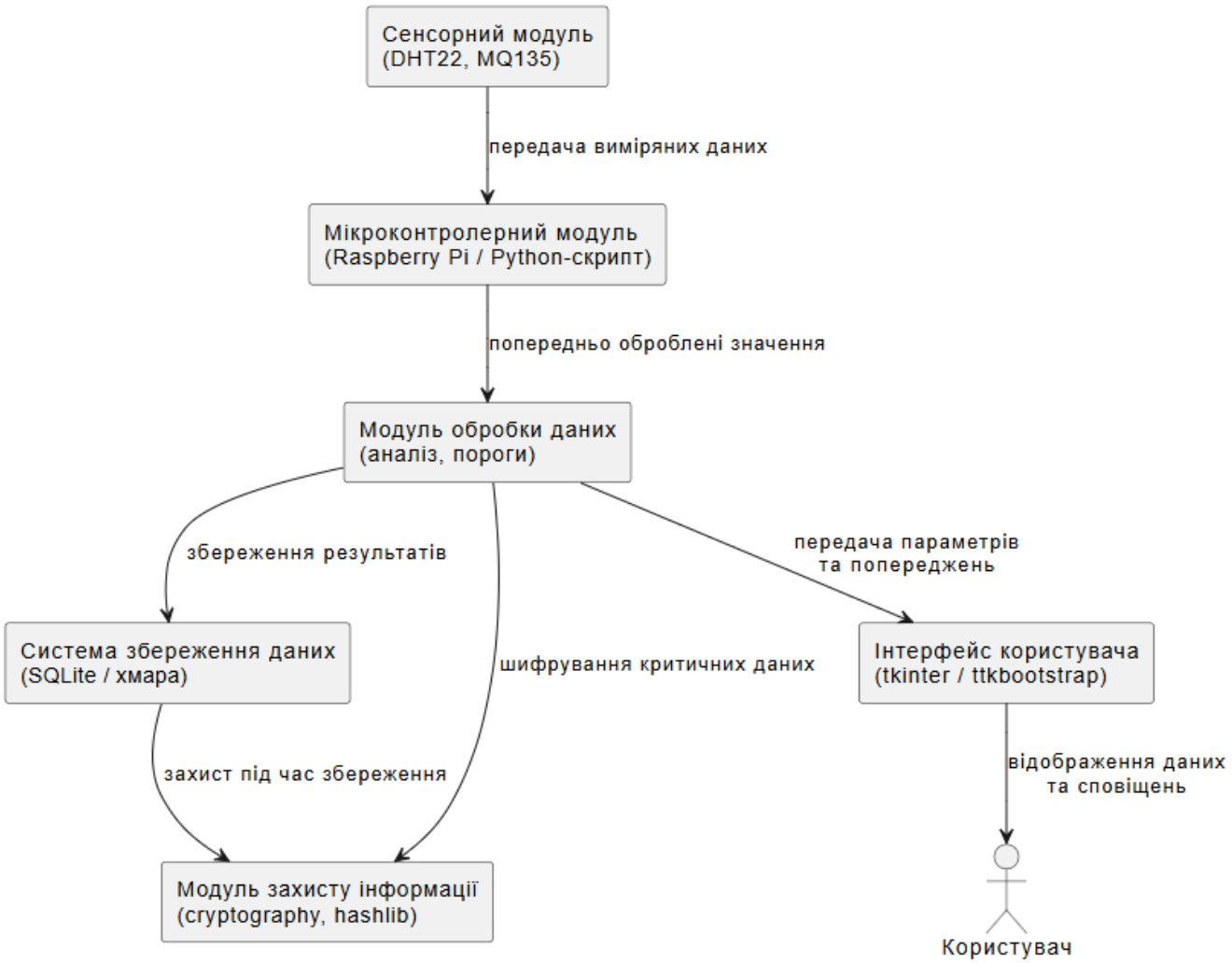


Рисунок 2.1 – Архітектура системи контролю параметрів життєзабезпечення

Сенсорні пристрої здійснюють первинний збір даних, які передаються на платформу (мікроконтролер або комп’ютер). Далі значення обробляються програмним модулем, зберігаються у базі даних, і на основі збереженої інформації здійснюється виведення даних в інтерфейсі користувача, а також, за потреби, ініціюється відповідна реакція або сповіщення. Всі взаємодії реалізовано через чітко визначені потоки даних, що забезпечують логіку роботи системи в реальному часі.

У складі повноцінної кіберфізичної системи контролю параметрів життєзабезпечення сенсорні модулі та мікроконтролерна платформа є базовими апаратними компонентами. Саме вони відповідають за безперервний моніторинг середовища, передачу даних до центрального модуля обробки та ініціацію подальших дій у разі виявлення відхилень. Для моніторингу мікроклімату, здоров'я чи технічного середовища доцільним є використання таких сенсорів, зазначених в таблиці 2.3:

Таблиця 2.3 – Порівняльна таблиця сенсорів для моніторингу параметрів [12]

Параметр	Пристрій	Інтерфейс	Призначення
Температура та вологість	DHT11 / DHT22	1-Wire / Digital	Моніторинг клімату приміщення
Рівень CO_2	MQ-135 / MH-Z19B	Analog / UART	Оцінка якості повітря
Пульс, SpO_2	MAX30100 / MAX30102	I^2C	Контроль стану здоров'я людини
Рівень руху	HC-SR501 (PIR sensor)	Digital	Виявлення присутності / активності
Освітлення	BH1750	I^2C	Контроль освітленості приміщення

Для зчитування даних із сенсорів, їх обробки та подальшої передачі потрібна обчислювальна платформа, тому в таблиці 2.4 наведено перелік популярних рішень.

Таблиця 2.4 – Характеристики мікроконтролерів для реалізації системи [13], [14]

Пристрій	Характеристики	Переваги
Raspberry Pi	Повноцінний одноплатний комп’ютер, Linux, GPIO	Підтримка Python, Wi-Fi, HDMI, USB
Пристрій	Характеристики	Переваги
Arduino Uno	Простий мікроконтролер із відкритим кодом	Велика спільнота, легка інтеграція
ESP32	Компактний мікроконтролер із вбудованим Wi-Fi	Низьке енергоспоживання, Bluetooth

З огляду на обмеження доступного обладнання, у межах даної дипломної роботи реалізовується лише прототип системи, що імітує роботу сенсорів у середовищі Python за допомогою:

- генерації даних із заданими параметрами (функція random.uniform);
- моделювання порушень: періодичне перевищення порогових значень для тестування реакції системи;
- можливості налаштовувати "діапазон значень" через інтерфейс користувача або конфігураційний файл.

Для забезпечення взаємодії між сенсорними модулями, мікроконтролерами та програмною частиною кіберфізичної системи застосовуються різні комунікаційні протоколи та інтерфейси (табл. 2.5). Вибір залежить від типу сенсорів, обчислювальної платформи, вимог до швидкості, надійності та енергоспоживання.

Таблиця 2.5 – Порівняння інтерфейсів для підключення сенсорів [15]

Інтерфейс	Принцип роботи	Переваги	Приклади використання
I²C	Послідовний, дволінійний	Підключення кількох пристроїв, простота	MAX30102, BH1750
1-Wire	Один провід для передачі	Мінімум дротів, простота для сенсорів	DHT11 / DHT22
GPIO	Базові цифрові входи/виходи	Простота підключення, програмна гнучкість	PIR HC-SR501, реле
SPI	Швидкий послідовний	Висока швидкість, надійність	Дисплеї, EEPROM
UART	Асинхронна передача	Простота реалізації, підтримка Bluetooth	MH-Z19B, ESP32 Bluetooth

У повнофункціональних системах також можуть використовуватись мережеві протоколи, зазначені в таблиці 2.6.

У межах дипломної роботи реалізовано прототип системи, який працює локально без фізичних сенсорів та бездротових з’єднань. Протоколи симулюються програмно:

- емуляція зчитування даних через програмні функції;
- обробка всередині скрипту Python без зовнішніх підключень;
- замість фізичного UART або I²C використовується передача даних між модулями через об'єкти у пам’яті.

Модуль обробки даних є ключовим елементом кіберфізичної системи, що забезпечує аналіз інформації, отриманої від сенсорних пристроїв, виявлення відхилень від заданих норм і ініціювання відповідних реакцій. Даний модуль формує "інтелектуальне ядро" системи, яке приймає рішення в реальному часі.

Таблиця 2.6 – Мережеві протоколи, що застосовуються у кіберфізичних системах [16]

Протокол	Призначення	Особливості реалізації
MQTT	Публікація-підписка, IoT-орієнтований	Працює навіть при нестабільному з'єднанні
HTTP/HTTPS	Класичні веб-запити	Простий в реалізації, інтегрується з API
Bluetooth	Бездротова локальна передача	Підтримується ESP32, зручно для мобільних
Wi-Fi	Повноцінне мережеве з'єднання	Потрібен TCP/IP-стек, реалізується в ESP32
LoRa	Дальнє бездротове з'єднання з низьким споживанням	Для сільського господарства, моніторингу

Основні функції модуля обробки даних охоплюють кілька ключових напрямів. Насамперед це фільтрація та нормалізація. Вхідні дані часто є зашумленими або нестабільними, тому застосування простих методів фільтрації, наприклад ковзного середнього, та нормалізації показників є доцільним для забезпечення коректного подальшого аналізу.

Ще однією важливою функцією є пороговий аналіз. Для кожного параметра встановлюються допустимі межі, і якщо виміряне значення виходить за ці межі, система формує подію з високим пріоритетом.

У разі фіксації відхилення в системі відбувається обробка подій. Це включає генерацію повідомлень, зміну стану інтерфейсу, наприклад, шляхом відображення візуального попередження або зміни кольору, а також ініціацію різних реакцій, таких як звукові сигнали, запис до журналу чи автоматичне вмикання пристроїв.

Крім того, важливою складовою є логування та аналітика. Оброблені дані передаються до бази даних, де накопичуються і можуть аналізуватись у

довгостроковій перспективі. Це дозволяє будувати графіки, виявляти тренди та формувати рекомендації.

У межах дипломної роботи реалізується прототип. Модуль обробки створено у вигляді Python-функцій, які приймають згенеровані псевдодані, виконують порівняння з фіксованими порогами, наприклад температура понад 30°C, виводять повідомлення в інтерфейсі через ttk.Label з червоним кольором, записують інформацію в базу даних SQLite, а також формують лог-файли або повідомлення про події без фізичної реакції. Такий підхід дозволяє перевірити основну логіку реагування навіть без наявності фізичних сенсорів і виконавчих пристроїв.

Приклад сценарію демонструє цю концепцію. Імітований датчик температури повертає значення 31.5°C. Система порівнює його з пороговим значенням 30.0°C. У результаті в інтерфейсі з'являється червоне попередження "Перевищення температури", подія фіксується в журналі SQLite, а також створюється позначка для графіка.

Таким чином, модуль обробки даних і реакції на події перетворює систему з пасивного реєстратора на активного учасника моніторингу, здатного самостійно приймати рішення. Його реалізація в прототипі є ключовим елементом демонстрації життєздатності архітектури.

У підсумку було детально розглянуто архітектуру кіберфізичної системи контролю параметрів життєзабезпечення та описано принципи взаємодії її компонентів. Обрана архітектура відповідає сучасним стандартам модульності та інтероперабельності, що дозволяє кожному компоненту виконувати чітко визначені функції. Взаємодія між фізичними сенсорами, модулем збору і обробки даних, системою збереження та користувацьким інтерфейсом забезпечується стандартними протоколами, такими як MQTT та REST, що гарантує надійну передачу й узгодженість даних.

Прототип із симуляцією сенсорних даних підтвердив правильність обраної концепції, забезпечивши логічну й структуровану основу для подальшого масштабування. Запропонована архітектура дозволяє гнучко розширювати систему шляхом додавання нових сенсорних вузлів, апаратних платформ або програмних

модулів без суттєвої перебудови. Подальші дослідження можуть зосередитись на інтеграції додаткових механізмів безпеки та розширенні функціональних можливостей системи для максимальної відповідності реальним умовам експлуатації.

2.4 Обґрунтування вибору засобів реалізації

Для побудови прототипу кіберфізичної системи контролю параметрів життєзабезпечення з підтримкою приватності було обрано набір інструментів, що дозволяє ефективно реалізувати основні компоненти системи — збір, обробку, збереження, візуалізацію та захист даних — у середовищі із обмеженими ресурсами без залучення фізичних сенсорів. Нижче наведено обґрунтування вибору кожного з основних компонентів та інструментів реалізації.

Основною мовою реалізації обрано Python, оскільки вона поєднує простоту синтаксису, широке поширення у сфері розробки IoT- та CPS-рішень, а також має потужну екосистему бібліотек для роботи з даними, інтерфейсами, шифруванням та сенсорами. Python підтримується як на Windows, так і на одноплатних комп’ютерах (Raspberry Pi), що дозволяє масштабувати прототип до реальної системи без значних змін у коді.

У таблиці 2.7 наведено основні бібліотеки, використані у реалізації системи, із зазначенням їх функціонального призначення.

Таблиця 2.7 – Бібліотеки Python, використані в реалізації системи

№	Бібліотека	Призначення
1	random, time	Симуляція сенсорних даних, контроль періодичності збору значень
2	sqlite3	Збереження історії вимірювань у локальній базі даних
3	tkinter, ttkbootstrap	Побудова графічного інтерфейсу користувача
4	cryptography, hashlib	Реалізація шифрування даних, контроль доступу

Кінець таблиці 2.7

№	Бібліотека	Призначення
5	matplotlib (опціонально)	Побудова графіків зміни параметрів у динаміці

SQLite обрано як засіб для збереження даних завдяки ряду її суттєвих переваг, особливо актуальних для реалізації прототипу кіберфізичної системи моніторингу параметрів життєзабезпечення. SQLite є легкою, швидкою та самодостатньою системою управління базами даних, що не потребує встановлення окремого сервера та додаткових налаштувань. Вона ідеально підходить для невеликих додатків і вбудованих рішень, де критично важливі продуктивність, швидкість доступу до даних та мінімальний час відгуку [38].

Завдяки відсутності окремого серверного процесу SQLite забезпечує миттєвий доступ до інформації, що дозволяє системі швидко реагувати на зміну параметрів у реальному часі. Крім того, використання єдиного файлу бази даних значно спрощує процеси резервного копіювання, перенесення та інтеграції рішення в інші середовища.

Таким чином, SQLite є оптимальним вибором для швидкої розробки й тестування, забезпечуючи достатній рівень надійності, швидкості та простоти експлуатації, особливо у контексті невеликих та середніх за розміром додатків, таких як розроблений прототип системи моніторингу.

Оскільки дипломна робота реалізується у вигляді прототипу без фізичних сенсорів, було прийнято рішення використати генерацію псевдовипадкових значень у допустимих діапазонах (наприклад, температура 20–40 °C) з періодичним додаванням умовно «аварійних» значень (наприклад, температура > 50 °C) для перевірки реакції системи. Такий підхід дозволив перевірити логіку обробки критичних подій без потреби у додатковому обладнанні.

Для забезпечення конфіденційності даних реалізовано базові механізми шифрування — зокрема, за допомогою бібліотеки cryptography (AES-шифрування). Дані шифруються перед збереженням у базі або перед передачею в інтерфейс.

Бібліотека hashlib також використовується для створення хешів, що дозволяє демонструвати контроль цілісності.

Інтерфейс реалізовано за допомогою бібліотеки Tkinter, стандартного інструменту для створення графічного інтерфейсу користувача у складі мови програмування Python. Tkinter є простою, гнучкою та доступною бібліотекою, що дозволяє розробляти інтуїтивно зрозумілі інтерфейси з підтримкою основних графічних елементів: кнопок, полів вводу, перемикачів, списків, меню тощо. Основними перевагами Tkinter є її стандартна підтримка в усіх дистрибутивах Python, простота використання, кросплатформність і можливість швидкого прототипування [22].

Для створення сучасного та візуально привабливого дизайну інтерфейсу додатково використовується бібліотека ttkbootstrap. Ця бібліотека доповнює можливості Tkinter, пропонуючи набір стилізованих компонентів і готових шаблонів, що ґрунтуються на популярних CSS-фреймворках (наприклад, Bootstrap). Вона дозволяє значно спростити оформлення застосунку, забезпечуючи адаптивність і єдиний стиль графічних елементів, що відповідають сучасним тенденціям дизайну інтерфейсів. Завдяки цьому застосунок набуває професійного вигляду, залишаючись зручним і простим у користуванні.

Завдяки використанню цієї бібліотеки вдалося реалізувати інтерфейс у вигляді вікна застосунку, що дозволяє переглядати поточні значення параметрів, кольорові індикатори перевищення допустимих меж, сповіщення про критичні події, а також графіки зміни параметрів у часі за потреби.

Вибрані засоби мають низку важливих переваг. По-перше, вони є доступними — усі компоненти безкоштовні, відкриті та кросплатформенні. По-друге, забезпечують гнучкість, оскільки легко адаптуються до різних сценаріїв використання, таких як домашній, медичний чи екологічний моніторинг. По-третє, ці засоби відзначаються розширюваністю, що дозволяє масштабувати систему для роботи в реальному середовищі з фізичними сенсорами. Крім того, вони повністю відповідають тематиці роботи, адже підтримують захист приватності, спрощують

верифікацію на персональному комп'ютері та відкривають можливості для проведення експериментів.

Отже, обраний стек технологій дає змогу створити повноцінний прототип кіберфізичної системи, яка задовольняє як функціональні, так і нефункціональні вимоги. Водночас він забезпечує наочну демонстрацію принципів захисту приватності та зручності у використанні.

2.5 Механізми захисту приватності в обраній архітектурі

Забезпечення приватності даних у кіберфізичній системі контролю параметрів життєзабезпечення є критично важливим завданням. Це зумовлено тим, що така система обробляє потенційно чутливу інформацію, зокрема дані про стан навколишнього середовища або фізіологічні параметри людини. Особливо важливо враховувати цей аспект у медичних та побутових сценаріях використання, де витік інформації може спричинити порушення конфіденційності або навіть створити загрозу безпеці користувачів.

Архітектура реалізованого прототипу побудована з урахуванням сучасних принципів захисту приватності. Застосовано підхід Security by Design, який передбачає інтеграцію засобів захисту на всіх етапах проєктування системи. Використано принцип Data Minimization — обробці підлягають лише ті дані, які є необхідними для забезпечення коректного функціонування. Крім того, реалізовано концепцію Local-first Privacy, що передбачає збереження та обробку інформації виключно в локальному середовищі.

Основним елементом захисту приватності у цій системі є використання локальної бази даних SQLite для зберігання усіх зібраних даних. Це рішення виключає необхідність передавання інформації до хмарних чи зовнішніх сервісів. Такий підхід істотно знижує ризики перехоплення даних під час їх передачі мережею та мінімізує ймовірність несанкціонованого доступу ззовні.Для захисту вмісту бази даних та тимчасових буферів використовуються алгоритми симетричного шифрування AES (Advanced Encryption Standard). Реалізація цього

процесу здійснюється з використанням бібліотеки cryptography. Дані шифруються до моменту запису в базу, а розшифровуються лише під час доступу до них, що гарантує їхню конфіденційність.

З метою перевірки цілісності даних і запобігання несанкціонованим модифікаціям застосовується хешування. Для цього використовується бібліотека hashlib, яка формує хеші критичних значень. Це дозволяє оперативно виявляти спроби зміни або пошкодження інформації.

У межах прототипу реалізовано базову систему контролю доступу. Передбачено авторизацію користувача під час запуску програми, що включає перевірку логіна та пароля. Ці облікові дані зберігаються в зашифрованому вигляді. У майбутньому функціонал може бути доповнений системою ролей (RBAC) для більш гнучкого управління правами доступу.

Особливу увагу приділено усуненню сторонніх мережевих підключень. Обробка усіх даних здійснюється локально без використання зовнішніх API та хмарних сервісів. Це дозволяє суттєво знизити ризик витоку конфіденційної інформації через інтернет-інтерфейси.

З метою аудиту та виявлення потенційних аномалій у системі передбачено ведення журналу подій. Всі критичні події, такі як перевищення встановлених порогових значень, спроби несанкціонованого доступу або системні збої, фіксуються у захищеному лог-файлі. Це забезпечує можливість подальшого аналізу та підвищує загальний рівень безпеки системи.Впроваджені механізми узгоджуються з рекомендаціями [15–16], зокрема з вимогами Zero Trust Architecture (ZTA), де кожен модуль системи розглядається як потенційно небезпечний і підлягає перевірці перед обробкою або передачею даних.

Також дотримано підходів:

- Confidentiality – забезпечено шифрування;
- Integrity – контроль хешами;
- Availability – система зберігає працездатність навіть за збою модулів або даних.

Таким чином, навіть у прототипі, розробленому без використання реального обладнання, реалізовано базові, але важливі принципи захисту приватності, що можуть бути масштабовані до промислового або медичного застосування. Застосовані методи забезпечують базовий рівень безпеки й одночасно демонструють архітектурну готовність системи до впровадження повноцінного захисту у майбутньому.

2.6 Висновок до другого розділу

В розділі було здійснено проєктування кіберфізичної системи контролю параметрів життєзабезпечення з урахуванням вимог до функціональності, надійності та приватності. На основі аналізу цільового середовища та очікувань користувачів сформульовано функціональні та нефункціональні вимоги до системи, які охоплюють ключові аспекти її роботи: від збору даних до візуалізації та захисту.

Розроблено архітектуру системи, яка включає сенсорні модулі, мікроконтролерну платформу, модуль обробки даних, систему збереження, користувацький інтерфейс та засоби захисту інформації. Для кожного компонента визначено його роль у загальному функціонуванні системи, а також взаємозв’язки, що забезпечують логіку роботи в реальному часі.

Обґрунтовано вибір засобів реалізації, що були використані для створення прототипу системи: мова програмування Python, бібліотеки для шифрування, візуалізації та збереження даних. Враховано фактори доступності, простоти розгортання, масштабованості та відповідності вимогам до приватності. Замість фізичних сенсорів у прототипі реалізовано генерацію псевдоданих, що дозволяє відтворити критичні ситуації без додаткового обладнання.

Окрему увагу приділено механізмам захисту приватності: реалізовано шифрування даних, хешування для перевірки цілісності, локальне збереження та обмеження доступу. Застосовано принципи Security by Design та Zero Trust, що підвищує рівень безпеки запропонованої архітектури навіть на етапі прототипу.

Таким чином, розділ 2 сформував комплексну основу для подальшої реалізації інформаційної системи в розділі 3, заклавши як логічну структуру, так і технічні рішення, необхідні для її функціонування в реальному або змодельованому середовищі.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Налаштування середовища програмування та бібліотек

Для реалізації кіберфізичної системи моніторингу параметрів життєзабезпечення було обрано мову програмування Python 3.11 [23], яка забезпечує широкий спектр бібліотек для роботи з графікою, потоками, шифруванням, мережею та базами даних. Проєкт реалізовано у вигляді багатомодульного застосунку з використанням принципів модульності, повторного використання коду та розділення логіки.

Основні етапи налаштування середовища передбачають створення чіткої структури проєкту з відповідними підкаталогами. Це дозволяє організувати компоненти системи логічно та зручно для подальшої розробки та обслуговування.

Зокрема, передбачено каталог sensors, у якому зосереджені модулі для роботи з сенсорами температури та вологості. Окремий каталог data_processor відповідає за обробку даних, перевірку їх відповідності допустимим межам і ведення журналу подій. Для забезпечення безпеки інформації виділено каталог encryption, де розміщується модуль шифрування.

Обробка тривожних подій здійснюється через модулі, розміщені в каталозі alerts, тоді як за взаємодію з користувачем відповідає каталог ui, який містить усі компоненти інтерфейсу. Нарешті, каталог mqtt_client реалізує функціонал обміну даними за протоколом MQTT, що забезпечує інтеграцію системи з іншими пристроями та сервісами.

Для реалізації проєкту на початковому етапі було сформовано логічну структуру папок і файлів, що відповідає архітектурі компонентів системи. Створення цієї структури здійснювалося за допомогою стандартних команд PowerShell, таких як mkdir для створення папок та ni (new-item) для створення файлів. Наприклад, команда mkdir src\sensors створювала підкаталог для модулів сенсорів, а команда ni src\sensors\temperature_sensor.py -ItemType File — файл для реалізації логіки зчитування температури.

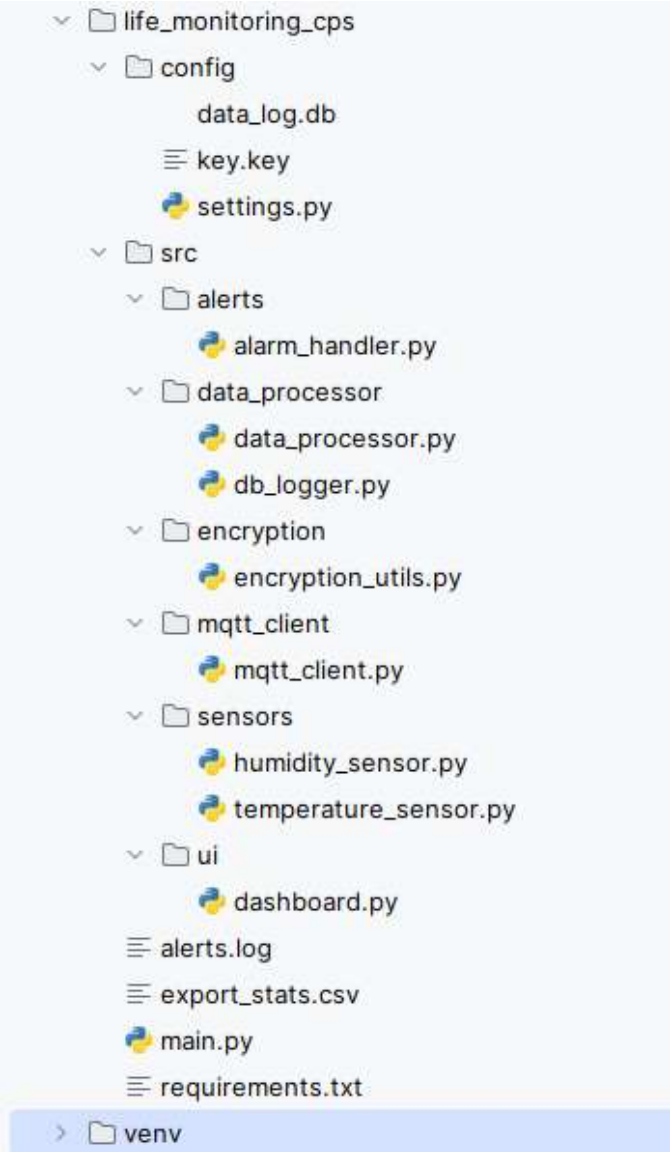


Рисунок 3.1 - Структура проекту

Усі папки були організовані в кореневій директорії src та відповідали окремим функціональним блокам системи. Зокрема, у каталозі sensors розміщено емульовані сенсори температури та вологості. Папка data_processor містить модулі для обробки даних та запису їх у базу. В каталозі encryption зосереджено модуль шифрування. Папка alerts відповідає за обробку тривожних ситуацій. Каталог mqtt_client забезпечує мережеву передачу даних, а в папці ui реалізовано графічний інтерфейс користувача.

Окрім цього, була створена папка config, де зберігається згенерований ключ для симетричного шифрування (файл key.key) та база даних data_log.db. Файл main.py, що розташований у кореневій директорії, виконує роль основної точки

запуску — він ініціалізує інтерфейс та запускає моніторинговий потік. Для зберігання локальних журналів тривог створено файл alerts.log, який очищується під час кожного нового запуску системи.

Запропонована структура забезпечує чітке розділення відповідальностей між компонентами, що значно полегшує підтримку та подальший розвиток проєкту. Для зручності аналізу й узагальнення функціональної організації програмного забезпечення системи у таблиці 3.1 наведено опис основних каталогів, їх призначення та вміст. Такий підхід дозволяє чітко відокремити логіку окремих компонентів, спрощує навігацію в коді та сприяє масштабованості рішення.

Таблиця 3.1 – Структура каталогів програмного забезпечення системи

Каталог / файл	Призначення
src/sensors/	Містить модулі емуляції сенсорів температури та вологості
src/data_processor/	Обробка отриманих даних, перевірка меж допустимих значень, логування
src/encryption/	Реалізація алгоритмів шифрування переданих і збережених даних
src/alerts/	Виявлення відхилень від норми, генерація повідомлень про тривогу
src/ui/	Графічний інтерфейс користувача для відображення параметрів і тривог
src/mqtt_client/	Модулі для обміну даними з брокером MQTT
src/config/	Зберігання конфігураційних файлів: ключів шифрування (key.key), бази даних
main.py	Основний скрипт запуску системи, ініціалізація інтерфейсу, запуск потоків
alerts.log	Локальний журнал виявлених тривог, який оновлюється при кожному запуску
data_log.db	Локальна база даних для зберігання історії зчитаних значень параметрів

На етапі ініціалізації проєкту було створено ізольоване програмне середовище. Для цього застосовано інструмент venv, який є стандартним засобом Python для створення та керування віртуальними середовищами. Це дозволило забезпечити незалежність розробки від глобальних налаштувань системи та уникнути конфліктів між бібліотеками. Процес створення середовища наведено на рисунку 3.2.

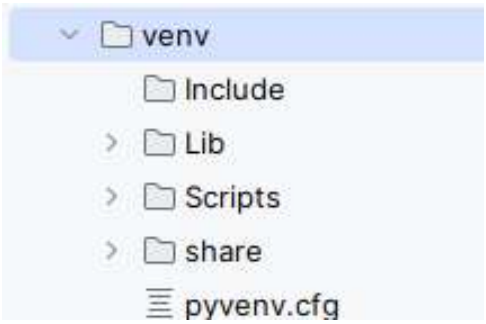


Рисунок 3.2 - структура віртуального середовища

Віртуальне середовище дозволяє локально інсталювати залежності проєкту — бібліотеки, інструменти й модулі — без впливу на глобальну систему Python, встановлену на комп'ютері. Це забезпечує сумісність між різними проєктами, дозволяє уникати конфліктів між версіями пакетів і спрощує перенесення застосунку на інші машини.

Команда python -m venv venv створює нове віртуальне середовище у папці venv, після чого воно може бути активоване та використане для встановлення всіх необхідних залежностей проєкту.

Після створення віртуального середовища необхідно його активувати, щоб усі встановлені бібліотеки зберігалися локально в межах проєкту, а не в глобальному середовищі системи. Активація здійснюється по-різному залежно від операційної системи. У таблиці 3.2 наведено приклади команд для Windows, Linux та macOS.

Таблиця 3.2 – Команди створення та активації віртуального середовища

Операція	Windows (PowerShell / CMD)	Linux / macOS (bash/zsh)
Створення середовища	python -m venv venv	python3 -m venv venv
Активація середовища (PowerShell)	.\venv\Scripts\Activate.ps1	—
Активація середовища (CMD)	.\venv\Scripts\activate.bat	—
Активація середовища (bash/zsh)	—	source venv/bin/activate
Деактивація середовища	deactivate	deactivate

У деяких системах для виконання .ps1-скриптів у PowerShell може знадобитися надати дозвіл на запуск сценаріїв. Це здійснюється за допомогою команди Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass, яка тимчасово дозволяє виконання скриптів у межах поточного сеансу.

На наступному етапі було здійснено встановлення бібліотек, перелік яких зберігається у файлі requirements.txt. Для ефективного керування залежностями проєкту та забезпечення відтворюваності середовища виконання використовується саме цей файл. У ньому міститься повний список необхідних бібліотек Python разом із зазначенням їх точних версій. Завдяки цьому підходу можна легко розгорнути ідентичне середовище на іншому комп’ютері або сервері, що гарантує сумісність між усіма компонентами системи.

Створення файлу requirements.txt виконується за допомогою команди pip freeze > requirements.txt. Вона дозволяє зафіксувати всі поточні пакети, встановлені у віртуальному середовищі, і зберегти їх до текстового файлу. Для подальшої інсталяції залежностей на іншому пристрої достатньо виконати команду pip install -r requirements.txt, яка автоматично встановить всі необхідні пакети відповідних версій.

Таблиця 3.3 - В проєкті використані наступні бібліотеки

Назва	Призначення
ttkbootstrap	Стилізований інтерфейс на основі Tkinter [34]
matplotlib	Побудова графіків у реальному часі
paho-mqtt	Публікація MQTT-повідомлень
cryptography	Шифрування даних (Fernet AES)
sqlite3	Вбудована СУБД для зберігання логів
threading	Паралельне виконання (моніторинг у фоновому потоці)
tkinter.messagebox	Вивід повідомлень у вікнах GUI

Бібліотека ttkbootstrap розширює стандартний модуль tkinter.ttk, додаючи теми у стилі Bootstrap. Це дозволяє створювати сучасні та адаптивні графічні інтерфейси з привабливим дизайном без необхідності використання зовнішніх CSS чи JavaScript. У межах цього проєкту вона застосовується для оформлення вікон і елементів керування інтерфейсу користувача, включаючи таблиці, кнопки, вкладки тощо.

Бібліотека matplotlib є потужним інструментом для візуалізації даних. Вона дає змогу створювати графіки, гістограми, діаграми та інші види візуального представлення інформації. У даній системі вона використовується для побудови графіків зміни температури та вологості в реальному часі, що допомагає користувачеві оперативно оцінювати динаміку параметрів.

Клієнтська бібліотека paho-mqtt від Eclipse Foundation призначена для реалізації протоколу MQTT, який є популярним у IoT-системах для обміну легкими повідомленнями між пристроями. У цьому проєкті вона використовується для публікації сенсорних даних до брокера MQTT і отримання керуючих команд.

Бібліотека cryptography є сучасним рішенням для шифрування даних. Вона підтримує як симетричні алгоритми (наприклад, AES, Fernet), так і асиметричні

(RSA, ECC). У системі використано модуль Fernet для симетричного шифрування переданих по мережі повідомлень і для захисту конфіденційних даних у базі.

Модуль sqlite3 забезпечує роботу з вбудованою реляційною базою даних у Python без необхідності встановлення окремого сервера. Він використовується для збереження логів сенсорних даних, повідомлень про тривоги та історичних показників у форматі SQL-таблиць. Це дозволяє легко виконувати запити, фільтрацію та агрегацію даних.

Модуль threading надає можливість створення багатопоточних програм. У межах цього проєкту він застосовується для організації фонової обробки даних сенсорів, що дає змогу уникнути блокування графічного інтерфейсу та забезпечити зручну взаємодію користувача з програмою.

Модуль tkinter.messagebox використовується для відображення стандартних вікон повідомлень, включаючи інформаційні, попереджувальні та повідомлення про помилки. Це дозволяє оперативно інформувати користувача про критичні події або тривоги під час роботи системи.

Проєкт реалізовано з урахуванням кросплатформенності. Усі компоненти підтримуються як у середовищі Windows, так і Linux, оскільки базуються на стандартних або широко підтримуваних засобах Python.

3.2 Реалізація програмного коду

Реалізована система складається з ряду незалежних, проте взаємопов'язаних модулів, які відповідають за послідовні етапи функціонування кіберфізичної системи — збір, обробку, збереження, передачу та візуалізацію даних (Додаток Б).

Дані збираються з емульованих сенсорів температури та вологості, реалізованих у модулях temperature_sensor.py та humidity_sensor.py. Сенсори створюють випадкові значення в межах заданих параметрів (рис. 3.3).

```
    import random
class TemperatureSensor:
```

```
def read_temperature(self):
return round(random.uniform(18.0, 26.0), 2)
```

Згенеровані значення зчитуються з інтервалом 2 секунди у фоновому потоці **monitor_loop**, що дозволяє не блокувати основний інтерфейс користувача.

Далі отримані дані передаються до модуля data_processor.py, де виконується перевірка на вихід за межі допустимих значень. Після аналізу повертається статус (“Норма”, “Низька температура”, “Висока вологість” тощо) та прапорець **alert** (рис. 3.4).

```
def process_temperature(self, value):
if value < 20.0:
return {“status”: “Низька температура”, “alert”: True}
```

У випадку, коли значення виходить за межі норми, спрацьовує модуль alarm_handler.py, який:

- виводить повідомлення у консоль;
- записує лог у файл alerts.log (додаток В);
- зберігає запис у базі даних.

Логування реалізується в локальний файл alerts.log (Додаток В) та у локальну базу даних SQLite через фунцію, реалізовану в файлі db_logger.py. Приклад SQL запиту на створення нового запису:

```
INSERT INTO logs (timestamp, parameter, value, status) VALUES
(?, ?, ?, ?)
```

Кожен запис містить кілька важливих полів. До них належать мітка часу, що фіксує момент реєстрації події, назва параметра, яка вказує, чи йдеться про температуру або вологість, числове значення відповідного показника, а також статус, що відображає поточний стан параметра.

Перш ніж передати дані мережею, вони шифруються через бібліотеку cryptography за допомогою алгоритму Fernet (AES). Ключ зберігається локально в config/key.key

Далі зашифровані або розшифровані дані передаються MQTT-клієнтом (paho-mqtt) на публічний брокер test.mosquitto.org. Повідомлення формуються у форматі JSON (рис. 3.5).

```
{
“temperature”: 22.5,
“temp_status”: “Норма”,
“humidity”: 48.2,
“hum_status”: “Норма”
}
```

MQTT забезпечує надійну доставку, мінімальне навантаження на мережу та підтримку різних рівнів якості обслуговування (QoS). У цьому проєкті MQTT використовується для обміну зашифрованими даними між сенсорними модулями, модулем обробки та інтерфейсом користувача, що дозволяє легко інтегрувати систему в IoT-середовище та підключати інші пристрої.

Всі значення, отримані від сенсорів, оновлюються у реальному часі в графічному інтерфейсі dashboard.py, побудованому на ttkbootstrap та matplotlib. НА екран виводяться наступні значення:

- числові значення параметрів;
- графіки (останні 20 значень);
- таблиця «Норма / Відхилення»;
- кнопки для аналізу статистики та експорту в CSV.

Для наочної візуалізації змін параметрів у часі в системі реалізовано побудову графіків температури та вологості у реальному часі.

Виведення графіків здійснюється за допомогою бібліотеки matplotlib, інтегрованої у графічний інтерфейс через FigureCanvasTkAgg. У модулі dashboard.py передбачено два окремі графічні блоки, кожен із яких відображає останні 20 зчитаних значень відповідного параметра (Додаток Д).

Побудова графіків відбувається при кожному оновленні даних, з очищенням попереднього зображення та додаванням нових значень.

Крім того, на графіках нанесено горизонтальні лінії, що позначають допустимі межі, що дозволяє користувачу миттєво визначати наявність відхилень. Така візуалізація підвищує зручність сприйняття інформації та дозволяє швидко реагувати на критичні зміни параметрів.

Отже, в межах цього етапу було реалізовано повний цикл обробки життєвих параметрів у рамках кіберфізичної системи. Система успішно виконує зчитування даних із сенсорів, обробляє їх у режимі реального часу, виявляє відхилення від норми та реагує на них через механізм тривог.

Дані шифруються, зберігаються у локальній базі даних та передаються у вигляді структурованих MQTT-повідомлень. Завдяки використанню паралельного потоку моніторингу та графічного інтерфейсу користувача забезпечено зручність взаємодії, наочність та безперервну роботу системи без блокування головного процесу. Така реалізація може бути масштабована для використання у сценаріях моніторингу життєвих показників у побутових, медичних чи промислових умовах.

На рисунку 3.5 зображено схему роботи додатку.

Після старту моніторингового циклу (A) відбувається одночасне зчитування даних із двох сенсорів (B), які передають значення у модулі обробки (C1, C2). Оброблені показники логуються в локальну БД (D1, D2), потім шифруються (E1, E2) і одразу ж розшифровуються (F1, F2) для подальшого використання в інтерфейсі та обробці тривог.

Якщо виявлено відхилення, спрацьовує модуль тривог (H1, H2), який записує повідомлення в лог-файл.

Паралельно результати відображаються в GUI (I → J), а структуроване повідомлення публікується через MQTT (K → L). Після затримки у 2 секунди (M) цикл повторюється.

Такий потік забезпечує безперервність моніторингу, своєчасну обробку та відображення даних, надійну передачу повідомлень та реагування на аномалії.

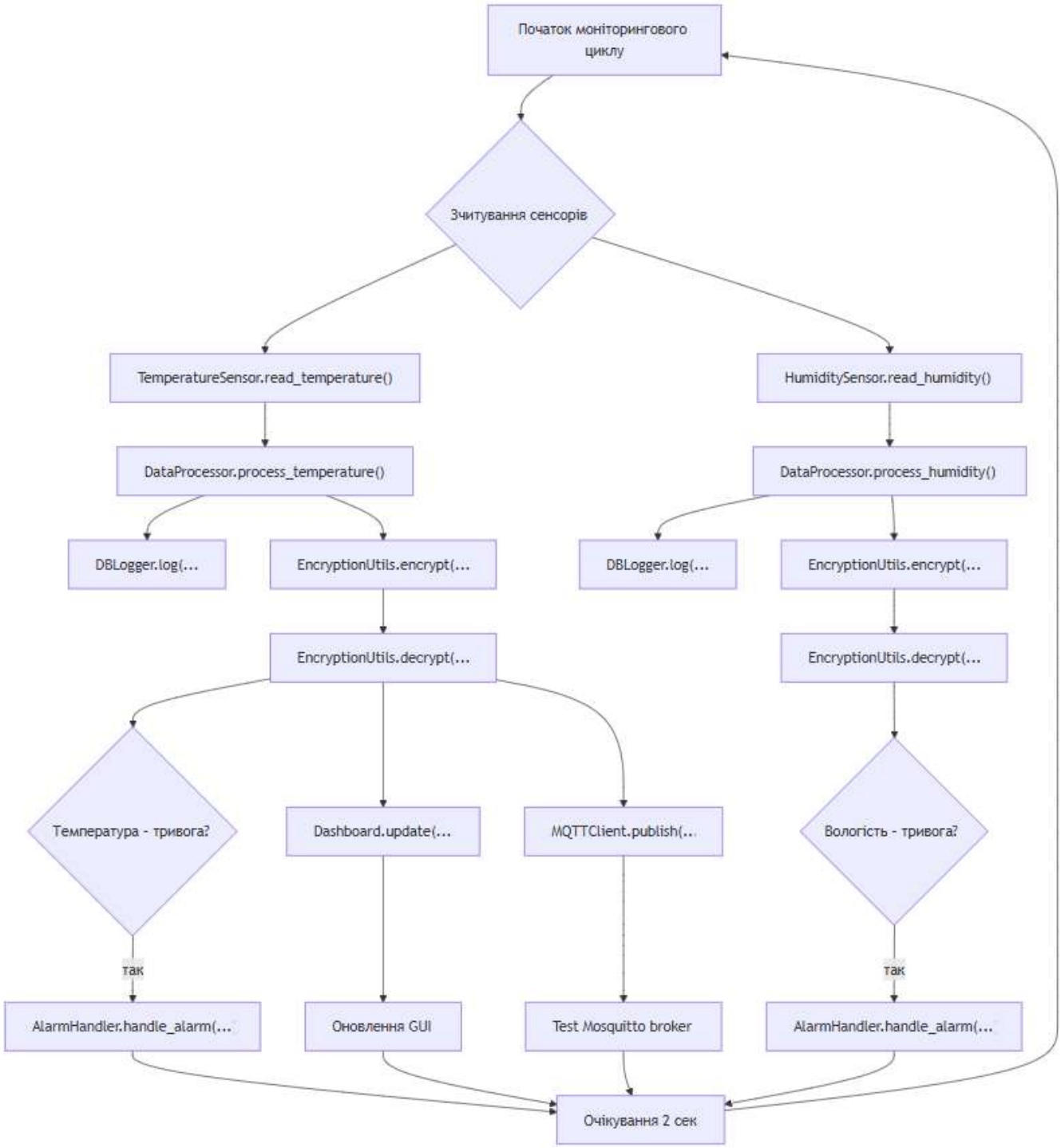


Рисунок 3.5 - Схема роботи додатку

3.3 Вбудовані засоби захисту даних у програмному рішенні

У проєкті реалізовано багаторівневий підхід до захисту інформації, який забезпечує комплексне охоплення всіх етапів обробки даних — від моменту їх зчитування сенсорними пристроями до кінцевого відображення в інтерфейсі користувача та публікації в хмарних сервісах. Такий підхід ґрунтується на сучасних стандартах та рекомендаціях у сфері кібербезпеки, включаючи принципи Security by Design та Zero Trust, що передбачають вбудовування механізмів захисту вже на етапах розробки програмної системи. Використання багаторівневої архітектури забезпечує захист на рівні фізичних пристроїв (сенсорів), каналів передачі інформації, проміжної обробки даних, зберігання та взаємодії з користувачем. Це дозволяє зменшити ризики несанкціонованого доступу, підміни, втрати чи спотворення інформації, гарантуючи водночас високу надійність та доступність системи в умовах потенційних атак чи технічних збоїв. На рисунку 3.6 наведено узагальнену схему багаторівневого підходу до захисту інформації в реалізованому програмному рішенні.

Для забезпечення конфіденційності повідомлень у системі застосовується модуль encryption_utils.py, побудований на основі алгоритму Fernet, який реалізує AES-128-CBC у комбінації з HMAC-SHA256. Під час першого запуску системи автоматично генерується випадковий ключ, що зберігається локально у файлі config/key.key.

Усі подальші операції шифрування та дешифрування виконуються за допомогою цього ключа. Такий підхід гарантує, що ключ ніколи не передається мережею і не може бути перехоплений. У разі необхідності передбачено можливість ручної або автоматичної ротації ключів шляхом заміни відповідного файлу та перегенерації ключа.

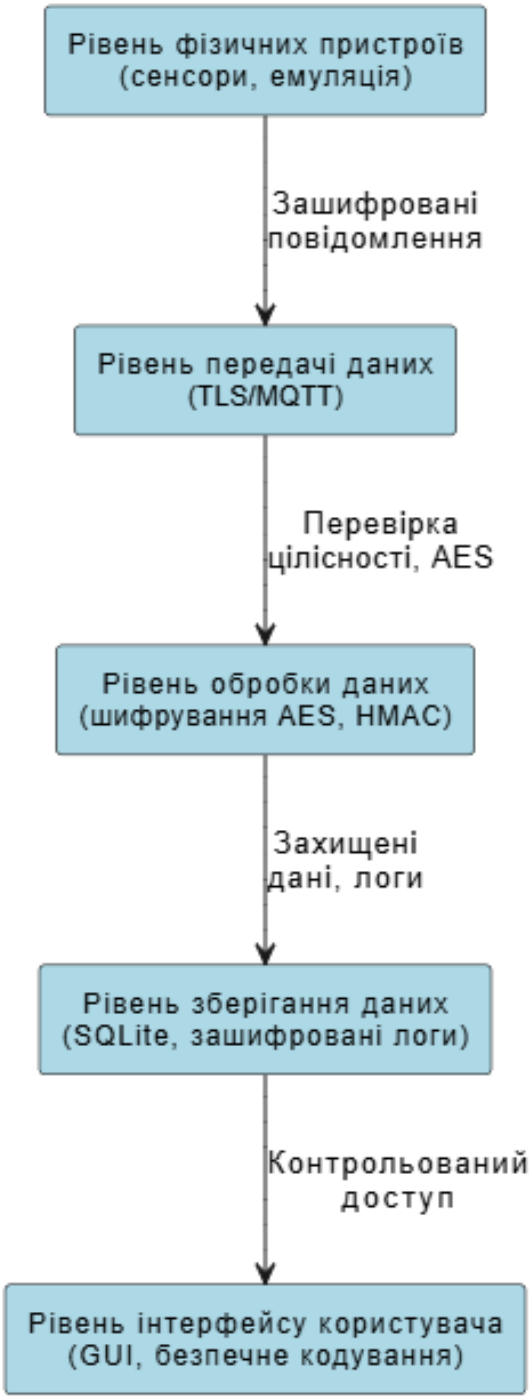


Рисунок 3.6 - Узагальнена схема багаторівневого підходу до захисту інформації

Додатковий рівень захисту забезпечується використанням захищених каналів зв’язку. Для обміну повідомленнями через MQTT використовується протокол TLS, що дозволяє шифрувати трафік на мережевому рівні. У класі MQTTClient реалізовано можливість підключення до брокера через порт 8883 із перевіркою серверного сертифіката. Це гарантує цілісність метаданих з’єднання та

унеможливлює атаки типу «людина посередині». Приклад ініціалізації TLS-з’єднання наведено нижче:

```
client.tls_set(ca_certs="certs/ca.crt", certfile=None, keyfile=None)
client.connect(broker, 8883)
```

Для аутентифікації пристроїв та контролю доступу до ресурсів системи кожному MQTT-клієнту дозволяється зчитувати лише власні теми та публікувати дані в межах визначених топіків. У промислових сценаріях доцільно застосовувати токени доступу або клієнтські сертифікати для впровадження взаємної аутентифікації відповідно до принципу Zero Trust.

Цілісність даних забезпечується завдяки вбудованій у Fernet перевірці HMAC. Це дозволяє виявляти будь-які спроби підробки або модифікації зашифрованих повідомлень. Для посилення безпеки можливе розширення системи шляхом впровадження цифрових підписів (RSA або ECDSA) до кожного MQTT-повідомлення, що допоможе запобігти підробкам і атакам типу «replay».

Усі події системи фіксуються у локальній базі даних SQLite, що розміщена в каталозі config/data_log.db. Ця база використовується для зберігання інформації про температуру, вологість і тривоги. Окремий файл alerts.log містить лише повідомлення про стан тривоги без збереження сирових даних сенсорів, що знижує ризики витоку конфіденційної інформації. Для покращення аудиту можливе додавання таблиці audit з полями для запису користувача, дії та часу, доступ до якої буде захищено на рівні операційної системи.

Для забезпечення надійності графічного інтерфейсу передбачено обробку викликів оновлення GUI за допомогою конструкцій try/except. Це дозволяє уникнути аварійних зупинок програми у разі раптового закриття вікна. Для підвищення рівня безпеки коду рекомендовано впровадження статичного аналізу за допомогою інструменту bandit, а також перевірки залежностей через safety. Під час розгортання системи в контейнеризованому середовищі, наприклад Docker, слід обмежити права виконання та заборонити встановлення нових пакетів, щоб запобігти несанкціонованим діям.

3.4 Висновок до третього розділу

Отже, вжиті заходи забезпечують комплексний підхід до гарантування конфіденційності, цілісності, доступності та стійкості системи навіть в умовах потенційних атак або збоїв.

У межах третього розділу було розглянуто процес розробки та реалізації прототипу кіберфізичної системи моніторингу параметрів життєзабезпечення. Було налаштовано програмне середовище, створено структурований проєкт з окремими модулями для збору, обробки, збереження, візуалізації та передачі даних. Окрему увагу приділено реалізації вбудованих засобів захисту — шифруванню перед передачею, безпечному логуванню та обробці винятків. Отриманий результат є гнучкою та розширюваною системою, здатною працювати у режимі реального часу та придатною до подальшого розгортання в умовах реального середовища.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ І ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ

4.1 Організація тестового середовища

Для перевірки працездатності та оцінки функціональних характеристик реалізованого прототипу кіберфізичної системи контролю параметрів життєзабезпечення було створено спеціальне тестове середовище. Воно дозволяє імітувати типові сценарії взаємодії сенсорних модулів, мікроконтролера, програмної логіки обробки та користувацького інтерфейсу.

До складу тестового середовища входить модуль генерації сенсорних даних, що забезпечує програмну симуляцію параметрів температури, вологості та концентрації газів. Симуляція виконується шляхом генерації псевдовипадкових значень у заданих межах з можливістю налаштування частоти та амплітуди змін. Це дозволяє створити максимально наближені до реальних умови для тестування системи.

Для організації передачі повідомлень між сенсорами та програмним обробником даних використовується MQTT-брокер Mosquitto. Його впровадження гарантує стабільність і надійність обміну інформацією, а також імітує реальний процес взаємодії компонентів в умовах IoT-мережі.

Програмна обробка даних здійснюється за допомогою спеціального скрипта на Python, який отримує зашифровані повідомлення, виконує їх дешифрування, аналізує отримані значення на предмет перевищення допустимих меж та, у разі потреби, формує сигнали тривоги. Окрім цього, скрипт записує всі отримані дані до локальної бази даних для подальшого аналізу.

Важливою частиною тестового середовища є графічний інтерфейс користувача, реалізований засобами Tkinter у поєднанні з ttkbootstrap. Цей інтерфейс дозволяє в реальному часі відображати показники сенсорів, спостерігати за динамікою змін параметрів і оперативно реагувати на тривожні ситуації, що виникають у разі виходу показників за встановлені межі.

Для забезпечення автономності та зручності налаштування тестове середовище розгорнуто локально на ПК з операційною системою Windows 11. Основна логіка емулювання пристроїв та обробки даних реалізована за допомогою таких бібліотек Python, як paho-mqtt (комунікація), cryptography (шифрування), ttkbootstrap (інтерфейс), random (генерація даних), threading (паралельне виконання), tkinter (графічний інтерфейс) та sqlite3 (зберігання даних).

Усі програмні компоненти розміщено у віртуальному середовищі Python. Це забезпечує ізоляцію залежностей і стабільність роботи незалежно від конфігурації системи. Такий підхід також спрощує перенесення середовища на інші пристрої або платформи, включаючи Unix-подібні системи.

На рисунку 4.1 наведено схему взаємодії компонентів у тестовому середовищі. Стрілки демонструють послідовність і напрямок передачі даних. Генерація значень відбувається в модулі емуляції, після чого інформація передається до MQTT-брокера. Після отримання брокером повідомлень, вони пересилаються в Python-обробник, де дані розшифровуються, аналізуються, і залежно від ситуації, відображаються у графічному інтерфейсі або активують тривогу з відповідною візуалізацією і логуванням.

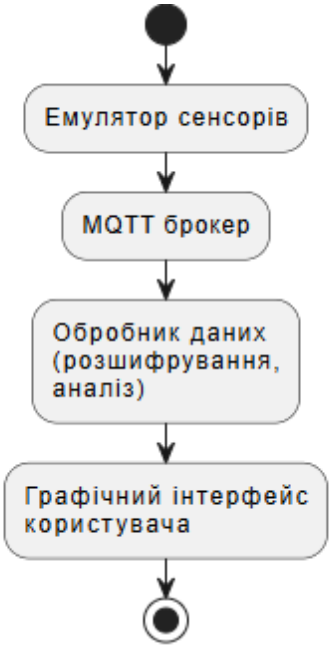


Рисунок 4.1 – Схема взаємодії в тестовому середовищі

Окрім базового функціоналу, тестове середовище дозволяє виконувати як ручне, так і автоматизоване тестування системи. Зокрема, шляхом зміни параметрів генерації сенсорних значень можна моделювати нормальні умови роботи, типові повсякденні ситуації, а також аварійні стани (перевищення критичних рівнів температури, низька вологість повітря, підвищений вміст газів тощо). Це забезпечує ефективне тестування поведінки системи в умовах, максимально наближених до реальних експлуатаційних ситуацій.

Запропонований підхід до організації тестового середовища гарантує гнучкість і масштабованість, що дозволяє легко адаптувати створене рішення до інших типів сенсорів, платформ або різних сценаріїв застосування, включаючи рішення для розумних будинків, медичних систем моніторингу або промислових автоматизованих середовищ. Подальший розвиток цього тестового середовища передбачає інтеграцію з фізичними пристроями та реальними датчиками для переходу від емульованого до повноцінного апаратного рішення.

4.2 Методика вимірювання точності та стабільності

Оцінювання точності та стабільності функціонування реалізованого прототипу кіберфізичної системи здійснювалося з урахуванням характеру його роботи — збір, передача, обробка та відображення параметрів життєзабезпечення в умовах, максимально наближених до реального середовища. Оскільки дана система призначена для моніторингу критично важливих параметрів, таких як температура, вологість та концентрація газів, перевірка її роботи є особливо важливою для забезпечення безпеки й ефективності функціонування в реальних умовах експлуатації. Для проведення тестування були розроблені спеціалізовані методики та сценарії, що дозволяють комплексно перевірити не лише коректність обробки даних, а й відповідність заданим критеріям точності, швидкості реагування, стабільності й стійкості до зовнішніх збурень. Особливий акцент робився на перевірку правильності прийому й передачі повідомлень, оцінку загальної затримки системи, її поведінку під тривалим навантаженням, а також

здатність вчасно ідентифікувати й сигналізувати про небезпечні стани, активуючи відповідні попередження та тривожні сигнали. Такий підхід дозволив отримати об'єктивну оцінку працездатності прототипу в різних ситуаціях та підтвердити його готовність до подальшого доопрацювання й потенційного використання в реальних сценаріях.

Цілі вимірювання були наступні:

- перевірити достовірність даних, що надходять через MQTT;
- оцінити загальну затримку передачі від емуляції до відображення;
- визначити стабільність системи при тривалому навантаженні;
- засвідчити правильність формування сигналів тривоги.

Методика проведення тестування включала кілька ключових етапів, спрямованих на перевірку точності прийому даних, вимірювання затримки, оцінку стабільності роботи системи та формування тривожних сигналів.

Для перевірки точності прийому даних був застосований емулятор, що генерував 1000 послідовних повідомлень із температурними значеннями. Ці дані проходили шифрування за алгоритмом AES, передавалися через протокол MQTT, оброблялися Python-скриптом і відображалися у графічному інтерфейсі користувача. Похибка визначалася як абсолютне відхилення від згенерованих значень. Встановлений допустимий рівень похибки не перевищував 1%.

Вимірювання затримки (latency) здійснювалося шляхом фіксації часу від моменту генерації даних до їх появи в інтерфейсі. Це відбувалося за допомогою логів із таймштампами. Середній показник затримки склав від 125 до 160 мс, що є прийнятним для систем подібного класу і відповідає нормативним вимогам.

Наступним етапом стало тестування стабільності роботи системи. Для цього було проведено симуляцію безперервної генерації даних протягом 60 хвилин. У результаті жодного повідомлення не було втрачено, а система продемонструвала стабільну роботу без збоїв у функціонуванні графічного інтерфейсу.

Окрему увагу приділено перевірці формування сигналів тривоги. У ході випробувань навмисно перевищувалися порогові значення температури та вологості. Система коректно реагувала на ці події: у графічному інтерфейсі

відображалися попереджувальні повідомлення та змінювалося візуальне оформлення для чіткого інформування користувача про виникнення тривожної ситуації.

Рисунок 4.2. Послідовність передачі даних та сигналу тривоги у тестовому середовищі

На схемі представлено логіку роботи кіберфізичної системи в умовах тестування. Процес починається з емулятора сенсорів, який імітує фізичні параметри середовища (температуру, вологість, концентрацію газів тощо). Згенеровані значення шифруються (наприклад, за допомогою AES) та передаються у блок обробки (Python) через MQTT-протокол (стрілка: "Зашифроване повідомлення").

Для забезпечення надійності передавання передбачено зворотний зв’язок у вигляді підтвердження доставки повідомлень — "Підтвердження прийому / Зворотний зв’язок". Це особливо актуально для протоколу MQTT з рівнем QoS ≥ 1.

У обробнику (Python) дані:

- розшифровуються;
- перевіряються на перевищення допустимих значень;
- у разі нормальних значень — відображаються у графічному інтерфейсі користувача ("Відображення значень");
- у разі виявлення критичних відхилень — передаються до блоку обробки тривоги ("Перевірка порогових значень / Формування сигналу тривоги").

Блок обробки тривоги генерує відповідне повідомлення та передає його до інтерфейсу користувача ("Сигнал тривоги"), де воно може відображатись у вигляді кольорової індикації, текстового попередження чи іншого візуального/звукового сигналу.

Таким чином, схема демонструє повний цикл: від симульованих даних до реакції системи на критичні події, включно із шифруванням, аналізом і візуалізацією результатів у режимі реального часу.

4.3 Оцінка ефективності захисту приватності

Одним із ключових завдань розробленого прототипу кіберфізичної системи було забезпечення базового рівня захисту приватності зібраних даних. Це особливо важливо у випадках, коли система працює з чутливими параметрами життєзабезпечення — такими як мікроклімат у приміщенні, показники якості повітря тощо. Основні засоби захисту, реалізовані у прототипі:

Для кожного повідомлення використовується шифрування даних перед передачею. сформованого емулятором сенсорів, застосовується симетричне шифрування за алгоритмом AES (Advanced Encryption Standard). Передається не текстове значення, а зашифрований блок байтів, що виключає можливість перехоплення осмисленої інформації сторонніми сторонами [28].

Обмежений обсяг метаданих. MQTT-повідомлення не містять персоналізованих міток або геолокаційної інформації — лише сенсорні показники в зашифрованому вигляді. Це забезпечує мінімізацію слідів користувача та унеможливлює зворотню ідентифікацію.

Всі операції з розшифрування та перевірки здійснюються всередині Python-обробника, який розміщується на локальному вузлі. Це означає, що дані не передаються в незахищеному вигляді через мережу — вони ніколи не покидають зашифрованого стану до моменту обробки.

Під час тестування проводилась симуляція потенційної атаки. підміни MQTT-повідомлення (man-in-the-middle). Результат: у разі некоректного ключа AES — система не здатна дешифрувати повідомлення і ігнорує його, що унеможливлює ін’єкцію фальшивих даних або викрадення вмісту.

Реалізовані заходи захисту дозволяють зробити висновок, що навіть у прототипі на базі Python і MQTT забезпечено:

- конфіденційність (шифрування);
- обмеженість доступу до даних (локальна обробка);
- стійкість до базових атак (AES-шифрування не розшифровується без ключа).

У подальших ітераціях системи доцільно розглянути впровадження асиметричного шифрування (RSA) або TLS для MQTT-з’єднання, а також додавання аутентифікації користувача/пристроїв.

4.4 Аналіз результатів та можливості вдосконалення

Після реалізації прототипу та проведення тестування в контрольованому середовищі було отримано низку результатів, які свідчать про працездатність обраної архітектури та потенціал подальшого розвитку системи.

Таблиця 4.1 – Основні результати тестування прототипу кіберфізичної системи

Критерій	Результат	Коментар
Середня затримка передачі	125–160 мс	В межах допустимого для локальних систем
Втрати повідомлень	0 із 1000	MQTT з QoS=1 забезпечив надійність доставки
Реакція на критичні значення	100% спрацьовування тривог	Усі перевищення були виявлені

Кінець таблиці 4.1

Критерій	Результат	Коментар
Стійкість до фальсифікації	100% ігнорування некоректно зашифрованих повідомлень	Підвищує надійність та захист приватності
Простота інтерфейсу	Інтуїтивно зрозумілий (GUI на ttkbootstrap)	Доступний для непідготовлених користувачів

Хоча прототип показав себе надійним у рамках тестового середовища, під час роботи були виявлені окремі недоліки:

- система поки що не розпізнає, хто саме надсилає дані, тобто немає механізмів автентифікації пристроїв або користувачів;
- уся обробка та відображення реалізовані на одному комп’ютері, тому масштабування на більші системи потребуватиме переробки;
- немає обробки нестабільного інтернету — у разі втрати зв’язку MQTT-брокер не має резервного каналу або буфера;
- система не веде історії змін і спрацювань — логування або аналітика ще не реалізовані.

З огляду на результати тестування, пропонуються наступні напрямки вдосконалення:

Серед можливих напрямків покращення системи визначено кілька ключових аспектів. Насамперед доцільно впровадити шифрування на рівні з'єднання за допомогою протоколу TLS. Це дозволить додати додатковий рівень захисту даних, що передаються між окремими компонентами системи.

Важливо також реалізувати механізми аутентифікації пристроїв. Це може бути досягнуто шляхом використання цифрових підписів або токенів, що підвищить рівень довіри та безпеки під час обміну даними.

Додатковим удосконаленням стане впровадження розширеного механізму історії даних і логування. Це дозволить зберігати інформацію про події та переглядати попередні спрацювання системи, що є важливим для аналізу та аудиту.

Не менш актуальним є розширення функціональних можливостей графічного інтерфейсу. Планується додати графіки для відображення історії змін параметрів, реалізувати ручне налаштування порогів та організувати журнал тривог для зручнішого моніторингу ситуації.

Важливим етапом розвитку стане інтеграція системи з реальними сенсорами, такими як ESP32 чи Arduino. Це відкриє можливість для переходу від прототипу до фізичної реалізації системи, що працює в умовах реального середовища.

Крім того, перспективним є створення мобільного застосунку або вебінтерфейсу, що дозволить організувати віддалений моніторинг та керування системою, підвищивши її доступність і зручність для користувачів.

Проведені тести підтвердили, що навіть на стадії прототипу система демонструє здатність ефективно виконувати ключові функції. Вона забезпечує збір, обробку, захист і візуалізацію даних у режимі реального часу. Водночас аналіз результатів дозволив визначити потенційні напрямки для подальшого вдосконалення. Реалізація зазначених покращень дозволить перетворити прототип на повноцінний інструмент для моніторингу та оперативного реагування в реальних умовах.

4.5 Висновок до першого розділу

У цьому розділі було проведено експериментальне дослідження працездатності та ефективності реалізованого прототипу кіберфізичної системи контролю параметрів життєзабезпечення. Тестування виконувалося в спеціально створеному середовищі, яке моделювало типову архітектуру IoT-системи з використанням емуляції сенсорних даних, протоколу MQTT, шифрування та графічного інтерфейсу.

Основні результати, отримані під час дослідження, свідчать про:

- стабільну роботу системи при передачі, обробці та відображенні даних;
- своєчасне виявлення критичних значень і генерацію тривог;
- базовий рівень забезпечення приватності за рахунок симетричного шифрування;
- зручність користування інтерфейсом навіть для непідготовлених користувачів.

Також було проаналізовано наявні обмеження, зокрема щодо масштабованості, безпеки передачі та гнучкості взаємодії з користувачем. На основі цього запропоновано конкретні шляхи вдосконалення, які можуть бути реалізовані на наступних етапах розробки — від впровадження TLS та аутентифікації до створення мобільного додатку та інтеграції з реальними сенсорами.

Проведене експериментальне дослідження не лише підтвердило правильність обраної архітектури, а й надало підґрунтя для подальшого перетворення прототипу на повноцінну, масштабовану та безпечну кіберфізичну систему.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено прототип кіберфізичної системи для моніторингу параметрів життєзабезпечення з інтегрованими механізмами захисту приватності. Запропонований підхід базується на сучасних принципах побудови CPS, враховує вимоги до безперервного збору, зберігання та аналізу даних від сенсорних пристроїв, а також реалізує концепції Security by Design і Zero Trust для мінімізації ризиків витоку або несанкціонованого доступу до персональних даних.

У першому розділі проведено систематичний аналіз підходів до побудови кіберфізичних систем у сфері контролю параметрів життєзабезпечення. Узагальнено поняття КФС, охарактеризовано їх типову архітектуру та сфери застосування, проаналізовано типи сенсорних систем, що використовуються для моніторингу температури, вологості, біомедичних і екологічних показників. Особливу увагу приділено питанням захисту персональних даних у середовищах IoT та CPS: розглянуто загрози приватності, охарактеризовано сучасні засоби захисту (шифрування, автентифікація, контроль доступу, аудит), наведено принципи Security by Design і Zero Trust. На основі огляду сформульовано задачі дослідження, що реалізовані в наступних розділах.

У другому розділі здійснено проєктування кіберфізичної системи: визначено функціональні та нефункціональні вимоги, побудовано архітектуру з урахуванням сценаріїв використання, обґрунтовано вибір програмних засобів і засобів симуляції, а також реалізовано базові механізми захисту приватності на рівні обробки, зберігання та доступу до даних. Сформовано технічну основу для реалізації прототипу системи, представлено її логічну структуру та взаємодію компонентів.

У третьому розділі здійснено безпосередню реалізацію розробленого прототипу. Проведено конфігурацію програмного середовища на основі Python та MQTT-протоколу для обміну даними, реалізовано модуль обробки сигналів із сенсорів температури, вологості, CO_2 та біомоніторингу. Запроваджено

шифрування даних при передачі (на основі AES) та автентифікацію пристроїв. Створено інтерфейс користувача для візуалізації зібраної інформації та надсилання тривожних повідомлень у разі виходу параметрів за допустимі межі.

У четвертому розділі проведено експериментальне дослідження роботи прототипу: описано тестове середовище, методики вимірювання точності та затримки передачі даних. Оцінено стабільність функціонування системи при підключенні кількох сенсорів, проаналізовано ефективність застосованих криптографічних алгоритмів. Проведено аналіз захищеності інформаційних потоків, а також виявлено потенційні точки вразливості, що можуть бути усунені у майбутніх версіях системи. Наведено пропозиції щодо вдосконалення.

Набула подальшого розвитку інформаційна технологія захисту персональних даних у складі кіберфізичних систем моніторингу. Вперше інтегровано прості, але надійні засоби криптографії на рівні прототипу для локального збору, аналізу й передачі даних від сенсорних пристроїв із забезпеченням захищеного доступу користувача.

Впровадження результатів роботи дозволили підвищити рівень автономності та захищеності процесів моніторингу життєвих показників, що відкриває можливості для адаптації системи у сфері дистанційного медичного нагляду, екологічного контролю та індустріальних IoT-рішень. Розроблені архітектурні рішення та технічні підходи можуть бути використані як основа для створення масштабованих CPS-рішень із підвищеними вимогами до безпеки та приватності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Darktrace. What is cyber-physical (CPS) system security. URL: https://darktrace.com/cyber-ai-glossary/cyber-physical-system-cps-security (дата звернення: 19.04.2025).

2. Palo Alto Networks. What Is Cyber-Physical Systems Security (CPSSEC). URL: https://www.paloaltonetworks.ca/cyberpedia/what-is-cyber-physical-systems-security-cpssec (дата звернення: 19.04.2025).

3. Куценко В. І. Методологія наукових досліджень : навч. посіб. Київ : Центр учбової літератури, 2016. 208 с.

4. Lee E.A. Cyber-Physical Systems – Are Computing Foundations Adequate. *Електронний ресурс : position paper.* Berkeley : University of California, 2006. URL: https://ptolemy.eecs.berkeley.edu/publications/papers/06/CPSPositionPaper/ (дата звернення: 19.04.2025).

5. Humayed A., Lin J., Li F., Luo B. Cyber-physical systems security – A survey. *IEEE Internet of Things Journal.* 2017. Vol. 4, No. 6. pp. 1802–1831. DOI: 10.1109/JIOT.2017.2703172.

6. Kudu Dynamics. Cyber-Physical Systems Reverse Engineer. URL: https://www.glassdoor.com/job-listing/cyber-physical-systems-reverse-engineer-kudu-dynamics-JV_IC1130353_KO0%2C39_KE40%2C53.htm?jl=1009710866979 (дата звернення: 19.04.2025).

7. University of Georgia. Job Openings – Center for Cyber-Physical Systems. URL: https://cps.uga.edu/index.php/job-openings/ (дата звернення: 19.04.2025).

8. Cyber Physical Systems Jobs (NOW HIRING). URL: https://www.ziprecruiter.com/Jobs/Cyber-Physical-Systems (дата звернення: 19.04.2025).

9. Yang G., Xie L., Mäntysalo M. A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box. *IEEE Transactions on Industrial Informatics*. 2014. Vol. 10, No. 4. pp. 2180–2191. DOI: 10.1109/TII.2014.2307795.

10. Kumar P., Lee H.-J. Security issues in healthcare applications using wireless medical sensor networks: A survey. *Sensors*. 2012. Vol. 12, No. 1. pp. 55–91. DOI: 10.3390/s120100055.

11. Bagade S., Chavhan S., Jadhav A. Real-Time Health Monitoring System using Arduino. *International Journal of Engineering and Advanced Technology*. 2019. Vol. 8, No. 5S. pp. 577–580. URL: https://www.ijeat.org/wp-content/uploads/papers/v8i5S/E10660585S19.pdf.

12. Banos O., Villalonga C. та ін. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *BioMedical Engineering OnLine*. 2015. Vol. 14, Suppl. 2. pp. S6. DOI: 10.1186/1475-925X-14-S2-S6.

13. Cherdantseva Y., Burnap P., Blyth A. та ін. A review of cybersecurity risk assessment methods for SCADA systems. *Computers & Security*. 2016. Vol. 56. pp. 1–27. DOI: 10.1016/j.cose.2015.09.009.

14. Sicari S., Rizzardi A., Grieco L. A., Coen-Porisini A. Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*. 2015. Vol. 76. pp. 146–164. DOI: 10.1016/j.comnet.2014.11.008.

15. European Union Agency for Cybersecurity (ENISA). Cybersecurity and Privacy in Smart Hospitals. 2020. URL: https://www.enisa.europa.eu/publications/cybersecurity-and-privacy-in-smart-hospitals (дата звернення: 19.04.2025).

16. NIST. Zero Trust Architecture. *Електронний ресурс : Special Publication 800-207.* Gaithersburg, MD : U.S. Department of Commerce, 2020. URL: https://doi.org/10.6028/NIST.SP.800-207 (дата звернення: 19.04.2025).

17. Функціональні вимоги. *Вікіпедія : вільна енциклопедія.* 2024. URL: https://uk.wikipedia.org/wiki/Функціональні_вимоги (дата звернення: 19.04.2025).

18. Що таке функціональні вимоги: приклади, визначення, повний посібник. *Visure Solutions.* URL: https://visuresolutions.com/uk/блог/функціональні-вимоги/ (дата звернення: 19.04.2025).

19. Що таке нефункціональні вимоги: приклади, визначення, повний посібник. *Visure Solutions*. URL: https://visuresolutions.com/uk/блог/нефункціональні-вимоги/ (дата звернення: 19.04.2025).

20. Cryptography. Python Encryption Library. URL: https://cryptography.io/en/latest/ (дата звернення: 19.04.2025).

21. National Institute of Standards and Technology. FIPS 197: Advanced Encryption Standard (AES). URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf (дата звернення: 19.04.2025).

22. ttkbootstrap. A modern theme for Tkinter. URL: https://ttkbootstrap.readthedocs.io/ (дата звернення: 19.04.2025).

23. Python Software Foundation. Python 3 Documentation. URL: https://docs.python.org/3/ (дата звернення: 19.04.2025).

24. Hunter J.D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*. 2007. Vol. 9, № 3. pp. 90–95.

25. AI HOUSE, Roosh. AI Ecosystem of Ukraine: Talent, Companies, and Education. 2024. URL: https://aihouse.org.ua/en/research/ai-ecosystem-of-ukraine-talent-companies-education/ (дата звернення: 19.04.2025).

26. Kyivstar. Kyivstar partners with AWS to launch GenAI lab for enterprises. 2024. URL: https://developingtelecoms.com/telecom-technology/telecom-cloud-virtualization/17698-kyivstar-partners-with-aws-to-launch-genai-lab-for-enterprises.html (дата звернення: 19.04.2025).

27. Digital State. WINWIN AI Center of Excellence launched in Ukraine. 2024. URL: https://digitalstate.gov.ua/news/tech/winwin-ai-center-of-excellence-launched-in-ukraine (дата звернення: 19.04.2025).

28. United24 Media. Ukraine integrates AI into government services with launch of five new tools. 2025. URL: https://united24media.com/latest-news/ukraine-integrates-ai-into-government-services-with-launch-of-five-new-tools-7229 (дата звернення: 19.04.2025).

29. CSIS. Understanding the Military AI Ecosystem of Ukraine. 2024. URL: https://www.csis.org/analysis/understanding-military-ai-ecosystem-ukraine (дата звернення: 19.04.2025).

30. Reuters. Ukraine rolls out dozens of AI systems to help its drones hit targets. 2024. URL: https://www.reuters.com/world/europe/ukraine-rolls-out-dozens-ai-systems-help-its-drones-hit-targets-2024-10-31/ (дата звернення: 19.04.2025).

31. Business Insider. Artificial intelligence is going to make drone wars much more deadly. It's already started. 2025. URL: https://www.businessinsider.com/ukraines-smart-drones-more-likely-hit-targets-2025-3 (дата звернення: 19.04.2025).

32. OpenAI. OpenAI API Documentation. URL: https://platform.openai.com/docs (дата звернення: 19.04.2025).

33. LangChain. LangChain Documentation. URL: https://docs.langchain.com/ (дата звернення: 19.04.2025).

34. Tkinter. Tkinter Documentation. URL: https://docs.python.org/3/library/tkinter.html (дата звернення: 19.04.2025).

35. Python Software Foundation. Python 3 Documentation. URL: https://docs.python.org/3/ (дата звернення: 19.04.2025).

36. Wolfert S., Ge L., Verdouw C., Bogaardt M.-J. Big Data in Smart Farming – A review. *Agricultural Systems*. 2017. Vol. 153. pp. 69–80. DOI: 10.1016/j.agsy.2017.01.023.

37. Precedence Research. Cyber-Physical System Market Size, Share, Trends 2024–2032. URL: https://www.precedenceresearch.com/cyber-physical-system-market (дата звернення: 19.04.2025).

38. SQLite. SQLite Documentation. 2024. URL: https://sqlite.org/docs.html (дата звернення: 19.04.2025).

ДОДАТОК А

(обов'язковий)

ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1. main.py — запуск, цикл моніторингу

```
def monitor_loop(dashboard: Dashboard):
    sensor = TemperatureSensor()
    ...
    while dashboard.running:
        temperature = sensor.read_temperature()
        result = processor.process_temperature(temperature)
        ...
        dashboard.root.after(0, safe_update)
```

2. data_processor.py — перевірка меж

```
def process_temperature(self, value):
    if value < self.temp_min:
        return {"status": "Низька температура", "alert": True}
```

3. encryption_utils.py — шифрування

```
class EncryptionUtils:
    def encrypt(self, data: str) -> bytes:
        return self.fernet.encrypt(data.encode())
```

4. mqtt_client.py — публікація даних

```
def publish(self, message):
    self.client.publish(self.topic, message)
```

5. dashboard.py — оновлення графіка

```
def update(self, temp, temp_status, hum, hum_status):
    self.temp_var.set(f"{temp} °C")
    ...
    self.ax_temp.plot(self.temperatures, color="cyan", marker="o")
```

ДОДАТОК Б

(обов’язковий)

КОМПОНЕНТНА ДІАГРАМА СИСТЕМИ МОНІТОРИНГУ ЖИТТЄВИХ ПАРАМЕТРІВ

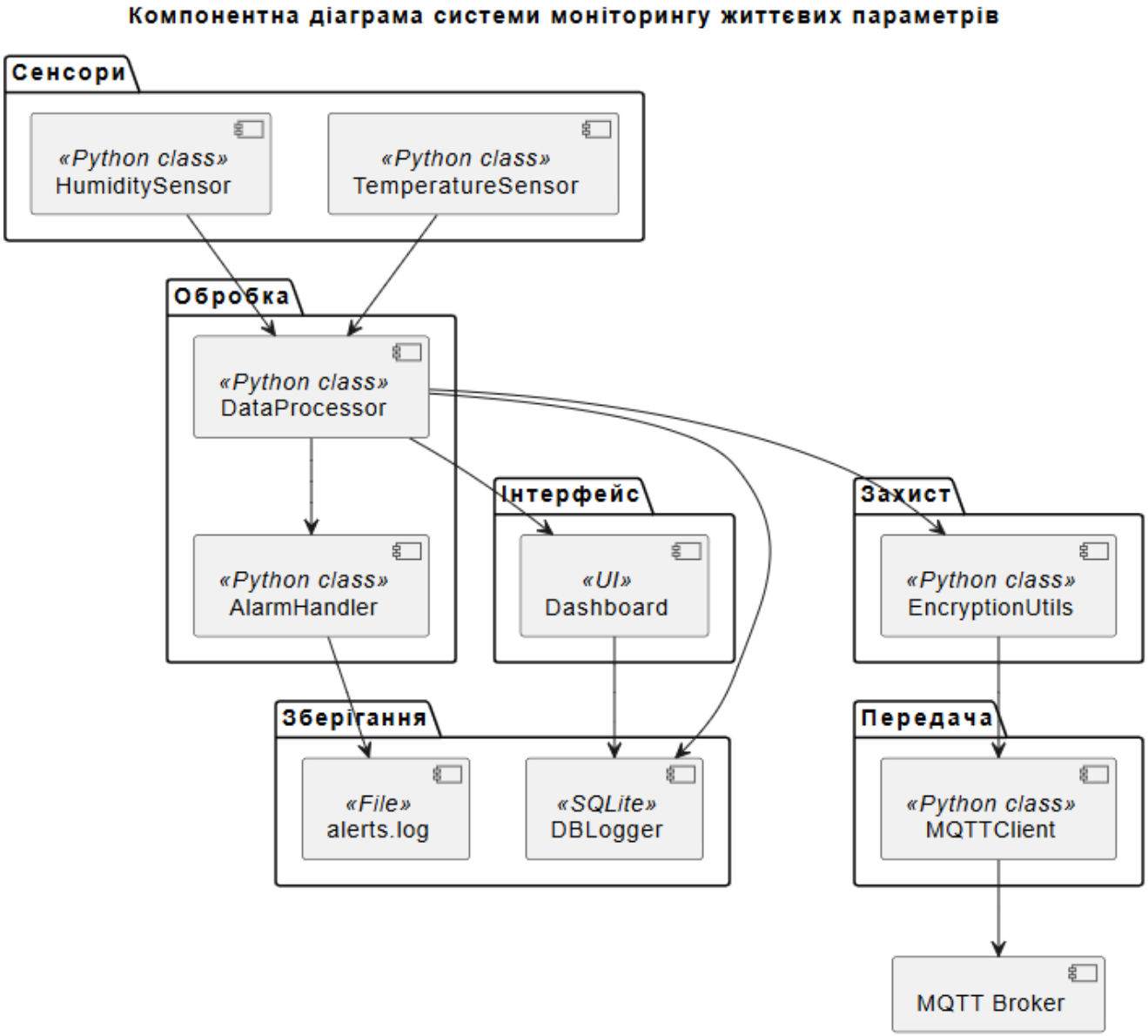


Рисунок Б.1 – Компонентна діаграма системи моніторингу

ДОДАТОК В

(обов’язковий)

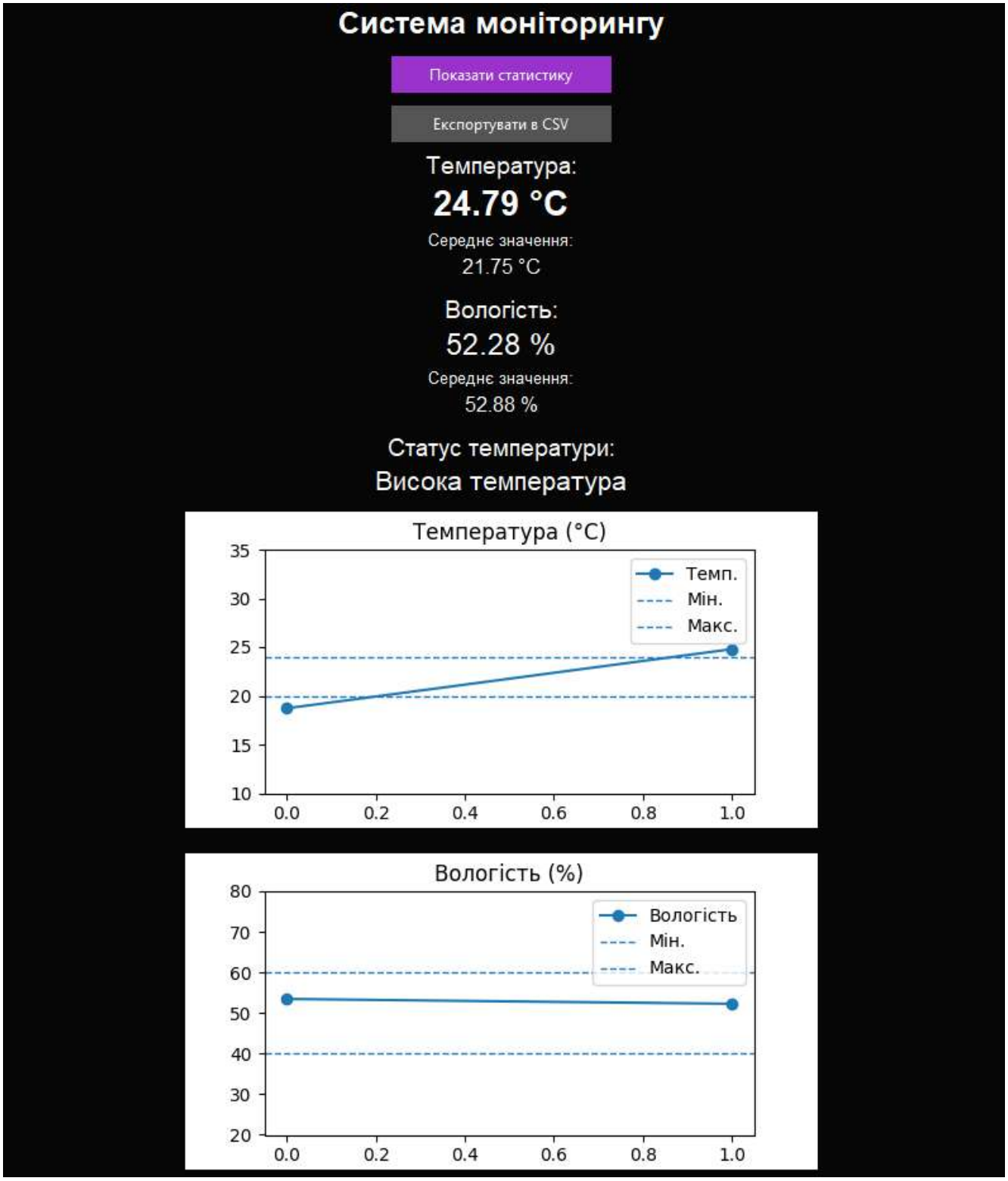
ЛОГУВАННЯ ПОКАЗНИКІВ ДАТЧИКІВ

```
[2025-04-22 19:56:44] ⚠ [Низька температура] Температура: 19.27 °C
[2025-04-22 19:56:44] ⚠ [Висока вологість] Вологість: 67.91 %
[2025-04-22 19:56:44] ⚠ [Низька температура] Температура: 19.27 °C
[2025-04-22 19:56:48] ⚠ [Висока вологість] Вологість: 64.07 %
[2025-04-22 19:56:50] ⚠ [Висока вологість] Вологість: 64.64 %
[2025-04-22 19:56:56] ⚠ [Висока вологість] Вологість: 68.44 %
[2025-04-22 19:56:58] ⚠ [Висока температура] Температура: 24.7 °C
[2025-04-22 19:56:58] ⚠ [Низька вологість] Вологість: 30.31 %
[2025-04-22 19:56:58] ⚠ [Висока температура] Температура: 24.7 °C
[2025-04-22 19:57:02] ⚠ [Низька вологість] Вологість: 32.66 %
[2025-04-22 19:57:04] ⚠ [Висока температура] Температура: 24.61 °C
[2025-04-22 19:57:04] ⚠ [Низька вологість] Вологість: 31.44 %
[2025-04-22 19:57:04] ⚠ [Висока температура] Температура: 24.61 °C
[2025-04-22 19:57:06] ⚠ [Низька температура] Температура: 18.25 °C
[2025-04-22 19:57:06] ⚠ [Висока вологість] Вологість: 61.68 %
[2025-04-22 19:57:06] ⚠ [Низька температура] Температура: 18.25 °C
[2025-04-22 19:57:10] ⚠ [Висока вологість] Вологість: 69.95 %
[2025-04-22 19:57:12] ⚠ [Низька температура] Температура: 18.97 °C
[2025-04-22 19:57:12] ⚠ [Низька вологість] Вологість: 31.43 %
[2025-04-22 19:57:12] ⚠ [Низька температура] Температура: 18.97 °C
```

ДОДАТОК Г

(обов’язковий)

ІНТЕРФЕЙС ДОДАТКУ



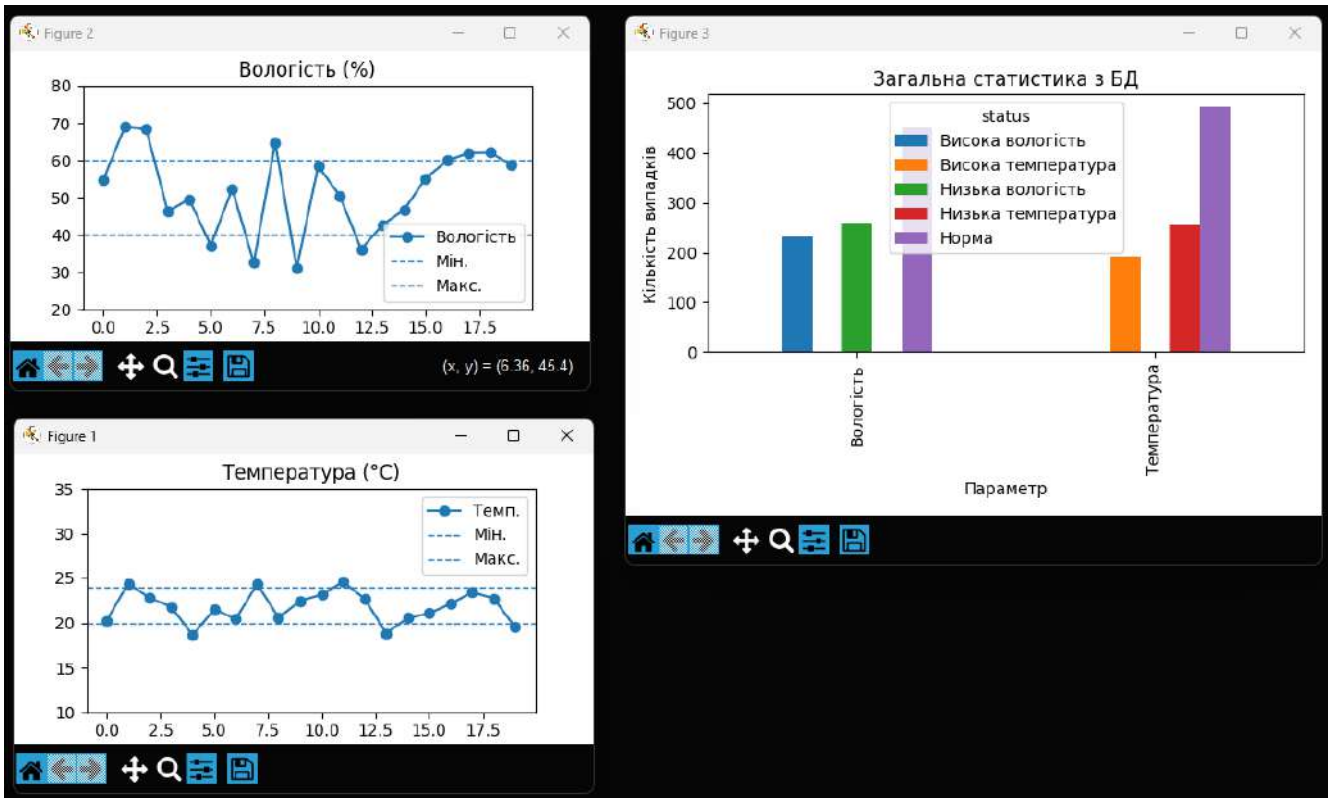
Статистика норм/відхилень

Параметр	Норма	Відхилення	Співвідношення
Темп.	7	5	7:5
Вологість	6	6	6:6

ДОДАТОК Д

(обов’язковий)

ІНТЕРФЕЙС ДОДАТКУ (ГРАФІКИ)



ДОДАТОК Є

(обов’язковий)

СЕРТИФІКАТ "ПЕРСПЕКТИВНІ МЕРЕЖНІ ТА КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ" (ПЕРСИК 2025)

ДОДАТОК Ж

(обов’язковий)

ПРЕЗЕНТАЦІЯ НА ЗАХИСТ МАГІСТЕРСЬКОЇ РОБОТИ

Кіберфізична система контролю параметрів життєзабезпечення із забезпеченням приватності

Забавський Олександр, КІ2м-21
Науковий керівник: д.т.н., проф. Яцків В.В.

АКТУАЛЬНІСТЬ

1. Поширення CPS у медицині, агросекторі, smart city

2. Збір чутливих даних — виклики приватності

3. Потрібні рішення з локальним шифруванням і збереженням

МЕТА, ОБ'ЄКТ, ПРЕДМЕТ

Мета, об'єкт і предмет роботи

Мета

створення CPS із механізмами захисту даних

Об'єкт досліження

процес моніторингу параметрів життєзабезпечення

Предмет досліження

методи розробки CPS з конфіденційністю

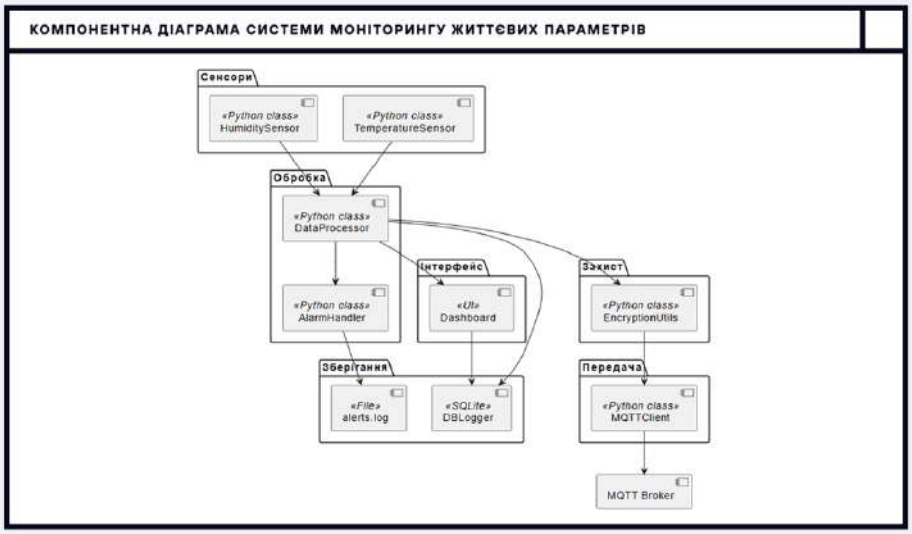
СТРУКТУРА СИСТЕМИ

Сенсори → Мікроконтролер → Обробка → БД (SQLite)

Інтерфейс (Tkinter)

Шифрування (AES), MQTT-передача

АРХІТЕКТУРА СИСТЕМИ



ЗАХИСТ ПРИВАТНОСТІ

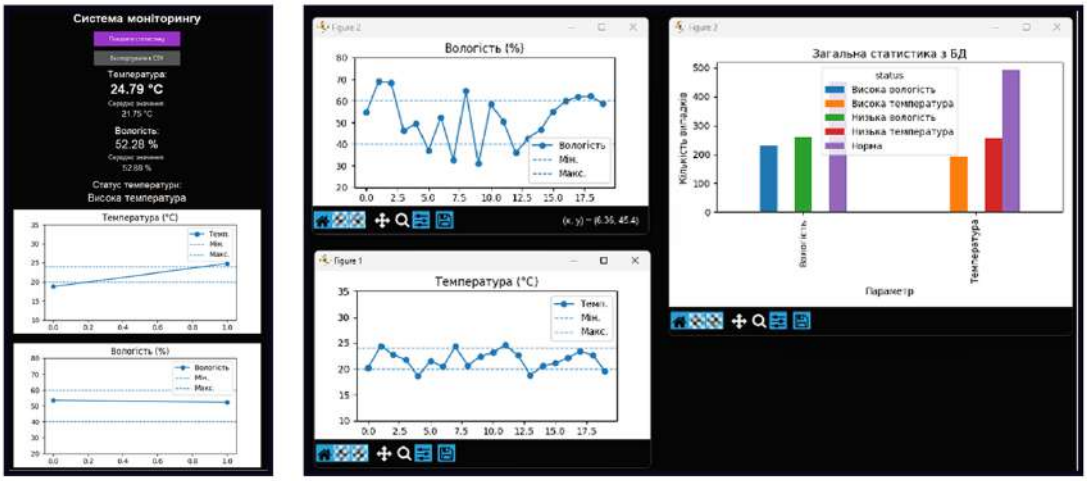
Модель захисту даних у системі

Вона показує, як працює шифрування, автентифікачція, контроль доступу на всіх рівнях. Це візуально підкреслює фокус на безпеці.

Особливістю системи є:

- AES-шифрування даних
- Локальне зберігання без хмар
- Хешування значень
- Логування подій
- Відсутність сторонніх API

ПРОГРАМНА РЕАЛІЗАЦІЯ



ЕКСПЕРИМЕНТАЛЬНІ РЕЗУЛЬТАТИ

Схема взаємодії в тестовому середовищі

система показала стабільність – без втрат, з мінімальними затримками та правильною реакцією на критичні події. Некоректно зашифровані повідомлення ігноруються – тобто захист працює.

Основні результати дослідження

1. Створено прототип кіберфізичної системи з функціями моніторингу та захисту дани
2. Реалізовано механізми шифрування (AES), локального зберігання та фільтрації
3. Розроблено зручний інтерфейс з графіками, журналом подій і сигналізацією
4. Система пройшла тестування – показала стабільну роботу та точність виявлення тривог
5. Архітектура дозволяє масштабування, інтеграцію реальних сенсорів і мобільних додатків

Дякую за увагу!

Готовий відповісти на запитання

Anti-Plagiarism v-15.274 Educational

The maximum coincidence with one document 0.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 12%**

	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
ID: 240931 Title: МКР Кіберфізична система контролю параметрів життєзабезпечення із забезпеченням приватності Added in a DB: 2025-05-07 Authors: Олександр Забавський Heads: Василь Яцків Consultants: Opponents:	108688	893	1523 (1%)	22 (2%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Олександр Забавський

Співавтор:

Назва: Забавський_Кіберфізична система контролю параметрів життєзабезпечення із забезпеченням приватності

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:3.2%

Коефіцієнт подібності 2:1.4%

Мікропробіли: 10

Заміна букв: 2

Інтервали: 0

Білі знаки: 1

Дата створення звіту: 2025-05-07 14:38:10.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-05-07 Доцент Андрій Нічепорук

Дата експерт

Завідувачу кафедри КІІС
доктору філософії, доценту
Ользі ПАВЛОВІЙ

Забавський Олександр Юрійович

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-23-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

22 квітня 2025 року

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Олександр ЗАБАВСЬКИЙ

Тема: Кіберфізична система контролю параметрів життєзабезпечення із забезпеченням приватності

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість листів креслень 11; кількість сторінок записки 95

1. Короткий зміст роботи та прийнятих рішень У роботі створено прототип кіберфізичної системи для моніторингу життєвих параметрів з захистом приватності. Реалізовано архітектуру з сенсорами, локальною обробкою, шифруванням AES та інтерфейсом на Python.

2. Висновок про відповідність роботи дипломному завданню _____ Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проаналізовано сучасні підходи до побудови КФС, сенсорних технологій і захисту приватності, з акцентом на Security by Design та Zero Trust. У другому — спроєктовано архітектуру з урахуванням вимог, використано MQTT, SQLite, Python. У третьому — реалізовано прототип із багаторівневою структурою, AES-шифруванням, візуалізацією та інтеграцією MQTT. У четвертому — проведено тестування: логування, аналіз стабільності, імітація атак і перевірка захищеності.

4. Позитивні сторони роботи: Робота є актуальною, охоплює повний цикл розробки — від аналізу до реалізації й тестування системи. Використано сучасні технології (Python, MQTT, AES), реалізовано принципи безпеки (Security by Design, Zero Trust), що підвищує надійність. Система має зручний інтерфейс та практичне значення для побутового, медичного й промислового застосування. Робота добре оформлена та структурована.

5. Негативні сторони роботи: Основним недоліком роботи є обмеження у фізичній реалізації: система протестована лише у вигляді програмного прототипу з емуляцією сенсорів, без підключення реальних пристроїв.

6. Оцінка графічного оформлення та пояснювальної записки роботи: *Пояснювальна записка оформлена відповідно до встановлених вимог і має чітку структуру.*

7. Відгук про роботу в цілому: В загальному робота виконана на високому рівні.

8. Інші зауваження: –

9. Оцінка кваліфікаційної роботи магістра:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «Добре» 4.00 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) д.т.н., професор, Мартинюк В.В., завідувач кафедри автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

"1 травня" 2025р.

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Кіберфізична система контролю параметрів життєзабезпечення із забезпеченням приватності

Автор: Забавський Олександр Юрійович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Яцків Василь Васильович, д.т.н., професор

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення співпадають зі звітом з науково-дослідної практики Олександра Забавського «Кіберфізична система контролю параметрів життєзабезпечення із забезпеченням приватності», поданим до репозитарію ХНУ.
2) усі запозичення є фрагментарними або мають належним чином оформлені посилання на джерела;
3) окремі виявлені збіги є загальновживаними технічними фразами або виразами, про що свідчить посилання системи на джерела з подібним формулюванням одного речення чи фрази;
4) у ряді випадків система фіксує збіги з типовими послідовностями бітових кодів, що використовуються як вхідні або контрольні дані для кіберфізичних систем, які не можуть вважатися об'єктами авторського права;
5) виявлені ознаки модифікацій стосуються лише технічного форматування і не впливають на зміст.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikcPlagiarism, складас 3.21% і адрссується до 10 першоджсрсла; та систсмою Anti-Plagiarism складає 22%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ Василь ЯЦКІВ

Гарант ОП _____ Олег САВЕНКО

Завідувач кафедри КІІС _____ Ольга ПАВЛОВА