

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

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень

Кіберфізична система моніторингу параметрів мікроклімату у медичному

закладі  
Назва теми

КІРКІ 210108.21.01.55 ПЗ  
Шифр

Галузь знань 12 «Інформаційні технології»  
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»  
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»  
Назва

Виконав: студент IV курсу, група КІ2-21-1 Мороз Олександр ЖОЛУДЕВ  
Підпис Ініціали, прізвище

Керівник [Підпис] Людмила КОРЕЦЬКА  
Підпис, дата Ініціали, прізвище

Нормоконтролер [Підпис] Тетяна КИСІЛЬ  
Підпис, дата Ініціали, прізвище

До захисту допускаю:  
зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем

[Підпис] Ольга ПАВЛОВА  
Підпис Ініціали, прізвище

«19» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Жолудєв Олександр Андрійович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система моніторингу параметрів мікроклімату у медичному закладі

Керівник проекту (роботи) Людмила Корецька, к.т.н., доц.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

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

Кіберфізична система моніторингу параметрів мікроклімату у медичному закладі

Аналіз складових системи та засобів її розробки

Опис програмної частини реалізація агенту мікроклімату лабораторії кіберфізичної системи

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

Алгоритм роботи програмного забезпечення

Схема концепції роботи кіберфізичної системи

Апаратне забезпечення проекту

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – кіберфізична система моніторингу параметрів мікроклімату у медичному закладі	01.03.2025	виконано
4	Робота над розділом 2 – аналіз складових системи та засобів її розробки	01.04.2025	виконано
5	Робота над розділом 3 – опис програмної реалізації агента мікроклімату лабораторії кібер-фізичної системи	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Керівник роботи

Підпис

Олександр ЖОЛУДСЬ  
Ініціали, прізвище

Підпис

Людмила КОРЕЦЬКА  
Ініціали, прізвище

Підпис с

Тема кваліфікаційної роботи  
параметрів мікроклімату  
Автор роботи  
Керівник роботи  
Пояснювальна записка  
Графічна частина

СЕНСОРНИЙ  
МОНІТОРИНГ, М

Метою дипломного проекту є дослідження  
кіберфізичного об'єкта та  
забезпечення моніторингом

Об'єктом дослідження є  
мікроклімату у медичному закладі

Предметом дослідження є  
збір, аналіз та обробка даних  
допомогою вбудованих  
інтерактивних інтерфейсів

Під час виконання роботи  
систематично здійснювалися  
даного дослідження

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Кіберфізична система моніторингу параметрів мікроклімату у медичному закладі».

Автор роботи: Олександр ЖОЛУДСЬ.

Керівник роботи: Людмила Корецька Олександрівна.

Пояснювальна записка: 79 с., 33 рис., 2 табл., 3 дод., 40 джерел.

Графічна частина: 3 креслення.

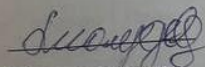
СЕНСОРНІ ДАТЧИКИ, КІБЕРФІЗИЧНА СИСТЕМА, АРХІТЕКТУРА, МОНІТОРИНГ, МЕСЕНДЖЕР-БОТ.

Метою дипломної роботи є визначення умов та особливостей застосування кіберфізичного обладнання, а також аналіз схожих кіберфізичних систем для забезпечення миттєвих повідомлень у месенджер за допомогою бота, а також моніторингом самого кіберфізичного пристрою за допомогою Blynk.

Об'єкт дослідження - процес забезпечення та контролю параметрів мікроклімату у приміщеннях медичного закладу.

Предмет дослідження - кіберфізична система моніторингу, яка забезпечує збір, аналіз та реагування на параметри мікроклімату у медичному закладі за допомогою вбудованих сенсорів, контролерів, програмного забезпечення та інтерактивних інтерфейсів.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.

  
Підпис студента

30.05.2025

Дата

ЖОЛУДСЬ  
звіще

КОРЕЦЬКА  
звіще

№ Р я д к а	Ф о р м а т	Позначення	Найменування	К і л - л и с т і в	№ е кз	П р и м і т к а
			<u>Текстові документи</u>			
1		КІРКІ 210108.21.01.55 ПЗ	Пояснювальна записка	60		
			<u>Графічні матеріали</u>			
2		КІРКІ 210108.21.01.55 Е8	Алгоритм роботи програмного забезпечення	1		
3		КІРКІ 210108.21.01.55 Е8	Логічна схема роботи системи	1		
4		КІРКІ 210108.21.01.55 Е8	Апаратне забезпечення проєкту	1		
			<u>Код проєкту</u>			
6		КІРКІ 210108.21.01.55	Програмний код для інтеграція з мікроконтролером	7		
КІРКІ 210108.21.01.55 ВП						
Зм	Арк	№ докум	Підпис	Дата	Літера	Аркуш
Розробив		Жолудєв	<i>[Signature]</i>	19.06.21	У	1
Перевір.		Корецька	<i>[Signature]</i>	19.06.21		1
Н. контр.		Кисіль	<i>[Signature]</i>	19.06.21	ХНУ, КІ2-21-1	
Затв.		Павлова	<i>[Signature]</i>	19.06.21		
Відомість проєкту						

## ЗМІСТ

<b>ВСТУП</b> .....	4
<b>1 КІБЕРФІЗИЧНА СИСТЕМА МОНІТОРИНГУ ПАРАМЕТРІВ МІКРОКЛІМАТУ У МЕДИЧНОМУ ЗАКЛАДІ</b> .....	6
1.1 Актуальність задачі .....	6
1.2 Аналіз та дослідження аналогів або інших систем .....	8
1.2.1 Розумна система поливу грядок .....	11
1.2.2 Система моніторингу довкілля .....	12
1.2.3 Система для моніторингу якості повітря .....	14
1.2.4 Додатковий аналіз сучасних систем мікроклімату .....	16
1.2.5 Аналіз вимог до програмно-технічного засобу та розробка технічного завдання .....	17
1.3 Аналіз вимог до проєктування системи .....	19
1.3.2 Функціональні вимоги .....	20
1.3.3 Вимоги до апаратного забезпечення .....	21
1.3.4 Вимоги до програмного забезпечення .....	23
1.4 Висновки до першого розділу .....	25
<b>2 АНАЛІЗ СКЛАДОВИХ СИСТЕМИ ТА ЗАСОБІВ ЇЇ РОЗРОБКИ</b> .....	26
2.1 Розробка алгоритму функціонування .....	26
2.2 Вибір мікроконтролера .....	28
2.3 Вибір компонентів кіберфізичної систем .....	36
2.4 Підключення компонентів .....	50
2.5 Висновки до другого розділу .....	53
<b>3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ АГЕНТУ МІКРОКЛІМАТУ ЛАБОРАТОРІЇ КІБЕР-ФІЗИЧНОЇ СИСТЕМИ</b> .....	54

КІРКІ 210108.21.01.55 ПЗ

Зм.	Арк.	№докум.	Підпис	Дата	Літера	Аркуш	Аркушів
Виконав		Олександр Жолудж	<i>[Signature]</i>	2023			
Перевір.		Людмила Корсуняк	<i>[Signature]</i>	2023			
Н.контр.		Тетяна КИСІЛЬ	<i>[Signature]</i>	2023			
Затверд.		Ольга ПАВЛОВА	<i>[Signature]</i>	2023			

Кіберфізична система моніторингу параметрів мікроклімату у медичному закладі  
Пояснювальна записка

ХНУ КІ2-21-1

3.1 Вимоги до програмного забезпечення .....	54
3.2 Вибір мови програмування .....	56
3.3 Використання GitHub для контролю версій .....	60
3.4 Опис програмної реалізації чат-боту.....	64
3.5 Підключення бібліотек.....	65
3.5 Інтеграція з месенджер Telegram.....	68
3.6 Інтеграція з Vlynk .....	70
3.7 Опис програмної частини .....	72
3.8 Керування Vlynk.....	74
3. 9 Висновки до третього розділу .....	76
<b>ВИСНОВКИ</b> .....	78
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ</b> .....	80
ДОДАТОК А .....	84
ДОДАТОК Б.....	85
ДОДАТОК В.....	86
ДОДАТОК С.....	87

## ВСТУП

Сучасні медичні установи дедалі частіше стикаються з необхідністю впровадження автоматизованих систем для моніторингу мікроклімату - зокрема вологості й температури. Від стабільних кліматичних умов у лікарняних приміщеннях залежать не лише швидкість одужання пацієнтів, а й справність медичного обладнання та комфорт персоналу.

Об'єкт дослідження – процес забезпечення та контролю параметрів мікроклімату у приміщеннях медичного закладу.

Предмет дослідження – кіберфізична система моніторингу, яка забезпечує збір, аналіз та реагування на параметри мікроклімату у медичному закладі за допомогою вбудованих сенсорів, контролерів, програмного забезпечення та інтерактивних інтерфейсів.

Кіберфізичні технології надають можливість у режимі реального часу фіксувати параметри середовища, аналізувати їх і оперативно вносити корективи, забезпечуючи стабільність температурно-вологісного режиму без участі людини. Це знижує ризики помилок та підвищує загальну ефективність функціонування медзакладів.

Мета роботи - розробити кіберфізичну систему для моніторингу вологістю і температурою у приміщеннях медичних закладів.

Для досягнення поставленої мети потрібно виконати наступні завдання:

- описати вимоги до мікроклімату в лікарнях;
- провести аналіз існуючих технічних рішень;
- підібрати компоненти кіберфізичних систем (КФС);
- розробити алгоритм збору й обробки даних, алгоритм роботи КФС;
- розробити програмне забезпечення кіберфізичної системи.

Кінцевий результатом виконання перерахованих завдань стане прототип КФС, здатний до автономного моніторингу мікрокліматичних параметрів у медичних закладах та оперативного інформування персоналу, що забезпечує

					КІРКІ 210108.21.01.55 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

підвищення якості лікування, покращенню умов праці медичного персоналу та продовженню строку служби обладнання, чутливого до змін клімату.

					КІРКІ 210108.21.01.55 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

# 1 КІБЕРФІЗИЧНА СИСТЕМА МОНІТОРИНГУ ПАРАМЕТРІВ МІКРОКЛІМАТУ У МЕДИЧНОМУ ЗАКЛАДІ

## 1.1 Актуальність задачі

У сучасних медичних закладах забезпечення оптимального мікроклімату є надзвичайно важливим - від температурного режиму, рівня вологості та чистоти повітря залежить не лише комфорт пацієнтів, але й ефективність роботи високоточних медичного обладнання, безпека операцій та загальний стан здоров'я людей [1]. Порушення нормативних вимог до температури та вологості може призвести до розвитку інфекційних процесів, погіршення функціонування електронного обладнання, а також пошкодження ушкодження будівельних і інтер'єрних матеріалів, що в довгостроковій перспективі негативно впливає на стабільність роботи лікарні [2]. Сучасна інфраструктура вимагає інтеграції різних технологічних рішень, зокрема систем збору й аналізу даних, що дозволяють оперативно і точно реагувати на будь-які відхилення від заданих параметрів мікроклімату.

В умовах війни в Україні, коли медична інфраструктура зазнає значних навантажень та пошкоджень, а ризики інфекційних захворювань зростають через антисанітарію та скупчення людей, забезпечення стабільного та контрольованого мікроклімату набуває ще більшого значення. Недостатній контроль може призвести до критичних збоїв, тому автоматизація цього процесу стає не просто бажаною, а життєво необхідною. Перебої з енергопостачанням, руйнування вентиляційних і кондиціонерних систем, а також збільшення навантаження на інфраструктуру викликають потребу розробки комплексних рішень для автоматичного моніторингу та керування мікрокліматом. У перспективі інтеграція сучасних технологій дозволяє не лише виявляти відхилення в режимах роботи, а й прогнозувати можливі несправності за допомогою алгоритмів машинного навчання [3], що забезпечує додатковий рівень безпеки. Розробка кіберфізичної

					КІРКІ 210108.21.01.55 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

системи, що дає змогу автоматично контролювати ці параметри, є надзвичайно актуальною для лікарень, операційних та лабораторій [4]. Застосування бездротових сенсорних мереж та IoT-технологій сприяє підвищенню точності вимірювань і забезпеченню централізованого миттєвого моніторингу всіма параметрами мікроклімату [5].

Автоматизовані системи не тільки реагують на зміни, але й накопичують дані, які можна використовувати для подальшого вдосконалення управління мікрокліматом. Для своєчасного виявлення небезпечних ситуацій може використовуватися датчик вібрації, що дозволяє відстежувати раптові механічні коливання та потенційні руйнування в конструкціях будівель. Також, для забезпечення безпеки пацієнтів та медичного персоналу, вкрай важливим є контроль рівня шкідливих газів у повітрі, адже навіть незначне підвищення концентрації токсинів може призвести до зниження працездатності або появи гострих симптомів у осіб з підвищеною чутливістю. Завдяки цьому з'являється можливість аналізувати історичні дані для виявлення тенденцій та розробки стратегічних планів з модернізації медичних установ. У підсумку такі системи стають інструментом стратегічного значення - гарантією безпеки, ефективності та стійкості медичної системи в часи кризи, дозволяючи вести системний аналіз та прогнозування потенційних ризиків. Крім того, зібрана інформація допомагає оптимізувати витрати на обслуговування та модернізацію інфраструктури.

Окрім контролю мікроклімату, важливим є моніторинг інших критичних параметрів, які можуть впливати на загальний стан безпеки в лікарняно-санітарному середовищі. Системна інтеграція всіх вищезазначених модулів у єдину кіберфізичну мережу створює міцну основу для майбутніх досліджень і впровадження інновацій у галузі охорони здоров'я.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

## 1.2 Аналіз та дослідження аналогів або інших систем

Ключовим етапом розробки будь-якої інноваційної системи, зокрема агента мікроклімату для лабораторій кіберфізичних систем (КФС), є уважний аналіз вже існуючих розробок та технологій. Такий аналіз дозволяє встановити ключові критерії для майбутнього проектування, збагнути поточні ринкові тенденції та виокремити особливості кожного рішення. Одночасно це допомагає виявити слабкі місця наявних систем, що може стати поштовхом для розробки нового функціоналу, який зробить агента унікальним та пристосованим до потреб лікувально-санітарного середовища. На сьогоднішній день представлено безліч систем для моніторингу мікроклімату, проте переважна більшість з них забезпечує моніторинг лише деяких параметрів, наприклад, температури, і не враховує комплексний характер завдань лікувально-санітарних умов. Для медичних закладів, де критично важливо підтримувати не тільки оптимальну температуру, але й належний рівень вологості, чистоту повітря та інші мікрокліматичні показники, це створює суттєві обмеження.

Особливої уваги варті рішення, збудовані на базі мікроконтролера ESP32, який завдяки своїй енергоефективності, можливостям бездротового передавання даних та відносно низькій вартості широко використовується в системах моніторингу. Системи на його основі зазвичай використовують різні датчики для зчитування температурних показників, вологості та інших параметрів. Проте багато з цих рішень мають обмежені можливості вони забезпечують односторонній контроль (наприклад, лише моніторинг температури), не дозволяючи інтегрованого керування або ручного налаштування критичних значень, що є надзвичайно важливим для медичних закладів. Окрім того, існуючі системи часто потребують встановлення додаткових програм або зовнішніх серверних рішень для обробки й аналізу даних, що ускладнює їх використання в умовах обмежених ресурсів лікарень та клінік [6,7]. Це особливо актуально в сучасних умовах, коли надійність системи має бути гарантом безпеки пацієнтів, а

					КІРКІ 210108.21.01.55 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

оперативність є засобом мінімізації ризиків, пов'язаних із змінами мікроклімату. Ще одним важливим недоліком є відсутність інтеграції існуючих систем з іншими елементами інфраструктури лікарень, зокрема з HVAC-системами або централізованими системами управління. Така ізольованість обмежує можливості автоматизації та створення єдиної платформи для комплексного управління мікрокліматом, що є критичним у випадках кризових ситуацій або перебоїв з енергопостачанням.

Тому, регулювання мікроклімату набуває надзвичайно важливого значення у різних сферах діяльності від агровиробництва до архівної справи та житлового будівництва. Аналіз аналогів існуючих кіберфізичних систем моніторингу параметрів мікроклімату дозволяє охопити як технологічні, так і економічні аспекти ринку, визначити ключові функції кожного рішення, зокрема розумної системи поливу грядок, системи моніторингу доквілля, системи для контролю якості повітря.

HVAC-системи класифікуються за такими параметрами:

- 1) за призначенням:
  - a) системи опалення;
  - b) системи вентиляції;
  - c) системи кондиціонування;
  - d) комплексні системи ОВК (HVAC).
- 2) за типом будівлі:
  - a) житлові;
  - b) комерційні;
  - c) промислові.
- 3) за типом обладнання:
  - a) централізовані;
  - b) децентралізовані;
  - c) системи вентиляції з природним спонуканням;
  - d) системи вентиляції з механічним спонуканням.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 9
Зм.	Арк.	№ докум.	Підпис	Дата		

4) за принципом роботи:

a) пасивні системи;

b) активні системи.

					КІРКІ 210108.21.01.55 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

### 1.2.1 Розумна система поливу грядок

Система базується на мікроконтролері ESP32, датчику вологості ґрунту та хмарному сервісі Blynk, які забезпечують моніторинг і контроль стану грядок, а також управління водяним насосом у разі необхідності виконання встановлених умов. Інтеграція сучасних технологій дозволяє системі швидко адаптуватися до змін зовнішнього середовища, забезпечуючи її стабільну і ефективну роботу в різноманітних кліматичних умовах.

Ключовими функціями системи є:

- контроль вологості ґрунту для визначення оптимальних моментів поливу та мінімізації втрат води;
- моніторинг погодних умов у реальному часі, що дозволяє своєчасно виявляти несприятливі зміни, які можуть вплинути на потреби рослин;
- віддалене управління через мобільний додаток, що забезпечує зручний доступ до системи навіть поза межами господарства;
- автоматичний режим поливу, який реалізує процес без участі користувача та сприяє рівномірному зволоженню всієї площі грядки.

Основними перевагами цієї системи є її енергоефективність, що дозволяє економно використовувати ресурси води та електроенергії, а також порівняно низька вартість впровадження. Пристрій характеризується простотою збірки апаратної частини, що полегшує монтаж та технічне обслуговування, а також можливістю модернізації для розширення функціональних можливостей. Детальну схему компонування елементів системи зображено на рисунку 1.1.

Однак система має певні недоліки: вона залежить від погодних умов, а її сумісність обмежується використанням виключно мікроконтролерів серії ESP32, що може ускладнювати інтеграцію з іншими типами обладнання. Крім того, відсутність функції ручного управління поливом може створювати труднощі при непередбачених змінах або екстрених ситуаціях. Система, що застосовується в агровиробництві, спрямована на забезпечення стабільних умов для росту рослин завдяки інтеграції датчиків для вимірювання температури, вологості, освітлення,

					КІРКІ 210108.21.01.55 ПЗ	Арк. 11
Зм.	Арк.	№ докум.	Підпис	Дата		







систему придатною для застосування як у стаціонарних, так і у мобільних проектах. Схему підключення компонентів можна побачити на рисунку 1.4.

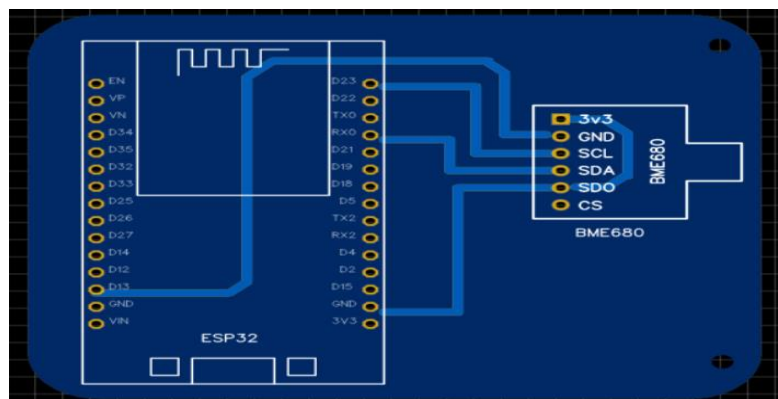


Рисунок 1.4 – Схема підключення системи моніторингу якості повітря

Однак, незважаючи на очевидні переваги, дана система має певні обмеження. Одним із недоліків є відсутність можливості гнучкої кастомізації: прямий зв'язок із датчиком BME680 означає, що система отримує набір даних, який може містити інформацію, що не завжди є корисною для певних завдань проекту. Наприклад, надмірна деталізація вимірювань деяких параметрів може ускладнювати аналіз даних, коли потрібно зосередитися лише на основних показниках якості повітря. Така залежність від одного типу датчика може обмежити адаптивність системи до спеціалізованих потреб користувача, змушуючи розробників шукати додаткові рішення для оптимізації обробки даних.

Отже, що система для моніторингу якості повітря є ефективним рішенням для первинного аналізу стану навколишнього середовища завдяки простоті реалізації та використанню мінімального набору компонентів. Проте важливо враховувати, що для деяких проектів може знадобитися додаткова настройка або розширення функціональності для відсіювання непотрібних даних та забезпечення більш цілеспрямованого моніторингу.

Зм.	Арк.	№ докум.	Підпис	Дата

## 1.2.4 Додатковий аналіз сучасних систем мікроклімату

Автоматизоване керування системою впливає на вентиляцію, зрошення, підігрів або охолодження залежно від отриманих показників. Завдяки використанню AI-алгоритмів, система здатна прогнозувати зміни мікроклімату з урахуванням погодних умов, що сприяє оптимізації витрат води та електроенергії [10].

Такий підхід може призводити до збільшення врожайності до 30 % у тепличних умовах, поліпшення здоров'я та продуктивності тварин, а також забезпечувати можливість віддаленого керування через мобільний додаток, що зменшує ризик захворювань завдяки підтриманню стабільних умов.

Водночас, ціна таких систем коливається від приблизно 5000 до 20000 гривень, залежно від масштабу господарства та рівня автоматизації. Проте, недоліками цих рішень є висока залежність від стабільного інтернет-з'єднання, значні початкові інвестиції для малих фермерських господарств, потреба у кваліфікованому персоналі для налаштування й обслуговування, а також ризики кібербезпеки при використанні онлайн-платформ.

Для архівів, музеїв, бібліотек та сховищ антикваріату застосовують спеціалізовані осушувачі повітря, які не лише знижують вологість, але й створюють стабільне середовище, що дозволяє уникнути руйнування чутливих матеріалів, таких як папір, шкіра, деревина та інші гігроскопічні елементи [10,11,12]. Ці прилади характеризуються високою точністю регулювання вологості (до  $\pm 3\%$ ) і можуть бути інтегровані із сигналізаційними системами, що спрацьовують у разі перевищення критичних параметрів. Крім того, вони запобігають утворенню конденсату, який може призвести до корозії металевих елементів або пошкодження експонатів, а також забезпечують фільтрацію повітря для видалення пилу та інших шкідливих домішок. Завдяки можливості програмування режимів роботи відповідно до типу зберезуваних матеріалів, такі системи сприяють продовженню терміну служби як експонатів, так і будівельних

					КІРКІ 210108.21.01.55 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

матеріалів у приміщеннях архівів та музеїв. Вартість подібних пристроїв зазвичай коливається від 3000 до 15000 гривень, хоча їх недоліками є обмежене застосування як універсального рішення, необхідність регулярного обслуговування, підвищене споживання електроенергії при тривалій роботі та вимога врахування об'єму приміщення для правильного вибору потужності.

Системи контролю мікроклімату для будівель орієнтовані на створення комфортних умов для людей у житлових і офісних приміщеннях, торгових центрах, лікарнях, школах та навіть у контексті розумних міст [13]. Недоліками таких систем є складність налаштування в уже збудованих об'єктах, потреба у постійному оновленні програмного забезпечення, залежність від кваліфікованих спеціалістів для встановлення й обслуговування, а також можливі проблеми сумісності з наявними HVAC-системами й ризики витоку персональних даних при використанні хмарних сервісів керування. Крім того, кожен тип систем має свої переваги та обмеження: висока енергоефективність є ключовою перевагою для будівельних рішень, тоді як для архівів особливо важливо забезпечувати безшумну роботу.

#### 1.2.5 Аналіз вимог до програмно-технічного засобу та розробка технічного завдання

У аналізі вимог наведено комплексний огляд потреб до кіберфізичної системи моніторингу параметрів мікроклімату у медичному закладі, що спирається на результати аналізу предметної області, нормативних обмежень та вивчення доступних технологічних рішень. В умовах війни в Україні, коли стабільність інфраструктури часто порушується, особлива увага приділяється надійності роботи таких систем навіть за складних зовнішніх обставин. Розглянуто два головні аспекти: по-перше, інтеграцію трьох критичних сенсорних підсистем (температура / вологість, вібрація, чадний газ); по-друге, організацію віддаленого моніторингу через Telegram-бота [14], який забезпечує своєчасне

					КІРКІ 210108.21.01.55 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

сповіщення відповідальних спеціалістів - незалежно від їхнього фізичного місцезнаходження.

Визначення потреб до системи починається з аналізу умов експлуатації у середовищі медичного закладу, де стабільність параметрів мікроклімату є критично важливою для збереження стерильності, належного функціонування обладнання та безпеки пацієнтів. В умовах частих перебоїв з електропостачанням і підвищених ризиків пошкодження інфраструктури, підтримання стабільного мікроклімату стає завданням підвищеної пріоритетності. Температура і вологість повинні залишатися у встановлених межах відповідно до нормативних актів, таких як ДБН та санітарні правила. Вібраційні навантаження мають бути мінімізовані, особливо у приміщеннях із чутливим обладнанням. Моніторинг рівня чадного газу є обов'язковим через можливе використання резервних джерел енергії в екстрених ситуаціях.

Технічне завдання також враховує потреби до масштабованості та адаптивності системи, оскільки в умовах воєнних дій можлива термінова зміна конфігурації приміщень або розгортання тимчасових медичних закладів. Сенсорні модулі мають підтримувати як дротову, так і бездротову передачу даних, що забезпечить максимальну гнучкість розміщення пристроїв у змінному середовищі. Другий важливий аспект - організація віддаленого моніторингу через Telegram-бот набуває особливої актуальності через те, що в умовах війни персонал може бути обмежений у можливості фізичного доступу до обладнання. Розробка програмного забезпечення передбачає не лише збір даних, миттєве повідомлення та аналіз мікрокліматичних параметрів у медичних закладах.

Задля забезпечення безперервної роботи навіть у разі перебоїв енергопостачання або руйнування окремих каналів зв'язку, технічне завдання встановлює високі вимоги до автономності сенсорних пристроїв. Передбачено використання акумуляторів та резервних ліній зв'язку.

Окрему увагу приділено інформаційній безпеці - в умовах воєнного часу питання захисту даних набуває особливої гостроти. Система передбачає

					КІРКІ 210108.21.01.55 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

шифрування переданих даних, автентифікацію користувачів, обмеження доступу до налаштувань та ретельне реєстрації всіх подій для забезпечення прозорості роботи і можливості аудиту в критичних ситуаціях [15].

Таким чином, технічне завдання сформульоване як чіткий, вимірюваний набір потреб до функціональних, технічних та експлуатаційних характеристик системи. Запропонована архітектура створює основу для подальшого етапу проектування та реалізації кіберфізичної системи моніторингу параметрів мікроклімату у медичних закладах, що працюють в умовах надзвичайних викликів та нестабільності.

### 1.3 Аналіз вимог до проектування системи

Завдання системи полягає у постійному моніторингу параметрів, що безпосередньо впливають на безпеку пацієнтів, функціональність медичного обладнання та точність лабораторних досліджень. З цією метою сенсори розташовуються в приміщеннях з особливими вимогами до мікроклімату, а саме: відділеннях інтенсивної терапії, операційних блоках, лабораторіях. Система генерує потік даних у реальному часі, їх обробка відбувається на місці з мінімальною затримкою, а при виході значень за встановлені межі, негайно відправляється тривожне сповіщення у Telegram-канал, що забезпечує миттєве реагування персоналу. Тому така архітектура в перспективі, дозволяє інтегрувати апаратне забезпечення (датчики та контролери) з хмарними сервісами та мобільними додатками, утворюючи цілісну кіберфізичну екосистему. У ній контрольовані об'єкти (лікарняні приміщення) перетворюються на "цифрових двійників", а їхній поточний стан відображається через об'єктивні числові показники, що доступні у будь-який час. Крім того, концепція передбачає можливість подальшого розширення функціоналу шляхом підключення додаткових датчиків, наприклад, для моніторингу рівня чадного газу та

					КІРКІ 210108.21.01.55 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

освітленості. Цей модульний підхід дає змогу швидко адаптувати систему до нових вимог без необхідності кардинальної перебудови базової інфраструктури.

Система повинна забезпечувати мультиплатформенну взаємодію, гарантуючи доступ до інформації як через мобільні додатки, так і через веб-інтерфейс для адміністратора, що дозволяє проводити глибший аналіз динаміки змін параметрів за тривалі періоди та формувати відповідні звіти.

### 1.3.2 Функціональні вимоги

Основою функцією системи є збір, валідація та передавання показників критичних параметрів. Для вимірювання температури та відносної вологості планується використання датчика SHT31 з точністю  $\pm 0,1^{\circ}\text{C}$  та  $\pm 2\%$  відповідно. Частота опитування встановлена на рівні десяти секунд, що забезпечує достатню деталізацію даних без надмірного навантаження на радіоканал. Моніторинг вібрацій буде здійснюватися за допомогою тензометричного акселерометра ADXL345, здатного фіксувати коливання у діапазоні від нуля до двохсот герц. Інтеграція цього датчика дозволить не тільки виявляти механічні несправності (наприклад, проблеми у роботі компресорів або вентиляційних систем), але й відкриває можливості для проведення превентивної діагностики обладнання. Контроль вмісту чадного газу буде реалізовано за допомогою датчика MQ-7, чутливого до діапазону 0-1000 ppm; враховуючи особливості аналогового виходу, передбачено апаратну та програмну калібровку показань. Уся сенсорна інформація одразу зберігається у локальній SPI-Flash-пам'яті, де зберігається щонайменше 24-годинна історія з інтервалом у одну хвилину. Застосування ковзного середнього по п'яти останніх вимірах дозволяє згладити короточасні стрибки, викликані випадковими коливаннями, та зменшити кількість хибних спрацьовувань. Передбачено механізм детекції аномалій на основі аналізу порогових значень (threshold anomaly detection) з можливістю гнучкого налаштування адаптивних порогів.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

Підключення до серверної частини забезпечується через Wi-Fi за протоколом з шифруванням TLS 1.2; на рівні QoS 1 гарантується доставка кожного повідомлення. У випадку втрати з'єднання Wi-Fi, модуль автоматично перемикається на резервний канал Bluetooth LE. У перспективі розглядається підтримка мереж LoRaWAN для забезпечення наддалекого зв'язку у великих лікарняних комплексах [16].

Взаємодія з користувачем відбувається через Telegram-бот: після запуску командою /start він реєструє ID чату, що дозволяє надалі надсилати сповіщення про перевищення порогових значень, а також обробляти команди управління (/status, /set\_threshold, /health тощо). Для підвищення зручності користування передбачені опції налаштування частоти сповіщень, встановлення персональних порогових значень тривоги для кожного сенсора та створення групових сповіщень для різних відділень медичного закладу.

Планується реалізація функції реєстрації усіх критичних подій із можливістю перегляду історії тривог та формування аналітичних звітів про ефективність реагування персоналу на сповіщення. Це дозволить не тільки контролювати стан середовища, але й оптимізувати внутрішні протоколи безпеки у лікарні.

### 1.3.3 Вимоги до апаратного забезпечення

Оскільки центральним обчислювальним елементом обрано модуль ESP32-WROOM-32 з двоядерним процесором Xtensa частотою 240 МГц, апаратна платформа повинна забезпечувати достатню кількість виводів для підключення усіх сенсорів та модулів живлення. Деталізовані характеристики основних компонентів подано у таблиці 1.1.

Передбачено також встановлення додаткового моніторингового порту UART для налагодження пристроїв у режимі реального часу без переривання

					КІРКІ 210108.21.01.55 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

основної роботи системи. Усі компоненти повинні мати сертифікацію відповідно до стандартів електробезпеки, що діють у медичних установах.

					КІРКІ 210108.21.01.55 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

Таблиця 1.1 - Характеристики компонентів розділити таблицю

Компонент	Характеристики	Примітки
MCU	ESP32-WROOM-32, 2×240 МГц, Wi-Fi / BT	Потребує вентиляційних отворів у корпусі для антен
Темп./вологісний датчик	SHT31, I <sup>2</sup> C, ±0,1 °C; ±2 % RH	Монтується у зоні рівномірного потоку повітря
Вібраційний датчик	ADXL345, I <sup>2</sup> C або SPI, 0–200 Hz	Кріпиться через антивібраційні прокладки
Датчик CO	MQ-7, аналоговий вихід, 0–1000 ppm	Потрібна калібровка, використовуються 12-бітні ADC-канали ESP32
Пам'ять	SPI-Flash 4 МБ	Використовується для кешу та системних логів
Живлення	5 В DC (USB або PoE), 3 А	Захист від переполюсовки; на виході стабілізовані 3,3 В

### 1.3.4 Вимоги до програмного забезпечення

Програмне забезпечення буде розроблено на базі середовища ESP-IDF версії 4.x із застосуванням стандарту C++17, що дає змогу ефективно організувати паралельне виконання завдань за допомогою FreeRTOS. Архітектура програмного коду чітко структурована й поділена на два ключових модуля

SensorManager виконує налаштування сенсорів, здійснює їх періодичне опитування, фільтрацію даних і передає результати у загальну чергу подій.

						КІРКІ 210108.21.01.55 ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата			

Модуль також обробляє необроблені ("сирі") вимірювання та обчислює усереднені значення для підвищення стабільності показників.

TelegramBot забезпечує взаємодію із Telegram-API через HTTPS-запити, відповідає за формування та надсилання сповіщень, а також обробляє команди, які надходять від користувачів. Вбудована підтримка пріоритетних push-сповіщень дозволяє оперативно передавати критичні тривоги.

Для розробки використовуються перевірені та стабільні бібліотеки: esp-mqtt, mbedTLS, Arduino-JSON та UniversalTelegramBot, які відзначаються хорошою документацією і активною підтримкою спільноти розробників. Механізм оновлення прошивки реалізований через захищене HTTPS-з'єднання із валідацією цифрового сертифіката сервера, що гарантує автентичність джерела оновлень і запобігає впровадженню шкідливого ПЗ [17,18].

Підсумовуючи, проведений аналіз довів, що запропонована архітектура повністю відповідає вимогам медичних закладів до цілодобового та надійного моніторингу мікрокліматичних параметрів. Використання високоточних сенсорів для контролю температури, вологості, вібраційних навантажень і рівня чадного газу дозволяє своєчасно виявляти навіть мінімальні відхилення від нормативних показників. Енергоефективна апаратна платформа на базі ESP32 забезпечує достатній рівень обчислювальних ресурсів для обробки даних у реальному часі без суттєвого енергоспоживання, що особливо важливо для систем із гарантованою автономністю.

Захищений канал передачі даних TLS-шифруванням гарантує цілісність та конфіденційність інформації, що критично для середовища із підвищеними вимогами до інформаційної безпеки, таких як медичні установи. Завдяки інтеграції з Telegram користувачі отримують сповіщення практично миттєво, незалежно від свого фізичного розташування, що значно скорочує час реакції на можливі загрози [19,20].

У перспективі архітектура системи допускає подальший розвиток - розширення спектру контрольованих параметрів, інтеграцію алгоритмів штучного

					КІРКІ 210108.21.01.55 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

інтелекту для прогнозної аналітики та застосування в інших критично важливих об'єктах, таких як фармацевтичні склади, центри реабілітації або біотехнологічні лабораторії. Запропонована система становить міцний фундамент для впровадження актуальних стандартів моніторингу в медичній галузі та надає великі перспективи для подальшої модернізації.

#### 1.4 Висновки до першого розділу

У результаті проведеного аналізу було підтверджено нагальну необхідність впровадження автоматизованої кіберфізичної системи для моніторингу мікроклімату в медичних установах, особливо в умовах війни, коли стабільність інфраструктури є критично важливою. Аналіз існуючих рішень виявив, що більшість із них не охоплюють весь спектр параметрів, не підтримують гнучке налаштування та не гарантують безпеку передачі даних. На основі цього сформульовано вимоги до нової системи, яка передбачає моніторинг температури, вологості, вібрацій та концентрації чадного газу з використанням енергоефективного модуля ESP32, високоточних сенсорів і Telegram-бота для сповіщення. Запропонована модульна архітектура забезпечує гнучкість, масштабованість і стійкість до перебоїв, а використання стандартів безпеки та протоколів шифрування гарантує захист даних. Отримані результати створюють надійну основу для розробки, впровадження та подальшої модернізації системи в медичних закладах.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2 АНАЛІЗ СКЛАДОВИХ СИСТЕМИ ТА ЗАСОБІВ ЇЇ РОЗРОБКИ

### 2.1 Розробка алгоритму функціонування

Файл `config.h` містить сукупність визначень, які відповідають за параметри підключення і апаратні налаштування: тут формалізовано ідентифікатори та токени для сервісів Blynk і Telegram, задано номери GPIO для інтерфейсів датчиків та індикаторів, а також встановлено поріг спрацьовування сенсора CO і часову константу антидребезгу. Окремо виділено структуру `ConnectionSettings`, в якій зосереджено SSID і пароль Wi-Fi мережі разом із токеном Blynk, що дозволяє організувати централізований контроль та уніфікований доступ до цих даних у будь-якій частині програми.

Основний файл `main.ino` починається із підключення низки бібліотек, необхідних для реалізації бездротового зв'язку, роботи з датчиками DHT, інтерфейсом Blynk, клієнтом Telegram та OLED-дисплеєм SH1106. В ході ініціалізації за допомогою функції `setup()` відбувається запуск серійного інтерфейсу для відлагодження, встановлення з'єднання з мережею Wi-Fi та Blynk, налаштування таймерів та конфігурація GPIO-пінів. Визначений часопрограмований механізм система періодичних викликів функцій через об'єкт `BlynkTimer` відповідає за передачу показань датчиків, обробку вхідних повідомлень і моніторинг стану сенсорів із заданими інтервалами. Після завершення етапу ініціалізації OLED-дисплей оновлюється показом поточного часу та температури, синхронізованого через NTP-сервіс. У безперервному циклі `loop()` підтримується зв'язок із платформою Blynk і виконуються заплановані задачі. Цей підхід забезпечує реактивність системи, дозволяючи своєчасно реагувати на зміну показників навколишнього середовища і команди користувача.

Функція `sendSensorData()` реалізує зчитування температури та вологості з датчика DHT і, у разі відмінності від попередніх значень, формує відповідні віртуальні пиші Blynk, а також оновлює графічне відображення на дисплеї.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

Алгоритм ретельно відслідковує стан зв'язку із точкою доступу і сервером Blynk за допомогою методу checkConnection(), який у разі втрати з'єднання ініціює повторну процедуру підключення до обох сервісів.

Блок-схема алгоритму програмної частини наведена на рисунку 2.1.

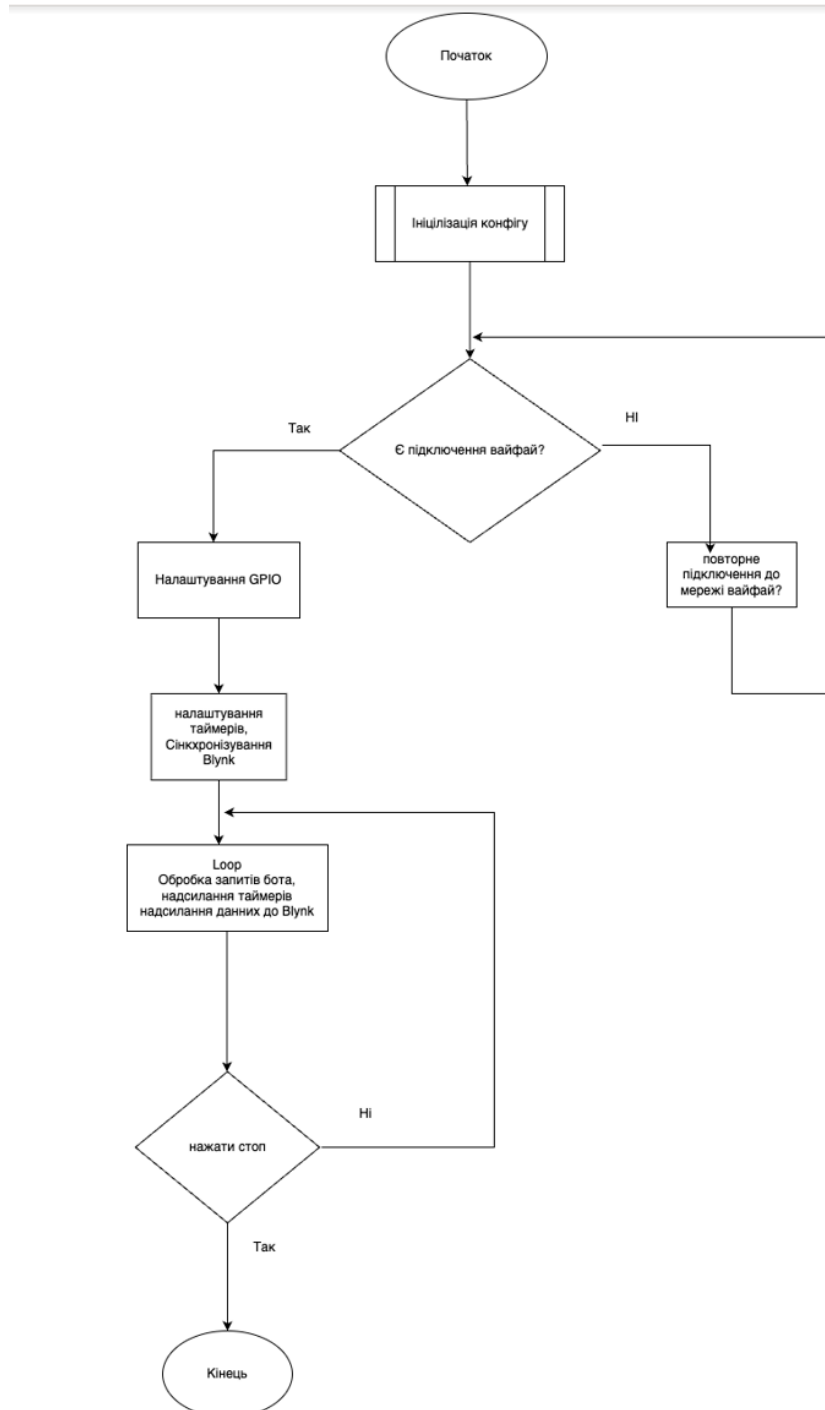


Рисунок 2.1 – Алгоритм роботи коду мікроагента

Важливою частиною рішення є модуль обробки аварійних ситуацій: функція `checkSensors()` читає цифровий вхід від вібраційного сенсора та аналогове значення СО-датчика, аналізує подію натискання кнопки скидання і відповідно керує станом світлодіоду. При виявленні перевищення порогового значення або фіксації вібрації система не лише вмикає індикатор, а й ініціює розсилку Telegram-повідомлень через асинхронні виклики `notifyTelegram()`, що гарантує миттєве інформування оператора без блокування основного потоку виконання. Аналогічно, команда `/gas` або `/vibration`, надіслана користувачем у чаті, викликає миттєву відповідь із поточними значеннями сенсорів.

## 2.2 Вибір мікроконтролера

У цьому огляді розглянуто чотири популярні апаратні платформи Arduino Uno, Arduino Nano ESP32, ESP32 Dev Board і Raspberry Pi 4/5. Їх характеристик наведені у таблиці 2.1. Ці мікроконтролери посідають ключові позиції у сучасній розробці електроніки, автоматизованих систем, побутових та промислових IoT-рішень. Кожна з них орієнтована на свій сегмент завдань, користувацький рівень і обсяг функцій, що забезпечується апаратною та програмною складовою. Вибір тієї чи іншої плати визначається не лише її технічними характеристиками, а й підтримуваними мовами програмування, можливістю масштабування, сумісністю з іншими модулями, доступом до спільноти розробників та ціновими обмеженнями. Саме тому доцільно розглядати їх у порівняльному контексті, враховуючи не лише сухі цифри з даташитів, а й практичні аспекти — від простоти налаштування до реальних сценаріїв використання у польових умовах. Такий підхід дозволяє не просто визначити, яка плата «краща», а яка буде оптимальною для конкретного проєкту чи задачі.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 28
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1 - Параметри мікроконтролерів

Параметр	Arduino Uno	Arduino Nano ESP32	ESP32 Dev Board	Raspberry Pi 4/5
Мікроконтролер / SoC	ATmega328P (8-біт AVR)	ESP32-S3 (32-біт, 2× Xtensa LX7)	ESP32-WROOM-32 (32-біт, 2× LX6)	BCM2711/BCM2712 (64-біт, 4× Cortex-A72)
Оперативна пам'ять	2 КБ SRAM	512 КБ SRAM	520 КБ SRAM	1–8 ГБ LPDDR4 / LPDDR4X
Постійна пам'ять	32 КБ Flash	16 МБ Flash + 8 МБ PSRAM	4 МБ Flash	microSD (до 512 ГБ)
Цифрові I/O пінів	14 (6 PWM)	22+ (всі підтримують PWM)	~30 GPIO	40 GPIO (частково PWM)
Аналогові входи	6 (10-біт)	8 (12-біт)	18 (12-біт ADC)	Відсутні (необхідний зовнішній АЦП)
Комунікаційні шини	UART, I <sup>2</sup> C, SPI	UART×2, I <sup>2</sup> C×2, SPI×2, I <sup>2</sup> S, RMT	UART, I <sup>2</sup> C, SPI, CAN, I <sup>2</sup> S, PWM, SDIO	UART, I <sup>2</sup> C, SPI, USB, CSI/DSI, Ethernet
Мережеві інтерфейси	Відсутні	Wi-Fi 4, Bluetooth 5 BLE	Wi-Fi 4, Bluetooth 4.2	Ethernet, Wi-Fi 5/6, Bluetooth 5
Живлення	7–12 В VIN / 5 В USB	5–18 В VIN / USB-C / 3.3 В	5 В USB / 3.3 В	5 В USB-C (до 3 А)
Габарити, мм	68 × 53	45 × 18	~50 × 25	88 × 58

Arduino Uno перевірена часом мікроконтролерна платформа, що стала своєрідним стандартом для навчання, хобі-розробок та швидкого прототипування. Її переваги це простота, велика кількість навчальних матеріалів, активна спільнота та передбачувана робота. Однак технічні обмеження 8-бітна архітектура, невеликий обсяг пам'яті, відсутність бездротових інтерфейсів і невисока тактова частота ускладнюють її використання в складніших або сучасніших проєктах, які вимагають обробки сигналів, зв'язку з мережею чи

одночасної роботи з багатьма модулями. На противагу цьому, Arduino Nano ESP32 та ESP32 Dev Board базуються на 32-бітних мікроконтролерах з двоядерними процесорами, що дозволя підвищити обчислювальну потужність, а й реалізувати паралельне виконання завдань. Вони мають більше пам'яті, більше інтерфейсів, зокрема підтримку Wi-Fi і Bluetooth, що відкриває двері до розробки бездротових сенсорних вузлів, IoT-інфраструктур, носимих пристроїв або мережевих контролерів. Raspberry Pi 4 і 5, своєю чергою, переходять у зовсім іншу лігу. Це вже не просто плата з мікроконтролером, а одноплатний комп'ютер із багатоядерним процесором, повноцінним графічним виходом, портами USB 3.0, Ethernet, Bluetooth 5 і підтримкою карт пам'яті до пів терабайта. Raspberry Pi здатен запускати повноцінні операційні системи на базі Linux, працювати з дисплеями, камерами, базами даних і хмарними сервісами, перетворюючи просту апаратну плату на центр керування, вебсервер або обчислювальний вузол. У результаті маємо чітке розмежування: Uno для найпростіших завдань, ESP32 для розумних вбудованих систем, Raspberry Pi для повноцінних багатофункціональних рішень з можливістю масштабування.

З програмної точки зору, Arduino Uno обмежується мовою C/C++ у середовищі Arduino IDE або через альтернативні інструменти на кшталт PlatformIO. Проте він не підтримує багатозадачність, скриптові мови чи інтернет-функціонал. Натомість плати на базі ESP32 як Nano ESP32, так і Dev Board відкривають ширші можливості: окрім C/C++ та ESP-IDF, вони підтримують MicroPython, JavaScript (через Espruino), Lua (NodeMCU) та інші середовища. Це робить їх зручними як для розробників-новачків, так і для професіоналів, яким важлива швидкість прототипування. Raspberry Pi, своєю чергою, є фактично настільним комп'ютером, здатним працювати з Python, C++, Go, Java, Node.js, Rust та багатьма іншими мовами, включно з повною підтримкою Docker, TensorFlow Lite, OpenCV, веб-фреймворків і графічних середовищ.

Щодо практичного використання, Arduino Uno найчастіше застосовується в простих проектах зі збору даних з датчиків або керування виконавчими

					КІРКІ 210108.21.01.55 ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

механізмами. Його обирають для навчання, бо він зрозумілий, стабільний і недорогий. Arduino Nano ESP32, маючи підтримку Wi-Fi і Bluetooth, більше орієнтований на розробку IoT-рішень, носимих пристроїв і бездротових сенсорів. ESP32 Dev Board є універсальною платформою для промислових контролерів, систем розумного дому, аудіоінтерфейсів, оскільки забезпечує багато GPIO-пінів, підтримку CAN-шини, I<sup>2</sup>S та інших спеціалізованих інтерфейсів. Raspberry Pi 4 або 5 використовується всюди, де потрібні складні обчислення, обробка зображень, керування інтерфейсами через повноцінну операційну систему або навіть запуск серверних рішень. Це ідеальний вибір для побудови вебсерверів, медіацентрів, STEM-лабораторій, систем розпізнавання об'єктів чи навчання нейромереж.

Цінове порівняння цих платформ підкреслює їх різне позиціонування на ринку. ESP32 Dev Board абсолютний лідер за співвідношенням ціна/можливості: за \$6-10 користувач отримує двоядерний мікроконтролер із Wi-Fi, Bluetooth, великою кількістю I/O-портів і підтримкою сучасних протоколів. Це дозволяє створювати повноцінні IoT-системи без суттєвих витрат, що особливо цінно в умовах обмеженого бюджету або при розробці масових пристроїв. Arduino Nano ESP32, маючи схожу апаратну базу, коштує трохи дорожче, але забезпечує зручніший форм-фактор і кращу інтеграцію з екосистемою Arduino. Arduino Uno, хоча й коштує у межах \$20-25, технічно поступається сучаснішим платам, однак зберігає свою популярність завдяки стабільності та простоті. Raspberry Pi 4/5, із ціною від \$35 до \$75, залишається найдорожчою, але й найбільш універсальною платформою. Вона може повністю замінити ПК у низці задач, що обґрунтовує вартість навіть у найвищій конфігурації. Таким чином, цінова різниця між платформами не просто демонструє розрив у продуктивності вона фактично відображає глибину можливостей і рівень інтеграції, який отримає розробник у результаті. Вибір тут не про «дешево» чи «дорого», а про доцільність інвестиції в залежності від технічного завдання.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

Тому, вибір платформи визначається масштабом і складністю проєкту, необхідністю мережевих функцій, обсягом даних для обробки та бюджетом. Якщо йдеться про простий прототип або навчальний проєкт оптимальним буде Arduino Uno. Для розумних IoT-пристроїв, що потребують бездротового підключення, краще підходить Arduino Nano ESP32 або ESP32 Dev Board, залежно від складності. У випадках, коли потрібен повноцінний комп'ютер із багатозадачністю, операційною системою, можливістю запуску серверів чи складних обчислень Raspberry Pi залишається безальтернативним вибором. Тому виходячи з потреб проєкту для реалізації кіберфізичної системи обрано Arduino ESP32.

Для створення системи моніторингу мікроклімату, необхідно реалізоване безперебійне функціонування сенсорів, які мають працювати злагоджено та передавати інформацію до центрального пристрою-комп'ютера, смартфона або планшета. Дані мають надходити в реальному часі, без будь-яких затримок, а система має бути зручною для користувача та максимально автономною. Щоб уникнути необхідності у додатковому обладнанні для обробки інформації, варто обрати платформу, здатну самостійно збирати, обробляти та передавати дані, при цьому підтримуючи бездротовий зв'язок.

Серед найпопулярніших платформ для збору та обробки даних - Arduino й ESP. Arduino ESP32 приваблює простотою: відкрите ПЗ, величезна спільнота, безліч готових бібліотек і датчиків (температурні сенсори, екрани, світлодіоди, реле, датчики руху тощо). Підключення через USB та живлення від зовнішнього джерела дозволяють швидко почати прототипування та тестування.

Переваги платформи Arduino/ESP32:

- простота розробки й навчання: готові бібліотеки, зразки коду, дружній IDE;
- низьке енергоспоживання: оптимально для батарейних застосунків;
- доступна вартість: від 3–5 \$ (ESP32) та ~20 \$ (Arduino Uno);
- прямий контроль GPIO: швидкий відгук без шару ОС;

					КІРКІ 210108.21.01.55 ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підпис	Дата		





мікроконтролерами, такими як ESP8266 чи Arduino Uno на базі ATmega328P. Насамперед, завдяки своєму двоядерному процесору Tensilica Xtensa LX6, який може працювати на частоті до 240 МГц, ESP32 забезпечує значно вищу обчислювальну потужність, що дозволяє виконувати складніші завдання та обробляти більше даних у реальному часі. Це особливо важливо для обробки та передачі даних до систем моніторингу, які потребують високої швидкості обробки та стабільної роботи.

Ще однією важливою перевагою є обсяг пам'яті, 520 КБ оперативної пам'яті та можливість підключення до 16 МБ флеш-пам'яті дають змогу розміщувати й запускати більш складні програми, що відкриває ширші можливості для реалізації амбітних проектів. Для сучасних застосувань, де потрібні висока продуктивність і надійність, це справді критично.

Крім того, ESP32 підтримує до 36 GPIO-пінів і широкий спектр периферійних інтерфейсів, включаючи I2C, SPI, UART, PWM. Це дозволяє легко підключати різні датчики, виконавчі механізми та інші зовнішні пристрої. Це дає змогу розширити кількість датчиків та модифікувати програмне забезпечення, у наслідок чого покрити моніторингом більшу частину площі санітарно-медичних закладів.

Більш того, на борту ESP32 вже є вбудовані датчики, такі як датчик температури та датчик Холла, що робить його ще універсальнішим для розробки різних інженерних рішень.

Для повноцінного функціонування системи було впроваджено ефективний механізм контролю за основними параметрами мікроклімату.. Це критично важливо як для забезпечення комфортних умов перебування пацієнтів, так і для дотримання принципів енергоефективності та безпеки роботи обладнання.

					КІРКІ 210108.21.01.55 ПЗ	Арк.
						35
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2.3 Вибір компонентів кіберфізичної систем

Сенсори серії MQ використовуються для побудови систем моніторингу повітря у різних середовищах від домашніх умов до промислових об'єктів. Вибір конкретного типу сенсора має критичне значення для точності та призначення системи, зважаючи на специфіку детекції, вихідні сигнали, енергоспоживання, габарити та чутливість до навколишніх факторів. У цьому контексті порівняння MQ-7 і MQ-135 дозволяє сформуванню обґрунтоване рішення щодо інтеграції в агент мікроклімату.

MQ-7 є вузькоспеціалізованим сенсором, який призначений для детекції чадного газу (CO) у повітрі. Його чутливий шар на основі діоксиду олова ( $\text{SnO}_2$ ) реагує переважно на CO, а циклічне нагрівання сенсора дозволяє підвищити вибірковість і зменшити похибку за рахунок очищення чутливої поверхні від небажаних домішок. Для цього використовується двофазне живлення, періоди «очищення» та «зчитування» відбуваються по черзі, що трохи ускладнює алгоритм, але суттєво покращує точність.

MQ-135, на відміну від MQ-7, є багатоконпонентним сенсором, який вловлює цілий спектр шкідливих речовин. Він чутливий до летких органічних сполук (VOCs), аміаку, оксидів азоту,  $\text{CO}_2$ , алкоголю, диму та навіть бензолу. Таким чином, MQ-135 можна розглядати як універсальний сенсор для визначення загальної якості повітря, але без здатності точно розрізняти концентрації окремих газів.

Вимірювальний діапазон MQ-7 значно ширший для CO до 10 000 ppm [21]. Це робить його придатним для виявлення потенційно небезпечних концентрацій у замкнених просторах, таких як лабораторії, гаражі чи технічні приміщення. Сенсор вимагає стабільного живлення 5 В і споживає до 150 мА, що варто враховувати при виборі джерела живлення, особливо в автономних системах [21,22].

					КІРКІ 210108.21.01.55 ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		



від потоків водяної пари або агресивних газів, які не входять у перелік детектованих, оскільки це може вплинути на довготривалу стабільність.

#### Практичні висновки та сценарії використання

Якщо необхідно зосередитися саме на виявленні чадного газу з високою точністю MQ-7 буде оптимальним варіантом, особливо якщо потрібне точне моніторування небезпечних умов. Його селективність до CO є головною перевагою.

MQ-135, у свою чергу, не дає такої точності для окремих речовин, але забезпечує широку картину загального забруднення повітря. Це ідеальний вибір для систем, що мають на меті визначати "погіршення повітря", тобто відхилення від норми у будь-якому напрямку. У контексті моніторингу мікроклімату MQ-135 може виступати як індикатор наявності забруднення, після чого вже MQ-7 підтверджує або спростовує наявність саме CO.

Інтеграція обох сенсорів у єдину систему дозволяє реалізувати дворівневу логіку: базовий контроль загального стану повітря (MQ-135) і поглиблений аналіз вмісту CO (MQ-7). Це забезпечує як запобіжне попередження, так і підтвердження небезпечного стану, що важливо для рішень у сфері безпеки.

Поєднання MQ-7 та MQ-135 у рамках одного агента мікроклімату відкриває можливості для більш гнучкого реагування системи, адаптивного аналізу даних та точнішого відображення ситуації. Для автоматизованих систем, що вимагають надійності, раннього попередження і простоти технічної реалізації, саме така комбінація дозволяє створити дійсно ефективну платформу моніторингу якості повітря.

Модуль КУ-002 є компактним та малопотужним компонентом первинного виявлення механічних впливів (вібрацій або ударів), що поєднує просту апаратну конструкцію з цифровим виходом, адаптованим для мікроконтролерних систем. У його основі пружинний механічний сенсор SW-18015P або SW-520D, чутливий до імпульсних механічних коливань. Принцип дії ґрунтується на моментальному замиканні електричного контакту між стрижнем і пружиною внаслідок

					КІРКІ 210108.21.01.55 ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

прискорення або удару, що призводить до фіксації логічного сигналу "LOW" на цифровому виході [24]. У нормальному стані вихід знаходиться в логічному "HIGH" завдяки вбудованому підтягувальному резистору номіналом 10 кОм. Такий підхід спрощує підключення без потреби в додаткових компонентах підтягування на платі мікроконтролера.

Технічні параметри модуля забезпечують його стабільну роботу у широкому температурному діапазоні (від  $-20$  до  $+70$  °C), при номінальному живленні 3,3-5 В DC та струмі споживання до 10 мА. При цьому сенсорна частина демонструє високу зносостійкість понад 100 000 циклів спрацювання що є прийнятним для довготривалої експлуатації у динамічних системах, наприклад, у мобільній робототехніці або розподілених охоронних вузлах [25]. Хоча чутливість сенсора не регулюється, вона є достатньою для виявлення коротких локальних вібрацій, притаманних ударам або механічному дотику, при цьому відсутність аналогового виходу обмежує застосування лише до бінарних сценаріїв (виявлено/не виявлено).

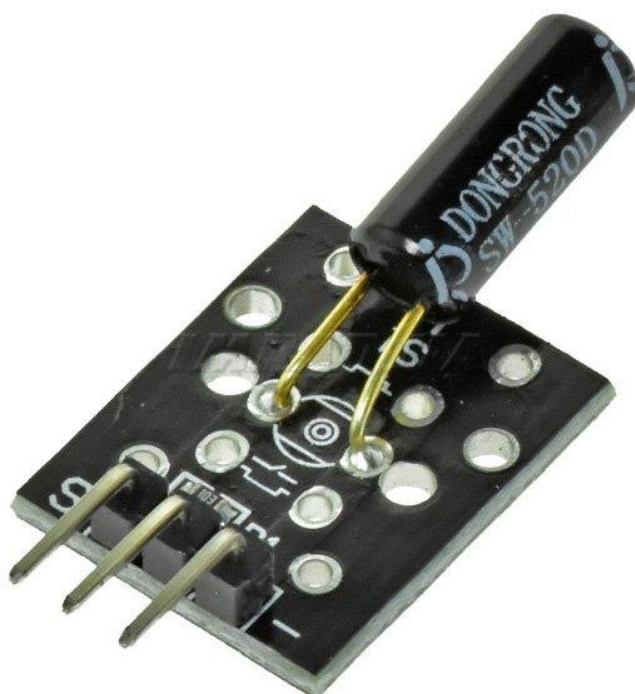


Рис 2.5 – Датчик вібрації

З погляду інтеграції, KY-002 надзвичайно зручний. Його трьохвивідна структура (VCC, GND, OUT) дозволяє підключити його безпосередньо до цифрового GPIO більшості сучасних мікроконтролерів, таких як плати сімейств Arduino, ESP32 або STM32 [25]. Програмно сенсор реалізується як проста кнопка: застосовується конфігурація INPUT\_PULLUP, зчитується логічний стан, при цьому бажано реалізувати програмне усереднення (debounce) для усунення паразитних імпульсів, що можуть виникати через фізичні коливання контактів. Для покращення стабільності сигналу на фізичному рівні доцільно доповнити модуль RC-фільтром низьких частот, що усуває високочастотні наводки та шумові сплески.

KY-002 знайшов широке застосування в охоронних системах як тригер детекції несанкціонованого впливу, в автономній робототехніці ,як простий колізійний сенсор, у системах обліку імпульсних навантажень, а також у побутових пристроях, де необхідно реагувати на зовнішній механічний вплив без складних сенсорних модулів. Проте слід зауважити, що цей сенсор не замінює акселерометр або пізоелектричні системи у задачах, де потрібне точне вимірювання амплітуди вібрацій, їх спектру або динаміки. У таких випадках доцільно комбінувати KY-002 із цифровими MEMS-акселерометрами або аналоговими сенсорами типу piezo для отримання повнішої картини механічної дії.

У системах моніторингу навколишнього середовища, розумного дому, метеостанцій та IoT-рішень вибір сенсора температури й вологості відіграє ключову роль не лише через точність вимірювань, а й через конструкційні, протокольні й енергетичні обмеження, які накладає конкретне середовище застосування. Чотири розглянуті сенсори АНТ21, DHT11, DHT22 (АМ2302) та ВМЕ280 ілюструють широкий спектр компромісів між ціною, розміром, інтерфейсною гнучкістю й функціональністю. Найпростіші у використанні, як-от DHT11, підходять для навчальних або побутових задач з низькими вимогами до

точності й оновлення даних. Їхня цінова доступність і однопровідна передача дозволяють легко інтегрувати їх у невеликі проєкти без потреби в складному ПЗ або апаратних драйверах. DHT22, як вдосконалена версія, суттєво розширює діапазон і точність, що робить його доцільним вибором для аматорських метеостанцій або автономних пристроїв з розширеним діапазоном температур. АНТ21, як сучасний MEMS-чип, демонструє приклад інтеграції промислових технологій у побутові сенсори мініатюрний форм-фактор, низьке енергоспоживання, вже калібровані виходи і надійний I<sup>2</sup>C-протокол дозволяють його застосування у вбудованих системах з жорсткими вимогами до стабільності та ресурсу [26]. BME280, зі свого боку, виходить за межі типового сенсора температури й вологості, додаючи точне барометричне вимірювання, що відкриває шлях до реалізації розширеної екологічної аналітики, визначення висоти, прогнозування погоди й адаптивного управління мікрокліматом. При цьому він поєднує мінімальне споживання з високою частотою опитування, що дозволяє використовувати його як у портативних, так і в стаціонарних системах. Отже, техніко-експлуатаційний вибір сенсора базується не лише на числових характеристиках, а на системній відповідності до архітектури пристрою, бюджету, умов навколишнього середовища та мети вимірювання.

Продовжуючи аналіз, варто окремо зосередитись на архітектурних і протокольних особливостях сенсорів АНТ21, DHT11, DHT22 (АМ2302) та BME280, які визначають їх придатність до конкретних класів вбудованих систем. АНТ21, будучи сенсором нового покоління, побудований на базі інтегрованого ASIC-чипа з цифровою обробкою сигналу та MEMS-структурою для вимірювання вологості, що забезпечує високу повторюваність, стабільність калібрування та лінійність у всьому діапазоні [27]. Його підтримка I<sup>2</sup>C на частотах до 400 кГц дає змогу безпосередньо інтегрувати сенсор у будь-який мікроконтролер, що має апаратну підтримку цієї шини, без додаткових інтерфейсних елементів або часової синхронізації, яка часто є критичною для однопровідних сенсорів [28].

					КІРКІ 210108.21.01.55 ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

DHT11 і DHT22, що використовують власний однопровідний протокол, демонструють іншу модель інтеграції з високою залежністю від програмної реалізації таймінгів, що ускладнює використання в середовищах із обмеженим обробленням переривань або при наявності RTOS. DHT22 покращує точність і стабільність у порівнянні з DHT11, однак обидва сенсори зберігають спільну обмеженість у частоті оновлення та відсутність внутрішньої буферизації, що знижує надійність при інтенсивному опитуванні [29]. Крім того, їхня архітектура не забезпечує апаратної компенсації температурної похибки вологості, що може бути критичним фактором у системах з широким температурним діапазоном.

VME280, у свою чергу, реалізований як повноцінний мультимодальний сенсор з цифровими інтерфейсами I<sup>2</sup>C і SPI, внутрішнім АЦП на 16 біт і підтримкою компенсаційних алгоритмів від виробника. Його гнучкість у конфігурації дозволяє не лише вибирати частоту опитування, а й керувати енергоспоживанням через режими «sleep», «forced» та «normal», що особливо важливо для автономних сенсорних вузлів з обмеженим живленням. Okремо слід зазначити наявність внутрішніх компенсаційних коефіцієнтів, збережених у регістрах, що дозволяє уникнути програмної калібровки, а також вивід абсолютного тиску, який є рідкістю серед побутових сенсорів.

Але при побудові системи виявилось що не всі піни мікроконтролера були зайняті, тому можна розширити моніторинг з 1 приміщення до 3 приміщень, додавши ще датчиків для решти кімнат, це робить систему більш доступною та вигідною для медичних закладів. Фізично датчики можна розташувати з урахуванням призначення кожного приміщення: у палаті DHT11 встановити на висоті близько півтора метра від підлоги, щоб оптимально фіксувати середнє значення температури й вологості, а MQ-7 розмістити поруч із вентиляційними решітками для своєчасного виявлення накопичення чадного газу. Для моніторингу вібраційної активності за допомогою сенсора KY-002 найкращим місцем буде зона поблизу крісел або диванних зон, де потенційно можливі механічні впливи. Датчики у реанімації може вимагати дещо іншого підходу:

					КІРКІ 210108.21.01.55 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

DHT11 корисно винести ближче до вікна, щоб оцінювати ефект провітрювання, а MQ-7 варто встановити на рівні, близькому до голови людини під час сну, аби швидше реагувати в разі накопичення шкідливих домішок [28]. В операційній датчики варто розміщувати таким чином, щоб вони охоплювали максимальну площу простору: температурний модуль краще встановити близько середини кімнати, а вібраційний сенсор у зоні, де монтуються меблі чи встановлено обладнання, що може створювати вібрації в процесі роботи. Обмін інформацією між мікроконтролером та користувачем організовано через єдиного Telegram-бота, який підтримує одночасний моніторинг усіх трьох локацій. У такому режимі користувач може надіслати запит на стан будь-якого приміщення або отримувати автоматизовані сповіщення у разі перевищення заданих порогів наприклад, аварійне повідомлення про надмірне накопичення CO або виявлення механічних коливань у зоні відпочинку. За допомогою бот-інтерфейсу стає можливим не лише одиначне отримання даних, а й формування загальних звітів, які у вигляді графічних чи табличних прикріплень надсилаються у чат протягом доби. Це відкриває можливість віддалено відстежувати стан мікроклімату та безпеки у реальному часі, а також коригувати інтервали зчитування даних за потреби. Така архітектура володіє рядом безперечних переваг: вона забезпечує економію апаратних ресурсів завдяки використанню одного ядра обробки для трьох незалежних зон, дозволяє централізовано керувати налаштуваннями та швидко масштабувати систему шляхом додавання нових вузлів із мінімальною переналаштуванням.

У реалізації мікроагенту було обрано спеціалізовані електронні компоненти, що дозволяють об'єднати різні пристрої в єдину систему управління. Одним із ключових елементів цієї системи стало двоканальне реле, яке показано на рисунку 2.4.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		





Рисунок 2.5 – 1.3-дюймовий модуль OLED дисплею

Цей компактний екран дозволяє у зручному форматі відображати основні параметри середовища: температуру повітря, рівень вологості, концентрацію чадного газу, рівень вібрацій тощо.

Наявність візуального інтерфейсу на дисплеї значно підвищує інформативність системи. Користувач у реальному часі отримує повну картину стану середовища та може швидко реагувати на зміни. Це особливо важливо для оперативного контролю ситуацій, що потребують швидкого втручання - наприклад, виявлення підвищеного рівня чадного газу або різкого падіння температури.

Вибір дисплея має OLED-дисплей, має суттєву перевагу в індикації системи. Він чітко показує дані навіть при різному освітленні, його видно під будь-яким кутом, він споживає мало енергії, що важливо для пристроїв, які працюють довго або на батареях. Крім того, на ньому можна відображати не тільки текст, а й графіки які будуть зручні для роботи системи мікроклімату.

Мікроконтролер у кіберфізичній системі - це її «мозок», але жодна система не працює без «чуттів». Саме датчики виконують роль очей, вух і шкіри агента мікроклімату, дозволяючи системі отримувати інформацію про навколишнє середовище. Вони зчитують температуру, вологість, рівень газів у повітрі, а також фіксують коливання чи вібрації. Усе це перетворюється на цифрові сигнали, з якими може працювати контролер. Від того, наскільки точно й надійно ці датчики передають дані, залежить, як ефективно працюватиме вся система: чи зможе вона

вчасно реагувати на зміни, підтримувати комфортні умови та попереджати ризики у санітарно-медичних закладах.

З огляду на цілі системи це моніторинг параметрів середовища було вирішено, що необхідно відстежувати температуру, вологість, концентрацію чадного газу та рівень вібрацій. Це базові показники, які дозволяють оцінювати загальний стан повітря, своєчасно помічати відхилення та запобігати критичним ситуаціям.

Відштовхуючись від технічних характеристик, для проєкту було обрано три датчики, які добре підходять за своїми можливостями та вартістю:

- DHT11 - для вимірювання температури та вологості;
- MQ-7 - для виявлення концентрації чадного газу (CO);
- KY-002 - для фіксації вібрацій.

DHT11 простий і недорогий датчик, який став популярним у багатьох навчальних і хобі-проєктах. Його часто використовують через легкість підключення: він має цифровий вихід, що дозволяє напряму підключати його до мікроконтролера без потреби в додаткових компонентах. DHT11 вимірює температуру в діапазоні від 0 до +50 °C з точністю  $\pm 2$  °C, а вологість - від 20% до 90% RH з точністю  $\pm 5\%$ . Для завдань, де потрібно контролювати базові параметри мікроклімату, цього цілком достатньо. Крім того, DHT11 споживає дуже мало енергії, що робить його хорошим вибором для проєктів, які працюють на батареях або потребують мінімального енергоспоживання. Його компактний корпус і простий однопровідний інтерфейс спрощують монтаж у різних умовах.

Проте варто врахувати й обмеження: DHT11 має відносно низьку частоту оновлення даних - нові значення надходять лише раз на 1-2 секунди. Також його точність менша, ніж у сучасніших моделей, таких як DHT22 чи SHT31. Але для задач, де потрібно отримувати загальну картину стану мікроклімату, цього цілком вистачає. Простота використання дозволяє розробникам більше часу приділяти створенню логіки системи, а не боротьбі з проблемами інтеграції чи складними алгоритмами.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 46
Зм.	Арк.	№ докум.	Підпис	Дата		

Наступний важливий елемент системи - датчик чадного газу MQ-7. Його головне завдання - виявляти наявність CO у повітрі. Це критично важливо, оскільки чадний газ невидимий, без запаху, але дуже небезпечний для людини. Навіть невеликі концентрації можуть стати загрозою для здоров'я, тому своєчасне виявлення цього газу допомагає уникнути серйозних наслідків. MQ-7 здатний визначати концентрацію CO у межах від 20 до 2000 ppm, що дозволяє фіксувати як низькі рівні, так і небезпечні перевищення [30]. Принцип його роботи базується на зміні електричного опору напівпровідникового матеріалу під дією чадного газу: при наявності CO опір змінюється, а мікроконтролер зчитує цей сигнал і перетворює його на корисні дані. Датчик має тривалий термін служби, стійкий до впливу інших газів, але потребує правильного калібрування та прогріву перед використанням – це гарантує точність показників. MQ-7 надійний інструмент для контролю якості повітря, який допомагає вчасно виявити небезпечні зміни у мікрокліматі та запобігти аварійним ситуаціям у лабораторії.

Для контролю вібрацій у системі обрано датчик KY-002. Його функція фіксація механічних коливань або ударів, які можуть сигналізувати про проблеми в обладнанні, зовнішній вплив або інші аномальні ситуації. Датчик простий у використанні: він має цифровий вихід, який змінює стан при виявленні вібрації – або високий, або низький рівень сигналу. Принцип роботи KY-002 ґрунтується на використанні пружинного контакту: при коливанні або ударі контакт замикається, формуючи сигнал для мікроконтролера [31]. Це дозволяє без складних розрахунків отримати базову інформацію про стан системи. KY-002 компактний, недорогий, простий у підключенні та підходить для базових завдань моніторингу. Його можна використовувати для фіксації небезпечних ситуацій – наприклад, якщо обладнання починає сильно вібрувати, це може свідчити про поломку або перевантаження. Своєчасна фіксація таких сигналів допомагає запобігти аваріям і забезпечити стабільну роботу системи.

Схематичне підключення датчиків до мікроконтролера наведено на рисунку 2.6

					КІРКІ 210108.21.01.55 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		



стабільне середовище для проведення досліджень, тестування та кінцевого роботи експлуатування. Цей набір сенсорів, основа для створення розумної системи, яка допомагає автоматизувати контроль умов і забезпечує комфортну та безпечну роботу в лабораторії.

План-схема приміщення з широким моніторингом (рисунок 2.7):



Рисунок 2.7 – Схема приміщення з широким покриттям

Проте водночас слід враховувати низку обмежень: по-перше, єдиний мікроконтролер стає критичною точкою відмови у разі збою всі три приміщення втрачають зв'язок із центром керування. По-друге, фізична довжина проводів або радіус дії бездротових модулів може обмежувати оптимальне розміщення датчиків, особливо якщо кімнати знаходяться на великій відстані чи мають складну архітектуру з кількома стінами. Такий підхід також потребує стабільного інтернет-з'єднання, адже передача даних у Telegram залежить від доступу до мережі. Водночас можливість інтеграції додаткових алгоритмів обробки даних та розширення функціональності через додавання програмних фільтрів і аналітичних модулів робить цю систему гнучким інструментом для контролю мікроклімату та безпеки віддалених приміщень.

## 2.4 Підключення компонентів

Усе починається з мікроконтролера ESP32 ,який є гнучким та доступним рішенням для реалізації кіберфізичних систем. Для старту або загрузки скетчу потрібно встановити Arduino IDE або PlatformIO, залежно від вимог та потреб проекту. У випадку з Arduino, обов'язково слід додати офіційне посилання на пакет плат ESP32, аби програма знала, як із ними працювати [33]. Після цього можна переходити до встановлення самих плат через Boards Manager і обрання потрібної моделі у меню (наприклад, ESP32 Dev Module). Не менш важливо перевірити COM-порт і впевнитися, що комп'ютер правильно розпізнає плату.

Після налаштування середовища розробки логічним кроком є збирання системи підключення сенсорів, дисплея, реле та інших модулів до ESP32. Тут усе крутиться навколо точності: важливо знати, які піни за що відповідають, які рівні напруги підтримуються і як уникнути помилок, які можуть вивести з ладу мікроконтролер або порушити роботу сенсорів.

Кожен компонент має своє місце й свої умови:

- DHT11, який відповідає за вимірювання температури та вологості, підключається до GPIO 21, а живлення отримує через стандартні лінії 3.3 V і GND. Для стабільної роботи може знадобитися підтягувальний резистор;
- MQ-7, сенсор для виявлення чадного газу, під'єднується до аналогового входу GPIO 36. Оскільки його живлення може вимагати 5 V, а вихідні сигнали іноді перевищують 3.3 V, потрібен стабілізатор і дільник напруги;
- KY-002, вібраційний модуль, використовує GPIO 19. Він дає простий цифровий сигнал при фіксації удару чи коливань;
- OLED-дисплей підключається через інтерфейс I<sup>2</sup>C, використовуючи GPIO 4 і GPIO 5. Для стабільності сигналу підтягувальні резистори на SDA та SCL;

					КІРКІ 210108.21.01.55 ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

– Двоканальне реле керується через GPIO 16 і 17. Виходи підключаються до мережі змінного струму, тому важливо дотримуватись правил електробезпеки [34]. Живлення модуля 5 V через стабілізатор.

Усі ці компоненти мають бути об'єднані правильно. Не менш важливе й правильне живлення це ESP32 має вбудований стабілізатор 3.3 V, але його потужності може не вистачити для всіх підключених елементів, особливо при використанні OLED-дисплеїв чи сенсорів з піковим енергоспоживанням. У таких випадках краще використовувати окремі стабілізатори або зовнішні джерела живлення.

Макетна плата зручний інструмент для початкового монтажу. Але навіть на ній варто дотримуватись порядку: групувати сенсори, розділяти лінії живлення, мінімізувати перехрещення проводів. Це знижує ризик помилок і спрощує подальше тестування. Після монтажу обов'язково перевірити кожен контакт, переконатися у відсутності "плаваючих" ліній та некоректних сигналів.

Усі основні технічні компоненти системи на базі ESP32:

1. ESP32 Dev Module із двоядерним процесором, Wi-Fi/Bluetooth, численними GPIO-пінами для сенсорів, реле та інтерфейсів передачі даних;
2. DHT11 цифровий сенсор для вимірювання температури та вологості повітря, легко інтегрується та працює з внутрішнім підтягувальним резистором;
3. MQ-7 аналоговий сенсор для визначення концентрації чадного газу (CO) у діапазоні 20-2000 ppm, вимагає стабільного живлення й попереднього прогріву;
4. KY-002 датчик вібрації з цифровим виходом, який виявляє механічні удари або коливання, підходить для базової індикації аварій;
5. OLED SH1106 (1.3") дисплей із інтерфейсом I<sup>2</sup>C для локального виводу значень температури, вологості та поточного часу;
6. Двоканальне реле виконавчий модуль для комутації навантажень (вентилятор, освітлення), керується через GPIO ESP32 (інверсна логіка LOW = ON);

					КІРКІ 210108.21.01.55 ПЗ	Арк. 51
Зм.	Арк.	№ докум.	Підпис	Дата		

7. Кнопка скидання тактовий перемикач, підключений до GPIO з підтягуванням, слугує для скидання аварійного режиму або ручного керування LED;

8. Індикаційний LED – світлодіод, що сигналізує про тривожні події, вмикається ESP32 при перевищенні рівня CO або вібрації;

9. Wi-Fi модуль (вбудований у ESP32) – забезпечує зв'язок з Telegram-ботом та Blynk через захищене з'єднання (SSL/TLS);

10. Програмне середовище:

Arduino IDE / PlatformIO – середовище розробки прошивки;

Бібліотеки програмного забезпечення WiFi.h, DHT.h, Adafruit\_SH110X.h, UniversalTelegramBot.h, ArduinoJson.h, WiFiClientSecure.h;

11. Зовнішнє живлення 5V окреме джерело живлення для модулів з високим енергоспоживанням (MQ-7, реле, OLED), забезпечує стабільну роботу ESP32.

Зрештою, усе зводиться до якості підключення й перевірки: навіть найкраща програма не спрацює без надійної апаратної основи. І навпаки правильно підключена система з простим кодом здатна стабільно працювати й виконувати критично важливі функції.

Саме на цьому етапі (рисунок 2.8), перевірка та базове налагодження закладається успіх усього проєкту. Це точка, де ідея починає ставати реальністю. Після завершення цієї частини можна сміливо переходити до написання коду, тестування логіки та налагодження взаємодії з користувачем через Telegram-бота або веб-інтерфейс.

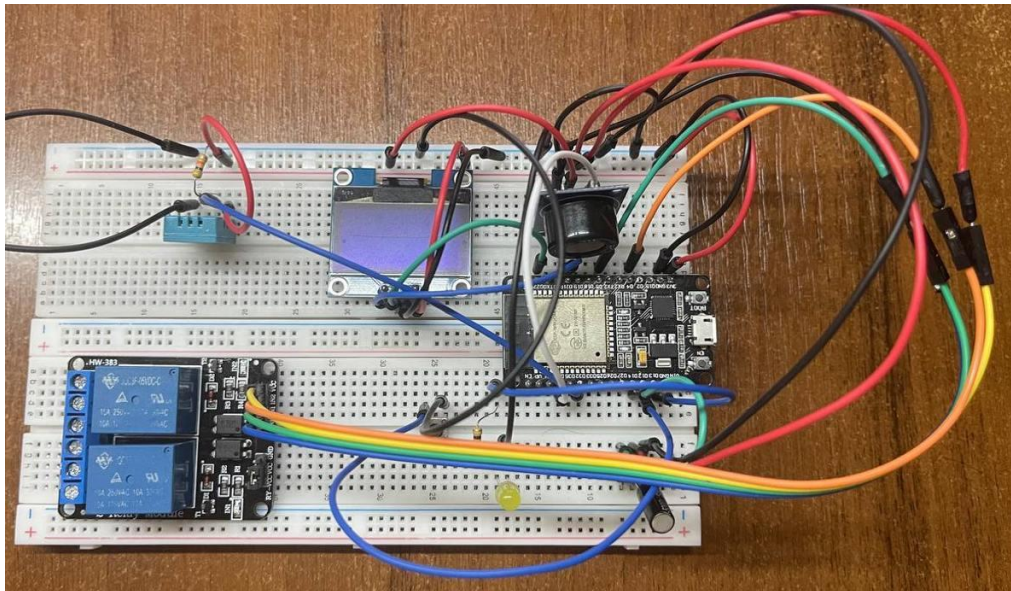


Рисунок 2.8 – Підключення сенсорів до мікроконтролера

## 2.5 Висновки до другого розділу

Системний аналіз ключових технічних аспектів програмного забезпечення, вибору мови програмування, середовища розробки, засобів контролю версій та апаратної платформи дозволив сформувавши цілісне бачення програмної архітектури мікрокліматичної IoT-системи. Отримані результати доводять, що успішна розробка вбудованих систем потребує комплексного підходу, де кожен вибір від коду до сенсора повинен бути обґрунтований не лише локальними характеристиками, а й відповідністю до загальної системної мети, енергетичних обмежень, обчислювального навантаження та сценаріїв використання. Після дослідження було встановлено, що мова програмування відіграє не другорядну, а стратегічну роль у досягненні ефективності системи. Використання C++ забезпечує не лише високу швидкість і контроль над пам'яттю, а й дозволяє підтримувати модульну, повторно використовувану архітектуру з можливістю масштабування. Таким чином, проект починає розвиватися одноразове рішення, а як підтримуваний технічний продукт. З погляду апаратної частини, детальне порівняння мікроконтролерних і одноплатних комп'ютерів дозволило точно визначити зони доцільного застосування кожного типу пристрою.

### 3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ АГЕНТУ МІКРОКЛІМАТУ ЛАБОРАТОРІЇ КІБЕР-ФІЗИЧНОЇ СИСТЕМИ

#### 3.1 Вимоги до програмного забезпечення

Кіберфізичні системи, які відповідають за моніторинг мікроклімату, коли мова заходить про сучасні лабораторії, є досить важливими. Саме вони забезпечують контроль над ключовими параметрами навколишнього середовища температурою, вологістю, тиском, рівнем газів та іншими показниками, що мають безпосередній вплив на перебіг експериментів і безпеку персоналу. У таких умовах навіть незначні коливання можуть зіпсувати результати досліджень або створити потенційні загрози для обладнання та людей. Тому стабільність і точність це не розкіш, а потреба. Автоматизовані системи моніторингу не обмежуються просто фіксацією значень. Вони виконують роль активного учасника процесу: у разі змін вмикаються відповідні пристрої такі як, кондиціонери, осушувачі, вентиляція чи фільтри. Це дає змогу підтримувати постійний мікроклімат без постійного втручання оператора. Важливо й те, що система може миттєво сповістити про критичні відхилення: якщо якийсь параметр виходить за межі допустимого діапазону, оператор одразу отримує сповіщення через повідомлення, сигнал тривоги або навіть автоматичну зупинку процесу. Крім того, автоматично формується база даних із повною історією змін, що дає змогу не лише проводити глибокий аналіз, а й робити прогнози, виявляти закономірності та своєчасно вживати заходів. У центрі цієї інфраструктури мікроконтролери. Це невеликі, але потужні пристрої, які поєднують у собі процесор, пам'ять та порти вводу-виводу, і працюють як «мозок» системи. Вони надзвичайно компактні, споживають мінімум енергії та можуть працювати безперервно у режимі реального часу. Сучасні моделі, наприклад ESP32, додатково обладнані бездротовими модулями Wi-Fi та Bluetooth. Це спрощує підключення до мережі та інтеграцію з іншими пристроями, без зайвих кабелів чи складних налаштувань. Для вимірювання кліматичних умов застосовується

					КІРКІ 210108.21.01.55 ПЗ	Арк. 54
Зм.	Арк.	№ докум.	Підпис	Дата		

широкий набір сенсорів. Температурні та гігрометричні датчики дають змогу точно фіксувати навіть найменші зміни, з точністю до десятих часток. Газові сенсори виявляють присутність потенційно небезпечних речовин, таких як чадний газ (CO) або вуглекислий газ (CO<sub>2</sub>), і сигналізують про небезпеку ще до того, як ситуація стане критичною. Додатково можуть використовуватись акселерометри для контролю вібрацій, фотодіоди для вимірювання рівня освітленості, барометри для моніторингу атмосферного тиску тощо. Усе це дає змогу побудувати дійсно комплексну картину умов у приміщенні. Загалом, система моніторингу створюється як єдина, автоматизована структура, де кожен елемент взаємодіє з іншими. Мікроконтролер збирає дані з усіх підключених сенсорів, аналізує їх відповідно до заданих алгоритмів, приймає рішення і активує виконавчі пристрої. У той же час, він підтримує зв'язок із зовнішніми сервісами, надаючи доступ до інформації у будь-який момент. Такий підхід дає змогу досягти високої точності вимірювань, стабільної роботи та адаптивності до конкретних умов незалежно від того, чи йдеться про лабораторію, дослідницький центр або промислове підприємство.

У системах контролю та моніторингу мікроклімату лабораторій можуть використовуватися різні датчики для забезпечення точного вимірювання та контролю умов середовища. Ось декілька типів датчиків, які переважно є в складі таких систем:

– датчики температури використовуються для вимірювання температури в приміщенні або на об'єкті. Найчастіше застосовують термістори та термопари, оскільки вони дають досить точні результати;

– датчики вологості допомагають визначати рівень вологи в повітрі чи ґрунті, що особливо важливо для лабораторних експериментів. Зазвичай це ємнісні датчики, хоча є моделі, які одночасно вимірюють і температуру, і вологість;

					КІРКІ 210108.21.01.55 ПЗ	Арк. 55
Зм.	Арк.	№ докум.	Підпис	Дата		

– датчики якості повітря потрібні для контролю рівня шкідливих газів, наприклад CO чи CO<sub>2</sub>, у повітрі. Часто працюють на основі хімічних реакцій, які реагують на певні гази.

– датчики освітленості створені на основі фотодіодів чи фоторезисторів, дозволяють визначати, наскільки освітлене приміщення або робоча зона;

– датчики тиску вимірюють рівень тиску газів або повітря, що важливо, наприклад, під час стерилізації або контролю середовища у лабораторіях;

– датчики руху використовуються для безпеки, наприклад, щоб відстежувати, хто заходить або виходить із певної зони. Хоча останнім часом це рішення стало менш популярним.

Водночас з'єднання таких систем із ширшою мережею породжує нові виклики: треба захищати інформацію та турбуватися про кібербезпеку. Щоб лабораторія функціонувала стабільно і безпечно, необхідно правильно управляти цими ризиками. Системи контролю мікроклімату стали важливою частиною сучасних лабораторій. Вони допомагають підтримувати необхідні умови для експериментів, а завдяки своїй точності, гнучкості та можливості інтеграції відкривають нові перспективи для розвитку інновацій у сфері кіберфізичних систем.

### 3.2 Вибір мови програмування

Для написання коду для мікроконтролерів використовують різні мови програмування, кожна з яких має свої сильні сторони й особливості, що залежать від вимог проєкту та вибраного мікроконтролера. Найчастіше застосовують мови C, C++, Python, Assembly, а іноді й JavaScript, якщо мова йде про специфічні платформи [35].

Вибір мови програмування при розробці мікроконтролерних або вбудованих систем безпосередньо залежить від обраної апаратної платформи, рівня абстракції проєкту, вимог до продуктивності та доступної екосистеми

					КІРКІ 210108.21.01.55 ПЗ	Арк. 56
Зм.	Арк.	№ докум.	Підпис	Дата		

інструментів. Для простих мікроконтролерів, таких як Arduino Uno з обмеженими ресурсами, доцільно використовувати мову C, яка дозволяє працювати максимально близько до «заліза» і забезпечує точний контроль над пам'яттю та периферією. Якщо ж мова йде про більш сучасні платформи, зокрема ESP32 або Arduino Nano ESP32, ефективнішим є використання C++ мови, що поєднує низькорівневу ефективність з високорівневою структурованістю, підтримкою об'єктів, інкапсуляції та модульного кодування, що критично важливо для підтримки великих проєктів і повторного використання коду [36]. У випадках, коли важлива швидкість розробки, а не обчислювальна ефективність, наприклад при створенні прототипів або інтеграції з веб-сервісами, доцільним може бути Python або JavaScript, що підтримуються на відповідних платформах як Raspberry Pi або ESP32 (через MicroPython або Espruino). Також враховуються сторонні компоненти ПЗ середовища розробки, наявність драйверів, бібліотек та прикладів, які можуть обмежити або, навпаки, розширити вибір мови. У підсумку, мова програмування обирається не ізольовано, а як частина загальної архітектури проєкту, де вона повинна гармонійно взаємодіяти з апаратною базою, типом задач та розробницьким інструментарієм.

Застосування C++ дає можливість працювати з популярними платформами, такими як Arduino або ESP, де існує велика база знань, прикладів, документації, що допомагає уникати типових помилок і прискорює створення програмного забезпечення для мікроконтролерів. Такий підхід дозволяє будувати стабільні та масштабовані системи, які легко оновлювати та розширювати залежно від завдань. Тому для реалізації проєкту обираємо C++.

Для програмування мікроконтролерів існує кілька популярних середовищ розробки, кожне з яких має свої особливості, переваги та певні обмеження. Найбільш поширеними серед них є Arduino IDE, PlatformIO та Espressif IDF. Кожне з цих середовищ орієнтоване на різні категорії розробників: від новачків, які тільки починають свій шлях у світі IoT, до досвідчених інженерів, що створюють складні комерційні продукти.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 57
Зм.	Арк.	№ докум.	Підпис	Дата		

Незважаючи на наявність сучасних багатofункціональних середовищ розробки, таких як PlatformIO чи Espressif IDF, у ряді випадків доцільно свідомо обирати простіше рішення зокрема Arduino IDE, особливо на етапі створення прототипу або під час індивідуальної розробки. Використання IDE із розширеними можливостями не завжди є виправданим, адже більшість функцій таких середовищ як-от управління залежностями, CI/CD-інтеграція, тестові фреймворки чи модульна побудова проєкту актуальні насамперед для командної роботи або проєктів із глибокою інфраструктурною частиною. У випадках, коли головне завдання полягає у швидкому розгортанні функціонального прототипу або тестуванні роботи мікроконтролера з периферією, надмірна складність інструментів лише ускладнює процес. Arduino IDE, попри свою простоту, забезпечує все необхідне для ефективної роботи з широким спектром плат і модулів, надаючи зручний доступ до бібліотек, прикладів та стабільної компіляції без зайвого конфігурування. Завдяки цьому розробник отримує не лише зниження порогу входу, але й пришвидшення всіх ключових етапів від написання коду до завантаження прошивки, що критично важливо у швидких ітераційних проєктах.

Arduino IDE одне з найпопулярніших середовищ розробки для мікроконтролерів, що здобуло широке визнання завдяки простоті використання, доступності та великій спільноті користувачів [37]. Його інтерфейс інтуїтивно зрозумілий навіть для тих, хто не має попереднього досвіду роботи з мікроконтролерами, а процес програмування зводиться до кількох базових кроків: написання, компіляція та завантаження коду на плату. Arduino IDE також надає величезну кількість готових бібліотек, прикладів та проєктів, що дозволяє швидко освоїти основи розробки та переходити до створення власних прототипів. Важливо, що середовище підтримує роботу з різними платами, включаючи популярні моделі на базі ESP32 та ESP8266, що відкриває широкі можливості для реалізації IoT-рішень.

Проте, Arduino IDE має і певні обмеження. Зокрема, при розробці складних проєктів із багаторівневою структурою коду виникають труднощі з управлінням

					КІРКІ 210108.21.01.55 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

залежностями, налаштуванням компіляції та організацією коду. У таких випадках доцільно використовувати більш гнучкі інструменти, як-от PlatformIO або Espressif IDF.

PlatformIO потужне середовище розробки, що підтримує велику кількість платформ і інтегрується з сучасними редакторами коду, такими як Visual Studio Code. Його можливості значно перевищують функціонал Arduino IDE: PlatformIO дозволяє керувати залежностями, автоматизувати збірку, проводити тестування, інтегрувати системи контролю версій та CI/CD, що є критично важливим для командної роботи та розробки великих комерційних продуктів [38]. Це середовище створене для професіоналів, які потребують контролю над кожним етапом розробки та розгортання.

Espressif IDF (Espressif IoT Development Framework) офіційне середовище розробки для мікроконтролерів ESP, таких як ESP32. Воно відкриває доступ до повного спектру функцій мікроконтролера, включаючи роботу з апаратними ресурсами на низькому рівні, управління енергоспоживанням, реалізацію бездротових інтерфейсів (Wi-Fi, Bluetooth), криптографічних функцій та інших складних задач. Espressif IDF є оптимальним вибором для розробки комерційних продуктів, де критично важлива продуктивність, енергоефективність та глибока оптимізація [39].

У цьому проєкті було прийнято рішення використовувати саме Arduino IDE як основне середовище розробки. Це рішення ґрунтується на кількох ключових причинах. По-перше, Arduino IDE дозволяє швидко налаштувати середовище та розпочати роботу без складної конфігурації, що особливо важливо на етапі створення прототипів. По-друге, наявність великої кількості бібліотек, прикладів та ресурсів значно прискорює процес розробки та дозволяє зосередитися на створенні функціоналу, а не на технічних деталях середовища. По-третє, Arduino IDE спрощує роботу з популярними модулями, датчиками та периферією, що є критично важливим для швидкої розробки IoT-рішень.

Крім того, Arduino IDE дозволяє створювати власні бібліотеки, що сприяє кращій структуризації коду, його оптимізації та повторному використанню в різних проєктах.

### 3.3 Використання GitHub для контролю версій

Контроль версій є критично важливим етапом розробки програмного забезпечення. Він дозволяє зберігати історію змін, забезпечує можливість співпраці між розробниками, допомагає уникати втрати даних і створює гнучку, масштабовану систему управління кодом. Серед доступних інструментів GitHub займає особливе місце бо GitHub, одна з найпопулярніших платформ, яка поєднує функціональність системи контролю версій Git із розширеними можливостями для спільної роботи, автоматизації процесів і управління проєктами.

Окрім базових можливостей контролю версій, GitHub надає інструменти, що дозволяють реалізувати повний життєвий цикл розробки програмного продукту. Наприклад, кожен репозиторій підтримує розділ Wiki, який є інтегрованою платформою для створення й підтримки документації [40]. Це дає змогу командам формувати детальні технічні описи, інструкції користувача, рекомендації щодо використання та інші матеріали. Наявність актуальної, зрозумілої документації також важливий фактор для залучення нових учасників до проєкту, а також для зменшення часу на навчання нових розробників.

Ще одним ключовим компонентом є система Issues, це гнучкий інструмент для управління завданнями, обговорення ідей, повідомлення про баги та відстеження прогресу в їх вирішенні. Кожне завдання можна маркувати тегами, що вказують на його пріоритет, тип (наприклад, баг, нова функція, технічний борг), а також призначати відповідальних. Це дозволяє чітко організувати роботу над проєктом, навіть якщо в ньому беруть участь десятки або сотні розробників, і гарантує, що важливі задачі не залишаться поза увагою.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

GitHub також надає потужні можливості для автоматизації процесів за допомогою GitHub Actions. Цей інструмент дозволяє створювати кастомні пайплайни для виконання різних завдань: автоматичне тестування коду, розгортання додатків, перевірку стилю коду (linting), надсилання повідомлень у Slack або Telegram тощо. Наприклад, можна налаштувати workflow, який автоматично перевіряє проходження тестів для всіх pull requests, а після успішного тестування виконує деплой проєкту на сервер або в хмарне середовище. Це значно підвищує стабільність продукту та зменшує ризик помилок через людський фактор.

Система Code Review через Pull Requests є ще одним важливим елементом GitHub. Вона дозволяє команді проводити перевірку коду перед його інтеграцією до основної гілки, що сприяє виявленню помилок на ранніх етапах, обговоренню архітектурних рішень та дотриманню стандартів кодування. Коментарі можна залишати прямо в рядках коду, що робить обговорення максимально прозорим і зрозумілим, особливо для нових членів команди.

Окрім цього, GitHub підтримує створення Release Notes, які є зручним способом інформування користувачів і розробників про зміни у нових версіях: виправлення багів, додані функції, покращення продуктивності тощо. Це дозволяє легко відстежувати розвиток продукту та підтримувати користувачів у курсі останніх оновлень.

GitHub також надає можливість форкувати (fork) репозиторії створювати власні копії проєкту, модифікувати код і, за потреби, надсилати зміни назад у вигляді pull request. Це особливо важливо для розвитку open-source-спільнот, де співпраця й обмін ідеями між розробниками з усього світу є основою прогресу.

Ще одна корисна функція Secrets Management, що дозволяє безпечно зберігати чутливі дані (API-ключі, токени доступу) для використання у workflow, мінімізуючи ризики витоку конфіденційної інформації. Завдяки цьому GitHub забезпечує високий рівень безпеки автоматизованих процесів.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

Також варто згадати інтеграцію з інструментами на кшталт Dependabot, який автоматично перевіряє залежності проєкту на наявність вразливостей, генерує pull requests для оновлення бібліотек до безпечних версій і допомагає підтримувати продукт захищеним від відомих експлойтів та вразливостей.

У середовищі Інтернету речей існує безліч варіантів побудови каналу зв'язку між сенсорним модулем і кінцевим користувачем або аналітичною системою. Найбільш поширеними є рішення на базі месенджерів, веб-додатків, електронної пошти, SMS- та push-повідомлень, а також інтерактивних чат-ботів. Кожен із цих каналів володіє власними сильними і слабкими сторонами, і часто найкращим підходом стає комбінування двох чи трьох методів залежно від конкретних завдань та вимог до швидкості реакції, обсягу переданих даних і зручності для кінцевого користувача.

Спрощена загальна концепція циклу роботи кіберфізичної системи наведена на рисунку 3.1.

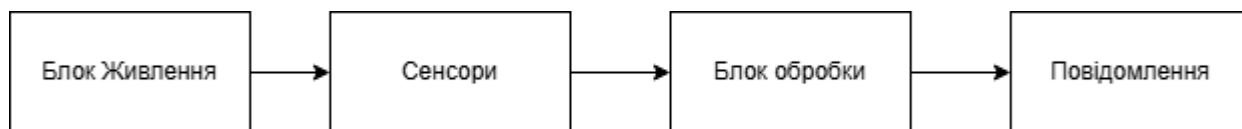


Рисунок 3.1 – Концепція циклу роботи кіберфізичної системи

Месенджери це найшвидший спосіб доставки коротких текстових сповіщень і зображень. У Telegram чи WhatsApp можна миттєво отримати повідомлення про перевищення порогових значень температури або вологості, переглянути графік змін або навіть натиснути кнопку управління пристроєм. Проте інтеграція залежить від сторонніх API, які можуть встановлювати обмеження на кількість запитів, а також потребує додаткових налаштувань для шифрування й автентифікації.

Веб-інтерфейс це повноцінний інформаційний портал із дашбордом, де в режимі реального часу відображаються деталізовані графіки, гістограми та логи подій. Завдяки гнучким фреймворкам на зразок React або Vue можна створити

кастомізовані віджети для відображення даних, інтерактивні фільтри та аналітичні панелі. Серед мінусів необхідність підтримувати серверну інфраструктуру, яка здатна обробляти велике навантаження, регулярно оновлювати бекенд і забезпечувати безпеку з'єднань.

Електронні листи використовують для формалізованої звітності переважно у B2B-середовищі чи для архівування даних. Раз на добу або за подіями система може формувати детальні звіти у вигляді PDF чи CSV-додатків, які зручно зберігати й пересилати керівництву або регуляторам. Однак у порівнянні з миттєвими каналами ці повідомлення надходять із затримкою, і ризик спрацювання спам-фільтрів неможливо ігнорувати.

SMS- та push-сповіщення відносно універсальні методи, оскільки SMS-повідомлення не вимагає наявності Інтернет-з'єднання, а push-нотифікації через Firebase або APNs забезпечують швидкий зворотний зв'язок у мобільних додатках. Головні обмеження: вартість тарифів для SMS і потреба в окремому клієнті для прийому push-сповіщень, зате така комбінація надійно покриває зони із слабким Інтернет-сигналом.

Забезпечення конфіденційності та цілісності даних має не менше значення, ніж самі вимірювання. Перш за все, для захисту HTTP/REST-запитів, MQTT-підписок та WebSocket-сесій використовують TLS/SSL, що дозволяє уникнути перехоплення та підміни пакетів даних. У месенджерах, які підтримують наскрізне шифрування, повідомлення шифруються безпосередньо між користувача--ми, і навіть оператор платформи не може прочитати вміст.

Для контролю доступу до API та сервісів практикують OAuth 2.0 або автентифікацію через JWT-токени. Ці механізми дозволяють гнучко налаштовувати рольові моделі, визначати права адміністратора, оператора або гостьового користувача, а також швидко відкликати чи поновлювати доступ без зміни основного коду. Додатково кожен важливий запит і подія (зміна стану пристрою, спроба підключення, виведення виконавчих команд) фіксуються в системах журналювання наприклад, InfluxDB для часових рядів чи SIEM-рішення

					КІРКІ 210108.21.01.55 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

для кореляції подій і виявлення аномалій. Така централізована обробка логів допомагає своєчасно реагувати на підозрілі активності та швидко відновлювати роботу системи після збоїв.

Сучасні IoT-проекти передбачають використання різних мов на клієнтській і серверній частинах. На мікроконтролерах (ESP32, STM32) зазвичай пишуть на C або C++ із застосуванням бібліотек Mosquitto чи Paho для MQTT та nlohmann/json для роботи з JSON-повідомленнями. Це забезпечує мінімальне затримки й економію ресурсів, критичних для вбудованих систем.

### 3.4 Опис програмної реалізації чат-боту

Успішне функціонування агента мікроклімату у санітарно-медичному закладі неможливе без чіткого визначення системних вимог, оскільки саме вони лежать в основі безперебійної роботи як апаратної, так і програмної частин. По-перше, ключовим аспектом є наявність стабільного підключення до мережі Інтернет. Оскільки система використовує платформу Blynk для передачі телеметрії та отримання команд у режимі реального часу, збої у з'єднанні можуть призводити до втрати даних або неможливості керувати пристроєм дистанційно. Аналогічно, Telegram-бот, який відповідає за оповіщення та керування, вимагає постійного онлайн-доступу до API Telegram: без цього користувач не зможе отримати повідомлення про критичні зміни параметрів або надіслати команду на виправлення ситуації.

Другим важливим пунктом є наявність активного облікового запису в Telegram і належним чином згенерованих токенів та ключів API. Токен від @BotFather для Telegram-бота та автентифікаційний ключ Blynk повинні бути інтегровані в код таким чином, щоб забезпечити достатній рівень безпеки й уникнути несанкціонованого доступу. Сам процес налаштування цих ключів передбачає розуміння базових принципів роботи мережевих API, а також уміння налаштувати середовище розробки для ESP32, Arduino IDE або PlatformIO. Без

					КІРКІ 210108.21.01.55 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

цих елементів коректна робота модуля зв'язку з хмарними сервісами неможлива, а код не зможе підключитися до потрібних ресурсів.

У апаратній складовій ядром системи є мікроконтролер ESP32 разом із трьома ключовими датчиками DHT11, MQ-7 і KY-002, які забезпечують вимірювання температури, вологості, рівня чадного газу та рівня вібрації відповідно. Сенсорні модулі підключені за допомогою Dupont-кабелів, що дає змогу оперативно змінювати конфігурацію при необхідності. Для візуалізації параметрів мікроклімату у реальному часі використовується OLED-дисплей діагоналлю 1.3 дюйми, на якому відображаються значення всіх сенсорів, час останнього оновлення даних та поточний режим роботи пристрою. Додатково передбачена тактова кнопка, що виконує дві ролі: коротке натискання перемикає інформаційні екрани, а тривале утримання ініціює процес калібрування і самодіагностики, перевіряючи зв'язок із хмарним сервером.

Програмна архітектура складається з трьох взаємопов'язаних модулів. На першому етапі ініціалізація налаштовується відповідність контактів мікроконтролера до сенсорів і дисплея, активується I<sup>2</sup>C-шина, запускається послідовний порт і встановлюється Wi-Fi-з'єднання. Далі в основному циклі з фіксованим інтервалом здійснюється опитування сенсорів, валідація отриманих даних, обчислення ковзаючих середніх для згладжування шумів та по черзі вмикаються реле, щоб уникнути пікових навантажень на живлення. На заключному етапі відбуваються підготовка коректних показників для користувача та передача їх у зручному форматі: у випадку негативних відхилень система проводить аналіз і автоматично приймає рішення щодо активації виконавчих пристроїв або сповіщення відповідального персоналу.

### 3.5 Підключення бібліотек

В сучасній розробці вбудованих систем кожен рядок коду це вкладення часу, зусиль та зосередженості програміста. З цієї причини застосування готових

					КІРКІ 210108.21.01.55 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

бібліотек перетворюється не на технологічне рішення, а на стратегічно важливий крок, який допомагає скоротити повторювані задачі, покращити стабільність системи і вивільнити ресурси для творчого процесу замість механічного конструювання власних «велосипедів». Саме бібліотеки є міцним фундаментом програмної архітектури, що дозволяє сфокусуватися на розробці логіки проекту, а не на низькорівневих технічних нюансах. Далі представлено огляд тринадцяти ключових бібліотек, що склали основу для агента мікроклімату у кіберфізичній системі, а також обґрунтування того, як вони гарантують стабільність, безпеку та адаптивність системи.

WiFi.h це місток у глобальну павутину. Ця бібліотека ховає під капотом складнощі налаштування радіомодуля, керування статусами з'єднання, реагування на події типу «розрив/відновлення Wi-Fi» та оптимізації споживання енергії. За її допомогою ESP32 за лічені частки секунди знову підключається до роутера, що дає змогу відправляти дані телеметрії на сервери або приймати інструкції від користувача, не вдаючись до написання низькорівневого коду для роботи із сокетом. Щоб працювати з шиною I<sup>2</sup>C, використовуємо Wire.h. Вона пропонує зручний інтерфейс для передачі даних на OLED-екран або, наприклад, на зовнішній RTC-модуль, надаючи функції для старту та завершення передачі, а також для відправки байтів даних. Таймаути, перевірка підтверджень (ACK) та захист від зависань вже включені у бібліотеку, що дозволяє уникнути непередбачуваних затримок у циклі опитування сенсорів.

BlynkSimpleEsp32.h відмикає двері до хмарної платформи Blynk, де можна на швидку руку згенерувати мобільну панель приладів, інтегрувати графіки історії та push-повідомлення. Ця бібліотека бере на себе всю мороку з TLS-підключеннями, аутентифікацією за токеном та обробкою даних. Розробнику залишається лише викликати Blynk.virtualWrite() для передачі даних.

Щоб отримати інформацію про клімат, використовується DHT.h. Бібліотека ховає під капотом алгоритми стартових імпульсів, побітового зчитування і перевірки контрольних сум, які часто стають каменем спотикання для

					КІРКІ 210108.21.01.55 ПЗ	Арк.
						66
Зм.	Арк.	№ докум.	Підпис	Дата		

початківців. Завдяки цьому час зчитування даних з DHT11 скорочується, а основний цикл програми не блокується.

Взаємодія з користувачем через Telegram-бот здійснюється через бібліотеку UniversalTelegramBot.h. Вона забезпечує HTTP-запити з опитуванням, кешування значень offset для уникнення дублювання повідомлень та аналіз відповідей у форматі JSON, що дає змогу сконцентруватися на виконанні команд. Робоча стабільність бота підтримується навіть при нестабільному інтернет-з'єднанні. Для забезпечення безпеки передачі даних мережею застосовується WiFiClientSecure.h. Ця бібліотека гарантує підтримку TLS/SSL, перевірку сертифікатів та SNI, за допомогою freertos-сумісної реалізації mbedTLS. Достатньо вказати fingerprint або повний сертифікат сервера – і весь обмін даними, наприклад, з Blynk чи Telegram, буде зашифрованим без додаткових зусиль.

Обробку структурованих даних гарантує бібліотека ArduinoJson.h. Застосування StaticJsonDocument забезпечує роботу бібліотеки без необхідності динамічного розподілу пам'яті в купі, що унеможливорює витoki пам'яті на ESP32. З її допомогою просто конструювати пакети даних з інформацією про температуру, вологість, рівень CO та позначкою часу для відправки у хмарний сервіс чи месенджер. Відображення інформації на екрані реалізується через Adafruit\_SH110X.h. Вона абстрагує взаємодію з драйвером SH1106/SH1107, підтримує буферизацію зображень та варіативні розміри шрифтів. Це дає змогу виводити текстову інформацію та навіть примітивні графічні елементи на компактному OLED-дисплеї, роблячи інтерфейс інформативним і доступним.

Для синхронізації часу використовується бібліотека time.h. Вона надає можливість отримати поточний час з використанням NTP та відформатувати його у відповідний формат за допомогою strftime(). Коректний час є надзвичайно важливим для функціонування контролерів та правильного визначення меж гістерезису, особливо в нічний час, коли коливання температури не такі інтенсивні.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

Пряме керування периферійним обладнанням реалізується за допомогою driver/gpio.h. Цей драйвер забезпечує взаємодію з апаратними регістрами, що дозволяє налаштовувати пріоритети переривань, оптимізувати час виконання та забезпечувати негайну реакцію на події, наприклад, при перевищенні рівня CO.

До додаткових, але значущих бібліотек належать SPIFFS (для зберігання журналів), ESPmDNS (для реєстрації пристрою в локальній мережі) та Update.h, яка відповідає за OTA-оновлення можливість завантаження нових версій прошивки без фізичного доступу до пристрою.

### 3.5 Інтеграція з месенджер Telegram

Однією з ключових особливостей запропонованої системи моніторингу мікроклімату є можливість взаємодії з нею через зручний і знайомий користувачам інтерфейс месенджер Telegram. Це рішення дозволяє отримувати актуальну інформацію про стан середовища та реагувати внаслідок критичних змін показників. Для цього було розроблено Telegram-бота, який працює у тісній зв'язці з мікроконтролером ESP32.

Першим кроком у реалізації функціоналу було створення самого бота в Telegram. Для цього використовується офіційний сервіс @BotFather. За допомогою простих команд /start рисунок 3.2 та починаємо роботу з ботом.

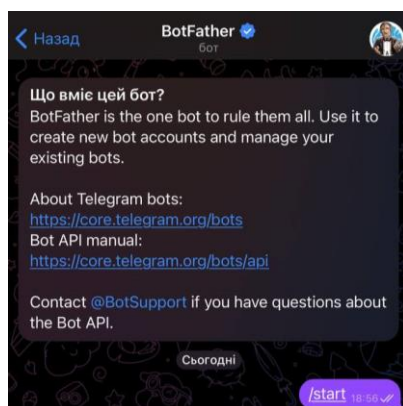


Рис 3.2 – Запуск бота BotFather

					КІРКІ 210108.21.01.55 ПЗ	Арк. 68
Зм.	Арк.	№ докум.	Підпис	Дата		

За допомогою `/newbot` створюється новий бот (рисунок 3.3), якому можна задати ім'я та унікальний ідентифікатор. У відповідь на створення Telegram надає токен спеціальний ключ, за яким ідентифікується бот і з яким надалі працюватиме ESP32.

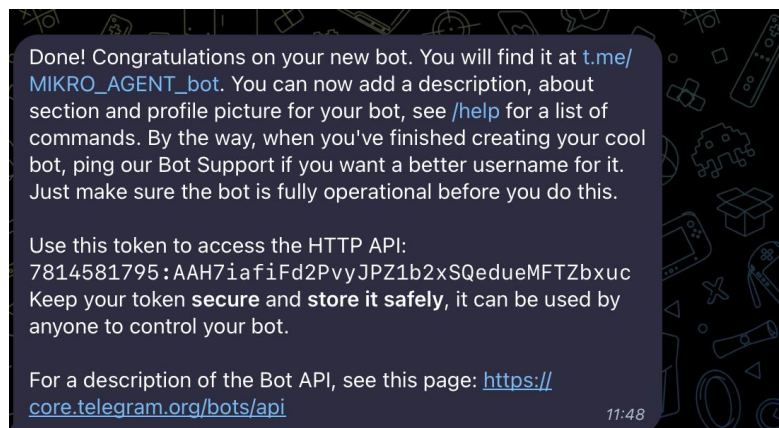


Рисунок 3.3 – Отримання токenu

Щоб налагодити комунікацію між мікроконтролером та Telegram, в прошивку ESP32 інтегрується підтримка Telegram Bot API. Через інтернет-з'єднання (Wi-Fi-модуль ESP32) плата надсилає й приймає HTTP-запити. Таким чином, вона може «слухати» нові повідомлення, обробляти команди користувача та надсилати відповіді. Система працює з API у режимі polling, періодично перевіряючи наявність нових команд від користувачів.

Зображення налаштування мікроконтролеру та отримання даних.

Бот отримав набір базових функцій, які повністю охоплюють потреби системи інформування бот уміє відповідати на запити користувача, надсилаючи поточні показники з датчиків (температура, вологість, концентрація чадного газу, наявність вібрацій)

У кодi мікроконтролера реалізовано обробники команд, які взаємодіють з відповідними функціями: зчитують значення з сенсорів, вмикають або вимикають GPIO-виводи, оновлюють конфігурацію. Кожна команда супроводжується відповідною дією та повідомленням з підтвердженням для користувача. У разі



вноситься у скетч у відповідному рядку коду, що ініціює підключення. Тільки після цього пристрій матиме можливість взаємодіяти з хмарним сервісом, передаючи телеметрію або отримуючи команди з інтерфейсу.

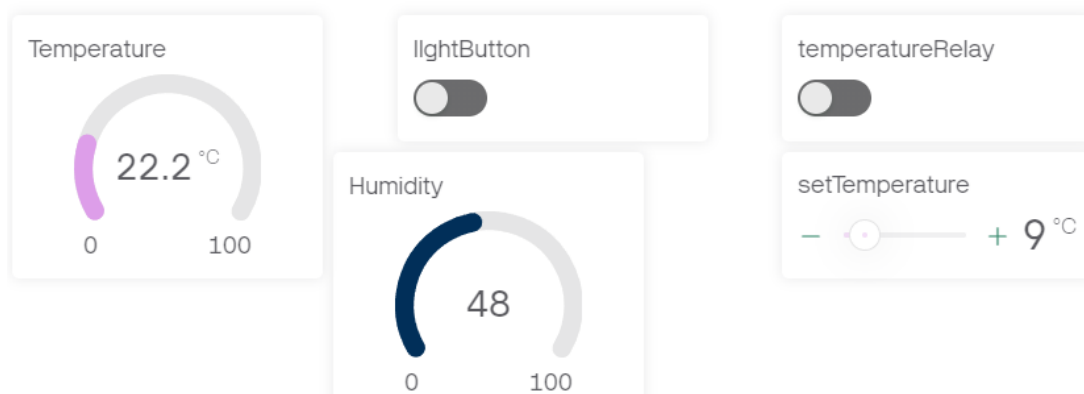


Рисунок 3.5 – Віджети для контролю пристрою

Окрім ключових функцій моніторингу та управління, платформа Blynk також дозволяє віддалене програмування вашого пристрою завдяки OTA (Over-the-Air). Це вкрай практично в умовах реального використання, адже дає змогу оновлювати прошивку мікроконтролера без фізичного з'єднання з комп'ютером. З Blynk OTA можна швидко корегувати код, усувати помилки або додавати нові функції, навіть якщо пристрій розташований у важкодоступному місці або працює безперервно.

Варто підкреслити, що Blynk пропонує декілька тарифних планів із різними функціональними можливостями. Зокрема, безкоштовна версія має обмеження на кількість віджетів до п'яти. Проте, враховуючи поставлену мету розробити агента мікроклімату для кіберфізичної системи, який буде не лише ефективним, але й економічно обґрунтованим використання саме безкоштовного варіанту виявилось повністю виправданим. Наявного ліміту цілком вистачило для реалізації основного функціоналу: моніторингу температури та вологості. Таким чином, проєкт зберігає необхідну гнучкість та функціональність без додаткових витрат.

Декілька технічних переваг у використанні Blynk:

- підтримка широкого спектра пристроїв дозволяє швидко підключити мікроконтролер до хмари без складних налаштувань;
- Vlynk забезпечує миттєвий обмін даними між мікроконтролером і мобільним застосунком, дозволяючи моніторити сенсори, дистанційно керувати пристроєм;
- підтримка віртуальних пінів дає змогу логічно керувати даними без прив'язки до конкретних фізичних пінів, що спрощує структуру коду та розширює можливості керування;
- вбудовані інструменти візуалізації дозволяють відображати дані у вигляді графіків, дисплеїв, кнопок, слайдерів та інші віджетах , це допомагає створити повноцінний інтерфейс керування без окремої розробки веб або мобільного UI.

Загалом, інтеграція з Vlynk спрощує взаємодію користувача з системою, забезпечує технічну зручність у підтримці, модифікації та масштабуванні проекту. Обрана конфігурація демонструє оптимальне поєднання між можливостями платформи та потребами конкретного завдання, що підтверджує раціональність цього рішення в контексті лабораторної кіберфізичної інфраструктури.

### 3.7 Опис програмної частини

Для реалізації проекту обрано мову програмування C++. Такий вибір обумовлений кількома факторами. По-перше, C++ поєднує високу продуктивність із можливостями об'єктно-орієнтованого програмування, що дозволяє створювати структурований код, який легше підтримувати та розширювати. По-друге, C++ забезпечує прямий доступ до апаратних ресурсів, що є важливим для оптимізації роботи мікроконтролера та ефективного управління ресурсами. По-третє, C++ має широку підтримку у спільноті розробників, що спрощує пошук рішень для задач,

а також дозволяє використовувати численні готові бібліотеки й інструменти, що значно прискорює процес розробки.

Програмна логіка системи на базі ESP32 побудована так, щоб забезпечити чітке розділення функцій, стабільну взаємодію між компонентами та гнучкість для подальшого розширення. У центрі уваги ефективне зчитування показників із фізичних сенсорів, аналіз отриманих даних, і передача інформації користувачу в максимально зручній та швидкій формі.

Підключення до Blynk розпочинається з отримання аутентифікаційного токена, який слугує ключем доступу до вашого проекту на платформі Blynk. Його побачити можна на рисунку 3.4.

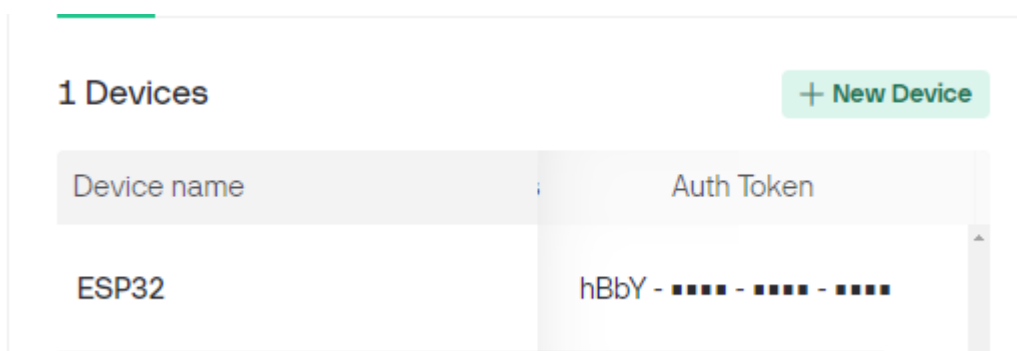


Рисунок 3.4 – Ідентифікатор для підключення пристрою

Усі параметри, від імені мережі Wi-Fi до порогів спрацювання сенсорів, структуровано зберігаються в окремому конфігураційному модулі. Це дозволяє змінювати налаштування централізовано, без потреби вносити правки в десятки функцій. Такий підхід скорочує ризик помилок і прискорює адаптацію коду для нових умов.

Особливістю реалізації є повна відмова від блокуючих функцій типу `delay()`. Замість них використано механізм таймерів на основі `millis()`, завдяки якому всі операції відбуваються паралельно та без затримок. Наприклад, дані з DHT11 оновлюються кожні 10 секунд, тоді як MQ-7 і вібраційний сенсор перевіряються кожні 100 мілісекунд. Це забезпечує як регулярне оновлення, так і миттєву реакцію у разі тривожних подій. За цим принципом побудовано блок-схему, яка

ілюструє алгоритм системи. OLED-дисплей використовується не просто як вивід інформації він діє як локальний інтерфейс. Щойно температура або вологість змінюються, дисплей автоматично оновлюється. Якщо значення ті самі, екран залишається статичним, що знижує навантаження на I<sup>2</sup>C-шину та подовжує строк служби компонента. Разом із температурою відображається й поточний час, який завжди точний завдяки синхронізації з сервером

Таку взаємодію легко масштабувати. Усі повідомлення надсилаються одразу всім активним чатам через шаблонну функцію, що обходить список `chat_id`. Для додавання нових користувачів не потрібно змінювати код вони просто надсилають перше повідомлення боту й автоматично потрапляють у потік повідомлень.

Сама структура коду гнучка: можна легко інтегрувати Blynk або іншу IoT-платформу, додавши нові методи надсилання. Наприклад, функція `notifyBlynk()` могла б дублювати дані на сервер Blynk для відображення у мобільному застосунку або активації автоматичних сценаріїв.

У підсумку, програмна частина проєкту реалізує повний цикл: від зчитування до інформування користувача з урахуванням ефективності, безпеки та зручності. Завдяки чіткій структурі та можливості масштабування ця система підходить не лише для навчальних експериментів, а й для практичного використання в інтелектуальному моніторингу навколишнього середовища в санітарно-медичних закладах.

### 3.8 Керування Blynk

Система дистанційного управління на базі ESP32 і платформи Blynk працює за принципом безперервного зв'язку між фізичним пристроєм і мобільним додатком. Через Wi-Fi мікроконтролер регулярно зчитує дані з сенсора DHT11, який вимірює температуру та вологість, і передає їх у хмару. Дані надходять на сервер Blynk по захищеному каналу, де миттєво обробляються і відображаються

					КІРКІ 210108.21.01.55 ПЗ	Арк. 74
Зм.	Арк.	№ докум.	Підпис	Дата		

на смартфоні користувача. У зворотному напрямку користувач може вносити зміни в параметри через інтерфейс який активує вбудовані функції керування, які відповідають за увімкнення або вимкнення реле, зміну цільової температури або ініціацію автоматичного режиму.

Система складається з трьох ключових компонентів: сенсора, який фіксує навколишні параметри; ESP32, що виконує роль логічного ядра з бездротовим інтерфейсом; та візуального інтерфейсу, реалізованого в застосунку Blynk. Передача даних між рівнями організована через віртуальні піни. Наприклад, температура записується на V0, вологість на V1, перемикач автоматизації V2, цільова температура V3. Зміна стану будь-якого елемента викликає відповідні дії всередині мікроконтролера, що дозволяє управляти фізичними об'єктами без затримок та ручного втручання.

Інтерфейс Blynk налаштовується за кілька хвилин: до робочого екрану додаються візуальні елементи кругові гейджі, перемикачі, слайдери. Один із гейджів показує температуру з точністю до десятих градуса, другий рівень вологості. Перемикач "Turn on Temp" активує режим автоматичного нагріву, а слайдер дозволяє встановити бажану температуру. Усе це пов'язується з відповідними віртуальними пінами, після чого користувач отримує повноцінний пульт керування в кишені.

Усередині робочого циклу ESP32 відбувається постійна перевірка стану віртуальних елементів. Основна логіка зосереджена в методах `Blynk.run()` і `BlynkTimer`, які дозволяють уникнути зависань і забезпечують стабільну роботу навіть у складних умовах. Температура зчитується щосекунди, вологість кожні п'ять секунд. Коли активується автоматичне керування, контролер порівнює поточне значення температури із заданим і приймає рішення про подачу сигналу на реле. Затримки, необхідні для коректної синхронізації, реалізовані без блокувань, що зберігає реактивність системи.

Побудована система демонструє високу стабільність, зручність та готовність до масштабування. Підключення додаткових сенсорів або реле не

					КІРКІ 210108.21.01.55 ПЗ	Арк. 75
Зм.	Арк.	№ докум.	Підпис	Дата		

вимагає зміни архітектури достатньо додати нові піні у Blynk. У перспективі можливе впровадження механізмів аварійного захисту: наприклад, автоматичне відключення живлення при перегріванні. Іншим напрямком розвитку є інтеграція історичних даних через Blynk History API для побудови графіків температури та вологості, що дозволить виявляти тенденції й аномалії. Усе це робить систему не просто засобом керування, а повноцінним інструментом моніторингу й аналізу.

### 3. 9 Висновки до третього розділу

У третьому розділі було послідовно розкрито особливості реалізації програмної частини агента мікроклімату, призначеного для функціонування у складі кіберфізичної системи. Було зосереджено увагу на ключових моментах в реалізації, яка визначає ефективність та надійність роботи: від вимог до стабільності підключення і точності вимірювань до впровадження каналів оперативного зворотного зв'язку. Архітектура рішення побудована з урахуванням потреб автономності, адаптивності та захищеності даних. Використання мови C++ стало виправданим вибором завдяки її поєднанню низькорівневої ефективності з гнучкою структурованістю, що дозволило реалізувати масштабований та підтримуваний код.

Arduino IDE було обрано як практичне середовище для розробки, яке дозволило зосередитись на логіці проєкту без потреби у складному налаштуванні інфраструктури. Завдяки інтеграції GitHub удалося забезпечити прозорість змін, історію розробки, а також полегшити роботу над кодом у командному середовищі. Підключення Telegram-бота та платформи Blynk дало змогу побудувати багаторівневу систему моніторингу, яка поєднує локальний контроль із мобільною взаємодією. Система оперативно сповіщає про зміни, відображає телеметрію в реальному часі та дозволяє віддалено керувати функціями.

Кодова база використовує неблокуючу модель, що дозволяє уникнути зависань і забезпечує рівномірну обробку подій навіть при високому навантаженні. Для візуалізації параметрів обрано OLED-дисплей, який, у

					КІРКІ 210108.21.01.55 ПЗ	Арк. 76
Зм.	Арк.	№ докум.	Підпис	Дата		

комбінації з бібліотекою графічного виводу, виконує роль локального інтерфейсу користувача. Реалізовано також функцію самодіагностики, що дозволяє системі автоматично перевіряти своє з'єднання з хмарними сервісами. Завдяки цьому рішення не обмежується прототипною реалізацією, а здатне працювати у стабільному режимі в умовах нестабільного інтернету або енергопостачання.

					КІРКІ 210108.21.01.55 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		77

## ВИСНОВКИ

У часи, коли світ щодня стикається з викликами війна, енергетична нестабільність, екологічні загрози контроль над середовищем, у якому ми працюємо, лікуємось і досліджуємо, стає не розкішшю, а життєвою необхідністю. Саме тому ідея створення автоматизованої системи моніторингу мікроклімату перестає бути теоретичним завданням і перетворюється на технологічний запити.

Вивчено типові системи, які використовуються для контролю мікроклімату, зокрема ті, що базуються на централізованих сенсорних мережах, та визначено їх основні недоліки висока вартість, складність інтеграції, залежність від стороннього ПЗ.

Ця робота доводить, що сучасні вбудовані системи на прикладі ESP32 у поєднанні з перевіреними мовами програмування, гнучкою архітектурою та відкритими платформами можуть сформувати надійний інструмент контролю навіть у найвимогливіших умовах. Проект не просто реалізував набір функцій. Він створив екосистему: датчики, мобільний контроль, безпека, збереження історії, швидка реакція усе це працює як єдине ціле.

Проведено аналіз нормативної документації та сучасних практик, які вказують на необхідність підтримки стабільної температури (переважно в межах 21–24 °C) та вологості (від 40 до 60%) у лікувальних приміщеннях, операційних та лабораторіях для забезпечення безпеки пацієнтів, коректної роботи обладнання та ефективної роботи персоналу.

Зараз ця система вже готова до інтеграції в реальні об'єкти від лабораторій до лікарень. У майбутньому вона здатна еволюціонувати, підключати нові сенсори для розширення спектру даних моніторингу, співпрацювати з іншими IoT-платформами, навчатися на накопиченій інформації та автоматизувати сценарії повідомлення у Телеграм бот.

За допомогою мікроконтролера ESP32 та середовища Arduino IDE цей проект приклад того, як із простих модулів MQ-7, DHT-11, KY002 і відкритого

					КІРКІ 210108.21.01.55 ПЗ	Арк. 78
Зм.	Арк.	№ докум.	Підпис	Дата		

програмного можна побудувати рішення, що здатне забезпечити стабільність у нестабільному світі.

Розроблено власну систему на базі ESP32 з Wi-Fi та підтримкою Blynk. Система збирає дані з сенсорів, передає їх на сервер, виявляє аномалії та сповіщає персонал. Створено прошивку, хмарне зберігання, а також Telegram-бота для моніторингу та швидкої реакції на зміни мікроклімату.

					КІРКІ 210108.21.01.55 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Pérez A., Hernández L., Gómez R. Diseño de una plataforma IoT para la gestión de parámetros medioambientales en hospitals. *IEEE Latin America Transactions*. 2021. Vol. 19, No 6. P. 987–995.
2. Клименко І. П., Яковенко С. О. Кіберфізичні системи у медицині: концепції та архітектура. *Вісник НТУ «ХПІ»*. 2020. № 43(12). С. 23–31.
3. Шевченко В. Г. Телеграм-боти для моніторингу мікроклімату: практичний підхід. *Вісник Чернівецького нац. ун-ту*. 2022. № 17. С. 99–108.
4. Sánchez F. Sistemas ciberfísicos en la atención médica : retos y oportunidades : tesis maestr. Madrid : Universidad Politécnica de Madrid, 2023. 150 hojas.
5. Pimentel M. A., Armstrong N. J., Si Y., Paris C. A Review of Microclimate Monitoring in Historical Buildings: Challenges and Innovations. *Journal of Cultural Heritage*. 2021. Vol. 47. P. 1–15.
6. Lee E. A., Seshia S. A. Introduction to Embedded Systems: A Cyber-Physical Systems Approach. Cambridge, MA : MIT Press, 2017. 580 p.
7. Song H., Xia F., Zhang X., Wang L., Sun P. Fog Computing for Healthcare: Architecture, Challenges, and Platform. *Future Generation Computer Systems*. 2019. Vol. 93. P. 1012–1023.
8. Гнатуш О. Ю. Моделі прогнозування мікроклімату в реальному часі. Львів : ЛНУ ім. І. Франка, 2023. 356 с.
9. Sethi P., Sarangi S. R. Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering*. 2017. Article ID 9324035. 12 p.
10. Pimentel M. A., Armstrong N. J., Si Y., Paris C. A Review of Microclimate Monitoring in Historical Buildings: Challenges and Innovations. *Journal of Cultural Heritage*. 2021. Vol. 47. P. 1–15.
11. Silva P., Andrade T., Santos F. Monitoramento de microclima em instituições de saúde usando IoT // *Rev. Bras. Eng. Biomédica*. 2019. Vol. 35, No 2. P. 20–29.

					KIPKI 210108.21.01.55 ПЗ	Арк. 80
Зм.	Арк.	№ докум.	Підпис	Дата		

12. Шевченко В. Г. Телеграм-боти для моніторингу мікроклімату: практичний підхід. *Вісник Чернівецького нац. ун-ту*. 2022. No 17. С. 99–108.
13. Müller K., Schmitt R. Cyber-physische Systeme in der Medizintechnik : Herausforderungen und Lösungen. *Z. für Informationssysteme in der Medizin*. 2018
14. Бондаренко О. П. Розумний будинок: автоматизовані системи управління мікрокліматом. Київ : НТУУ «КПІ», 2019. 243 с.
15. Суржиков А. І. Технології Інтернету речей для контролю якості повітря в лікарнях. Харків : ХНУ, 2021. 180 с.
16. IEEE Std 802.15.4-2020. IEEE Standard for Low-Rate Wireless Networks. New York : IEEE, 2020. 420 p.
17. Бондаренко О. П. Розумний будинок: автоматизовані системи управління мікрокліматом. Київ : НТУУ «КПІ», 2019. 243 с.
18. Суржиков А. І. Технології Інтернету речей для контролю якості повітря в лікарнях. Харків : ХНУ, 2021. 180 с
19. Орел І. С. Алгоритми машинного навчання для аналізу мікрокліматичних даних: дис. канд. техн. наук. Одеса : Одеська політехніка, 2021. 162 с.
20. Rajkumar R., Lee I., Sha L., Stankovic J. Cyber-Physical Systems: The Next Computing Revolution. *Proceedings of the IEEE*. 2017. Vol. 100, No 1. P. 366–382.
21. Pinto T. A., Sousa L. G., Santos J. L. Low-Cost Microcontroller-Based System for Real-Time Indoor Microclimate Monitoring. *Sensors (Basel)*. 2020. Vol. 20, No 3. Article 799.
22. Dupont S., Laurent B. Systèmes cyber-physiques et Internet des objets : enjeux et perspectives. *Revue Internationale de Robotique*. 2019. Vol. 15, No 2. P. 45–60.
23. Martin P., Lefèvre V. Surveillance du microclimat en milieu hospitalier : technologies et methods. *Technologies de Santé*. 2021. Vol. 10, No 4. P. 65–79.
24. Durand E. Conception d'un système IoT pour le contrôle du climat intérieur : étude de cas : дис. д-ра. Grenoble : Université Grenoble Alpes, 2022. 220 s.

					КІРКІ 210108.21.01.55 ПЗ	Арк. 81
Зм.	Арк.	№ докум.	Підпис	Дата		

25. García M., López J. Sistemas de monitorización del microclima en edificios sanitarios : una revision. *Revista Iberoamericana de Instrumentación*. 2020. Vol. 25, No 1. P. 112–124

26. Korenivska O. L.; Benedytskyi V. B.; Andreiev O. V.; Medvediev M. G. A System for Monitoring the Microclimate Parameters of Premises Based on the Internet of Things and Edge Devices. *Journal of Edge Computing*. 2023. Vol. 2, No 2. P. 125–147.

27. Mota A.; Serôdio C.; Briga-Sá A.; Valente A. Implementation of an Internet of Things Architecture to Monitor Indoor Air Quality: A Case Study During Sleep Periods. *Sensors*. 2025. Vol. 25, No 6. Art. 1683.

28. Popescu D., Iordache M. Rețele de senzori fără fir pentru monitorizarea microclimatului clinicălor. *Revista Română de Inginerie Medicală*. 2024. Vol. 14, Nr. 2. P. 67–78.

29. Hasan A., Vivek M., Singh R. 5G-Enabled Smart Hospitals: Innovations in Patient Care and Facility Sustainability. *Journal of Medical Systems*. 2024. Vol. 48, No 6. Art. 92.

30. Lee T., Wang J., Chen L. An Analysis of Advanced IoT Technologies and Smart Nursing Systems. *Frontiers in Medicine*. 2024. Vol. 11. Art. 1471527.

31. Berkani M. R. A., Chouchane A., Himeur Y. et al. An Intelligent Edge-Deployable Indoor Air Quality Monitoring and Activity Recognition Approach // *arXiv preprint*. 2023. arXiv:2311.03920.

32. Thomas J., Smith S., McKenna P. Enhancing Healthcare through Sensor-Enabled Digital Twins in Smart Environments: A Comprehensive Analysis. *Sensors*. 2024. Vol. 24, No 9. Art. 2793.

33. Milesight Technology. *Enhancing Indoor Air Quality with WELL-Certified LoRaWAN AM319 9-in-1 Sensor*. Company Blog Post. 2024.

34. Omdena Collaborative. *Predictive Healthcare 2025: Transforming Patient Outcomes with AI-Driven Insights*. Tech Blog. 2025.

					KIPKI 210108.21.01.55 ПЗ	Арк. 82
Зм.	Арк.	№ докум.	Підпис	Дата		

35. Higgs C., Brown L., Patel J. Towards a Healthier Hospital Environment: The Role of Digital Twin Analytics. *IET Digital Healthcare Technology*. 2025. Vol. 2, No 1. P. 15–30.

36. OneSight Solutions Technical Team. *LoRaWAN Indoor Air Quality & Environment Sensors: Technical Overview*. London : OneSight Solutions, 2024. 18 p.

37. Gao Y., Jiang Q., Luo D., Song W. One-Point-Reference-Based Approach for Multi-Indoor Microclimate Prediction Using Machine Learning. *Sustainable Cities and Society*. 2025. Vol. 99. Art. 104030.

38. Zhang K., Wu Y., Chen H., Li X. Federated Learning-Enabled IoT Framework for Indoor Air Quality Monitoring and Ventilation Control. *Sustainable Built Environment*. 2025. Vol. 14, No 3. P. 301–317.

39. Chandra S., Edwards D., Huang Y. Innovations in Air Quality Monitoring: Sensors, IoT and Future Research Directions. *Sensors*. 2025. Vol. 25, No 7. Art. 2070

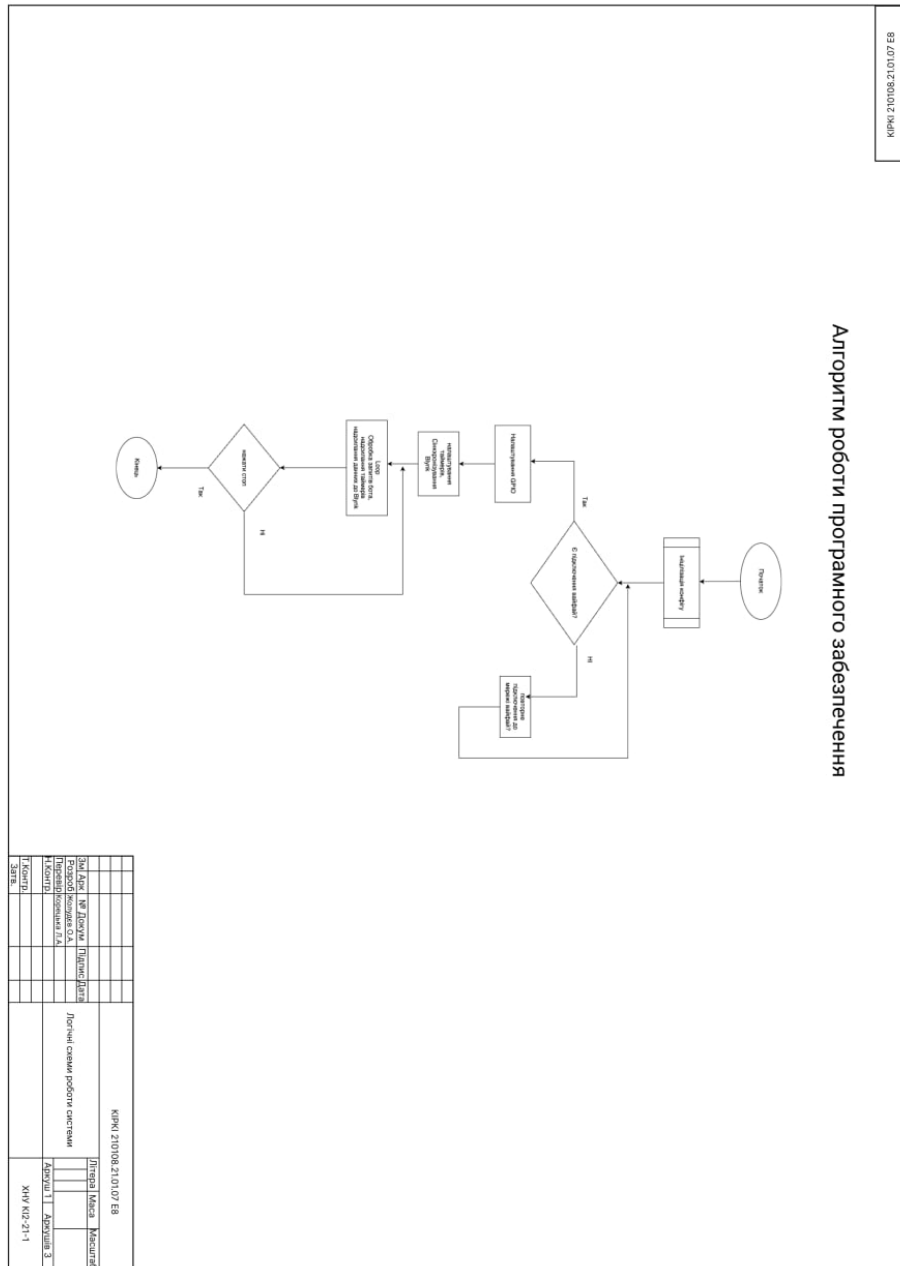
40. Parkavi A., Sowmya B. J., Alex S. A., Supreeth S., Shruthi G. Air Quality and Dust Level Monitoring Systems in Hospitals Using IoT. *Discover Internet of Things*. 2025. Vol. 5. Art. 23.

					КІРКІ 210108.21.01.55 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		83

# ДОДАТОК А

(обов'язковий)

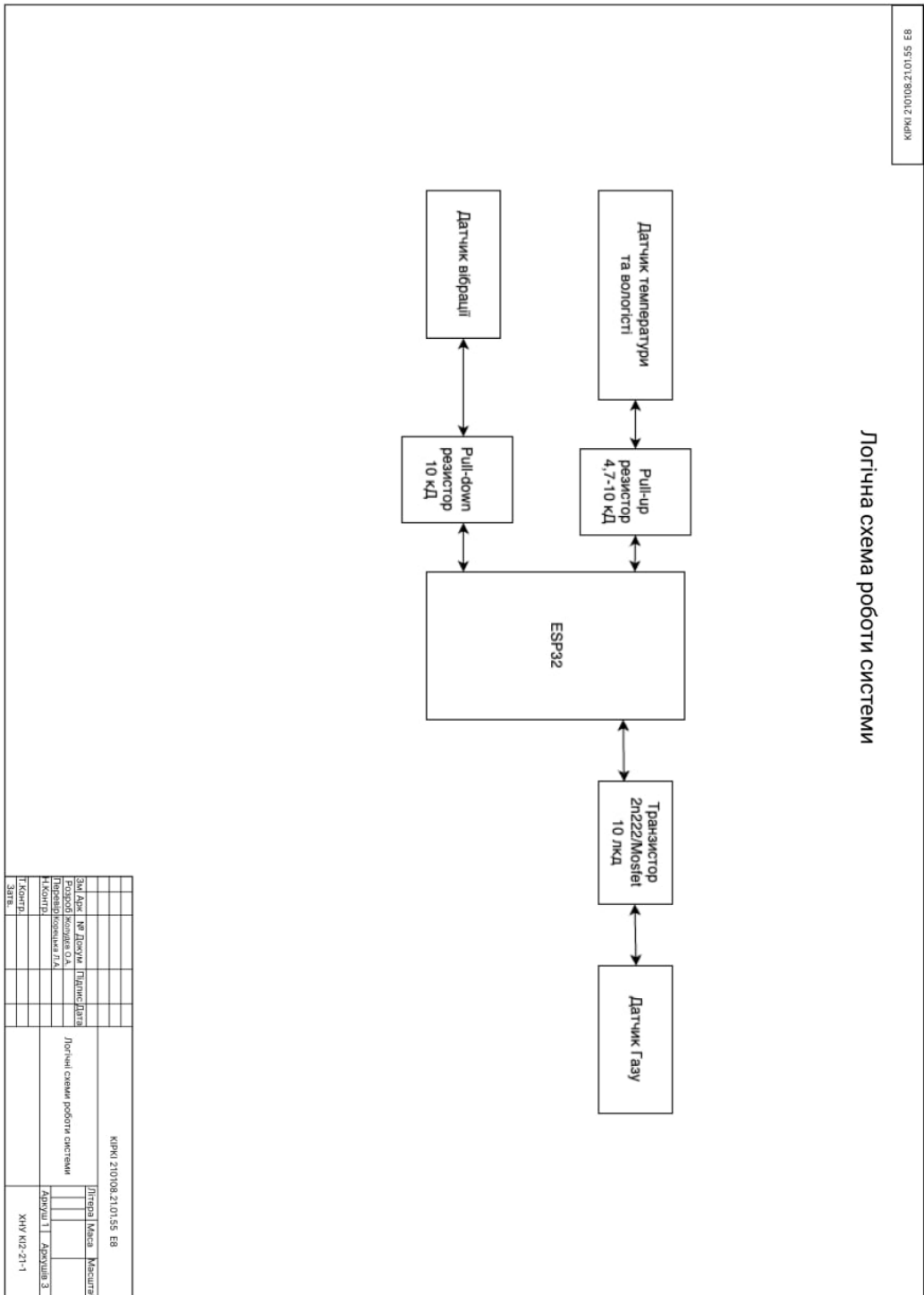
## КОПІЯ КРЕСЛЕННЯ «АЛГОРИТМ РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»



# ДОДАТОК Б

(обов'язковий)

## КОПІЯ КРЕСЛЕННЯ «ЛОГІЧНА СХЕМА РОБОТИ СИСТЕМИ»



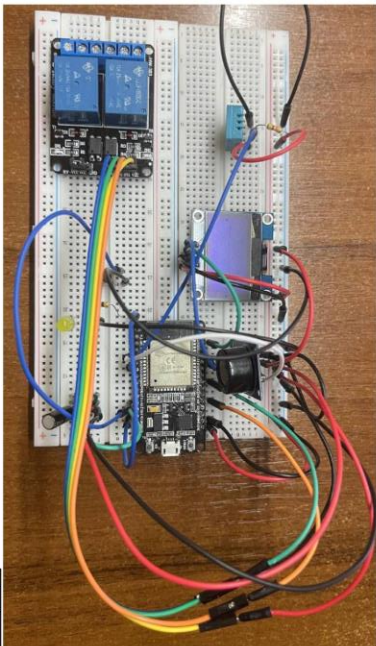
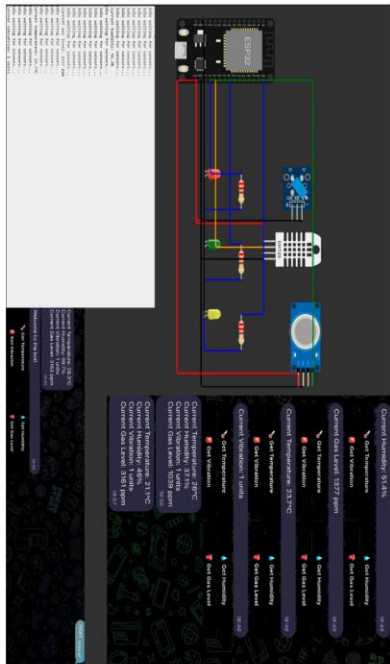
# ДОДАТОК В

(обов'язковий)

## КОПІЯ КРЕСЛЕННЯ «АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ»

КІРЖИ.2101008.21.01.55. E8

### Апаратне забезпечення проекту



КІРЖИ.2101008.21.01.55. E8			
ЗМІ	Кіде	№ Докум.	Підпис/Дата
Розроб	Жукова О.А.		
Перевір	Коваленко Г.А.		
Контр.			
Затв.			
Апаратне забезпечення проекту			Ітерація
Кіріш			Місяць
Закінч.			Масштаб
Затв.			Архив
Затв.			Затв.

# ДОДАТОК С

## ПРОГРАМНИЙ КОД ДЛЯ ІНТЕГРАЦІЇ З МІКРОКОНТРОЛЕРОМ

(обов'язковий)

```
// ===== config.h =====  
  
#ifndef CONFIG_H  
#define CONFIG_H  
  
#define BLYNK_TEMPLATE_ID "TMPL4-WKIXdY3"  
#define BLYNK_TEMPLATE_NAME "ESP32"  
#define BLYNK_AUTH_TOKEN "hBbY1GVX2qaLOdSYnqLUTzE5SUstEgcI"  
#define TELEGRAM_BOT_TOKEN  
"7814581795:AAH7iafiFd2PvyJPZ1b2xSQedueMFTZbxuc"  
  
#define VIBRATION_SENSOR_PIN GPIO_NUM_14  
#define LED_PIN GPIO_NUM_13  
#define RESET_BUTTON_PIN GPIO_NUM_12  
#define MQ7_SENSOR_PIN GPIO_NUM_34  
  
#define CO_THRESHOLD 3000  
#define DEBOUNCE_DELAY 50  
#define DHTPIN GPIO_NUM_32  
#define DHTTYPE DHT11  
  
#define GPIO_INPUT_PIN_SEL \  
((1ULL << VIBRATION_SENSOR_PIN) | (1ULL << RESET_BUTTON_PIN))  
#define GPIO_OUTPUT_PIN_SEL ((1ULL << LED_PIN))  
  
#define NTP_SERVER "pool.ntp.org"  
  
struct ConnectionSettings {  
    const char* ssid = "Wokwi-GUEST"; // Default Wokwi AP
```

```

const char* password = "";
const char* blynkToken = BLYNK_AUTH_TOKEN;
};

#endif // CONFIG_H

// main.ino
#include <Adafruit_SH110X.h>
#include <ArduinoJson.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <UniversalTelegramBot.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <Wire.h>
#include <driver/gpio.h>
#include <time.h>

#include "config.h"

ConnectionSettings connSettings;
DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;
WiFiClientSecure secured_client;
UniversalTelegramBot bot(TELEGRAM_BOT_TOKEN, secured_client);
Adafruit_SH1106G display(128, 64, &Wire);

bool ledState = false;

String keyboardJson =
  "["
  "[{"text\":" 🌡 Temperature\","callback_data\":" /temperature"}],"
  [{"text\":" 🌧 Humidity\","callback_data\":" /humidity"}],"
  [{"text\":" 📶 CO Level\","callback_data\":" /gas"}],"

```

```

    [{"text\\":\\" 📢 Vibration\\",\\"callback_data\\":\\"/vibration\\"}],
    [{"text\\":\\" 💡 Turn LED OFF\\",\\"callback_data\\":\\"/ledoff\\"}]]"
  ]";

void checkConnection() {
  if (WiFi.status() != WL_CONNECTED) {
    Serial.println("WiFi lost, reconnecting...");
    WiFi.disconnect();
    WiFi.begin(connSettings.ssid, connSettings.password);
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("\nWiFi reconnected. IP: " + WiFi.localIP().toString());
  }

  if (!Blynk.connected()) {
    Serial.println("Blynk lost, reconnecting...");
    Blynk.connect();
  }
}

void setup() {
  Serial.begin(115200);
  Serial.println("\n=== ESP32 Starting ===");

  setupWiFiAndBlynk();

  gpio_config_t io_conf = {};

  io_conf.intr_type = GPIO_INTR_DISABLE;
  io_conf.mode = GPIO_MODE_OUTPUT;
  io_conf.pin_bit_mask = GPIO_OUTPUT_PIN_SEL;
  io_conf.pull_up_en = GPIO_PULLUP_DISABLE;

```

```

io_conf.pull_down_en = GPIO_PULLDOWN_DISABLE;
gpio_config(&io_conf);
io_conf.mode = GPIO_MODE_INPUT;
io_conf.pin_bit_mask = GPIO_INPUT_PIN_SEL;
io_conf.pull_up_en = GPIO_PULLUP_ENABLE;
io_conf.pull_down_en = GPIO_PULLDOWN_DISABLE;
gpio_config(&io_conf);

timer.setInterval(10000L, sendSensorData);
timer.setInterval(1000L, handleNewMessages);
timer.setInterval(500L, checkSensors);
timer.setInterval(30000L, checkConnection);
Blynk.syncAll();
// display.begin(0x3C, true);
// display.display(); delay(500);
// display.clearDisplay();

configTime(3 * 3600, 0, NTP_SERVER);

Serial.println("Setup complete\n");
}
void setupWiFiAndBlynk() {
  Serial.print("Connecting to WiFi SSID: ");
  Serial.println(connSettings.ssid);
  WiFi.begin(connSettings.ssid, connSettings.password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi connected. IP: " + WiFi.localIP().toString());

  Serial.println("Connecting to Blynk...");
  Blynk.begin(connSettings.blynkToken, connSettings.ssid,
             connSettings.password);

```

```

secured_client.setInsecure();
Serial.println("Blynk ready\n");
}

void sendSensorData() {
  Serial.println("[sendSensorData] Reading DHT");
  static float lastTemp = NAN, lastHum = NAN;
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  if (!isnan(temperature) && !isnan(humidity)) {
    if (temperature != lastTemp || humidity != lastHum) {
      Serial.printf("Temperature: %.1f C, Humidity: %.1f %%\n", temperature,
                    humidity);
      Blynk.virtualWrite(V0, temperature);
      Blynk.virtualWrite(V1, humidity);
      lastTemp = temperature;
      lastHum = humidity;
    }
  } else {
    Serial.println("Failed to read from DHT sensor!");
  }

  struct tm timeinfo;
  if (getLocalTime(&timeinfo)) {
    char buf[16];
    strftime(buf, sizeof(buf), "%H:%M", &timeinfo);
    display.clearDisplay();
    display.setTextSize(3);
    display.setTextColor(SH110X_WHITE);
    display.setCursor(0, 0);
    display.printf("%.1f C", temperature);
    display.setTextSize(2);
    display.setCursor(0, 48);

```

```

    display.println(buf);
    display.display();
}
}

void checkSensors() {
    int vibration = gpio_get_level(VIBRATION_SENSOR_PIN);
    int coLevel = analogRead(MQ7_SENSOR_PIN);
    bool buttonPressed = (gpio_get_level(RESET_BUTTON_PIN) == 0);

    Serial.printf("[checkSensors] Vib=%d, CO=%d, Button=%d\n", vibration, coLevel,
        buttonPressed);

    if (buttonPressed && ledState) {
        ledState = false;
        gpio_set_level(LED_PIN, 0);
        notifyTelegram("Info: LED turned OFF manually.");
    } else if ((vibration == 1 || coLevel > CO_THRESHOLD) && !ledState) {
        ledState = true;
        gpio_set_level(LED_PIN, 1);
        notifyTelegram("Alert: LED turned ON due to sensor activation.");
    }
}

void handleNewMessages() {
    int count = bot.getUpdates(bot.last_message_received + 1);
    if (count > 0) Serial.printf("[handleNewMessages] New: %d\n", count);
    for (int i = 0; i < count; i++) {
        String chat_id = bot.messages[i].chat_id;
        String txt = bot.messages[i].text;
        Serial.printf("Message from %s: %s\n", chat_id.c_str(), txt.c_str());
        handleCommands(chat_id, txt);
    }
}
}

```

```

void handleCommands(String chat_id, String text) {
  if (text == "/start") {
    bot.sendMessageWithInlineKeyboard(chat_id, "Choose a command:", "",
                                      keyboardJson);
    return;
  }

  if (text == "/temperature" || text == "/humidity") {
    executeCommand(chat_id, text);
  } else if (text == "/ledoff") {
    ledState = false;
    gpio_set_level(LED_PIN, 0);
    bot.sendMessage(chat_id, "LED is turned OFF", "");
  } else if (text == "/gas") {
    int coLevel = analogRead(MQ7_SENSOR_PIN);
    bot.sendMessage(chat_id, "Current CO level: " + String(coLevel), "");
  } else if (text == "/vibration") {
    int vibration = gpio_get_level(VIBRATION_SENSOR_PIN);
    String resp = (vibration == 1) ? "Vibration detected! 📢" : "No vibration.";
    bot.sendMessage(chat_id, resp, "");
  }

  bot.sendMessageWithInlineKeyboard(chat_id, "Select another command:", "",
                                    keyboardJson);
}

void executeCommand(String chat_id, String cmd) {
  if (cmd == "/temperature")
    sendMessage(chat_id, dht.readTemperature(), "Temperature");
  if (cmd == "/humidity") sendMessage(chat_id, dht.readHumidity(), "Humidity");
}

void sendMessage(String chat_id, float value, String type) {

```

```

String resp = !isnan(value) ? "Current " + type + " is: " + String(value) +
                (type == "Temperature" ? " °C" : " %")
                : "Failed to read " + type + ".";
bot.sendMessage(chat_id, resp, "");
}

void notifyTelegram(String msg) {
  int count = bot.getUpdates(bot.last_message_received + 1);
  for (int i = 0; i < count; i++) {
    String chat_id = bot.messages[i].chat_id;
    bot.sendMessage(chat_id, msg, "");
  }
}

void loop() {
  Blynk.run();
  timer.run();
  delay(10);
}

```

## 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 Жолудев

**Співавтор:**

**Назва:** Жолудев\_Кіберфізична система моніторингу параметрів мікроклімату у медичному закладі

**Експерт:**

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 1.5%

**Коефіцієнт подібності 2:** 0.6%

**Мікропробіли:** 0

**Заміна букв:** 5

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2025-06-19 08:56:17.0

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 укріття 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 процедур. Таким чином робота не приймається.

Обґрунтування:

2025-06-19

Дата

Доцент Андрій Нічепорук

експерт

## Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 10%

ID: 246860 Title: БКР Кіберфізична система моніторингу параметрів мікроклімату у медичному закладі Added in a DB: 2025-06-19 Authors: Олександр Жолудєв Heads: Людмила КОРЕЦЬКА Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	107329	1594	2531 (2%)	35 (2%)

### Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Жолудев Олександр Андрійович

Тема: Кіберфізична система моніторингу параметрів мікроклімату у медичному закладі

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 79

1. Метою роботи є розроблення прототипу кіберфізичної системи (КФС), що в режимі реального часу відстежує температуру, вологість, концентрацію CO та вібраційні навантаження приміщень медичного закладу, інформує персонал через Telegram-бот і забезпечує дистанційне керування за допомогою Blynk. У роботі проведено аналіз нормативної бази, вибрано сенсорну апаратуру (DHT11, MQ-7, KY-002) та мікроконтролер ESP32, розроблено алгоритми збору й обробки даних у FreeRTOS.

2. Виконано всі позиції виданого завдання: сформульовано технічні вимоги, спроектовано апаратну та програмну архітектуру, створено алгоритмічні, принципів та структурні схеми, реалізовано демонстраційну збірку й оформлено звітні матеріали. Відхилень від поставленої мети не виявлено.

3. Розділ 1 демонструє глибокий системний огляд сучасних КФС, аналізує ризики воєнного часу, подає детальне обґрунтування вибору показників моніторингу й нормативних меж.

Розділ 2 подає порівняльний аналіз платформ Arduino-Uno/ESP32/Raspberry Pi, вміщує таблиці технічних параметрів і ґрунтовно аргументує вибір ESP32.

Розділ 3 містить повний стек програмної реалізації: конфігураційні файли, блок-схеми FreeRTOS-задач, приклади запитів до Telegram-API. Автор коректно застосував бібліотеки UniversalTelegramBot, Arduino-JSON, що відповідає сучасним трендам IoT-розробок. Таким чином, дипломник активно спирався на останні розробки у галузі вбудованих систем та безпечного передавання даних.

4. Позитивні сторони роботи: висока практична цінність проекту для медзакладів у кризових умовах; модульна архітектура, що легко масштабувати; комплексне поєднання сенсорів, мобільного сповіщення; якісно оформлені графічні матеріали та зрозумілі інструкції з розгортання.

5. Негативні сторони роботи: обмежена тривалість стендових випробувань; відсутність розширеного аналізу кіберзахисту при можливих атаках; не подано оцінки енергоефективності в умовах автономного живлення.

6. Записка структурована за вимогами ХНУ, містить 3 ілюстрації, список літератури з 40 позицій; 2 таблиці, рисунки пронумеровані коректно. Оформлення відповідає діючим стандартам.

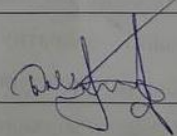
7. Кваліфікаційна робота виконана на високому науково-технічному рівні, демонструє вміння автора самостійно формувати інженерні задачі, застосовувати сучасні апаратно-програмні засоби та критично оцінювати отримані результати. Розроблений прототип може слугувати основою для подальших досліджень у галузі медичних IoT-систем.

8. Інші зауваження: Немає

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Шокаршин  
Ремис Анатолійович, к.т.н., доцент кафедри АСУТ

"19" 06 2025 р.

 (підпис)

Завідувачу кафедри КПС  
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Олександра ЖОЛУДСЬКА  
ІІІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-1

#### 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 (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) 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 (Strike-Plagiarism 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.

19.06 2025 року

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 моніторингу параметрів мікроклімату у медичному закладі

Автор: Олександр Жолудев  
Спеціальність: 123- Комп'ютерна інженерія  
Освітня програма: освітньо-професійна  
Науковий керівник: Людмила Корецька, 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 розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі українськими скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 1.5% і адресується до 401 першоджерела; та системою Anti-Plagiarism складає 1.0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

Людмила КОРЕЦЬКА

Андрій НІЧЕПОРУК

Ольга ПАВЛОВА