

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

КВАЛІФІКАЦІЙНА РОБОТА
на тему: «Веб-додаток для моніторингу здоров'я з
використанням IoT технологій»

на здобуття освітнього ступеня вищої освіти бакалавр
зі спеціальності 122 Комп'ютерні науки
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні науки
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

(підпис) Віталій ТИМЧЕНКО
(Ім'я, ПРІЗВИЩЕ здобувача)

Виконав:
здобувач вищої освіти
група КНД-41

Віталій ТИМЧЕНКО

Керівник:
науковий ступінь,
вчене звання

Юрій КАТКОВ
Д.Т.Н., доцент

Рецензент:
науковий ступінь,
вчене звання

(Ім'я, ПРІЗВИЩЕ)

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Комп'ютерних наук

Ступінь вищої освіти бакалавр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри Комп'ютерних наук

_____ Віктор ВИШНІВСЬКИЙ

« _____ » _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

_____ Тимченко Віталію Володимировичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Веб-додаток для моніторингу здоров'я з використанням IoT технологій.

керівник кваліфікаційної роботи Юрій КАТКОВ д.т.н., професор,

(Ім'я, ПРИЗВИЩЕ науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. № 56

2. Строк подання кваліфікаційної роботи «30» травня 2025р.

3. Вихідні дані до кваліфікаційної роботи:

- Науково-технічна література з питань, пов'язаних з аналізом можливостей використання IoT у сфері охорони здоров'я.
- Огляд можливостей використання IoT у сфері охорони здоров'я;
- Огляд апаратних рішень на базі ESP32 та сенсорів;
- Огляд веб-інтерфейсу для відображення медичних даних;

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

4.1 Аналіз можливостей використання IoT-технологій у системах моніторингу здоров'я;

4.2 Дослідження застосування IoT-технологій у системах моніторингу здоров'я.

4.3 Представлення результатів і рекомендацій щодо впровадження IoT у медичний моніторинг.

5. Перелік графічного матеріалу: *презентація*

1) Тема дипломної роботи.

2) Мета роботи. Об'єкт дослідження. Предмет дослідження. Постановка завдання дослідження.

3) Актуальність роботи.

4) Огляд існуючих інструментів та методів.

5) Огляд основних можливостей.

6) Демонстрація роботи розроблених інструментів .

7) Оцінка переваг та недоліків.

8) Висновки та перспективи розвитку проєкту.

9) Апробація результатів дослідження.

Дата видачі завдання 25 лютого 02.2025 р

КАЛЕНДАРНИЙ ПЛАН

№ п	Назва етапів бакалаврської Роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури.....	25.02-26.02.25	виконане
2	Дослідження поставленої задачі.....	26.02-20.03.25	виконане
3	Проведення аналізу сучасних IoT-технологій у медичній сфері.....	25.02- 26.02.25	виконане
4	Визначення мети, завдань та методів дослідження.....	25.02- 20.03.25	виконане
5	Дослідження апаратної частини (ESP32, сенсори) та можливостей їх використання.....	21.03-27.03.25	виконане
6	Розробка та тестування веб-системи для моніторингу фізіологічних показників.....	28.03-30.03.25	виконане
7	Аналіз результатів, перевірка працездатності та стабільності системи.....	01.04-07.04.25	виконане
8	Підготовка тексту розрахунково- пояснювальної записки та презентації.....	08.04-17.04.25	виконане
9	Завершення оформлення всієї роботи.....	18.04-24.04.25	виконане
10	Розробка обов'язкових демонстраційних матеріалів.....	25.04-28.04.25	виконане
11	Доповідь та презентація.....	29.04-20.05.25	виконане
12	Попередній захист роботи	До 30.05	виконане

Здобувач вищої освіти

(підпис)

Віталій ТИМЧЕНКО

(Ім'я, ПРІЗВИЩЕ)

Керівник



кваліфікаційної роботи

(підпис)

Юрій КАТКОВ

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина бакалаврської роботи 70 с., 45 рис., 5 табл., 48 джерел

Мета роботи: Підвищення ефективності контролю за станом здоров'я шляхом розробки веб-додатку, що використовує IoT-пристрої та технологію Webhook..

Об'єкт дослідження: Процес віддаленого моніторингу стану здоров'я людини..

Предмет дослідження: Веб-додаток, що використовує IoT-пристрої та технологію Webhook.

Короткий зміст роботи:

Наукове завдання: Оцінка ефективності розробленого веб-додатку, що використовує IoT-пристрої та технологію Webhook..

Методи дослідження : Методи дослідження заснований на методах системного підходу, а саме: теоретичні методи (метод порівняльного аналізу, метод системного аналізу); аналітичні методи (метод метрики ефективності, метод логування) та практичні методи (метод моделювання, метод проєктування програмного забезпечення, експериментальний метод (практична реалізація), метод тестування та налагодження).

Актуальність теми: Створення зручних та ефективних систем віддаленого моніторингу, які дозволяють не тільки збирати дані з датчиків фізіологічного стану, але й своєчасно реагувати на критичні зміни є сучасним. Особливої важливості набуває застосування технологій сповіщення, таких як Webhook, що забезпечують миттєву інтеграцію з веб-сервісами та платформами. Така інтеграція дозволяє оперативно передавати дані у веб-браузер користувача, медичний портал або сервер аналітики, що значно підвищує ефективність роботи IoT-системи, що є актуальним.

Галузь використання: Медицина і охорона здоров'я. Фітнес і здоровий спосіб життя. Наукові дослідження. Освіта і навчання. Смарт-міста.

Ключові слова: веб-браузер, Webhook, IoT-пристрої.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	10
ВСТУП.....	11
1 АНАЛІЗ МОЖЛИВОСТЕЙ ВИКОРИСТАННЯ ІoT-ТЕХНОЛОГІЙ У СИСТЕМАХ МОНІТОРИНГУ ЗДОРОВ'Я	14
1.1 Аналіз шляхів впровадження ІoT у галузі медицини та побутового моніторингу	14
1.2 Аналіз сучасних методів дистанційного контролю фізіологічних показників	21
1.3 Аналіз прикладів застосування ІoT у системах контролю стану здоров'я.....	26
1.4 Аналіз тенденції розвитку веб-рішень для віддаленого моніторингу стану здоров'я.....	30
2 ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ІoT-ТЕХНОЛОГІЙ У СИСТЕМАХ МОНІТОРИНГУ ЗДОРОВ'Я.....	34
2.1 Дослідження можливостей контролера ESP32 для медичних ІoT-проектів.....	34
2.2 Дослідження можливостей застосування пристроїв збору даних: датчики, модулі зв'язку та живлення	38
2.3 Побудова схеми ІoT-пристрою для зчитування показників	41
2.4 Взаємодія пристрою з мережею через Wi-Fi або Bluetooth	47
3 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО РЕАЛІЗАЦІЇ ВЕБ-СИСТЕМИ ДЛЯ МОНІТОРИНГУ ЗДОРОВ'Я	52
3.1 Рекомендації щодо побудови алгоритма обробки даних з ІoT-пристрою у веб-середовищі	52
3.2 Рекомендації щодо налаштування середовища для програмування	

мікроконтролера ESP32і	54
3.3 Рекомендації щодо підключення зовнішніх бібліотек та написання програмного коду	58
3.4 Рекомендації щодо розгортання та тестування прототипу платформи моніторингу ментації щодо покращення безпеки IoT-систем	65
ВИСНОВКИ.....	81
ПЕРЕЛІК ПОСИЛАНЬ.....	83
ДОДАТОК А. Програмний код в середовищі Arduino IDE, який реалізує процес зчитування показників та їх відправку до ThingSpeak.....	89
ДОДАТОК Б. Кодовий лістинг з параметрами підключення до платформи ThingSpeak через бездротову мережу Wi-Fi.....	91
ДОДАТОК В. - Кодовий лістинг з параметрами підключення до платформи ThingS.....	92
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (презентація).....	93

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API	-	Application Programming Interface
CMS	-	Content Management System
gRPC	-	Google Remote Procedure Call
UI	-	User Interface
FBS	-	Filmbox
glTF	-	GL Transmission Format
gRPC	-	Google Remote Procedure Call
bpy	-	Blender Python
HTTP	-	HyperText Transfer Protocol
RPG	-	Role-Playing Game
NPC	-	Non-Playable Character
FPS	-	Frames Per Second
UX	-	User Experience
JSON	-	User Experience

ВСТУП

В атестаційній роботі бакалавра розглядається актуальна проблема розробки ефективних інструментів для розробки веб-додатків, що дозволяють у режимі реального часу здійснювати моніторинг стану здоров'я користувача, є сучасним викликом та важливим напрямом інженерних досліджень..

Обґрунтування вибору теми та її актуальність. Вибір теми обумовлений необхідністю створення зручних та ефективних систем віддаленого моніторингу, які дозволяють не тільки збирати дані з датчиків фізіологічного стану, але й своєчасно реагувати на критичні зміни. Особливої важливості набуває застосування технологій сповіщення, таких як Webhook, що забезпечують миттєву інтеграцію з веб-сервісами та платформами. Така інтеграція дозволяє оперативно передавати дані у веб-браузер користувача, медичний портал або сервер аналітики, що значно підвищує ефективність роботи IoT-системи, що є актуальним.

Ступінь вивченої проблеми. Проблема створення IoT-рішень для моніторингу здоров'я є предметом численних наукових і технічних досліджень, особливо в контексті застосування мікроконтролерів типу ESP32, хмарних сервісів обробки даних, API та webhook-технологій. Значна кількість публікацій присвячена побудові архітектури таких систем, однак питання інтеграції механізмів миттєвого сповіщення у контексті медичних застосунків залишається відкритим [1, 3, 4, 7, 8, 14].

Постановка завдання дослідження. Розробка веб-додатку для системи моніторингу стану здоров'я з інтеграцією IoT та медичних пристроїв, полягає в створенні інноваційного інструменту – веб-додатку, який поєднує технології IoT, веб-розробку, медичні знання та дозволяє збирати, обробляти, аналізувати та візуалізувати дані про стан здоров'я користувачів у реальному часі для покращення якості медичного обслуговування та підтримки здоров'я користувачів. Тому виникає завдання розробки веб-додатку для моніторингу

фізіологічного стану користувача з використанням IoT технологій та реалізація миттєвого сповіщення за допомогою механізму Webhook.

Завданнями дослідження є:

- Аналіз можливостей впровадження IoT у сферу медичного моніторингу;
- Дослідження архітектури IoT-пристрою на базі ESP32;
- Вивчення способів інтеграції Webhook для сповіщення про критичні події;
- Розробка прототипу веб-додатку та тестування його працездатності.

Об'єкт дослідження: Процес віддаленого моніторингу стану здоров'я людини.

Предмет дослідження: Веб-додаток, що використовує IoT-пристрої та технологію Webhook.

Мета роботи: Підвищення ефективності контролю за станом здоров'я шляхом розробки веб-додатку, що використовує IoT-пристрої та технологію Webhook.

Метод дослідження. У дослідженні використано методи теоретичного аналізу, системного підходу, моделювання і розробки прототипу з подальшим тестуванням функціоналу у реальних умовах.

Завдання роботи:

1. Провести аналіз технологічних передумов щодо впровадження IoT-рішень у сферу моніторингу здоров'я з акцентом на використання механізму Webhook для забезпечення миттєвого надсилання сповіщень.
2. Дослідити методи реалізації системи збору, обробки та управління даними про фізіологічні показники користувача, отриманих з IoT-пристроїв, із використанням Webhook-технології.
3. Розробити практичні рекомендації щодо побудови архітектури веб-додатку для моніторингу здоров'я на основі IoT-технологій із впровадженням функціоналу миттєвого сповіщення через Webhook.

Наукове завдання – Оцінка ефективності розробленого веб-додатку, що використовує IoT-пристрої та технологію Webhook..

Результати дослідження. У результаті дослідження розроблено прототип веб-додатку для моніторингу фізіологічних показників користувача, який поєднує можливості IoT-пристрою на базі ESP32 та механізму Webhook. Система забезпечує збір, обробку та візуалізацію даних у режимі реального часу, а також реалізує функціонал миттєвих сповіщень у разі виявлення критичних змін параметрів. Запропонована модель підвищує ефективність моніторингу здоров'я та відкриває перспективи для подальшого використання в системах домашньої медицини та телемедицинських платформах.

Наукова новизна. Новизна полягає у поєднанні IoT-архітектури з механізмом миттєвого сповіщення Webhook, що дозволяє реалізувати високоефективну систему моніторингу для медичних і побутових задач.

Практична значущість результатів дослідження – результати дослідження можуть бути використані при розробці систем домашнього нагляду за станом здоров'я, у телемедицинських рішеннях, а також у спеціалізованих медичних установах для віддаленої діагностики.

Апробація результатів надається в 1 тезисах в рецензованих наукових журналах і виданнях.

Матеріали були опубліковані:

В тезисах:

1. Тимченко В.В. СТВОРЕННЯ ВЕБ-ДОДАТКУ МОНІТОРИНГ СТАНУ ЗДОРОВ'Я ПІД ЧАС ІНТЕГРАЦІЇ ІОТ ТА МЕДИЧНИХ ПРИСТРОЇВ// VI Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT». Збірник тез 15 квітня 2025. – К.: ДУІКТ, https://duikt.edu.ua/ua/news-1-570-13858-zaproshennya-na-vi-mizhnarodnu-naukovo-tehnichnu-konferenciyu-suchasniy-stan-ta-perspektivi-rozvitku-iot_kafedra-informaciynih-sistem-ta-tehnologiy

1 АНАЛІЗ МОЖЛИВОСТЕЙ ВИКОРИСТАННЯ ІОТ-ТЕХНОЛОГІЙ У СИСТЕМАХ МОНІТОРИНГУ ЗДОРОВ'Я

1.1 Аналіз шляхів впровадження ІоТ у галузі медицини та побутового моніторингу

Інтернет речей (Internet of Things, ІоТ) став однією з ключових технологій цифрової трансформації у багатьох сферах, зокрема й у галузі охорони здоров'я. Завдяки можливості постійного збору, передачі та обробки фізіологічних даних у реальному часі, ІоТ відкриває нові підходи до моніторингу стану здоров'я, лікування хвороб, профілактики ускладнень та організації медичної допомоги.

Традиційна модель надання медичних послуг ґрунтувалася на епізодичному відвідуванні пацієнтом закладу охорони здоров'я. Лікар мав доступ до обмеженого обсягу даних, які збиралися під час огляду або лабораторного аналізу. Водночас хвороби, особливо хронічні, мають динамічний характер, і проміжна втрата інформації могла призводити до помилок у діагностиці або лікуванні [1].

ІоТ дозволяє цю проблему вирішити шляхом забезпечення постійного моніторингу фізіологічних показників пацієнтів у режимі реального часу. За допомогою носимих пристроїв (wearables), медичних сенсорів, «розумного» одягу або вбудованих імплантів система фіксує життєво важливі параметри – такі як пульс, артеріальний тиск, рівень кисню в крові, температура тіла, рівень цукру, електрокардіограма тощо. Ці дані передаються через мережі зв'язку (Wi-Fi, Bluetooth, LTE, LoRa) на хмарні платформи або локальні сервери, де обробляються, аналізуються й візуалізуються для лікарів та пацієнтів [1].

Результат впровадження ІоТ у сфері охорони здоров'я є перехід від реактивної моделі лікування до проактивної та превентивної медицини. Пацієнт більше не змушений чекати прояву симптомів або чергового медичного огляду –

система сама виявляє відхилення, сповіщає користувача чи медичного працівника, а в разі критичної ситуації – ініціює автоматичну дію, наприклад, виклик швидкої допомоги.

IoT змінює підходи до лікування хронічних хвороб (діабет, гіпертонія, аритмія тощо), адже надає змогу вести безперервний контроль у домашніх умовах, що зменшує кількість госпіталізацій, покращує якість життя пацієнтів і водночас знижує навантаження на лікарні.

Інша важлива перевага – оптимізація роботи медичних закладів. IoT-технології використовуються не лише для моніторингу пацієнтів, а й для керування медичним обладнанням, логістикою, автоматизацією постачань, контролем умов зберігання ліків (наприклад, температури в холодильниках). Це підвищує ефективність внутрішніх процесів, знижує ризики і втрати.

У військовій, польовій та екстреній медицині IoT-технології використовуються для телемедицини, де завдяки віддаленому з'єднанню лікар може проводити консультацію, переглядати показники пацієнта та давати рекомендації без фізичної присутності.

Роль IoT у трансформації охорони здоров'я є фундаментальною. Вона полягає у зміні парадигми надання медичних послуг: від централізованої до персоналізованої, від епізодичної до безперервної, від реактивної до превентивної. У найближчому майбутньому ці технології стануть основою нової моделі цифрової медицини, де пацієнт буде активним учасником процесу, а не лише об'єктом лікування.

У сфері моніторингу здоров'я на основі IoT-технологій існує два основні напрями використання пристроїв: професійні медичні системи та споживчі гаджети (consumer-grade wearables). Кожна з цих категорій має свої особливості, переваги й обмеження, що визначають їх застосування, точність та надійність.

Призначення та сфера застосування:

– Професійні медичні системи використовуються в клініках, госпіталях, реабілітаційних центрах, а також для постійного моніторингу пацієнтів із серйозними діагнозами. Такі системи сертифіковані відповідно до медичних

стандартів (наприклад, FDA, CE) і забезпечують високий рівень достовірності показників.

– Споживчі гаджети орієнтовані на масовий ринок і використовуються для загального моніторингу фізичної активності, сну, серцевого ритму, рівня стресу тощо. Вони не вимагають медичної сертифікації й призначені більше для фітнесу та профілактики, ніж для діагностики [2].

Точність та надійність:

– Медичні пристрої (наприклад, портативні ЕКГ, глюкометри, оксиметри, тонометри) забезпечують високу точність, перевірену у клінічних дослідженнях. Вони мають калібрування, проходять технічне обслуговування, і можуть зберігати показники у медичних картках.

– Гаджети (наприклад, Xiaomi Mi Band, Apple Watch, Fitbit, Samsung Galaxy Watch) демонструють нижчу точність, оскільки використовують спрощені методи вимірювання. Наприклад, фотоплетизмографія для пульсу в умовах фізичної активності може давати похибки.

Обробка та передача даних:

– Професійні пристрої передають дані через спеціалізовані медичні шлюзи або сервери, часто – з використанням шифрування та автентифікації. Вони інтегруються з електронними медичними записами (EMR) та системами eHealth.

– Споживчі пристрої переважно працюють через мобільні додатки та хмарні сервіси виробників. Дані зазвичай зберігаються у власних облікових записах користувача та мають обмежений медичний контекст [3].

Зручність та доступність:

– Медичні пристрої часто потребують спеціальних знань для налаштування або інтерпретації результатів, і не завжди зручні для щоденного носіння.

– Гаджети є компактними, легкими, зручними у використанні, мають зручний інтерфейс, автоматично відстежують активність і сповіщають користувача про зміни стану.

Вартість:

– Професійне обладнання значно дорожче, часто призначене для багаторазового використання медичним персоналом.

– Споживчі пристрої доступні широкому колу користувачів, мають демократичну ціну, але обмежені в функціональності для медичного застосування.

З метою наочного представлення відмінностей між професійними медичними системами та споживчими IoT-пристроями, у таблиці 1.1 наведено їх порівняння за ключовими критеріями.

Таблиця 1.1 – Порівняння професійних медичних систем та споживчих IoT-гаджетів

Критерій	Професійні медичні системи	Споживчі гаджети (фітнес-браслети, смарт-годинники)
Призначення	Клінічне використання, діагностика, лікування	Самомоніторинг, фітнес, профілактика
Сертифікація	Підлягають медичній сертифікації (FDA, CE)	Відсутність або мінімальна сертифікація
Точність вимірювань	Висока, клінічно підтверджена	Помірна, залежить від умов використання
Типи показників	Розширений набір: ЕКГ, SpO2, артеріальний тиск, глюкоза	Обмежений набір: пульс, SpO2, активність, сон
Передача даних	Через захищені медичні канали, інтеграція з eHealth	Через мобільні додатки, Bluetooth/Wi-Fi
Інтерфейс	Призначений для лікарів та технічного персоналу	Інтуїтивно зрозумілий інтерфейс для звичайних користувачів
Зручність у використанні	Потребує спеціального навчання, громіздкіші	Компактні, легкі, прості у повсякденному застосуванні
Вартість	Висока	Середня або низька
Цільова аудиторія	Лікарі, медичні заклади, пацієнти під наглядом	Широке коло користувачів, які слідкують за здоров'ям

Застосування IoT-технологій у сфері охорони здоров'я охоплює різні рівні медичної допомоги – від високоспеціалізованих лікарень до амбулаторних клінік і домашнього середовища. У кожному з цих випадків Інтернет речей виконує свою функціональну роль, спрямовану на підвищення якості медичних послуг, оперативності реакції та зниження навантаження на персонал (рис. 1.1).



Рисунок 1.1 Основні напрями впровадження IoT у сфері охорони здоров'я [4]

У лікарнях та медичних центрах впровадження IoT має системний характер і охоплює як моніторинг пацієнтів, так і управління медичною інфраструктурою. Пацієнтам у палатах або реанімаціях підключають багатофункціональні IoT-пристрої, які безперервно фіксують життєво важливі показники (пульс, дихання, температура, ЕКГ). Дані надходять до єдиного центру моніторингу, де лікарі можуть оперативно реагувати на критичні зміни [4].

Крім того, IoT-системи використовуються для інвентаризації та контролю медичного обладнання, моніторингу температурних режимів холодильників з ліками, автоматизації розкладів процедур. Такий підхід забезпечує прозорість процесів і зменшує кількість помилок, пов'язаних з людським фактором.

Амбулаторні клініки та медичні кабінети, у поліклініках та приватних медичних центрах IoT переважно застосовується для первинної діагностики та

контролю пацієнтів із хронічними захворюваннями. Наприклад, носимі пристрої або компактні медичні сенсори дозволяють швидко виміряти тиск, рівень глюкози, сатурацію – результати автоматично передаються в електронну медичну картку або мобільний додаток лікаря.

Це значно скорочує час прийому, підвищує точність діагностики, а також дозволяє виявляти проблеми ще до появи яскраво виражених симптомів. Також застосовуються смарт-черги, електронні браслети для пацієнтів, мобільні форми взаємодії з лікарем.

Найбільш динамічний розвиток IoT отримав у домашніх умовах, завдяки широкій доступності недорогих фітнес-браслетів, розумних ваг, тонометрів та термометрів з Bluetooth/Wi-Fi. У поєднанні з мобільними додатками ці пристрої дозволяють звичайним користувачам щоденно контролювати свій стан здоров'я, отримувати поради щодо фізичної активності, сну, харчування.

Особливо цінним є використання IoT у догляді за людьми похилого віку, пацієнтами з обмеженою мобільністю або хронічними хворобами. Завдяки автоматичним сповіщенням та зв'язку з родичами чи лікарем, навіть незначні відхилення у здоров'ї можуть бути вчасно виявлені.

У ХХІ столітті технологічна трансформація охорони здоров'я набуває все більшого масштабу, і серед ключових рушіїв цього процесу є технології IoT, які тісно інтегруються з іншими цифровими інноваціями. Найактуальнішими тенденціями на сьогодні є розвиток телемедицини, мобільних додатків для здоров'я та концепція «розумних» лікарень (Smart Hospitals). Кожен із цих трендів забезпечує гнучкість, доступність і підвищену ефективність медичних послуг (рис. 1.2).



Рисунок 1.2 – Взаємозв'язок сучасних трендів в охороні здоров'я [5]

Телемедицина – це надання медичних послуг на відстані за допомогою телекомунікаційних технологій. Завдяки IoT-пристроям пацієнт може вимірювати життєво важливі показники у себе вдома, а лікар отримує ці дані в режимі реального часу через захищені канали зв'язку.

Ключові можливості телемедицини:

- Відеоконсультації з лікарем через спеціалізовані платформи;
- Дистанційне спостереження за хронічними пацієнтами;
- Термінові консультації для віддалених населених пунктів;
- Інтеграція з електронними медичними записами.

Особливо телемедицина виявила свою ефективність під час пандемії COVID-19, коли обмеження на пересування вимагали дистанційної взаємодії між лікарем і пацієнтом.

Мобільні додатки для здоров'я, смартфон став центром персонального моніторингу здоров'я. Існують тисячі мобільних додатків, які дозволяють:

- Вести щоденник самопочуття, фіксувати симптоми;
- Підключатися до фітнес-браслетів, смарт-годинників, пульсометрів;
- Отримувати рекомендації з харчування, фізичної активності, сну;

- Керувати медичними нагадуваннями (прийом ліків, візити до лікаря).

Завдяки API та Bluetooth-з'єднанню мобільні додатки здатні автоматично отримувати дані з IoT-пристроїв і передавати їх лікарю чи зберігати в хмарі для подальшого аналізу [5].

Концепція «розумної» лікарні (Smart Hospital) – це медичний заклад, в якому широко застосовуються технології IoT, штучного інтелекту, автоматизації та аналітики для покращення діагностики, лікування й управління ресурсами.

Ключові риси:

- Централізовані системи моніторингу пацієнтів;
- IoT-інфраструктура для відстеження розташування обладнання, персоналу, пацієнтів;
- Смарт-палати з інтерактивним керуванням середовищем (світло, температура, кнопки виклику);
- Інтегровані електронні медичні записи з аналітикою на основі AI.

Такі лікарні не лише підвищують якість лікування, а й суттєво оптимізують витрати, зменшують ризик медичних помилок і покращують загальний досвід пацієнта.

Сучасні тренди в галузі охорони здоров'я демонструють зсув у бік персоналізованої, віддаленої та технологічно орієнтованої медицини. Завдяки IoT у поєднанні з телемедичними платформами, мобільними застосунками та інтелектуальною лікарняною інфраструктурою стає можливою нова модель надання медичних послуг, де на першому місці – пацієнт, зручність і безпека.

1.2 Аналіз сучасних методів дистанційного контролю фізіологічних показників

У системах моніторингу здоров'я на основі IoT-технологій ключовим елементом є вимірювання біомедичних (фізіологічних) показників, які відображають стан організму в реальному часі. Найпоширенішими з них є температура тіла, частота серцевих скорочень (пульс), артеріальний тиск та рівень

насичення крові киснем (SpO_2) [5]. Кожен із цих параметрів має практичне значення як для щоденного самоаналізу, так і для клінічного застосування, з метою узагальнення параметрів, які використовуються у системах IoT-моніторингу здоров'я, у таблиці 1.2 представлено короткий опис основних біомедичних показників, що підлягають автоматичному вимірюванню в рамках інтелектуальних пристроїв.

Таблиця 1.2 – Основні біомедичні показники для IoT-моніторингу здоров'я [5]

Показник	Нормальні значення	Значення для моніторингу	Типові IoT-датчики
Температура тіла	36,1–37,2 °C	Підвищення при інфекціях, зниження при переохолодженні	MLX90614, DS18B20
Пульс (ЧСС)	60–100 уд/хв	Виявлення тахікардії, брадикардії, стресу	MAX30102, AD8232 (ЕКГ)
Артеріальний тиск	~120/80 мм рт. ст.	Виявлення гіпертонії/гіпотонії, контроль при серцевих хворобах	Осцилометричні тонометри
SpO_2 (сатурація)	95–100%	Зниження при дихальних розладах, COVID-19	MAX30100, пульсоксиметри

У сучасних IoT-пристроях для моніторингу стану здоров'я застосовуються різні технології вимірювання біомедичних параметрів. Вибір технології залежить від типу фізіологічного показника, точності, вимог до енергоспоживання та умов використання. Найпоширенішими є оптичні, електронні та імпульсні технології. Розглянемо кожен з них докладніше [6].

Оптичні сенсори працюють за принципом фотоплетизмографії (PPG) – аналізу змін поглинання світла при проходженні через тканини. Такий метод використовується для вимірювання:

- Частоти серцевих скорочень (ЧСС);
- Рівня насичення крові киснем (SpO_2);
- Судинного тону або варіабельності серцевого ритму.

Пристрої з оптичними сенсорами зазвичай мають світлодіоди (зелені, червоні або ІЧ) та фотодетектори, що зчитують зміну відбитого або пропущеного світла залежно від кровонаповнення.

Приклади сенсорів: MAX30102, MAX30100, TCRT1000.

Переваги:

- Безконтактність або мінімальний контакт зі шкірою;
- Підходить для носимих пристроїв;
- Низьке енергоспоживання.

Недоліки:

- Менша точність при русі або поганому приляганні до шкіри;
- Чутливість до освітлення та кольору шкіри.

Електронні технології – ці технології базуються на електричних характеристиках тіла, наприклад, опору, напруги чи провідності. Вони застосовуються для вимірювання:

- Температури тіла (за допомогою терморезисторів або термопар);
- Електрокардіограми (ЕКГ) – шляхом зчитування електричних сигналів серця;
- М'язової активності (ЕМГ) чи мозкової активності (ЕЕГ).

Приклади сенсорів: DS18B20 (температура), AD8232 (ЕКГ), LM35, термістори NTC/PTC.

Переваги:

- Висока точність;
- Пряма залежність від фізіологічного параметра;
- Використовується у медичних сертифікованих пристроях.

Недоліки:

- Потреба в щільному контакті з тілом;
- Чутливість до вологи, шумів або електромагнітних завад.

Імпульсні технології використовуються у пристроях, які фіксують зміни в тиску, частоті або русі на основі імпульсів електричних або механічних сигналів. Вони актуальні для вимірювання:

- Артеріального тиску (осцилометричний метод);
- Механічних коливань пульсу;
- Швидких змін у ритмі серця або м'язів.

Приклади сенсорів: BMP280 (тиск), сенсори сили (FSR), акселерометри (для непрямого визначення пульсу).

Переваги:

- Можливість створення портативних пристроїв без манжети;
- Інтеграція з іншими сенсорами в комплексні IoT-рішення.

Недоліки:

- Не завжди достатня точність для клінічного використання;
- Потреба в калібруванні.

Розмаїття технологій вимірювання дає змогу адаптувати IoT-рішення до конкретних медичних задач. Оптичні сенсори ідеальні для носимих пристроїв, електронні – для точного медичного моніторингу, а імпульсні – для створення мобільних та енергоефективних платформ. У сучасних системах часто застосовується гібридний підхід, коли в одному пристрої поєднуються кілька типів технологій для підвищення точності та функціональності [7].

Передача даних у системах IoT-моніторингу здоров'я є критично важливою складовою, оскільки вона забезпечує безперервний потік інформації від сенсорів до оброблювальних платформ, мобільних застосунків або хмарних сховищ. Вибір технології передачі залежить від умов експлуатації, енергоспоживання, обсягу даних і необхідного радіусу дії. Найпоширенішими є Wi-Fi, Bluetooth та LoRa (табл. 1.3).

Таблиця 1.3 – Порівняльна характеристика технологій передачі даних в IoT-системах охорони здоров'я [7]

Параметр	Wi-Fi	Bluetooth / BLE	LoRa
Радіус дії	До 30 м у приміщенні	До 10–20 м	До 10 км (у відкритій місцевості)
Швидкість передачі	Висока (до 100 Мбіт/с)	Середня (до 2 Мбіт/с для BLE)	Низька (кілька кбіт/с)
Енергоспоживання	Високе	Дуже низьке (оптимально для носимих пристроїв)	Дуже низьке
Застосування	Домашні IoT-системи, передача у хмару	Носимі пристрої, підключення до смартфонів	Сільські медичні пункти, автономні пристрої
Складність реалізації	Середня	Низька	Висока (потребує базової станції або шлюзу)

Незважаючи на стрімкий розвиток IoT-пристроїв для медичного моніторингу, точність отриманих даних досі залишається серйозним викликом. У порівнянні з професійним обладнанням, носимі та споживчі пристрої можуть демонструвати суттєві відхилення через різноманітні фактори:

Погане прилягання сенсора до шкіри, наприклад оптичні сенсори на зап'ясті не завжди щільно контактують зі шкірою, особливо при русі або потовиділенні, що призводить до похибок у вимірюванні пульсу чи SpO_2 .

Артефакти руху, під час фізичної активності дані зчитуються нестабільно. Сенсори можуть "сприймати" вібрації або рух руки як зміну пульсу, що викривлює результати.

Зовнішнє освітлення (для оптичних сенсорів), вологість або температура можуть впливати на роботу пристроїв. Наприклад, надто яскраве світло може "засвітити" фотодіод сенсора.

Наявність перешкод або шумів, електронні сенсори, як-от ЕКГ, можуть "ловити" електромагнітні завади з навколишнього середовища, що викликає шум на графіках.

Невідповідність калібрування, багато споживчих IoT-пристроїв не проходять індивідуальне калібрування. Без цього сенсор може демонструвати стабільну, але неточну серію вимірювань [8].

1.3 Аналіз прикладів застосування IoT у системах контролю стану здоров'я

Інтернет речей (IoT) у сфері охорони здоров'я активно впроваджується у формі пристроїв, які забезпечують постійний моніторинг фізіологічного стану людини, фіксують параметри життєдіяльності та передають їх у хмарні сервіси або мобільні додатки. Застосування IoT-технологій охоплює як масові споживчі рішення, так і високоспеціалізовані медичні системи. Нижче наведено аналіз відомих кейсів, інноваційних рішень, а також оцінку їх функціональності, точності та потенційних ризиків.

Apple HealthKit – це програмна екосистема для iOS (рис. 1.3), яка дозволяє збирати, аналізувати та візуалізувати медичні й фітнес-дані з різних пристроїв (Apple Watch, сторонні датчики). HealthKit інтегрується з клінічними системами, підтримує автоматичний облік пульсу, кроків, калорій, фаз сну, SpO₂. Дані можуть зберігатися у Apple Health Records – електронній медичній картці [9].



Рисунок 1.3 – Програмна екосистема Apple HealthKit [9]

- Функціональність: широка, інтелектуальні алгоритми аналізу
- Точність: висока, сертифіковані сенсори в Apple Watch
- Інтеграція: з iOS-додатками та сторонніми платформами (наприклад, MyFitnessPal, Medisafe)

Google Fit – це аналог Apple HealthKit для Android-пристроїв. Додаток синхронізується з фітнес-браслетами (рис. 1.4), смарт-годинниками та дозволяє стежити за фізичною активністю, частотою серцевих скорочень, вагою тощо. Google Fit підтримує API для сторонніх розробників, що робить його універсальним хабом IoT-пристроїв у сфері здоров'я [10].



Рисунок 1.4 – Google Fit Android-пристроїв [10]

- Функціональність: середня, основні показники
- Зручність: проста інтеграція з Android-середовищем
- Ризики: залежність від точності сторонніх пристроїв

Xiaomi Mi Band – це популярний бюджетний фітнес-трекер, що вимірює пульс, кроки, сон, SpO₂, рівень стресу. Дані передаються в мобільний додаток Mi Fit або Zepp Life. Пристрій має Bluetooth-з'єднання та може використовуватись у повсякденному житті для базового контролю здоров'я (рис. 1.5).



Рисунок 1.5 – Фітнес-трекер Xiaomi Mi Band [11]

- Переваги: низька вартість, довгий час автономної роботи
- Недоліки: обмежена точність, особливо під час руху
- Призначення: для самоспостереження, а не клінічного застосування

Medtronic Guardian Connect – це професійна медична система для моніторингу рівня глюкози у хворих на діабет. Вона складається з сенсора, встановленого на шкірі, та мобільного додатка (рис. 1.6). Пристрій вимірює рівень глюкози кожні 5 хвилин і попереджає про ризик гіпо- або гіперглікемії [11].

- Точність: висока, сертифікований медичний пристрій
- Цільова аудиторія: пацієнти з інсулінозалежним діабетом
- Функції: прогнози рівня глюкози, обмін з лікарем, звіти



Рисунок 1.6 – Medtronic Guardian Connect [11]

Розумна футболка (Smart Shirt) – це текстиль з вбудованими біосенсорами, які зчитують параметри ЧСС, дихання, температури, пози тіла. Дані можуть передаватись через Bluetooth у мобільний додаток або сервер. (рис. 1.7).

Приклад: Hexoskin Smart Garment або Sensoria Shirt.

- Переваги: зручність для постійного носіння, висока точність
- Недоліки: висока вартість, обмежена доступність

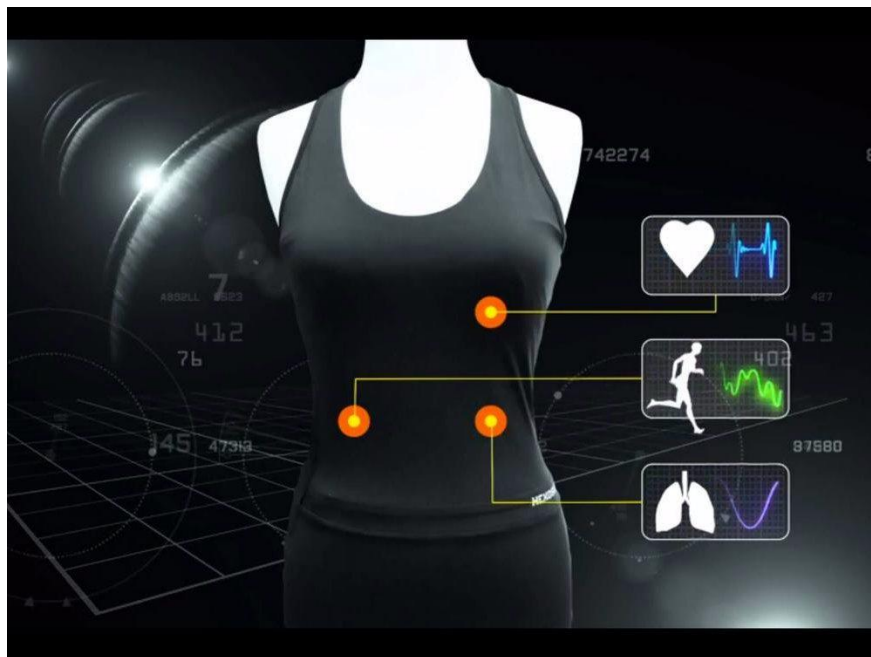


Рисунок 1.7 – Розумна футболка (Smart Shirt) [11]

(Smart Patches) – мініатюрні медичні пристрої, які прикріплюються до шкіри і вимірюють глюкозу, температуру, електрокардіограму (рис. 1.8).



Рисунок 1.8 – Пластир для кріплення Smart Patches [12]

Приклад: BioSticker (розробка компанії BioIntelliSense).

- Переваги: непомітність, безперервність вимірювань
- Недоліки: обмежений час використання (до 7 днів), висока вартість

ІоТ-пристрої відіграють все важливішу роль у моніторингу стану здоров'я як у повсякденному житті, так і в медичній практиці. Від доступних фітнес-браслетів до високоточних медичних систем – усі ці рішення мають потенціал покращити якість життя, запобігти ускладненням та зменшити навантаження на медичну систему. Водночас їх впровадження потребує врахування точності, ризиків безпеки та рівня відповідальності користувача [12].

1.4 Аналіз тенденції розвитку веб-рішень для віддаленого моніторингу стану здоров'я

З розвитком цифрових технологій та зростанням попиту на дистанційне спостереження за станом здоров'я з'являється все більше веб-рішень, що інтегруються з ІоТ-пристроями та забезпечують користувачам і лікарям

можливість візуалізувати, аналізувати й обмінюватися медичними даними. Такі рішення базуються на сучасних фронтенд і бекенд-технологіях, хмарних інфраструктурах та API-сервісах, створюючи цілісні екосистеми для віддаленого моніторингу [13].

Веб-інтерфейси для лікарів і користувачів

Сучасні веб-інтерфейси виконують роль основного засобу взаємодії між пацієнтом, медичним закладом і пристроями моніторингу.

- Для пацієнтів це зручні панелі, де відображаються показники пульсу, тиску, SpO₂, температура тощо. Дані часто подаються у вигляді графіків, таблиць, попереджень.

- Для лікарів веб-інтерфейси надають доступ до історії вимірювань, дозволяють виявляти тенденції, формувати медичні звіти, ставити діагнози віддалено.

Така взаємодія особливо корисна для управління хронічними хворобами, телемедичних консультацій, дистанційного нагляду за пацієнтами після операцій або в умовах карантину.

Хмарні сервіси для збору/візуалізації медичних даних

Однією з тенденцій є перехід до хмарних платформ, які забезпечують централізоване зберігання даних з IoT-пристроїв, обробку інформації в реальному часі та масштабовану візуалізацію.

Типові хмарні сервіси:

- Google Cloud, AWS IoT, Microsoft Azure Health – надають інструменти для обробки медичних даних, створення облікових записів користувачів, налаштування прав доступу;

- Firebase – популярне рішення для малих і середніх IoT-проектів з вбудованими базами даних, авторизацією та реальногочасовою синхронізацією.

Переваги хмар:

- Віддалений доступ 24/7;
- Масштабованість і автоматичне резервне копіювання;
- Безпечна передача даних (HTTPS, шифрування, автентифікація) [14].

Використання REST API для обміну даними

REST API (Representational State Transfer Application Programming Interface) став стандартом у розробці IoT- та веб-рішень, оскільки забезпечує простий і масштабований спосіб обміну даними між пристроями, клієнтами та серверами.

Переваги REST API у системах моніторингу здоров'я:

- Передача даних від мікроконтролера (наприклад, ESP32) до сервера або хмари у форматі JSON;
- Запити GET/POST для отримання чи відправки фізіологічних параметрів;
- Аутентифікація користувачів для захисту даних.

Використовуючи REST API, IoT-пристрій може надсилати дані на веб-платформу щосекунди або при кожному новому вимірюванні, що дозволяє створити динамічну систему моніторингу в реальному часі.

Актуальні технології: MQTT, Firebase, Node.js, React

У сучасній архітектурі веб-систем для моніторингу здоров'я найчастіше застосовуються наступні технології:

- MQTT (Message Queuing Telemetry Transport) – легкий мережевий протокол обміну повідомленнями, ідеальний для IoT-пристроїв. Використовується у випадках, коли необхідна реальночасова передача даних з низьким енергоспоживанням, особливо в системах з десятками тисяч сенсорів.
- Firebase – платформа Google для створення серверної частини без власного сервера. Включає базу даних (Firestore), систему аутентифікації, хмарні функції, push-сповіщення. Firebase чудово підходить для швидкого створення прототипів IoT-рішень, зокрема – медичних веб-додатків.
- Node.js – серверне середовище для запуску JavaScript. Завдяки своїй асинхронній природі Node.js дозволяє ефективно обробляти потік даних від десятків IoT-пристроїв. Добре інтегрується з базами даних (MongoDB, Firebase) і API [15].
- React – JavaScript-бібліотека для створення динамічних інтерфейсів користувача. Використовується для розробки адаптивних та інтерактивних веб-

додатків, які дозволяють зручно переглядати медичні дані на комп'ютерах, планшетах і смартфонах.

Тенденції розвитку веб-рішень у сфері моніторингу здоров'я свідчать про зростання значення інтеграції IoT з веб-технологіями для забезпечення ефективного, безпечного та персоналізованого медичного обслуговування. Комбінація сучасних фреймворків, протоколів та хмарних сервісів дозволяє створювати гнучкі системи, які дають змогу лікарям і пацієнтам спільно керувати здоров'ям у цифровому середовищі.

2 ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ІОТ-ТЕХНОЛОГІЙ У СИСТЕМАХ МОНІТОРІНГУ ЗДОРОВ'Я

2.1 Дослідження можливостей контролера ESP32 для медичних ІоТ-проектів

ESP32 є сучасним мікроконтролером із підтримкою бездротових технологій Wi-Fi та Bluetooth, що широко застосовується для створення систем Інтернету речей (IoT). Пристрій розроблений компанією *Espressif Systems* та відзначається високою продуктивністю, енергоефективністю і функціональною гнучкістю, що забезпечує його популярність серед розробників рішень для моніторингу та автоматизації [16].

Основні технічні характеристики ESP32 включають:

- подвійне ядро Tensilica Xtensa LX6 із тактовою частотою до 240 МГц;
- підтримка бездротових технологій Wi-Fi (IEEE 802.11 b/g/n) та Bluetooth v4.2 / BLE (Bluetooth Low Energy);
- обсяг оперативної пам'яті до 520 КБ SRAM;
- вбудована flash-пам'ять обсягом від 4 МБ (залежно від моделі);
- наявність більш ніж 30 цифрових і аналогових входів/виходів (GPIO), включаючи ADC, DAC, SPI, I²C, UART, PWM;
- реалізація енергоощадних режимів роботи (deep sleep, light sleep);
- апаратна підтримка алгоритмів криптографії (AES, SHA-2, RSA, ECC);
- вбудований датчик температури та датчик Холла (в окремих модифікаціях);
- підтримка технології OTA (Over-The-Air) для дистанційного оновлення прошивки.

Застосування IoT-технологій у сфері моніторингу стану здоров'я потребує використання контролерів, які поєднують стабільність, енергоефективність та безпеку обробки персональних даних. У цьому контексті ESP32 має низку переваг, що визначають його доцільність у подібних розробках:

- підтримка двох основних бездротових інтерфейсів – Wi-Fi та Bluetooth Low Energy, що дає можливість організувати як локальні мережеві з'єднання, так і передавати дані на віддалені сервери;
- висока енергоефективність завдяки наявності режимів енергозбереження, що дозволяє використовувати пристрої на базі ESP32 для тривалого автономного моніторингу;
- достатня обчислювальна потужність для обробки фізіологічних даних у режимі реального часу, зокрема для попередньої фільтрації, статистичної обробки та виявлення критичних відхилень показників;
- апаратна реалізація засобів криптографічного захисту, яка забезпечує конфіденційність переданих медичних даних та відповідає сучасним стандартам безпеки;
- широка сумісність із різними типами сенсорів та медичних пристроїв завдяки наявності універсальних інтерфейсів підключення, таких як SPI, I²C, UART та інші;
- доступність численних програмних бібліотек, документації та активної спільноти розробників, що спрощує створення, тестування та масштабування IoT-систем;
- можливість використання ОТА-оновлень для дистанційного оновлення програмного забезпечення без необхідності фізичного втручання, що підвищує надійність та зручність обслуговування системи [17].

Вибір ESP32 для створення прототипу системи моніторингу здоров'я зумовлений низкою технічних, економічних та експлуатаційних чинників, які забезпечують ефективну реалізацію поставлених завдань. Серед основних аргументів на користь вибору ESP32 можна виділити такі:

- вбудована підтримка Wi-Fi та Bluetooth Low Energy, що дозволяє одночасно реалізувати локальну взаємодію з мобільними пристроями та віддалене з'єднання з серверною частиною або хмарними платформами;
- наявність енергоефективних режимів роботи, які забезпечують зменшення енергоспоживання, що є ключовим чинником для носимих або портативних пристроїв моніторингу здоров'я;
- інтеграція функціоналу OTA-оновлень, яка забезпечує можливість дистанційного оновлення прошивки пристрою, що дозволяє оперативно усувати вразливості та додавати нові функції без необхідності фізичного доступу;
- широка доступність програмного забезпечення для розробки, зокрема Arduino IDE, PlatformIO та ESP-IDF, що значно спрощує реалізацію та налагодження IoT-системи;
- оптимальне співвідношення вартості та функціональності, що робить ESP32 економічно вигідним рішенням для розробки масових медичних IoT-пристроїв [18].

Ураховуючи ці переваги, застосування ESP32 дозволяє створити стабільну, безпечну та економічно обґрунтовану платформу для реалізації системи віддаленого моніторингу стану здоров'я, що відповідає сучасним вимогам телемедицини та домашнього медичного контролю.

У системах медичного моніторингу на базі IoT-технологій вибір мікроконтролера є ключовим фактором, що визначає стабільність роботи пристрою, енергоефективність, точність збору та безпеку переданих даних. Серед широкого спектра доступних рішень мікроконтролер ESP32 відзначається оптимальним поєднанням продуктивності, надійності та функціональних можливостей, що робить його доцільним вибором для побудови систем дистанційного контролю фізіологічних показників [19]. Його технічні характеристики дозволяють інтегрувати різноманітні сенсори, реалізовувати бездротову передачу даних через Wi-Fi або Bluetooth, а також забезпечувати захист персональної інформації пацієнтів завдяки апаратній підтримці криптографії. У табл. 2.1 наведено основні характеристики мікроконтролера

ESP32 та їх переваги для застосування в системах медичного моніторингу здоров'я.

Таблиця 2.1 – Основні характеристики та переваги мікроконтролера ESP32 для медичних IoT-проектів [18]

Характеристика	Опис	Переваги для IoT-моніторингу здоров'я
Тип процесора	Подвійне ядро Tensilica Xtensa LX6, до 240 МГц	достатня обчислювальна потужність для обробки даних у реальному часі;
Бездротові технології	Wi-Fi (802.11 b/g/n), Bluetooth v4.2 / BLE	одночасна передача даних локально та віддалено;
Оперативна пам'ять	До 520 КБ SRAM	можливість роботи з потоками даних від кількох сенсорів;
Flash-пам'ять	Від 4 МБ	зберігання необхідних параметрів, конфігурацій та логів;
Кількість GPIO	Понад 30 цифрових і аналогових входів/виходів	можливість підключення різних біомедичних сенсорів;
Підтримка криптографії	AES, SHA-2, RSA, ECC	забезпечення безпеки та конфіденційності медичних даних;
Енергоощадні режими	Deep sleep, light sleep	тривалий час автономної роботи для портативних пристроїв;
Можливість ОТА-оновлень	Підтримується	зручне дистанційне оновлення прошивки та функціоналу;
Середовище розробки	Arduino IDE, PlatformIO, ESP-IDF	простота розробки та наявність великої кількості бібліотек;
Вартість	Низька (економічне рішення для масових проектів)	зниження загальних витрат на розробку пристроїв;

2.2 Дослідження можливостей застосування пристроїв збору даних: датчики, модулі зв'язку та живлення

Ефективність функціонування системи моніторингу стану здоров'я на базі IoT значною мірою залежить від правильно підібраних апаратних компонентів, які забезпечують зчитування, передачу та зберігання фізіологічних даних користувача. Серед основних складових таких систем виділяють сенсорні модулі для вимірювання життєво важливих показників, модулі живлення для підтримки стабільної роботи пристрою, а також засоби бездротової передачі інформації [20].

У системах IoT-моніторингу здоров'я сенсори виконують функцію збору базових біомедичних параметрів, що відображають фізіологічний стан людини. Серед найбільш поширених показників, що контролюються подібними пристроями, є частота серцевих скорочень (пульс), рівень насичення крові киснем (SpO_2), температура тіла та артеріальний тиск. Для кожного з цих параметрів існують специфічні типи сенсорів, які мають свої особливості, переваги та обмеження.

Для моніторингу частоти серцевих скорочень широко застосовується сенсор MAX30102, який базується на оптичному методі фотоплетизмографії. Цей датчик забезпечує вимірювання пульсу та SpO_2 , використовуючи червоні та інфрачервоні світлодіоди у поєднанні з фотодетектором. Серед його переваг – компактність, низьке енергоспоживання та висока чутливість, однак точність може знижуватися при активних рухах або слабкому контакті із шкірою.

Для визначення температури тіла часто використовують DS18B20 – цифровий термометр із інтерфейсом 1-Wire, що забезпечує стабільне та точне вимірювання температури в діапазоні від $-55\text{ }^{\circ}\text{C}$ до $+125\text{ }^{\circ}\text{C}$. В окремих випадках застосовуються інфрачервоні сенсори, такі як MLX90614, які дають змогу виконувати безконтактні вимірювання температури поверхні тіла, що зручно для використання в носимих або побутових пристроях.

Контроль артеріального тиску здебільшого здійснюється за допомогою осцилометричних тонометрів, які використовують сенсори тиску, наприклад

ВМР280 або подібні варіанти. Ці сенсори здатні забезпечувати точне вимірювання змін тиску в манжеті та перетворення цих даних на діагностично значущі показники [21].

Вибір сенсорних компонентів визначається необхідною точністю, енергоспоживанням, зручністю інтеграції з мікроконтролером, а також специфікою медичних завдань системи моніторингу.

Оскільки пристрої моніторингу стану здоров'я, побудовані на базі IoT-технологій, зазвичай мають портативний характер і повинні працювати автономно впродовж тривалого часу, питання організації живлення є надзвичайно важливим. Найчастіше для таких систем застосовуються літій-іонні (Li-Ion) або літій-полімерні (Li-Po) акумулятори, які мають високу енергоємність, відносно компактні розміри та низький рівень саморозряду.

Одним із ключових аспектів є впровадження механізмів енергозбереження, зокрема використання режимів *deep sleep* і *light sleep*, які підтримуються ESP32 та подібними мікроконтролерами. У цих режимах контролер та периферійні модулі переходять у стан мінімального енергоспоживання, що дозволяє значно подовжити термін автономної роботи пристрою.

Важливим аспектом проектування є вибір акумуляторів відповідної ємності, здатних забезпечити безперервну роботу пристрою протягом визначеного періоду. Для підвищення надійності роботи елементів живлення доцільно використовувати схеми контролю заряду, які виконують функції обмеження перезарядження, захисту від короткого замикання та надмірного розряду. Це сприяє стабільності роботи системи та зменшує ризики, пов'язані з виходом з ладу енергетичних компонентів. Окрім цього, за певних умов доцільним є застосування додаткових джерел живлення, зокрема модулів відновлюваної енергетики, що дозволяє підвищити автономність роботи пристрою та зменшити залежність від традиційних методів підзаряджання [22].

Для забезпечення передачі даних від IoT-пристроїв до кінцевого користувача або серверної частини системи використовуються різноманітні технології бездротового зв'язку. Вибір відповідного типу зв'язку залежить від

характеристик застосування, відстані передачі, енергоспоживання та обсягу переданих даних.

Bluetooth Low Energy (BLE) є оптимальним рішенням для короткодистанційної передачі даних від сенсорів до мобільного пристрою або локального IoT-шлюзу. Основними перевагами Bluetooth є:

- низьке енергоспоживання;
- простота реалізації;
- широка підтримка з боку сучасних смартфонів та планшетів.

До недоліків Bluetooth відносяться обмежений радіус дії (до 10–20 метрів) та обмежена швидкість передачі великих обсягів даних.

Wi-Fi (802.11 b/g/n) забезпечує високу швидкість передачі даних і є зручним для інтеграції з хмарними платформами та локальними серверами. Серед переваг цієї технології можна відзначити:

- високу пропускну здатність;
- можливість роботи з існуючими мережами Wi-Fi;
- гнучкість у побудові мережевої інфраструктури.

Однак недоліками Wi-Fi є відносно високе енергоспоживання та обмежений радіус дії в межах приміщення (до 30 метрів).

LoRa (Long Range) є технологією з наднизьким енергоспоживанням і великим радіусом дії (до 10 км на відкритій місцевості), що робить її доцільною для використання в сільських, віддалених або малодоступних районах. Основні переваги LoRa включають:

- великий радіус дії;
- низьке енергоспоживання;
- можливість побудови приватних мереж.

Недоліками LoRa є низька швидкість передачі даних (кілька кбіт/с) і необхідність додаткового обладнання для побудови мережі (LoRa-шлюзи).

Вибір конкретної технології бездротового зв'язку повинен ґрунтуватися на аналізі умов експлуатації, необхідної швидкості передачі, енергоспоживання та відстані між пристроями та кінцевим пунктом збору даних. У багатьох системах

доцільним є комбінування кількох технологій для досягнення оптимального балансу між автономністю, надійністю та функціональністю (табл 2.2).

Таблиця 2.2 – порівняння технологій бездротового зв'язку (Bluetooth, Wi-Fi, LoRa) [23].

Параметр	Bluetooth / BLE	Wi-Fi	LoRa
Радіус дії	10–20 м	до 30 м (у приміщенні)	до 10 км (у відкритій місцевості)
Швидкість передачі	до 2 Мбіт/с (BLE)	до 100 Мбіт/с	кілька кбіт/с
Енергоспоживання	дуже низьке	високе	дуже низьке
Складність реалізації	низька	середня	висока (потрібні шлюзи)
Застосування	фітнес-трекери, медичні сенсори	домашні IoT-системи, хмарна передача	віддалений моніторинг, сільські райони

2.3 Побудова схеми IoT-пристрою для зчитування показників

Процес віддаленого моніторингу стану здоров'я пацієнта в системах IoT розпочинається зі зчитування ключових фізіологічних показників за допомогою спеціалізованих медичних пристроїв (рис. 2.1). До таких показників належать: вага, артеріальний тиск, рівень глюкози в крові, насиченість крові киснем (SpO_2), частота серцевих скорочень, а також електрокардіограма (ЕКГ).

Ці дані фіксуються за допомогою переносних медичних приладів, оснащених вбудованими сенсорами для зручного та оперативного збору інформації. Далі отримані результати автоматично передаються через бездротові канали зв'язку до Bluetooth IoT шлюзу, який координує обмін інформацією між медичними датчиками та смартфоном пацієнта для подальшого опрацювання та аналізу [23].



Рисунок 2.1 – Архітектура віддаленого медичного моніторингу із застосуванням IoT-технологій [24]

Смарт-годинник із багатофункціональним дисплеєм забезпечує зручний доступ до ключових фізіологічних показників, включаючи частоту серцевого ритму, підрахунок кроків та якість сну. Завдяки синхронізації з мобільним пристроєм годинник автоматично передає сповіщення про вхідні дзвінки, текстові повідомлення та повідомлення з різних застосунків [24].

Важливою перевагою таких пристроїв є їх повна сумісність з інтелектуальними IoT-шлюзами, що дозволяє інтегрувати смарт-годинник у єдину екосистему віддаленого моніторингу та управління здоров'ям, створюючи комфортне середовище для користувача (Рис.2.2).

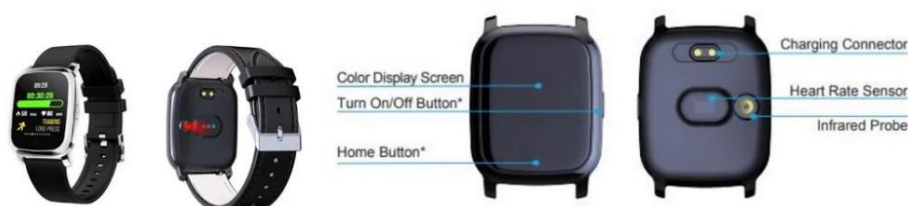


Рисунок 2.2 – Приклад інтерфейсу багатофункціонального Smart Watch для збору даних про активність та здоров'я [24]

Основні технічні характеристики Smart Watch для моніторингу стану здоров'я:

- Bluetooth-чіпсет: Nordic 52832.
- Датчик руху (G-сенсор): Kionix-KX023.
- Температурний сенсор: інфрачервоний датчик.

- Сенсор серцевого ритму: HRS3313 (інфрачервоний).
- Версія Bluetooth: 5.04.
- Дальність дії Bluetooth: 8–10 метрів.

Пам'ять та зберігання:

- Об'єм пам'яті (RAM + ROM + Flash): 64 КБ + 512 КБ + 4 МБ.
- Термін зберігання даних: до 3 днів.

Конструктивні особливості:

- Водонепроникність: рівень захисту IP67.
- Тип заряджання: магнітний кабель.
- Матеріал корпусу: поєднання загартованого скла, цинкового сплаву та ABS-пластику.
- Матеріал ремінця: метал, TPU або шкіра (залежно від модифікації).
- Габарити корпусу: 51,7 мм × 37,3 мм × 11,5 мм.
- Вага пристрою: 48 г.

Дисплей:

- Тип екрану: TFT IPS.
- Діагональ: 1,30 дюйма.
- Роздільна здатність: 240 × 240 пікселів.

Енергоспоживання:

- Ємність акумулятора: 140 мА·год.
- Час активної роботи: до 7 днів.
- Час роботи в режимі очікування: до 30 днів [25].

Серед ключових апаратних рішень для побудови систем Інтернету речей варто виділити низку спеціалізованих модулів та пристроїв, що мають енергоефективну архітектуру та високий рівень інтеграції.

DSOM-060N – це модульна система, побудована на базі процесора NXP I.MX6ULLL, який належить до лінійки енергоощадних процесорів, орієнтованих на широке застосування в IoT-рішеннях, де важлива оптимізація вартості та споживання енергії.

DSM-050 представляє собою вбудований Bluetooth-модуль, що підтримує версію 5.2 та характеризується низьким енергоспоживанням. Основою модуля є високоінтегрований бездротовий чіп nRF52840, доповнений набором периферійних пристроїв, що робить його ідеальним варіантом для Bluetooth Low Energy застосувань.

DSGK-930 RK3328 – це промислова материнська плата, яка базується на процесорі Rockchip RK3288. Завдяки широкому набору інтерфейсів для введення та виведення даних, плата адаптована до складних промислових завдань та автоматизації [26].

Важливою складовою систем IoT для моніторингу здоров'я є також тонометр нового покоління, створений на базі звіту WBP202. Це портативний пристрій у вигляді зап'ястка, який забезпечує високу точність вимірювання артеріального тиску. Прилад здатен зчитувати показники в п'ять разів більше точок, що підвищує надійність результатів. Окрім цього, пристрій має функцію виявлення нерегулярного серцебиття, зберігає до 500 груп вимірювань та оснащений зручною манжетою з D-кільцем для комфортного використання. Просте керування одним дотиком робить цей тонометр зручним рішенням для щоденного моніторингу артеріального тиску.

Основні технічні характеристики IoT-монітора артеріального тиску:

- Ергономічний форм-фактор із можливістю комфортного носіння на зап'ястку, що дозволяє проводити вимірювання артеріального тиску та пульсу в будь-який зручний момент.
- Вбудований літєвий акумулятор, який забезпечує автономну роботу пристрою без необхідності частого підзаряджання.
- Підтримка функції обміну та синхронізації даних з іншими пристроями через бездротові інтерфейси.
- Оснащений великим екраном з чітким відображенням результатів вимірювань, що підвищує зручність користування навіть для людей з порушеннями зору [27].

– Можливість хмарного зберігання результатів та віддаленого обміну інформацією для доступу до даних із будь-якої точки світу.

Наявність трьох варіантів манжет:

- Велика (32–42 см) – опційно для дорослих користувачів;
- Стандартна (22–32 см) – основна для дорослих;
- Мала (17–22 см) – додатковий варіант для користувачів із вузьким зап'ястям.

Підтримка двох окремих облікових записів користувачів, кожен з яких здатен зберігати до 500 груп результатів вимірювань, що дозволяє використовувати пристрій для кількох осіб без втрати даних. (Рис.2.3)



Рисунок 2.3 – Пристрій для вимірювання артеріального тиску та пульсу з функціями IoT [26]

Технічні характеристики IoT-монітора артеріального тиску [28]:

- Тип пристрою: Bluetooth-монітор тиску (версія Bluetooth 4).
- Час одного вимірювання: від 20 до 45 секунд.
- Радіус дії Bluetooth: до 10 метрів.
- Акумулятор: літій-іонний, перезаряджуваний.
- Автономна робота: не менше 24 годин на одному заряді.
- Час повної зарядки: не перевищує 2 години.
- Окружність манжети: від 22 см до 32 см (стандартний розмір для дорослих)

Діапазон і точність вимірювань:

- Систолічний тиск: 30–255 мм рт.ст. (4.0–34.0 кПа).
- Діастолічний тиск: 10–220 мм рт.ст. (1.3–29.3 кПа).

- Частота пульсу (PR): 40–240 ударів на хвилину з точністю $\pm 5\%$ або ± 3 уд/хв (залежно від більшого значення).

Пам'ять та зберігання:

- Обсяг пам'яті: до 500 груп збережених результатів.

Параметри Bluetooth-з'єднання:

- Потужність передавача: 4 дБм.
- Чутливість приймача: -80 дБм при 0.1% BER.
- Зміщення частоти: ± 20 кГц.
- Діапазон робочих частот: від 2.405 до 2.480 ГГц.
- Смуга пропускання: 2 МГц.
- Тип модуляції: GFSK.
- Еквівалентна потужність ізотопного випромінювання (EIRP): менше 10 мВт.

Фізичні параметри:

- Габарити пристрою: 130 мм $\times$ 55 мм $\times$ 30 мм.
- Вага: не вказано (Н/Д).

Сертифікація та відповідність стандартам:

- Наявні сертифікати: CE, FDA, ISO 13485.

Body Composition Scale – це сучасний пристрій для контролю фізіологічних параметрів, який дозволяє не лише фіксувати загальну вагу користувача, але й аналізувати такі показники, як відсоток жирової тканини, м'язова маса, кісткова маса та рівень води в організмі.

Основою роботи таких пристроїв є метод біоелектричного імпедансного аналізу (BIA), при якому через тіло людини пропускається слабкий електричний струм. Це дає змогу отримати точні дані про співвідношення різних компонентів маси тіла. Для підвищення точності розрахунків використовуються додаткові дані користувача, такі як зріст, вага, вік та стать.

Подібні пристрої є корисним інструментом для людей, які прагнуть контролювати зміни у своєму тілі, зокрема для тих, хто працює над зменшенням жирової маси або нарощуванням м'язів. Серед прикладів таких пристроїв варто

відзначити Withings Body+ Scale – високотехнологічну модель, яка забезпечує вимірювання відсотка жирової та водної складової, а також м'язової та кісткової маси [29].

Ця модель підтримує до восьми індивідуальних профілів, сумісна з мобільними пристроями на базі Android та iOS і може інтегруватися з популярними фітнес-застосунками. Крім того, передбачено різноманітні режими роботи, зокрема для спортсменів, вагітних жінок та дітей, що дозволяє персоналізувати підхід до контролю здоров'я.

2.4 Взаємодія пристрою з мережею через Wi-Fi або Bluetooth

Портативний шлюз DSGW-340 IoT є компактним багатофункціональним пристроєм, який забезпечує передачу медичних даних від пацієнта до хмарної системи. Завдяки підтримці технології Bluetooth 5.2, шлюз легко з'єднується з медичними сенсорами та приладами, а отримані показники відправляє на сервер через Wi-Fi 2.4G або LTE Cat.1 [30].

Цей пристрій орієнтований на безперервний моніторинг життєвих показників та віддалене спостереження за станом здоров'я користувачів. Завдяки своїй мобільності та простоті інтеграції DSGW-340 є зручним рішенням для побудови систем дистанційного контролю здоров'я.

Ключові технічні характеристики шлюзу:

- живлення через порт USB 5V Type-C;
- сучасний процесор BG24;
- підтримка Bluetooth 5.2, LTE Cat.1, Wi-Fi 2.4G;
- можливість роботи від літієвого акумулятора.

Основні функціональні компоненти цього шлюзу та їх взаємозв'язок наочно представлені на рисунку 2.4 у вигляді блок-схеми апаратної архітектури.

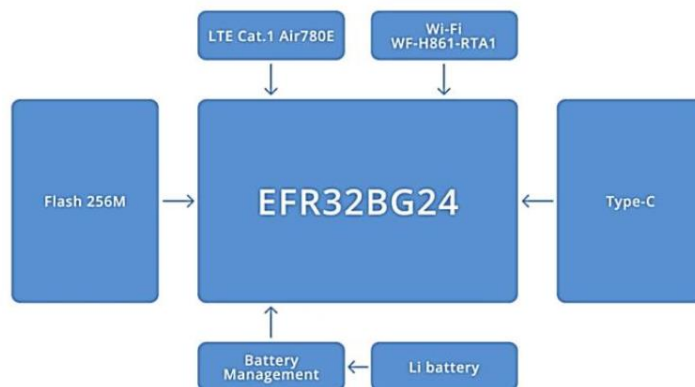


Рисунок 2.4 – Схематичне зображення взаємодії компонентів Bluetooth-шлюзу з Wi-Fi та LTE мережами [30].

Механічні параметри та конструктивні особливості портативного Bluetooth-шлюзу для передачі даних через Wi-Fi/LTE відображено на рисунку 2.5



Рисунок 2.5 – Схематичне зображення механічної конфігурації шлюзу для бездротової передачі даних [30].

Технічні характеристики портативного Bluetooth-шлюзу для Wi-Fi/LTE
Портативний шлюз Bluetooth для Wi-Fi/LTE має низку ключових характеристик, що забезпечують стабільну та надійну роботу в умовах віддаленого збору і передачі даних.

Електроживлення та індикація:

- Вхідна напруга адаптера: 100–240 В (AC), 50–60 Гц.
 - Вихідна напруга: 5 В, струм 1 А, роз'єм USB Type-C [31].
 - Індикатори роботи:
1. Індикатор живлення блимає з частотою 0,2 Гц після увімкнення.

2. BLE-індикатор блимає 2 Гц під час встановлення мережевого з'єднання та 0,5 Гц після його завершення.
3. WLAN та LTE індикатори блимають з частотою 0,5 Гц при відсутності з'єднання.
4. При натисканні аварійної кнопки на 3 секунди всі індикатори блимають 5 Гц протягом 30 секунд.
5. Підсвічування вимкнене за замовчуванням, активується кнопкою живлення і автоматично вимикається через 10 секунд.

Органи керування:

- Кнопка скидання для Wi-Fi та LTE розташована на перемикачах.
- Окрема кнопка для підключення пристрою з функцією скидання протоколів (утримання 5 секунд).
- Кнопка екстреного виклику активується тривалим натисканням протягом 3 секунд.

Основні технічні параметри:

- Тип встановлення: переносна модель.
- SIM-карта: Micro SIM.
- Робоча температура: 0...40 °C.
- Температура зберігання: -40...85 °C.
- Продуктивність передачі даних

Bluetooth Low Energy (BLE):

- Потужність передачі: 8 дБм.
- Радіус дії: до 100 метрів на відкритій місцевості.
- Чутливість прийому:
 - 1) -103 дБм при 125 kbs,
 - 2) -99 дБм при 500 kbs,
 - 3) -96 дБм при 1 Mbit/s,
 - 4) -92 дБм при 2 Mbit/s.
- Частотний діапазон: 2400–2483,5 МГц, зміщення частоти: ± 20 кГц.

LTE Cat.1:

- Підтримувані діапазони:
 - 1) LTE-FDD: B1/B3/B5/B8.
 - 2) LTE-TDD: B34/38/39/40/41.
- Швидкість передачі даних:
 - 1) LTE-FDD: до 10 Mbit/s (вниз) / 5 Mbit/s (вгору).
 - 2) LTE-TDD: до 8.2 Mbit/s (вниз) / 3.4 Mbit/s (вгору).
- Потужність передачі: 23 дБм $\pm$ 2 дБм.
- Чутливість прийому: менше -99 дБм.
- Імпеданс антени: 50 Ом (потребує зовнішньої антени).

Wi-Fi:

- Стандарти: 802.11b/g/n.
- Швидкість передачі даних: до 150 Mbit/s (в режимі 802.11n).
- Підтримка агрегації кадрів: A-MPDU та A-MSDU.
- Інтервал захисту: 0.4 мкс.
- Частотний діапазон: 2412–2484 МГц.

DSM-042 є енергоефективним вбудованим модулем, який базується на високоінтегрованому бездротовому радіопроцесорі EFR32MG21A020F768IM32-B і підтримує технологію Zigbee для стабільної та економної передачі даних [32].

Щодо характеристик шлюзу DSGW-210B, цей пристрій виконує функцію конвертації сигналів із технології LoRaWAN у Ethernet або Wi-Fi, що забезпечує його універсальність для інтеграції з мережею The Things Network та використання у різноманітних промислових рішеннях. Вбудований Ethernet-порт і стандартний режим точки доступу Wi-Fi роблять цей шлюз зручним для широкого спектра IoT-застосувань.

Ще одним елементом системи є маяк-датчик температури та вологості Beacon, створений компанією Dusun, який відправляє актуальні показники температури та вологості в режимі реального часу завдяки використанню технології Beacon.

Серед інших пристроїв вирізняється шлюз роумінгу Bluetooth DSGW-021, який поєднує двосмуговий модем 4G Cat4, процесор MT7628AN із тактовою

частотою 600 МГц та оперативну пам'ять обсягом до 128 МБ. Цей шлюз функціонує під управлінням OpenWrt, що відкриває широкі можливості для налаштування і розширення мережесих рішень [33].

Пристрій підтримує Wi-Fi на частотах 2,4 ГГц і 5 ГГц, а також технології Zigbee 3.0, Bluetooth 5.2 та Z-Wave. Bluetooth-функціонал шлюзу охоплює сканування пристроїв, підтримку GATT, BLE Mesh та BLE роумінг. Завдяки використанню стандарту Wi-Fi IEEE 802.11ac та можливостям LTE 4G, шлюз DSGW-021 (рис. 2.6) забезпечує стабільну і високошвидкісну передачу даних, що дозволяє одночасно підключати кілька пристроїв для ефективного обміну інформацією [34].



Рисунок 2.6 – Шлюз DSGW-021 для роумінгу Bluetooth з функціями бездротової передачі даних через 4G та Wi-Fi [30].

3 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО РЕАЛІЗАЦІЇ ВЕБ-СИСТЕМИ ДЛЯ МОНИТОРИНГУ ЗДОРОВ'Я

3.1 Рекомендації щодо побудови алгоритма обробки даних з IoT-пристрою у веб-середовищі

Для досягнення ефективної роботи запропонованої системи моніторингу стану здоров'я людини було проаналізовано оптимальні підходи до формування алгоритмічного забезпечення процесу збору, обробки та передачі даних з IoT-пристрою. Результатом цього аналізу стала розробка схеми роботи програми, що представлена на рисунку 3.1.

Початковим етапом реалізації алгоритму є ініціалізація апаратних інтерфейсів I2C та UART, а також підключення необхідних бібліотек, які забезпечують функціонування пристроїв і передавання інформації. Зокрема, використовуються:

- Wire.h – для підтримки обміну даними через I2C;
- Protocentral_MAX30205.h – для роботи з температурним сенсором;
- PulseSensorPlayground.h – для обробки сигналів з датчика пульсу;
- WiFi.h – для організації бездротової передачі даних по Wi-Fi;
- ThingSpeak.h – для інтеграції з хмарним IoT-сервером ThingSpeak [35].

До початку основного циклу роботи системи виконується налаштування мережевого з'єднання через IP-протокол, що дозволяє забезпечити безперервну передачу даних.

В основному циклі алгоритму здійснюється опитування датчика температури тіла через I2C та фіксація частоти серцевих скорочень з допомогою ADC-каналу, підключеного до відповідного сенсора. Зібрані показники

відображаються на OLED-дисплеї (через I2C) та дублюються через послідовний порт для оперативного моніторингу в реальному часі.(Рис.3.1).

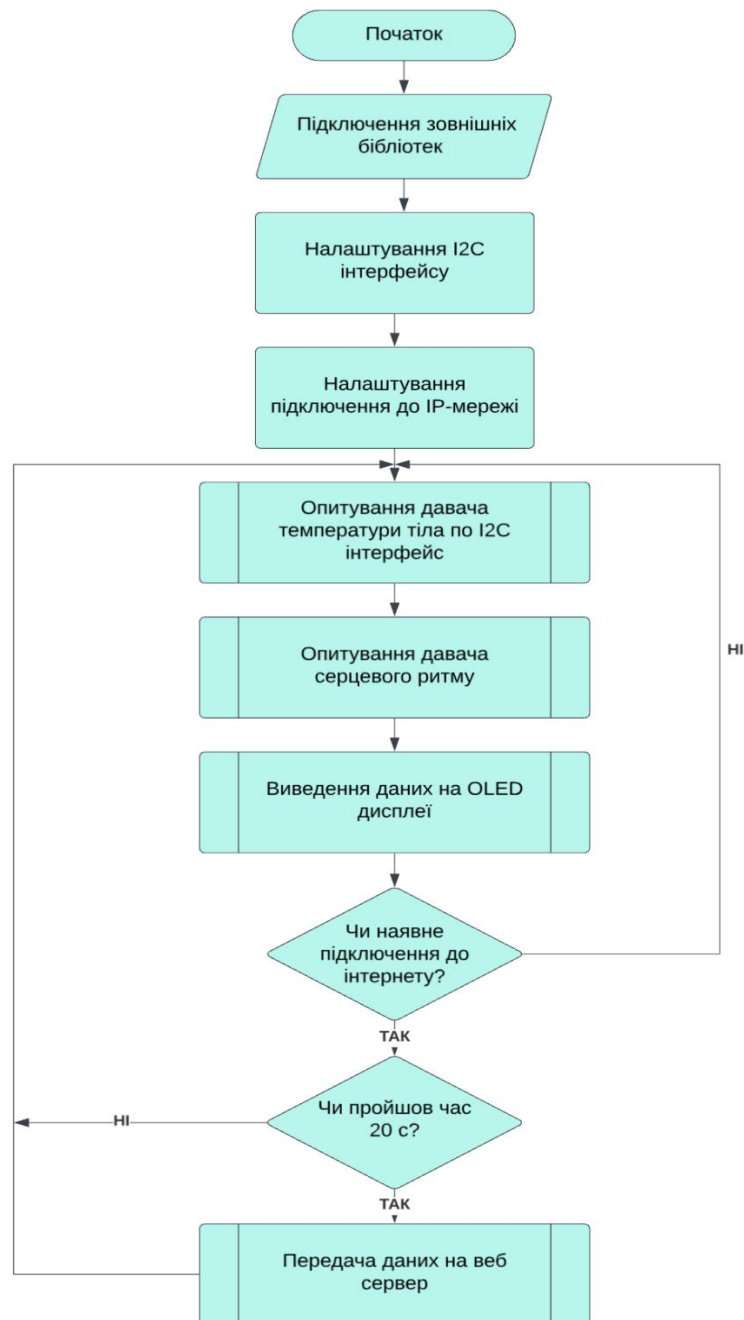


Рисунок 3.1 – Блок-схема програмного алгоритму для системи IoT-моніторингу фізіологічних показників [31].

Перед передачею даних на сервер виконується перевірка наявності підключення до Інтернету. За умови активного з'єднання показники періодично надсилаються до веб-сервера за допомогою REST API-запитів, що дозволяє організувати віддалене збереження та аналіз отриманих даних [36].

3.2 Рекомендації щодо налаштування середовища для програмування мікроконтролера ESP32i

Для створення програмного забезпечення, призначеного для роботи з мікроконтролером ESP32, у межах реалізації проєкту було обрано мову програмування Processing, яка базується на синтаксисі C/C++. Такий вибір обумовлений простотою інтеграції з апаратною частиною, широкою підтримкою бібліотек для роботи з датчиками та мережевими інтерфейсами, а також гнучкістю налаштувань [37].

Як середовище розробки застосовується Arduino IDE, що забезпечує зручний інтерфейс для написання, компіляції та завантаження коду на мікроконтролер. Це середовище є популярним серед розробників IoT-рішень завдяки своїй простоті, доступності та широкій спільноті підтримки.

Зовнішній вигляд вікна Arduino IDE, яке використовується під час розробки програмного забезпечення в даному проєкті, представлено на рисунку 3.2.

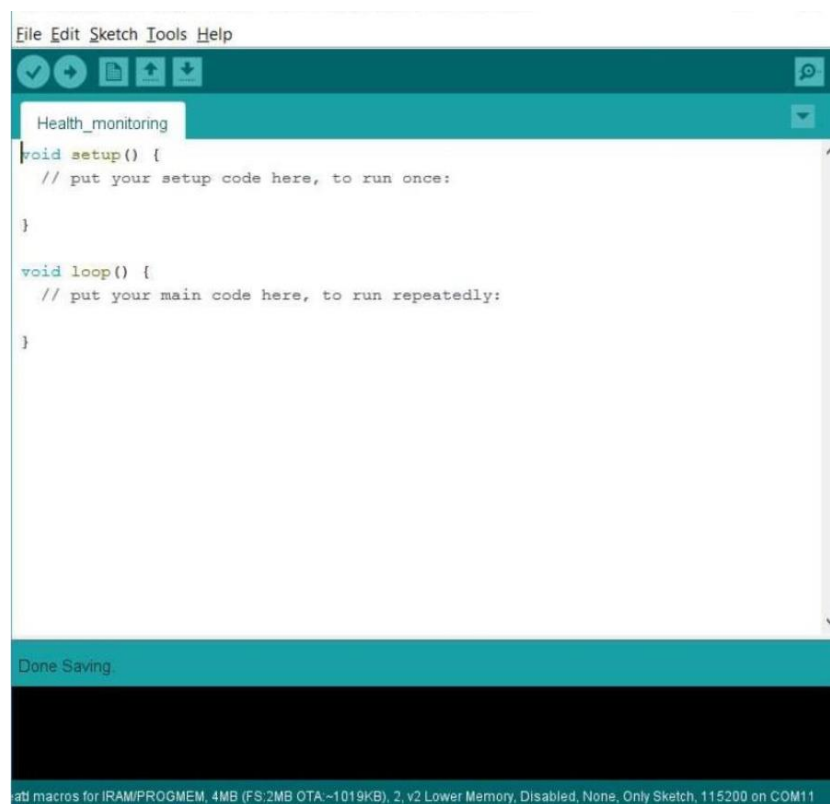


Рисунок 3.2 – Активне вікно інтегрованого середовища розробки Arduino IDE[33].

Arduino IDE є універсальним кросплатформеним середовищем для розробки вбудованого програмного забезпечення, орієнтованим на роботу з мікроконтролерами різних типів, зокрема ESP32, який використовується в даному проєкті. Це безкоштовне рішення відзначається зручністю інтеграції з численними бібліотеками, що значно полегшує процес створення та налагодження програм для IoT-пристроїв [38].

Вибір саме Arduino IDE для розробки обумовлений низкою переваг, серед яких варто виділити:

- невеликі вимоги до обсягу пам'яті;
- низьке споживання ресурсів комп'ютера;
- інтуїтивно зрозумілий інтерфейс, що спрощує роботу навіть для початківців;
- швидке та зручне підключення до апаратної частини;
- повна підтримка операційної системи Windows;
- можливість використовувати різні мови програмування;
- наявність широкого спектра функцій та інструментів, необхідних для розробки вбудованого ПЗ.

Програмний код, створений в Arduino IDE, традиційно називається «скетч».

Перед компіляцією кожен проєкт потребує збереження у вигляді окремого файлу, що дозволяє уникнути помилок під час компіляції та подальшої прошивки мікроконтролера.

Для того щоб мати можливість розробляти програмне забезпечення для мікроконтролера ESP32 безпосередньо в середовищі Arduino IDE, необхідно попередньо встановити відповідний модуль підтримки цієї платформи. Це дає змогу розширити функціональність середовища та забезпечити коректне налаштування для роботи з ESP32 [39].

Процедура встановлення модуля передбачає внесення спеціального посилання на JSON-файл конфігурації до налаштувань Arduino IDE. Для цього потрібно відкрити меню «File» → «Preferences» та вставити відповідне посилання в поле для додаткових менеджерів плат.

Цей етап є необхідним для завантаження актуальних пакетів підтримки ESP32, що забезпечують правильну компіляцію та прошивку мікроконтролера. Вигляд відповідного вікна налаштувань продемонстровано на рисунку 3.3.

`https://dl.espressif.com/dl/package\_esp32\_index.json`

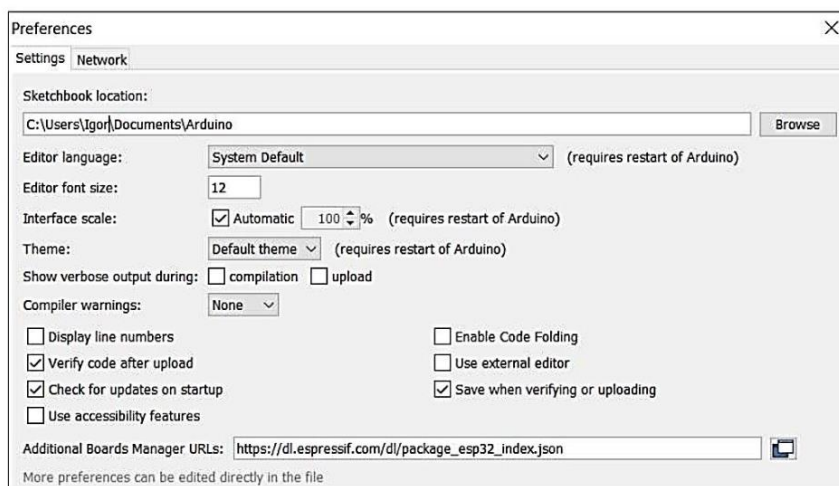


Рисунок 3.3 – Кроки встановлення плати ESP32 в Arduino IDE через конфігураційне посилання JSON[38].

Для повноцінної роботи з мікроконтролером ESP32 в середовищі Arduino IDE після додавання конфігураційного посилання необхідно встановити відповідний пакет підтримки плати. Це здійснюється через вікно «Boards Manager», яке можна відкрити в меню «Tools» → «Board» → «Boards Manager» [40].

У вікні менеджера плат потрібно виконати пошук за ключовим словом "esp32" та встановити знайдений пакет. Цей модуль містить усі необхідні компоненти для правильного налаштування компіляції та завантаження прошивки на ESP32.

Вигляд процесу пошуку та встановлення пакета підтримки для ESP32 в Arduino IDE наведено на рисунку 3.4.

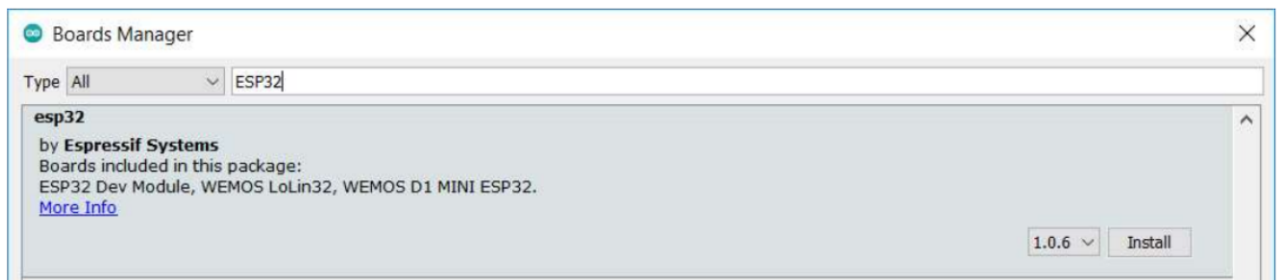


Рисунок 3.4 – Кроки інсталяції пакета підтримки ESP32 через Boards Manager в Arduino IDE [40].

Після успішного встановлення пакета підтримки ESP32 в Arduino IDE користувач отримує можливість обрати необхідний тип мікроконтролера зі списку доступних плат. Для цього слід перейти до меню «Tools» → «Board», де стане доступним пункт «ESP32 Dev Module», який відповідає за налаштування параметрів компіляції та прошивки для даної платформи [41].

Цей вибір є необхідним для коректної роботи з ESP32 та забезпечує правильне визначення конфігурації мікроконтролера. Вигляд меню з обраною платою представлено на рисунку 3.5.

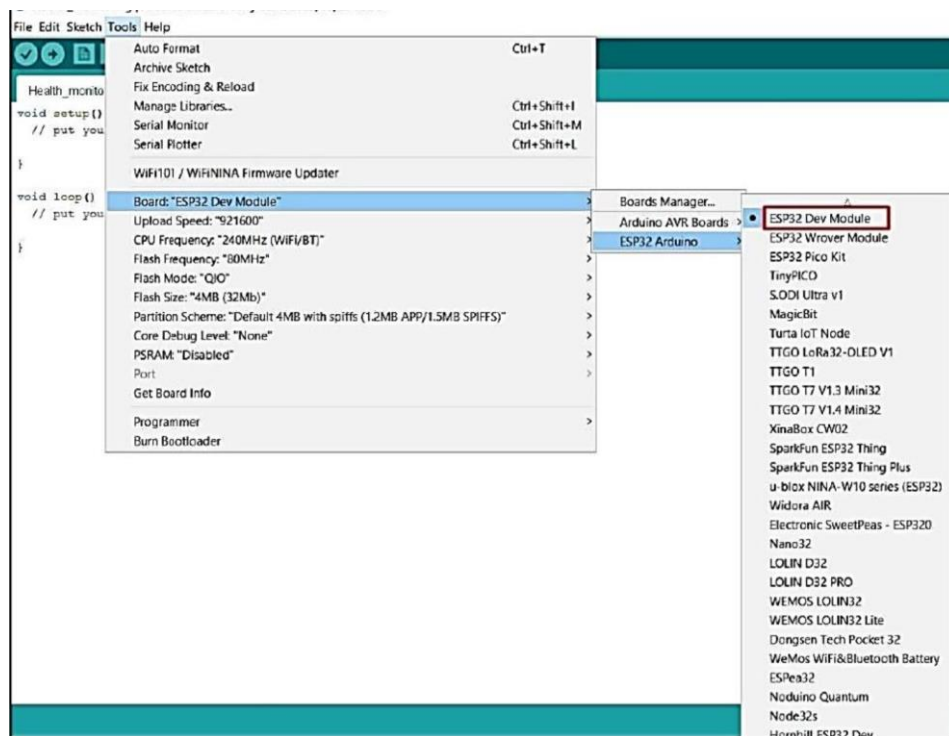


Рисунок 3.5 – Вибір мікроконтролера ESP32 для компіляції та прошивки в Arduino IDE [41].

3.3 Рекомендації щодо підключення зовнішніх бібліотек та написання програмного коду

Бібліотеки в Arduino IDE – це набір готових функцій, класів та структур даних, що дозволяють значно спростити процес програмування, зокрема під час роботи з різними модулями, датчиками та іншими периферійними пристроями. Підключення таких бібліотек до проєкту дає змогу зосередитися на основній логіці роботи програми, не витрачаючи час на реалізацію базових низькорівневих операцій.

Сьогодні існує велика кількість бібліотек, які розповсюджуються вільно та можуть бути завантажені з Інтернету. Кожна бібліотека має власну структуру у вигляді каталогу, що містить необхідні файли для компіляції – заголовочні файли (.h), реалізації (.cpp) та інші допоміжні ресурси.

Під час збирання проєкту Arduino IDE автоматично додає функціонал підключених бібліотек, що значно спрощує роботу розробника. Для підключення бібліотеки достатньо вставити відповідну директиву `#include` на початку коду, зазначивши файл заголовка необхідної бібліотеки [42].

Існує кілька способів встановлення бібліотек у Arduino IDE: вручну, шляхом копіювання файлів у спеціальну директорію, або через вбудований менеджер бібліотек (Library Manager), який доступний у меню «Tools» → «Manage Libraries».

У межах цього проєкту для роботи з датчиком серцевого ритму було використано саме підхід встановлення бібліотеки через Library Manager, що дозволило швидко знайти потрібний пакет за назвою та інтегрувати його до проєкту. Вигляд процесу встановлення бібліотеки продемонстровано на рисунку 3.6.



Рисунок 3.6 – Послідовність підключення зовнішньої бібліотеки PulseSensor у середовищі Arduino IDE [42]

У розробленій системі було використано програмну реалізацію опитування датчика серцевого ритму, що дозволяє отримувати актуальні дані про фізіологічний стан користувача в режимі реального часу. Для організації взаємодії з цим датчиком у програмному коді було підключено спеціалізовану бібліотеку PulseSensorPlayground.h, яка забезпечує всі необхідні функції для коректної роботи сенсора [43].

Підключення цієї бібліотеки виконується на початку програми за допомогою директиви `#include`, що дозволяє використовувати вбудовані методи для зчитування, обробки та аналізу даних серцевого ритму без необхідності самостійної реалізації низькорівневих алгоритмів.

Застосування готової бібліотеки значно спрощує процес розробки, підвищує стабільність роботи системи та дозволяє зосередитися на побудові логіки збору і передачі отриманих даних для подальшого моніторингу.

У програмному коді було створено екземпляр класу PulseSensorPlayground з ім'ям `pulseSensor`, який відповідає за ініціалізацію та подальшу роботу з датчиком серцевого ритму. Цей об'єкт забезпечує доступ до методів бібліотеки, необхідних для налаштування, зчитування та обробки даних пульсу.

У межах функції `setup()` було реалізовано налаштування параметрів для створеного об'єкта `pulseSensor`, що забезпечує коректну взаємодію із сенсором серцевого ритму. На цьому етапі виконується ініціалізація основних режимів роботи датчика, зокрема вказується пін для зчитування сигналу, встановлюються порогові значення та визначаються додаткові параметри обробки.

Ці дії необхідні для підготовки пристрою до стабільної роботи та забезпечення точності зібраних даних. Конфігурація об'єкта у функції `setup()` наведено на рисунку 3.7.

```

1  Serial.begin(115200);
2  // Configure the PulseSensor manager.
3  pulseSensor.analogInput(PULSE_INPUT);
4  pulseSensor.blinkOnPulse(PULSE_BLINK);
5  pulseSensor.fadeOnPulse(PULSE_FADE);
6  pulseSensor.setSerial(Serial);
7  pulseSensor.setOutputType(OUTPUT_TYPE);
8  pulseSensor.setThreshold(THRESHOLD);
9
10 // Skip the first SAMPLES_PER_SERIAL_SAMPLE in the loop().
11 samplesUntilReport = SAMPLES_PER_SERIAL_SAMPLE;
12
13 // Now that everything is ready, start reading the PulseSensor
14 if (pulseSensor.begin()) {
15     Serial.println("We created a pulseSensor Object!");
16 }

```

Рисунок 3.7 – Ініціалізація та конфігурація датчика серцевого ритму за допомогою об'єкта `pulseSensor` [43].

В основному циклі роботи програми реалізовано опитування датчика серцевого ритму з подальшим зчитуванням показників пульсу. Отримані дані з аналогового входу оперативно передаються для відображення через послідовний порт (Serial Monitor), що дозволяє контролювати результати вимірювань у режимі реального часу [44].

Фрагмент коду, що реалізує цей процес у головному циклі програми, представлено на рисунку 3.8.

```

33 if (pulseSensor.sawNewSample()) {
34     if (--samplesUntilReport == (byte) 0) {
35         samplesUntilReport = SAMPLES_PER_SERIAL_SAMPLE;
36         pulseSensor.outputSample();
37         if (pulseSensor.sawStartOfBeat()) {
38             pulseSensor.outputBeat();
39         }
40     }
41 }

```

Рисунок 3.8 – Реалізація опитування датчика пульсу та виведення результатів у Serial Monitor[44].

Сформований сигнал із датчика серцевого ритму, а також результати обробки цих даних можна переглядати безпосередньо через монітор послідовного порту, який доступний у середовищі Arduino IDE. Це дозволяє в реальному часі відстежувати зміни серцевого ритму та контролювати коректність роботи системи [45].

Вигляд вікна відображення результатів на моніторі послідовного порту наведено на рисунку 3.9.

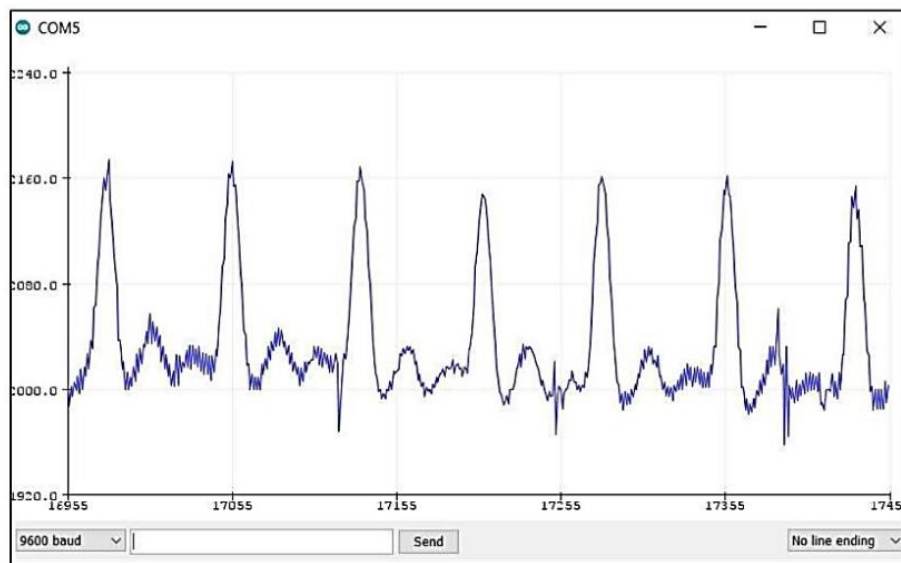


Рисунок 3.9 – Виведення інформації з датчика пульсу на монітор послідовного порту в Arduino IDE[45].

Для організації взаємодії з датчиком температури тіла MAX30205 у проєкті застосовується спеціалізована бібліотека, яка забезпечує необхідний функціонал для зчитування та обробки температурних даних. Підключення цієї бібліотеки до проєкту є обов'язковою умовою для коректної роботи датчика.

Процес додавання відповідної бібліотеки представлено на рисунку 3.10.

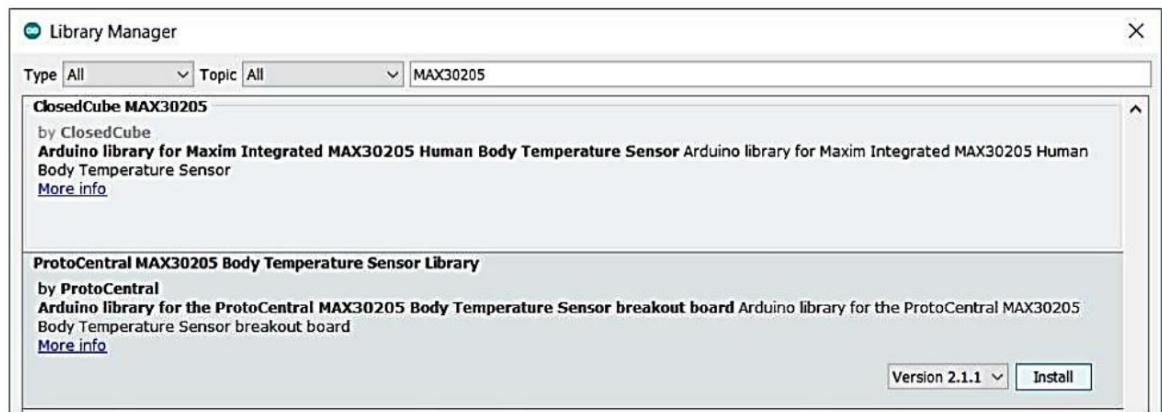


Рисунок 3.10 – Процес встановлення зовнішньої бібліотеки ProtoCentral MAX30205 для інтеграції датчика температури тіла[45].

Для забезпечення коректної роботи системи та можливості отримання даних з датчика температури тіла MAX30205, необхідно попередньо підключити до проекту відповідну бібліотеку. Це дозволяє отримати доступ до функцій, що відповідають за ініціалізацію, налаштування та зчитування показників температури через стандартний інтерфейс взаємодії [46].

Імпорт необхідної бібліотеки є першим етапом реалізації обробки температурних даних у межах розроблюваного програмного забезпечення.

У рамках функції `setup()` реалізовано перевірку наявності з'єднання датчика температури MAX30205 через інтерфейс I^2C . Цей етап дозволяє переконатися, що пристрій правильно підключено до мікроконтролера та готовий до зчитування даних. Наявність коректного підключення є критично важливою умовою для подальшої роботи системи моніторингу. Приклад реалізації цього процесу наведено на рисунку 3.11.

```

48 Serial.begin(9600);
49 Wire.begin();
50 // Scan for temperature in every 30 sec until a sensor is found.
51 while(!tempSensor.scanAvailableSensors()){
52   Serial.println("Couldn't find the temperature sensor.");
53   delay(30000);
54 }

```

Рисунок 3.11 – Реалізація перевірки підключення датчика MAX30205 до мікроконтролера через інтерфейс I2C[46]

У основному циклі програми реалізовано виклик функції зчитування температурних даних від датчика MAX30205. Це дозволяє в реальному часі отримувати актуальні значення температури тіла та передавати їх для подальшої обробки або відображення [47].

Такий підхід забезпечує безперервний моніторинг фізіологічних показників та своєчасне оновлення інформації, необхідної для аналізу стану користувача.

У основному циклі програми здійснюється запуск функції зчитування температури тіла шляхом виклику відповідного методу, який отримує актуальні дані від підключеного датчика (Рис.3.12)

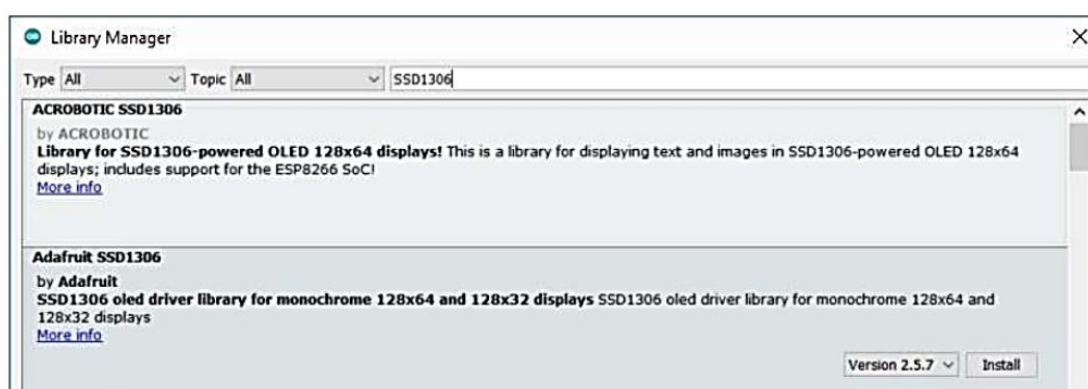


Рисунок 3.12 – Процес встановлення бібліотеки Adafruit SSD1306 для конфігурації OLED дисплея [46]

Для реалізації відображення результатів моніторингу на OLED-дисплеї в проєкті використовується бібліотека Adafruit SSD1306, яка забезпечує зручний інтерфейс для виведення графічної та текстової інформації на екран. Додавання та налаштування цієї бібліотеки представлено на рисунку 3.12.

У рамках програмної реалізації було застосовано бібліотеку Wire.h, яка відповідає за обмін даними через інтерфейс I2C, а також Adafruit_SSD1306.h, що використовується для керування OLED-дисплеєм та організації виведення інформації на екран. Таке поєднання дозволяє ефективно взаємодіяти з апаратними компонентами системи та забезпечує зручне відображення результатів моніторингу.

У програмному коді визначаються параметри розмірів OLED-дисплея, зокрема встановлюється роздільна здатність екрана 128×64 пікселі. Це необхідно

для правильного налаштування області виведення та коректної роботи зображення на екрані.

У програмному коді ініціалізовано екземпляр об'єкта для роботи з OLED-дисплеєм, що дозволяє виконувати всі необхідні операції з відображення інформації на екрані.

У межах функції `setup()` здійснено ініціалізацію та базове налаштування OLED-дисплея, що включає запуск пристрою, перевірку його доступності та підготовку до відображення інформації.

Кодова реалізація функції `loop()`, що відповідає за відображення актуальних даних серцевого ритму на OLED-дисплеї, наведена на рисунку 3.13.

```

76- if (x > 127) {
77   display.clearDisplay();
78   x = 0;
79   lastx = x;
80 }
81 ThisTime = millis();
82 Signal = pulseSensor.getBeatsPerMunite();
83 display.setTextColor(SSD1306_BLACK, SSD1306_WHITE);
84 int y = 60 - (Signal / 16); // 4096 divide by 64
85 display.drawLine(lastx, lasty, x, y, SSD1306_WHITE);
86 lasty = y;
87 lastx = x;
88
89- if (Signal > UpperThreshold) {
90-   if (BeatComplete) {
91     BPM = ThisTime - LastTime;
92     BPM = int(60 / (float(BPM) / 1000));
93     BPMTiming = false;
94     BeatComplete = false;
95   }
96-   if (BPMTiming == false) {
97     LastTime = millis();
98     BPMTiming = true;
99   }
100 }
101
102- if ((Signal < LowerThreshold) && (BPMTiming)) {
103   BeatComplete = true;
104   display.setCursor(0, 50);
105   display.print("BPM = ");
106   display.println(BPM);
107   display.display();
108   x++;
109 }

```

Рисунок 3.13 – Програмна реалізація виведення результатів моніторингу серцевого ритму на OLED-дисплей [45].

У даному циклі здійснюється безперервне зчитування показників пульсу та їх подальше виведення на екран у зручному для користувача вигляді. Результат візуалізації цих даних на OLED-дисплеї продемонстровано на рисунку 3.14.



Рисунок 3.14 – Відображення на OLED-дисплеї даних щодо частоти серцевого ритму [45].

3.4 Рекомендації щодо розгортання та тестування прототипу платформи моніторингу менації щодо покращення безпеки IoT-систем

У процесі організації передачі та збереження результатів моніторингу показників здоров'я, таких як частота серцевого ритму та температура тіла, було обрано хмарну IoT-платформу ThingSpeak. Це рішення забезпечує не лише віддалене зберігання даних, а й їх подальший аналіз та обробку завдяки інтегрованим можливостям мови програмування Matlab, що є однією з переваг цієї платформи.

Особливу увагу було приділено можливості гнучкого налаштування каналів для прийому показників від IoT-пристроїв та візуалізації отриманих результатів у вигляді графіків і таблиць. Додатковим фактором на користь ThingSpeak стала наявність безкоштовного тарифного плану, що дозволяє використовувати сервіс у некомерційних цілях без додаткових витрат.

Схематичне зображення принципу роботи ThingSpeak у контексті цього проєкту наведено на рисунку 3.15. Платформа виступає посередником між IoT-пристроєм, який збирає медичні показники, та користувачем, що має змогу переглядати та аналізувати дані в зручному веб-інтерфейсі.

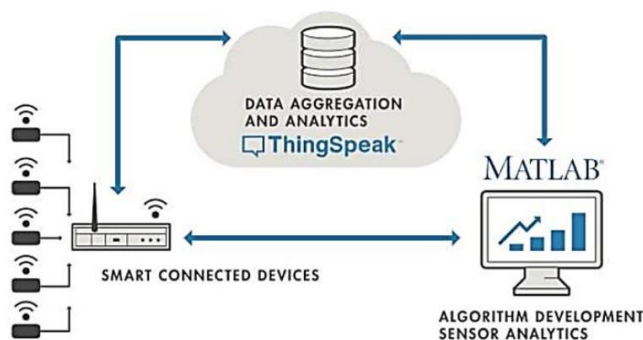


Рисунок 3.15 – Принцип функціонування IoT-платформи ThingSpeak для дистанційного контролю медичних параметрів[45].

У сучасних умовах розвитку інформаційних технологій системи збору та моніторингу даних відіграють ключову роль у вирішенні завдань різних галузей, зокрема в медичній сфері. Одним із перспективних напрямів є застосування систем віддаленого моніторингу здоров'я, побудованих на основі концепцій Інтернету речей (IoT), що дозволяє організувати ефективний контроль фізіологічних показників людини.

У межах цієї роботи запропоновано впровадження системи віддаленого моніторингу стану здоров'я людини, яка використовує мікроконтролер ESP32 спільно з хмарною IoT-платформою ThingSpeak. Основна увага приділена створенню каналу в ThingSpeak для збору, збереження та візуалізації даних із датчиків пульсу та температури тіла, а також організації передачі цих показників у режимі реального часу.

Платформа ThingSpeak виступає ефективним інструментом для збору та подальшої обробки даних від IoT-пристроїв, зокрема від різноманітних сенсорів (температури, вологості, тиску тощо). Її функціональність дозволяє інтегрувати зовнішні пристрої з хмарним середовищем для централізованого зберігання даних і їх аналізу.

Однією з важливих переваг ThingSpeak є можливість використання вбудованого інструментарію для аналізу даних на основі Matlab, що значно розширює можливості обробки інформації без необхідності експорту до сторонніх

середовищ. Це робить платформу універсальним рішенням для організації моніторингових систем різного рівня складності.

Ключовим елементом платформи ThingSpeak є канал, який слугує основним середовищем для зберігання та організації даних. У межах такого каналу можуть бути налаштовані поля для запису показників, а також вказано додаткову інформацію, зокрема місцезнаходження пристрою та статус системи.

Після створення каналу ThingSpeak надає можливість не лише зберігати дані, але й виконувати їх подальшу обробку, візуалізацію та аналіз за допомогою інструментів MATLAB. Крім цього, платформа підтримує автоматичну реакцію на певні події через надсилання сповіщень, твітів або HTTP-запитів, що дозволяє оперативно реагувати на зміну параметрів моніторингу.

Стандартний сценарій роботи з ThingSpeak охоплює кілька основних етапів [48]:

- створення каналу та налаштування полів для збору даних;
- відправлення даних від IoT-пристроїв до каналу;
- аналіз, обробка та візуалізація отриманої інформації за допомогою доступних інструментів.

У межах цього проєкту для прикладу демонструється зчитування температури та вологості за допомогою датчика DHT11, підключеного через модуль NodeMCU. Отримані показники в автоматичному режимі передаються до каналу ThingSpeak, що дозволяє зручно спостерігати за станом навколишнього середовища через Інтернет із будь-якої точки світу (Рис.3.16)

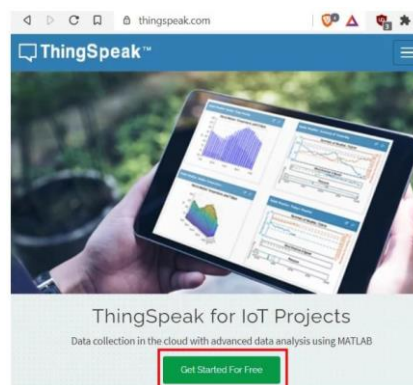


Рисунок 3.16 – Початковий екран IoT-платформи ThingSpeak для налаштування моніторингу [45].

Для завершення процедури реєстрації на платформі необхідно вказати відповідну інформацію в реєстраційній формі та підтвердити дію, натиснувши кнопку «Зареєструватися». Це дозволить створити обліковий запис для подальшого налаштування системи моніторингу (Рис.3.17).



Рисунок 3.17 – Діалогове вікно створення облікового запису на ThingSpeak[45]

Після авторизації в системі ThingSpeak за допомогою облікового запису ThingSpeak або MathWorks, для створення нового каналу збору даних необхідно виконати кілька послідовних дій:

- увійти до особистого кабінету платформи;
- перейти до розділу «Мої канали»;
- обрати опцію «Новий канал» для запуску процесу створення та налаштування нового каналу моніторингу.

Цей етап є початковим для організації зберігання та візуалізації даних, які надходять від IoT-пристроїв (Рис.3.18).

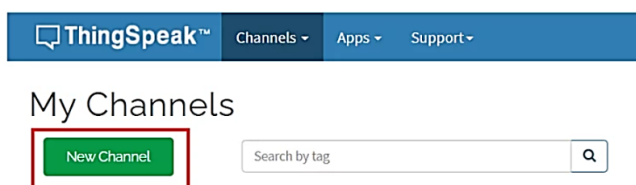


Рисунок 3.18 – Діалогове вікно налаштування параметрів нового каналу в ThingSpeak

У межах реалізації даного проєкту було створено та налаштовано окремий канал у системі ThingSpeak під назвою «Health Monitoring System», який призначений для збору та обробки медичних даних. Для організації моніторингу в канал були додані поля зі змінними: «Pulse Rate» для фіксації частоти серцевих скорочень та «Body Temperature» для реєстрації температури тіла.

Процес налаштування та створення каналу демонструється на рисунку 3.19.

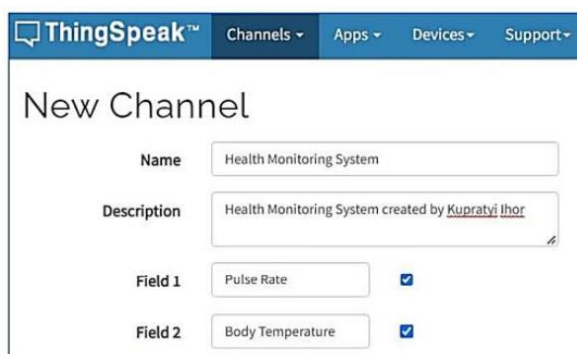


Рисунок 3.19 – Послідовність дій під час створення каналу в ThingSpeak для збору медичних даних [45]

Для організації збору даних з датчика DHT11, який використовується для вимірювання температури та вологості, під час створення каналу в ThingSpeak необхідно задати відповідну назву каналу, наприклад, «Вимірювання» (або іншу, згідно з потребами проєкту). За бажанням можна також додати короткий опис каналу, натиснувши кнопку «Додайте опис».

Далі слід визначити кількість полів, які будуть використовуватись для збереження даних. Для прикладу:

- Поле 1 – температура;
- Поле 2 – вологість.

У разі використання датчика BME280, який додатково підтримує вимірювання атмосферного тиску, можна налаштувати три поля: температура, тиск та вологість. При цьому кількість полів може бути змінена відповідно до конкретних потреб системи моніторингу.

Після збереження налаштувань каналу користувач отримує доступ до п'яти основних вкладок:

- Приватний перегляд;
- Публічний перегляд;
- Налаштування каналу;
- Ключі API;
- Імпорт/експорт даних.

Це дає можливість гнучко керувати каналом та обирати зручний спосіб перегляду й обміну інформацією (Рис.3.20).

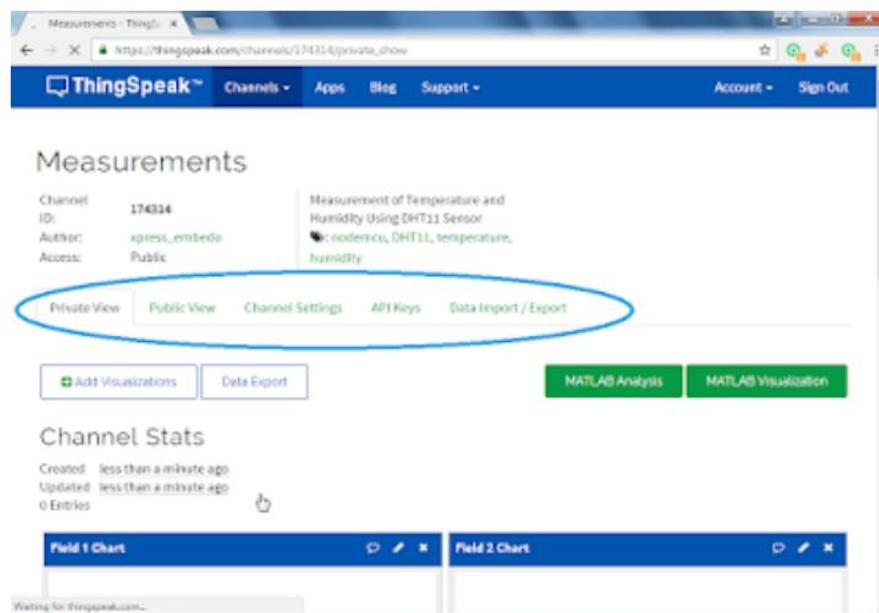


Рисунок 3.20 – Інтерфейс вкладок каналу в ThingSpeak після створення та налаштування[45].

У ThingSpeak передбачена можливість налаштовувати відображення графіків для кожного поля каналу. Користувач може змінювати заголовок графіка, підписи осей X та Y, а також коригувати масштаб відображення даних як для Поле 1, так і для Поле 2. Ці параметри можуть бути налаштовані окремо для приватного та публічного перегляду каналу.

Доступні варіанти редагування параметрів візуалізації представлено на рисунку 3.21.

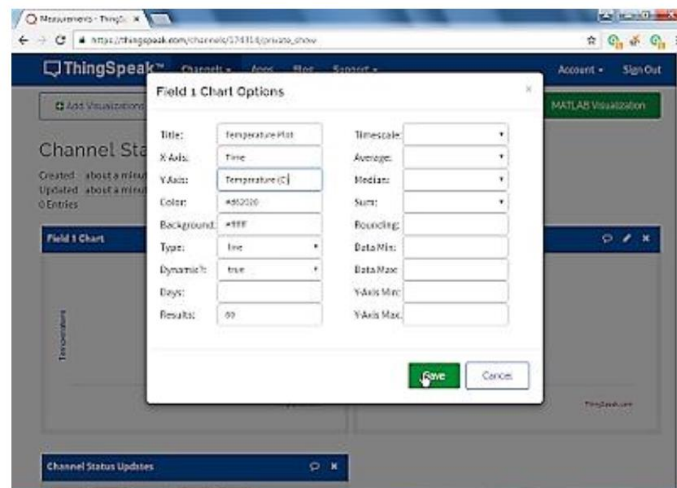


Рисунок 3.21 – Налаштування відображення даних у ThingSpeak: заголовок графіка, осі та масштабування [45]

Після завершення налаштування каналу в ThingSpeak необхідно перейти до фінального етапу – оновлення програмного коду в Arduino IDE для модуля NodeMCU. Для цього потрібно скопіювати ідентифікатор каналу (Channel ID) та ключ API для запису даних (Write API Key), які можна знайти у вкладці «API Keys» в налаштуваннях створеного каналу.

Ці параметри необхідно вставити у відповідні змінні прошивки, що дозволить організувати передачу даних від пристрою до ThingSpeak. Без коректного внесення цих даних комунікація між NodeMCU та хмарною платформою буде неможливою.

Приклад внесення Channel ID та Write API Key у програмний код NodeMCU наведено на рисунку 3.22.

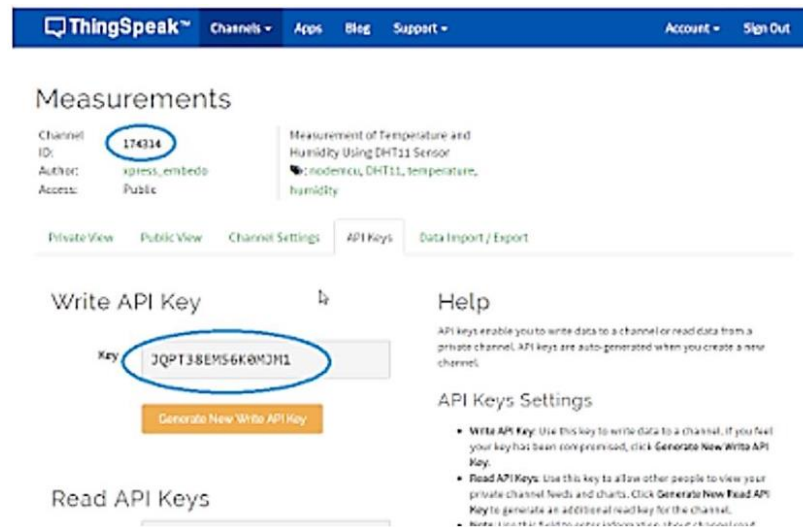


Рисунок 3.22 – Отримання ідентифікатора каналу та ключа API для налаштування зв'язку з платформою ThingSpeak [45]

Після завершення налаштування проєкту та внесення необхідних змін до програмного коду виконується прошивка модуля NodeMCU з оновленою версією прошивки. Це дозволяє розпочати передачу даних із датчиків до створеного каналу на платформі ThingSpeak.

У результаті можна спостерігати графічне відображення отриманих значень температури та вологості безпосередньо в інтерфейсі ThingSpeak. Приклади візуалізації цих даних подано на рисунку 3.23 та рисунку 3.24.

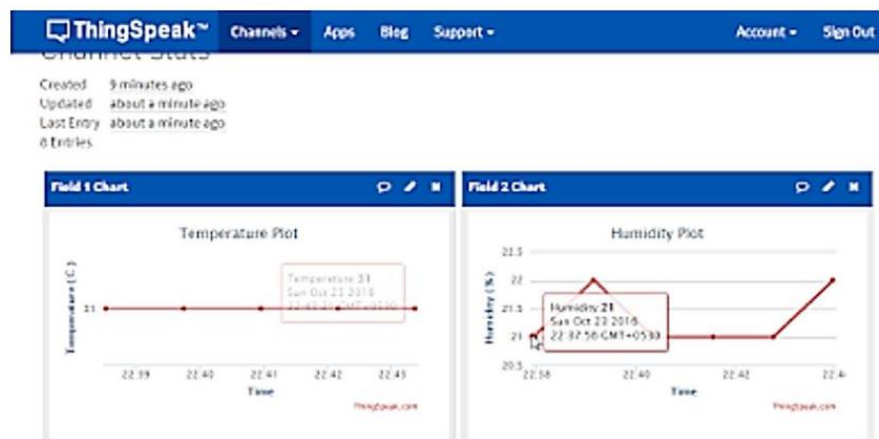


Рисунок 3.23 – Графіки температури та вологості в ThingSpeak через кілька хвилин після активації системи [45]

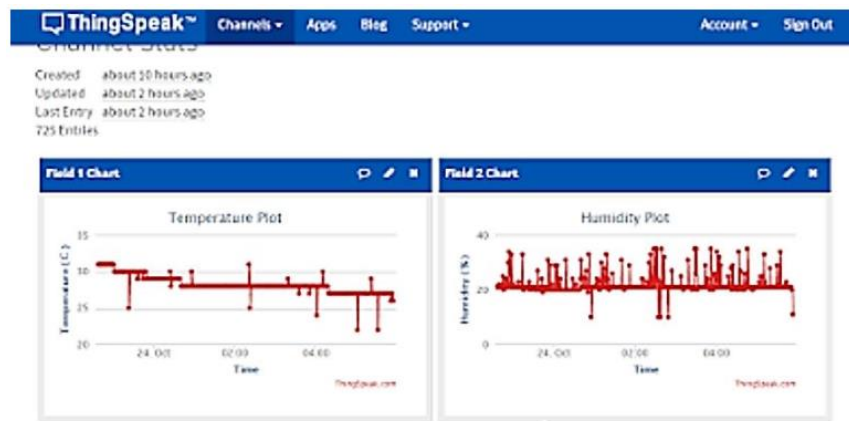


Рисунок 3.24 – Графіки температури та вологості в ThingSpeak через 8 годин моніторингу [45]

На наступному етапі реалізації необхідно додати до проєкту бібліотеку ThingSpeak, яка дозволяє організувати передачу зібраних даних із мікроконтролера NodeMCU до створеного каналу на хмарній платформі ThingSpeak.

Нижче представлено приклад програмного коду, розробленого в середовищі Arduino IDE, який реалізує процес зчитування показників та їх відправку до ThingSpeak (Додаток А).

Далі буде розглянуто послідовність кроків для налаштування мобільного застосунку ThingView, який дозволяє переглядати дані з каналу ThingSpeak на смартфоні під керуванням операційної системи Android. Цей застосунок забезпечує зручний доступ до результатів моніторингу в режимі реального часу

Після запуску мобільного застосунку ThingView (див. рисунок 3.25) необхідно натиснути кнопку «Додати» для створення нового підключення до каналу ThingSpeak. Цей крок є стартовим для налаштування перегляду даних із вашого IoT-пристрою в мобільному додатку.

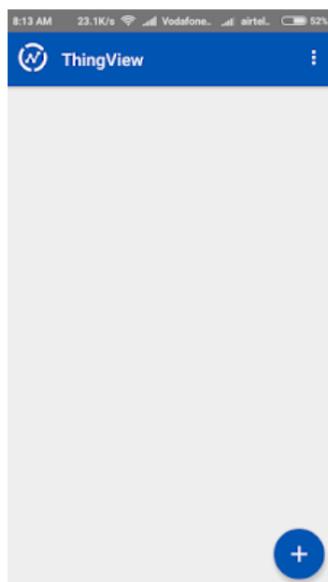


Рисунок 3.25 – Мобільний додаток ThingView для моніторингу показників IoT-пристроїв [46]

Розташування та вигляд цієї кнопки представлено на рисунку 3.26.

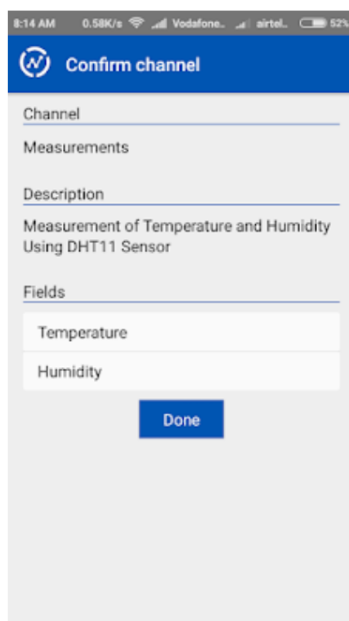


Рисунок 3.26 – Екран налаштувань підключення каналу в мобільному додатку ThingView[46]

Після натискання кнопки «Додати» в мобільному застосунку ThingView потрібно ввести ідентифікатор каналу (Channel ID), після чого здійснити пошук і вибрати потрібний канал зі списку знайдених. Це дозволяє зв'язати застосунок із

відповідним каналом для відображення актуальних даних моніторингу. Процес показано на рисунку 3.27.

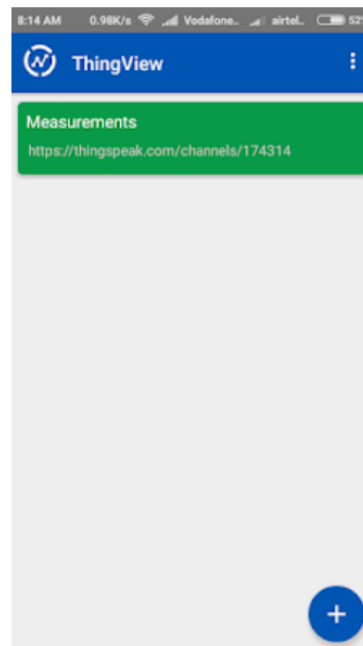


Рисунок 3.27 – Налаштування підключення до каналу через ідентифікатор у додатку ThingView [46]

Після успішного додавання каналу в мобільному застосунку ThingView з'являється можливість переглядати графіки та отримані дані безпосередньо на екрані смартфона. Це дозволяє стежити за змінами параметрів моніторингу в режимі реального часу, не використовуючи ПК чи браузер.

Відображення графіків у ThingView показано на рисунку 3.28.

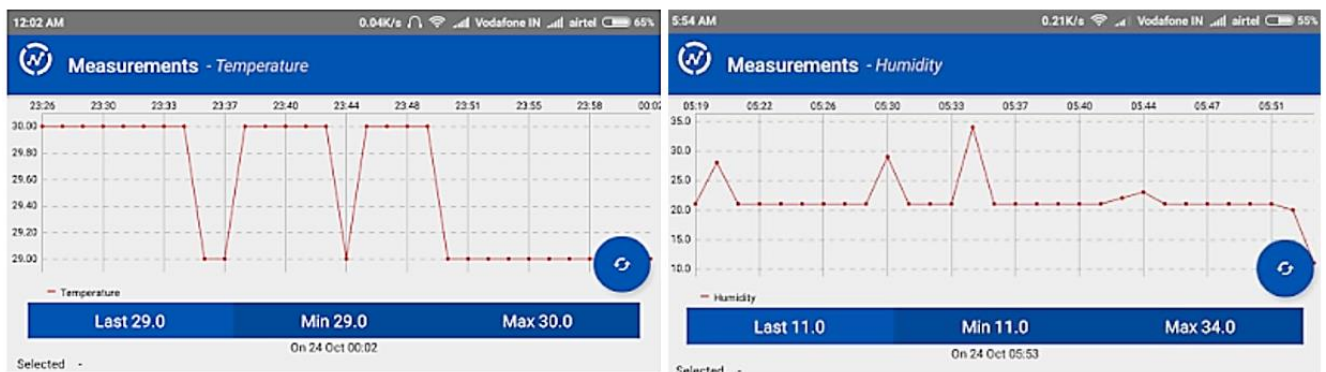


Рисунок 3.28 – Перегляд графіків температури та вологості в ThingView на смартфоні [46]

Передача поточкових даних у системі ThingSpeak може здійснюватися в режимі реального часу між каналами за допомогою таких протоколів, як MQTT або REST API. Це забезпечує швидку та стабільну передачу інформації від IoT-пристроїв до хмарної платформи для подальшої обробки та аналізу.

У межах безкоштовного тарифного плану користувач може створити до чотирьох каналів у своєму обліковому записі. Для кожного каналу доступно до восьми полів для зберігання різних типів даних, таких як температура, вологість, тиск тощо. Канали можуть бути налаштовані як публічні (доступні для всіх) або приватні (з обмеженим доступом).

Створення каналів і відображення їх списку наведено на рисунку 3.29.

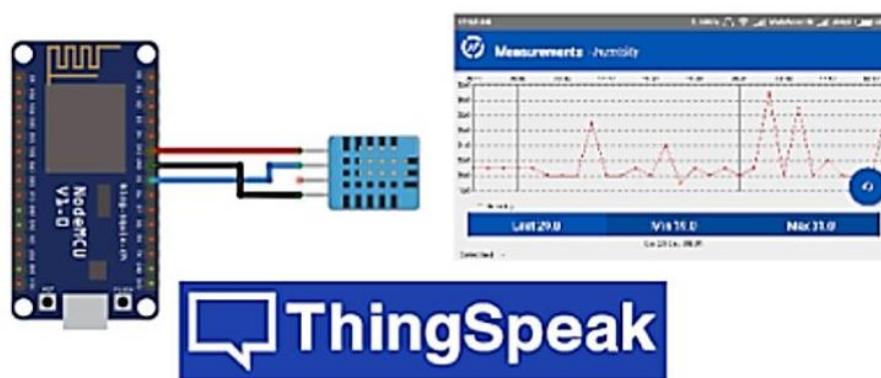


Рисунок 3.29 – Схематичне зображення передачі даних у ThingSpeak з використанням MQTT та REST API [46]

Для реалізації передачі даних до платформи ThingSpeak в межах цього проєкту використовується мікроконтролер ESP32, який забезпечує стабільне з'єднання з хмарною системою. Оптимальним рішенням для організації такого підключення є застосування функціоналу зовнішньої бібліотеки ThingSpeak, яка містить готові методи для надсилання та обробки даних. Підключення цієї бібліотеки в Arduino IDE наведено на рисунку 3.30.



Рисунок 3.30 – Інтеграція бібліотеки ThingSpeak для налаштування передачі даних у мікроконтролері ESP32 [46]

На початку програмного коду в Arduino IDE для модуля ESP32 необхідно вимкнути апаратні переривання, оскільки цей мікроконтролер не підтримує їх використання під час роботи з певними датчиками та бібліотеками Arduino. Це є важливою умовою для коректної взаємодії пристрою з периферійними компонентами системи.

Крім цього, на старті програми обов'язково потрібно задати параметри підключення до Wi-Fi мережі – вказати назву мережі (SSID) та пароль. Також необхідно внести налаштування каналу на ThingSpeak, а саме: Channel ID та API ключ для запису (Write API Key).

Ці параметри дозволяють ESP32 передавати зібрані дані до відповідного каналу для подальшого зберігання та аналізу. Процес внесення цих налаштувань показано у Додатку Б. (Кодовий лістинг з параметрами підключення до платформи ThingSpeak через бездротову мережу Wi-Fi).

Для організації відправки даних про частоту серцевого ритму та температуру тіла на хмарну платформу ThingSpeak розроблено відповідний код, представлений у Додатку В.

Передача інформації здійснюється з використанням функції `getBeatsPerMinute()` для визначення показників пульсу та `getTemperature()` для зчитування температури тіла.

Для забезпечення коректності та фільтрації даних у коді реалізовано перевірку вхідних значень за допомогою умовного оператора `if`, який дозволяє відправляти лише ті значення, що потрапляють у встановлені межі: від 40 до 150

ударів на хвилину для пульсу та від 35 до 41 градуса Цельсія для температури. Це дозволяє виключити з передачі випадкові або некоректні результати, тим самим підвищуючи точність моніторингу фізіологічних параметрів (Рис.3.31)



Рисунок 3.31 – Візуалізація отриманих показників здоров'я в "Private View" каналу ThingSpeak[46]

Після авторизації в персональному обліковому записі на платформі ThingSpeak користувач отримує можливість переглядати графічне відображення переданих даних у розділі «Private View». Це дозволяє контролювати актуальні результати моніторингу параметрів здоров'я в зручному візуальному форматі.

Залежно від поставлених завдань та рівня доступу, адміністратор каналу може налаштувати параметри конфіденційності – відкрити доступ до зібраної інформації для вибраного кола осіб або зробити її публічною. Такі налаштування здійснюються через вкладку «Sharing».

Крім того, інструмент «Data Import/Export» надає змогу експортувати зібрані дані у вигляді файлу для подальшої обробки або ж імпортувати необхідну інформацію у вже створений канал, що розширює можливості інтеграції даних у різні аналітичні системи (Рис.3.32)

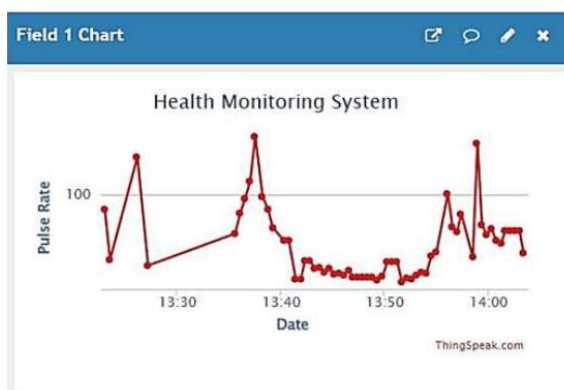


Рисунок 3.32 – Графічне представлення результатів вимірювання серцевого ритму в інтерфейсі ThingSpeak[46]

Платформа ThingSpeak є ефективним інструментом для організації збору, зберігання та візуалізації даних від IoT-пристроїв навіть у межах безкоштовного тарифного плану. Це дає можливість легко створювати прості системи Інтернету речей, які можуть бути використані, зокрема, для реалізації базових елементів концепції «Розумного будинку».

У межах третього розділу, присвяченого темі «Програмне забезпечення IoT в галузі охорони здоров'я», проведено аналіз та реалізацію програмної частини системи віддаленого моніторингу фізіологічного стану людини. Представлено розроблену блок-схему алгоритму функціонування програми, обґрунтовано вибір необхідних інструментів для створення проєкту та здійснено налаштування середовища розробки для модуля ESP32 з підключенням зовнішніх бібліотек.

Окремо описано процес створення програмного забезпечення, що включає зчитування показників з датчиків пульсу та температури тіла, а також їх відображення на OLED-дисплеї для локального моніторингу.

Також реалізовано віддалену передачу отриманих даних на хмарну платформу ThingSpeak, що дозволяє контролювати та аналізувати показники в режимі реального часу. Детально розглянуто процес створення та налаштування каналу в ThingSpeak, а також програмну логіку відправки даних з модуля ESP32 для подальшого відображення результатів моніторингу.

ВИСНОВКИ

У даній роботі було розглянуто можливості використання технологій Інтернету речей (IoT) для створення ефективних систем віддаленого моніторингу стану здоров'я людини. Було проаналізовано актуальні підходи до впровадження IoT-рішень у медичну сферу та побутові проекти, що дозволяє автоматизувати процеси збору, зберігання та обробки фізіологічних даних.

В межах дослідження проаналізовано сучасні методи дистанційного контролю життєвих показників, такі як частота серцевих скорочень, артеріальний тиск, температура тіла та інші параметри, що мають вирішальне значення для своєчасної діагностики та оцінки стану здоров'я користувачів.

Окрему увагу було приділено аналізу апаратної складової IoT-рішень, зокрема можливостей застосування контролера ESP32, типів датчиків, модулів зв'язку та джерел живлення. Вивчено схему побудови IoT-пристроїв для зчитування фізіологічних показників та розглянуто принципи їхньої взаємодії з мережею через Wi-Fi та Bluetooth.

У процесі роботи виокремлено рекомендації щодо побудови алгоритму обробки даних від IoT-пристроїв у веб-середовищі, що включають налаштування середовища для програмування, підключення необхідних бібліотек та написання програмного коду для мікроконтролера ESP32.

Було розроблено рекомендації щодо розгортання прототипу системи моніторингу здоров'я, а також окреслено можливості тестування роботи IoT-платформи для перевірки її функціональності та стабільності.

У завершенні роботи виокремлено ключові аспекти безпеки IoT-систем, що дозволяє забезпечити конфіденційність та цілісність передаваних даних, а також мінімізувати потенційні ризики несанкціонованого доступу до медичної інформації.

Результати дослідження. У результаті дослідження розроблено прототип веб-додатку для моніторингу фізіологічних показників користувача, який поєднує

можливості IoT-пристрою на базі ESP32 та механізму Webhook. Система забезпечує збір, обробку та візуалізацію даних у режимі реального часу, а також реалізує функціонал миттєвих сповіщень у разі виявлення критичних змін параметрів. Запропонована модель підвищує ефективність моніторингу здоров'я та відкриває перспективи для подальшого використання в системах домашньої медицини та телемедицинських платформах.

Наукова новизна. Новизна полягає у поєднанні IoT-архітектури з механізмом миттєвого сповіщення Webhook, що дозволяє реалізувати високоефективну систему моніторингу для медичних і побутових задач.

Практична значущість результатів дослідження – результати дослідження можуть бути використані при розробці систем домашнього нагляду за станом здоров'я, у телемедицинських рішеннях, а також у спеціалізованих медичних установах для віддаленої діагностики.

Апробація результатів надається в 1 тезисах в рецензованих наукових журналах і виданнях.

Матеріали були опубліковані:

В тезисах:

1. Тимченко В.В. СТВОРЕННЯ ВЕБ-ДОДАТКУ МОНІТОРИНГ СТАНУ ЗДОРОВ'Я ПІД ЧАС ІНТЕГРАЦІЇ ІОТ ТА МЕДИЧНИХ ПРИСТРОЇВ// VI Науково-технічна конференція «Сучасний стан та перспективи розвитку IoT». Збірник тез 15 квітня 2025. – К.: ДУІКТ, https://duikt.edu.ua/ua/news-1-570-13858-zaproshennya-na-vi-mizhnarodnu-naukovo-tehnichnu-konferenciyu-suchasniy-stand-ta-perspektivi-rozvitku-iot_kafedra-informaciynih-sistem-ta-tehnologiy

ПЕРЕЛІК ПОСИЛАНЬ

1. Espressif Systems. ESP32 Technical Reference Manual [Електронний ресурс]. – Режим доступу: <https://www.espressif.com/en/support/documents/technical-documents>. – Назва з екрана. – Дата звернення: 25.04.2025.
2. Espressif Systems. ESP-IDF Programming Guide [Електронний ресурс]. – Режим доступу: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/>. – Назва з екрана. – Дата звернення: 25.04.2025.
3. Google Developers. Firebase Documentation [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs>. – Назва з екрана. – Дата звернення: 25.04.2025.
4. Node.js Foundation. Node.js Documentation [Електронний ресурс]. – Режим доступу: <https://nodejs.org/en/docs/>. – Назва з екрана. – Дата звернення: 25.04.2025.
5. React.js Community. React Official Documentation [Електронний ресурс]. – Режим доступу: <https://react.dev/learn>. – Назва з екрана. – Дата звернення: 25.04.2025.
6. Всесвітня організація охорони здоров'я. Global Strategy on Digital Health 2020–2025 [Електронний ресурс]. – Режим доступу: <https://www.who.int/docs/default-source/documents/gsh4dh.pdf>. – Назва з екрана. – Дата звернення: 25.04.2025.
7. Al-Sarawi S., Anbar M., Alieyan K., Alzubaidi M. Internet of Things (IoT) communication protocols: Review // 2017 8th International Conference on Information Technology (ICIT). – 2017. – P. 685–690. – DOI: 10.1109/ICITECH.2017.8079928.
8. Islam S. M. R., Kwak D., Kabir M. H., Hossain M., Kwak K. S. The Internet of Things for health care: A comprehensive survey // IEEE Access. – 2015. – Vol. 3. – P. 678–708. – DOI: 10.1109/ACCESS.2015.2437951.

9. Sethi P., Sarangi S. R. Internet of Things: Architectures, protocols, and applications // Journal of Electrical and Computer Engineering. – 2017. – Article ID 9324035. – DOI: 10.1155/2017/9324035.
10. Zhou L., Wang Y., Song Y. MQTT protocol and its application in IoT systems // International Journal of Future Generation Communication and Networking. – 2019. – Vol. 12, No. 1. – P. 47–58.
11. Chatterjee S., Kar A. K. Regulation and healthcare IoT: Security and privacy perspectives // Health Policy and Technology. – 2018. – Vol. 7, No. 3. – P. 264–275. – DOI: 10.1016/j.hlpt.2018.04.010.
12. Farahani B., Firouzi F., Chakrabarty K. Healthcare IoT: Benefits, challenges, and opportunities // IEEE Internet of Things Magazine. – 2021. – Vol. 4, No. 2. – P. 37–41. – DOI: 10.1109/IOTM.001.2100038.
13. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions // Future Generation Computer Systems. – 2013. – Vol. 29, No. 7. – P. 1645–1660. – DOI: 10.1016/j.future.2013.01.010.
14. Rajalakshmi P., Ramesh M. V. Smart healthcare systems using IoT // Internet of Things: Concepts and Applications. – Singapore: Springer, 2018. – P. 239–268.
15. Singh R., Rajan R., Sharma P., Ahirwar S. ESP32-based health monitoring system for COVID-19 // International Journal of Advanced Science and Technology. – 2020. – Vol. 29, No. 7. – P. 13336–13346.
16. Gupta N., Chauhan R., Goel D. A review of Webhooks and REST APIs for integrating IoT systems // International Journal of Computer Applications. – 2022. – Vol. 184, No. 38. – P. 1–6.
17. Parra-Arnau J. Data minimization and privacy by design: The case of Webhooks in healthcare IoT // Computer Standards & Interfaces. – 2018. – Vol. 59. – P. 45–56. – DOI: 10.1016/j.csi.2018.03.007.

18. Park J., Kim M., Lee J. IoT data security using lightweight cryptography for wearable healthcare devices // *Sensors*. – 2020. – Vol. 20, No. 17. – Article 4820. – DOI: 10.3390/s20174820.
19. Raza S., Wallgren L., Voigt T. Lightweight security solutions for the Internet of Things // *IEEE Internet of Things Journal*. – 2017. – Vol. 4, No. 6. – P. 1910–1923. – DOI: 10.1109/JIOT.2017.2685589.
20. Ali M., Awad A. IoT-based remote health monitoring and medical data management system // *Health Informatics Journal*. – 2021. – Vol. 27, No. 4. – P. 1–16. – DOI: 10.1177/14604582211043339.
21. Tabish R., Akram R., Hamad S., Khan J. A 3G/WiFi-enabled 6LoWPAN-based U-healthcare system for ubiquitous real-time monitoring and data logging // *Proc. Middle East Conf. Biomed. Eng. (MECBME)*. – Feb. 2014. – P. 277–280.
22. Imadali S., Karanasiou A., Petrescu A., Sifniadis I., Veque V., Angelidis P. eHealth service support in IPv6 vehicular networks // *Proc. IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*. – Oct. 2012. – P. 579–585.
23. Imadali S., Karanasiou A., Petrescu A., Sifniadis I., Veque V., Angelidis P. eHealth service support in IPv6 vehicular networks // *Proc. IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*. – Oct. 2012. – P. 579–585.
24. Zhao W., Chaowei W., Nakahira Y. Medical application on Internet of Things // *Proc. IET Int. Conf. Commun. Technol. Appl. (ICCTA)*. – Oct. 2011. – P. 660–666.
25. Віддалений моніторинг ЕКГ за допомогою шлюзу Bluetooth [Електронний ресурс]. – Режим доступу: <https://www.dusuniot.com/uk/blog/remote-ecg-monitoring-using-bluetooth-gateway/>. – Назва з екрана. – Дата звернення: 25.04.2025.
26. Гоженко А., Кульбіда М., Кочет О. Профілактична стратегія медичної науки – шлях до підвищення ефективності охорони здоров'я // *Вісник НАН України*. – 2011. – № 12. – С. 64–69.
27. Збереження інформації про температуру та вологість в ThingSpeak [Електронний ресурс]. – Режим доступу:

https://www.radioradar.net/radiofan/miscellaneous/storage_thing_speak_information_temperature_humidity.html. – Назва з екрана. – Дата звернення: 25.04.2025.

28. Злепко С. М., Белзецький Р. С. Система дистанційного моніторингу за станом здоров'я людини // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2008. – С. 217–219.

29. Микитишин А. Г., Митник М. М., Стухляк П. Д., Пасічник В. В. Комп'ютерні мережі : навч. посіб. – Львів : Магнолія 2006, 2013. – 256 с.

30. Паламар А. М., Купратий І. Г. Система для дистанційного моніторингу стану здоров'я пацієнтів на основі Інтернету медичних речей // Матеріали X наук.-техн. конф. Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології». – Тернопіль : ТНТУ, 2022. – С. 85.

31. Портативний шлюз Bluetooth до Wi-Fi/LTE DSGW-340 для переносних медичних пристроїв [Електронний ресурс]. – Режим доступу: <https://www.dusuniot.com/uk/product-specification/dsgw-340-ble-to-wifi-lte-gateway/>. – Назва з екрана. – Дата звернення: 25.04.2025.

32. Система дистанційного моніторингу стану за допомогою IoT, як це працює? [Електронний ресурс]. – Режим доступу: <https://www.dusuniot.com/>. – Назва з екрана. – Дата звернення: 25.04.2025.

33. Козловський В. О., Лесько О. Й., Кавецький В. В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. – Вінниця : ВНТУ, 2021. – 42 с.

34. Postel J., Reynolds J. Telnet Protocol Specification. STD 8, RFC 854, May 1983 [Електронний ресурс]. – Режим доступу: <https://datatracker.ietf.org/doc/html/rfc854>. – Назва з екрана. – Дата звернення: 25.04.2025.

35. Herzberg B., Bekerman D., Zeifman I. Breaking Down Mirai: An IoT DDoS Botnet Analysis. Imperva Incapsula, Oct. 2016 [Електронний ресурс]. – Режим доступу: <https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html>. – Назва з екрана. – Дата звернення: 25.04.2025.

36. Siddavaatam P., Sedaghat R., Cheng M. H. An Adaptive Security Framework with Extensible Computational Complexity for Cipher Systems // Proc. 11th Int. Conf. for Internet Technology and Secured Transactions. – Barcelona, Spain, 5–7 December 2016.
37. Fischlin M., Janson C., Mazaheri S. Backdoored Hash Functions: Immunizing HMAC and HKDF // Proc. 2018 IEEE 31st Computer Security Foundations Symposium. – Oxford, UK, 9–12 July 2018.
38. Perrazzone J. B., Yu P. L., Sadler B. M., Blum R. S. Cryptographic side-channel signaling and authentication via fingerprint embedding // IEEE Transactions on Information Forensics and Security. – 2018. – Vol. 13. – P. 2216–2225.
39. Білоус С. В. Технології Інтернету речей в системах телемедицини : монографія. – Харків : ХНУРЕ, 2020. – 172 с.
40. Климов В. В., Шевченко В. С., Панасюк І. О. Інтернет речей: архітектура, протоколи, стандарти та платформи : навч. посіб. – Київ : КНУ ім. Т. Шевченка, 2021. – 268 с.
41. Kumar N., Mallick P. K. The Internet of Things: Insights into the building blocks, component interactions, and architecture layers // Procedia Computer Science. – 2018. – Vol. 132. – P. 109–117. – DOI: 10.1016/j.procs.2018.05.158.
42. Tan L., Wang N. Future Internet: The Internet of Things // Proc. 3rd Int. Conf. on Advanced Computer Theory and Engineering (ICACTE). – 2010. – P. V5-376–V5-380. – DOI: 10.1109/ICACTE.2010.5579493.
43. Khan R., Khan S. U., Zaheer R., Khan S. Future Internet: The Internet of Things architecture, possible applications and key challenges // Proc. 10th Int. Conf. on Frontiers of Information Technology. – 2012. – P. 257–260. – DOI: 10.1109/FIT.2012.53.
44. Firebase Authentication Documentation [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs/auth>. – Назва з екрана. – Дата звернення: 25.04.2025.

45. Bluetooth SIG. Bluetooth Core Specification v5.2 [Электронный ресурс]. – Режим доступа: <https://www.bluetooth.com/specifications/bluetooth-core-specification/>. – Назва з екрана. – Дата звернення: 25.04.2025.

46. International Telecommunication Union (ITU). Recommendation ITU-T Y.2060: Overview of the Internet of Things [Электронный ресурс]. – Режим доступа: <https://www.itu.int/rec/T-REC-Y.2060-201206-I/en>. – Назва з екрана. – Дата звернення: 25.04.2025.

47. Hossain M. S., Muhammad G., Alamri A. Smart healthcare monitoring: A voice pathology detection paradigm for smart cities // Multimedia Tools and Applications. – 2017. – Vol. 76. – P. 18489–18503. – DOI: 10.1007/s11042-016-4122-3.

48. Arduino. Arduino Nano 33 IoT Board Documentation [Электронный ресурс]. – Режим доступа: <https://docs.arduino.cc/hardware/nano-33-iot>. – Назва з екрана. – Дата звернення: 25.04.2025.

**Програмний код в середовищі Arduino IDE, який реалізує процес
зчитування показників та їх відправку до ThingSpeak.**

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ThingSpeak.h>
#define DHTPIN D5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
const char* ssid = "WiFiName";
const char* password = "Password";
WiFiClient client;
unsigned long myChannelNumber = 174314;
const char * myWriteAPIKey = "JQPT38EM56K0MJM1";
uint8_t temperature, humidity;
void setup()
{
  Serial.begin(115200);
  dht.begin();
  delay(10);
  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
```

```

Serial.println("WiFi connected");
// Print the IP address
Serial.println(WiFi.localIP());
ThingSpeak.begin(client);
}

void loop()
{
  static boolean data_state = false;
  temperature = dht.readTemperature();
  humidity = dht.readHumidity();
  Serial.print("Temperature Value is :");
  Serial.print(temperature);
  Serial.println("C");
  Serial.print("Humidity Value is :");
  Serial.print(humidity);
  Serial.println("%");

  // Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8
different
  // pieces of information in a channel. Here, we write to field 1.
  if( data_state )
  {
    ThingSpeak.writeField(myChannelNumber, 1, temperature, myWriteAPIKey);
    data_state = false;
  }
  else
  {
    ThingSpeak.writeField(myChannelNumber, 2, humidity, myWriteAPIKey);
    data_state = true;
  }
  delay(30000); // ThingSpeak will only accept updates every 15 seconds.
}

```

**Кодовий лістинг з параметрами підключення до платформи ThingSpeak
через бездротову мережу Wi-Fi**

```
#define USE_ARDUINO_INTERRUPTS false
#include <Wire.h>
#include "Protocentral_MAX30205.h"
#include <PulseSensorPlayground.h>
#include <WiFi.h>
#include <ThingSpeak.h>

MAX30205 tempSensor;
const char ssid[] = "Wi-Fi_SSID";
const char password[] = "xxxxxxxxxx";
WiFiClient client;
const long CHANNEL = 1985020;
const char *WRITE_API = "EQRK5VI8xxxxxxxx";

long prevMillisThingSpeak = 0;
int intervalThingSpeak = 20000; // write interval is 20 seconds
```

Рисунок Б.1 - Кодовий лістинг з параметрами підключення до платформи
ThingSpeak через бездротову мережу Wi-Fi

Лістинг передачі вимірюваних параметрів здоров'я через ThingSpeak

```
192 int myBPM = pulseSensor.getBeatsPerMinute(); // read pulse rate
193 float myTEMP = tempSensor.getTemperature(); // read temperature
194
195 if (millis() - prevMillisThingSpeak > intervalThingSpeak) {
196     // Set the fields with the values
197     if (myBPM > 40 && myBPM < 150) {
198         ThingSpeak.setField(1, myBPM);
199     }
200     if (myTEMP > 35 && myTEMP < 41) {
201         ThingSpeak.setField(2, myTEMP);
202     }
203     // Write to the ThingSpeak channel
204     int x = ThingSpeak.writeFields(CHANNEL, WRITE_API);
205     if (x == 200) {
206         Serial.println("Channel update successful.");
207     } else {
208         Serial.println("Problem updating channel. HTTP error code " + String(x));
209     }
210     prevMillisThingSpeak = millis();
211 }
```

Рисунок В.1 – Лістинг передачі вимірюваних параметрів здоров'я через ThingSpeak

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (презентація)



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Державний університет інформаційно-комунікаційних технологій

ДИПЛОМНА РОБОТА
на ступінь вищої роботи бакалавр
із спеціальності 122 Комп'ютерні науки

«Веб-додаток для моніторингу здоров'я з використанням IoT технологій»

Виконав: студент 4 курсу, групи КНД-43

Тимченко Віталій Володимирович

Керівник: д.т.н., доцент, Катков Ю.І.

Київ - 2025

1

ЗАГАЛЬНА ХАРАКТЕРИСТИКА ДИПЛОМНОЇ РОБОТИ

- **Мета роботи:** Підвищення ефективності контролю за станом здоров'я шляхом розробки веб-додатку, що використовує IoT-пристрої та технологію Webhook..
- **Об'єкт дослідження:** Процес віддаленого моніторингу стану здоров'я людини..
- **Предмет дослідження:** Веб-додаток, що використовує IoT-пристрої та технологію Webhook.
- **Наукове завдання:** Оцінка ефективності розробленого веб-додатку, що використовує IoT-пристрої та технологію Webhook..



2

АКТУАЛЬНІСТЬ ТЕМИ

Створення зручних та ефективних систем віддаленого моніторингу, які дозволяють не тільки збирати дані з датчиків фізіологічного стану, але й своєчасно реагувати на критичні зміни є своєчасним. Особливої важливості набуває застосування технологій сповіщення, таких як Webhook, що забезпечують миттєву інтеграцію з веб-сервісами та платформами. Така інтеграція дозволяє оперативно передавати дані у веб-браузер користувача, медичний портал або сервер аналітики, що значно підвищує ефективність роботи IoT-системи, що є актуальним.



3

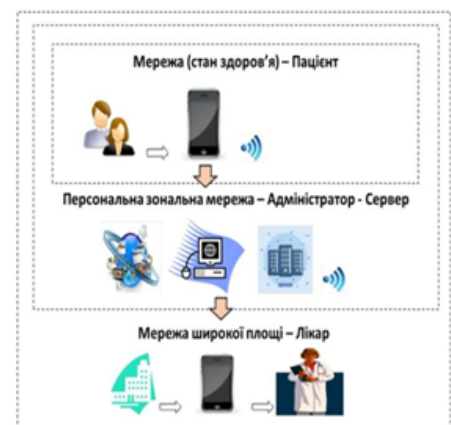
ПОСТАНОВКА ЗАВДАННЯ

Розробка веб-додатку для системи моніторингу стану здоров'я з інтеграцією IoT та медичних пристроїв, полягає в створенні інноваційного інструменту – веб-додатку, який поєднує технології IoT, веб-розробку, медичні знання та дозволяє збирати, обробляти, аналізувати та візуалізувати дані про стан здоров'я користувачів у реальному часі для покращення якості медичного обслуговування та підтримки здоров'я користувачів.

Тому виникає завдання розробки веб-додатку для моніторингу фізіологічного стану користувача з використанням IoT технологій та реалізація миттєвого сповіщення за допомогою механізму Webhook.

Завданнями дослідження є:

1. Аналіз можливостей впровадження IoT у сферу медичного моніторингу;
2. Дослідження архітектури IoT-пристрою на базі ESP32;
3. Вивчення способів інтеграції Webhook для сповіщення про критичні події;
4. Розробка прототипу веб-додатку та тестування його працездатності.



4

1 АНАЛІЗ МОЖЛИВОСТЕЙ ВИКОРИСТАННЯ ІОТ-ТЕХНОЛОГІЙ У СИСТЕМАХ МОНІТОРІНГУ ЗДОРОВ'Я



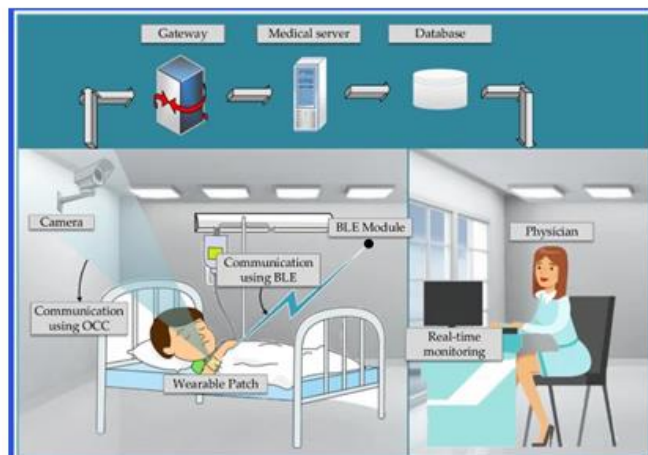
5

2 ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ ІОТ-ТЕХНОЛОГІЙ У СИСТЕМАХ МОНІТОРІНГУ ЗДОРОВ'Я



6

3 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО РЕАЛІЗАЦІЇ ВЕБ-СИСТЕМИ ДЛЯ МОНІТОРИНГУ ЗДОРОВ'Я



7

Висновки

Результати дослідження. У результаті дослідження розроблено прототип веб-додатку для моніторингу фізіологічних показників користувача, який поєднує можливості IoT-пристрою на базі ESP32 та механізму Webhook. Система забезпечує збір, обробку та візуалізацію даних у режимі реального часу, а також реалізує функціонал миттєвих сповіщень у разі виявлення критичних змін параметрів. Запропонована модель підвищує ефективність моніторингу здоров'я та відкриває перспективи для подальшого використання в системах домашньої медицини та телемедицинських платформах.

Наукова новизна. Новизна полягає у поєднанні IoT-архітектури з механізмом миттєвого сповіщення Webhook, що дозволяє реалізувати високоефективну систему моніторингу для медичних і побутових задач.

Практична значущість результатів дослідження – результати дослідження можуть бути використані при розробці систем домашнього нагляду за станом здоров'я, у телемедицинських рішеннях, а також у спеціалізованих медичних установах для віддаленої діагностики.

8

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЛЕННЯ

Матеріали були опубліковані:

В тезисах:

1. Тимченко В.В. СТВОРЕННЯ ВЕБ-ДОДАТКУ МОНІТОРИНГ СТАНУ ЗДОРОВ'Я ПІД ЧАС ІНТЕГРАЦІЇ ІОТ ТА МЕДИЧНИХ ПРИСТРОЇВ// VI Науково-технічна конференція «Сучасний стан та перспективи розвитку ІоТ». Збірник тез 15 квітня 2025. – К.: ДУІКТ, https://duikt.edu.ua/ua/news-1-570-13858-zaproshennya-na-vi-mizhnarodnu-naukovo-tehnichnu-konferenciyu-suchasniy-stand-ta-perspektivi-rozvitku-iot_kafedra-informaciy-nih-sistem-ta-tehnologiy