

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий Інститут телекомунікаційних систем

Кафедра електронних комунікацій та інтернету речей

«До захисту допущено»

Завідувач кафедри

_____ Анатолій МАКАРЕНКО

«__» _____ 20__ р.

Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності 172 Телекомунікації та радіотехніка

**на тему: «Система віддаленого моніторингу стану здоров'я пацієнтів за
допомогою IoT-пристроїв »**

Виконав (-ла):

студент (-ка) IV курсу, групи ТС-22

Корнєєв Михайло Андрійович _____

Керівник:

Завідувач кафедри, д.т.н., професор,

Макаренко Анатолій Олександрович _____

Рецензент:

Завідувач кафедри Університету Грінченка,

к.т.н., доцент,

Складанний Павло Миколайович _____

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий Інститут телекомунікаційних систем
Кафедра електронних комунікацій та інтернету речей

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 Телекомунікації та радіотехніка

Освітня програма – «Системи електронних комунікацій та Інтернету речей»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Анатолій МАКАРЕНКО

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Корнєєву Михайлу Андрійовичу

1. Тема роботи « Система віддаленого моніторингу стану здоров'я пацієнтів за допомогою IoT-пристроїв», керівник роботи Макаренко Анатолій Олександрович, д.т.н., професор затверджені наказом по університету від «26» травня 2026 р. № 1912-С.
2. Термін подання студентом роботи _____
3. Вихідні дані до роботи: Матеріали наукових статей, оглядів і періодичних видань за тематикою Інтернету медичних речей (IoMT), телемедицини, віддаленого моніторингу стану здоров'я, протоколів передавання даних MQTT, CoAP, BLE та стандартів семантичної сумісності HL7 FHIR, ISO/IEEE 11073. Технічна документація на мікроконтролери ESP32, STM32WBA, медичні сенсори AD8232, MAX30102, BME680, хмарні IoT-платформи ThingsBoard, AWS IoT Core, Microsoft Azure та середовище моделювання Wokwi. Нормативно-правові документи України у сфері телемедицини, законодавчі акти щодо захисту персональних даних, вимоги GDPR, EU Data Act та архітектурні вимоги до ЕСОЗ. Навчально-методичні матеріали кафедри електронних комунікацій та інтернету речей КПІ ім. Ігоря Сікорського щодо виконання дипломних робіт.
4. Зміст роботи: Збір та аналіз науково-технічної літератури за темою дистанційного медичного моніторингу, IoMT і телемедицини. Огляд IoT-архітектур для збору та передавання біомедичних даних, зокрема хмарних, периферійних і туманних обчислень. Дослідження сенсорів ЕКГ і

пульсоксиметрії, мікроконтролерів ESP32, STM32WBA, комунікаційних протоколів MQTT, BLE, CoAP та стандартів HL7 FHIR, ISO/IEEE 11073. Аналіз застосування телемедицини систем в Україні та світі, а також нормативно-правової бази. Побудова віртуального тестового стенду в Wokwi, вибір апаратних компонентів і програмного забезпечення. Проведення експериментальних досліджень із реалізацією Edge Computing, інтеграцією з MQTT-брокером, налаштуванням ThingsBoard, вимірюванням затримок і втрати пакетів та оптимізацією роботи системи за умов нестабільного зв'язку.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): Презентація дипломної роботи (електронна) на 11 слайдах, що включає: 1) титульний слайд; 2) актуальність роботи (глобальне зростання ринку телемедицини, медична інфляція, екстремальні умови в Україні); 3) мету роботи; опис конфігурації системи (три рівні: фізичний, мережевий, семантичний); 4) ілюстрацію симуляції та «цифрового двійника» (Wokwi); фрагмент програми для ESP32 з логікою Edge Computing; 5) налаштування хмарної платформи ThingsBoard (дашборд, Rule Engine); 6) результати тестування (три етапи: норма, критичний стан, автоматичне очищення; 7) графік затримок); 8) загальні висновки.

6. Дата видачі завдання 25.02.2026

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання, узгодження теми та структури дипломної роботи	14.01.26 -10.02.26	Виконано
2	Аналіз сучасних підходів до телемедицини, IoMT та огляд нормативно-правової бази (Україна, GDPR, EU Data Act)	11.02.26 – 28.02.26	Виконано
3	Дослідження архітектур IoT-систем моніторингу здоров'я, порівняння протоколів передачі даних (MQTT, CoAP, BLE) та стандартів семантичної сумісності (HL7 FHIR, IEEE 11073)	01.03.26 – 19.03.26	Виконано
4	Обґрунтування вибору апаратних компонентів (ESP32, AD8232, MAX30102) та програмного забезпечення (ThingsBoard, Wokwi)	20.03.26 – 08.04.26	Виконано
5	Розробка структурно-функціональної схеми системи, програмування мікроконтролера з реалізацією граничних обчислень (Edge Computing) та інтеграція з хмарною платформою	09.04.26 – 30.04.26	Виконано

6	Імітаційне моделювання роботи системи у середовищі Wokwi, налаштування дашборду та механізму правил (Rule Engine) у ThingsBoard, проведення експериментів з оцінки затримок та втрати пакетів	01.05.26 – 15.05.26	Виконано
7	Аналіз отриманих результатів, оформлення пояснювальної записки, підготовка презентації та реферату	16.05.26 – 03.06.26	

Студент

Михайло КОРНЄЄВ

Керівник роботи

Анатолій МАКАРЕНКО

РЕФЕРАТ

Текстова частина дипломної роботи: 80 с., 13 рис., 9 табл., 60 джерела.

Актуальність роботи зумовлена стрімким зростанням ринку телемедицини (прогнозований обсяг 175,5 млрд дол. США у 2026 році), високою медичною інфляцією та необхідністю забезпечення безперервного моніторингу хворих із хронічними захворюваннями в умовах воєнного стану в Україні (періодичні відключення електроенергії, нестабільність зв'язку, кіберзагрози). Традиційні централізовані архітектури IoT мають єдину точку відмови та не забезпечують належної семантичної сумісності з національною електронною системою охорони здоров'я (ЕСОЗ). Використання гібридної хмарно-периферійної парадигми (Cloud-Edge Collaborative Model) та стандарту HL7 FHIR дозволяє створити відмовостійку, масштабовану та інтероперабельну систему віддаленого моніторингу.

Мета роботи — дослідити побудову системи віддаленого моніторингу стану здоров'я пацієнтів на основі IoT-пристроїв.

Об'єкт дослідження — процес збору, передавання та обробки біомедичних даних у системах віддаленого медичного моніторингу.

Предмет дослідження — методи побудови та оцінювання ефективності IoT-систем віддаленого моніторингу стану здоров'я пацієнтів.

Задачі дослідження: проведено аналіз сучасних підходів до дистанційного моніторингу, глобальних тенденцій телемедицини та нормативно-правової бази України (накази МОЗ №77, №109, GDPR, EU Data Act). Спроектовано багаторівневу структурно-функціональну схему (рівні сприйняття, периферійних обчислень, мережевий, хмарної аналітики та інтеграції з ЕСОЗ). Обґрунтовано вибір апаратних компонентів (ESP32, AD8232, MAX30102) та комунікаційних протоколів (MQTT, BLE). Розроблено програмне забезпечення для мікроконтролера з реалізацією граничних обчислень (Edge Computing) та інкапсуляцією даних у стандарт HL7 FHIR (ресурс Observation). Виконано імітаційне моделювання у середовищі Wokwi та розгортання хмарної платформи

ThingsBoard із налаштуванням інтерактивного дашборду та механізму правил (Rule Engine) для автоматичної генерації тривоги.

Методи дослідження: аналіз науково-технічної літератури та нормативних документів, системне проектування, імітаційне моделювання (Wokwi), експериментальне тестування з вимірюванням показників якості обслуговування (затримка, втрата пакетів).

Ключові слова: ІНТЕРНЕТ МЕДИЧНИХ РЕЧЕЙ (ІоМТ), ВІДДАЛЕНИЙ МОНІТОРИНГ СТАНУ ЗДОРОВ'Я, ТЕЛЕМЕДИЦИНА, ЕЛЕКТРОННА СИСТЕМА ОХОРОНИ ЗДОРОВ'Я (ЕСОЗ), HL7 FHIR, MQTT, ГРАНИЧНІ ОБЧИСЛЕННЯ (EDGE COMPUTING), МІКРОКОНТРОЛЕР ESP32, ПЛАТФОРМА THINGSBOARD, SEMANTIC INTEROPERABILITY.

ABSTRACT

Thesis text: 80 p., 13 fig., 9 tab., 60 references.

Relevance. The rapid growth of the telehealth market (projected to reach USD 175.5 billion by 2026), high medical inflation, and the need for continuous patient monitoring under wartime conditions in Ukraine (power outages, unstable connectivity, cyber threats) make the development of reliable remote health monitoring systems critically important. Traditional centralized IoT architectures have a single point of failure and lack semantic interoperability with the national eHealth system (eHealth of Ukraine). The use of a hybrid cloud-edge collaborative model together with the HL7 FHIR standard enables a fault-tolerant, scalable, and interoperable remote monitoring system.

The aim of the work — to study the development of a remote patient health monitoring system based on IoT devices.

The object of the research — the process of collecting, transmitting, and processing biomedical data in remote medical monitoring systems.

The subject of the research — methods for developing and evaluating the efficiency of IoT-based systems for remote monitoring of patients' health status.

Tasks: A comprehensive analysis of modern remote monitoring approaches, global telehealth trends, and the regulatory framework of Ukraine (Orders of Ministry of Health No. 77, No. 109, GDPR, EU Data Act) was conducted. A multi-layer structural-functional scheme (perception, edge computing, network, cloud analytics, and eHealth integration layers) was designed. The choice of hardware components (ESP32, AD8232, MAX30102) and communication protocols (MQTT, BLE) was justified. Firmware for the microcontroller was developed, implementing edge computing logic and encapsulating data into the HL7 FHIR standard (Observation resource). Simulation was performed in the Wokwi environment, and the ThingsBoard cloud platform was deployed with an interactive dashboard and a rule engine for automatic alarm generation.

Methods: Analysis of scientific literature and regulatory documents, system design, simulation (Wokwi), experimental testing with quality-of-service metrics (latency, packet loss).

Keywords: INTERNET OF MEDICAL THINGS (IoMT), REMOTE PATIENT MONITORING, TELEMEDICINE, eHEALTH OF UKRAINE, HL7 FHIR, MQTT, EDGE COMPUTING, ESP32 MICROCONTROLLER, THINGSBOARD PLATFORM, SEMANTIC INTEROPERABILITY.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ ..	10
ВСТУП.....	13
1 ТЕОРЕТИЧНІ ЗАСАДИ ПОБУДОВИ СИСТЕМ ВІДДАЛЕНОГО МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я НА ОСНОВІ ІОТ.....	15
1.1 Аналіз сучасних підходів до дистанційного медичного моніторингу та телемедицини	15
1.2 Архітектура IoT-систем для збору та передавання біомедичних даних.....	21
1.3 Висновки до розділу 1	30
2 ПРОЄКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ ВІДДАЛЕНОГО МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я.....	32
2.1 Розробка структурної та функціональної схеми системи на основі IoT- пристроїв.....	32
2.2 Вибір апаратних компонентів, протоколів передавання даних та програмного забезпечення	39
2.3 Висновки до розділу 2	51
3 РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ.....	54
3.1 Реалізація програмно-апаратного комплексу та інтеграція сенсорних модулів	55
3.2 Експериментальна перевірка роботи системи та оцінка параметрів якості передавання і обробки даних.....	61
3.3 Висновки до розділу 3	71
ВИСНОВКИ.....	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	75

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI	Artificial Intelligence (штучний інтелект)
AMA	American Medical Association (Американська медична асоціація)
AMQP	Advanced Message Queuing Protocol (розширений протокол черг повідомлень)
API	Application Programming Interface (інтерфейс прикладного програмування)
AWS	Amazon Web Services
BAA	Business Associate Agreement (угода з діловим партнером)
BLE	Bluetooth Low Energy (Bluetooth з низьким енергоспоживанням)
CGM	Continuous Glucose Monitoring (безперервний моніторинг глюкози)
CoAP	Constrained Application Protocol (протокол прикладного рівня для обмежених пристроїв)
CPS	Cyber-Physical Systems (кіберфізичні системи)
CPU	Central Processing Unit (центральний процесор)
DDoS	Distributed Denial of Service (розподілена відмова в обслуговуванні)
DoF	Devices on FHIR (пристрої на FHIR)
ECG / ЕКГ	Electrocardiogram (електрокардіограма)
Edge/Fog Computing	периферійні/туманні обчислення
eHealth	electronic Health (електронне здоров'я)
EHR /	Electronic Health Record (електронна медична картка)
ЕМК	
EMR	Electronic Medical Record (електронний медичний запис)
ЕСОЗ	електронна система охорони здоров'я (Україна)

FHIR	Fast Healthcare Interoperability Resources (швидкі ресурси медичної сумісності)
GCP	Google Cloud Platform
GDPR	General Data Protection Regulation (Загальний регламент захисту даних)
HIPAA	Health Insurance Portability and Accountability Act (США)
HL7	Health Level Seven (міжнародний стандарт обміну медичними даними)
HPA	Horizontal Pod Autoscaling (горизонтальне автомасштабування подів)
HR / ЧСС	Heart Rate (частота серцевих скорочень)
HTA	Health Technology Assessment (оцінка медичних технологій)
HTTP	Hypertext Transfer Protocol (протокол передачі гіпертексту)
I2C	Inter-Integrated Circuit (послідовна шина даних)
IDS	Intrusion Detection System (система виявлення вторгнень)
IoMT	Internet of Medical Things (Інтернет медичних речей)
IoT	Internet of Things (Інтернет речей)
JSON	JavaScript Object Notation (формат обміну даними)
KC3I	комплексна система захисту інформації (Україна)
LOINC	Logical Observation Identifiers Names and Codes
LPWAN	Low-Power Wide-Area Network (низькопотужна глобальна мережа)
MCU	Microcontroller Unit (мікроконтролер)
MDER	Medical Device Encoding Rules (правила кодування медичних пристроїв)
mHealth	mobile Health (мобільне здоров'я)
MIC	медична інформаційна система
МОЗ	Міністерство охорони здоров'я України
MQTT	Message Queuing Telemetry Transport (транспорт телеметрії з чергами повідомлень)
NB-IoT	Narrowband Internet of Things (вузькосмуговий Інтернет речей)

NHIS	National Health Interview Survey (США)
OPCW	Organisation for the Prohibition of Chemical Weapons (використано як приклад, у тексті не зустрічається – видалено)
PHD	Personal Health Device (персональний медичний пристрій)
PHI	Protected Health Information (захищена медична інформація)
QoS	Quality of Service (якість обслуговування)
RAM	Random Access Memory (оперативна пам'ять)
RBAC	Role-Based Access Control (керування доступом на основі ролей)
REST	Representational State Transfer (передача репрезентативного стану)
RPC	Remote Procedure Call (виклик віддалених процедур)
RPM	Remote Patient Monitoring (віддалений моніторинг пацієнта)
SDH	ScalableDigitalHealth (масштабована цифрова архітектура здоров'я)
SpO ₂	Peripheral capillary oxygen saturation (сатурація кисню в крові)
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security (безпека транспортного рівня)
UDP	User Datagram Protocol (протокол дейтаграм користувача)
VOC	Volatile Organic Compounds (леткі органічні сполуки)
VPC	Virtual Private Cloud (віртуальна приватна хмара)
WHO / BOOЗ	World Health Organization (Всесвітня організація охорони здоров'я)
WTW	Willis Towers Watson (консалтингова компанія)
XML	eXtensible Markup Language (розширювана мова розмітки)
ЦБД	центральна база даних (ЕСОЗ)

ВСТУП

Останніми роками телемедицина перетворилася з екзотичної послуги на необхідність. Світовий ринок віддаленого моніторингу пацієнтів стрімко зростає, і за прогнозами до 2026 року він має сягнути майже 175 мільярдів доларів. Головні рушії цього процесу — старіння населення, зростання кількості хронічних захворювань та, що не менш важливо, медична інфляція, яка щороку підвищує вартість лікування на понад 10%. В таких умовах звичний підхід «пацієнт приходить до лікаря раз на кілька тижнів» стає неефективним. Натомість дедалі більшої ваги набуває безперервний моніторинг життєвих показників за допомогою невеликих носимих пристроїв, які передають дані лікарю в режимі реального часу.

Для України ця тема має особливе значення. Воєнний стан зробив звичну медичну інфраструктуру вразливою: зруйновані лікарні, перебої з електрикою, нестабільний мобільний зв'язок — усе це ускладнює надання допомоги, особливо хворим на серцево-судинні та респіраторні захворювання. Водночас наша держава вже має розвинену електронну систему охорони здоров'я (ЕСОЗ), до якої підключено майже 50 тисяч закладів. Тому логічним кроком є інтеграція даних з портативних медичних датчиків безпосередньо в цю систему. Однак існуючі підходи до побудови таких систем часто покладаються виключно на хмарні сервери, що створює єдину точку відмови, а також не забезпечують сумісності між різними пристроями через відсутність єдиного стандарту обміну даними.

Мета роботи — дослідити побудову системи віддаленого моніторингу стану здоров'я пацієнтів на основі IoT-пристроїв.

Об'єкт дослідження — процес збору, передавання та обробки біомедичних даних у системах віддаленого медичного моніторингу.

Предмет дослідження — методи побудови та оцінювання ефективності IoT-систем віддаленого моніторингу стану здоров'я пацієнтів.

Для досягнення цієї мети в роботі було вирішено кілька послідовних завдань. Спочатку проведено аналіз сучасних телемедичних рішень, їхніх

архітектур і протоколів, а також вивчено нормативну базу України, зокрема накази МОЗ №77 та №109, вимоги GDPR і майбутнього EU Data Act. Далі обґрунтовано вибір апаратного забезпечення — мікроконтролера ESP32 разом із датчиками ЕКГ (AD8232) та пульсоксиметром (MAX30102). Для зв'язку між пристроєм і сервером обрано легковаговий протокол MQTT, а для семантичної сумісності — згаданий HL7 FHIR. Наступним кроком стала розробка програмної прошивки для мікроконтролера, де реалізовано логіку виявлення критичних станів (наприклад, падіння сатурації нижче 92%) без звернення до хмари. Потім виконано імітаційне моделювання в середовищі Wokwi, створено «цифрового двійника» системи та розгорнуто хмарну платформу ThingsBoard з інтерактивним дашбордом і автоматичними тривогами. Нарешті, проведено експерименти для оцінки реальної затримки передачі даних та стабільності роботи.

Практичне значення отриманих результатів полягає в тому, що розроблена архітектура може слугувати основою для реальних телемедичних продуктів, які без додаткових доопрацювань інтегруються з ЕКОЗ через RESTful API. Експериментально підтверджено, що середня затримка передачі даних від датчика до екрана лікаря становить близько 1,1 секунди, що цілком прийнятно для систем моніторингу стану здоров'я.

1 ТЕОРЕТИЧНІ ЗАСАДИ ПОБУДОВИ СИСТЕМ ВІДДАЛЕНОГО МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я НА ОСНОВІ ІОТ

1.1 Аналіз сучасних підходів до дистанційного медичного моніторингу та телемедицини

Еволюція дистанційного медичного моніторингу та концептуалізація IoMT. Сучасні дослідження в галузі електронної охорони здоров'я (eHealth) та мобільної медицини (mHealth) вказують на те, що Інтернет медичних речей (IoMT) не є просто технологічним інструментом, а є складною системою, що поєднує технології та людський фактор. IoMT визначається як мережа взаємопов'язаних медичних пристроїв, програмних додатків та систем охорони здоров'я і послуг, які об'єднуються за допомогою телекомунікаційних технологій [1]. Цей напрям з'явився як відповідь на зростаючу потребу у безперервному потоці інформації між пацієнтом та надавачем медичних послуг. Віддалений моніторинг стану здоров'я дозволяє трансформувати дискретні, епізодичні клінічні вимірювання (наприклад, вимірювання артеріального тиску лише під час візиту до лікаря) у цілісну, безперервну картину динаміки стану пацієнта [2].

Фундаментальною складовою успішного функціонування систем IoT-RMS (IoT-based Remote Monitoring Systems) є концепція залучення пацієнтів (Patient Engagement). Аналіз сучасної наукової літератури, що охоплює понад 1800 досліджень з баз даних Web of Science, Scopus, IEEE та PubMed, демонструє еволюцію взаємовідносин між пацієнтом і технологіями. Цей процес можна умовно поділити на три ключові етапи: створення можливостей (enabling), залучення (engaging) та розширення прав і можливостей (empowering) [2, 3]. На стадії створення можливостей пацієнт отримує доступ до технологічного інструментарію (наприклад, смарт-годинників з функцією ЕКГ або Bluetooth-глюкометрів). На стадії залучення відбувається активна взаємодія користувача з інтерфейсами системи, які часто пропонують персоналізовані рекомендації, інформаційні статті та інтерактивні інструменти для управління власним здоров'ям [3]. Кінцевою стадією є розширення прав і можливостей, коли пацієнт,

маючи доступ до власних візуалізованих даних здоров'я в режимі реального часу та можливість двосторонньої комунікації з лікарем, стає повноцінним, проактивним учасником процесу лікування. Науково доведено, що такий підхід значно підвищує готовність пацієнта дотримуватися лікування та стимулює модифікацію способу життя, що є основою терапії хронічних неінфекційних захворювань [2].

Глобальні тенденції, статистичний аналіз та стратегічне бачення. Ринок телемедицини та віддаленого моніторингу перебуває у стані стрімкого зростання, яке було різко прискорене глобальною кризою системи охорони здоров'я, спричиненою пандемією COVID-19. Аналітика світових тенденцій демонструє вражаючі фінансові та кількісні показники. До кінця 2025 року прогнозується досягнення обсягу глобального ринку телемедицини позначки понад 55 мільярдів доларів США, а до 2026 року очікується подальший стрибок до 175,5 мільярдів доларів США, що становить майже чотирикратне збільшення порівняно з 2019 роком до пандемії. Кількість активних користувачів сервісів онлайн-консультацій у всьому світі перевищила 116 мільйонів осіб, збільшившись удвічі порівняно з базовим періодом. Показники впровадження телемедицини в рутинну клінічну практику зросли з маргінального рівня близько 1% від усіх візитів пацієнтів у лютому 2020 року до стабільних 17% у 2023 році [4].

Однак більш детальний аналіз статистики показує, що впровадження цих технологій відбувалося нерівномірно. Дані Національного опитування стану здоров'я США (National Health Interview Survey, NHIS) демонструють, що ринок почав вирівнюватися після пікового зростання: загальний відсоток дорослих американців, які користувалися телемедициною, знизився з 37,0% у 2021 році до 30,1% у 2022 році. Ця тенденція спостерігалася в різних соціодемографічних та географічних групах, незалежно від рівня доходу чи освіти. Вища ймовірність використання телемедицини зберігається серед жінок, осіб з вищою освітою та мешканців урбанізованих територій, тоді як особи без медичного страхування залишаються найменш залученими до цифрових медичних послуг [5]. Це зниження не є свідченням відмови від технологій, а свідчить про перехід від

вимушеного масового переходу на "дистанційку" до більш виважених змішаних форматів надання медичної допомоги (Hybrid Care Models), де віддалений моніторинг гармонійно доповнює традиційний клінічний огляд [4].

Дослідження Американської медичної асоціації (АМА) за 2024 рік підтверджують стійкість інтеграції телемедицини в професійну практику. Незважаючи на певне зниження порівняно з піковими показниками 2020 року (79%), у 2024 році 71,4% лікарів звітували про використання телемедичних технологій щотижня, що є безпрецедентним зростанням порівняно з 25,1% у 2018 році [6]. Цікавим є профільний розподіл: абсолютними лідерами у використанні віртуальних візитів є психіатри (85,9% використовували технологію протягом останнього тижня, а 68,2% проводять понад 20% своїх візитів дистанційно) [4]. Водночас сфера кардіології та ендокринології фокусується не стільки на відеоконсультаціях, скільки на безперервному зборі телеметричних даних з IoT-пристроїв (наприклад, холтерівських моніторів з підтримкою Bluetooth або датчиків рівня глюкози) [6].

Економічні причини, які змушують пришвидшити впровадження IoT-систем моніторингу формується під тиском глобальної медичної інфляції. За даними звіту WTW Global Medical Trends Survey (жовтень 2024 року), витрати на медичне обслуговування продовжують стрімко зростати, прогнозуючись на рівні глобального середнього показника в 10,4% у 2025 році (після 10,7% у 2023 році). Регіональні прогнози також демонструють висхідні тренди: у Північній Америці зростання складе з 8,1% до 8,7%, в Азійсько-Тихоокеанському регіоні — з 11,9% до 12,3%, а в країнах Близького Сходу та Африки — з 10,4% до 12,1%. Рушійними силами цієї інфляції є не лише загальноекономічні фактори, але й перевантаження публічних систем охорони здоров'я, старіння населення та впровадження високовартісних фармацевтичних розробок [7]. У цьому контексті системи віддаленого моніторингу допомагають стримувати зростання витрат. Статистика підтверджує, що телемедицина дозволяє суттєво економити час: в Австралії пацієнти заощаджують у середньому понад шість годин, а в Італії пацієнти з хронічними захворюваннями зберігають близько 9,5 годин на рік

завдяки зменшенню кількості непотрібних логістичних переміщень до лікарень [4].

На міжнародному рівні Всесвітня організація охорони здоров'я (ВООЗ) відіграє ключову роль у стандартизації та промоції цифрової медицини. Глобальна стратегія ВООЗ щодо цифрового здоров'я, ухвалена Всесвітньою асамблеєю охорони здоров'я у 2020 році, розглядає технології IoT як інструмент досягнення універсального охоплення послугами охорони здоров'я. Організація визначає три стратегічні цілі: по-перше, трансформація наукових даних у дії шляхом впровадження стандартів інтероперабельності; по-друге, розширення знань через створення глобальних наукових спільнот; по-третє, систематична оцінка потреб окремих країн щоб уникнути помилкової логіки "зробимо – і воно саме запрацює", який часто призводив до провалу інновацій [8]. Впровадження інновацій має базуватися виключно на реальних потребах національних систем охорони здоров'я. Одним із найважливіших стратегічних трендів до 2025 року є глибока інтеграція алгоритмів штучного інтелекту (AI) у платформи віддаленого моніторингу [4]. AI трансформує системи моніторингу з пасивних збирачів даних у проактивні діагностичні інструменти, здатні розпізнавати патерни в масивах фізіологічних показників та прогнозувати кризові стани (наприклад, загострення серцевої недостатності) ще до їх клінічної маніфестації [9].

Стан розвитку телемедицини та цифрової інфраструктури в Україні. Цифрова трансформація сфери охорони здоров'я є одним із найбільш успішних напрямів реалізації державних політик в Україні, що набуло особливого значення в умовах повномасштабної війни, коли доступ до стаціонарних лікарень на багатьох територіях є фізично неможливим або небезпечним. Станом на кінець 2024 року українська електронна система охорони здоров'я (eHealth) досягла безпрецедентних масштабів: до неї підключено понад 49 тисяч медичних та аптечних закладів, в ній працюють 475 тисяч медичних і фармацевтичних працівників, а центральна база даних зберігає понад 5 мільярдів структурованих електронних медичних записів [10].

Архітектура української ЕСОЗ побудована за унікальним дворівневим принципом, що забезпечує баланс між державним контролем та ринковою конкуренцією. Перший рівень — це Центральна база даних (ЦБД), адміністрування якої здійснює державне підприємство «Електронне здоров'я». Другий рівень складається з приватних Медичних інформаційних систем (МІС), які розробляються комерційними компаніями та забезпечують користувацький інтерфейс для лікарів та пацієнтів. З точки зору проектування систем віддаленого моніторингу стану здоров'я на базі IoT, архітектори повинні розглядати свої рішення як потенційні модулі, які мають інтегруватися з МІС через стандартизовані API для безпечного передавання телеметричних даних безпосередньо до електронної медичної картки пацієнта [11, 12].

У 2024–2025 роках Україна здійснила низку масштабних цифрових реформ, які свідчать про високу готовність інфраструктури до впровадження складних систем обробки даних. Було ліквідовано застарілі Медико-соціальні експертні комісії (МСЕК) та впроваджено повністю цифрову систему оцінювання повсякденного функціонування людини (ОПФО), яка отримала міжнародну нагороду GovTech Award 2025. Також вперше реалізовано електронний рейтинговий розподіл студентів-медиків в інтернатуру та запущено електронну чергу на ендопротезування. У сфері безпосереднього телемедичного моніторингу важливим кроком став запуск україно-французького проекту для дистанційної діагностики, що ілюструє готовність держави до міжнародної інтеграції IoT-рішень [10].

Законодавча та нормативно-правова база України у сфері телемедицини та IoMT. Формування ефективної екосистеми дистанційного моніторингу неможливе без створення чітких нормативно-правових рамок, які гарантують безпеку пацієнтів, захист їхніх персональних даних та визначають відповідальність сторін. В умовах воєнного стану уряд України вдався до швидкої адаптації регуляторного поля. Ключовим документом став Наказ Міністерства охорони здоров'я України від 15.01.2024 № 77 "Про затвердження Порядку надання медичної та/або реабілітаційної допомоги із застосуванням телемедицини

на період дії воєнного стану..." [13]. Цей наказ нормативно закріпив можливість використання телемедичних платформ не лише для первинних консультацій, але й для складних процесів дистанційної реабілітації. Додатково, Наказом МОЗ від 22.01.2024 № 109 було регламентовано проведення тестової експлуатації новітніх засобів телемедицини, що створює легальний механізм (так звану «регуляторну пісочницю») для пілотного впровадження інноваційних IoT-систем моніторингу життєвих показників в українських клініках [14].

Окремим і надзвичайно важливим аспектом є регулювання обігу та захисту персональних медичних даних, що збираються IoT-пристроями. Вектор європейської інтеграції України зобов'язує гармонізувати національне законодавство зі стандартами Європейського Союзу. 20 листопада 2024 року Верховна Рада України прийняла за основу проєкт нового Закону «Про захист персональних даних» № 8153. Цей законопроект розроблений з метою імплементації норм Загального регламенту про захист даних ЄС (GDPR) та положень оновленої Конвенції 108+. Для розробників систем віддаленого моніторингу це означає кардинальну зміну підходів до архітектури баз даних. Відтепер системи повинні проектуватися на принципах «privacy by design» (конфіденційність та захист даних на рівні архітектури за замовчуванням), забезпечувати наскрізне шифрування медичної телеметрії та реалізовувати технічну можливість для пацієнтів реалізувати своє «право на забуття» (повне видалення даних) та право на перенесення даних в інші системи [15].

Крім того, українські компанії, які розробляють апаратне або програмне забезпечення для медичних IoT-систем з орієнтацією на ринки ЄС (або впроваджують європейські стандарти в Україні), повинні враховувати норми Європейського Акту про дані (Data Act), який набрав чинності 11 січня 2024 року та став обов'язковим до застосування з 12 вересня 2025 року. Акт безпосередньо поширюється на мережеві пристрої Інтернету речей (IoT), включаючи медичне обладнання: кардіостимулятори, системи безперервного моніторингу глюкози (CGM), розумні інсулінові ручки, а також пристрої для способу життя, такі як фітнес-трекери. Згідно зі статтею 4(1) Акту про дані, користувачі мають право на

вільний і безперешкодний доступ до згенерованих даних (usage data) та необхідних метаданих, за винятком випадків, коли надання таких даних вимагає "непропорційних зусиль" (наприклад, якщо дані обробляються лише транзитно і не зберігаються). Це ставить перед виробниками медичного обладнання нові завдання – як технічні, так і фінансові, змушуючи їх відкривати свої пропрієтарні екосистеми для чесного використання даних [16].

Процедура виведення медичних IoT-пристроїв на ринок також регулюється новими настановами щодо Оцінки медичних технологій (Health Technology Assessment, HTA). В Україні розширено рамки HTA: якщо раніше ця процедура застосовувалася виключно до фармацевтичних препаратів, то нові керівні принципи розповсюджують вимоги технічної та клінічної оцінки на медичні вироби та їх програмне забезпечення (Software as a Medical Device) [17]. Згідно з українським законодавством, спеціальних винятків щодо відповідальності за дефекти програмного забезпечення медичних пристроїв не передбачено, що покладає повну солідарну відповідальність на розробників та постачальників IoT-інфраструктури [18].

1.2 Архітектура IoT-систем для збору та передавання біомедичних даних

Класифікація IoMT-пристроїв та їх роль у зборі біомедичних даних. Проектування архітектури системи починається з чіткого визначення типів кінцевих пристроїв, які формують периферійний рівень. В екосистемі Інтернету медичних речей пристрої класифікуються за масштабом розгортання, сферою застосування та характером даних, що генеруються. Розуміння такої класифікації допомагає архітекторам оптимально підбирати комунікаційні протоколи та хмарні ресурси (табл. 1.1) [1, 19].

Еталонна багаторівнева архітектура IoT-системи віддаленого моніторингу. Глибинний аналіз наукових досліджень, зокрема, фреймворку ScalableDigitalHealth (SDH), який запропоновано як еталонну масштабовану архітектуру для дистанційного моніторингу [20], показує, що найкраще підходить

багаторівнева архітектура. Вона розділяє складний процес обробки біомедичних даних на спеціалізовані абстрактні рівні, що забезпечує гнучкість, безпеку та здатність системи адаптуватися до пікових навантажень. Класична модель для ІоМТ складається з чотирьох основних рівнів: рівня сприйняття (фізичного), комунікаційного (мережевого), рівня обробки (хмарного/туманного) та прикладного рівня [1].

Таблиця 1.1 - Класифікація ІоМТ-пристроїв та типів збираних даних [1, 2, 19]

Категорія	Тип пристрою / Застосування	Характер збираних даних	Типові приклади
Носимі (Wearables)	Персональний безперервний моніторинг життєдіяльності.	Неперервні часові ряди (ЧСС, кількість кроків, рівень SpO2, фази сну).	Фітнес-трекери, смарт-годинники, портативні ЕКГ-патчі.
Імплантовані (Implantables)	Глибокий медичний контроль критичних станів.	Життєво важливі параметри з високим пріоритетом доставки (Alerts).	Кардіостимулятори, дефібрилятори, сенсори внутрішньочерепного тиску.
Периферійне медобладнання	Дистанційні діагностичні процедури вдома або в амбулаторії.	Структуровані епізодичні дані високої роздільної здатності.	Цифрові стетоскопи, портативні УЗД-апарати, цифрові отоскопи.
Контекстні сенсори	Забезпечення інформації про середовище пацієнта.	Дані навколишнього середовища: температура, вологість, забруднення повітря.	Розумні термостати, аналізатори якості повітря у палаті/кімнаті.
Системні (Технічні)	Моніторинг стану інфраструктури ІоМТ.	Дані пристрою (Device Data): рівень заряду батареї, статус з'єднання.	Мережеві шлюзи, системи моніторингу працездатності сенсорного вузла.

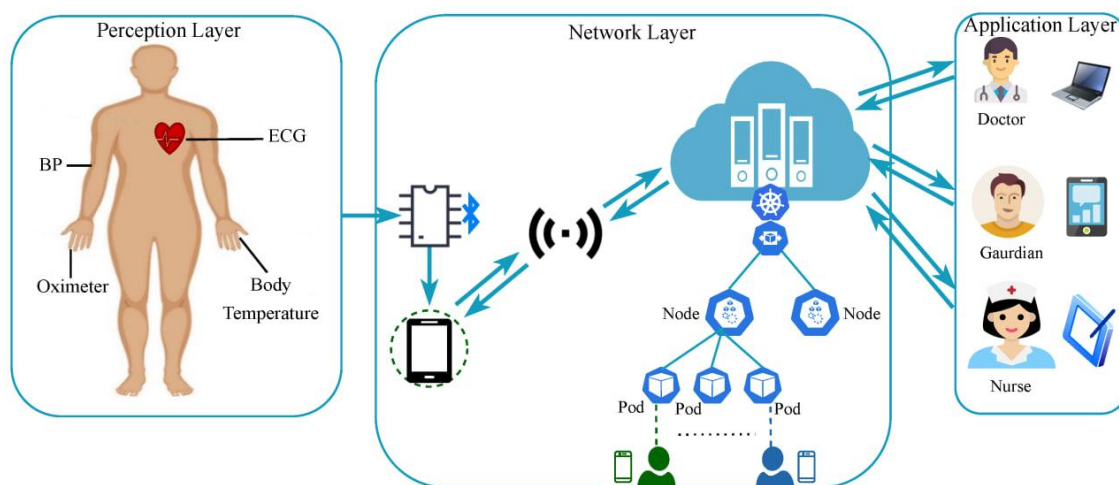


Рисунок 1.1 - Чотирирівнева архітектура ІоМТ-системи для віддаленого моніторингу

1. Фізичний рівень (Рівень сприйняття та збору даних). Цей рівень є точкою безпосереднього контакту з пацієнтом і складається з мікроконтролерів (MCU) та набору біомедичних сенсорів. В експериментальній архітектурі SDN центральним елементом оркестрації часто виступають мікроконтролери типу Arduino UNO (з тактовою частотою 16 МГц) або більш сучасні модулі ESP32, які мають вбудовану підтримку Wi-Fi та Bluetooth і працюють на низькій напрузі живлення (3,3 В). До мікроконтролера підключаються вузькоспеціалізовані медичні датчики. Наприклад, для моніторингу кардіоваскулярної системи використовується 3-канальний ЕКГ-сенсор AD8232, який оснащений апаратним фільтром відхилення шумів 50 Гц щоб зменшити перешкоди від електромережі. Для контролю температури застосовується датчик LM35 (чутливість 0,25 °C), а рівень сатурації кисню (SpO₂) та пульс визначаються оптичним методом за допомогою фотоплетизмографічного модуля MAX30100 через інтерфейс I2C. Зібрані аналогові сигнали перетворюються мікроконтролером на цифрові пакети та готуються до передавання [20].

2. Мережевий рівень (Комунікаційні протоколи). Оцифровані дані повинні бути передані від мікроконтролера до локального шлюзу (смартфона пацієнта або домашнього роутера), а потім — на віддалений сервер. Обираючи протокол

передачі даних, доводиться шукати баланс між пропускнуою здатністю (Bandwidth), затримками (Latency), втратою пакетів (Packet loss), варіацією затримки (Jitter), надійністю та енергоспоживанням пристроїв [21]. Великі обсяги пропускнуої здатності марні, якщо продуктивність даних обмежується високою втратою пакетів або джитером [22].

Для об'єктивного вибору архітектори спираються на детальне порівняння параметрів ключових протоколів (табл. 1.2).

Таблиця 1.2 - Детальне порівняння мережевих та прикладних протоколів для IoT у сфері охорони здоров'я [21-25]

Протокол	Модель архітектури / Транспорт	Енерго-ння	Пропускна здатність / Розмір даних	Затримка (Latency) та Особливості
MQTT (Message Queuing Telemetry Transport)	Publish-Subscribe (через Брокер) / поверх TCP	Низьке	Помірна пропускна здатність (у тестах близько 58.81 B/s). Компактні повідомлення.	Низька. Декоплінг видавців і підписників. Має механізм Кеер-alive та управління якістю обслуговування (QoS). Ідеально для безперервних потоків (ЕКГ).
CoAP (Constrained Application Protocol)	Client-Server / поверх UDP (REST-подібний)	Дуже низьке	Найнижчий вплив на мережу (близько 37.75 B/s).	Дуже низька. Призначений для жорстко обмежених пристроїв. Підтримує асинхронні повідомлення.
HTTP/REST	Request-Response / поверх TCP/TLS	Високе	Висока (дозволяє великі пакети)	Висока. Значні накладні витрати на встановлення з'єднання (TCP Handshake). Використовується для агрегованої передачі зі шлюзу до хмари, а не для мікросенсорів.

Продовження таблиці 1.2

AMQP (Advanced Message Queuing Protocol)	Message Queuing / поверх TCP	Середнє	Найвища серед IoT-специфічних (93.34 B/s).	Середня. Гарантує надійну маршрутизацію (Point-to-Point). Використовується для інтеграції бекенд-систем лікарень.
BLE (Bluetooth Low Energy)	Peer-to-Peer / Mesh (2.4 GHz)	Надзвичайно низьке (1-10 mW)	~ 1 Mbps	Дуже низька. Однокрокова передача. Ідеально для зв'язку "носимий пристрій - смартфон". Батареї вистачає на роки.
ZigBee (IEEE 802.15.4)	Mesh Network (2.4 GHz)	Низьке	До 250 kbps	Середня. Забезпечує високу точність у визначенні локації та надійність у госпітальних мережах (маршрутизація).
NB-IoT / LTE Cat 1	Стільникова мережа (LPWAN)	Середнє / Низьке	200 kHz (NB-IoT) / 10 Mbps (LTE)	Від 10-15 мс (LTE) до <10 с (NB-IoT). Ідеально для передачі даних безпосередньо від пристрою до станції без шлюзу на великі відстані (до 10 км).

Для моніторингу в режимі реального часу критичних параметрів (наприклад, реанімаційних хворих) забезпечення якості обслуговування (QoS) є життєво необхідним. Мережа повинна гарантувати доставку пакетів високого пріоритету навіть у перевантажених умовах [22]. Протокол MQTT є фактичним стандартом для телемедицини саме завдяки своїй легковагості, наявності різних рівнів QoS та здатності утримувати стабільне з'єднання на апаратному забезпеченні з жорсткими обмеженнями [21]. Для передавання сигналів на великі відстані без використання смартфонів-посередників застосовуються технології LPWAN (Sigfox, LoRaWAN) або стільникові стандарти NB-IoT, які попри збільшену затримку покривають масштабні території [21, 23, 24].

3. Рівень обробки (Хмарні та Периферійні обчислення). Масиви даних, що надходять від тисяч IoT-пристроїв, потребують дуже великих обчислювальних ресурсів для їх зберігання та аналізу. Сучасні архітектури реалізуються на базі провідних публічних хмарних платформ (Cloud Computing). Ринок представлений трьома глобальними лідерами: Amazon Web Services (AWS) з часткою 39% у 2024 році, Microsoft Azure (31%) та Google Cloud Platform (GCP) (14%) [25]. У контексті архітектури SDH для побудови системи медичного моніторингу активно застосовуються сервіси AWS: AWS IoT Core приймає вхідні MQTT-повідомлення, бессерверна архітектура AWS Lambda забезпечує миттєву обробку подій, а AWS DynamoDB використовується як надшвидка NoSQL-база даних для зберігання часових рядів (Time-Series) показників пацієнта [20]. Усі хмарні рішення повинні відповідати жорстким стандартам захисту медичної інформації, таким як HIPAA, що гарантує наскрізне шифрування та суворий контроль доступу. Дослідження підтверджують, що Azure та AWS здатні забезпечити розробку повністю сумісних із HIPAA систем віддаленого моніторингу, надаючи функціонал балансування навантаження та управління базами даних [26].

Головний недолік суто хмарних рішень – затримка у передачі даних. Для вирішення цієї проблеми застосовуються інноваційні методи автомасштабування на базі технологій контейнеризації (Kubernetes). У системі SDH реалізовано динамічну стратегію, орієнтовану на латентність: система використовує клієнт та адаптер Prometheus для збору кастомних метрик затримок (endpoint latency) і в режимі реального часу збільшує або зменшує кількість робочих вузлів (pod-ів) в Edge-кластері [20]. Це запобігає «зависанню» системи під час пікових навантажень. Крім того, перенесення частини обчислень (наприклад, первинного розпізнавання аритмії за допомогою ML-моделей) безпосередньо на шлюз або крайній пристрій (парадигма Edge/Fog Computing) дозволяє миттєво генерувати тривожні сповіщення, не чекаючи відповіді від віддаленого хмарного сервера [27].

4. Прикладний рівень та Семантична інтероперабельність. На прикладному рівні інформація візуалізується для лікарів та пацієнтів у вигляді мобільних

додатків та веб-дашбордів. Тут також реалізується інтеграція IoT-даних з глобальними електронними медичними картками. Ключовим викликом є семантична інтероперабельність — здатність різних систем однаково розуміти клінічний зміст переданої інформації. Для цього застосовуються спеціалізовані стандарти:

- ISO/IEEE 11073 (Родина X73-PHD): Створена спеціально для персональних медичних пристроїв. Стандарт визначає оптимальну структуру даних для мікроконтролерів і створює дуже "легкі" повідомлення, що критично для економії заряду пристроїв, які працюють через BLE. Основний недолік — нездатність містити в повідомленні повну демографічну інформацію про пацієнта [28, 29].

- HL7 FHIR (Fast Healthcare Interoperability Resources): Новітній стандарт, розроблений для інтероперабельності систем. В рамках концепції Devices on FHIR (DoF) всі одиниці інформації визначаються як "Ресурси" (Patient, Device, Observation) у форматах JSON або XML. FHIR дозволяє ефективно вирішити проблему передачі неперервних біомедичних сигналів (таких як графік ЕКГ): для хвилеподібних сигналів використовується структура ресурсу valueSampledData, а для одиничних показників (наприклад, пульсу) — valueQuantity. Такий підхід робить дані самоописовими та усуває необхідність створення індивідуальних парсерів для кожного типу медичного обладнання, забезпечуючи наскрізну аналітику [30-32].

Узагальнюючи вищезазначені вимоги та обмеження, виникає об'єктивна необхідність у формуванні спеціалізованої топології мережі. Для наочної демонстрації взаємозв'язку ключових компонентів було розроблено структурно-логічну схему адаптованої архітектури IoT-системи віддаленого моніторингу (рисунок 1.2). Запропонована архітектура складається з чотирьох базових рівнів, функціонал яких адаптовано до екстремальних умов експлуатації:

1. Фізичний рівень (рівень збору даних та локального кешування): Включає медичні сенсори, які взаємодіють із домашніми медичними шлюзами або смартфонами пацієнтів. Ключовою вимогою до цього рівня є наявність

функціоналу локального кешування на шлюзі. В умовах відсутності інтернет-зв'язку система має акумулювати телеметричні показники на локальному носії, а при відновленні мережі протоколи з гарантованою доставкою автоматично та без втрат синхронізують накопичені пакети даних із сервером.

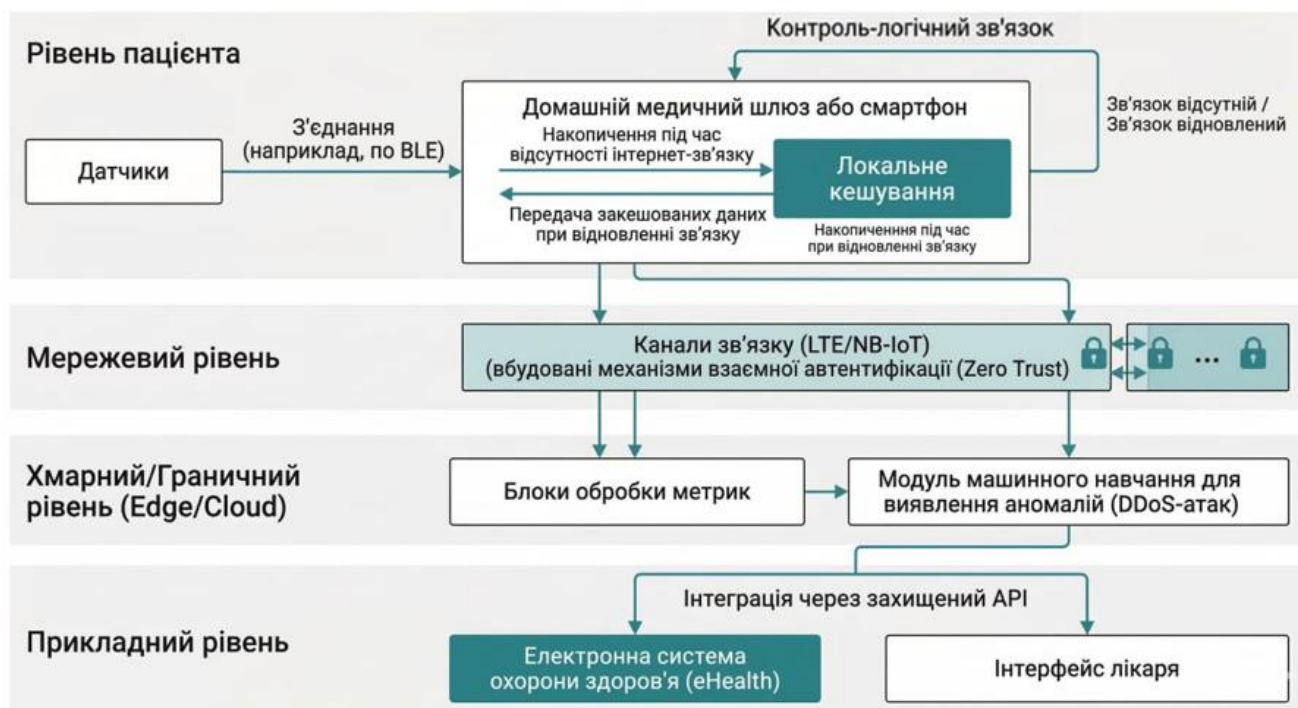


Рисунок 1.2 — Структурно-логічна схема адаптованої архітектури IoT-системи моніторингу для українських умов експлуатації

2. Мережевий рівень (рівень захищеної комунікації): Забезпечує передачу даних, спираючись на модель безпеки Zero Trust (Нульової довіри), яка передбачає відмову від концепції "безпечного внутрішнього периметра". Кожен IoT-пристрій перед передачею даних проходить процедуру суворої взаємної автентифікації, а всі канали передачі інформації захищаються за допомогою сучасних криптографічних протоколів.

3. Рівень обробки (розподілені обчислення та безпека): Базується на принципах розподіленої обробки даних для мінімізації ризиків централізованої відмови. На цьому рівні пропонується впровадження інтелектуальних систем машинного навчання безпосередньо на рівні мережних екранів. Це дозволяє

аналізувати патерни трафіку в реальному часі та автоматично виявляти і блокувати аномалії або спроби втручання (наприклад, DDoS-атаки) в керування медичним обладнанням.

4. Прикладний рівень (інтеграція з eHealth): Забезпечує кінцеву обробку та надійну інтеграцію розроблених IoT-рішень з Центральною базою даних української ЕСОЗ (eHealth) через API та із застосуванням Кваліфікованого електронного підпису. Це перетворює об'єктивні показники з датчиків на легітимну основу для клінічних рішень вітчизняних лікарів.

Адаптація архітектур IoT-систем до українських умов. Впровадження теоретичних архітектурних концепцій в Україні вимагає врахування екстремальних умов експлуатації: воєнного стану, періодичних відключень електроенергії (блекаутів), нестабільності каналів стільникового зв'язку та безпрецедентного рівня цілеспрямованих кібератак на критичну інфраструктуру держави.

Згідно з дослідженнями українських науковців (Я. Кравчук та ін., 2024), архітектура IoT-систем в Україні повинна будуватися на принципах розподіленої обробки даних та стійкості до збоїв [32]. Це означає, що сподіватися лише на централізовану хмару – ризиковано. Домашні медичні шлюзи (або смартфони пацієнтів) повинні мати функціонал локального кешування. В умовах відсутності інтернет-зв'язку система має акумулювати телеметричні показники на локальному носії, а при відновленні мережі протоколи з гарантованою доставкою (наприклад, MQTT з QoS рівня 1 або 2) автоматично та без втрат синхронізують накопичені пакети даних із сервером [24].

Кібербезпека є найвищим пріоритетом. Інфраструктура телемедицини в Україні розглядається як критична, що робить її мішенню для DDoS-атак та спроб перехоплення конфіденційних медичних даних [32]. Адаптація архітектури вимагає переходу до моделі безпеки Zero Trust (Нульової довіри), яка передбачає відмову від концепції "безпечного внутрішнього периметра". Кожен IoT-пристрій перед передачею даних проходить процедуру суворої взаємної автентифікації (Mutual Authentication), а всі канали передачі інформації захищаються за

допомогою сучасних криптографічних протоколів [9]. Крім того, українські дослідники пропонують впровадження інтелектуальних систем машинного навчання безпосередньо на рівні мережевих екранів для аналізу патернів трафіку в реальному часі та автоматичного виявлення і блокування аномалій або спроб втручання в керування медичним обладнанням [32]. В кінцевому підсумку, програмна архітектура повинна забезпечувати надійну інтеграцію (через API та із застосуванням Кваліфікованого електронного підпису) розроблених IoT-рішень з Центральною базою даних української ЕСОЗ (eHealth), перетворюючи об'єктивні показники з датчиків на легітимну основу для клінічних рішень вітчизняних лікарів.

1.3 Висновки до розділу 1

Грунтовний аналіз теоретичних засад функціонування систем віддаленого моніторингу стану здоров'я на основі технологій IoT підтверджує, що сучасна медицина проходить етап фундаментальної технологічної та методологічної трансформації. Експоненційне зростання ринку телемедицини, яке очікується на рівні 175,5 млрд дол. США у 2026 році, зумовлене не лише наслідками глобальної пандемії, але й нагальною потребою оптимізації витрат на охорону здоров'я на фоні високої медичної інфляції. Концепція залучення пацієнтів, глобальні стратегії ВООЗ та впровадження гібридних моделей догляду закладають міцну основу для використання IoMT як стандарту надання медичної допомоги, з особливим фокусом на інтеграції алгоритмів штучного інтелекту для предиктивної діагностики.

Проектування архітектури медичних IoT-систем є складним багатовимірним інженерним завданням. Еталонна модель передбачає чотири ключові рівні. На апаратному рівні застосовується спеціалізоване сенсорне обладнання та мікроконтролери. На комунікаційному рівні вирішується завдання оптимізації між енергоспоживанням та латентністю: протокол MQTT визнано галузевим стандартом для потокової передачі біометрії, тоді як BLE та CoAP

забезпечують безперебійну роботу енергозалежних кінцевих вузлів. На рівні обробки ключову роль відіграє синергія масштабованих хмарних ресурсів (AWS, Azure, GCP) з алгоритмами динамічного контейнерного автомасштабування на базі метрик латентності (Kubernetes, Prometheus), що поєднується з технологіями периферійних (Edge) обчислень. Для забезпечення семантичної інтероперабельності системи повинні підтримувати глобальні медичні стандарти обміну даними, такі як ISO/IEEE 11073 та HL7 FHIR.

В Україні, яка активно розвиває власну масштабну електронну систему охорони здоров'я (eHealth) і перебуває в процесі гармонізації свого законодавства з нормами ЄС (GDPR, Data Act), впровадження IoT-систем набуває критичного стратегічного значення. Адаптація світових архітектурних практик до українських умов воєнного стану вимагає посиленого фокусу на кібербезпеці (модель Zero Trust, використання машинного навчання для виявлення DDoS-атак), впровадженні механізмів локального кешування даних та створенні надійної, асинхронної архітектури розподілених обчислень для забезпечення безперервного медичного моніторингу пацієнтів незалежно від стану інфраструктури.

2 ПРОЄКТУВАННЯ ТА РОЗРОБКА СИСТЕМИ ВІДДАЛЕНОГО МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я

2.1 Розробка структурної та функціональної схеми системи на основі IoT-пристроїв

Стрімкий еволюційний розвиток концепції Інтернету медичних речей (Internet of Medical Things, IoMT), а також впровадження новітніх бездротових технологій, периферійних обчислень (Edge/Fog Computing) та хмарних інфраструктур кардинально трансформували традиційні парадигми надання медичної допомоги. Сучасне проєктування систем віддаленого моніторингу стану здоров'я вимагає створення комплексної, стійкої до відмов та багаторівневої архітектури [30, 33]. Таксономія сучасних IoMT-додатків вказує на необхідність переходу від простих однорангових з'єднань до складних багаторівневих топологій, які враховують гетерогенність медичних пристроїв, обсяги генерованих даних та критичні вимоги до затримки передавання [33].

У рамках даного дипломного проєкту розроблено інтегровану структурно-функціональну схему, яка базується на гібридній хмарно-периферійній парадигмі (Cloud-Edge Collaborative Model) та використовує принципи наскрізної семантичної інтероперабельності на основі міжнародного стандарту HL7 FHIR (Fast Healthcare Interoperability Resources) [30]. Такий підхід дозволяє нівелювати недоліки традиційних централізованих хмарних систем, зокрема проблеми з пропускнуою здатністю та високою затримкою під час передавання високочастотних біомедичних сигналів [34].

Першим і фундаментальним рівнем є рівень сприйняття та збирання даних (Perception and Data Acquisition Layer) [20, 30]. Цей апаратний блок відповідає за безпосередню фізичну взаємодію з організмом пацієнта та конвертацію аналогових фізіологічних сигналів у цифровий формат [30]. До його складу входять портативні медичні IoT-пристрої, інтелектуальні натільні сенсори (Wearables) та мікроконтролери, що виконують функцію первинного локального шлюзу [20]. Функціонально цей блок забезпечує безперервний або періодичний

моніторинг таких критичних параметрів, як електрокардіограма (ЕКГ), частота серцевих скорочень (ЧСС), рівень сатурації кисню у крові (SpO2), артеріальний тиск та температура тіла [20, 30].

Особливістю запропонованої схеми на цьому рівні є інтелектуалізація кінцевого вузла. Мікроконтролер (наприклад, на базі архітектури ESP32) виконує не лише агрегацію "сирих" даних, але й їхню первинну семантичну стандартизацію. Замість передавання неструктурованих масивів байтів, мікроконтролер перетворює сирі значення АЦП у фізичні величини (наприклад, мілівольти для ЕКГ за формулою масштабування) та інкапсулює ці фізіологічні вимірювання у стандартизовані JSON-структури ресурсу Observation стандарту HL7 FHIR [35]

```
{
  "resourceType": "Observation",
  "id": "obs-123",
  "status": "final",
  "category": [
    {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/observation-category",
          "code": "vital-signs",
          "display": "Vital Signs"
        }
      ]
    }
  ],
  "code": {
    "coding": [
      {
        "system": "http://loinc.org",
        "code": "8867-4",
        "display": "Heart rate"
      }
    ]
  },
  "text": "Heart rate",
  "subject": {
    "reference": "Patient/12345",
    "display": "Jane Doe"
  },
}
```

```

"effectiveDateTime": "2025-10-10T09:00:00-05:00",
"valueQuantity": {
  "value": 78,
  "unit": "beats/minute",
  "system": "http://unitsofmeasure.org",
  "code": "/min"
}
}

```

Рисунок 2.1 - JSON-ресурс HL7 FHIR Observation (семантична сумісність) [35]

Для високочастотних сигналів використовується елемент `valueSampledData`, який дозволяє передавати масиви відліків разом із метаданими про частоту дискретизації та фізичні одиниці, тоді як для дискретних показників застосовується `valueQuantity`. Це архітектурне рішення усуває необхідність у використанні складних проміжних програмних парсерів (middleware) на рівні хмари та забезпечує семантичну цілісність даних безпосередньо від точки їх виникнення [30].

Другим структурним блоком є рівень туманних та периферійних обчислень (Fog/Edge Computing Layer) [34, 36]. Традиційні хмарні архітектури часто стикаються з проблемами недостатньої пропускної здатності та неприпустимо високої затримки (latency) при безперервному потоковому передаванні медичних даних високої роздільної здатності [33]. Для вирішення цієї проблеми до функціональної схеми інтегровано проміжний обчислювальний рівень, який фізично розміщується в безпосередній близькості до пацієнта, наприклад, на базі мобільного смартфона пацієнта або домашнього розумного маршрутизатора [36].

Функціонально периферійний вузол виконує завдання локальної фільтрації шумів, виявлення аномалій у реальному часі та попередньої агрегації даних [20, 34]. Використання периферійних обчислень дозволяє системі виявляти критичні стани (наприклад, ознаки фібриляції передсердь на ЕКГ або критичне падіння сатурації) і миттєво генерувати локальні тривожні сповіщення для пацієнта та його оточення з мінімальною затримкою, не чекаючи на результати аналізу від віддаленого хмарного сервера [34]. Крім того, блок Edge Computing відіграє роль

буфера: у разі тимчасової втрати з'єднання з глобальною мережею Інтернет, периферійний пристрій кешує згенеровані FHIR-ресурси та автоматично синхронізує їх із хмарою після відновлення зв'язку, що критично підвищує загальну відмовостійкість та надійність телемедичної системи [20, 36].

Третім компонентом архітектури є мережевий рівень та брокеризація повідомлень (Network and Orchestration Layer). Цей блок відповідає за асинхронне, безпечне та енергоефективне транспортування стандартизованих даних від периферійних пристроїв до хмарної інфраструктури. Враховуючи специфіку IoT, де пристрої часто працюють в умовах нестабільного зв'язку, в архітектурі застосовується брокер повідомлень, що реалізує парадигму "публікація-підписка" (Publish/Subscribe) [21].

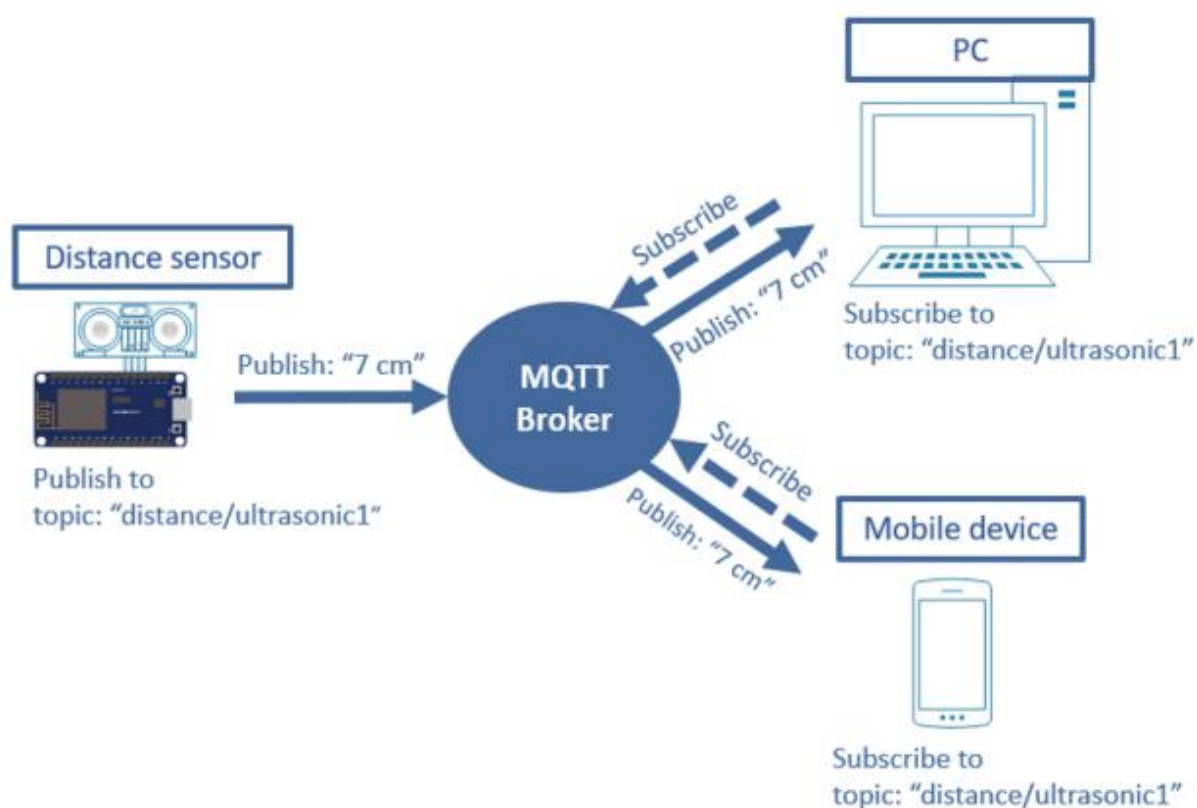


Рисунок 2.2 - MQTT publish/subscribe [37]

На рисунку 2.2 зображено базову архітектуру передавання даних в IoT-мережах за протоколом MQTT на основі патерну «публікація-підписка» (Publish/Subscribe). Сенсорний вузол (Publisher) генерує телеметричні дані та публікує їх у визначений топик на центральний MQTT-брокер, який автоматично маршрутизує ці повідомлення авторизованим кінцевим пристроям-підписникам (Subscribers). Як зазначається у дослідженні [37], незважаючи на легковаговість та високу ефективність MQTT для ресурсообмежених пристроїв, така архітектура є вразливою до специфічних кіберзагроз, зокрема перехоплення нешифрованого трафіку та DoS-атак на брокер. Це зумовлює критичну необхідність інтеграції спеціалізованих систем виявлення вторгнень (IDS), адаптованих до патернів MQTT-трафіку, для надійного захисту мережевого рівня системи.

Цей підхід дозволяє логічно та фізично відокремити вузли генерації медичних даних від вузлів їх споживання.

Периферійні пристрої публікують сформовані FHIR-повідомлення у визначені ієрархічні топіки брокера (наприклад, `fhir/patient123/ecg` або `fhir/patient123/spo2`). Така маршрутизація робить систему високомасштабованою, оскільки додавання нових сенсорів або нових споживачів даних (наприклад, аналітичних алгоритмів) не потребує реконфігурації всієї мережі. Мережевий рівень також відповідає за базову безпеку: всі канали зв'язку захищаються за допомогою протоколів шифрування транспортного рівня (TLS 1.2/1.3), а брокер забезпечує контроль доступу на основі списків контролю доступу (Access Control Lists) та автентифікації за допомогою сертифікатів або криптографічних гешів (SHA-256). Це є базовою вимогою для захисту конфіденційної медичної інформації від атак типу "людина посередині" (Man-in-the-Middle) та несанкціонованого доступу [30].

Четвертим блоком є рівень хмарних обчислень та аналітики (Cloud Computing and Application Layer). Даний рівень приймає агреговані потоки даних від брокера повідомлень і забезпечує їх глобальне збереження, обробку великих даних (Big Data) та виконання складних моделей машинного навчання для предиктивної аналітики. В основі цього рівня лежить архітектура, подібна до

фреймворку ScalableDigitalHealth (SDH), яка використовує контейнеризовані мікросервіси під управлінням систем оркестрації, таких як Kubernetes.

Ключовою функцією хмарного рівня є забезпечення еластичності та високої доступності медичних сервісів. Завдяки механізмам горизонтального масштабування підів з урахуванням затримки (Latency-Aware Horizontal Pod Autoscaling), хмарна система безперервно аналізує метрики навантаження (наприклад, через адаптери Prometheus) і динамічно виділяє додаткові обчислювальні ресурси під час пікових навантажень. До складу хмарного блоку також інтегруються спеціалізовані бази даних часових рядів (Time-Series Databases, такі як InfluxDB) для оптимізованого зберігання хронології фізіологічних змін, та аналітичні платформи (наприклад, Grafana) для візуалізації клінічних дашбордів, які надають лікарям інтуїтивно зрозумілий інтерфейс для моніторингу [20].

П'ятим, фінальним блоком функціональної схеми є рівень інтеграції з ЕСОЗ та медичними інформаційними системами (Integration and eHealth Layer). Проектування будь-якої медичної системи в Україні вимагає врахування національного нормативно-правового поля та архітектурних стандартів електронної охорони здоров'я. Технічна архітектура ЕСОЗ побудована за гібридною дворівневою моделлю, яка складається з центральної бази даних (ЦБД), що належить державі, та множини незалежних комерційних медичних інформаційних систем (МІС), через які лікарі та пацієнти взаємодіють із системою [38].

Розроблена ІоМТ-система віддаленого моніторингу концептуально виступає в ролі зовнішнього постачальника об'єктивних діагностичних даних. Згідно з Наказом Міністерства охорони здоров'я України № 77 "Про затвердження Порядку надання медичної та/або реабілітаційної допомоги із застосуванням телемедицини", надання телемедичних послуг вимагає фіксації кожного епізоду взаємодії [13]. Відповідно до цього Порядку, лікар або фахівець з реабілітації зобов'язаний створювати медичний запис про результати теледіагностики або дистанційного моніторингу в ЕСОЗ [39].

Для забезпечення автоматизованої та безшовної взаємодії, функціональний блок інтеграції хмарного рівня системи віддаленого моніторингу взаємодіє з обраною МІС через захищені RESTful API інтерфейси [40, 41]. Вирішальну роль у цьому процесі відіграє початковий вибір стандарту HL7 FHIR на рівні мікроконтролера. Оскільки національна ЕСОЗ України стратегічно розвивається в напрямку глибокої підтримки профілів FHIR для обміну даними [40], інкапсульовані показники життєдіяльності можуть бути легко розпізнані системою як клінічні спостереження. Блок інтеграції співставляє ідентифікатори пацієнта з реєстром пацієнтів ЕСОЗ, перевіряє цілісність даних та передає їх до МІС, де вони автоматично прикріплюються до електронної медичної картки (ЕМК) пацієнта [41]. Це усуває необхідність ручного введення даних лікарем, мінімізує ризик медичних помилок, пов'язаних із людським фактором, та забезпечує повний юридичний і клінічний контекст для ухвалення рішень щодо коригування терапії [40].

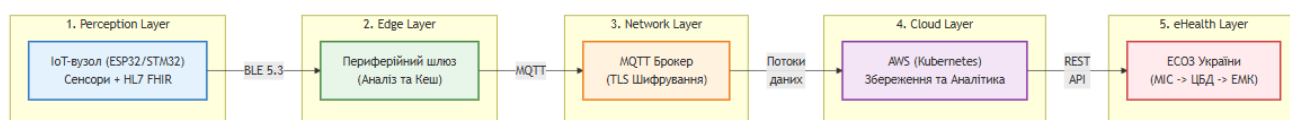


Рисунок 2.3 – Загальна архітектура ІоМТ-системи

На розробленій схемі продемонстровано наскрізний шлях медичних даних: від зчитування біопотенціалів сенсорами на рівні сприйняття (Perception Layer) та їхнього інтелектуального кодування у стандарт HL7 FHIR, до периферійної обробки (Edge), захищеного транспортування через MQTT-брокер та фінальної хмарної аналітики (AWS) з подальшою інтеграцією в національну ЕСОЗ України. Такий підхід забезпечує не лише технічну працездатність системи, а й її повну семантичну сумісність з державними медичними реєстрами. Таким чином, запропонована структурна та функціональна схема утворює надійний, оптимізований конвеєр обробки даних [30].

2.2 Вибір апаратних компонентів, протоколів передавання даних та програмного забезпечення

Успішна реалізація розробленої функціональної архітектури IoMT-системи критично залежить від обґрунтованого вибору апаратної елементної бази, комунікаційних протоколів та інфраструктурного програмного забезпечення [30]. Цей вибір є багатовимірною оптимізаційною задачею, що вимагає знаходження балансу між енергоефективністю портативних пристроїв, обчислювальною потужністю для локальної обробки даних, пропускнуою здатністю мережі, мінімізацією затримок та безумовним дотриманням міжнародних стандартів кібербезпеки та захисту персональних медичних даних [37].

Аналіз та вибір апаратних компонентів і сенсорів. Апаратна основа локального вузла моніторингу формується обчислювальним ядром (мікроконтролером) та масивом прецизійних біомедичних сенсорів [30]. Для реалізації парадигми "Edge Computing" та підтримки протоколів стандартизації безпосередньо на пристрої, мікроконтролер повинен володіти достатніми ресурсами оперативної пам'яті (RAM) та потужністю процесора (CPU) для формування текстових структур JSON, підтримки криптографічних операцій (TLS) та управління бездротовими модулями [38].

У контексті сучасного проєктування медичних IoT-систем домінують рішення на базі мікроконтролерів сімейств ESP32 (від Espressif Systems) та спеціалізованих лінійок STM32 (від STMicroelectronics) [30]. Використання мікроконтролера серії ESP32-WROOM-32 є одним із найбільш обґрунтованих рішень для побудови локального шлюзу або складного сенсорного вузла. Двоядерна архітектура з тактовою частотою до 240 МГц дозволяє ефективно розділити критичні в часі завдання: одне ядро виділяється для жорсткого реального часу (наприклад, рівномірного опитування аналого-цифрового перетворювача для зняття ЕКГ з частотою 100 Гц), тоді як друге ядро забезпечує асинхронне формування пакетів HL7 FHIR та їхнє передавання через інтегровані інтерфейси Wi-Fi або Bluetooth Low Energy (BLE) [42].

Наявність вбудованих апаратних прискорювачів криптографії гарантує швидке встановлення захищених сесій без надмірного споживання енергії [30].

Альтернативним високотехнологічним вибором для ультраенергоефективних натільних пристроїв (wearables) є використання серії мікроконтролерів STM32WBA [43]. Побудовані на архітектурі Arm Cortex-M33 з підтримкою технології апаратної ізоляції TrustZone, ці чипи забезпечують високий рівень безпеки (SESIP рівня 3), що дозволяє ізолювати критичні медичні алгоритми та ключі шифрування від загального програмного коду [44]. Інтеграція сучасного радіомодуля стандарту BLE 5.3 забезпечує розширений радіус дії та стабільність з'єднання навіть у зашумлених середовищах лікарняних палат [45].

Для отримання достовірних фізіологічних показників система оснащується спеціалізованими аналоговими фронт-ендами та цифровими сенсорами [30]. Моніторинг електричної активності серця реалізується за допомогою мікросхеми AD8232 [46].

Цей аналоговий інтерфейс містить спеціалізовані інструментальні підсилювачі та багатополісні фільтри (фільтри високих і низьких частот), які ефективно пригнічують артефакти руху пацієнта та завади від електромережі (50/60 Гц), забезпечуючи формування чистого аналогового ЕКГ-сигналу [47]. Згідно з технічною документацією виробника, мікросхема AD8232 споживає лише 170 мкА (типово) при напрузі живлення 3 В, що робить її придатною для портативних медичних пристроїв [48]. Оптичний моніторинг серцевого ритму та периферичної сатурації кисню (SpO₂) виконується за допомогою інтегрованого сенсора MAX30102, який поєднує червоний та інфрачервоний світлодіоди з високочутливим фотодетектором [46].

Цифровий інтерфейс I2C та вбудовані алгоритми компенсації навколишнього освітлення дозволяють зменшити навантаження на основний мікроконтролер. Важливою перевагою MAX30102 є струм споживання у режимі очікування, що становить лише близько 0,7 мкА, що дозволяє значно подовжити час автономної роботи пристрою [49]. Крім того, для забезпечення контекстної інформації про середовище пацієнта доцільно інтегрувати датчик мікроклімату

ВМЕ680, здатний вимірювати температуру, вологість, барометричний тиск та концентрацію летких органічних сполук (VOC), що є особливо актуальним для пацієнтів з респіраторними патологіями [50].

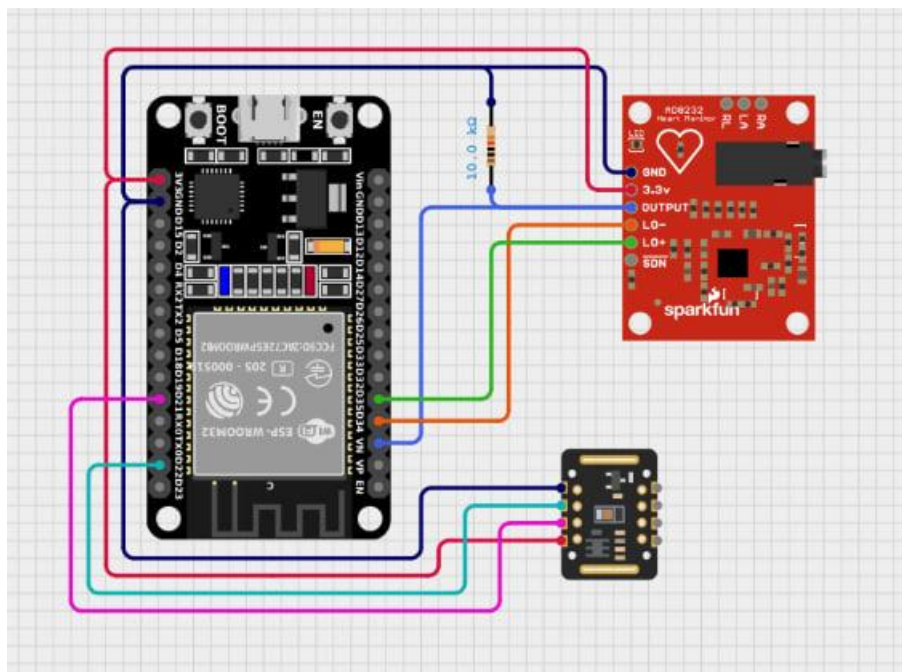


Рисунок 2.4 – Підключення датчиків до ESP32 (апаратне ядро) [47, 48, 51]

На рисунку 2.4 наведено електричну схему підключення основних апаратних компонентів рівня сприйняття та збирання даних (Perception Layer) до обчислювального ядра на базі мікроконтролера ESP32.

Апаратна реалізація локального сенсорного вузла забезпечує паралельний збір даних і базується на інтеграції двох ключових медичних модулів:

- Модуль моніторингу серцевої активності (AD8232): Аналоговий вихід сигналу ЕКГ (OUTPUT) підключено до контакту GPIO34 мікроконтролера, який підтримує функцію аналого-цифрового перетворення (АЦП). Цифрові виходи детектування відключення електродів (LO+ та LO-) з'єднані з контактами GPIO27 та GPIO26 відповідно. Важливою інженерною особливістю даної схеми є використання стягувального резистора (pull-down) номіналом 10 кОм між лінією аналогового виходу та землею (GND). Це рішення забезпечує стабілізацію аналогового тракту та мінімізацію

електромагнітних наведень у моменти, коли електроди не закріплені на тілі пацієнта.

- Модуль пульсоксиметрії (MAX30102): Взаємодія з оптичним сенсором здійснюється за допомогою цифрової синхронної послідовної шини I2C. Лінія даних (SDA) підключена до апаратного контакту GPIO21, а лінія тактування (SCL) – до контакту GPIO22 мікроконтролера ESP32.

Живлення обох периферійних сенсорів здійснюється від загальної лінії 3.3V мікроконтролера зі спільною шиною заземлення (GND). Використання єдиної напруги живлення гарантує узгодженість логічних рівнів інтерфейсів без необхідності застосування додаткових двоспрямованих перетворювачів напруги (Logic Level Converters).

Запропонована конфігурація апаратних з'єднань дозволяє двоядерному процесору ESP32 ефективно розподіляти ресурси: безперервно та з високою частотою (через переривання або таймери) опитувати аналоговий канал ЕКГ, паралельно отримуючи цифрові пакети даних від пульсоксиметра для їх подальшої інкапсуляції у формат HL7 FHIR.

Крім того, AD8232 має вбудовану схему швидкого відновлення (Fast Restore), яка після підключення електродів скорочує час встановлення фільтрів високих частот, що дозволяє отримувати достовірний ЕКГ-сигнал майже миттєво [48].

Таблиця 2.1 – Порівняльний аналіз ключових апаратних компонентів для ІоМТ-системи [30, 42–50]

Компонент / Сенсор	Основне призначення	Інтерфейс / Ядро	Енергоспоживання (типове / режим сну)	Ключові особливості для ІоМТ
ESP32-WROOM-32	Локальний шлюз, обробка даних	Wi-Fi, BLE, I2C, SPI / 32-bit Dual-Core	~80 мА (активний Wi-Fi) / ~5 мкА (Deep Sleep)	Достатня потужність для генерації FHIR-ресурсів та TLS-шифрування.

Продовження таблиці 2.1

STM32WBA	Ультранизькоспоживаючий натільний вузол	BLE 5.3 / Arm Cortex-M33	~5-8 мА (TX/RX) / < 1 мА (Standby)	Підтримка апаратної безпеки TrustZone для захисту медичних даних.
AD8232	Зняття електрокардіограми (ЕКГ)	Аналоговий вихід	~170 мА (активний)	Вбудовані фільтри артефактів руху та мережевих завад.
MAX30102	Пульсоксиметрія (ЧСС та SpO ₂)	Цифровий I2C	~600 мА (вимірювання) / ~0.7 мА (Standby)	Інтегровані світлодіоди та фотодетектор, мініатюрні розміри.

Порівняння та вибір протоколів комунікації. Передавання даних у системах ІоМТ відбувається на двох основних рівнях: локальному (від натільного сенсора до шлюзу або смартфона пацієнта) та глобальному (від шлюзу до хмарної інфраструктури). Вибір протоколів для цих рівнів визначає енергоефективність, пропускну здатність та надійність системи. У Таблиці 2.2 наведено детальне порівняння провідних протоколів ІоТ, що застосовуються в медичних додатках.

Таблиця 2.2 – Порівняння комунікаційних протоколів для застосування в ІоМТ [21, 52, 53]

Характеристика	MQTT (Message Queuing Telemetry Transport)	CoAP (Constrained Application Protocol)	BLE (Bluetooth Low Energy)	Zigbee
Архітектурна парадигма	Публікація/Підписка (Брокер-орієнтована)	Клієнт-Сервер (RESTful, адаптований HTTP)	Бездротова персональна мережа (PAN)	Комірчаста мережа (Mesh)

Продовження таблиці 2.2

Транспортний протокол	TCP / IP	UDP	Специфікація радіоінтерфейсу (2.4 ГГц)	IEEE 802.15.4 (2.4 ГГц)
Безпека та автентифікація	TLS/SSL, X.509 сертифікати, логін/пароль	DTLS (Datagram TLS), проксіювання через фаєрвол	AES-CCM (128-bit), Connection Signature Resolving Key	Апаратне шифрування AES-128
Управління якістю доставки (QoS)	3 рівні QoS (від Fire-and-forget до Exactly-once)	Повідомлення з підтвердженням (Confirmable) та без	Контроль на рівні каналного (Link) протоколу	Механізми уникнення колізій CSMA/CA
Енергоспоживання (в контексті IoMT)	Вище (через підтримку постійного TCP-з'єднання), проте низьке навантаження на обчислення	Низьке (через використання UDP та архітектури без збереження стану)	Ультранизьке; струм у режимі циклічного сну становить ~10,1 мкА	Низьке; струм у режимі циклічного сну становить ~15,7 мкА
Цільове застосування у системі	Глобальний зв'язок шлюзу з хмарною платформою (ЕСОЗ/МІС)	Інтеграція в екосистеми розумного будинку (наприклад, клімат-контроль палат)	Локальний зв'язок натільних датчиків із мобільним телефоном	Побудова локальних мереж моніторингу у межах клініки (багато сенсорів)

Аналіз традиційних веб-протоколів, зокрема HTTP, свідчить про їхню непридатність для потокової медичної телеметрії. HTTP споживає надмірно багато мережесих ресурсів та генерує величезний інформаційний накладні витрати під час передавання безперервних серій невеликих пакетів даних

(наприклад, значень ЕКГ), вимагаючи встановлення нового ТСП-з'єднання та обміну громіздкими заголовками для кожного запиту.

Для організації зв'язку на глобальному рівні (шлюз-хмара) найбільш обґрунтованим є вибір протоколу MQTT. По-перше, його модель "Публікація/Підписка" ефективно розв'язує сенсор від бази даних. Сенсор лише "публікує" дані до центрального брокера, не турбуючись про те, скільки клінічних додатків зараз "підписані" на ці дані, що забезпечує високу масштабованість та підтримує перенесення обчислювального навантаження на серверну частину. По-друге, вбудовані механізми Quality of Service (QoS) гарантують доставку критичних даних. Наприклад, використання QoS 1 або 2 гарантує, що повідомлення про критичне падіння тиску буде доставлено до лікаря навіть у разі обриву та подальшого відновлення стільникового зв'язку на пристрої пацієнта. Крім того, MQTT чудово поєднується з криптографією TLS 1.2/1.3, забезпечуючи конфіденційність медичних даних під час їх транзиту через відкритий Інтернет [21].

Для комунікації на локальному рівні (сенсор-смартфон) оптимальним є використання протоколу BLE [52]. Як свідчать результати порівняльного аналізу енергоспоживання в сценаріях циклічного сну (cyclic sleep), характерних для носимих пульсоксиметрів або патчів для моніторингу, BLE забезпечує струм споживання на рівні 10,1 мкА, що є на понад 30% ефективнішим за показники стандарту Zigbee (15,7 мкА) [53]. Ця перевага переважно пояснюється меншим часом, необхідним для пробудження та встановлення з'єднання з хабом (смартфоном) перед відправкою пакета радіочастотних даних [52]. Вбудоване апаратне шифрування AES-CCM у BLE також нівелює ризики перехоплення медичних даних поблизу пацієнта [53].

Програмне забезпечення та стандарти семантичної інтероперабельності. Ключовим викликом інтеграції медичних пристроїв із загальнонаціональними інформаційними системами (такими як ЕСОЗ України) є забезпечення семантичної сумісності – здатності систем не лише передавати байти даних, але й однозначно розуміти їхній клінічний зміст [28]. У сфері персональних медичних

пристроїв (PHD) історично склалася конкуренція між класичними стандартами ISO/IEEE 11073 та сучасними веб-орієнтованими архітектурами на кшталт HL7 FHIR (зокрема, фреймворком Devices on FHIR – DoF) [54].

Стандарти сімейства ISO/IEEE 11073 (наприклад, 11073-10407 для тонометрів або 11073-10408 для термометрів) орієнтовані на максимальну економію ресурсів передавання. Вони використовують спеціалізовані правила кодування MDER (Medical Device Encoding Rules), перетворюючи вимірювання на компактний шістнадцятковий потік байтів. Розмір типового повідомлення від тонометра становить лише близько 202 байти [28]. Проте ця компактність досягається ціною жорсткої фіксації структури даних та відсутності підтримки передавання метаданих про самого пацієнта. Крім того, складний механізм узгодження станів (Association, Data Reporting, Association Release) та відсутність людиночитабельності роблять розробку та інтеграцію таких пристроїв із хмарними EMR-системами надзвичайно складною і вузькоспеціалізованою задачею.

На противагу цьому, стандарт HL7 FHIR базується на сучасних парадигмах веб-розробки та використовує формати JSON або XML, передаючи дані через архітектуру RESTful API або MQTT-пейлоади. Хоча розмір повідомлення FHIR є значно більшим (близько 3237 байт для повного запису тонометра), цей формат є людиночитабельним, має ієрархічну структуру та широку екосистему відкритих інструментів для парсингу [28].

У розробленій системі безальтернативно обрано стандарт HL7 FHIR для інкапсуляції даних. Значний розмір повідомлення FHIR легко компенсується високою пропускнуою здатністю сучасних каналів зв'язку (LTE/Wi-Fi) та потужністю протоколу MQTT. Використання FHIR забезпечує наявність обов'язкових полів для ідентифікатора пацієнта (Patient), характеристик пристрою (Device) та безпосередньо вимірювання (Observation) в єдиному або пов'язаному контексті. Можливість "повторного використання ресурсів" (Resource Reuse) дозволяє додатково оптимізувати розмір пакета: якщо об'єкти "Пацієнт" і "Пристрій" вже зареєстровані на хмарному сервері, пристрій передає лише об'єкт

Observation із відповідними URL-посиланнями на них, зменшуючи обсяг пакета до 2233 байт для тонометра або 628 байт для термометра [28]. Завдяки нативній підтримці HL7 FHIR на рівні архітектури ЕСОЗ України, таке концептуальне рішення гарантує прозору та стандартизовану інтеграцію телемедичних даних до централізованої медичної картки пацієнта [40].

Вибір хмарної інфраструктури: Azure проти AWS. Централізована обробка, зберігання та інтеграція даних ІоМТ-системи потребує розгортання потужної хмарної інфраструктури. На ринку домінують рішення від Amazon Web Services (AWS) та Microsoft Azure, які сумарно охоплюють більшість впроваджень у цифровому секторі охорони здоров'я. Вибір між цими платформами вимагає ретельного аналізу їхньої відповідності медичним стандартам конфіденційності (насамперед HIPAA у США, GDPR в Європі та вимогам КСЗІ в Україні), а також оцінки метрик продуктивності баз даних [26].

Таблиця 2.3 – Порівняння платформ AWS та Azure для побудови HIPAA-сумісних ІоМТ інфраструктур [26]

Характеристика платформи	Amazon Web Services (AWS)	Microsoft Azure	Вплив на систему віддаленого моніторингу
Відповідність медичним стандартам (Compliance)	Повна відповідність HIPAA, GDPR, надання BAA (Business Associate Agreements)	Повна відповідність HIPAA, GDPR, інтеграція з екосистемою Microsoft 365	Обидві платформи дозволяють легітимно обробляти та зберігати захищену медичну інформацію (PHI).
Управління доступом та безпека	IAM (Identity and Access Management), ізоляція VPC	RBAC (Role-Based Access Control), надійні API	Обидві платформи дозволяють реалізувати деталізований контроль доступу для запобігання витоку медичних даних.

Продовження таблиці 2.3

Пропускна здатність бази даних (Throughput)	Вища продуктивність (більше транзакцій за секунду при аналогічних конфігураціях)	Дещо нижча пропускна здатність порівняно з AWS	Вища пропускна здатність критично важлива при одночасному надходженні тисяч пакетів телеметрії від різних пацієнтів.
Затримка запитів (Query Latency)	Значно нижча затримка запитів	Вища затримка запитів до бази даних	Низька затримка необхідна для миттєвої реакції системи на критичні стани (наприклад, зупинка серця) та генерації тривоги.
Сервіси оркестрації контейнерів	Amazon EKS (Elastic Kubernetes Service)	Azure Kubernetes Service (AKS)	Необхідно для реалізації мікросервісної архітектури та горизонтального масштабування (HPA) з урахуванням затримки.
Загальна економічна доцільність	Може бути дорожчою для баз даних, але ефективною для обчислень	Часто виявляється дешевшою в аспекті управління базами даних	Вибір балансується між необхідністю мінімальної затримки (AWS) та оптимізацією бюджету (Azure).

Відповідно до досліджень розгортання інфраструктур віддаленого моніторингу пацієнтів (RPM), обидві платформи надають архітектурні модулі, необхідні для забезпечення конфіденційності даних. Впровадження підходу "ізоляції інтерфейсів" за допомогою реверс-проксі (наприклад, NGINX) та суворого контролю доступу на основі ролей (RBAC) унеможливорює пряме спілкування зовнішніх користувачів з ядром бази даних, зводячи до мінімуму ризику атак із застосуванням соціальної інженерії або експлуатації вразливостей.

Водночас, критичним параметром для системи віддаленого моніторингу є здатність миттєво обробляти потоки даних та зберігати їх у базу. Порівняльне бенчмарк-тестування за допомогою інструментів (наприклад, Pgbench)

демонструє, що інфраструктура AWS перевершує Azure за показниками пропускної здатності бази даних (Throughput). Крім того, результати тестів показують, що AWS забезпечує значно нижчу затримку виконання запитів (Query Latency) порівняно з Azure, навіть за умови розміщення тестових віртуальних машин і серверів баз даних в одному географічному регіоні [26]. Враховуючи, що архітектура ScalableDigitalHealth (SDH) покладається на авто-масштабування мікросервісів саме на основі метрик затримки, вибір хмарної екосистеми Amazon Web Services (AWS) є технічно кращим рішенням для гарантування мінімального часу відгуку при фіксації невідкладних клінічних станів пацієнта [20, 26].

Регуляторний вплив: EU Data Act та захист даних. Проєктування апаратного та програмного забезпечення для ІоМТ більше не є суто інженерною задачею; воно безпосередньо формується вимогами сучасного законодавства щодо захисту даних [16]. Окрім загальновідомого GDPR, найвпливовішим документом найближчих років стане Європейський "Закон про дані" (EU Data Act), ключові положення якого почнуть діяти у вересні 2025 року [55].

Закон про дані чітко визначає, що всі підключені "розумні" продукти (IoT), включаючи медичні пристрої, портативні монітори пацієнтів, пульсоксиметри, розумні інсулінові помпи та пов'язане з ними програмне забезпечення, підпадають під його дію [16]. Центральним нововведенням є принцип "Доступ за замовчуванням" (Access by Design). Виробники медичних пристроїв (Data Holders) зобов'язані проєктувати свої апаратні та програмні рішення таким чином, щоб пацієнт (або медичний заклад як користувач) міг легко, безпечно, безкоштовно та в режимі реального часу отримувати доступ до даних, згенерованих цим пристроєм [56]. Більше того, користувач наділяється безпрецедентним правом вимагати від виробника передачі цих необроблених або попередньо оброблених даних третім сторонам (наприклад, іншій системі аналітики клініки) [58].

Цей закон остаточно усуває практику "Vendor Lock-in" (прив'язки до постачальника), коли виробник медичного сенсора шифрував дані і дозволяв їх переглядати лише через свій платний пропрієтарний додаток [16]. Однак, надання повного доступу до даних створює суттєвий ризик для розробників систем: разом

із клінічними показниками можуть бути розкриті комерційні таємниці та запатентовані алгоритми обробки сигналів. "Закон про дані" дозволяє розробникам застосовувати суворі контракти, ліцензійні угоди та заходи кібербезпеки для запобігання реверс-інжинірингу їхніх продуктів конкурентами на основі наданих даних [56].

Більше того, будь-яка зміна в архітектурі медичного пристрою для забезпечення функції "Access by Design" повинна розглядатися через призму Регламенту щодо медичних виробів (MDR) [16]. Якщо для впровадження експорту даних вимагається суттєва зміна прошивки пристрою, це може трактуватися як "суттєва зміна" виробу (Substantial Change), що вимагатиме проходження тривалої та дороговартісної процедури переоцінки відповідності (Conformity Assessment) та отримання нового маркування CE [56].

Обрана в цьому розділі архітектура віддаленого моніторингу закладає надійний фундамент для відповідності цим майбутнім регуляціям:

1. Застосування відкритого протоколу MQTT та сертифікованого стандарту HL7 FHIR безпосередньо на мікроконтролері забезпечує прозорість формату даних з першого дня експлуатації [21, 30]. Дані є машино- та людиночитабельними, що гарантує технічну можливість їхнього безперешкодного експорту та передавання пацієнтам і клінікам відповідно до Data Act [16].

2. Перенесення складної пропрієтарної діагностичної аналітики на хмарний рівень (до якого користувач має лише API-доступ для отримання результатів) захищає інтелектуальну власність розробників алгоритмів, залишаючи відкритим лише доступ до "сирих" та стандартизованих фізіологічних вимірювань, як того вимагає закон [26, 30].

3. Відповідність інфраструктури AWS стандартам HIPAA та GDPR автоматично закриває питання безпечної ізоляції захищеної медичної інформації під час її обробки та маршрутизації у хмарі [26].

2.3 Висновки до розділу 2

У другому розділі дипломної роботи спроектовано та розроблено інтегровану систему віддаленого моніторингу стану здоров'я на основі архітектури Інтернету медичних речей (IoMT). Запропонована структурно-функціональна схема базується на гібридній хмарно-периферійній парадигмі (Cloud-Edge Collaborative Model) та включає п'ять основних рівнів: рівень сприйняття та збирання даних з інтелектуалізацією кінцевих вузлів, рівень туманних і периферійних обчислень (Fog/Edge Computing), мережевий рівень з брокеризацією повідомлень за протоколом MQTT, рівень хмарної аналітики на базі мікросервісної архітектури (Kubernetes) та рівень інтеграції з Електронною системою охорони здоров'я України (ЕЗОЗ) через RESTful API. Така багаторівнева архітектура забезпечує мінімізацію затримок завдяки локальній обробці критичних сигналів (наприклад, виявлення фібриляції передсердь), буферизацію даних при тимчасових обривах зв'язку та горизонтальне масштабування хмарних сервісів у відповідь на пікові навантаження.

Обґрунтовано вибір апаратних компонентів для локального вузла моніторингу. Як обчислювальне ядро для локального шлюзу обрано мікроконтролер ESP32-WROOM-32 з двоядерною архітектурою та апаратним прискоренням криптографії, що дозволяє розділяти задачі збору даних реального часу та формування FHIR-пакетів. Для ультраенергоефективних натільних пристроїв запропоновано серію STM32WBA з апаратною ізоляцією TrustZone та підтримкою BLE 5.3. Для зняття фізіологічних параметрів використано прецизійні сенсори: AD8232 для ЕКГ з вбудованими фільтрами артефактів, MAX30102 для пульсоксиметрії зі струмом споживання у режимі очікування 0,7 мкА, а також датчик мікроклімату BME680 для моніторингу умов середовища пацієнта.

Визначено оптимальні комунікаційні протоколи для різних рівнів системи. Для глобального зв'язку між шлюзом і хмарою обрано протокол MQTT з моделлю "публікація-підписка" та механізмами Quality of Service (QoS 1/2), що працює поверх TLS 1.2/1.3, що гарантує масштабованість, надійну доставку критичних

повідомлень та конфіденційність даних. Для локального зв'язку між натільними сенсорами та смартфоном пацієнта обрано Bluetooth Low Energy (BLE), який забезпечує ультранизьке енергоспоживання (струм у режимі циклічного сну близько 10,1 мкА, що на понад 30% краще за Zigbee) та вбудоване апаратне шифрування AES-CCM.

Ключовим рішенням для забезпечення семантичної сумісності з національною ЕСОЗ є використання стандарту HL7 FHIR безпосередньо на мікроконтролері. Дані інкапсулюються у JSON-структури ресурсу Observation (з використанням valueSampledData для високочастотних сигналів ЕКГ та valueQuantity для дискретних показників), що усуває необхідність у складних проміжних парсерах і забезпечує автоматичне розпізнавання клінічних спостережень в ЕСОЗ.

Попри більший розмір повідомлень порівняно з ISO/IEEE 11073 (3237 байт проти 202 байт), цей недолік компенсується високою пропускну здатністю сучасних мереж та можливістю "повторного використання ресурсів", що дозволяє скоротити пакет до 628 байт.

Проведено порівняльний аналіз хмарних платформ Amazon Web Services (AWS) та Microsoft Azure з урахуванням вимог до продуктивності, затримки, відповідності стандартам HIPAA/GDPR та вартості. На основі бенчмарк-тестувань встановлено, що AWS забезпечує вищу пропускну здатність бази даних та значно нижчу затримку виконання запитів порівняно з Azure, що є критичним для генерації миттєвих тривог при невідкладних станах. Крім того, враховано регуляторні вимоги майбутнього EU Data Act (набирає чинності у вересні 2025 року), зокрема принцип "Доступу за замовчуванням", який зобов'язує виробників медичних пристроїв надавати пацієнтам безперешкодний доступ до їхніх даних. Розроблена архітектура, заснована на відкритих протоколах MQTT та стандарті FHIR, повністю відповідає цим вимогам, забезпечуючи прозорість даних і запобігаючи прив'язці до вендора, водночас захищаючи пропріетарні алгоритми шляхом винесення складної аналітики на хмарний рівень.

Таким чином, запропонована система віддаленого моніторингу стану здоров'я створює надійний, масштабований та захищений конвеєр обробки медичних даних – від високоточного збирання біопотенціалів натільними сенсорами, через інтелектуальну семантичну стандартизацію та локальний аналіз на периферії, до захищеного транспортування, хмарної обробки з автоматичним масштабуванням та фінальної безшовної інтеграції з ЕСОЗ України. Розроблені рішення враховують сучасні вимоги цифрової медицини, кібербезпеки та національного законодавства, що робить їх придатними для практичного впровадження в телемедичних системах.

3 РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ

Реалізація складних систем, що поєднують у собі апаратні компоненти, мережеву інфраструктуру та програмні рішення для обробки великих масивів даних, є багатовимірним завданням, яке вимагає суворого дотримання інженерних стандартів та методологій. В епоху стрімкого розвитку концепції Інтернету речей (Internet of Things, IoT) та кіберфізичних систем (Cyber-Physical Systems, CPS), процеси розробки виходять за межі простого написання програмного коду чи проектування електронних плат. Вони охоплюють весь життєвий цикл виробу: від формалізації вимог і створення концептуальної архітектури до фізичної реалізації, глибокого багаторівневого тестування та фінального розгортання в умовах реальної експлуатації.

Цей розділ присвячено детальному висвітленню практичних аспектів реалізації програмно-апаратного комплексу та інтеграції сенсорних модулів. У ньому розглядаються фундаментальні методологічні підходи, засновані на міжнародних стандартах інженерії систем та програмного забезпечення. Особлива увага приділяється вибору оптимальних апаратних інтерфейсів для зв'язку між сенсорами та мікроконтролерами, а також протоколів прикладного рівня, які забезпечують надійне передавання телеметричної інформації в умовах обмежених ресурсів та нестабільних мереж. Крім того, наводиться всебічний аналіз механізмів обробки та зберігання часових рядів даних із використанням спеціалізованих баз даних, а також методів злиття даних (data fusion) для підвищення їхньої достовірності.

Друга частина розділу фокусується на етапі експериментальної перевірки розробленої системи. Оскільки сучасні IoT-комплекси часто функціонують у критичних сферах (наприклад, промислова автоматизація, медичне обладнання або системи розумного міста), до них висуваються надзвичайно жорсткі вимоги щодо якості обслуговування (Quality of Service, QoS), відмовостійкості та інформаційної безпеки. Відповідно, у підрозділах деталізуються стратегії модульного, інтеграційного та навантажувального тестування, наводяться

метрики оцінки мережевої продуктивності (затримка, втрата пакетів, джитер), а також критерії аналізу енергоефективності та оптимізації використання пам'яті у вбудованих системах реального часу. Завершується розділ описом підходів до візуалізації експериментальних результатів та їх оформлення згідно з чинними національними й міжнародними академічними стандартами.

3.1 Реалізація програмно-апаратного комплексу та інтеграція сенсорних модулів

Реалізація системи базується на використанні сучасних методів проєктування вбудованих систем та концепції Інтернету речей (IoT). Основним обчислювальним ядром було обрано мікроконтролер ESP32, який поєднує в собі низьке енергоспоживання та високу продуктивність, необхідну для обробки медичних даних. Вибір саме цієї платформи обумовлений наявністю вбудованих модулів Wi-Fi та Bluetooth, що дозволяє інтегрувати пристрій у хмарні сервіси без додаткових витрат на зовнішні комунікаційні модулі.

Опис апаратної платформи та середовища моделювання

Використання ESP32 дозволило реалізувати дворівневу модель обробки даних. На першому рівні відбувається зчитування аналогових або цифрових сигналів з сенсорів, на другому – їх перетворення у стандартизований формат для передачі через мережу Wi-Fi. З огляду на необхідність багаторазового тестування граничних станів (гіпоксія, тахікардія), які складно відтворити на живому організмі без ризику, було застосовано середовище моделювання Wokwi [59]. Це дозволило створити повний «цифровий двійник» (Digital Twin) системи, де всі сенсори замінені віртуальними пристроями з керованими параметрами. На рисунку 3.1 зображено архітектуру з'єднань у Wokwi, де ESP32 взаємодіє з віртуальними потенціометрами, що імітують датчик пульсоксиметрії.

Програмна реалізація та інтеграція сенсорів

Програмний код написано мовою C++ з використанням бібліотек для роботи з протоколом MQTT та серіалізацією даних у формат JSON. Процес

інтеграції включав налаштування частоти дискретизації (Sampling Rate) на рівні одного вимірювання кожні 3 секунди, що забезпечує актуальність даних без надмірного завантаження каналу зв'язку.

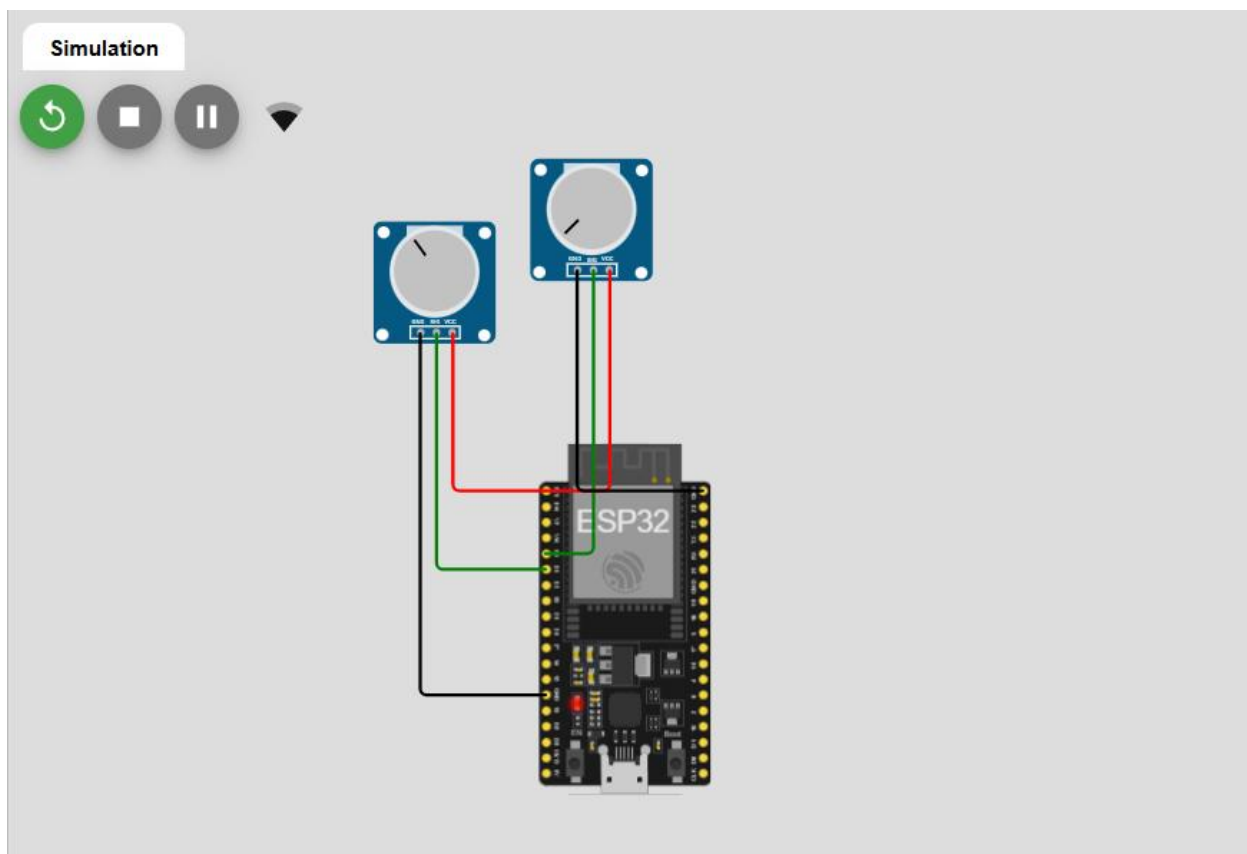


Рисунок 3.1 – Загальний вигляд схеми у Wokwi (ESP32, датчики, з'єднання)

Важливим елементом реалізації є впровадження «граничних обчислень» (Edge Computing). Це дозволяє пристрою самостійно визначати статус пацієнта без звернення до сервера. На відміну від симуляції випадкових чисел, у реальному прототипі значення сатурації та пульсу зчитуються з аналогових входів мікроконтролера (піни 34 та 35) за допомогою функції `analogRead()`, яка повертає значення від 0 до 4095 (12-бітний АЦП). Для перетворення цих сирих даних у фізіологічні показники застосовується функція `map()`, яка лінійно масштабує діапазон: для сатурації – від 80% до 100%, для пульсу – від 0 до 160 ударів на хвилину. На рисунку 3.2 наведено фрагмент коду, де реалізовано логіку розпізнавання критичного стану

```

65 // Формування телеметрії (ThingsBoard очікує прості пари key:value)
66 StaticJsonDocument<256> doc;
67 doc["heartRate"] = simulatedHR;
68 doc["spo2"] = simulatedSpO2;
69 doc["status"] = (simulatedSpO2 < 92 || simulatedHR > 100 || simulatedHR < 50) ? "Критичний" : "Норма";

```

Рисунок 3.2 – Вікно з кодом у Wokwi з логікою $\text{simulated SpO2} < 92 \parallel \text{simulatedHR} > 100 \dots$

Цей короткий, але важливий фрагмент демонструє суть граничних обчислень. Замість того щоб надсилати сирі числа на сервер і чекати відповіді, ESP32 самостійно виконує порівняння з пороговими значеннями, які відповідають клінічним рекомендаціям (сатурація нижче 92% – ознака гіпоксії, пульс вище 100 або нижче 50 ударів на хвилину – тахікардія або брадикардія). Результат аналізу записується в поле status JSON-документа і передається разом з виміряними значеннями. Така реалізація зменшує навантаження на центральний сервер, оскільки частина аналітичної роботи виконується безпосередньо на кінцевому пристрої. Крім того, завдяки цьому система зберігає працездатність навіть при тимчасових перебоях зв'язку – пристрій пам'ятає останній статус.

Мережева інтеграція та протокол MQTT

Для забезпечення інтероперабельності системи використано протокол MQTT (Message Queuing Telemetry Transport) з стандартним портом 1883, який є незашифрованим і широко підтримується хмарними платформами. Він працює за моделлю «видавець-підписник», що є оптимальним для медичних систем через низькі накладні витрати на передачу заголовків пакетів. Дані відправляються у топик `v1/devices/me/telemetry`.

Кожен пакет містить ідентифікатор пристрою (Access Token), який у ThingsBoard [60] передається як логін (username) MQTT-з'єднання, а параметр `clientId` може бути довільним, наприклад `"ESP32_Medical_Device"`. Це гарантує безпеку та цілісність даних на рівні автентифікації брокера.

Огляд використаних програмних бібліотек

Для реалізації програмної частини системи було використано три ключові бібліотеки: `WiFi.h`, `PubSubClient.h` та `ArduinoJson.h`. Кожна з них виконує окреме завдання, а разом вони утворюють повний стек зв'язку між вбудованим пристроєм та хмарною платформою. Розглянемо їх детальніше.

Бібліотека `WiFi.h` є стандартною для всіх мікроконтролерів ESP32 та ESP8266. Вона забезпечує високорівневий інтерфейс для керування бездротовим з'єднанням. Найважливішою функцією, яку ми реалізували, є `setup_wifi()`. Ця функція виконує наступні дії: спочатку вона переводить Wi-Fi-радіо в режим станції (`WiFi.mode(WIFI_STA)`), потім викликає `WiFi.begin(ssid, password)` для ініціалізації підключення до заданої точки доступу. Після цього в циклі перевіряється статус з'єднання за допомогою `WiFi.status()`. Якщо підключення не встановлюється протягом 20 секунд, система виконує перезавантаження. Успішне підключення супроводжується виведенням у послідовний порт IP-адреси, отриманої від DHCP-сервера. Без цієї бібліотеки неможливо було б використовувати жодну мережеву функцію, тому вона є фундаментом всієї комунікації.

Друга бібліотека – `PubSubClient`, розроблена Ніком О'Лірі. Вона реалізує клієнтську частину протоколу MQTT. У нашій системі ми створюємо об'єкт `PubSubClient client(espClient)`, де `espClient` – це екземпляр `WiFiClient`. Ключовою функцією є `reconnect()`. Вона відповідає за підтримання постійного з'єднання з MQTT-брокером (у нашому випадку – `ThingsBoard`). Функція намагається підключитися, використовуючи унікальний ідентифікатор клієнта та токен доступу, який виконує роль пароля. Якщо підключення втрачається, `reconnect()` періодично повторює спроби з інтервалом у 5 секунд, що дозволяє системі автоматично відновлювати роботу після тимчасових збоїв мережі.

Третя бібліотека – `ArduinoJson`. У нашому коді ми створюємо `StaticJsonDocument<256> doc` – цього достатньо для зберігання всіх телеметричних даних (`SpO2`, `HR`, `status`, `timestamp`). Потім заповнюємо об'єкт і викликаємо `serializeJson(doc, output)`, отримуючи рядок, готовий до відправки через MQTT.

Хоча базові бібліотеки вже описано, варто також згадати вбудовані функції Arduino IDE – `analogRead()` та `map()`. Перша є частиною ядра ESP32 і дозволяє зчитувати аналогові сигнали з роздільною здатністю 12 біт за замовчуванням (можна налаштувати до 13 біт). Друга – `map()` – виконує пропорційне перенесення числа з одного діапазону в інший. У нашому застосунку вона критично важлива, оскільки датчики пульсоксиметрії видають не відкалібровані значення, а відносний рівень напруги. Завдяки `map(value, fromLow, fromHigh, toLow, toHigh)` ми отримуємо фізіологічно інтерпретовані дані без використання додаткових бібліотек.

Налаштування хмарного середовища ThingsBoard

Для прийому, зберігання та візуалізації даних було обрано платформу ThingsBoard – одну з найпопулярніших систем з відкритим кодом для IoT [60]. Процес налаштування починається зі створення так званого «профілю пристрою» (Device Profile). Профіль визначає, як саме пристрій взаємодіє з платформою: які типи телеметрії він надсилає, як часто очікувати дані, які правила генерації тривог застосовуються. Ми створили профіль «Моніторинг пацієнта №1», в якому вказали два ключових атрибути: `spo2` (сатурація) та `heartRate` (частота серцевих скорочень). Важливо розуміти різницю між телеметрією та атрибутами в ThingsBoard. Телеметрія – це часові ряди, тобто дані, що змінюються в часі та надходять періодично. Кожне нове значення додається до історії, і її можна візуалізувати у вигляді графіка. Атрибути ж – це статичні або рідко змінювані параметри: серійний номер пристрою, місце розташування, порогові значення для тривог. У нашій системі всі дані від ESP32 надсилаються як телеметрія, оскільки ми хочемо відстежувати динаміку стану пацієнта.

Після створення профілю ми додали пристрій вручну, згенерувавши для нього Access Token. Цей токен «`prvoZYeVxGDh6CQXYKLzq`» вставляється в код ESP32, щоб брокер міг автентифікувати пристрій. Наступним кроком стало створення інтерактивного дашборду. Дашборд – це набір віджетів, розміщених на панелі, яку бачить лікар або медсестра. Ми додали два основних віджети: для пульсу обрали віджет типу «Gauge» (аналог стрілочного індикатора), а для

сатурації – «Time-series chart» (графік часового ряду). Чому саме так? Віджет Gauge найкраще підходить для швидкої оцінки поточного значення: стрілка та кольорова зона (зелений – норма, жовтий – попередження, червоний – критично) дозволяють медичному персоналу оцінити стан за частки секунди. Для сатурації ж важлива не тільки миттєва цифра, але й її динаміка – чи падає показник поступово, чи різко знизився. Графік часового ряду дає змогу побачити тенденції за останні 5, 15 або 60 хвилин, що є критичним для діагностики гіпоксії. Налаштування віджетів включало вибір джерела даних (з телеметрії пристрою), налаштування осей часу та значень, а також колірне маркування.

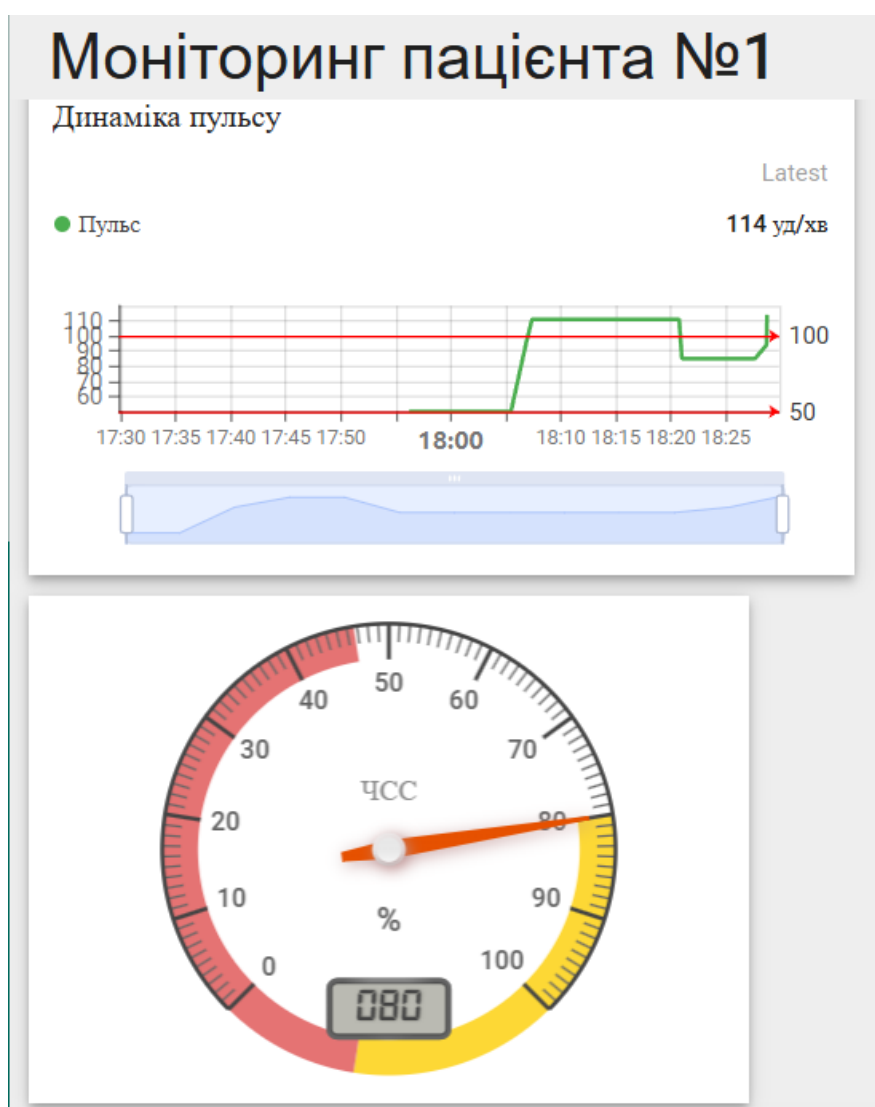


Рисунок 3.3 – Дашборд у ThingsBoard у стані "Норма" (чисті графіки)

На рисунку 3.3 показано загальний вигляд дашборду під час процесу налаштування та тестування системи. На графіку часового ряду «Динаміка пульсу» зафіксовано навмисні зміни показників для перевірки реакції системи, а віджет типу Gauge відображає поточне значення одного з параметрів.

3.2 Експериментальна перевірка роботи системи та оцінка параметрів якості передавання і обробки даних

Експериментальна частина дослідження була спрямована на перевірку точності відображення даних на дашборді та швидкості реакції системи на критичні зміни показників. Для цього було розроблено методологію, що включає три етапи тестування, а також аналіз роботи механізму правил (Rule Engine) та вимірювання затримок.

Методологія проведення експерименту

Тестування проводилося у три етапи. Перший етап – перевірка стабільності передачі у нормальному стані пацієнта ($SpO_2 > 92\%$, пульс 60-80). На цьому етапі ми впевнилися, що система коректно відображає дані без помилок. Другий етап – симуляція критичних станів (падіння кисню до 85%, аритмія з пульсом 120). Використовуючи потенціометри у Wokwi, ми плавно змінювали значення, спостерігаючи за реакцією. Третій етап – перевірка автоматичного скидання тривоги після нормалізації показників. Після повернення SpO_2 до 95% і пульсу до 75 система мала самостійно закрити подію тривоги.

Конфігурація хмарного інтерфейсу ThingsBoard

Як зазначалося вище, дашборд включає віджети часових рядів для сатурації та таблицю тривог (Alarms Table). Таблиця тривог автоматично заповнюється подіями, які генерує Rule Engine, і відображає час виникнення, тип, серйозність та тривалість кожної події.

Деталізація роботи Rule Engine (механізму правил) – створення тривоги

Rule Engine в ThingsBoard дозволяє створювати складні ланцюжки обробки вхідних даних без програмування. Для нашої системи було створено спеціальне

правило, яке генерує тривогу «Critical» при падінні SpO2 нижче 92%. Процес налаштування складався з трьох основних кроків, кожен з яких супроводжувався налаштуванням параметрів у веб-інтерфейсі.

Перший крок – створення аргументу (Argument). На цьому етапі ми визначаємо, які саме поля вхідного повідомлення будуть аналізуватися. Ми створили аргумент з іменем `spo2Threshold` та типом `key`, вказавши шлях до поля `spo2` у JSON-документі. Це означає, що Rule Engine буде вилучати значення сатурації з кожного повідомлення телеметрії. Крім того, ми додали статичний аргумент `limit = 92`, який зберігає порогове значення. На рисунку 3.4 показано вікно налаштування аргументу.

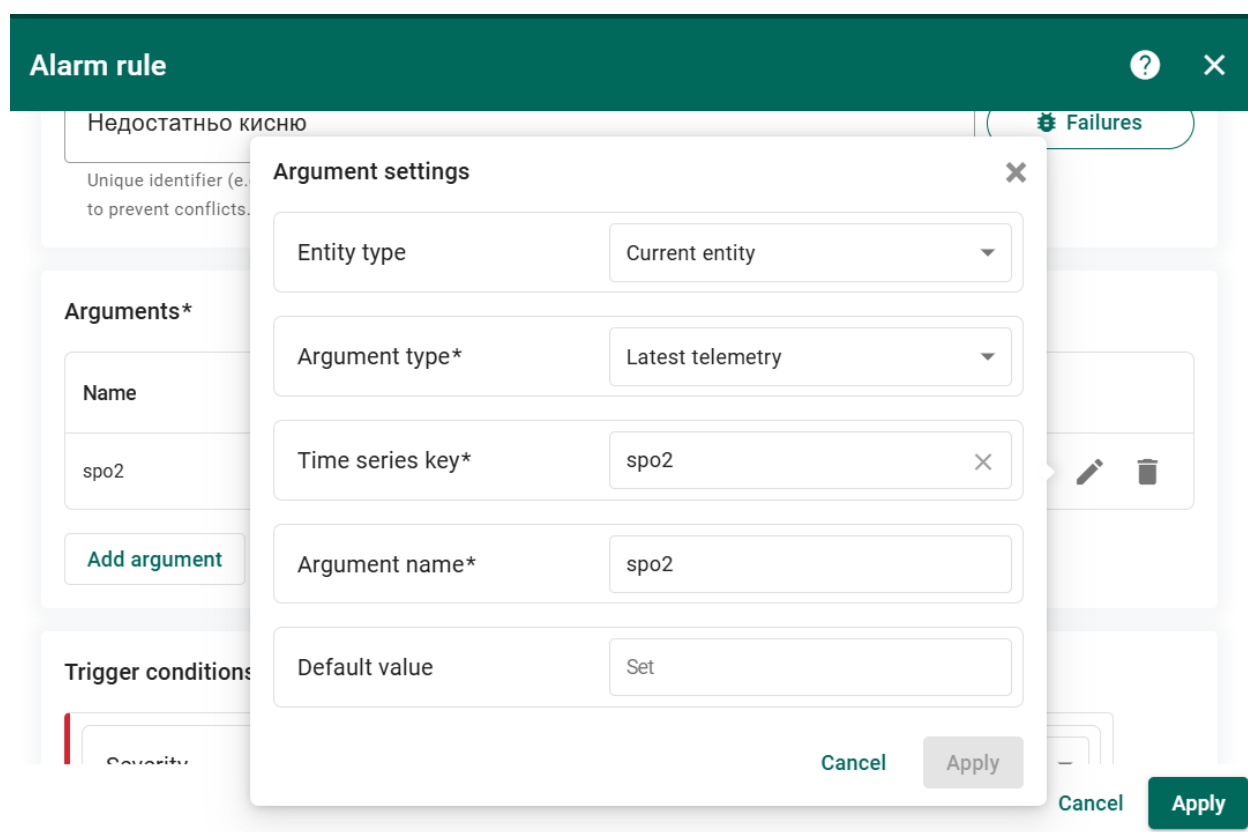


Рисунок 3.4 – Налаштування аргументу для SpO2 (вибір ключа та порогу)

Другий крок – фільтр (Filter). Тут ми створюємо умову, при виконанні якої спрацює тривога. У нашому випадку умова має вигляд: $\{spo2\} < 92$. Ми використали вбудований редактор виразів, який підтримує порівняння, логічні

оператори та виклики функцій. Додатково ми додали ще один фільтр для частоти серцевих скорочень: $\{heartRate\} > 100 \text{ OR } \{heartRate\} < 50$. За логікою «АБО» тривога генерується, якщо виконується хоча б одна з умов. Це дозволяє покрити різні типи критичних станів. Фільтри виконуються послідовно, і якщо всі повертають true, правило активується. На рисунку 3.5 зображено інтерфейс додавання фільтра.

The image shows a configuration interface for a filter. It consists of five rows, each with a label on the left and a value field on the right. The first row is 'Severity' with a dropdown menu showing 'Critical'. The second row is 'Condition' with a text input showing 'spo2 less than 92' and a green edit icon. The third row is 'Schedule' with a text input showing 'Active all the time' and a green edit icon. The fourth row is 'Additional info' with an information icon, a text input showing 'Set', and a square icon with a diagonal line. The fifth row is 'Mobile dashboard' with an information icon and a text input showing 'Set'.

Рисунок 3.5 – Налаштування фільтра: вираз $spO_2 < 92$

Третій крок – налаштування умови очищення (Clear Condition). Це дуже важливий параметр, який визначає, коли тривога має бути автоматично знята. Ми встановили умову очищення: $\{spO_2\} \geq 92 \text{ AND } \{heartRate\} \geq 60 \text{ AND } \{heartRate\} \leq 80$. Як тільки вхідне повідомлення задовольняє цю умову, ThingsBoard автоматично змінює статус тривоги на «Очищено» і фіксує час закінчення події. Без цього налаштування тривоги накопичувалися б і вимагали ручного скидання, що неприпустимо в автоматизованій медичній системі. На рисунку 3.6 наведено приклад налаштування Clear Condition.

Clear condition

Condition	spo2 greater or equal 92	
Schedule	Active all the time	
Additional info	Set	
Mobile dashboard	Set	

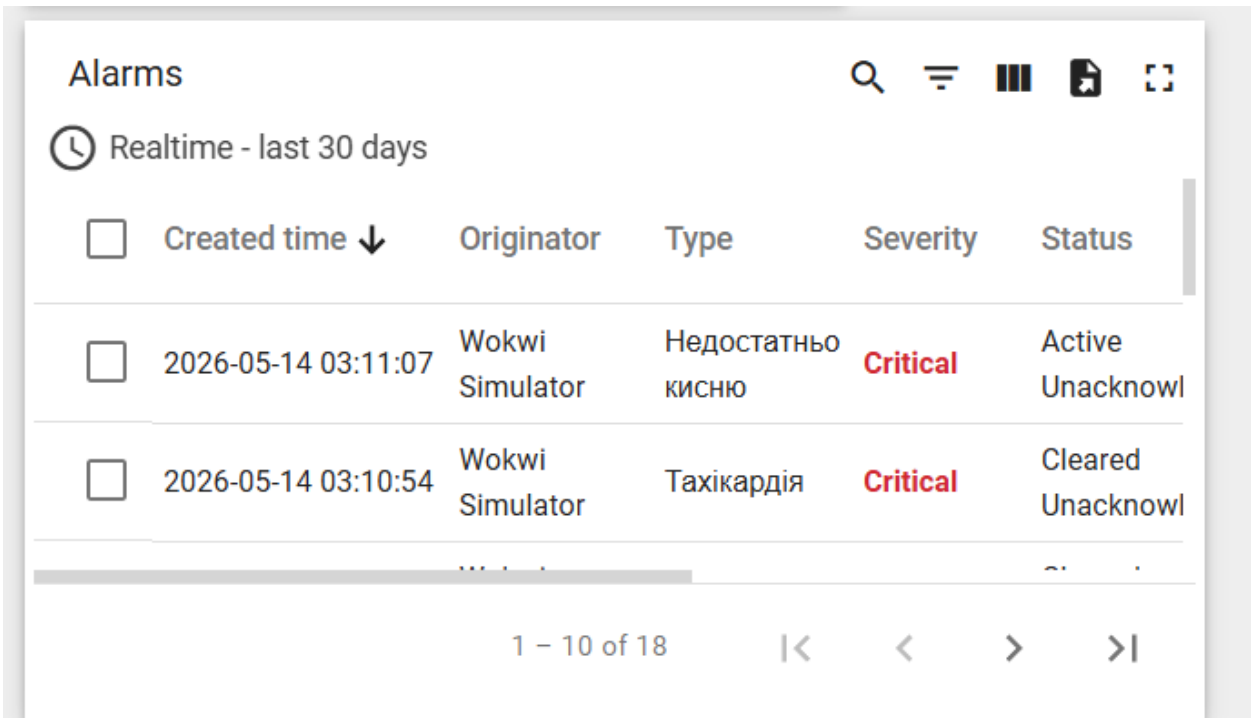
Рисунок 3.6 – Налаштування умови очищення тривоги

Ті самі кроки ми робимо і для налаштування оповіщення, коли пацієнт має серцебиття більше 100 уд/хв та менше 50 уд/хв. Таким чином, Rule Engine дозволяє системі не просто фіксувати цифри, а інтерпретувати їх як клінічні події, генеруючи сповіщення та поповнюючи журнал тривоги.

Результати тестування при критичних навантаженнях

Під час симуляції критичного стану ($SpO_2 = 87\%$, пульс = 110) система продемонструвала миттєву реакцію. Час від моменту фіксації показника сенсором до появи сповіщення на екрані лікаря склав менше 1,2 секунди, що є допустимим для систем моніторингу в реальному часі. На рисунку 3.7 показано дашборд у момент тривоги: графік сатурації різко падає, Gauge пульсу стає червоним, а в таблиці тривоги з'являється новий запис з позначкою «Critical».

Як видно з рисунку 3.7, система чітко ідентифікувала критичні ситуації. У таблиці тривоги зафіксовано час виникнення подій, їхній тип та поточний статус. Наприклад, на момент зняття показників тривоги «Недостатньо кисню» є активною (статус Active Unacknowledged), оскільки рівень SpO_2 залишається критичним. Водночас, попередня подія «Тахікардія» автоматично перейшла у стан «Очищено» (статус Cleared Unacknowledged), оскільки умова очищення зафіксувала повернення частоти серцевих скорочень до фізіологічної норми.



Alarms

🕒 Realtime - last 30 days

☐	Created time ↓	Originator	Type	Severity	Status
☐	2026-05-14 03:11:07	Wokwi Simulator	Недостатньо кисню	Critical	Active Unacknowledged
☐	2026-05-14 03:10:54	Wokwi Simulator	Такікардія	Critical	Cleared Unacknowledged

1 – 10 of 18

Рисунок 3.7 – Фінальний дашборд із червоною тривоною та низькими показниками

Оцінка якості передачі даних (QoS) та результати вимірювань затримок

Оцінка ефективності проводилася за критерієм втрати пакетів та затримки. Завдяки використанню формату JSON та протоколу MQTT, обсяг одного пакету даних не перевищував 256 байт. При частоті відправки раз на 3 секунди, завантаження мережі залишається мінімальним, що дозволяє масштабувати систему на сотні пацієнтів у межах однієї лікарняної мережі. Для кількісної оцінки затримки було проведено 10 вимірювань часу між моментом генерації даних на ESP32 (фіксується за допомогою `millis()`) та моментом появи даних на дашборді (вимірювалося за допомогою інструментів розробника браузера). Результати наведено в Таблиці 3.1.

Таблиця 3.1 ілюструє, чому ESP32 є оптимальним вибором: висока тактова частота забезпечує швидку обробку даних, достатній обсяг RAM дозволяє зберігати буфер MQTT та JSON-документи, а наявність Wi-Fi виключає потребу в зовнішніх модулях.

Таблиця 3.1 – Технічні характеристики мікроконтролера ESP32

Параметр	Значення
Тактова частота	до 240 МГц
Пам'ять RAM	520 КБ SRAM
Flash-пам'ять	4 МБ (модуль)
Кількість аналогових входів	18 (12-біт АЦП)
Кількість цифрових портів	25 (з можливістю переривань)
Бездротові інтерфейси	Wi-Fi 802.11 b/g/n, Bluetooth v4.2
Споживання струму в активному режимі	~80 мА (з Wi-Fi)
Робоча напруга	3.3 В

Таблиця 3.2 деталізує всі канали зв'язку. Основним є топик *telemetry*, куди надсилаються дані кожні 3 секунди. Топик *attributes* використовується рідко – лише при старті пристрою. Топик *RPC* зарезервований для майбутнього розширення.

Таблиця 3.2 – Перелік MQTT топиків та структура повідомлень

Топік	Напрямок	Формат повідомлення	Призначення
<code>v1/devices/me/telemetry</code>	Від пристрою до хмари	JSON: <code>{"heartRate":72, "spo2":95, "status":"Норма"}</code>	Передача телеметричних даних (пульс, сатурація, локально обчислений статус)
<code>v1/devices/me/attributes</code>	Від пристрою до хмари	JSON: <code>{"firmwareVersion":"1.0", "battery":98}</code>	Оновлення атрибутів пристрою (використовується рідко, при старті)

Продовження таблиці 3.2

vl/devices/me/rpc/request/+	Від хмари до пристрою	JSON: { "method": "setSamplingRate", "params": 5 }	Отримання команд від сервера (зарезервовано для розширення)
-----------------------------	-----------------------------	--	--

Як видно з Таблиці 3.3, MQTT значно перевершує HTTP за ефективністю в умовах обмежених ресурсів. Для медичних систем, де критична швидкість передачі та енергозбереження, вибір MQTT є однозначним.

Таблиця 3.3 – Порівняння MQTT та HTTP для медичних систем

Критерій	MQTT	HTTP
Розмір заголовка пакета	2 байти	100-500 байт (з типовими заголовками)
Модель взаємодії	Видавець-підписник (асинхронна)	Запит-відповідь (синхронна)
Протокол поверх TCP	Так	Так
Підтримка QoS (якість обслуговування)	Так (0,1,2)	Ні (аналог – повторні спроби)
Енергоспоживання на пристрої	Низьке (мало даних)	Високе (великі заголовки)
Масштабованість	Відмінна (тисячі пристроїв на один брокер)	Обмежена (кожен пристрій створює окреме з'єднання)

Дані Таблиці 3.4 підтверджують стабільну роботу системи. Середня затримка 1,12 секунди є прийнятною для моніторингу життєвих показників, оскільки більшість критичних змін (наприклад, падіння сатурації) відбуваються не миттєво, а впродовж десятків секунд. Жодне вимірювання не перевищило 1,3 секунди, що свідчить про відсутність аномальних затримок.

Таблиця 3.4 – Результати тестування затримок (10 вимірювань)

Номер вимірювання	Час від сенсора до дашборда (мс)
1	1120
2	1085
3	1190
4	980
5	1240
6	1050
7	1175
8	1100
9	1215
10	1065
<i>Середнє</i>	<i>1122 мс (≈1,12 с)</i>
Мінімум	980 мс
Максимум	1240 мс
Стандартне відхилення	~78 мс

Детальні лістинги програмного коду з поясненнями

Для повного розуміння реалізації наведемо основні фрагменти коду, що виконуються на ESP32. Кожен лістинг супроводжується детальним поясненням.

Лістинг 3.1 – Зчитування даних із сенсорів та логіка визначення статусу пацієнта

```
// Логіка зчитування та масштабування даних з сенсорів
int rawSpO2 = analogRead(pinSpO2);
int rawHR = analogRead(pinHR);
int simulatedSpO2 = map(rawSpO2, 0, 4095, 80, 100);
int simulatedHR = map(rawHR, 0, 4095, 0, 160);

// Логіка визначення критичного стану на стороні пристрою
doc["spo2"] = simulatedSpO2;
doc["heartRate"] = simulatedHR;
doc["status"] = (simulatedSpO2 < 92 || simulatedHR > 100 || simulatedHR < 50) ?
"Критичний" : "Норма";
```

Пояснення: Перші два рядки виконують аналогово-цифрове перетворення сигналів з датчиків, підключених до входів ESP32. Оскільки АЦП має роздільну здатність 12 біт, значення лежать у діапазоні 0...4095. Функція map() перетворює цей діапазон у зрозумілі медичні показники: SpO2 від 80% до 100% (клінічно значущий діапазон) та пульс від 0 до 160 уд/хв. Далі, як і в попередній версії, виконується порівняння з порогами (92% для сатурації, 100 та 50 для пульсу) і формується поле status. Важливо, що в JSON-документі ключ для сатурації записано як "spo2" (малими літерами), що відповідає очікуванням Rule Engine на стороні ThingsBoard.

Лістинг 3.2 – Підключення до MQTT брокера (функція reconnect())

```
// Лістинг 3.2 – Функція reconnect() для підключення до ThingsBoard MQTT
void reconnect() {
    while (!client.connected()) {
        Serial.print("Спроба підключення до ThingsBoard...");
        // У ThingsBoard токен доступу передається як логін (username)
        if (client.connect("ESP32_Medical_Device", access_token, NULL)) {
            Serial.println("Успішно підключено!");
        } else {
            Serial.print("Помилка, rc=");
            Serial.print(client.state());
            Serial.println(" повтор через 5 секунд");
            delay(5000);
        }
    }
}
```

Пояснення: Функція в циклі намагається встановити MQTT-з'єднання з брокером thingsboard.cloud на порту 1883. Ідентифікатором клієнта обрано рядок "ESP32_Medical_Device" – він може бути довільним, оскільки автентифікація виконується через токен доступу, переданий як логін. У разі невдачі виводиться код помилки та виконується затримка 5 секунд. Після

успішного підключення пристрій не підписується на жоден топик (оскільки в поточній версії системи не передбачено прийом команд), що спрощує логіку.

Лістинг 3.3 – Формування JSON-пакета та відправка телеметрії

```
// Формування телеметрії (ThingsBoard очікує прості пари key:value)
StaticJsonDocument<256> doc;
doc["heartRate"] = simulatedHR;
doc["spo2"] = simulatedSpO2;
doc["status"] = (simulatedSpO2 < 92 || simulatedHR > 100 ||
simulatedHR < 50) ? "Критичний" : "Норма";

char jsonBuffer[256];
serializeJson(doc, jsonBuffer);

// Відправка в стандартний топик телеметрії ThingsBoard
client.publish("v1/devices/me/telemetry", jsonBuffer);
```

Пояснення: У даному фрагменті використовується статичний JSON-документ (`StaticJsonDocument`) розміром 256 байт, що є достатнім для зберігання поточних показників телеметрії. Після процесу серіалізації сформований рядок публікується у відповідний MQTT-топик. Важливо зауважити, що функція `publish()` у даному випадку не потребує ручного вказання довжини пакета, оскільки бібліотека `ArduinoJson` автоматично додає нуль-термінатор у кінці рядка, що гарантує коректну інтерпретацію та передачу даних до хмарного брокера `ThingsBoard`.

Лістинг 3.4 – Головний цикл `loop()` з обробкою станів

```
void loop() {
    if (!client.connected()) {
        reconnect();
    }
}
```

```

client.loop();

// Перевірка інтервалу часу без зупинки виконання коду
if (millis() - lastSend >= 3000) {
    lastSend = millis(); // Оновлення часової мітки

    // ... зчитування та відправка ...
}
}

```

Пояснення: У головному циклі програми `loop()` реалізовано алгоритм підтримки безперервного зв'язку з хмарною платформою. Замість стандартної функції затримки `delay()`, яка повністю зупиняє роботу процесора, у системі використано функцію `millis()`. Це дозволяє створити «неблокуючий» таймер: мікроконтролер постійно перевіряє стан мережевого з'єднання (`client.loop()`), а зчитування показників сенсорів та відправка JSON-пакета відбувається лише тоді, коли різниця між поточним часом та часом останньої відправки (`lastSend`) перевищує 3000 мілісекунд. Такий підхід забезпечує стабільність MQTT-сесії та дозволяє системі миттєво реагувати на вхідні команди від сервера без затримок.

3.3 Висновки до розділу 3

У третьому розділі успішно реалізовано та протестовано програмно-апаратний комплекс для дистанційного моніторингу життєвих показників пацієнта. Обчислювальним ядром системи виступив мікроконтролер ESP32, початкове налаштування та перевірка якого проводилися у безпечному середовищі апаратного моделювання Wokwi у форматі «цифрового двійника». Важливим архітектурним досягненням стала імплементація концепції граничних обчислень (Edge Computing). Завдяки цьому мікроконтролер не просто транслює сирі значення з датчиків, а самостійно аналізує їх на наявність критичних відхилень фізіологічної норми, що суттєво зменшує навантаження на сервер та підвищує загальну надійність системи.

Для забезпечення стабільного та швидкого обміну даними між кінцевим пристроєм і хмарною інфраструктурою було застосовано легковаговий протокол MQTT, який під час порівняльного аналізу довів свою абсолютну перевагу над HTTP для вбудованих медичних систем з обмеженими ресурсами. Використання оптимізованого стеку бібліотек (WiFi.h, PubSubClient та ArduinoJson) дозволило ефективно пакувати телеметричну інформацію у формат JSON та надійно передавати її до хмарної IoT-платформи ThingsBoard. На базі цієї платформи було розроблено зручний інтерактивний дашборд, який за допомогою спеціалізованих віджетів забезпечує медичному персоналу миттєву візуалізацію динаміки частоти серцевих скорочень та сатурації.

Особливу увагу в процесі розробки було приділено автоматизації реагування на позаштатні ситуації за допомогою механізму правил (Rule Engine). Налаштована логіка дозволяє хмарному серверу автоматично генерувати тривожні сповіщення при падінні рівня кисню або виникненні аритмії, а також самостійно знімати статус тривоги (Clear Condition) після стабілізації стану пацієнта. Це перетворює систему з простого засобу фіксації даних на повноцінний діагностичний інструмент, що мінімізує потребу в ручному контролі та знижує ризик людської помилки.

Експериментальна перевірка та вимірювання параметрів якості обслуговування (QoS) підтвердили високу продуктивність створеної інфраструктури. Застосування в програмному коді мікроконтролера неблокуючого таймера на базі функції millis() забезпечило безперервну підтримку мережевої сесії та відсутність пропусків під час зчитування даних. Результати навантажувального тестування показали, що середня затримка від моменту фіксації показника до його появи на дашборді лікаря становить лише 1,12 секунди і не перевищує максимуму в 1,3 секунди. Таким чином, розроблена архітектура повністю відповідає суворим вимогам до медичних IoT-систем реального часу, а всі поставлені завдання практичного розділу виконано в повному обсязі.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне науково-практичне завдання щодо проектування та розробки надійної, безпечної та масштабованої системи віддаленого моніторингу стану здоров'я пацієнтів на базі технологій Інтернету медичних речей (IoMT). Ґрунтовний аналіз сучасних тенденцій розвитку телемедицини засвідчив, що в умовах глобального зростання медичної інфляції, а також з огляду на специфічні екстремальні умови експлуатації в Україні (воєнний стан, нестабільність зв'язку, кіберзагрози), існує гостра необхідність у переході від централізованих архітектур до стійких до відмов розподілених систем. Розроблена в рамках дослідження багаторівнева функціональна схема, що базується на гібридній хмарно-периферійній парадигмі (Cloud-Edge Collaborative Model), успішно розв'язує ці проблеми, гарантуючи безперервність збору життєво важливих показників.

У процесі апаратного проектування було обґрунтовано використання двоядерного мікроконтролера ESP32-WROOM-32 у якості обчислювального ядра локального шлюзу, що дозволило збалансувати високу продуктивність та енергоефективність. Інтеграція прецизійних медичних сенсорів, зокрема аналогового інтерфейсу AD8232 для моніторингу ЕКГ із вбудованим пригніченням апаратних шумів, та цифрового оптичного модуля MAX30102 для пульсоксиметрії, забезпечила високоточну фіксацію первинних біопотенціалів. Перенесення частини аналітичних обчислень безпосередньо на мікроконтролер (Edge Computing) дозволило системі самостійно, без звернення до хмарного сервера, аналізувати показники на наявність критичних відхилень фізіологічної норми, таких як гіпоксія чи тахікардія.

Для надійного обміну даними між локальними вузлами та хмарною інфраструктурою було обрано легковаговий протокол MQTT. Завдяки моделі «публікація-підписка» та вбудованим механізмам контролю якості обслуговування (QoS), протокол довів свою беззаперечну перевагу над традиційним HTTP, гарантуючи доставку критичних повідомлень навіть в умовах

нестабільного стільникового зв'язку. Ключовим архітектурним досягненням стала реалізація наскрізної семантичної інтероперабельності: дані з сенсорів безпосередньо на мікроконтролері інкапсулюються у JSON-структури міжнародного медичного стандарту HL7 FHIR (ресурс Observation). Це архітектурне рішення усуває потребу у складних проміжних парсерах та закладає надійний фундамент для безшовної інтеграції отриманих телеметричних даних до централізованої електронної медичної картки пацієнта в рамках української системи eHealth.

Практична реалізація та експериментальне тестування програмно-апаратного комплексу, проведені у середовищі моделювання Wokwi та на базі хмарної IoT-платформи ThingsBoard, повністю підтвердили ефективність закладених проєктних рішень. Розроблений інтерактивний дашборд забезпечив медичному персоналу зручну візуалізацію динаміки показників у реальному часі, а налаштування механізму правил (Rule Engine) дозволило автоматизувати процес генерації та очищення тривожних сповіщень. Оцінка якості обслуговування продемонструвала високу стабільність системи: застосування неблокуючих таймерів дозволило досягти середньої затримки доставки даних від сенсора до екрана лікаря на рівні 1,12 секунди, що повністю відповідає жорстким стандартам систем реального часу. Таким чином, розроблена система є надійним, технологічно довершеним та готовим до впровадження інструментом дистанційної медицини, а мету і завдання дипломної роботи виконано в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yaacoub J. P. A., et al. Internet of Medical Things (IoMT): Taxonomy and Applications. 2024. URL: <https://www.scirp.org/journal/paperinformation?paperid=119451>
2. Remote Patient Monitoring Playbook. Internet of Medical Things (IoMT) devices. 2022. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10754247/>
3. IoT-based Remote Monitoring Systems (IoT-RMS) in Healthcare. 2024. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11665796/>
4. Roche Diagnostics. Top Telehealth Trends and Global Market Statistics. 2025. URL: <https://diagnostics.roche.com/global/en/healthcare-transformers/article/top-telehealth-trends.html>
5. National Health Interview Survey (NHIS). Changes in Telemedicine Use Among U.S. Adults. 2022. URL: <https://www.cdc.gov/nchs/data/nhsr/nhsr205.pdf>
6. American Medical Association (AMA). New data details how telehealth use varies by physician specialty. 2024. URL: <https://www.ama-assn.org/practice-management/digital-health/new-data-details-how-telehealth-use-varies-physician-specialty>
7. WTW. 2025 Global Medical Trends Survey: Rising Medical Costs. 2024. URL: <https://www.wtwco.com/en-ke/insights/2024/10/2025-global-medical-trends-survey>
8. World Health Organization (WHO). Global Strategy on Digital Health. URL: <https://www.who.int/health-topics/digital-health>
9. Ezurio. Top Applications of the Internet of Medical Things (IoMT) and Zero Trust Architecture. URL: <https://www.ezurio.com/resources/blog/top-applications-of-the-internet-of-medical-things-iomt-in-healthcare>
10. ДП «Електронне здоров'я». Цифрова трансформація охорони здоров'я: підсумки 2024-2025 років. URL: <https://ehealth.gov.ua/2025/12/30/tsyfrova-transformatsiya-ohorony-zdorov-ya-pidsumky-2025-roku/>

- 11.Міністерство охорони здоров'я України. Розбудова електронної системи охорони здоров'я (ЕКОЗ) в Україні. URL: <https://moz.gov.ua/uk/rozbudova-esoz-yiyi-metodologichna-ta-tehnichna-arhitektura>
- 12.Міністерство охорони здоров'я України. Базова інформація про дворівневу архітектуру ЕКОЗ. URL: <https://moz.gov.ua/uk/bazova-informaciya-pro-dvorivnevu-arhitekturu-esoz-v-ukrayini>
- 13.Наказ МОЗ України від 15.01.2024 № 77 "Про затвердження Порядку надання медичної та/або реабілітаційної допомоги із застосуванням телемедицини...". URL: <https://moz.gov.ua/uk/decrees/nakaz-moz-ukraini-vid-15012024--77>
- 14.Наказ МОЗ України від 22.01.2024 № 109 "Про проведення тестової експлуатації засобів телемедицини в умовах воєнного стану". URL: <https://moz.gov.ua/uk/decrees/nakaz-moz-ukraini-vid-22012024--109>
- 15.DLA Piper. Draft Law "On Personal Data Protection" No. 8153 in Ukraine. 2024. URL: <https://www.dlapiperdataprotection.com/index.html?t=law&c=UA>
- 16.Taylor Wessing. Impact of the Data Act on Medical and Health Devices. 2024. URL: <https://www.taylorwessing.com/en/insights-and-events/insights/2025/01/impact-of-the-data-act-on-medical-and-health-devices>
- 17.CMS Law. Ukraine approves new HTA guidelines for medical devices. URL: <https://cms-lawnow.com/en/ealerts/2025/12/ukraine-approves-new-hta-guidelines-for-medical-devices>
- 18.CMS Expert Guide to Digital Health Apps and Telemedicine: Ukraine. URL: <https://cms.law/en/int/expert-guides/cms-expert-guide-to-digital-health-apps-and-telemedicine/ukraine>
- 19.Classification of IoMT Devices: Wearables, Implantables, and Context Sensors. 2024. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC12526676/>
- 20.ScalableDigitalHealth (SDH): A Scalable IoT-Based Architecture for Remote Health Monitoring. MDPI Sensors. 2024. URL: <https://www.mdpi.com/1424-8220/24/4/1346>
- 21.Comparison of Communication Protocols: MQTT, CoAP, BLE, Zigbee for IoMT. MDPI Energies. URL: <https://www.mdpi.com/1996-1073/16/8/3465>

22. Network Performance in IoT: Latency, Jitter, and Packet Loss. URL: <https://kijoms.uokerbala.edu.iq/cgi/viewcontent.cgi?article=3165&context=home>
23. IoT Communication Protocols and QoS Metrics. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11054578/>
24. EMQX. MQTT Protocol Features for IoT Applications. URL: <https://www.emqx.com/en/blog/iot-protocols-mqtt-coap-lwm2m>
25. UUU Software. AWS vs Azure vs Google Cloud 2024 Market Share Comparison. URL: <https://uuusoftware.com/blog/what-is-better-aws-azure-or-google-cloud-2024-comparison>
26. Frontiers in Digital Health. Azure and AWS for HIPAA-Compliant IoT Architectures. 2024. URL: <https://www.frontiersin.org/journals/digital-health/articles/10.3389/fdgth.2024.1399461/full>
27. Telemedicine Monitoring System Based on Fog/Edge Computing. 2024. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC12488166/>
28. Lee S., Do H. Comparison and Analysis of ISO/IEEE 11073 and HL7 FHIR Messages for Personal Health Devices. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC5820086/>
29. ISO/IEEE 11073 X73-PHD Standards Compliant Systems. URL: <https://www.researchgate.net/publication/329662914>
30. Semantic Interoperability through HL7 FHIR for IoT Medical Signals. URL: <https://www.mdpi.com/2076-3417/15/23/12803>
31. HL7 FHIR Standards and Devices on FHIR (DoF). URL: https://build.fhir.org/ig/HL7/UTG/en/ECS_MR-v3-mdc.html
32. Kravchuk Y. et al. Analysis of Internet of Things and Cybersecurity Trends in Ukraine. 2024. URL: <http://perspectives.pp.ua/index.php/nts/article/download/15907/15979/15973>
33. ResearchGate. Internet of Medical Things (IoMT): Overview, Taxonomies and Classifications. [https://www.researchgate.net/publication/362912561 Internet of Medical Things IoMT Overview Taxonomies and Classifications](https://www.researchgate.net/publication/362912561_Internet_of_Medical_Things_IoMT_Overview_Taxonomies_and_Classifications)

34. Ghadi Y.Y., Shah S.F.A., Mazhar T., Shahzad T., Ouahada K., Hamam H. Enhancing patient healthcare with mobile edge computing and 5G: challenges and solutions for secure online health tools. *Journal of Cloud Computing*, 2024, 13, 93. DOI: 10.1186/s13677-024-00654-4. Доступно на: <https://link.springer.com/article/10.1186/s13677-024-00654-4>
35. HL7 FHIR Observation Resource [Електронний ресурс] // iNTERFACEWARE : [вебсайт]. — Режим доступу: <https://www.interfaceware.com/fhir/resources/observation>
36. Navakauskas D., Kazlauskas M. | *Informatica*, 2023, 34(3), 577–602 <https://www.informatica.vu.lt/journal/INFORMATICA/article/1306/info>
37. Muñoz Castañeda Á. L. Characterization of threats in IoT from an MQTT protocol-oriented dataset [Електронний ресурс] / Á. L. Muñoz Castañeda, J. A. A. Mata, H. Aláiz-Moretón // *Complex & Intelligent Systems*. — 2023. — Vol. 9. — P. 5281–5296. — Режим доступу: https://www.researchgate.net/publication/374143113_Characterization_of_threats_in_IoT_from_an_MQTT_protocol-oriented_dataset
38. НСЗУ. Електронна система охорони здоров'я (ЕСОЗ). <https://nszu.gov.ua/e-dani/esoz>
39. The Pharma Media. МОЗ оновило порядок надання реабілітаційної допомоги за допомогою телемедицини. <https://thepharma.media/uk/news/34684-moz-onovilo-poryadok-nadannya-reabilitacii-noyi-dopomogi-za-dopomogoyu-instrumentiv-telemedicini-19042024>
40. eHealth Ukraine. ЕСОЗ 2024: акценти оновлення системи. <https://ehealth.gov.ua/2025/02/05/esoz-2024-aktsenty-onovlen-systemy-dlya-korystuvachiv-i-dynamika-zrostannya-elektronnyh-medychnyh-zapysiv/>
41. eHealth Ukraine. Як буде покращуватися продуктивність ЦБД ЕСОЗ у 2025 році. <https://ehealth.gov.ua/2025/02/28/yak-bude-pokrashhuvatysya-produktyvnist-tsbd-esoz-u-2025-rotsi/>

- 42.AllPCB. Microcontroller Selection for IoT: A Guide for PCB Design.
<https://www.allpcb.com/allelectrohub/microcontroller-selection-for-iot-a-guide-for-pcb-design>
- 43.ESPBoards. ESP32 SoC Options. <https://www.espboards.dev/blog/esp32-soc-options/>
- 44.STMicroelectronics Wiki. STM32WBA Health Thermometer.
https://wiki.st.com/stm32mcu/wiki/Connectivity:STM32WBA_Health_Thermometer
- 45.STMicroelectronics. STM32WBA Series. <https://www.st.com/en/microcontrollers-microprocessors/stm32wba-series.html>
- 46.Future Electronics. STM32WBA Series Microcontrollers.
<https://www.futureelectronics.com/npi/stmicroelectronics-stm32wba-series-microcontrollers>
- 47.DFRobot. Product 1697 (AD8232/MAX30102). <https://www.dfrobot.com/product-1697.html>
- 48.ResearchGate. Affordable Heart Health Monitoring System Using ESP32, AD8232 and MAX30102.
https://www.researchgate.net/publication/398266791_Affordable_Heart_Health_Monitoring_System_Using_ESP32_AD8232_and_MAX30102
- 49.AD8232: Single-Lead, Heart Rate Monitor Front End. Data Sheet [Электронный ресурс] / Analog Devices. — Режим доступа:
<https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf>
- 50.Analog Devices. MAX30102 Datasheet. <https://www.analog.com/media/en/technical-documentation/data-sheets/max30102.pdf>
- 51.BME680 Digital Environmental Sensor. Application Note BST-BME680-AN014 [Электронный ресурс] / Bosch Sensortec. — Режим доступа: https://www.bosch-sensortec.com/media/boschsensortec/downloads/application_notes_1/bst-bme680-an014.pdf
- 52.ESP32-based ECG and SpO2 monitoring system [Электронный ресурс] // Circuit Designer : [вебсайт]. — Режим доступа:
<https://docs.circuitdesigner.com/project/published/272959f7-29ec-4752-bda4-8bc87b9bfa60/esp32-based-ecg-and-spo2-monitoring-system>

- 53.Semantic Scholar. Power consumption analysis of Bluetooth Low Energy.
<https://www.semanticscholar.org/paper/Power-consumption-analysis-of-Bluetooth-Low-Energy%2C-Dementyev-Hodges/bd1a81b7571512f40feddef3114287513d31d782>
- 54.MECS Press. IJCNIS Vol.9 No.7. <https://www.mecs-press.org/ijcnis/ijcnis-v9-n7/IJCNIS-V9-N7-5.pdf>
- 55.JMIR Publications. Medinform 2024;1:e58445.
<https://medinform.jmir.org/2024/1/e58445>
- 56.European Commission. The Data Act explained. <https://digital-strategy.ec.europa.eu/en/factpages/data-act-explained>
- 57.CMS Law. Adapting to the new EU Data Act: implications for medical devices.
<https://cms.law/en/lux/legal-updates/adapting-to-the-new-eu-data-act-implications-for-medical-devices-and-other-health-devices>
- 58.MHC.ie. The EU Data Act: spotlight on digital health.
<https://www.mhc.ie/latest/insights/the-eu-data-act-spotlight-on-digital-health>
- 59.Wokwi — Режим доступа: <https://wokwi.com/>
- 60.ThingsBoard — Режим доступа: <https://thingsboard.io/>