

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)  
(рівень вищої освіти)

Моделі ефективності бездротових протоколів передачі даних у  
ІоТ-системах моніторингу показників здоров'я людини  
(тема)

Виконав: здобувач II курсу, групи СКСм-24-1

Склярова В.А.

(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані

комп'ютерні системи

(повна назва освітньої програми)

Керівник роботи Хаханова Г.В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

\_\_\_\_\_

(підпис)

Чумаченко С.В.


(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
Кафедра Автоматизації проєктування обчислювальної техніки  
Рівень вищої освіти другий (магістерський)  
Спеціальність 123 Комп'ютерна інженерія  
Тип програми Освітньо-професійна  
Освітня програма Спеціалізовані комп'ютерні системи

ЗАТВЕРДЖУЮ:

Зав. кафедри   
(підпис)  
«  »    2025 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Скляровій Валерії Андріївні  
(прізвище, ім'я, по батькові)

- Тема роботи Моделі ефективності бездротових протоколів передачі даних у IoT-системах моніторингу показників здоров'я людини  
затверджена наказом університету від 07 11 2025 р. № 1012 Ст
- Термін подання студентом роботи до екзаменаційної комісії 16 12 2025 р.
- Вихідні дані до роботи Arduino Uno,  
Датчик пульсоксиметра MAX30100,  
Датчик DS18B20,  
модуль ESP8266,  
середовище розробки Arduino IDE
- Перелік питань, що потрібно опрацювати в роботі Архітектура та принципи функціонування IoT-систем медичного призначення.  
Апаратна платформа мікроконтролерної.  
Дослідження ефективності протоколів передачі даних IoT.  
Розробка програмної частини проєкту.  
Тестування мікроконтролерної системи.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) 18 слайдів

---

---

---


6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1 )


Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	01.09.2025 – 02.09.2025	
2	Аналіз проблемної галузі, постановка завдання	03.09.2025 – 10.09.2025	
3	Вибір інструментальних засобів та розробка	11.09.2025 – 18.09.2025	
4	Розробка програми для мікроконтролера	19.09.2025 – 10.10.2025	
5	Програмна реалізація веб-сторінки	11.11.2025 – 25.11.2025	
6	Тестування розробленої системи	26.11.2025 – 03.12.2025	
7	Оформлення пояснювальної записки	04.12.2025 – 18.12.2025	
8	Оформлення графічного матеріалу	19.12.2025 – 25.12.2025	
9	Перевірка виконаного проекту керівником	26.12.2025 – 27.12.2025	
10	Захист роботи	22.12.2025	

Дата видачі завдання 01 вересня 2025 р.

Студент   
(підпис)

Керівник роботи  доц. каф. АПОТ Хаханова Г.В.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи магістра містить 69 сторінок, 20 рисунків, 4 таблиці, 15 джерел за переліком посилань.

ARDUINO UNO, BLUETOOTH, WI-FI, C++, ARDUINO IDE, МОБІЛЬНИЙ ПРИСТРІЙ, МЕДИЧНИЙ МОНІТОРИНГ, ESP8266, МІКРОКОНТРОЛЕРНА СИСТЕМА, БЕЗДРОТОВІ ПРОТОКОЛИ ПЕРЕДАЧІ ДАНИХ

Метою роботи є дослідження та побудова моделей ефективності бездротових протоколів передачі даних у IoT-системах дистанційного моніторингу здоров'я. Для цього проаналізовано архітектуру медичних IoT-систем та особливості передачі біометричних даних. Розроблено апаратну частину на базі платформи Arduino Uno, яка об'єднує сенсори серцевого ритму та сатурації, температури тіла і мікроклімату у приміщенні. Організовано канал зв'язку за допомогою Wi-Fi модуля ESP8266. У середовищі Arduino IDE створено програмне забезпечення, що реалізує цифрову обробку сигналів та передачу телеметрії на хмарний сервіс ThingSpeak. Проведено експериментальне порівняння бездротових технологій Wi-Fi та Bluetooth за показниками стабільності та швидкодії. У результаті роботи побудовано моделі ефективності протоколів та створено діючий прототип системи для контролю стану пацієнта в реальному часі.

## ABSTRACT

The explanatory note of the master's qualification work contains 69 pages, 20 figures, 4 tables, 15 sources according to the list of links.

ARDUINO UNO, BLUETOOTH, WI-FI, C++, ARDUINO IDE, MOBILE DEVICE, MEDICAL MONITORING, ESP8266, MICROCONTROLLER SYSTEM, WIRELESS DATA TRANSMISSION PROTOCOLS

The purpose of the work is to study and build models of the efficiency of wireless data transmission protocols in IoT systems for remote health monitoring. For this purpose, the architecture of medical IoT systems and the features of biometric data transmission were analyzed. The hardware part was developed based on the Arduino Uno platform, which combines heart rate and saturation sensors, body temperature and indoor microclimate. A communication channel was organized using the ESP8266 Wi-Fi module. Software was created in the Arduino IDE environment that implements digital signal processing and telemetry transmission to the ThingSpeak cloud service. An experimental comparison of Wi-Fi and Bluetooth wireless technologies in terms of stability and speed was conducted. As a result of the work, models of protocol effectiveness were built and a working prototype of a system for monitoring the patient's condition in real time was created.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП .....	9
1 АРХІТЕКТУРА ТА ПРИНЦИПИ ФУНКЦІОНУВАННЯ ІОТ-СИСТЕМ МЕДИЧНОГО ПРИЗНАЧЕННЯ .....	11
1.1 Поняття та принципи функціонування систем Інтернету речей.....	11
1.2 Застосування ІоТ у медицині.....	12
1.3 Принципи організації передачі даних у медичних ІоТ-системах .....	15
1.4 Перспективи розвитку ІоТ-систем у медицині .....	19
1.5 Порівняльна характеристика технологій бездротової передачі даних у медичних системах .....	19
1.6 Постановка задачі .....	21
2 АПАРАТНА ПЛАТФОРМА МІКРОКОНТРОЛЕРНОЇ СИСТЕМИ.....	23
2.1 Arduino .....	23
2.2 Плата Arduino Uno R3.....	25
2.3 Wi-Fi модуль ESP8266.....	28
2.4 Bluetooth-модуль HC-06 .....	30
2.5 Біометричний модуль MAX30102.....	32
2.6 Датчик температури DS18B20.....	34
2.7 Датчик температури та вологості DHT11.....	36
3 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ПРОТОКОЛІВ ПЕРЕДАЧІ ДАНИХ ІОТ .....	39
3.1 Експериментальне дослідження показників ефективності передачі даних через Bluetooth.....	39
3.2 Експериментальне дослідження показників ефективності передачі даних через Wi-Fi.....	46

3.2.1 Дослідження протоколу HTTP .....	47
3.2.2. Дослідження протоколу MQTT .....	48
3.2.3. Порівняльний аналіз результатів .....	49
4.1 Структурна та функціональні схеми системи.....	51
4.2 Програмна реалізація мікроконтролерної системи .....	55
4.3 Організація взаємодії системи з хмарною IoT-платформою ThingSpeak .....	59
5 ТЕСТУВАННЯ МІКРОКОНТРОЛЕРНОЇ СИСТЕМИ.....	62
ВИСНОВКИ.....	66
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	68
ДОДАТОК А.....	70
ДОДАТОК Б .....	79

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АЦП – аналого-цифровий перетворювач.

API (Application Programming Interface) – інтерфейс прикладного програмування (ключі доступу).

BPM (Beats Per Minute) – частота серцевих скорочень.

HTTP (HyperText Transfer Protocol) – протокол передачі даних.

I2C (Inter-Integrated Circuit) – шина даних для підключення.

IoMT (Internet of Medical Things) – Інтернет медичних речей.

IoT (Internet of Things) – Інтернет речей.

SpO<sub>2</sub> – рівень насичення крові киснем.

UART (Universal Asynchronous Receiver-Transmitter) – інтерфейс для зв'язку з Wi-Fi та Bluetooth модулями.

Wi-Fi – технологія бездротової локальної мережі.

## ВСТУП

Актуальність теми кваліфікаційної роботи, пов'язаної з дослідженням моделей ефективності бездротових протоколів у системах моніторингу здоров'я, полягає в тому, що зі стрімким розвитком технологій Інтернету медичних речей (ІоМТ) зростає потреба в автоматизації процесів діагностики, які підвищують якість життя та безпеку пацієнтів. Одним із таких критично важливих процесів є організація надійної передачі біометричних даних, адже сучасні системи охорони здоров'я стикаються з проблемою своєчасності отримання інформації та необхідністю оптимізації каналів зв'язку. Створення моделей ефективності бездротових протоколів дозволяє дослідити можливості застосування інтернету речей, сенсорних технологій і хмарних сервісів для підвищення стабільності функціонування систем дистанційного нагляду за станом людини.

У сучасному світі системи дистанційного моніторингу, які набули значного поширення та продовжують активно розвиватися, охоплюючи дедалі ширше коло медичних та профілактичних задач, являють собою інтегрований комплекс інтелектуальних технологій, спрямованих на автоматизацію збору вітальних показників. Завдяки поєднанню біометричних сенсорів, мікроконтролерів і засобів бездротового зв'язку вони дозволяють користувачеві або лікарю дистанційно контролювати серцевий ритм, температуру тіла та інші аспекти фізичного стану в реальному часі.

Одним із ключових і водночас складних технічних рішень, що реалізуються у рамках концепції медичного ІоТ, є вибір та оптимізація протоколів бездротової передачі даних, які забезпечують зв'язок між пацієнтом та хмарним сервісом. Інтеграція правильно підібраних протоколів у систему моніторингу дозволяє створити ефективний механізм контролю здоров'я без втрати критично важливих пакетів даних, що особливо актуально для ситуацій, які вимагають миттєвого реагування на зміни в організмі.

Метою кваліфікаційної роботи є розроблення та дослідження моделей ефективності бездротових протоколів передачі даних у IoT-системах моніторингу показників здоров'я людини, призначених для збору даних із датчиків та їх трансляції на віддалені термінали. Основою розробленої системи є апаратна платформа на базі Arduino Uno та модуля ESP8266, яка забезпечує обробку даних, отриманих від сенсорів пульсоксиметру, температури тіла, а також температури та вологості приміщення, у якому перебуває пацієнт, і передає інформацію на хмарний сервіс ThingSpeak. Це дозволяє здійснювати моніторинг у режимі реального часу, аналізувати стабільність з'єднання та порівнювати ефективність різних методів передачі даних. Технології Інтернету речей забезпечують інтеграцію всіх компонентів системи у єдину взаємодіючу мережу, що гарантує здатність системи не лише фіксувати поточні показники, а й забезпечувати надійний канал зв'язку, підвищуючи загальний рівень інформативності та енергоефективності комплексу.

# 1 АРХІТЕКТУРА ТА ПРИНЦИПИ ФУНКЦІОНУВАННЯ ІОТ-СИСТЕМ МЕДИЧНОГО ПРИЗНАЧЕННЯ

## 1.1 Поняття та принципи функціонування систем Інтернету речей

Інтернет речей (Internet of Things, IoT) – це концепція, що передбачає об'єднання фізичних пристроїв у мережу для автоматизованого збору, обміну та аналізу даних без безпосередньої участі людини. У таких системах різні об'єкти – сенсори, контролери, мобільні пристрої, сервери, побутова техніка, транспортні засоби тощо – взаємодіють між собою за допомогою бездротових або дротових каналів зв'язку.

Основна ідея IoT полягає у створенні єдиного інформаційного середовища, у якому пристрої можуть не лише передавати дані, але й приймати рішення, виконувати команди, реагувати на зміни навколишнього середовища. Це забезпечує новий рівень автоматизації та комфорту, а також відкриває можливості для ефективного управління технічними, побутовими й медичними процесами.

Типова архітектура IoT-системи включає кілька рівнів (рисунок 1.1):

- рівень пристроїв (сенсорів) – фізичні елементи, які вимірюють параметри середовища або об'єкта (температуру, вологість, тиск, пульс, рух тощо);
- рівень збору та попередньої обробки даних – мікроконтролери або шлюзи, що агрегують сигнали від сенсорів, виконують базову фільтрацію та передають інформацію далі;
- комунікаційний рівень – бездротові або дротові протоколи, які забезпечують передачу даних до серверів або хмарних платформ;
- рівень зберігання та обробки даних – бази даних або хмарні сервіси, де інформація накопичується, аналізується й використовується для прийняття рішень.

– рівень користувача – інтерфейс доступу до даних (мобільний додаток, веб-панель, монітор), який дозволяє переглядати показники, статистику та повідомлення про критичні зміни параметрів.

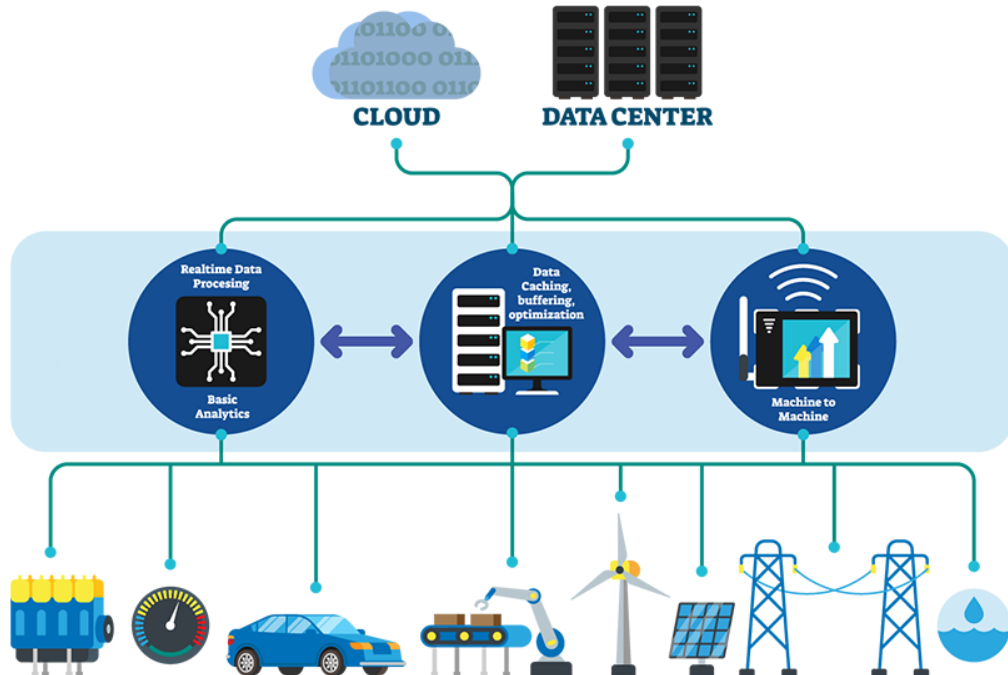


Рисунок 1.1 – Складові Інтернету речей

## 1.2 Застосування IoT у медицині

Інтернет речей сьогодні є одним із найдинамічніших напрямів розвитку сучасних технологій. Його впровадження у медичну сферу, відоме як Internet of Medical Things (IoMT), радикально змінює підходи до діагностики, лікування та профілактики захворювань. Суть цих технологій полягає у створенні мережі взаємопов'язаних пристроїв – сенсорів, мікроконтролерів, мобільних додатків і серверів, – які спільно забезпечують безперервний збір, обробку та аналіз даних про стан здоров'я людини.

IoT у медицині передбачає, що всі пристрої, які вимірюють фізіологічні параметри пацієнта, обмінюються інформацією між собою та з хмарними сервісами через бездротові протоколи. Це дозволяє лікарям і пацієнтам у режимі реального часу отримувати актуальні відомості про основні життєві показники – температуру тіла, пульс, рівень кисню в крові, артеріальний тиск чи частоту дихання. Важливо, що такі дані можуть зберігатися та аналізуватися тривалий час, формуючи базу для оцінки динаміки стану здоров'я.

Найбільше поширення технології IoT отримали у сфері дистанційного моніторингу пацієнтів. Це особливо актуально для людей із хронічними захворюваннями, які потребують постійного контролю показників, але не завжди мають можливість часто відвідувати лікаря. Переносні пристрої, такі як розумні браслети чи спеціальні медичні сенсори, передають дані до мобільного додатка або на сервер лікаря. У разі, якщо система фіксує відхилення від норми, може автоматично надходити сповіщення лікарю чи родичам пацієнта. Такий підхід дозволяє виявляти критичні зміни на ранніх етапах і запобігати ускладненням.

Велике значення IoT має і для розвитку телемедицини. Під час онлайн-консультацій лікар може мати доступ до поточних даних про стан пацієнта, що підвищує точність діагностики та ефективність лікування. У сучасних клініках дедалі частіше впроваджуються «розумні» системи, які автоматизують роботу медичних закладів: відстежують роботу обладнання, контролюють параметри середовища у лабораторіях, оптимізують споживання ресурсів.

Значного поширення набувають також системи, що допомагають пацієнтам дотримуватися призначеного лікування. Наприклад, спеціальні пристрої або мобільні додатки можуть нагадувати про час прийому ліків, а інтелектуальні упаковки для медикаментів здатні фіксувати факт відкриття й передавати відповідні дані. У разі відхилення від графіка система може надсилати сповіщення пацієнту чи лікарю.

Однією з ключових переваг використання IoT у медицині є можливість забезпечення безперервного моніторингу пацієнта без потреби його постійної присутності в лікарні. Такий підхід не лише підвищує комфорт пацієнтів, а й зменшує навантаження на медичні установи. Завдяки автоматичним повідомленням лікар може оперативно реагувати на зміни стану здоров'я, що особливо важливо у випадках, коли від швидкості реакції залежить життя людини. Крім того, аналіз великого обсягу зібраних даних дозволяє персоналізувати лікування, підбираючи терапію з урахуванням індивідуальних особливостей пацієнта.

Прикладами сучасних рішень на основі IoT є системи Philips HealthSuite, що забезпечують збір та аналітику медичних даних на хмарній платформі, а також популярні споживчі пристрої Fitbit чи Xiaomi Mi Band, які відстежують фізичну активність, сон та серцевий ритм (рисунок 2.2). До спеціалізованих медичних пристроїв належать, наприклад, портативний ЕКГ-монітор AliveCor KardiaMobile чи пульсоксиметр iHealth Wireless Pulse Oximeter, який синхронізується зі смартфоном і дозволяє лікарю контролювати рівень кисню в крові пацієнта на відстані.



Рисунок 1.2 – Фітнес-трекери Xiaomi Mi Band

Разом із тим впровадження IoT у медицину супроводжується низкою викликів. Найважливішими з них є питання захисту персональних даних і безпеки передачі інформації, адже медичні відомості належать до конфіденційних. Не менш актуальною є проблема сумісності різних пристроїв і платформ, оскільки виробники часто використовують власні стандарти. Додатковою складністю залишається обмежена енергоефективність переносних сенсорів, які повинні працювати тривалий час без підзарядки.

Попри ці труднощі, потенціал IoT у медицині залишається надзвичайно високим. Технології інтелектуального моніторингу сприяють переходу від реактивної моделі лікування, коли втручання відбувається вже після появи симптомів, до проактивної, що базується на постійному спостереженні й попередженні ускладнень. Такий підхід дозволяє не лише підвищити якість життя пацієнтів, а й створює передумови для розвитку профілактичної медицини, де головним завданням є не лікування, а збереження здоров'я.

Таким чином, застосування IoT у медицині відкриває новий етап у розвитку систем охорони здоров'я, об'єднуючи інженерні, інформаційні та біомедичні технології в єдиний інтелектуальний комплекс. Його подальший розвиток сприятиме створенню більш ефективних, доступних і безпечних засобів контролю за станом людини, що в перспективі може стати основою для нової моделі взаємодії між лікарем, пацієнтом і технологіями.

### 1.3 Принципи організації передачі даних у медичних IoT-системах

Одним із ключових елементів будь-якої IoT-системи, зокрема медичної, є механізм передачі даних між сенсорними пристроями, мікроконтролерами, шлюзами та хмарними сервісами (рисунок 2.3). Надійність, швидкість і безпека цього процесу визначають ефективність функціонування всієї системи моніторингу стану здоров'я людини.



Рисунок 1.3 – Механізм передачі даних

Передача даних у медичних IoT-системах має свої особливості, зумовлені специфікою зібраної інформації. Йдеться про біомедичні сигнали – температуру тіла, пульс, рівень кисню в крові, артеріальний тиск, електрокардіограму тощо. Ці дані є конфіденційними, а тому потребують захищеної передачі, а також мають бути доставлені вчасно, без затримок і втрат.

Організація обміну даними у таких системах зазвичай побудована за багаторівневою структурою. На нижньому рівні розміщуються сенсорні модулі, які безпосередньо вимірюють параметри організму. Вони підключаються до мікроконтролера або спеціального вузла збору даних, що виконує попередню обробку сигналів, перетворення аналогових величин у цифрові та фільтрацію шумів. На наступному етапі відбувається передача цих даних через бездротовий канал зв'язку до проміжного пристрою – шлюзу або смартфона користувача.

Шлюз, як правило, виступає посередником між локальною мережею сенсорів і глобальною мережею Інтернет. Його завдання полягає в тому, щоб забезпечити з'єднання між менш потужними пристроями, які використовують

енергоєфективні протоколи, та хмарними сервісами, де дані зберігаються, аналізуються та візуалізуються. Для цього шлюзи часто підтримують декілька типів інтерфейсів – наприклад, Bluetooth для локального з'єднання з датчиками та Wi-Fi або стільниковий зв'язок 4G/5G для доступу до Інтернету.

У медичних IoT-системах особлива увага приділяється вибору бездротового протоколу передачі даних. Він визначає не лише дальність і швидкість з'єднання, а й енергоспоживання пристроїв, що має велике значення для переносних сенсорів. Для коротких відстаней, наприклад, під час зв'язку сенсорів із мобільним телефоном пацієнта, найчастіше використовуються технології Bluetooth Low Energy (BLE) або ZigBee. Вони забезпечують низьке енергоспоживання при достатній пропускній здатності для передавання медичних показників.

Для більш віддалених підключень застосовуються Wi-Fi, LoRaWAN або стільникові мережі, які дозволяють передавати дані безпосередньо до хмарних серверів. У системах стаціонарного моніторингу, наприклад у лікарнях, Wi-Fi залишається одним із найпоширеніших рішень, оскільки він забезпечує високу швидкість та стабільність передачі. Натомість для віддалених або автономних систем, що функціонують у режимі мінімального енергоспоживання, перевагу надають LoRa або NB-IoT.

Передача медичних даних організовується за допомогою спеціалізованих протоколів прикладного рівня, таких як MQTT або HTTP/HTTPS. Протокол MQTT, який часто використовується в IoT-системах, побудований за моделлю «видавець-підписник» і оптимізований для передавання невеликих обсягів даних через нестабільні або обмежені канали зв'язку. Його головна перевага – мінімальне навантаження на мережу та можливість швидкої обробки повідомлень у реальному часі.

У випадку, коли потрібно забезпечити сумісність із веб-сервісами або хмарними платформами, може застосовуватися протокол HTTP або його захищена версія HTTPS. Він дозволяє надсилати дані безпосередньо до серверів, наприклад, до платформи ThingSpeak, AWS IoT Core, Google Cloud

IoT чи аналогічних. Залежно від конфігурації системи, дані можуть передаватися з різною періодичністю – у режимі реального часу, з фіксованими інтервалами або за подіями (наприклад, при виході показників за межі норми).

Одним із принципів організації передачі даних у медичних IoT-системах є надійність зв'язку. Система має бути стійкою до втрати пакетів, перебоїв живлення або тимчасової відсутності Інтернету. Для цього в протоколах реалізуються механізми підтвердження доставки, повторної передачі даних або буферизації повідомлень. Важливим також є захист інформації, що передається. Усі передані дані шифруються з використанням алгоритмів SSL/TLS, AES або інших криптографічних методів, що забезпечує конфіденційність і цілісність біомедичних показників.

Ще одним важливим принципом є синхронізація часу між пристроями. Для правильного відображення динаміки показників необхідно, щоб кожне вимірювання мало точну часову позначку. Це дозволяє будувати графіки змін стану пацієнта, виявляти тенденції та аномалії. Синхронізація може здійснюватися через мережеві протоколи NTP або локальні часові сигнали.

Важливу роль відіграє також енергоефективність передачі даних. Переносні сенсори, які працюють від акумуляторів, повинні підтримувати баланс між частотою передачі та споживанням енергії. Тому часто застосовуються стратегії, за яких дані надсилаються не постійно, а лише при зміні показників або за визначеним інтервалом часу. Це дозволяє значно продовжити автономну роботу пристроїв.

У комплексі всі ці принципи забезпечують стабільне функціонування медичних IoT-систем. Вони дозволяють реалізувати безперервний, надійний і безпечний обмін даними між сенсорами, контролерами, мобільними пристроями та хмарними сервісами. Завдяки цьому створюється основа для ефективного моніторингу фізіологічних параметрів людини в реальному часі, що підвищує якість медичних послуг, своєчасність діагностики та персоналізацію лікування.

## 1.4 Перспективи розвитку IoT-систем у медицині

Інтернет речей у медицині продовжує активно розвиватися, і цей напрям є одним із найперспективніших у сфері цифрових технологій. Поєднання сенсорних пристроїв, хмарних сервісів і аналітики даних відкриває нові можливості для моніторингу здоров'я, профілактики захворювань і покращення якості медичних послуг.

Сучасні тенденції розвитку IoT у медицині спрямовані на:

- розширення функціональності сенсорів – підвищення точності, зменшення енергоспоживання та інтеграція кількох параметрів у одному пристрої;
- використання штучного інтелекту – автоматичний аналіз великих масивів даних для прогнозування стану пацієнта або виявлення аномалій;
- хмарні технології та Edge-комп'ютинг – забезпечення швидкої обробки та передачі даних у режимі реального часу;
- покращення безпеки даних – розробка нових методів шифрування та автентифікації для захисту медичної інформації;
- інтеграція з телемедициною – віддалений моніторинг і консультації лікарів без необхідності особистого відвідування медзакладів.

У майбутньому очікується широке впровадження «розумних» медичних платформ, здатних працювати автономно, аналізувати дані про здоров'я та формувати індивідуальні рекомендації. Це сприятиме переходу від лікування до превентивної медицини, де основний акцент робиться на попередженні захворювань.

## 1.5 Порівняльна характеристика технологій бездротової передачі даних у медичних системах

Вибір стандарту бездротового зв'язку є визначальним етапом проєктування систем ІоМТ, оскільки від цього залежать ключові параметри

пристрою: автономність роботи, радіус дії, швидкість оновлення даних та надійність доставки критичних повідомлень.

Wi-Fi є найбільш поширеним стандартом для організації локальних мереж з виходом в Інтернет:

- переваги: висока пропускна здатність (до сотень Мбіт/с), що дозволяє передавати великі масиви даних без затримок. Легкість інтеграції з хмарними платформами (ThingSpeak, AWS) без додаткових шлюзів, оскільки Wi-Fi роутери є майже в кожному приміщенні;

- недоліки: високе енергоспоживання (активна передача може споживати 200–300 мА), що ускладнює використання в мініатюрних пристроях із батарейним живленням, а також чутливість до перешкод від стін та інших пристроїв;

Bluetooth Low Energy – спеціалізована версія Bluetooth, оптимізована для IoT:

- переваги: низьке енергоспоживання, підтримка прямого з'єднання зі смартфонами та планшетами, які виступають шлюзами для передачі даних у хмару;

- недоліки: обмежений радіус дії (зазвичай до 10–30 м у приміщенні), низька швидкість передачі даних, якої достатньо лише для періодичних показників, але не для потокового відео чи аудіо.

ZigBee – протокол для створення розподілених сенсорних мереж:

- переваги: використання комірчастої топології, де кожен пристрій є ретранслятором, що дозволяє покривати великі площі та забезпечує високу надійність: якщо один вузол вийде з ладу, сигнал піде іншим шляхом, і низьке енергоспоживання.

- недоліки: необхідність використання спеціального шлюзу, оскільки смартфони та ПК не підтримують цей стандарт напряму, низька швидкість передачі (до 250 кбіт/с).

LoRaWAN – технологія низькошвидкісної передачі даних на великі відстані:

– переваги: великий радіус дії, висока проникаюча здатність сигналу, дуже низьке енергоспоживання;

– недоліки: низька пропускна здатність (від 0.3 до 50 кбіт/с), велика затримка передачі, не підходить для моніторингу в реальному часі, лише для періодичних звітів.

Для наочності основні параметри розглянутих стандартів було зведено у таблицю 1.1.

Таблиця 1.1 – Порівняльний аналіз бездротових технологій передачі даних

Характеристика	Wi-Fi	Bluetooth LE	ZigBee	LoRaWAN
Радіус дії	30–50 м	10–30 м	10–100 м (Mesh)	> 1 км
Пропускна здатність	до 600 Мбіт/с	1–2 Мбіт/с	250 кбіт/с	0.3–50 кбіт/с
Енергоспоживання	Високе	Дуже низьке	Низьке	Дуже низьке
Топологія мережі	Зірка	Точка-точка, Зірка, Комірка	Комірка	Зірка із зірок
Складність підключення	Низька (прямий IP)	Середня (потрібен шлюз/телефон)	Висока (потрібен координатор)	Висока (потрібен шлюз LoRa)
Вартість модуля	Низька	Низька	Середня	Середня/Висока

## 1.6 Постановка задачі

Для реалізації мети дослідження необхідно розробити апаратно-програмний комплекс дистанційного моніторингу стану пацієнта, який забезпечує автоматизований збір фізіологічних показників (серцевий ритм, сатурація кисню, температура тіла) та параметрів мікроклімату з їх подальшою бездротовою передачею на хмарну платформу ThingSpeak. Спроектувати структурну та функціональну схеми пристрою, створити програмне забезпечення для платформи Arduino, що реалізує цифрову обробку сигналів, та провести експериментальне порівняння ефективності протоколів передачі

даних Wi-Fi та Bluetooth.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз архітектури сучасних ІоМТ-систем та порівняльну характеристику бездротових технологій (Wi-Fi, Bluetooth) у контексті медичного застосування;
- обґрунтувати вибір елементної бази (сенсори MAX30102, DS18B20, DHT11) та комунікаційних інтерфейсів для побудови енергоефективного вузла моніторингу;
- розробити апаратну частину прототипу на базі мікроконтролера Arduino Uno з інтеграцією модулів бездротового зв'язку ESP8266 та HC-06;
- реалізувати програмні алгоритми первинної обробки біосигналів, детекції серцевого ритму та формування пакетів даних для передачі на сервер;
- провести експериментальне дослідження та побудувати моделі ефективності використання каналів Wi-Fi та Bluetooth за критеріями стабільності, затримки та енергоспоживання.

## 2 АПАРАТНА ПЛАТФОРМА МІКРОКОНТРОЛЕРНОЇ СИСТЕМИ

### 2.1 Arduino

Arduino – це універсальна апаратно-програмна платформа з відкритим вихідним кодом, призначена для розробки та прототипування електронних пристроїв, систем автоматизації та робототехніки. Платформа об'єднує в собі стандартизовані апаратні модулі та спеціалізоване програмне середовище, що дозволяє суттєво спростити процес взаємодії з мікроконтролерами різної архітектури.

Особливістю платформи є її відкритість. Принципові електричні схеми та топологія друкованих плат публікуються розробниками у вільному доступі. Це дозволяє інженерам не лише вільно використовувати та модифікувати існуючі рішення, але й створювати власні плати або інтегрувати схемотехнічні рішення Arduino у промислові вироби без порушення авторських прав.

Апаратна складова платформи базується на друкованих платах із встановленим мікроконтролером (найчастіше сімейства AVR від Microchip, а також ARM Cortex). Ключовою інженерною перевагою є стандартизація розташування виводів. Це забезпечує сумісність базового модуля з широким спектром плат розширення, так званих «шилдів» (рисунок 2.1). Шилди – це додаткові модулі, які встановлюються безпосередньо на плату мікроконтролера, формуючи модульну структуру. Такий підхід дозволяє масштабувати функціонал системи без необхідності пайки та розробки складних друкованих плат на етапі макетування.

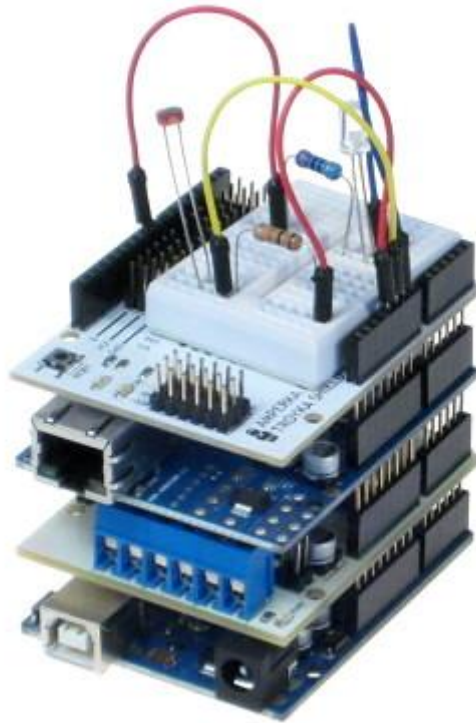


Рисунок 2.1 – Плата розширення Arduino

Завдяки поєднанню доступної апаратної бази та розвиненої спільноти, платформа Arduino стала стандартом для створення IoT-рішень (Internet of Things). Вона дозволяє реалізовувати складні алгоритми збору даних із сенсорів, їх первинної обробки та передачі через мережеві інтерфейси. Проєкти на базі Arduino здатні працювати як автономно, керуючи виконавчими механізмами, так і у складі розподілених систем, взаємодіючи з програмним забезпеченням на серверах або мобільних пристроях.

Для даної магістерської роботи вибір платформи Arduino обумовлений необхідністю швидкої інтеграції різноманітних компонентів (біометричних датчиків та комунікаційних модулів) та наявністю широкої підтримки бібліотек.

## 2.2 Плата Arduino Uno R3

Як базовий елемент вузла моніторингу в даній роботі обрано плату Arduino Uno R3. Це найбільш розповсюджена версія платформи, побудована на базі 8-бітного мікроконтролера ATmega328P виробництва Microchip. Вибір саме цієї модифікації зумовлений оптимальним співвідношенням обчислювальної потужності, кількості портів введення-виведення та енергоспоживання для задач збору біометричних даних (рисунок 2.2).

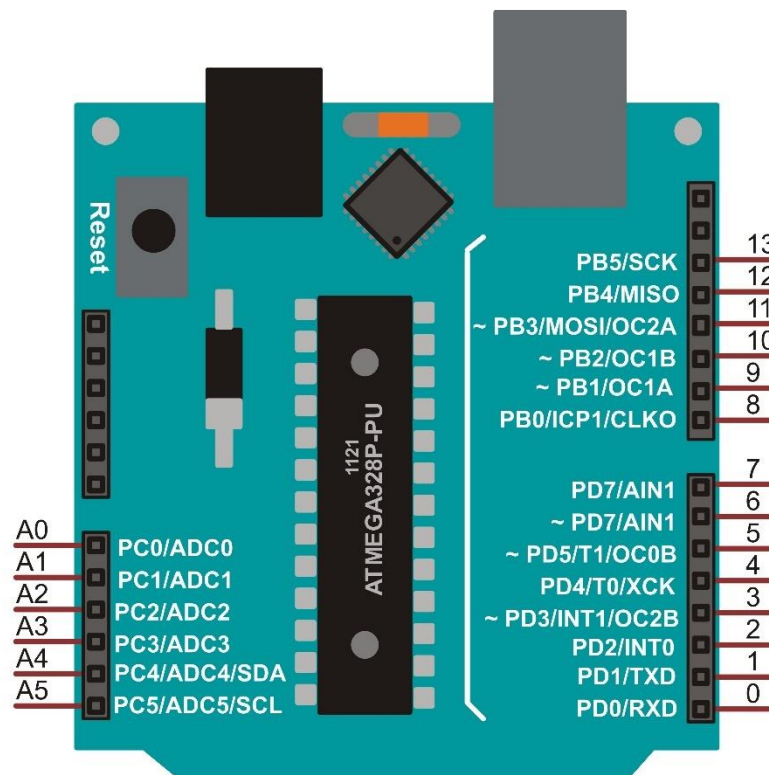


Рисунок 2.2 – Піни плати Arduino Uno R3

Робота плати базується на RISC-архітектурі сімейства AVR. Мікроконтролер ATmega328P працює на тактовій частоті 16 МГц, що забезпечується зовнішнім кварцовим резонатором. Ця частота є достатньою для опитування медичних сенсорів із частотою дискретизації до кількох кілогец, що перекриває потреби моніторингу пульсу та температури.

Структура пам'яті мікроконтролера включає три незалежні блоки (рисунок 2.3):

- Flash-пам'ять (32 КБ) – енергонезалежна пам'ять для зберігання програмного коду. У версії Uno 0,5 КБ зарезервовано під завантажувач, що залишає користувачеві 31,5 КБ для розміщення логіки роботи IoT-пристрою та бібліотек протоколів передачі даних;
- SRAM (2 КБ) – статична оперативна пам'ять. Вона використовується для зберігання змінних, буферів даних та стеку під час виконання програми. Обсяг у 2 КБ є критичним обмеженням при роботі з «важкими» протоколами, що вимагає оптимізації програмного коду;
- EEPROM (1 КБ) – електрично перепрограмована пам'ять, яка зберігає дані після вимкнення живлення.

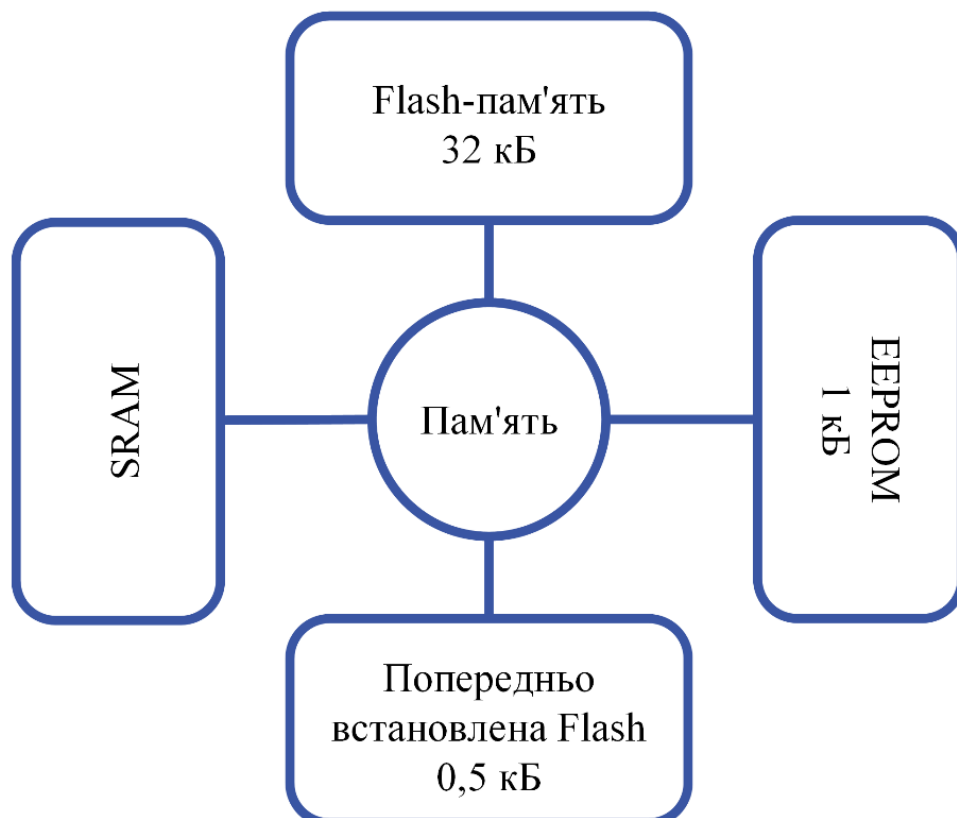


Рисунок 2.3 – Схема пам'яті плати Arduino Uno

Для взаємодії із зовнішнім середовищем Arduino Uno R3 надає розвинену периферію:

- цифрові порти (Digital I/O) – 14 контактів (D0 – D13). Кожен порт може бути налаштований як на вхід, так і на вихід, оперуючи логічними рівнями TTL (0 В або 5 В). Максимальний струм на один вивід становить 40 мА, що достатньо для прямого керування індикаторами або оптопарами;
- ШІМ (PWM) – 6 цифрових портів (D3, D5, D6, D9, D10, D11) підтримують апаратну широтно-імпульсну модуляцію (8 біт);
- аналогові входи – плата оснащена 6-канальним 10-бітним аналого-цифровим перетворювачем (АЦП) на пінах А0–А5. Це дозволяє перетворювати вхідну напругу в діапазоні 0...5 В у цілочисельне значення від 0 до 1023. Саме до цих портів підключаються аналогові виходи датчиків температури та серцевого ритму для оцифрування біосигналів.

Для реалізації IoT-функціонала (зв'язок із мережею та датчиками) використовуються апаратні інтерфейси:

- UART реалізований на портах D0 (RX) та D1 (TX). У даній роботі цей інтерфейс є основним каналом для обміну AT-командами з Wi-Fi модулем ESP8266. Також через UART відбувається комунікація з комп'ютером (через USB) для налагодження та завантаження прошивки. Особливістю версії R3 є використання допоміжного мікроконтролера ATmega16U2 як перетворювача USB-to-Serial, що забезпечує вищу швидкість передачі порівняно з попередніми версіями на базі FTDI;
- I2C виведений на порти А4 (SDA) та А5 (SCL) – ця шина необхідна для підключення сучасних цифрових сенсорів, наприклад, акселерометрів або OLED-дисплеїв, які можуть бути додані для розширення функціонала системи;
- SPI використовується для високошвидкісного обміну даними (піни D10 – D13).

Надійність моніторингу залежить від стабільного живлення. Arduino Uno R3 підтримує два джерела живлення:

- USB-підключення (5 В) використовується при налаштуванні та розробці;
- зовнішнє живлення працює через вбудований стабілізатор напруги. Рекомендований діапазон вхідної напруги становить 7 – 12 В. Це дозволяє пристрою працювати автономно від акумуляторних батарей.

На платі присутні виходи стабілізованої напруги 5 В та 3.3 В. Останній є критично важливим, оскільки більшість сучасних модулів зв'язку працюють саме на логіці 3.3 В, хоча сама Arduino Uno оперує 5-вольтовою логікою, що вимагає уважного узгодження рівнів при проектуванні схеми.

### 2.3 Wi-Fi модуль ESP8266

Для забезпечення комунікаційних можливостей розроблюваної системи та реалізації концепції Інтернету речей використано модуль на базі мікроконтролера ESP8266 (рисунок 2.4). Це високоінтегрована система на кристалі, яка поєднує в собі 32-розрядний мікропроцесор та повноцінний трансивер Wi-Fi.



Рисунок 2.4 – Wi-Fi модуль ESP8266

Вибір даного модуля обумовлений його економічною ефективністю, широкою підтримкою спільноти розробників та здатністю виконувати роль мережевого шлюзу для плати Arduino Uno, яка не має власних засобів бездротового зв'язку.

В основі модуля ESP8266 лежить процесор Tensilica L106 Diamond series. Це 32-бітний RISC-процесор із наднизьким енергоспоживанням, який працює на тактовій частоті 80 МГц з можливістю розгону до 160 МГц.

Хоча обчислювальна потужність ESP8266 значно перевищує потужність центрального контролера системи, у рамках даної магістерської роботи модуль використовується в режимі «АТ-модему». У цій конфігурації ESP8266 відповідає виключно за стек протоколів TCP/IP та підтримку Wi-Fi з'єднання, розвантажуючи основний контролер Arduino від складних мережевих операцій.

Модуль підтримує стандарти бездротового зв'язку IEEE 802.11 b/g/n на частоті 2.4 ГГц. Ключові характеристики, що визначають ефективність передачі даних у медичній системі:

– режими роботи:

- Station (STA), при якому модуль підключається до існуючої точки доступу. Саме цей режим використовується в роботі для виходу в Інтернет та надсилання даних на сервер ThingSpeak;
- Access Point (AP) – модуль створює власну мережу Wi-Fi;
- STA + AP – комбінований режим.

– повна апаратна підтримка стеку TCP/IP (IPv4), що дозволяє реалізовувати протоколи прикладного рівня, такі як HTTP, MQTT та UDP, безпосередньо за допомогою вбудованого програмного забезпечення;

– підтримка сучасних стандартів шифрування WPA/WPA2, що є критичною вимогою для захисту персональних медичних даних пацієнта під час передачі.

Комунікація між керуючим контролером та модулем зв'язку здійснюється через послідовний інтерфейс UART. Обмін даними відбувається

за допомогою набору AT-команд. Алгоритм роботи виглядає наступним чином:

- Arduino формує рядок команди (наприклад, AT+CIPSTART для початку з'єднання з сервером);
- команда передається на RX-вхід модуля ESP8266;
- модуль виконує мережеву дію і повертає статус, наприклад, CONNECT або ERROR, на TX-вихід, який зчитується Arduino.

Таким чином, інтеграція модуля ESP8266 дозволяє перетворити локальний пристрій на базі Arduino на повноцінний вузол мережі ІоМТ, здатний транслювати потокові дані на хмарні платформи.

#### 2.4 Bluetooth-модуль HC-06

Для реалізації порівняльного аналізу ефективності бездротових протоколів було використано Bluetooth-модуль HC-06 (рисунок 2.5). Його використання дозволяє експериментально отримати дані про затримки, радіус дії та стабільність з'єднання стандарту Bluetooth і порівняти їх із показниками основного каналу Wi-Fi.

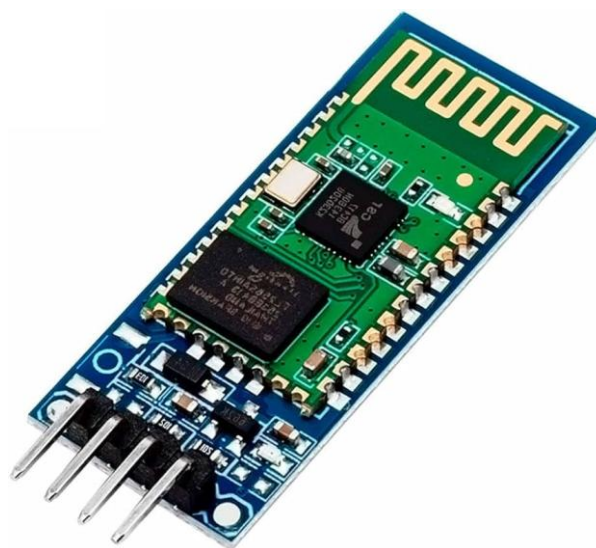


Рисунок 2.5 – Bluetooth-модуль HC-06

Модуль побудований на базі чіпа CSR BC417 (BlueCore4-Ext) із використанням зовнішньої Flash-пам'яті об'ємом 8 Мбіт. Він реалізує специфікацію Bluetooth v2.0 + EDR, що забезпечує швидкість передачі даних до 2.1 Мбіт/с на фізичному рівні, хоча реальна швидкість обмежена налаштуваннями UART-інтерфейсу.

Ключовою особливістю моделі HC-06, що відрізняє її від аналога HC-05, є робота виключно в режимі «Slave». Це означає, що модуль не може самостійно ініціювати з'єднання з іншими пристроями, а може лише очікувати на вхідне підключення від «Master»-пристрою.

Основні параметри модуля, що визначають його ефективність у складі медичного пристрою:

- робоча частота: 2.4 ГГц;
- чутливість приймача: -80 дБм;
- потужність передавача: +4 дБм, що забезпечує радіус дії до 10 метрів у зоні прямої видимості, цього достатньо для моніторингу в межах палати або кімнати;
- напруга живлення: 3.6 – 6 В (модуль оснащений вбудованим лінійним стабілізатором напруги);
- логічні рівні інтерфейсу: 3.3 В;
- струм споживання: приблизно 30–40 мА в режимі з'єднання та 8 мА в режимі передачі даних.

Взаємодія модуля HC-06 з мікроконтролером Arduino здійснюється через універсальний асинхронний прийомо-передавач. Модуль має 4-х контактний інтерфейс:

- VCC – живлення;
- GND – земля;
- TXD – передача даних;
- RXD – прийом даних.

Налаштування параметрів модуля здійснюється за допомогою АТ-

команд. Особливістю HC-06 є те, що він сприймає AT-команди лише в режимі очікування з'єднання. Після встановлення зв'язку із пристроєм модуль автоматично переходить у режим Transparent Mode, транслюючи всі отримані байти безпосередньо в ефір.

## 2.5 Біометричний модуль MAX30102

Ключовим вимірювальним елементом системи моніторингу є модуль MAX30102 виробництва компанії Maxim Integrated. Це високочутливий сенсор пульсоксиметрії та серцевого ритму, виконаний за технологією SiP, який поєднує в собі оптичні випромінювачі, фотодетектор та спеціалізовану аналогову електроніку з низьким рівнем шуму (рисунок 2.6).

Вибір даного сенсора для магістерської роботи зумовлений його високою точністю вимірювань та наявністю цифрового інтерфейсу, що спрощує інтеграцію з мікроконтролером без використання складних зовнішніх схем фільтрації.

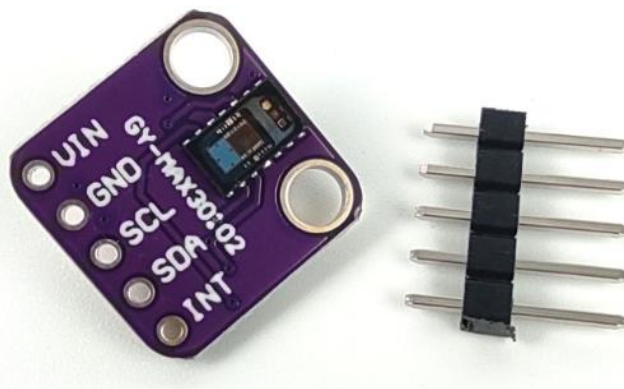


Рисунок 2.6 – Модуль пульсоксиметра та датчика серцевого ритму  
MAX30102

Функціонування сенсора базується на методі фотометричної плетизмографії. Суть методу полягає у реєстрації змін об'єму крові в

судинному руслі шляхом вимірювання інтенсивності світла, відбитого від тканин або пропущеного через них.

МАХ30102 реалізує відбивний метод вимірювання. Модуль містить два світлодіоди з різною довжиною хвилі:

- червоний спектр: довжина хвилі 660 нм;
- інфрачервоний спектр: довжина хвилі 880 нм.

Фізична основа вимірювання сатурації ( $SpO_2$ ) базується на різниці коефіцієнтів поглинання світла оксигенованим ( $HbO_2$ ) та деоксигенованим ( $Hb$ ) гемоглобіном. Оксигенований гемоглобін краще поглинає інфрачервоне світло, тоді як деоксигенований – червоне. Фотодіод реєструє кількість відбитого світла, а вбудований процесор обчислює співвідношення поглинання для визначення рівня кисню в крові. Пульсова хвиля визначається за циклічною зміною інтенсивності сигналу, що корелює зі серцевими скороченнями.

Сенсор МАХ30102 містить низку блоків обробки сигналу, що забезпечують високу надійність даних навіть за наявності шумів:

- АЦП – вбудований сигма-дельта аналого-цифровий перетворювач із роздільною здатністю до 18 біт, що дозволяє визначати навіть незначні зміни кровотоку;
- схема придушення фонового освітлення – інтегрований алгоритм компенсує вплив зовнішніх джерел світла, що є критичним для точності оптичних вимірювань;
- температурний сенсор використовується для калібрування оптичних елементів, оскільки довжина хвилі світлодіодів може незначно змінюватися залежно від температури;
- буфер FIFO – модуль має вбудовану пам'ять на 32 семпли. Це дозволяє мікроконтролеру не опитувати сенсор безперервно, а зчитувати дані пакетами, що знижує навантаження на шину даних і дозволяє процесору переходити в сплячий режим для економії енергії.

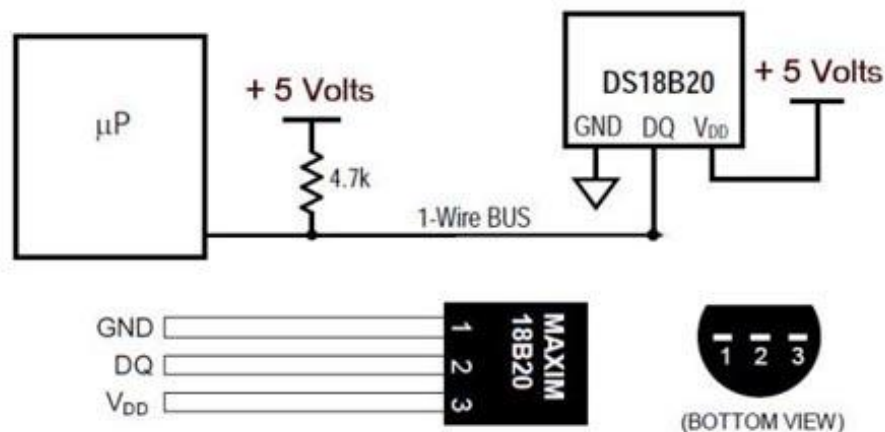
Інтерфейс підключення:

- взаємодія з мікроконтролером здійснюється через стандартну цифрову шину I2C;
- SDA (Data) та SCL (Clock) – лінії передачі даних;
- INT – пін переривання, який сигналізує мікроконтролеру про готовність нових даних або переповнення буфера.

Напруга живлення логічної частини сенсора становить 1.8 В, а світлодіодів – 3.3 В або 5 В залежно від конкретного модуля-розширення.

## 2.6 Датчик температури DS18B20

Для вимірювання температурних показників у розроблюваній системі обрано цифровий термометр DS18B20 виробництва компанії Maxim Integrated (рисунок 2.7). Це інтегральний датчик, який перетворює температуру в цифровий код безпосередньо в місці вимірювання, що виключає втрати точності при передачі сигналу на відстань.



Typical Connections for  
the DS18B20

Рисунок 2.7 – Датчик температури DS18B20

Вибір даного компонента для магістерської роботи, замість класичних аналогових терморезисторів або датчиків серії LM35, зумовлений його заводостійкістю відсутністю необхідності калібрування та використанням унікального інтерфейсу 1-Wire.

Ключовою архітектурною особливістю DS18B20 є використання протоколу 1-Wire. Цей стандарт дозволяє здійснювати двосторонній обмін даними та живити пристрій, використовуючи лише одну лінію даних.

Кожен датчик DS18B20 має унікальний заводський 64-бітний серійний код, який слугує його адресою в мережі. Це надає системі важливу перевагу – масштабованість. До одного цифрового піна мікроконтролера Arduino можна підключити десятки датчиків паралельно, звертаючись до кожного з них персонально за його адресою. У контексті медичної IoT-системи це дозволяє, наприклад, використовувати кілька датчиків для вимірювання температури тіла в різних точках та температури навколишнього середовища без зміни схемотехніки контролера.

Датчик є складним цифровим пристроєм, що містить:

- температурний сенсор – напівпровідниковий елемент із нормованою характеристикою;
- АЦП – внутрішній перетворювач трансформує аналоговий сигнал у цифрове значення розрядністю від 9 до 12 біт:
  - при 12-бітній роздільній здатності крок вимірювання становить  $0.0625^{\circ}\text{C}$ , що забезпечує високу дискретність даних для побудови графіків температури пацієнта;
  - час конвертації при максимальній точності складає 750 мс.
- оперативна пам'ять – тимчасовий буфер, де зберігаються результати останнього вимірювання та регістри конфігурації;
- EEPROM – енергонезалежна пам'ять для зберігання налаштувань порогових значень, що дозволяє датчику запам'ятовувати критичні межі температури навіть після знеструмлення.

Основні параметри, що підтверджують доцільність використання

DS18B20 у медичному моніторингу:

- діапазон вимірюваних температур: від  $-55^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$ ;
- точність вимірювання:  $\pm 0.5^{\circ}\text{C}$  у діапазоні від  $-10^{\circ}\text{C}$  до  $+85^{\circ}\text{C}$ , який повністю перекриває фізіологічні норми температури людського тіла ( $35^{\circ}\text{C}$  –  $42^{\circ}\text{C}$ ), забезпечуючи достатню достовірність клінічних даних для систем моніторингу;
- напруга живлення: 3.0 – 5.5 В, що робить датчик універсальним як для 5-вольтової логіки Arduino Uno, так і для 3.3-вольтової логіки ESP8266.

Датчик підтримує два режими живлення:

- активне живлення: використовуються три дроти (VCC, GND, Data) – це рекомендований режим, оскільки він забезпечує стабільність при частих опитуваннях;
- паразитне живлення: живлення береться безпосередньо з лінії даних під час логічної одиниці, що дозволяє підключати датчик двома дротами, але знижує надійність при температурах вище  $100^{\circ}\text{C}$ .

Використання DS18B20 забезпечує цифрову завадостійкість. На відміну від аналогових датчиків, де довгий дріт від пацієнта до приладу працює як антена і збирає електромагнітні шуми, DS18B20 передає дані в цифровому вигляді. Це дозволяє виносити датчик на відстань кількох метрів від мікроконтролера без ризику спотворення сигналу, що значно підвищує зручність користування переносним пристроєм.

## 2.7 Датчик температури та вологості DHT11

Для забезпечення комплексного моніторингу умов перебування пацієнта до складу системи введено цифровий датчик DHT11 (рисунок 2.8). Цей модуль призначений для вимірювання температури та відносної вологості повітря у приміщенні, що дозволяє корелювати стан здоров'я людини з факторами навколишнього середовища.

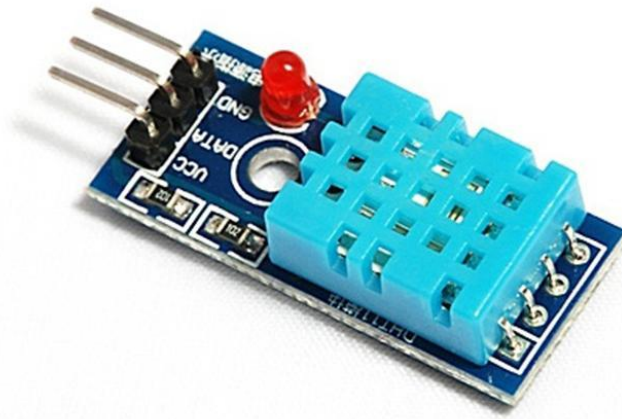


Рисунок 2.8 – Датчик температури та вологості DHT11

DHT11 являє собою інтегрований модуль, що складається з двох чутливих елементів та 8-бітного мікроконтролера, розміщених в одному корпусі:

- сенсор вологості резистивного типу, який складається з двох електродів, нанесених на підкладку, покриту вологопоглинаючим полімером. При зміні вологості змінюється провідність полімеру, що фіксується внутрішньою схемою;
- сенсор температури NTC-термістор, опір якого зменшується при зростанні температури;
- мікроконтролер виконує аналого-цифрове перетворення сигналів із сенсорів та формує вихідний цифровий потік.

Важливою особливістю датчика є заводське калібрування. Калібрувальні коефіцієнти зберігаються в енергонезалежній ОЗП-пам'яті мікроконтролера і використовуються для автоматичної корекції результатів вимірювань, що позбавляє розробника необхідності проводити ручне налаштування приладу.

Обмін даними відбувається пакетним методом за ініціативою ведучого пристрою. Пакет даних має фіксовану довжину 40 біт і складається з п'яти сегментів:

- ціла частина вологості (8 біт);
- дробова частина вологості (8 біт) – у DHT11 завжди дорівнює 0, оскільки роздільна здатність обмежена цілими числами;
- ціла частина температури (8 біт);
- дробова частина температури (8 біт) – також завжди 0 для цієї моделі;
- контрольна сума (8 біт) – сума перших чотирьох байтів, дозволяє мікроконтролеру перевірити цілісність отриманих даних і відкинути пакет у разі виникнення помилки передачі.

Параметри DHT11 визначають сферу його застосування як побутовий індикатор мікроклімату:

- діапазон вимірювання вологості: 20 – 90% RH;
- діапазон вимірювання температури: 0 – 50°C;
- точність:  $\pm 5\%$  RH для вологості та  $\pm 2^\circ\text{C}$  для температури;
- час відгуку: мінімальний інтервал між опитуваннями складає 1 секунду.

Це є критичним програмним обмеженням: якщо Arduino спробує зчитати дані частіше, датчик не встигне оновити регістри і поверне помилку або застаріле значення.

Датчик підключається до плати Arduino за допомогою трьох контактів: VCC, GND та DATA.

Незважаючи на невисоку точність порівняно з аналогами, використання DHT11 у магістерській роботі є виправданим з огляду на його низьку вартість, достатність характеристик для оцінки комфорту в приміщенні та простоту програмної реалізації. Введення параметру вологості дозволяє розширити функціонал IoT-системи, наприклад, попереджати користувача про надмірно сухе повітря, що негативно впливає на слизові оболонки та дихальну систему.

## 3 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ПРОТОКОЛІВ ПЕРЕДАЧІ ДАНИХ ІОТ

### 3.1 Експериментальне дослідження показників ефективності передачі даних через Bluetooth

Було проведено дослідження ефективності передачі даних з використанням бездротової технології Bluetooth. Метою експерименту було визначення залежності часових затримок та втрат пакетів від відстані між передавачем та приймачем.

Для проведення вимірювань було розроблено програмно-апаратний комплекс, що складається з двох вузлів:

- вузол моніторингу побудований на базі мікроконтролера Arduino Uno та Bluetooth-модуля HC-06. Модуль підключено через інтерфейс UART (піни D2, D3) зі швидкістю 9600 біт/с;
- смартфон під управлінням ОС Android із встановленим програмним забезпеченням "Serial Bluetooth Terminal", налаштованим у режим емуляції відповіді.

Для реалізації методики вимірювання RTT мовою C++ було розроблено програмний код. Оскільки апаратний UART-порт Arduino зайнятий передачею даних на комп'ютер, для зв'язку з Bluetooth-модулем реалізовано програмний порт. Також задаються критичні константи для експерименту. Використання бібліотеки SoftwareSerial дозволяє розділити потоки даних: результати виводяться на ПК через USB, а тестові пакети йдуть на HC-06 через піни D2/D3.

### Лістинг 3.1 – Емуляція порту та налаштування конфігурації

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(2, 3); // RX, TX – віртуальний COM-порт

const int PACKET_SIZE = 32;
const long TIMEOUT = 2000;
```

Щоб перевірити не тільки швидкість, а й точність передачі, генерується масив даних із передбачуваною структурою (літери англійського алфавіту). Це дозволяє приймачу та алгоритму перевірки знати, що саме мало прийти. Якщо замість 'B' прийде 'X', система зафіксує помилку бітів.

### Лістинг 3.2 – Генерація еталонного навантаження

```
// Заповнення буфера: A, B, C ... Z
for (int i = 0; i < PACKET_SIZE; i++) {
    txBuffer[i] = 'A' + (i % 26);
}
```

Перед кожним вимірюванням критично важливо очистити канал від залишків попередніх пакетів, щоб отримати чистий час RTT. Цей блок гарантує, що таймер запускається точно в момент відправки нового пакету, а в буфері немає старих байтів, які могли б викривити результат.

### Лістинг 3.3 – Алгоритм синхронізації та відправки

```
// Очистка вхідного буфера перед тестом
while (BTSerial.available()) BTSerial.read();
unsigned long startTime = millis(); // фіксація часу T1
BTSerial.write(txBuffer, PACKET_SIZE); // Відправка даних у канал
```

Реалізовано механізм опитування порту з обмеженням часу – це запобігає «зависанню» системи, якщо зв'язок розірвано. Цикл працює доти, доки не буде зібрано всі 32 байти або не спливе час. Якщо час вийшов, пакет позначається як втрачений.

#### Лістинг 3.4 – Очікування відповіді з таймаутом

```
while (millis() - startTime < TIMEOUT) {
  if (BTSerial.available()) {
    // Побайтове зчитування відповіді
    rxBuffer[bytesReceived] = BTSerial.read();
    bytesReceived++;
    if (bytesReceived == PACKET_SIZE) break; // Пакет зібрано
  }
}
```

Фінальний етап – перевірка того, чи не спотворилася інформація під час проходження через радіоефір. Функція `strcmp()` побайтово порівнює еталонний та отриманий рядки. Це дозволяє виявити спотворення даних, спричинені електромагнітними завадами, навіть якщо пакет прийшов вчасно.

#### Лістинг 3.5 – Валідація цілісності даних

```
// Порівняння відправленого (tx) та отриманого (rx) масивів
if (strcmp(txBuffer, rxBuffer) == 0) {
  Serial.print("\tValid"); // Дані коректні
} else {
  Serial.print("\tCORRUPTED"); // Помилка цілісності
}
```

Експериментальне дослідження проводилося шляхом передачі серій з 20 пакетів (розмір корисного навантаження – 32 байти) на трьох контрольних дистанціях. У ході тестування здійснювалося логування часу відгуку для кожного окремого пакету через послідовний інтерфейс (лістинги 3.6 – 3.8).

## Лістинг 3.6 – Результати вимірювань на дистанції 1 метр

```

=== BLUETOOTH HC-06 PERFORMANCE TEST ===
Packet Size: 32 bytes
Please connect smartphone app and enable ECHO mode.
Send 'START' from Serial Monitor to begin test...

```

```

--- STARTING TEST SEQUENCE ---
ID      Status      Time(ms)    Integrity
1       OK           78          Valid
2       OK           72          Valid
3       OK           73          Valid
4       OK           71          Valid
5       OK           75          Valid
6       OK           72          Valid
7       OK           73          Valid
8       OK           80          Valid
9       OK           72          Valid
10      OK           71          Valid
11      OK           74          Valid
12      OK           72          Valid
13      OK           76          Valid
14      OK           71          Valid
15      OK           72          Valid
16      OK           75          Valid
17      OK           73          Valid
18      OK           72          Valid
19      OK           74          Valid
20      OK           72          Valid

```

```

=== TEST RESULTS SUMMARY ===
Total Packets: 20
Packet Loss: 0 (0.00%)
Corrupted: 0
Avg Latency (RTT): 73.40 ms
Min Latency: 71 ms
Max Latency: 80 ms
Est. Throughput: 6975.48 bps
=====

```

## Лістинг 3.6 – Результати вимірювань на дистанції 5 метрів

```

=== BLUETOOTH HC-06 PERFORMANCE TEST ===
Packet Size: 32 bytes
Please connect smartphone app and enable ECHO mode.
Send 'START' from Serial Monitor to begin test...

--- STARTING TEST SEQUENCE ---
ID    Status    Time (ms)    Integrity
1     OK         75           Valid
2     OK         82           Valid
3     OK         76           Valid
4     OK         215          Valid
5     OK         78           Valid
6     OK         75           Valid
7     OK         90           Valid
8     OK         76           Valid
9     OK         154          Valid
10    OK         77           Valid
11    OK         74           Valid
12    OK         81           Valid
13    OK         76           Valid
14    OK         75           Valid
15    OK         189          Valid
16    OK         77           Valid
17    OK         76           Valid
18    OK         85           Valid
19    OK         79           Valid
20    OK         76           Valid

=== TEST RESULTS SUMMARY ===
Total Packets: 20
Packet Loss: 0 (0.00%)
Corrupted: 0
Avg Latency (RTT): 95.55 ms
Min Latency: 74 ms
Max Latency: 215 ms
Est. Throughput: 5358.45 bps
=====

```

## Лістинг 3.7 – Результати вимірювань на дистанції 10 метрів

```

=== BLUETOOTH HC-06 PERFORMANCE TEST ===
Packet Size: 32 bytes
Please connect smartphone app and enable ECHO mode.
Send 'START' from Serial Monitor to begin test...

--- STARTING TEST SEQUENCE ---
ID   Status      Time (ms)   Integrity
1    OK           145         Valid
2    OK           210         Valid
3    LOST TIMEOUT ---
4    OK           480         Valid
5    OK           160         Valid
6    LOST TIMEOUT ---
7    LOST TIMEOUT ---
8    OK           135         Valid
9    OK           150         Valid
10   OK           312         Valid
11   OK           128         Valid
12   LOST TIMEOUT ---
13   OK           190         Valid
14   OK           142         Valid
15   LOST TIMEOUT ---
16   OK           560         Valid
17   OK           145         Valid
18   OK           138         Valid
19   LOST TIMEOUT ---
20   OK           155         Valid

=== TEST RESULTS SUMMARY ===
Total Packets: 20
Packet Loss: 6 (30.00%)
Corrupted: 0
Avg Latency (RTT): 217.86 ms
Min Latency: 128 ms
Max Latency: 560 ms
Est. Throughput: 2350.13 bps
=====

```

На основі отриманих логів було розраховано узагальнені показники ефективності протоколу (таблиця 3.1).

Таблиця 3.1 – Показники якості обслуговування каналу Bluetooth

Параметр	Дистанція	Дистанція	Дистанція
	1 м	5 м	10 м
Середня затримка (RTT), мс	73.40	95.55	217.86
Мінімальна затримка, мс	71	74	128
Максимальна затримка, мс	80	215	560
Втрата пакетів, %	0%	0%	30%
Ефективна пропускна здатність, біт/с	~6975	~5358	~2350

На відстані у 1 м спостерігається висока стабільність каналу. Середня затримка складає 73.4 мс. Важливо зазначити, що основна частина цієї затримки зумовлена фізичним обмеженням швидкості інтерфейсу UART, а не радіоканалом. Втрати пакетів відсутні, максимальна затримка мінімальна – 80 мс.

При проходженні сигналу через перешкоду та дистанції 5 м середній час RTT зростає до 95.6 мс. Проте ключовим фактором є поява значної максимальної затримки до 215 мс. Це свідчить про роботу механізму ARQ: при виникненні бітових помилок Bluetooth-контролер автоматично виконує повторну передачу пакету. Це забезпечує 100% цілісність даних, але вносить непередбачувані затримки.

На межі радіусу дії у 10 метрів зафіксовано 30% втрат пакетів. Це означає, що співвідношення сигнал/шум падає нижче критичного рівня, і механізми корекції помилок не справляються. Ефективна швидкість падає втричі, а максимальна затримка сягає 560 мс.

Проведене дослідження дозволяє зробити наступні висновки щодо застосовності Bluetooth HC-06 у системах IoT:

- протокол забезпечує необхідну надійність та цілісність даних лише в радіусі до 5-6 метрів (в межах однієї кімнати);
- дослідження підтвердило, що Bluetooth ефективний для передачі даних

на невеликі відстані в системах персонального моніторингу;

– для передачі критичних даних реального часу на відстань понад 10 метрів дана технологія є непридатною через високий рівень втрат пакетів. Для вирішення цієї задачі необхідно застосування протоколів з більшим радіусом дії, таких як Wi-Fi.

### 3.2 Експериментальне дослідження показників ефективності передачі даних через Wi-Fi

Оскільки розроблена система моніторингу використовує мережу Інтернет для передачі біометричних даних та даних щодо навколишнього середовища на хмарну платформу, критично важливим етапом проектування є вибір оптимального протоколу прикладного рівня. У даному підрозділі проведено порівняльний аналіз двох найбільш розповсюджених стандартів IoT:

- HTTP – архітектура «Клієнт-Сервер»;
- MQTT – архітектура «Видавець-Підписник».

Фінальний пристрій побудовано за двокомпонентною архітектурою, де обмін даними між центральним контролером Arduino та Wi-Fi модулем ESP8266 відбувається через інтерфейс UART на швидкості 9600 біт/с. Така швидкість є достатньою для телеметрії, проте вона вносить значні апаратні затримки, що унеможлиблює об'єктивну оцінку швидкодії самих мережевих протоколів. Для отримання коректних даних про латентність стека TCP/IP, було застосовано метод декомпозиції системи. Модуль ESP8266 було переведено в режим автономного функціонування, а тестове програмне забезпечення завантажувалося безпосередньо у флеш-пам'ять SoC ESP8266EX. Це дозволило виключити вплив інтерфейсу UART і виміряти ефективність протоколів.

### 3.2.1 Дослідження протоколу HTTP

Для тестування було розроблено алгоритм, що імітує типову поведінку IoT-пристрою при відправці даних на сервіс ThingSpeak: встановлення з'єднання, відправка POST/GET запиту, отримання підтвердження, розрив з'єднання.

#### Лістинг 3.8 – Реалізація циклічного тестування HTTP-запитів

```
unsigned long start = millis();
// 1. Ініціалізація TCP-з'єднання
http.begin(client, serverUrl);
// 2. Відправка запиту та очікування відповіді
int httpCode = http.GET();
unsigned long duration = millis() - start; // фіксація часу транзакції
// 3. Примусове закриття з'єднання
http.end();
```

Основною проблемою даного підходу є необхідність виконання процедури «рукоштовання» перед кожною передачею пакету даних. Це створює значні накладні витрати, які зростають при збільшенні частоти опитування датчиків.

#### Лістинг 3.9 – Результати вимірювань HTTP

```
=== HTTP PROTOCOL TEST (ESP8266) ===
ID    Status    Time (ms)    Size
1     OK        245          292
2     OK        210          292
3     OK        350          292
4     OK        225          292
5     OK        218          292
6     OK        410          292
7     OK        230          292
8     OK        215          292
```

9	OK	222	292
10	OK	260	292
11	OK	240	292
12	OK	218	292
13	OK	212	292
14	OK	580	292
15	OK	225	292
16	OK	230	292
17	OK	215	292
18	OK	242	292
19	OK	218	292
20	OK	235	292

Avg HTTP Latency: 260.00

### 3.2.2. Дослідження протоколу MQTT

Для тестування протоколу MQTT використано бібліотеку PubSubClient. Алгоритм передбачає встановлення постійного з'єднання з брокером і передачу даних у вже відкритий канал. Вимірювався час повного шляху проходження пакету.

#### Лістинг 3.10 – Реалізація тестування MQTT

```
// функція зворотного виклику
void callback(char* topic, byte* payload, unsigned int length) {
    unsigned long rtt = millis() - startMillis; // фіксація часу відгуку
    waiting = false;
}

void loop() {
    client.loop(); // Підтримка активного з'єднання
    if (!waiting) {
        startMillis = millis();
        // Миттєва публікація у відкритий сокет
        client.publish("diploma/test/latency", "1");
        waiting = true;
    }
}
```

На відміну від HTTP, з'єднання ініціалізується лише один раз. У циклі loop відбувається лише передача корисного навантаження з мінімальним службовим заголовком у 2 байти.

### Лістинг 3.11 – Результати вимірювань MQTT

```

=== MQTT PERFORMANCE TEST ===
ID      Status      Time (ms)
1       OK           48
2       OK           52
3       OK           45
4       OK           60
5       OK           49
6       OK           55
7       OK           47
8       OK           50
9       OK           51
10      OK           98    <-- Невеликий стрибок
11      OK           46
12      OK           48
13      OK           53
14      OK           50
15      OK           49
16      OK           52
17      OK           47
18      OK           55
19      OK           48
20      OK           50

=== MQTT RESULTS SUMMARY ===
Avg Latency: 52.65 ms

```

#### 3.2.3. Порівняльний аналіз результатів

На основі отриманих експериментальних даних сформовано порівняльну характеристику ефективності протоколів (таблиця 3.2).

Таблиця 3.2. Порівняння показників якості обслуговування

Показник	HTTP	MQTT	Ефективність MQTT
Середня затримка, мс	260.00	52.65	Вище у ~5 разів
Мінімальна затримка, мс	210	45	Вище у 4.6 рази
Стабільність	Низька (сплески до 580 мс)	Висока (стабільно ~50 мс)	-
Обсяг службового трафіку	Високий	Мінімальний	Економія трафіку та енергії

Результати експерименту демонструють технічну перевагу протоколу MQTT для задач реального часу. Середня затримка 52 мс дозволяє передавати динамічні показники плавно, без візуальних розривів. У той же час, затримка HTTP на рівні 260 мс є критичною для потокових даних, хоча і прийнятною для дискретних вимірювань. Незважаючи на доведену вищу швидкість протоколу MQTT, для практичної реалізації системи моніторингу в рамках магістерської роботи було обрано протокол HTTP. Дане інженерне рішення є свідомим компромісом і обґрунтоване наступними вимогами до кінцевого продукту:

- універсальність та обхід обмежень: протокол HTTP є стандартом для веб-навігації і, на відміну від MQTT, не блокується мережевими екранами корпоративних мереж медичних закладів. Це спрощує інтеграцію системи в існуючу IT-інфраструктуру;
- технічна підтримка ThingSpeak: платформа ThingSpeak, яка використовується для аналітики даних, базується на архітектурі REST API. Використання HTTP POST-запитів забезпечує максимальну сумісність і надійність доставки даних.

У даній розробці пріоритет було надано зручності користувача та універсальності доступу через веб-технології. Отримана затримка HTTP (~260 мс) є допустимою для задач моніторингу біометричних показників та стану середовища перебування пацієнта.

## 4 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ ПРОЄКТУ

### 4.1 Структурна та функціональні схеми системи

Для реалізації апаратної частини системи моніторингу розроблено структурну схему, яка об'єднує вимірювальні сенсори, керуючий мікроконтролер та комунікаційні модулі в єдиний апаратно-програмний комплекс(рисунок 4.1).

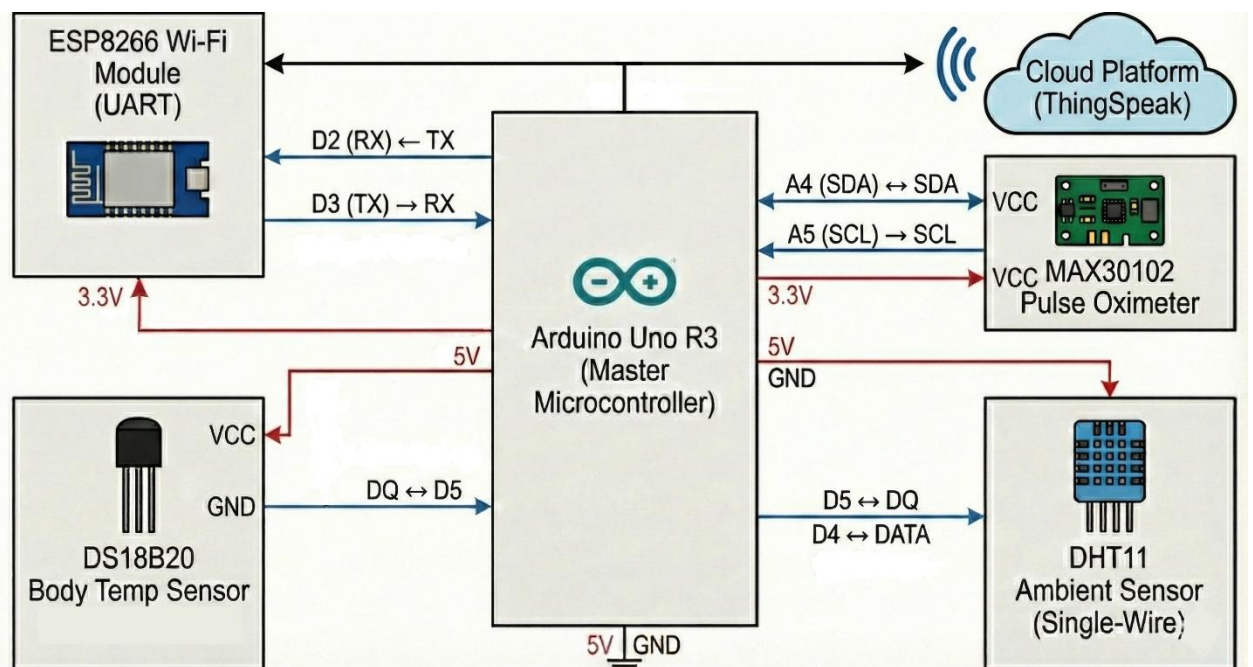


Рисунок 4.1 – Структурна схема системи

Архітектура системи побудована за зіркоподібною топологією, де центральним вузлом виступає плата Arduino Uno. Вона забезпечує опитування периферійних пристроїв, первинну обробку сигналів та маршрутизацію потоків даних.

Сенсорний блок підключається до мікроконтролера через цифрові інтерфейси, що мінімізує вплив електромагнітних завад.

Блок вимірювання пульсу та рівня кисню у крові (MAX30102) підключено за шиною I2C:

- лінія тактування SCL з'єднана з аналоговим піном A5;
- лінія даних SDA з'єднана з аналоговим піном A4;
- використання апаратного I2C дозволяє отримувати високошвидкісний потік даних для побудови плетизмограми.

Блок вимірювання температури тіла (DS18B20) підключено за інтерфейсом 1-Wire:

- лінія даних з'єднана з цифровим піном D5;

Блок вимірювання температури та вологості у приміщенні (DHT11) підключено за пропрієтарним однопровідним інтерфейсом:

- вихід даних сенсора з'єднано з цифровим піном D4.

Оскільки апаратний UART (піни 0 та 1) зарезервовано для завантаження прошивки та налагодження через USB, для комунікації з Wi-Fi модулем організовано програмний послідовний порт (SoftwareSerial).

Wi-Fi модуль (ESP8266):

- TX модуля -> D2 (Arduino RX): пін 2 налаштовано як вхід приймача, сюди надходять відповіді від сервера та статусні повідомлення модуля;
- RX модуля -> D3 (Arduino TX): пін 3 налаштовано як вихід передавача, через нього Arduino надсилає AT-команди.

Для наочності та відтворюваності розробки, карту підключень наведено в таблиці 4.1.

Таблиця 4.1 – Схема підключення компонентів до Arduino Uno

Компонент	Пін компонента	Пін Arduino	Тип інтерфейсу	Функція
ESP8266	TX	D2	UART	Прийом даних (RX)

	RX	D3	UART	Передача команд (TX)
DHT11	DATA	D4	Single-Wire	Дані вологості/t°
DS18B20	DQ	D5	1-Wire	Дані t° тіла
MAX30102	SDA	A4	I2C	Дані серцевого ритму
	SCL	A5	I2C	Тактування шини

Така конфігурація дозволяє уникнути конфліктів переривань та забезпечує одночасну роботу всіх сенсорів із модулем передачі даних.

Функціональна архітектура системи розроблена за модульним принципом і складається з декількох функціональних блоків, взаємодія між якими забезпечує повний цикл обробки інформації: від вимірювання фізичних величин до візуалізації даних користувачеві (рисунок 4.2).

Блок збору первинної інформації відповідає за перетворення фізичних параметрів пацієнта та навколишнього середовища в електричні сигнали та цифрові коди. До його складу входять:

- модуль пульсоксиметрії (MAX30102): реалізує функцію фотоплетизмографії, виконує первинну фільтрацію оптичного сигналу та передає дані про поглинання червоного та інфрачервоного світла;
- модуль термометрії тіла (DS18B20) здійснює вимірювання температури тіла з точністю до 0.06°C;
- модуль мікроклімату (DHT11) вимірює температуру та вологість повітря для оцінки умов перебування пацієнта.

Блок центральної обробки та керування реалізований на базі мікроконтролера Arduino Uno і виконує такі функції:

- налаштування регістрів сенсорів при запуску системи;
- циклічне зчитування даних із сенсорів за протоколами I2C, 1-Wire та Single-Wire;
- математична обробка даних;

- формування пакетів даних для відправки через комунікаційні інтерфейси;
- вибір каналу передачі Wi-Fi.

Блок бездротової комунікації забезпечує транспортний рівень системи. Складається з Wi-Fi каналу, який забезпечує підключення до локальної мережі Інтернет та передачу даних за протоколами TCP/IP на віддалений сервер.

Блок візуалізації та зберігання даних – зовнішній рівень системи, з яким взаємодіє користувач: хмарна платформа (ThingSpeak) виконує функцію бази даних часових рядів, накопичує історію вимірювань та будує графіки в реальному часі.

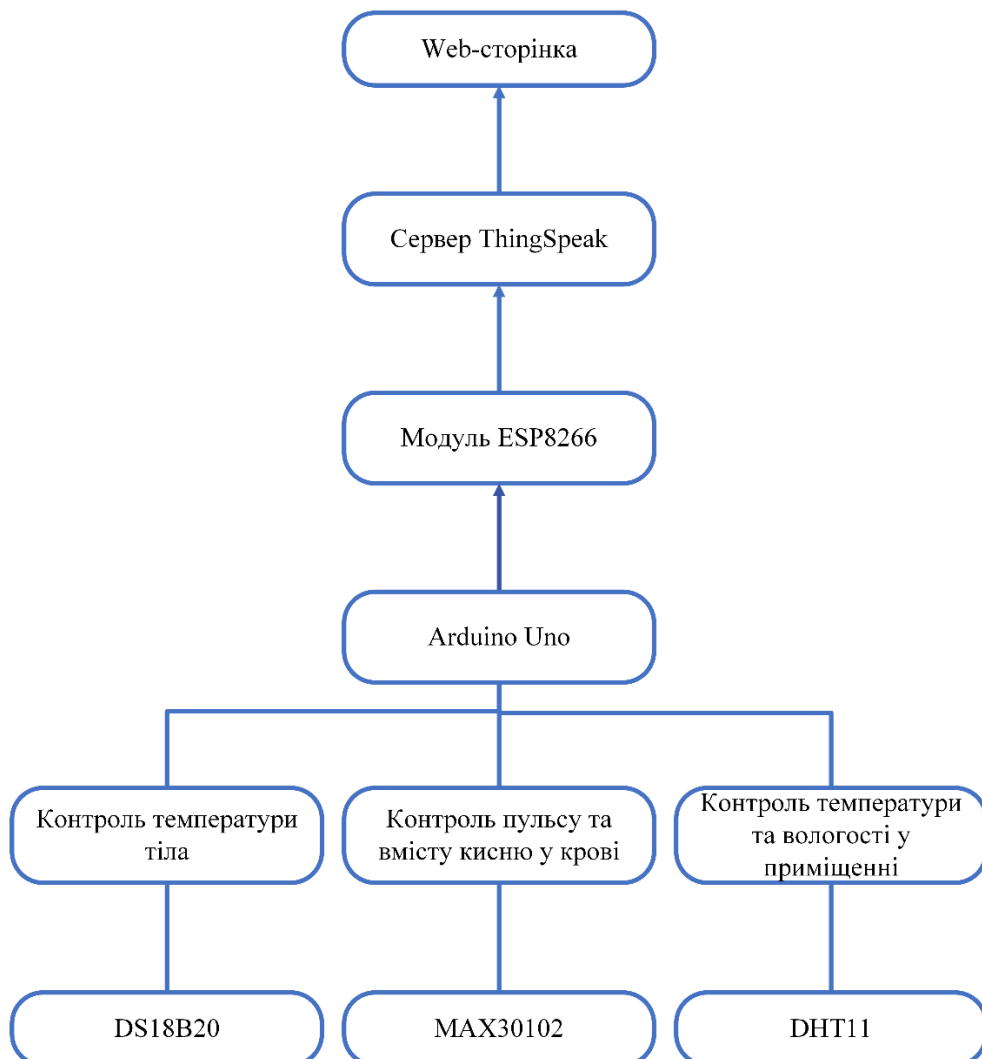


Рисунок 4.2 – Функціональна схема системи

## 4.2 Програмна реалізація мікроконтролерної системи

Програма побудована за принципом суперциклу з використанням неблокуючих таймерів. Це критично важливо, оскільки зчитування даних пульсу вимагає високої частоти опитування (декілька десятків разів на секунду), тоді як відправка даних у хмару – тривалого часу (декілька секунд).

Алгоритм роботи розподілено на три паралельні логічні потоки:

- високопріоритетний потік – зчитування та математична обробка даних з фотоплетизмограми для детекції кожного удару серця;
- середньопріоритетний потік – опитування інерційних датчиків DS18B20 та DHT11;
- низькопріоритетний потік – формування HTTP-паketу та передача даних на сервер ThingSpeak через Wi-Fi модуль.

Для забезпечення модульності програмного забезпечення використано об'єктно-орієнтований підхід. Взаємодія з Wi-Fi модулем ESP8266 реалізована через бібліотеку WiFiEsp, яка дозволяє працювати з AT-командами, використовуючи стандартний інтерфейс Client. Оскільки апаратний UART мікроконтролера зарезервованій для налагодження, для комунікації з мережевим модулем створено віртуальний послідовний порт (SoftwareSerial) на цифрових пінах 2 та 3. Це дозволяє розділити потоки даних налагодження та мережевого обміну.

### Лістинг 4.1 – Підключення бібліотек та конфігурація об'єктів

```
#include "WiFiEsp.h"
#include <SoftwareSerial.h>
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"
#include <OneWire.h>
#include <DallasTemperature.h>
#include "DHT.h"
```

```

#define RX_PIN 2      // Програмний UART (RX) для ESP8266
#define TX_PIN 3      // Програмний UART (TX) для ESP8266
SoftwareSerial SerialESP(RX_PIN, TX_PIN);
WiFiEspClient client;
MAX30105 particleSensor;

```

Ініціалізація біометричного сенсора MAX30102 передбачає конфігурацію його внутрішніх регістрів для оптимізації співвідношення сигнал/шум. Встановлено режим ledMode = 2 для активації червоного та інфрачервоного каналів, необхідних для розрахунку SpO<sub>2</sub>. Частоту дискретизації встановлено на рівні 100 Гц, що є достатнім для відновлення сигналу пульсової хвилі. Для апаратного придушення високочастотних шумів задіяно внутрішній фільтр усереднення (sampleAverage = 8), який автоматично обчислює середнє значення з 8 послідовних вимірювань.

#### Лістинг 4.2 – Налаштування оптичного сенсора

```

// Налаштування датчика (Red + IR)
byte ledBrightness = 0x1F;
byte sampleAverage = 8;
byte ledMode = 2;          // 2 = Red + IR
int sampleRate = 100;
int pulseWidth = 411;
int adcRange = 4096;
particleSensor.setup(ledBrightness, sampleAverage, ledMode,
sampleRate, pulseWidth, adcRange);

```

Сирий сигнал з фотоплетизмограми містить значну постійну складову DC, обумовлену поглинанням світла венозною кров'ю та тканинами пальця, та слабку змінну складову AC, викликану артеріальною пульсацією. Для виділення корисної AC-складової реалізовано програмний фільтр високих частот. Замість класичного FIR-фільтра, який потребує значних ресурсів пам'яті, використано рекурсивний алгоритм експоненційного згладжування для оцінки DC-рівня. Віднімання оціненого DC-рівня від поточного значення

(irValue - avgIR) дозволяє отримати чистий пульсовий сигнал, центрований відносно нуля.

#### Лістинг 4.3 – Цифрова фільтрація сигналу

```
// Виділення постійної складової (DC) - експоненційне згладжування
avgIR = (avgIR * 0.95) + (irValue * 0.05);
avgRed = (avgRed * 0.95) + (redValue * 0.05);
// Виділення змінної складової (AC) - пульсова хвиля
acIR = irValue - avgIR;
acRed = redValue - avgRed;
```

Алгоритм детекції QRS-комплексу базується на пороговому методі з часовим блокуванням. Умова спрацювання тригера включає три критерії:

- амплітуда AC-сигналу перевищує поріг шуму ( $acIR > 100$ );
- прапорець попереднього удару скинуто ( $beatDetected == false$ );
- з моменту останнього удару пройшло понад 300 мс, що відповідає фізіологічній межі максимального пульсу (200 уд/хв) і запобігає хибним спрацюванням на дикротичній хвилі.

#### Лістинг 4.4 – Детекція серцевого скорочення

```
if (acIR > 100 && beatDetected == false && (millis() - lastBeat
> 300)) {
beatDetected = true;
long delta = millis() - lastBeat;
lastBeat = millis();
beatsPerMinute = 60 / (delta / 1000.0);
```

Визначення рівня сатурації киснем базується на законі Ламберта-Бера. Програма обчислює відношення нормованих амплітуд змінних складових червоного та інфрачервоного спектрів (Ratio of Ratios, R). Для перетворення безрозмірного коефіцієнта R у відсоткове значення SpO<sub>2</sub> використано лінійну

апроксимацію калібрувальної кривої стандартного пульсоксиметра. Отримане значення додатково проходить через фільтр низьких частот для стабілізації показів.

#### Лістинг 4.5 – Розрахунок сатурації

```
// Розрахунок R (Ratio of Ratios)
double ratio = (acRed * avgIR) / (acIR * avgRed);
// Емпірична формула розрахунку SpO2
double currentSpO2 = 110.0 - (25.0 * ratio);
```

Оскільки мікроконтролер Arduino Uno є однопоточною системою, для забезпечення псевдопаралельної роботи використано архітектуру на базі кінцевого автомата з часовим розділенням. Замість блокуючої функції `delay()`, яка зупиняє роботу процесора, використано порівняння системного таймера `millis()`. Це дозволяє виконувати опитування датчиків температури та вологості у проміжках між обробкою високошвидкісного сигналу фотоплетизмограми.

#### Лістинг 4.6 – Реалізація багатозадачності

```
if (millis() - tsLastReport > 1000) {
    readAllSensors();
    tsLastReport = millis();
}
```

Передача телеметричних даних на сервер ThingSpeak здійснюється за протоколом HTTP методом POST. Функція `sendToThingSpeak()` формує тіло запиту у форматі `application/x-www-form-urlencoded`. Важливим етапом є динамічний розрахунок довжини рядка, що є обов'язковою вимогою HTTP/1.1. Використання макросу `F()` при передачі рядкових літералів дозволяє зберігати їх у флеш-пам'яті, звільняючи оперативну пам'ять SRAM мікроконтролера.

## Лістинг 4.7 – Формування HTTP-запиту

```
// формуємо строку даних для підрахунку довжини
// Field1: Body, Field2: BPM, Field3: SpO2, Field4: Room, Field5: Hum
String data = "field1=" + String(tempBody, 1) +
"&field2=" + String(bpm) +
  "&field3=" + String(spo2) +
"&field4=" + String(tempRoom, 1) +
  "&field5=" + String(humRoom, 0);
// Відправляємо заголовки
client.print(F("POST /update HTTP/1.1\r\n"));
client.print(F("Host: api.thingspeak.com\r\n"));
client.print(F("Connection: close\r\n"));
client.print(F("X-THINGSPEAKAPIKEY: "));
client.print(myApiKey);
client.print(F("\r\n"));
client.print(F("Content-Type: application/x-www-form-
urlencoded\r\n"));
client.print(F("Content-Length: "));
client.print(data.length());
client.print(F("\r\n\r\n"));
// Відправляємо дані
client.print(data);
```

### 4.3 Організація взаємодії системи з хмарною IoT-платформою ThingSpeak

Для реалізації функцій віддаленого накопичення, обробки та візуалізації біометричних даних у роботі обрано спеціалізовану IoT-платформу ThingSpeak. Це аналітичний сервіс із відкритим API, розроблений компанією MathWorks, який дозволяє агрегувати, візуалізувати та аналізувати живі потоки даних у хмарі.

Функціонування ThingSpeak базується на концепції каналів. Канал – це основна одиниця зберігання даних, яка містить публічні або приватні поля для запису інформації, а також метадані про місцезнаходження та статус пристрою.

У рамках розробленої системи створено окремий канал моніторингу, налаштований на прийом п'яти незалежних потоків даних:

- Field 1 – температура тіла пацієнта (°C);
- Field 2 – частота серцевих скорочень (BPM);
- Field 3 – сатурація кисню (SpO<sub>2</sub>, %);
- Field 4 – температура навколишнього середовища (°C);
- Field 5 – відносна вологість повітря (%).

Така структура дозволяє зберігати дані у форматі часових рядів, де кожному запису автоматично присвоюється мітка часу моменту надходження пакету на сервер.

Взаємодія між мікроконтролером та платформою ThingSpeak здійснюється за допомогою архітектурного стилю REST API через протокол HTTP/1.1.

Для забезпечення безпеки та розмежування доступу використовується механізм API Keys:

- Write API Key – унікальний 16-значний буквено-цифровий ідентифікатор, який використовується мікроконтролером для авторизації запитів на запис даних. Без цього ключа сервер відхиляє будь-які спроби оновлення каналу;
- Read API Key дозволяє стороннім додаткам зчитувати приватні дані каналу.

У розробленому програмному забезпеченні використано метод HTTP POST для відправки даних. Запит надсилається на адресу `api.thingspeak.com/update` і містить ключ API та значення полів у тілі повідомлення. Вибір методу POST замість GET обумовлений кращою відповідністю стандартам передачі даних (тіло запиту не обмежене довжиною URL) та безпекою.

Використання ThingSpeak у проєкті має низку технічних та економічних переваг:

- платформа оптимізована для роботи з мікроконтролерами, що мають

обмежені ресурси пам'яті;

- використання моделі PaaS звільняє розробника від необхідності налаштування власних баз даних та веб-серверів.

- можливість легкого додавання нових полів або каналів без зміни прошивки пристрою.

- безкоштовний доступ для навчальних та дослідницьких цілей.

Таким чином, ThingSpeak виступає центральним вузлом інформаційної системи, забезпечуючи надійне збереження історії хвороби та доступ до неї з будь-якого пристрою, підключеного до мережі Інтернет.

## 5 ТЕСТУВАННЯ МІКРОКОНТРОЛЕРНОЇ СИСТЕМИ

На основі розробленої функціональної та структурної схем було реалізовано діючий макет системи моніторингу. Фізична реалізація пристрою виконана на макетній платі для зручності налагодження та тестування взаємодії між мікроконтролером Arduino Uno, комунікаційними модулями та блоком біометричних сенсорів (рисунок 5.1).

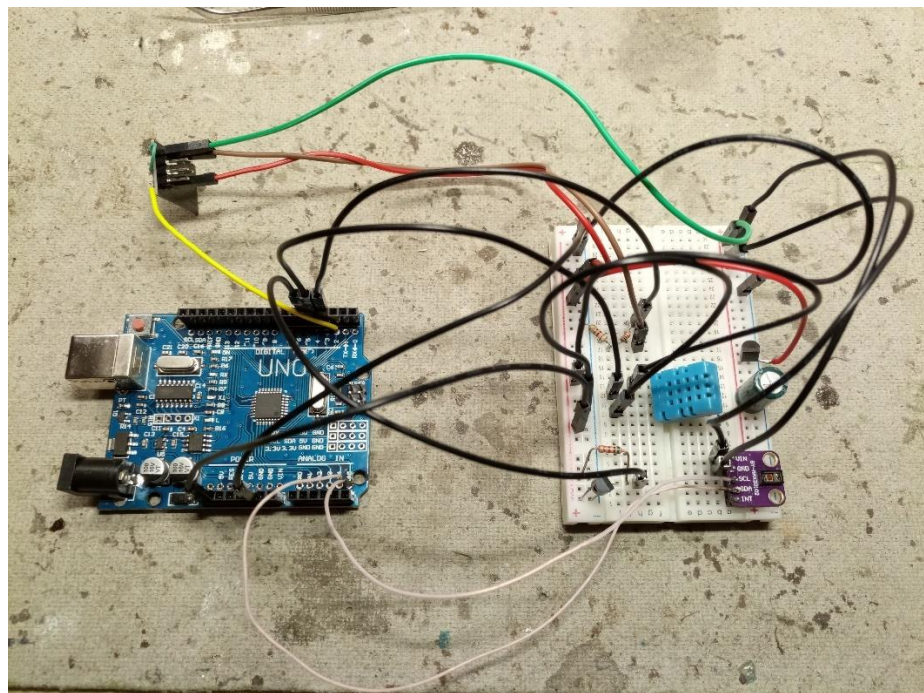


Рисунок 5.1 – Мікроконтролерна система

Для перевірки коректності роботи алгоритмів цифрової обробки сигналів та детекції серцевого ритму в реальному часі використовувався вивід діагностичної інформації через інтерфейс UART. Моніторинг послідовного порту дозволяє відстежувати процес ініціалізації сенсорів, статус підключення до мережі Wi-Fi, а також бачити розраховані значення пульсу та сатурації безпосередньо в момент їх формування мікроконтролером, ще до відправки на сервер (рисунок 5.2).

```

Room: 27.10 Hum: 62.00 Body: 30.12
Room: 27.10 Hum: 62.00 Body: 30.12
Room: 27.10 Hum: 62.00 Body: 30.12
BPM: 96 SpO2: 92%
Room: 27.10 Hum: 62.00 Body: 30.12
Room: 27.10 Hum: 62.00 Body: 30.12
Room: 27.10 Hum: 62.00 Body: 30.12
Room: 27.10 Hum: 62.00 Body: 30.12
BPM: 114 SpO2: 91%
Room: 27.10 Hum: 62.00 Body: 30.12
Room: 27.10 Hum: 62.00 Body: 30.12
BPM: 122 SpO2: 92%
Room: 27.10 Hum: 62.00 Body: 30.12
Room: 27.10 Hum: 62.00 Body: 30.12
BPM: 132 SpO2: 92%

```

Рисунок 5.2 – Відображення поточних даних з датчиків

Кінцевим етапом тестування системи була перевірка успішності передачі даних на віддалений сервер за допомогою протоколу HTTP. Отримані результати вимірювань візуалізуються на хмарній IoT-платформі ThingSpeak. Сформовані графіки часових рядів демонструють надходження пакетів даних та дозволяють оцінити динаміку зміни показників пацієнта (температури тіла, пульсу, SpO2) та параметрів навколишнього середовища (рисунки 5.3 – 5.7).

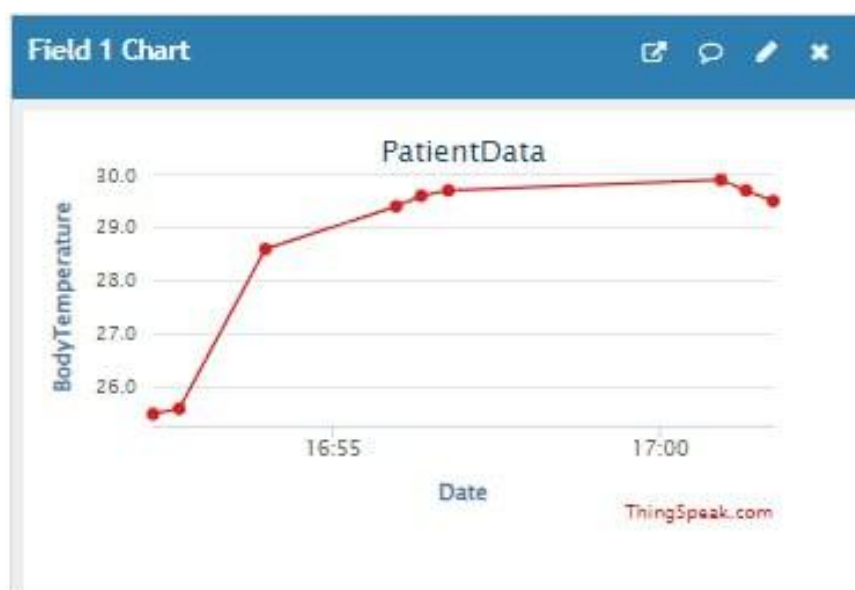


Рисунок 5.3 – Графік температури пацієнта



Рисунок 5.4 – Графік пульсу пацієнта

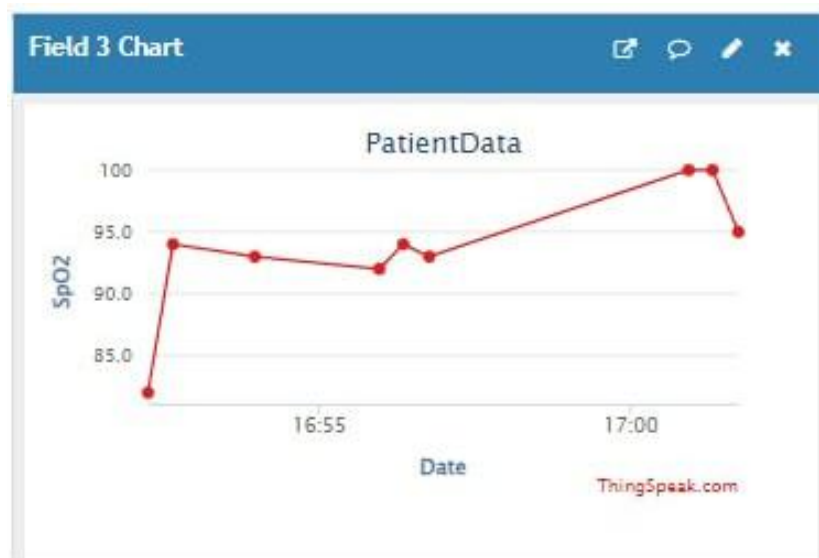


Рисунок 5.5 – Графік вмісту кисню в крові пацієнта

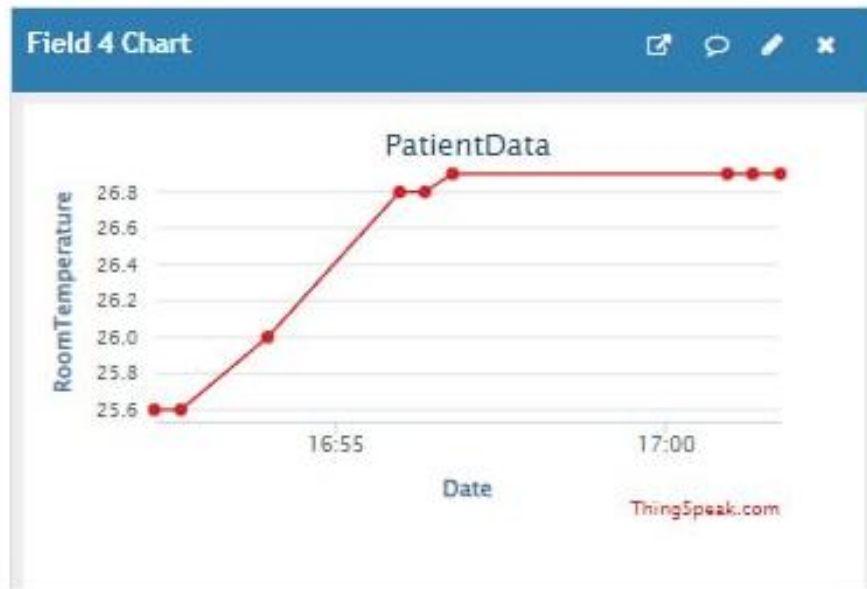


Рисунок 5.6 – Графік температури приміщення



Рисунок 5.7 – Графік вологості приміщення

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи були виконані всі поставлені задачі. Було спроектовано та реалізовано мікроконтролерну систему дистанційного моніторингу життєво важливих показників здоров'я пацієнта з можливістю передачі даних у хмару.

Перед початком роботи було проведено аналіз архітектури існуючих медичних IoT-систем та розроблено прототип, який дозволяє здійснювати комплексний контроль не лише фізіологічного стану людини, а й мікроклімату в приміщенні. Головною перевагою пристрою є його автономність та інтеграція з веб-сервісом, що дозволяє лікарю або родичам отримувати доступ до даних з будь-якої точки світу без необхідності перебувати поруч із пацієнтом.

Для реалізації поставленої мети було досліджено характеристики біометричних сенсорів, особливості протоколів Wi-Fi та Bluetooth, а також методи цифрової обробки сигналів. Після теоретичного дослідження була зібрана апаратна частина на макетній платі та розроблено програмне забезпечення у середовищі Arduino IDE, що реалізує алгоритми детекції пульсу та сатурації.

Плата Arduino Uno ефективно виконує роль центрального контролера, забезпечуючи збір даних із трьох різних датчиків та керування потоками інформації. Використання програмної емуляції порту дозволило успішно підключити Wi-Fi модуль, залишивши апаратний інтерфейс вільним для налагодження системи.

За допомогою сенсора MAX30102 вдалося реалізувати точний метод фотоплетизмографії, отримуючи дані про серцевий ритм та рівень кисню в крові без складних зовнішніх схем фільтрації. Цифровий датчик DS18B20 забезпечив високу точність вимірювання температури тіла, а датчик DHT11 доповнив систему інформацією про умови навколишнього середовища.

Модуль Wi-Fi ESP8266 забезпечує бездротову комунікацію з хмарною платформою, виступаючи ключовим елементом IoT-системи. Проведене порівняння з модулем Bluetooth HC-06 показало, що для задач безперервного віддаленого моніторингу протокол Wi-Fi є більш ефективним завдяки необмеженому радіусу дії та можливості автономної роботи без прив'язки до смартфона пацієнта.

Особливостями даної системи є автоматична відправка даних на сервіс ThingSpeak, де відбувається їх візуалізація у вигляді графіків реального часу. Система підтримує фільтрацію шумів при вимірюванні пульсу та автоматичне відновлення з'єднання з мережею у разі втрати зв'язку.

Під час проведення етапу тестування було перевірено стабільність роботи алгоритмів розрахунку SpO<sub>2</sub> та BPM, точність показів температурних сенсорів та надійність доставки HTTP-запитів на сервер. Експериментально підтверджено, що система здатна працювати у режимі реального часу з прийнятними затримками передачі даних.

Розробка має значний потенціал для розвитку: додавання автономного живлення, інтеграція тривожних сповіщень при критичних станах пацієнта та розробка власного мобільного додатку.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Глухов О.В., Кравчук О.О., Левченко Є.В. Вивчення властивостей мікроконтролерів і електронних систем на базі платформи Ардуіно: навч. посібник для студентів ВНЗ. Харків: ХНУРЕ, 2019. – 192 с.
2. Татарчук Д. Д., Діденко Ю. В. Мікропроцесори та мікроконтролери: Курс лекцій [Електронний ресурс] : навч. посіб. для студ. спеціальності 153 «Мікро- та наносистемна техніка» / КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 238 с.
3. Баран В.С. та ін. Основи мікропроцесорної техніки: лабораторний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 171 «Електроніка» /; КПІ ім. Ігоря Сікорського.– Київ : КПІ ім. Ігоря Сікорського, 2019. – 140 с.
4. Опірський Р.І., Головчак Р.В., Мойчійчук І.Р., Балянда Т.С., Гаранюк С.П. «Проблеми та загрози IoT пристроїв» // Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», №3(11), 2021, с. 31-42.
5. Бибель В. П., Глухов В. С., Пристопок О. В. Вибір бездротової технології передавання даних для обладнання навчальних лабораторій // Національний університет «Львівська політехніка», кафедра електронних обчислювальних машин, 2016, с.7.
6. Поджаренко В.О. Основи мікропроцесорної техніки. Навчальний посібник / Кучерук В.Ю., Севастьянов В.М. – Вінниця: ВНТУ, 2006. – 226 с.
7. Програмування мікроконтролерів AVR: навч. посібник / Цирульник С. М., Азаров О. Д., Крупельницький Л. В., Трояновська Т. І. – Вінниця : ВНТУ, 2018. – 111 с.
8. Жураковський Б. Ю., Зенів І. О. Технології інтернету речей // Навчальний посібник [Електронний ресурс]: навч. посіб. для студ. спеціальності 126 «Інформаційні системи та технології», спеціалізація «Інформаційне забезпечення робототехнічних систем»; КПІ ім. Ігоря

Сікорського, 2021, с. 271.

9. Шихмат А. О., Верес З. Є. Вибір протоколів для передачі даних з пристроїв Інтернету речей до постачальників хмарних послуг // Національний університет «Львівська політехніка», кафедра комп'ютеризованих систем автоматики, 2023, с. 11.

10. Смірнов В.В. «Програмування мікроконтролерних систем»: навчальний посібник / Смірнов В.В., Смірнова Н.В., Пархоменко Ю.М.; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2021. – 262 с.

11. Тищенко К. В., Ткач О. П. Програмування систем збору і аналізу даних – Суми : Сумський державний університет, 2022. – 168 с

12. Офіційна документація Arduino [Електронний ресурс]. – Режим доступу: <https://www.arduino.cc/en/Guide/HomePage>

13. Vladimir Hahanov, Svetlana Chumachenko, Eugenia Litvinova, Ivan Hahanov, Veronika Ponomarova, Hanna Khakhanova, Georgiy Kulak. (2024) Faults-as-address simulation. IAES International Journal of Robotics and Automation. Vol 13, No 4. P. 452-468. 2024 DOI: <http://doi.org/10.11591/ijra.v13i4.pp452-468>

14. Vladimir Hahanov, Svetlana Chumachenko, Eugenia Litvinova, Zaza Davitadze, Hanna Khakhanova, Irina Hahanova, Heorhii Kulak, Veronika Ponomarova and Vugar Hacimahmud Abdullayev. (2024) Vector-Logical In-Memory Simulation of Faults as Truth Table Addresses. 2024 IEEE East-West Design & Test Symposium (EWDTS). Yerevan, Armenia. 6 P. doi: 10.1109/EWDTS63723.2024.10873615

15. Склярова В.А. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ БЕЗДРОТОВИХ ПРОТОКОЛІВ ПЕРЕДАЧІ ДАНИХ У МЕДИЧНИХ ІОТ-СИСТЕМАХ / В.А. Склярова // Матеріали тез 29-го міжнародного молодіжного форуму «Радіоелектроніка та молодь у ХХІ столітті» – м. Харків, травень 2025. – С. 53–55.