

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 перший (бакалаврський)
(рівень вищої освіти)

Модель пульсометра на базі мікроконтролера
з інтеграцією у мобільний застосунок

(тема)

Виконав: здобувач IV курсу, групи КІУКІ-21-7

Малков Є.І.

(прізвище, ініціали)

Спеціальність 123 – Комп'ютерна інженерія

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма

Комп'ютерна інженерія

(повна назва освітньої програми)

Керівник доц. Рахліс Д.Ю.

(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 проектування обчислювальної техніки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія
(шифр і назва)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«06» травня 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувача Малкова Єгора Ігоровича
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Модель пульсометра на базі мікроконтролера з інтеграцією у мобільний застосунок

затверджена наказом по університету від "21" 05 2025 р. № 403 Ст.

2. Термін подання здобувачем роботи до екзаменаційної комісії 20.06.2025

3. Вихідні дані до роботи (проекту) _____

SoC ESP32-WROOM-32D

3D принтер Creality Ender-3; SolidWorks.

Мови програмування: C, JavaScript, TypeScript

Акумулятор li-ion 3.7v, MT3608, TP4056, перемикач

Сенсор пульсу MAX30102

ОС Android

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

Огляд існуючих рішень

Моделювання апаратної та програмної частини

Вибір компонентів для реалізації апаратної частини

Створення прототипу пульсометра на базі SoC та корпусу для нього

Програмування алгоритму роботи пристрою.

Розробка програмного застосунку для ОС Android

Комплексне тестування пристрою та програмного застосунку

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____ 14 слайдів _____


6. Консультанти розділів роботи (проекту)

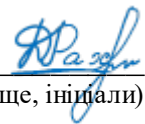
Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання етапів проекту (роботи)	Примітка
1	Видача теми проекту, узгодження і затвердження теми	06.05.2025 – 10.05.2025	виконано
2	Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів	11.05.2025 – 13.05.2025	виконано
3	Розробка моделі пристрою	14.05.2025 – 17.05.2025	виконано
4	Прототипування моделі	18.05.2025 – 25.05.2025	виконано
5	Розробка алгоритму роботи та налагодження	25.05.2025 – 05.06.2025	виконано
6	Тестування пристрою	06.06.2025 – 07.06.2025	виконано
7	Оформлення пояснювальної записки	08.06.2025 – 13.06.2025	виконано
8	Перевірка виконаного проекту керівником, допуск до захисту	14.06.2025 – 17.06.2025	виконано
9	Захист проекту	20.06.2025	

Дата видачі завдання 06.05.2025

Здобувач _____  _____
(підпис)

Керівник роботи (проекту) _____  _____
(підпис) (посада, прізвище, ініціали)

доц. Рахліс Д.Ю.

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи містить 71 сторінку, 42 рисунка, 2 таблиці, 3 додатки та 12 джерел посилань.

КОНТРОЛЬ СЕРЦЕВОГО РИТМУ, ЧАСТОТА СЕРЦЕВИХ СКОРОЧЕНЬ, МІКРОКОНТРОЛЕР, СЕНСОР, BLE, SOC, REACT NATIVE, ПУЛЬСОВІ ЗОНИ.

Метою кваліфікаційної роботи є розробка моделі пульсометра на базі мікроконтролера та інтеграція його у мобільний застосунок для операційної системи Android. В ході виконання роботи було цілісно досліджено принцип на якому базується метод вимірювання частоти серцевих скорочень, а також проаналізовано існуючі комерційні рішення подібних приладів.

Запропонована модель пульсометра в якості головного блока керування використовує плату від EspressifSystems на базі мікроконтролера ESP32-C3, та MAX30102 в якості оптичного сенсора для вимірювання ЧСС. З метою забезпечення стабільної передачі даних між пульсометром та мобільним застосунком використано протокол BLE, який дозволяє суттєво знизити енергоспоживання пристрою при збереженні надійності з'єднання. Пульсометр зібрано в ергономічний корпус, що дозволяє комфортно користуватись приладом, також прилад обладнано сигнальними світлодіодами, що забезпечує всебічну комунікацію із користувачем під час використання. Праграмна частина комплексу реалізована за допомогою ReactNativeCLI, а в якості системи керування БД обрана SQLite. Мобільний застосунок забезпечує візуалізацію даних у режимі реального часу, а також дає змогу зберігати сесії для подальшого аналізу та перегляду середніх показників ЧСС та інформації у вигляді графіків.

Особливу увагу приділено зручності інтерфейсу, який було реалізовано у Figma, з урахуванням принципів доступності та інтуїтивної навігації.

ABSTRACT

The explanatory note of the qualification work contains 71 pages, 42 figures, 2 tables, 3 appendices, and 12 reference sources.

HEART RATE CONTROL, HEART RATE, MICROCONTROLLER, SENSOR, BLE, SOC, REACT NATIVE, PULSE ZONES.

The purpose of the qualification work is to develop a model of a heart rate monitor based on a microcontroller and integrate it into a mobile application for the Android operating system. During the work, the principle on which the heart rate measurement method is based was comprehensively investigated, and existing commercial solutions for similar devices were analyzed. The proposed heart rate monitor model uses an Espressif Systems board based on the ESP32-C3 microcontroller as the main control unit, and MAX30102 as an optical sensor for measuring heart rate. In order to ensure stable data transmission between the heart rate monitor and the mobile application, the BLE protocol was used, which allows significantly reducing the device's power consumption while maintaining connection reliability. The heart rate monitor is assembled in an ergonomic case, which allows for comfortable use of the device, and the device is also equipped with signal LEDs, which provides comprehensive communication with the user during use. The software part of the complex is implemented using React Native CLI, and SQLite was chosen as the database management system. The mobile application provides real-time data visualization, and also allows you to save sessions for further analysis and viewing of average heart rate and information in the form of graphs.

Particular attention was paid to the convenience of the interface, which was implemented in Figma, taking into account the principles of accessibility and intuitive navigation.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	9
1.1 Моніторинг ЧСС як елемент цифрового самоспостереження	9
1.2 Огляд існуючих рішень.....	10
1.3 Мета роботи та постановка технічного завдання	13
2 МОДЕЛЬ ПУЛЬСОМЕТРА НА БАЗІ SOC	15
2.1 Вибір компонентів для реалізації апаратної частини комплексу	15
2.2 Моделювання принципової електричної схеми пристрою	23
2.3 Реалізація програми для апаратної частини комплексу	25
2.3.1 Алгоритм роботи пристрою	25
2.3.2 Програмне середовище розробки	27
2.3.3 Ініціалізація компонентів	29
2.3.4 Зчитування, обробка та передача даних	30
2.3.5 Оптимізація та енергозбереження	32
2.4 Проектування та розробка корпусу пульсометра	33
2.5 Складання та монтаж апаратної частини пристрою	37
2.6 Відлагодження проєкта в ArduinoIDE.....	40
3 РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ОС ANDROID	43
3.1 Вибір програмних технологій для реалізації застосунку	43
3.2 Архітектура мобільного застосунку та логіка роботи.....	44
3.3 Проектування інтерфейсу користувача.....	47
3.4 База даних для мобільного застосунку.....	52
3.5 Реалізація функції реєстрації профілю користувача	53
3.6 Сканування та підключення до BLE-пристрою.....	55
3.7 Отримання та обробка даних ЧСС	55
3.8 Запис і збереження сесій в локальну базу даних	56
3.9 Відображення графіків, статистики та історії.....	57
4 ТЕСТУВАННЯ ТА ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ КОМПЛЕКСУ	58
ВИСНОВКИ	69
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	71
ДОДАТОК А.....	72
ДОДАТОК Б.....	99
ДОДАТОК В.....	101

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД (База даних) – структура для зберігання та обробки даних.

Гц (Hz) – герц, одиниця частоти, кількість циклів за секунду.

мс (ms) – мілісекунда, одна тисячна частина секунди.

ПЗ – програмне забезпечення.

ЧСС (Частота серцевих скорочень) – кількість скорочень серця за хвилину.

BLE (BluetoothLowEnergy) – енергоефективна версія Bluetooth, призначена для бездротового обміну даними з низьким енергоспоживанням.

BLE-пристрій – пристрій, що передає або приймає дані за допомогою BluetoothLowEnergy.

BLE-профіль – набір правил і структур даних для обміну інформацією між BLE-пристроями.

GUI (GraphicalUserInterface) – візуальна частина програмне забезпечення, з якою взаємодіє користувач.

I²C – послідовний інтерфейс для підключення периферійних пристроїв до мікроконтролерів.

SoC (SystemonChip) – мікросхема, яка поєднує процесор, пам'ять, інтерфейси введення/виведення та інші компоненти на одному чипі.

UART (UniversalAsynchronousReceiver-Transmitter) – стандарт асинхронної послідовної передачі даних.

ВСТУП

Контроль фізіологічних параметрів організму стає все більш актуальним серед користувачів, які ведуть активний спосіб життя або прагнуть підвищити обізнаність щодо власного фізичного стану. Одним із ключових показників, що характеризує рівень навантаження на серцево-судинну систему, є ЧСС. Вимірювання пульсу дозволяє об'єктивно оцінити інтенсивність активності, відстежувати зміни у фізичному стані та приймати обґрунтовані рішення щодо регулювання фізичного навантаження.

Більшість доступних на ринку пристроїв для вимірювання ЧСС мають низку обмежень: закриту архітектуру, обмежений доступ до даних, відсутність можливості локального зберігання інформації або її обробки користувачем. Усе це знижує ефективність їх використання в умовах, коли важливо мати простий, надійний і керований інструмент для самопостереження.

Завдяки розвитку мікроелектроніки та вбудованих систем на кристалі (SoC), стало можливим створення компактних автономних пристроїв, що поєднують функції збору, обробки та бездротової передачі даних. З іншого боку, програмний застосунок для смартфона відкриває можливості для перегляду пульсу в реальному часі, запису вимірювань, графічного аналізу й візуалізації змін ЧСС. Подібні рішення орієнтовані насамперед на користувачів, які бажають краще розуміти реакції власного організму на повсякденні чи тренувальні навантаження, а також здійснювати самостійний контроль у зручній і доступній спосіб, без прив'язки до закритих екосистем або складних комерційних платформ.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Моніторинг ЧСС як елемент цифрового самоспостереження

Протягом останніх років спостерігається стійка тенденція до зростання попиту на персональні пристрої для контролю фізичного стану організму. Однією з ключових причин цього є поширення культури активного способу життя, що передбачає регулярні тренування, слідкування за самопочуттям і бажання бачити власний прогрес у кількісному вимірі. Усе більше людей прагнуть не лише займатись фізичною активністю, але й аналізувати вплив навантажень на організм у доступній і зрозумілій формі.

Відповідно до звіту IDC WorldwideQuarterlyWearableDeviceTracker, лише у 2023 році по всьому світу було продано понад 500 мільйонів носимих пристроїв[1], більшість із яких – це фітнес-браслети та смарт-годинники. Зростання продажів пов'язане не лише з розвитком технологій, а й із збільшенням інтересу до базового самоспостереження, зокрема за пульсом, кількістю кроків, витраченими калоріями тощо. Серед усіх фізіологічних показників, що доступні для вимірювання в побутових умовах, частота серцевих скорочень (ЧСС) посідає одне з провідних місць. Це обумовлено тим, що ЧСС напряду відображає реакцію організму на навантаження, рівень аеробної підготовки та швидкість відновлення після тренування. На відміну від крокоміра чи лічильника калорій, пульс є об'єктивним показником внутрішнього фізіологічного стану, який важко фальсифікувати чи оцінити неточно.

Особливу увагу користувачі приділяють пульсовим зонам – діапазонам ЧСС, які відповідають різним рівням інтенсивності навантаження. Виділяють п'ять основних зон:

- зона 1 – дуже легке навантаження (відновлення, розминка);
- зона 2 – легке аеробне навантаження (розвиток витривалості);

- зона 3 – помірне навантаження (зона ефективного тренування);
- зона 4 – інтенсивне навантаження (поліпшення аеробної потужності);
- зона 5 – максимальна інтенсивність (анаеробна витривалість).

Тренування в тій чи іншій зоні дозволяє досягати конкретних цілей: зниження ваги, покращення загальної фізичної форми, підготовка до змагань тощо. Знання та контроль своєї ЧСС у реальному часі, а також моніторинг та збереження статистики, дозволяє користувачу регулювати навантаження, уникати перевтоми й отримувати кращий результат при меншому ризику. Водночас, усе більше людей використовують вимірювання пульсу не лише під час занять спортом, а й у повсякденному житті – для оцінки стресу, сну, відновлення після роботи або прогулянок. Таким чином, функція моніторингу ЧСС стала частиною повсякденної цифрової культури, а пульсометр – звичним засобом самоусвідомлення, поряд зі смартфоном чи додатком для розрахунку калорій.

Ці обставини формують постійно зростаючий попит на прості, автономні та доступні рішення, які дозволяють у зручній формі отримувати ключову інформацію про власний організм. Зростання цього сегменту пояснюється не медичними потребами, а інтересом до особистого контролю та керування власною фізичною активністю.

1.2 Огляд існуючих рішень

На ринку засобів для персонального моніторингу ЧСС найпоширенішими є три типи пристроїв: фітнес-браслети, розумні годинники та автономні нагрудні пульсометри. Кожен з них має свої особливості в реалізації вимірювання, передачі даних та взаємодії з користувачем.

1. Фітнес-браслет Xiaomi MiBand 7.

Xiaomi Mi Band 7 (рис. 1.1) – популярний фітнес-браслет, орієнтований на масового користувача [2]. Для вимірювання ЧСС пристрій використовує

фотоплетизмографію (ФПГ) – метод оптичного зчитування пульсу на основі аналізу поглинання світла кров'ю.



Рисунок 1.1 – Фітнес-браслет Xiaomi Mi Band 7

Сенсор встановлено на тильній стороні браслета й працює в режимі безперервного моніторингу або за запитом користувача. Передача даних відбувається за допомогою BLE на смартфон, де використовується застосунок Mi Fit. У додатку можна аналізувати статистику за день, тиждень чи місяць, отримувати повідомлення про відхилення та задавати нагадування про тренування.

Плюси:

- компактність і довготривала автономність (до 14 днів);
- зручна візуалізація даних у застосунку;
- доступна велика кількість сумісних модулів і датчиків;
- компактні розміри, низька вартість.

До мінусу можна віднести їх залежність від хмарної синхронізації.

2. Смарт-годинник Garmin Forerunner 255.

Garmin Forerunner 255 (рис. 1.2)– пристрій професійного рівня для спортсменів і користувачів, що серйозно займаються тренуваннями. Годинник також використовує технологію оптичного ФПГ-датчика, однак реалізовану з покращеною схемою розміщення світлодіодів і

багатоканальним зчитуванням, що забезпечує високу точність навіть під час інтенсивного руху.



Рисунок 1.2 – Смарт-годинник Garmin Forerunner 255

Передача даних здійснюється через ANT+, що дозволяє підключати пристрій як до смартфона, так і до зовнішніх тренажерів або нагрудних пульсометрів. Застосунок – Garmin Connect, який підтримує глибоку аналітику, персоналізовані тренування, VO₂max, тощо.

Плюси:

- висока точність вимірювання ЧСС;
- підтримка ANT+;
- професійний рівень аналітики.

Мінуси:

- висока вартість;
- залежність від хмарної синхронізації.

3. Нагрудний датчик Polar H10.

Polar H10 (рис. 1.3)– класичний приклад електродного пульсометра, що працює на основі аналізу електрокардіографічного сигналу. Датчик складається з еластичного ремня з вбудованими електродами, які щільно

прилягають до шкіри, забезпечуючи високоточне зчитування сигналу серцевого ритму.



Рисунок 1.3 – Нагрудний датчик Polar H10

Polar H10 не має дисплея чи вбудованого інтерфейсу – вся взаємодія здійснюється через BLE. Пристрій можна підключати до смартфонів, годинників, тренажерів або професійного ПЗ.

Плюси:

- висока точність завдяки ЕКГ-зчитуванню;
- широка сумісність із застосунками та пристроями.

До мінусів можна віднести потребу у зволоженні електродів для коректної роботи.

Таким чином, сучасні пристрої для моніторингу частоти серцевих скорочень охоплюють широкий спектр сценаріїв застосування – від щоденного фітнес-контролю до професійного спортивного моніторингу з високою точністю. Кожен із розглянутих типів має свої переваги, обумовлені рівнем інтеграції, точністю, зручністю та функціональністю. Проте їхня складність, вартість, прив'язка до закритих екосистем або орієнтація на вузькі задачі створюють простір для розвитку альтернативних підходів у цій сфері – простіших, адаптивних і автономних.

1.3 Мета роботи та постановка технічного завдання

Проведений аналіз існуючих пристроїв для моніторингу частоти серцевих скорочень показав, що такі рішення користуються широким

попитом серед людей, зацікавлених у контролі своєї фізичної активності. Проте більшість із них орієнтовані або на масовий сегмент із мінімальною функціональністю, або на професійний рівень із високою вартістю та надлишковою складністю.

Метою даної роботи є створення компактного носимого пристрою, призначеного для персонального вимірювання та фіксації частоти серцевих скорочень у режимі запису за ініціативою користувача, з можливістю перегляду пульсу в реальному часі та аналізу зібраної інформації після завершення сесії. Інформація про пульс повинна бути наочно подана у вигляді числових значень та графіків, з урахуванням середніх показників, розподілу за пульсовими зонами та часових характеристик.

Основні функціональні вимоги до системи:

- безперебійна передача даних від носимого пристрою до мобільного застосунку під час активної сесії запису;
- можливість запуску та завершення сесії запису з мобільного застосунку;
- автоматичне збереження даних у локальну базу даних мобільного застосунку після завершення запису;
- наочна побудова графіка ЧСС, обчислення середнього значення, виведення базової статистики;
- доступ до історії записів із можливістю перегляду ключових параметрів кожної сесії;
- інтуїтивно зрозумілий інтерфейс застосунку без потреби в складному налаштуванні.

Таким чином, поставлена задача полягає в реалізації персонального інструменту моніторингу пульсу, орієнтованого на простоту, автономність, носимість і доступність для користувача без спеціальної підготовки. Особливу увагу має бути приділено зручності використання в реальних умовах повсякденного життя, а також наочності подання результатів.

2 МОДЕЛЬ ПУЛЬСОМЕТРА НА БАЗІ SOC

2.1 Вибір компонентів для реалізації апаратної частини комплексу

Для реалізації апаратної частини було обрано набір перевірених і доступних компонентів, що забезпечують необхідну функціональність, компактність і автономність пристрою. Основними критеріями вибору стали: наявність вільної технічної документації, сумісність між собою, стабільність роботи та достатня поширеність на ринку, а також, один з ключових факторів при виборі елементів стала необхідність зменшення габаритів пристрою без втрати функціональності. Обирання компонентів відбувалося виходячи з умов створення носимого пристрою з автономним живленням та бездротовою передачею даних.

Основою апаратної частини виступає мікроконтролер ESP32-C3FH4 на базі ESP32-C3 Type-C (рис. 2.1), який є сучасною реалізацією системи на кристалі з архітектурою RISC-V.



Рисунок 2.1 – ESP32-C3 Type-C

ESP32-C3 має вбудований BLE5.0-модуль [4] та достатній об'єм пам'яті. Окрім основних характеристик, ESP32-C3 вирізняється низьким енергоспоживанням: у режимі активної передачі BLE він споживає приблизно 80–90 мА, тоді як ESP32-WROOM-32D у схожих умовах може споживати до 180–200 мА. У режимі сну (deep sleep) ESP32-C3 потребує менше ніж 5 мкА, що майже вдвічі ефективніше за WROOM-версію, яка

споживає приблизно 10 мкА, що критично важливо для портативних пристроїв із живленням від акумулятора. Завдяки компактним розмірам та широким можливостям периферії, мікроконтролер забезпечує надійне зчитування, обробку та передачу даних без потреби у додаткових зовнішніх модулях. Інтерфейс USB Type-C полегшує програмування пристрою, що також спрощує прототипування та обслуговування. Детальні характеристики наведено у таблиці 2.1.

Таблиця 2.1 – Технічні характеристики ESP32-C3

Параметр	Значення
Мікроконтролер	Одноядерний ESP32-C3FH4 (32-bit RISC-V)
Тактова частота	160 МГц
Пам'ять ROM	384 КБ
Пам'ять SRAM	400 КБ (кеш 16 КБ)
Пам'ять SRAM (RTC)	8 КБ
Флеш-пам'ять	4 МБ
Вхідна напруга	3.3 – 5 В
Робочий струм	До 500 мА
Wi-Fi	2.4 ГГц, до 150 Мбіт/с
Bluetooth	Bluetooth 5 (LE)
GPIO / Інтерфейси	13 GPIO, SPI x1, I2S x2, UART x1, PWM, ADC
USB	Type-C
Розміри	23 × 18 мм

Таким чином можна сказати, що ESP32-C3 Type-C, у порівнянні з популярним ESP32-WROOM-32, є більш компактним і енергоефективним варіантом, що дозволяє інтегрувати його у носимий пристрій з мінімальними фізичними обмеженнями, та значно підвищити автономність пристрою, а також ергономічність та комфорт при використанні пристрою.

Під час обирання методу вимірювання ЧСС, були проаналізовані існуючі технології для вирішення цієї задачі.

Системи для вимірювання ЧСС можна умовно поділити на контактні (рис. 2.2) та безконтактні пристрої (оптичні, зображено на рисунку 2.3). Перші базуються на знятті електричних потенціалів серцевої діяльності (електрокардіографія), другі – на фотоплетизмографії [3], що реєструє зміну оптичної щільності тканин, спричинену пульсацією крові.



Рисунок 2.2 – ESP32-C3 Type-C



Рисунок 2.3 – Оптичний датчик MAX30102 для зчитування ЧСС

Контактні методи, зокрема ЕКГ, забезпечують високу точність та клінічну достовірність, але потребують складнішої апаратної реалізації, ретельного розміщення електродів та, як правило, не є зручними для щоденного використання в побутових умовах. Натомість оптичні методи, попри чутливість до артефактів (рухи, освітлення), є зручними для інтеграції в носимі пристрої, легко адаптуються до побутового використання та мають достатню точність для задач фітнес-моніторингу або базового контролю стану здоров'я.

Фотоплетизмографія (ФПГ) є оптичним методом, що дозволяє реєструвати зміни об'єму крові в м'яких тканинах на основі зміни поглинання або відбиття світлового випромінювання. Цей метод базується на тому, що кров має властивість поглинати світло в певному спектральному діапазоні, і зміни її об'єму під час кожного серцевого циклу спричиняють відповідні коливання інтенсивності світла, яке проходить через тканини [10] або відбивається від них.

У класичному виконанні ФПГ-системи використовуються два основні компоненти: джерело світла (найчастіше світлодіод) та фотодетектор (наприклад, фотодіод), які можуть бути розміщені як навпроти одне одного (трансмівна схема), так і поруч (рефлексивна схема). У першому випадку (рис. 2.4) світло проходить через тканину (наприклад, палець), і фотодетектор реєструє його інтенсивність після проходження.

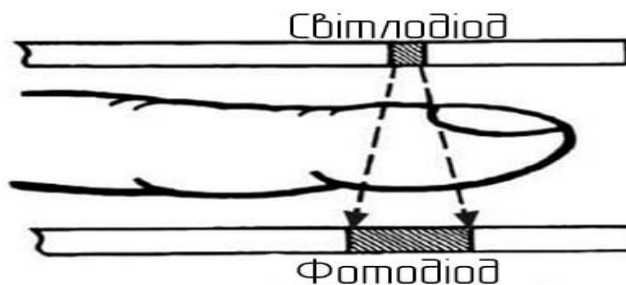


Рисунок 2.4 – Трансмівна схема ФПГ-пристрою

У другому (рис. 2.5) – світло проникає в тканини, частково відбивається від кровоносних судин і прилеглих структур, а детектор фіксує рівень цього відбиття. Саме така компоновка використовується в МАХ30102.

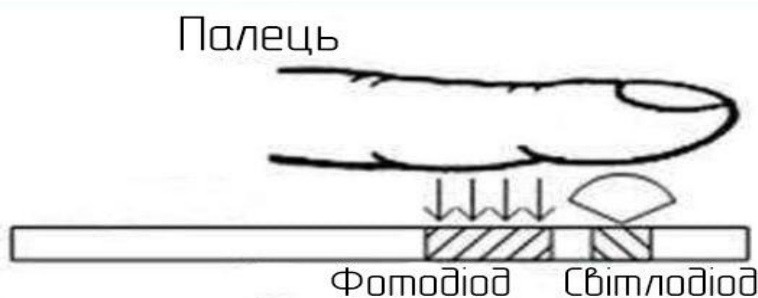


Рисунок 2.5 – Рефлексивна схема ФПГ-пристрою

Світлодіоди випромінюють світло одного або кількох конкретних кольорів (довжин хвиль), зазвичай у червоному або інфрачервоному діапазоні, оскільки ці довжини хвиль найбільше взаємодіють із гемоглобіном – основним носієм кисню в крові. Чим вищий об’єм крові в зоні реєстрації, тим більше світла поглинається, і тим менше доходить до фотодетектора. У фазі систоли, коли серце викидає кров у судини, кількість крові в тканинах збільшується, що спричиняє зниження інтенсивності зареєстрованого світла. У фазі діастоли, навпаки, інтенсивність зростає. Такі періодичні зміни й формують плетизмографічний сигнал, з якого можна виділити ритмічні піки, відповідні до кожного серцевого скорочення.

Для запобігання шумів важливо застосовувати алгоритми згладжування та фільтрації, а також правильно обрати частоту дискретизації, яка зазвичай становить 25–100 Гц для забезпечення достатньої роздільної здатності в часовому домені [3]. Детальні технічні характеристики обраного датчику для зчитування ЧСС МАХ30102, наведено у таблиці 2.2.

Таблиця 2.2 – Характеристики МАХ30102

Параметр	Значення
Вимірювані параметри	Частота пульсу, SpO ₂
Напруга живлення	3.3 В (вбудований стабілізатор 1.8 В)
Споживаний струм (режим вимірювання)	1.2 мА
Споживаний струм (режим сну)	До 10 мкА
Інтерфейс	I ² C
Напруга інтерфейсу I ² C	3.3 В
Максимальна частота I ² C	400 кГц
Розміри	18.5 × 14.5 × 3 мм

Завдяки простоті реалізації, низькій вартості, компактності та енергоефективності фотоплетизмографія стала основою більшості сучасних комерційних фітнес-трекерів, розумних годинників, пульсометрів та інших носимих пристроїв. Вона не лише дозволяє визначати ЧСС, а й створює потенціал для подальшого розширення функціоналу пристроїв – наприклад, для оцінки рівня стресу, якості сну, фізичного навантаження тощо, тому саме така технологія була обрана для використання у проектуванні носимого пульсометра.

Ще одним ключовим компонентом, зважаючи на те, що пристрій обов'язково має бути носимим, у складі апаратної частини є портативне джерело живлення. З огляду на вимогу компактності та портативності пристрою, було обрано літій-полімерний акумулятор ємністю 0.8А із габаритами 30×40×6 мм (рис 2.6).



Рисунок 2.6 – Літій-полімерний акумулятор

Такий акумулятор забезпечує баланс між місткістю, габаритами та вагою, що критично важливо для реалізації носимого пристрою. Менші акумулятори можуть обмежити час автономної роботи, тоді як більші – ускладнюють конструкцію, збільшують масу і зменшують комфорт використання.

Для зарядки акумулятора використовується модуль TP4056 (рис. 2.7).

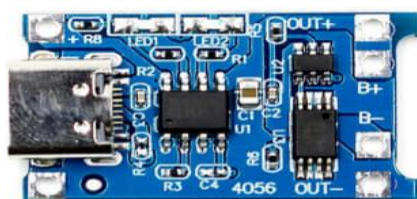


Рисунок 2.7 – Модуль зарядки акумуляторів TP4056

Він забезпечує стабільну зарядку з керованим струмом і захистом від перезаряду та глибокого розряду. Його широке розповсюдження, Type-C роз'єм, надійність і простота інтеграції роблять його стандартним рішенням у проєктах з живленням від Li-Po акумуляторів.

Модуль TP4056 має на платі 2 індикаторних СМД-світлодіода, що сигналізують про процес заряду – світлодіод червоного кольору та індикація, що акумулятор повністю зарядився – світлодіод синього кольору.

Так як СМД-світлодіод не буде видно зовні корпусу майбутнього приладу, у модуль TP4056 було внесено зміни – заміна двох СМД-світлодіодів на один двокольоровий світлодіод 3мм (рис. 2.8).

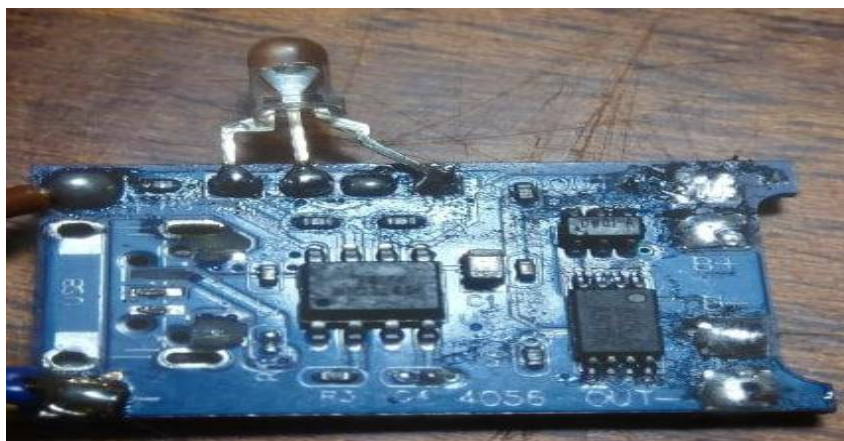


Рисунок 2.8 – Змінений модуль TP4056

Оскільки напруга акумулятора змінюється в діапазоні 3.0–4.2В, а для стабільної роботи ESP32-C3 потрібна стабільна напруга 5В, у схемі застосовано підвищувальний перетворювач напруги MT3608 (рис 2.9).



Рисунок 2.9 – DC-DC перетворювач напруги MT3608

Цей модуль забезпечує стабільне живлення мікроконтролера незалежно від рівня заряду акумулятора, що особливо важливо для збереження працездатності пристрою в умовах зниженого заряду.

Для переривання живлення приладу від акумулятора коли він не використовується обрано тумблер перемикання (рис. 2.10 (а)) та двокольоровий світлодіод для індикації подачі живлення та розряду акумулятора (рис 2.10 (б)).

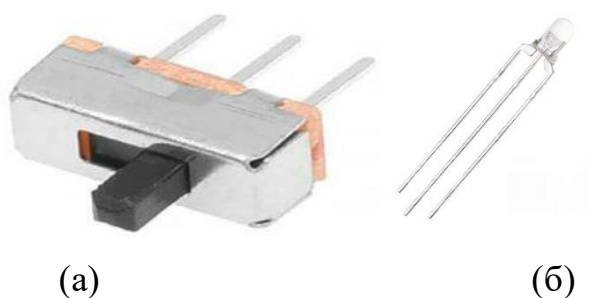


Рисунок 2.10 –Тумблер перемикання та двокольоровийсвітлодіод

Разом конфігурація, що включає ESP32-C3 Type-C, датчик MAX30102, акумулятор Li-Po 3.7В 0.8А, модуль зарядки TP4056, підвищувальний перетворювач MT3608, перемикач живлення (тумблер) та зелений світлодіод, дозволяє забезпечити:

- безпечну зарядку акумулятора;
- стабільну напругу живлення навіть при частковому розряді акумулятора;
- мінімальні габарити елементів живлення без шкоди для тривалості автономної роботи;
- повну енергетичну незалежність без прив'язки до зовнішніх джерел живлення під час використання;
- зручну інтеграцію у носиму конструкцію з низьким енергоспоживанням та постійною передачею даних;

– можливість ручного вмикання або вимикання живлення за допомогою компактного тумблера, що підвищує контроль користувача над роботою пристрою;

– візуальне підтвердження подачі живлення за допомогою зеленого індикаторного світлодіоду та червоного – розряду акумулятора.

Таким чином, саме така апаратна компоновка повністю відповідає вимогам до компактного, енергоефективного, носимого пристрою з автономним живленням, стабільною роботою та зручною експлуатацією. Це забезпечує не лише технічну відповідність поставленим технічним завданням, але й практичну зручність у повсякденному використанні, зокрема під час фізичних навантажень.

2.2 Моделювання принципової електричної схеми пристрою

Для моделювання принципової електричної схеми прототипу пульсометра було використано середовище Circuit Diagram IDE. Це онлайн-інструмент для створення простих електричних схем, який дозволяє візуально розмістити компоненти та з'єднати їх згідно з логікою роботи пристрою. Завдяки цьому стало можливим графічно представити основні функціональні елементи системи та перевірити логіку їх взаємозв'язку.

На схемі було спроектовано підключення основних компонентів пристрою, що зображено на рисунку 2.11.

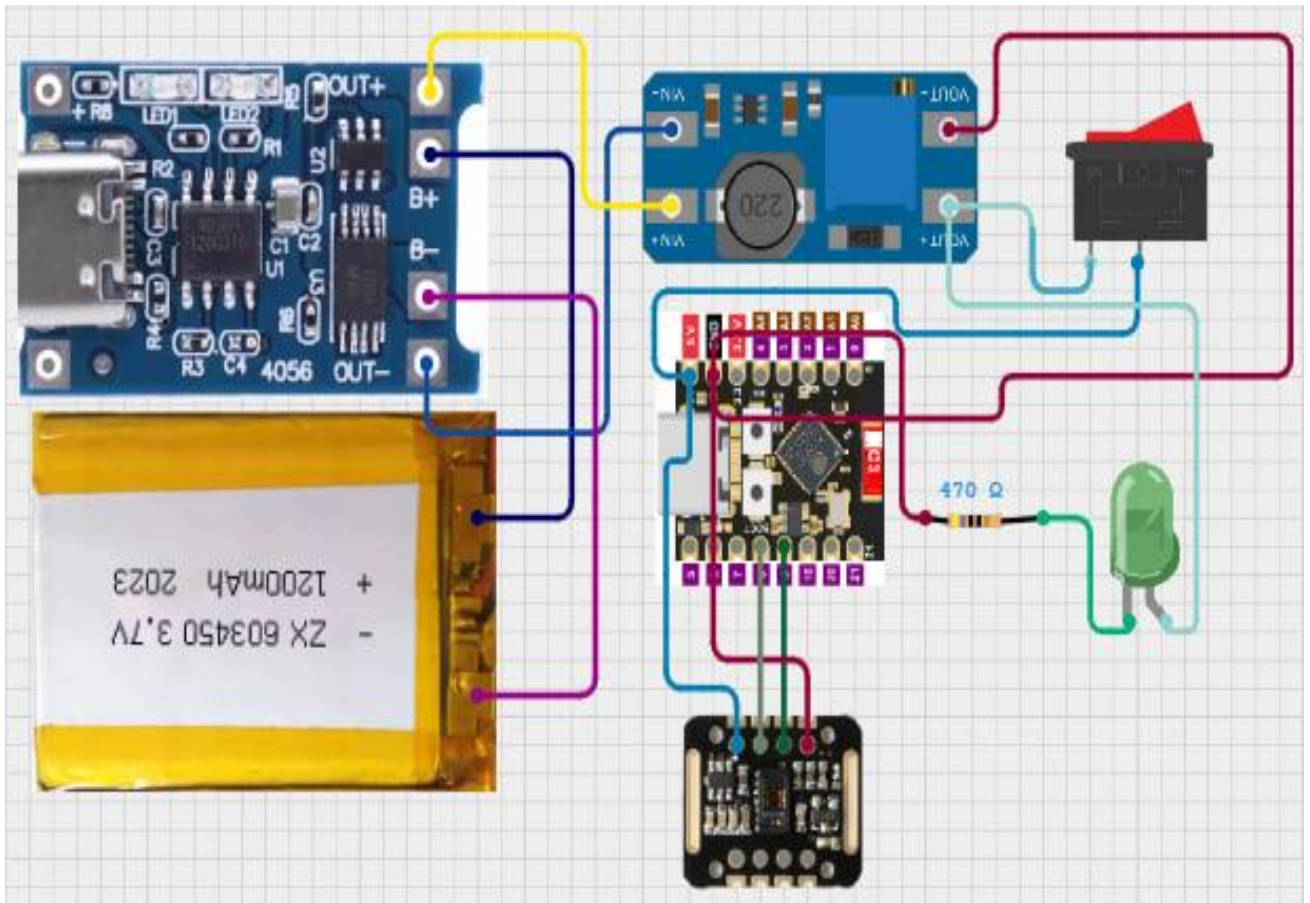


Рисунок 2.11 – Принципова електрична схема пристрою

Опис схеми підключення:

1) мікроконтролер ESP32-C3 – пін 5B– підключено до виходу підвищувального перетворювача (VOUT+) з напругою 5В через тумблер, таким чином він розриває живлення ESP32-C3; GND – до загальної землі схеми. Також для індикації подачі живлення та розряду акумулятора встановлений двокольоровий світлодіод (зелений/червоний);

2) датчик пульсу MAX30102:

- VIN – підключено до піну 3V3 ESP32;
- GND – до GND ESP32;
- SDA – до GPIO8;
- SCL – до GPIO9.

3) акумулятор підключений до B- та B+ (TP4056) відповідно, а OUT- та OUT+ (TP4056) до VIN- та VIN+ (MT3608) відповідно.

Описане та зображене на схемі підключення компонентів, забезпечує повну функціональність прототипу пульсометра. Усі компоненти з'єднані таким чином, щоб гарантувати стабільне живлення мікроконтролера ESP32-S3 та датчика MAX30102, а також надати можливість вмикати й вимикати пристрій за допомогою апаратної кнопки. Подача живлення здійснюється від літій-полімерного акумулятора через модуль заряду TP4056 і підвищувальний перетворювач, що формує постійну напругу 5 В для ESP32. Одночасно із запуском живлення активується індикатор у вигляді зеленого світлодіода, який сигналізує про роботу системи. Обмін даними між мікроконтролером і сенсором пульсу здійснюється по шині I²C, що дозволяє точно й безперервно зчитувати значення ЧСС.

Загалом, реалізована схема є компактною, енергоефективною та функціонально завершеною – вона виконує всі необхідні завдання для реєстрації й передачі пульсу. Це підтверджує правильність обраної архітектури пристрою та демонструє практичну придатність рішення для створення переносного пульсометра.

2.3 Реалізація програми для апаратної частини пристрою

2.3.1 Алгоритм роботи пристрою

Програмна логіка пристрою пульсометра побудована на поділенні його функціонування на кілька послідовних етапів, які реалізуються після подачі живлення. Основна задача програми – забезпечити надійне зчитування сигналу з оптичного датчика пульсу, обчислення значення ЧСС, контроль наявності фізичного контакту з пальцем, та передача результату у вигляді цифрових даних по BLE у смартфон, наглядно алгоритм роботи пристрою зображено графічно на рисунку 2.12.

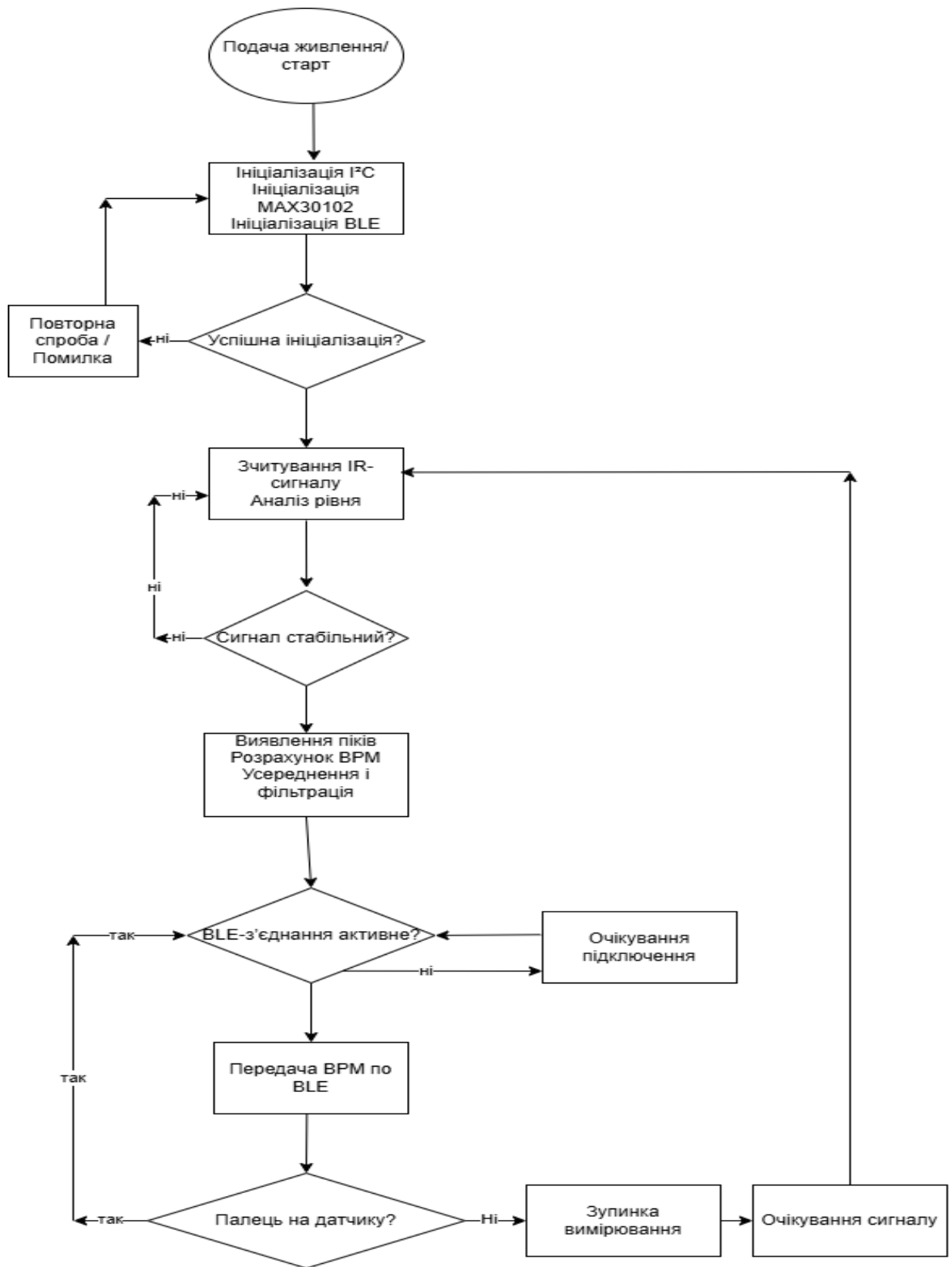


Рисунок 2.12 – Граф-схема алгоритму роботи пристрою

Після ввімкнення пристрою запускається стандартна процедура ініціалізації, у межах якої активуються апаратні інтерфейси, зокрема I²C для

взаємодії з датчиком MAX30102, та BLE-інтерфейс для передачі даних. Якщо ініціалізація будь-якого з модулів завершується помилкою, система не переходить до основного циклу, а виконує повторну спробу або виводить код помилки (в базовій реалізації – через серійну консоль).

У разі успішної ініціалізації пристрій переходить у стан очікування стабільного сигналу від пальця користувача. На цьому етапі відбувається безперервне зчитування значень інфрачервоного сигналу з фотодіода MAX30102. Якщо інтенсивність сигналу не досягає встановленого порогу, пристрій продовжує чекати. Як тільки сигнал стабілізується, система переходить у режим обчислення ЧСС. Вимірювання ЧСС виконується шляхом обробки потоку IR-даних, які аналізуються через вікно ковзного середнього, виявлення піків та інтервалів між ними. Обчислене значення пульсу (оновлюється один раз на 2 секунди з урахуванням усереднення та фільтрації шумів).

Далі значення пульсу передається по BLE у відповідну характеристику користувацького сервісу. Якщо з'єднання зі смартфоном активне, передача виконується негайно. У протилежному випадку пристрій зберігає останнє значення локально (в оперативній пам'яті) та періодично перевіряє наявність нового з'єднання. Крім того, реалізований механізм обробки втрати сигналу (наприклад, якщо користувач прибрав палець з датчика). У цьому випадку передача зупиняється, значення пульсу скидається, і пристрій повертається у стан очікування стабільного сигналу.

2.3.2 Програмне середовище розробки

Для розробки програмного забезпечення мікроконтролера ESP32-C3 SuperMini було використано Arduino IDE. Це середовище є одним із найпоширеніших у галузі прототипування та розробки вбудованих систем [9] завдяки простому інтерфейсу, відкритій архітектурі та великій кількості сумісних бібліотек. Arduino IDE підтримує ESP32 через офіційний пакет плат

від Espressif, що дозволяє швидко розгорнути проєкт без складного налаштування середовища.

Проєкт реалізовано мовою C++, яка є стандартом для середовища Arduino. Ця мова забезпечує прямий доступ до апаратного рівня, дозволяє точно контролювати часові затримки, регістри, периферійні інтерфейси, і при цьому підтримує об'єктно-орієнтовані конструкції, зручні для модульної розробки.

У програмному коді були задіяні такі зовнішні бібліотеки:

- Wire.h – стандартна бібліотека для роботи з інтерфейсом I²C; забезпечує обмін даними між ESP32 і датчиком MAX30102;

- MAX30105.h – бібліотека від SparkFun для роботи з оптичними сенсорами MAX30102/30105; надає функції ініціалізації, налаштування режимів роботи, зчитування IR-сигналу;

- heartRate.h – допоміжна бібліотека для розрахунку ЧСС на основі аналізу піків сигналу; включає базові алгоритми виявлення пульсових хвиль;

- BLEDevice.h, BLEUtils.h, BLEServer.h, BLE2902.h – набір бібліотек для реалізації BLE-сервера на ESP32. Дозволяють створити користувацький сервіс, оголосити характеристики, здійснювати передачу даних та керувати підключенням клієнтів (наприклад, мобільного додатку).

Arduino IDE було обрано з огляду на такі переваги:

- простота встановлення та конфігурації;
- вбудована підтримка ESP32 через пакет "esp32 by Espressif Systems";
- велика спільнота та документація;
- швидка інтеграція з бібліотеками для BLE та сенсорів;
- підтримка завантаження прошивки через USB без додаткових налаштувань.

У проєкті використовувалась наступна конфігурація плати: Тип плати: ESP32C3 Dev Module; Частота процесора: 160 МГц; Flash Frequency: 80 МГц; Upload Speed: 115200 бод; Порт: USB (через USB-UART міст); Формат прошивки: .bin; Режим завантаження: стандартний (натискання Boot + Reset);

Проект компілювався без додаткових оптимізацій, у стандартному режимі для платформ ESP32.

2.3.3 Особливості реалізації алгоритму роботи пристрою

Після подачі живлення першим етапом роботи програми є ініціалізація всіх апаратних компонентів пристрою. Це забезпечує правильну конфігурацію внутрішніх інтерфейсів мікроконтролера ESP32-C3 та перевірку доступності підключених зовнішніх пристроїв.

Комунікація між ESP32-C3 та сенсором пульсу MAX30102 здійснюється за протоколом I²C. У програмі використовується стандартна бібліотека Wire.h, яка дозволяє визначити номери виводів SDA та SCL та встановити тактову частоту шини: `Wire.begin(8, 9); Wire.setClock(400000);`

Це забезпечує стабільний високошвидкісний обмін даними із сенсором.

Для взаємодії з оптичним сенсором використовується бібліотека MAX30105.h, яка сумісна з MAX30102. Ініціалізація сенсора виконується за допомогою функції `particleSensor.begin(Wire)`. Якщо пристрій не відповідає, програма виводить повідомлення про помилку в серійну консоль, реалізацію наведено на лістингу 2.1.

Лістинг 2.1 – Ініціалізація сенсора MAX30102

```
if (!particleSensor.begin(Wire)) {  
    Serial.println("Не знайдено сенсор  
MAX30102.Перезапуск...");  
    while (1); }  
particleSensor.setup();  
particleSensor.setPulseAmplitudeRed(0x0A);  
particleSensor.setPulseAmplitudeIR(0x0A);
```

Для створення BLE-сервера використовується набір бібліотек від Espressif: BLEDevice, BLEServer, BLEUtils та BLE2902. Під час ініціалізації задається ім'я пристрою, створюється сервер, сервіс та характеристика для передачі значення ЧСС, що наведено на лістингу 2.2.

Лістинг 2.2 – Ініціалізація Bluetooth Low Energy (BLE)

```
BLEDevice::init("StrongHeart");
BLEServer *pServer = BLEDevice::createServer();
BLEService *pService = pServer->createService(SERVICE_UUID);
BLECharacteristic *pCharacteristic = pService-
>createCharacteristic(
    CHARACTERISTIC_UUID,
    BLECharacteristic::PROPERTY_NOTIFY);
pCharacteristic->addDescriptor(new BLE2902());
pService->start();
BLEAdvertising *pAdvertising = BLEDevice::getAdvertising();
pAdvertising->start();
```

Ця конфігурація дозволяє мобільному додатку приєднатися до пристрою та отримувати сповіщення з оновленим значенням пульсу в режимі реального часу. Після завершення ініціалізації всі ключові компоненти перевіряються на наявність помилок. Якщо хоча б один з етапів не пройдено – пристрій зупиняється до усунення помилки. Успішна ініціалізація переводить систему в основний цикл очікування стабільного сигналу.

2.3.4 Зчитування, обробка та передача даних

Після успішної ініціалізації пристрій переходить до основного циклу роботи, який виконується безперервно у реальному часі. Основною метою цього циклу є зчитування оптичного сигналу з сенсора MAX30102, обробка отриманих даних для визначення ЧСС, та передача результатів на зовнішній пристрій через BLE.

Сенсор MAX30102 формує сигнал на основі інфрачервоного випромінювання, яке частково поглинається кров'ю під час кожного серцевого скорочення. Цей сигнал зчитується у вигляді цифрових значень з ADC. Для цього використовується метод `particleSensor.getIR()`, який повертає значення поточного вимірювання інтенсивності інфрачервоного світла, що наведено на лістингу 2.3.

Лістинг 2.3 – Зчитування та обробка сигналу, визначення ЧСС

```
long irValue = particleSensor.getIR();
if (checkForBeat(irValue)) {
long delta = millis() - lastBeat;
    lastBeat = millis();
    bpm = 60000 / delta;
}
```

Зчитування проводиться з певною частотою (наприклад, 100 Гц), залежно від конфігурації сенсора. У разі низької амплітуди сигналу (відсутній палець або нестабільний контакт) обробка ЧСС не виконується. Для обробки сигналу використовується базовий алгоритм виявлення піків, наданий у бібліотеці heartRate.h. Алгоритм знаходить моменти часу між пульсовими хвилями (інтервали R-R) та на їх основі розраховує миттєве значення ЧСС у ударах на хвилину (BPM). Розраховане значення проходить усереднення для зменшення шумів, а також накладаються обмеження допустимого діапазону (наприклад, 40–180 BPM), що дозволяє відкинути хибні сплески.

Після обчислення ЧСС його значення передається по BLE у відповідну характеристику. Для цього використовується метод setValue() та notify(): pCharacteristic->setValue(bpm); pCharacteristic->notify(); Передача відбувається кожні 2 секунди або при зміні значення. Якщо з'єднання з клієнтом відсутнє, пристрій продовжує обробку сигналу, але не виконує надсилання, зменшуючи тим самим енергоспоживання.

У випадку, коли палець прибирається з сенсора, амплітуда IR-сигналу різко падає. Це фіксується програмою як втрата контакту, після чого:

- вимірювання ЧСС припиняється;
- передається значення 0 або припиняється передача;
- система переходить у режим очікування стабільного сигналу.

Цей механізм, наведено на лістингу 2.4, дозволяє уникнути генерації помилкових показників та підтримує коректність відображення даних на стороні клієнта.

Лістинг 2.4 – Обробка втрати сигналу

```
const int irThreshold = 5000;
bool fingerDetected = false;
long irValue = particleSensor.getIR();
if (irValue > irThreshold) {
  if (!fingerDetected) {
    Serial.println("Палець виявлено");
    fingerDetected = true;
  }
  if (checkForBeat(irValue)) {
    long delta = millis() - lastBeat;
    lastBeat = millis();
    bpm = 60000 / delta;
    if (bpm > 40 && bpm < 180) {
      pCharacteristic->setValue(bpm);
      pCharacteristic->notify();
    }
  } else {
    if (fingerDetected) {
      Serial.println("Палець втрачено");
      fingerDetected = false;
    }
    bpm = 0;
    pCharacteristic->setValue(bpm);
    pCharacteristic->notify();
  }
}
```

Таким чином, цикл зчитування, обробки та передачі даних забезпечує повний функціонал пристрою у реальному часі, з автоматичною реакцією на зміни стану контакту з пальцем та підтримкою енергоефективної передачі лише у разі активного з'єднання з клієнтом.

2.3.5 Оптимізація та енергозбереження

З урахуванням того, що пристрій працює від автономного джерела живлення – літій-полімерного акумулятора, у процесі розробки програмного забезпечення було приділено увагу зменшенню енергоспоживання та загальній оптимізації роботи. Це дозволяє збільшити тривалість безперервного використання пристрою без необхідності частого підзаряджання.

Одним із базових кроків оптимізації стало обмеження частоти передачі даних по BLE. Значення частоти серцевих скорочень оновлюється на стороні клієнта не постійно, а з інтервалом у 2 секунди. Такий підхід дозволяє знизити навантаження на модуль Bluetooth, зменшити кількість бездротових передач і тим самим зекономити енергію без шкоди для точності

відображення. Також реалізовано автоматичний контроль наявності пальця на датчику. Якщо інфрачервоний сигнал нижчий за встановлений поріг, пристрій припиняє обробку даних і передачу ЧСС, переходячи у стан очікування. Це значно зменшує обчислювальне навантаження на мікроконтролер у моменти бездіяльності, що прямо впливає на зниження енергоспоживання.

Для максимально можливого зниження енергоспоживання приладу було впроваджено режим “light sleep” у період між вимірюваннями. Режим “light sleep” дозволяє суттєво зменшити енергоспоживання пристрою без втрати працездатності. У цьому режимі центральний процесор ESP32 переходить у сплячий стан, проте периферія, включаючи I²C та BLE, залишається активною або готовою до швидкого пробудження. Під час “light sleep” передача даних не припиняється, з’єднання BLE залишається стабільним, а мікроконтролер пробуджується кожні 100 мілісекунд для зчитування сигналу. Таким чином, у період між вимірюваннями значно знижується загальне енергоспоживання – до 1–2 мА у порівнянні з активним режимом (~80–90 мА), що значно продовжує автономну роботу пристрою без шкоди для точності або швидкості оновлення даних.

2.4 Проектування та розробка корпусу пульсометра

Оскільки розроблюваний пристрій призначений для носіння на тілі користувача, важливою частиною є створення зручного, компактного й ергономічного корпусу. Корпус не лише забезпечує захист електронних компонентів від механічних пошкоджень, пилу та вологи, але й відіграє ключову роль у зручності використання пульсометра у повсякденних умовах або під час фізичної активності.

Усі компоненти пристрою – мікроконтролер ESP32-C3 SuperMini, сенсор МАХ30102, плата живлення (TP4056 з boost-конвертером), акумулятор та індикатор живлення – повинні бути раціонально розміщені

всередині корпусу з урахуванням їх габаритів, точок підключення та напрямку орієнтації.

Проектування корпусу здійснюється в середовищі SolidWorks, яке дозволяє точно моделювати усі габарити та кріплення з урахуванням допусків, міжцентрових відстаней та функціональних отворів.

Під час створення 3D-моделі корпусу враховано:

- необхідний простір для кожного з модулів;
- відстані між платами для уникнення коротких замикань;
- отвори для виходу світлодіода, доступу до кнопки та роз'єму для зарядки акумулятора;
- отвір для виходу контактної частини сенсора.

У поточному варіанті корпусу передбачено два основних функціональних сегменти (рис. 2.13): нижня основа, в яку вмонтовується акумулятор та електроніка, та верхня частина, в яку щільно вмонтовується сенсор.



Рисунок 2.13 – Детальна модель корпусу приладу

Бокова кришка для доступу до всієї електроніки, з отворами для зарядки, кнопки та світлодіода. Для під'єднання кабелю заряджання передбачено отвір під роз'єм Type-C.

Розміри корпусу в первинному проєкті становлять:

- довжина: 55 мм;
- ширина: 35 мм;
- висота: 20 мм.

Моделювання завершено з урахуванням допусків на усадку пластика при друку, а також технологічних обмежень 3D-принтера.

Для виготовлення фізичного прототипу використовується 3D-принтер Creality Ender-3, зображено на рисунку 2.14.

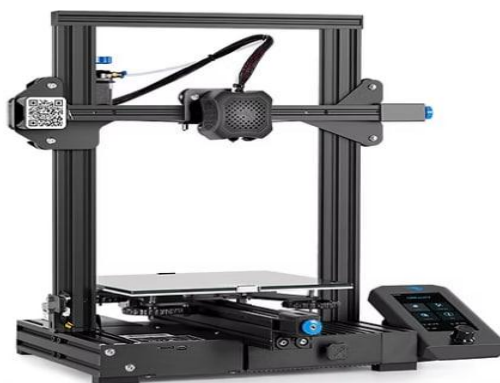


Рисунок 2.14 – 3D-принтер Creality Ender-3

Принтер дозволяє друкувати з високою точністю при товщині шару 0.2 мм. Матеріалом друку є ABS-пластик, який забезпечує достатню механічну міцність, термостійкість та стійкість до вологи, що особливо актуально для натільних пристроїв.

Файл моделі експортується з SolidWorks у формат .STL, що є стандартним для адитивного виробництва, після чого обробляється у слайсері, у даному випадку було використано власний слайсер від Creality, призначений для генерації G-коду з урахуванням технічних характеристик принтера та заданими параметрами. Налаштування моделі у слайсері перед друком зображено на рисунку 2.15.

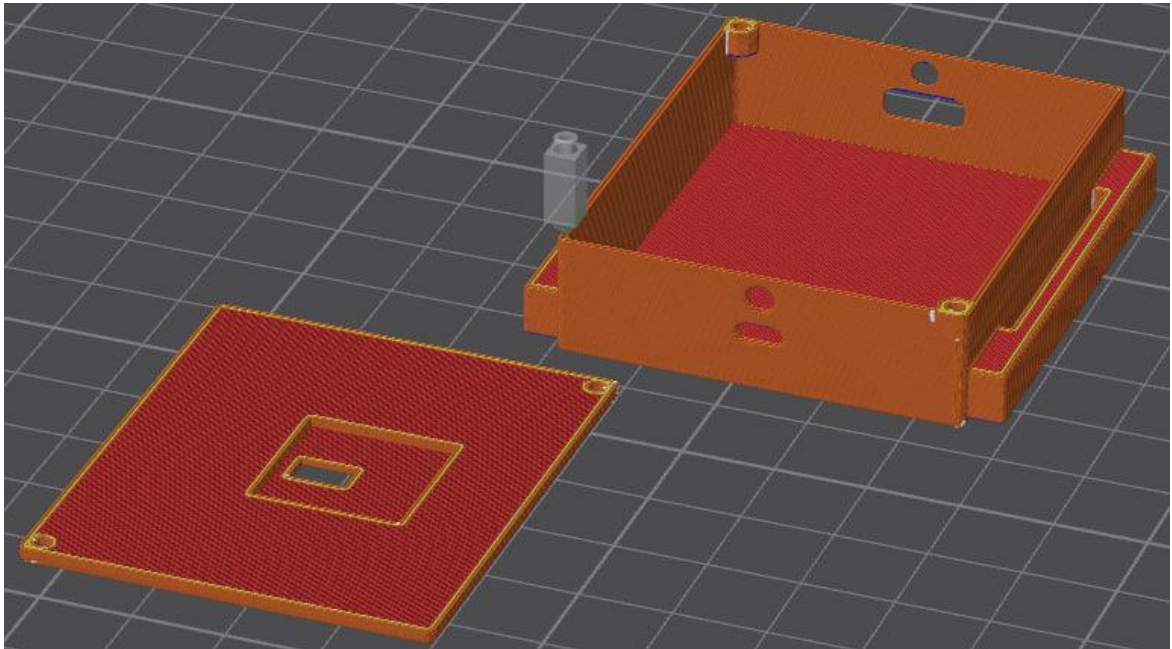


Рисунок 2.15 –3D-модель корпусу у слайсері Creality

У процесі налаштування друку в слайсері були встановлені параметри, що забезпечують оптимальне поєднання якості та надійності виготовлення корпусу. Товщина шару становила 0.2 мм, що дозволяє досягнути балансу між деталізацією та швидкістю друку. Ширина екструзії була встановлена на рівні 0.4 мм – стандартне значення для сопла діаметром 0.4 мм. Ступінь заповнення обрано на рівні 30 %, як компроміс між міцністю конструкції та економією матеріалу. Температура екструдера під час друку складала 240 °С, що відповідає рекомендованим параметрам для ABS-пластику, а температура нагрівального стола – 100 °С, що забезпечує належне зчеплення моделі з поверхнею та зменшує ризик викривлення під час охолодження. Швидкість друку було встановлено на рівні 50 мм/с. Охолодження моделі під час друку було вимкнено, оскільки ABS-пластик чутливий до перепадів температур, що може призвести до деформацій. Роздрукована модель корпусу приладу зображена на рисунку 2.16.

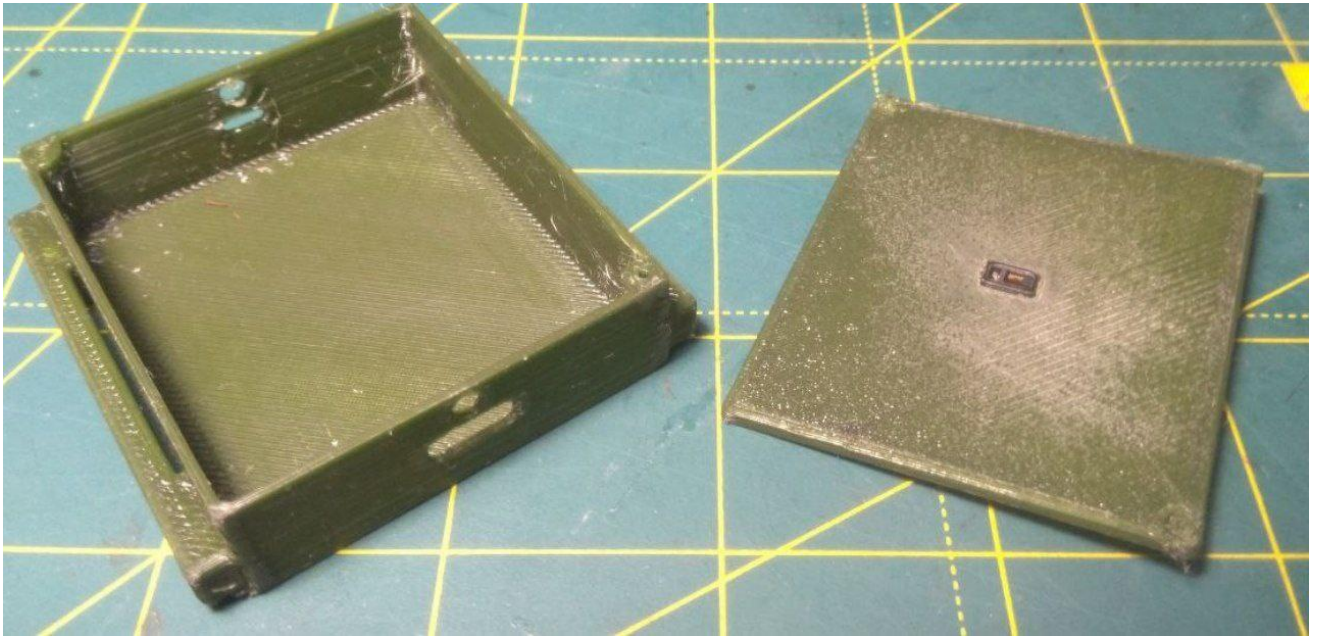


Рисунок 2.16 – Надрукований корпус

Зазначені налаштування дали змогу отримати прототип із достатньою точністю і якістю поверхні для подальшого складання і тестування електронної частини пристрою.

2.5 Складання та монтаж апаратної частини пристрою

Після завершення етапів проектування схеми, корпусу та реалізації прошивки розпочинається процес безпосереднього складання апаратної частини пульсометра, тобто спаювання компонентів.

Основна мета цього етапу – фізично інтегрувати усі компоненти пристрою в єдину працездатну систему, забезпечивши надійність електричних з'єднань і правильну геометрію монтажу відповідно до розробленої 3D-моделі. Перед підключенням живлення на ESP32-C3, налагоджуємо регульовальним гвинтом потрібний вольтаж, та вольтметром перевіряємо вольтаж на виході з DC-DCперетворювача (рис. 2.17), так як на пін VINESP32-C3, маємо подати строго 5В (похибка вольтметра ± 0.01 В).

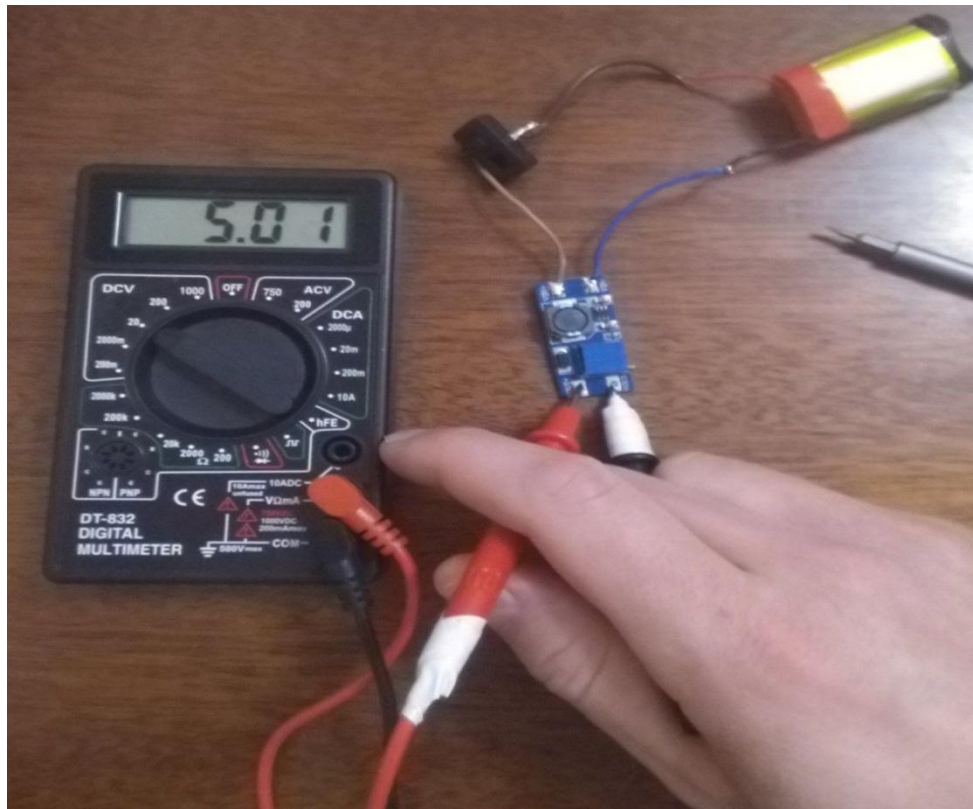


Рисунок 2.17 – Перевірка вольтажу на виході з DC-DCперетворювача

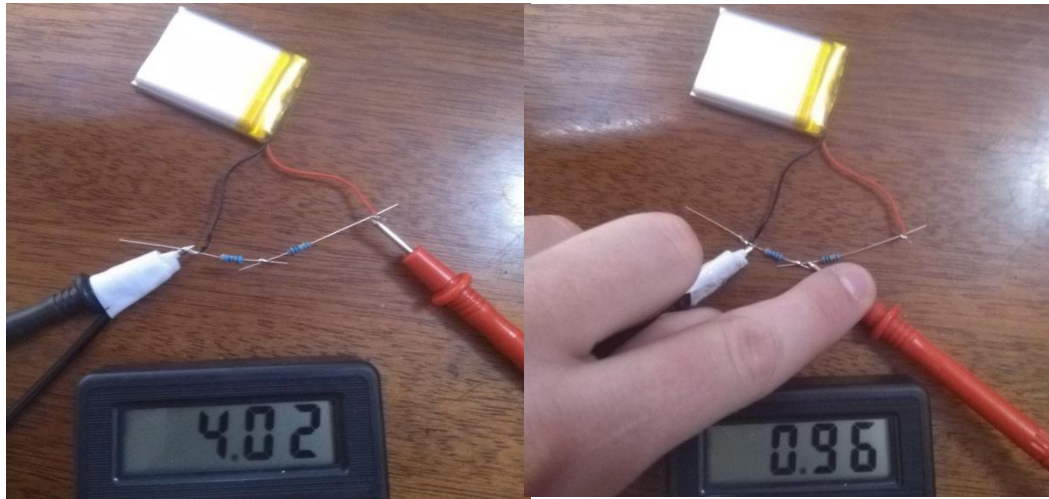
Ключові паяні з'єднання:

- шина I²C (SDA, SCL) між ESP32-C3 та MAX30102;
- живлення 3.3 В та GND до сенсоровід ESP32;
- вихід TP4056 до акумулятора та входу DC-DC модуля;
- підключення кнопки живлення та світлодіода.

Для реалізації індикації заряду акумулятора як у застосунку, так і апаратно за допомогою світлодіода, було зібрано дільник напруги на резисторах 100 кОм та 33 кОм. Дільник підключено до аналогового входу ESP32-C3, що дозволяє контролеру вимірювати напругу на акумуляторі. Розрахунок напруги на виході дільника здійснено за формулою:

$$V_{out} = V_{in} \cdot R_2 / (R_1 + R_2).$$

Згідно з цим розрахунком, при максимальній напрузі 4.2В на акумуляторі, на виході дільника формується приблизно 1.04В (рис. 2.18 (а), рис. 2.18 (б)), що є безпечним для аналогового піна мікроконтролера, який дозволяє подати напругу до 1.1В.



(a)

(б)

Рисунок 2.18 – Заміри напруги на виході акумулятора (а) та на ділянці напруги (б)

Значення напруги використовуються як для графічного відображення у мобільному застосунку, так і для керування станом світлодіода, що сигналізує про рівень заряду.

Після завершення пайки, всі компоненти акуратно фіксуються всередині надрукованого корпусу (рис. 2.19).



Рисунок 2.19 – Фіксація компонентів пристрою у корпусі

Для цього використовуються передбачені місця для компонентів, гвинтів та термоклей. Сенсор виводиться в отвір на верхній частині корпусу таким чином, щоб його контактна поверхня трохи виступала за стінку корпусу.

Особлива увага приділяється перевірці полярності підключень, механічній стійкості з'єднань та захисту плати від можливих коротких замикань, що здійснюється за допомогою обгортання кожного компоненту термоусадкою.

2.6 Відлагодження проєкта в ArduinoIDE

Після складання апаратної частини було проведено первинне функціональне тестування для перевірки працездатності основних модулів перед подальшою інтеграцією з мобільним застосунком. Метою тестування було переконатися у стабільному зчитуванні сигналу з сенсора MAX30102, коректності роботи алгоритму розрахунку ЧСС, а також виявленні наявності/відсутності пальця на сенсорі.

Для цього прилад було підключено до комп'ютера через USB-інтерфейс, а в середовищі Arduino IDE відкрито монітор порту. Програма, прошита у мікроконтролер, виводила такі дані у реальному часі: обчислене значення BPM, а також службові повідомлення на кшталт "Finger detected", "Collectiondata", "70 BPM" (рис. 2.20).

На фрагменті зчитування в серійному моніторі видно, що прилад при ввімкненні постійно перевіряє, чи є палець на сенсорі, якщо сенсор щільно прилягає до шкіри – починається збір інформації про пульс у буфер для стабілізації значень. Буфер вміщує в себе 5 значень пульсу, які збираються протягом перших 10 секунд після того, як шкіра торкнулась сенсора.

Якщо до цього моменту пройшло все успішно, конект шкіри із сенсором не розірвано та усі 5 значень в буфері валідні – починається виведення та передача даних поточного ЧСС по BLEу реальному часі.

```
14:42:38.453 -> [INFO] Waiting for finger...
14:42:38.593 -> [INFO] Finger detected. Measuring...
14:42:38.686 -> [INFO] Collecting data...
14:42:38.686 -> [SLEEP] Entering light sleep for 100ms...
14:42:38.686 -> [SLEEP] Woke up from light sleep
14:42:40.596 -> [INFO] Collecting data...
14:42:40.689 -> [SLEEP] Entering light sleep for 100ms...
14:42:40.689 -> [SLEEP] Woke up from light sleep
14:42:42.599 -> [INFO] Collecting data...
14:42:42.692 -> [SLEEP] Entering light sleep for 100ms...
14:42:42.692 -> [SLEEP] Woke up from light sleep
14:42:44.602 -> [INFO] Collecting data...
14:42:44.696 -> [SLEEP] Entering light sleep for 100ms...
14:42:44.743 -> [SLEEP] Woke up from light sleep
14:42:46.616 -> [INFO] Collecting data...
14:42:46.709 -> [SLEEP] Entering light sleep for 100ms...
14:42:46.709 -> [SLEEP] Woke up from light sleep
14:42:48.615 -> [INFO] Collecting data...
14:42:48.755 -> [SLEEP] Entering light sleep for 100ms...
14:42:48.755 -> [SLEEP] Woke up from light sleep
14:42:50.658 -> [INFO] Collecting data...
14:42:50.752 -> [SLEEP] Entering light sleep for 100ms...
14:42:50.752 -> [SLEEP] Woke up from light sleep
14:42:52.659 -> [DATA] BPM: 65
14:42:52.751 -> [SLEEP] Entering light sleep for 100ms...
14:42:52.751 -> [SLEEP] Woke up from light sleep
14:42:54.659 -> [DATA] BPM: 67
14:42:54.751 -> [SLEEP] Entering light sleep for 100ms...
14:42:54.751 -> [SLEEP] Woke up from light sleep
14:42:56.661 -> [DATA] BPM: 69
14:42:56.757 -> [SLEEP] Entering light sleep for 100ms...
14:42:56.757 -> [SLEEP] Woke up from light sleep
14:42:58.665 -> [DATA] BPM: 71
14:42:58.758 -> [SLEEP] Entering light sleep for 100ms...
```

Рисунок 2.20 –Значення з монітору порта

Так як дані з сенсора зчитуються раз на 2 секунди, то ми маємо можливість знизити енергоспоживання нашого пристрою за допомогою режиму “lightsleep”.

Цей режим не є найефективнішим у сенсі зниження енергоспоживання, найефективніший – “deep sleep”, але при цьому режимі повністю вимикається периферія нашого SoC приладу, зокрема BLE, який у нашому випадку має працювати безперервно для надійного та стабільного з’єднання із програмним застосунком на смартфоні. Тому найкращій вихід з нашого становища – впровадження “lightsleep” режиму, що вимикає усі непотрібні

для нашої задачі прилади, при цьому залишає периферію, зокрема BLE у робочому стані, не порушуючи вимоги безперервної передачі даних. Завдяки увімкненню всіх компонентів приладу усього на 100мс кожні 2000мс, вдалося знизити енергоспоживання з 80мА/год до 4мА/год.

Таким чином, функціональне тестування підтвердило коректність роботи всіх ключових компонентів пристрою – від стабільного зчитування ІЧ-сигналу та точного визначення ЧСС до безперервної передачі результатів по BLE у режимі реального часу. Впровадження енергозберігаючого режиму “light sleep” дозволило суттєво знизити середнє енергоспоживання без порушення вимог до точності вимірювань і надійності зв’язку. Це забезпечує практичну придатність пристрою до тривалого автономного використання та відповідає сучасним вимогам до носимих електронних рішень.

3 РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ КЕРУВАННЯ ПУЛЬСОМЕТРОМ

Сучасні носимі пристрої для моніторингу фізіологічних параметрів вимагають не лише точного апаратного забезпечення, але й ефективного програмного інтерфейсу для взаємодії з користувачем. У межах цього проєкту було реалізовано мобільний застосунок для оперативної системи Android, призначений для приймання даних з пульсометра по каналу BLE, відображення поточного значення ЧСС у реальному часі, запису тренувальних сесій, аналізу результатів та перегляду історії вимірювань.

3.1 Вибір стеку технологій для реалізації застосунку

Для розробки мобільного клієнта, який взаємодіє з апаратною частиною пульсометра через інтерфейс BLE, було обрано стек технологій, що забезпечує гнучкість, кросплатформеність та зручність розробки. Основним середовищем виступає React Native CLI [5] – кросплатформена фреймворк-система, що базується на JavaScript та дозволяє створювати нативні застосунки для Android з єдиною кодовою базою.

На відміну від Expo, використання React Native CLI дає змогу мати повний контроль над нативними модулями, що особливо важливо для інтеграції з BLE-бібліотеками, роботою з низькорівневим Bluetooth API та налаштуванням власних нативних компонентів. Такий підхід дає змогу забезпечити стабільну роботу застосунку з нестандартними BLE-пристроями.

Для локального зберігання даних, таких як історія вимірювань, середній пульс за сесію, часові мітки тощо, використовується база даних SQLite. Завдяки використанню бібліотек типу react-native-sqlite-storage, забезпечено надійне збереження та швидкий доступ до даних без потреби в зовнішньому з'єднанні з сервером.

Тестування застосунку здійснювалось у режимі реального часу, без використання емулятора – шляхом підключення фізичного Android-пристрою через USB до робочого середовища розробки. Це дозволило коректно перевірити роботу Bluetooth-модуля, який в більшості емуляторів Android не підтримується, а також одразу діагностувати поведінку застосунку в реальних умовах використання.

Вибір описаного стеку технологій обумовлений такими факторами:

- повна підтримка BLE через нативні модулі;
- кросплатформеність (із фокусом на Android);
- гнучка структура проекту та можливість оптимізації під специфіку роботи з медичними даними;
- доступність великої кількості плагінів та документації.

Таким чином, обрана технологічна основа забезпечує повний контроль над логікою застосунку, ефективну роботу з BLE-пристроєм, зручне збереження локальних даних і подальше масштабування функціоналу у разі потреби.

3.2 Архітектура мобільного застосунку та логіка роботи

Мобільний застосунок, що розробляється в межах цього проекту, є ключовим елементом взаємодії користувача з апаратною частиною пульсометра. Його архітектура базується на чітко структурованих модулях, що забезпечують обробку, збереження та візуалізацію даних, отриманих з пристрою через BLE. Архітектура застосунку побудована з урахуванням принципів розділення відповідальностей. Основні функціональні блоки включають:

- BLE-модуль: відповідає за сканування пристроїв, підключення до пульсометра, отримання та обробку даних у реальному часі;
- модуль обробки даних: забезпечує первинну фільтрацію значень, агрегацію для обчислення середнього пульсу та зон ЧСС;

- модуль збереження даних: реалізований на основі SQLite, дозволяє створювати, зберігати та переглядати історію сесій вимірювання;
- модуль UI: складається з екранів з'єднання, запису сесії, перегляду результатів та навігації між ними.

Кожна сесія вимірювання ініціюється користувачем вручну, після чого застосунок починає приймати значення ЧСС у реальному часі, відображаючи їх у вигляді числового значення та номеру поточної пульсової зони, логіку роботи наглядно наведено на рисунку 3.1.

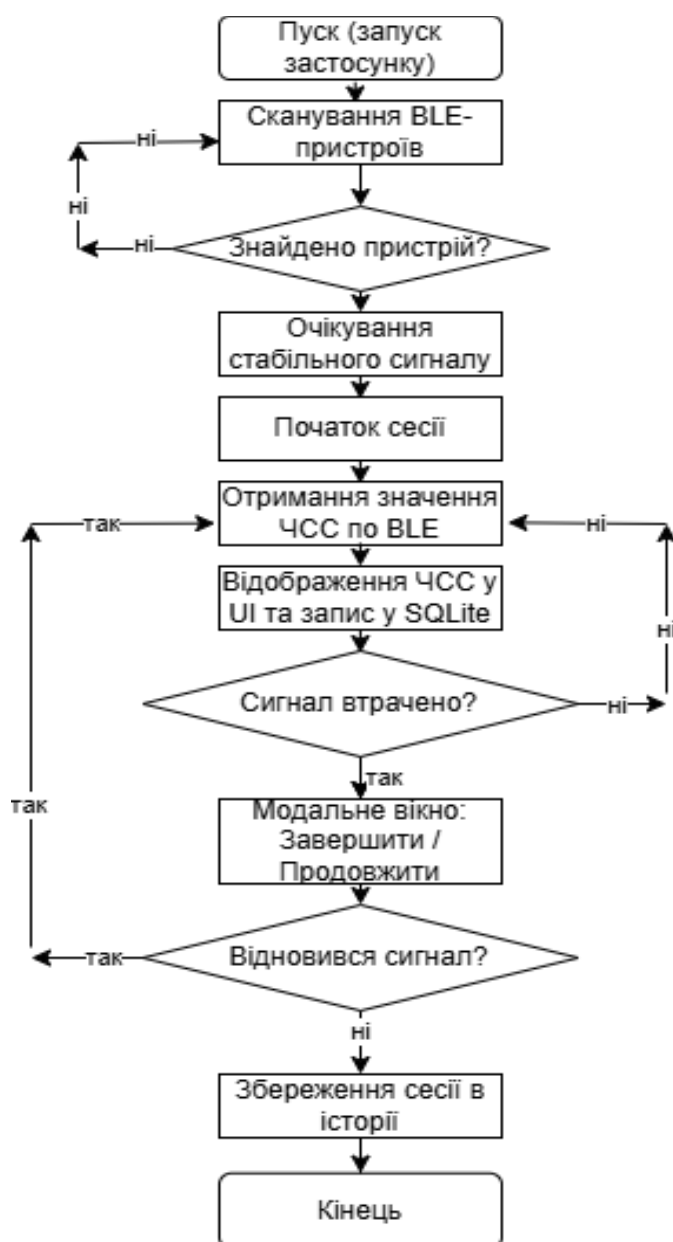


Рисунок 3.1 – Граф схема алгоритму роботи програмного застосунку

Дані зберігаються локально в базі SQLite з позначенням часу початку/завершення сесії. У разі втрати зв'язку з пристроєм, застосунок призупиняє запис і виводить модальне повідомлення з пропозицією зберегти або продовжити сесію після відновлення сигналу, та запобігає втрати даних.

Окрім логіки роботи, доцільно також представити структурну модель архітектури застосунку, яка ілюструє взаємодію ключових компонентів системи. На рисунку 3.2 наведено узагальнену структурну схему застосунку.

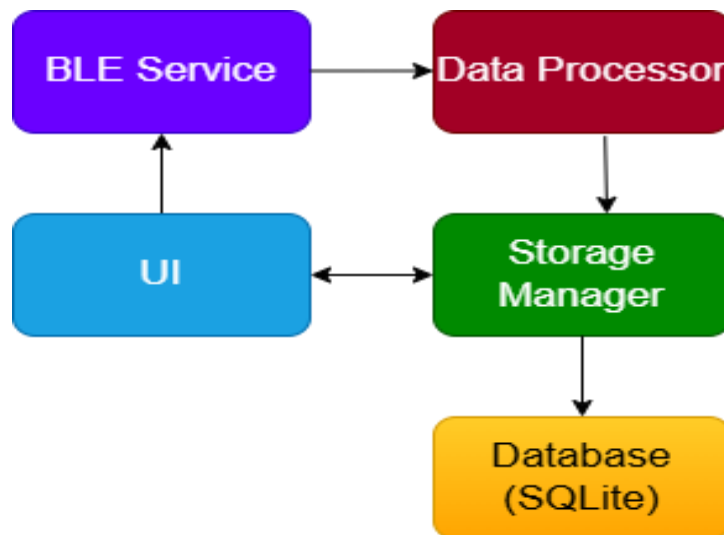


Рисунок 3.2 – Структурна схема модулів мобільного застосунку

У наведеній схемі кожен модуль виконує окрему функціональну роль:

- модуль UI (User Interface) – відповідає за взаємодію з користувачем, відображення стану пристрою, ЧСС, історії вимірювань;
- модуль BLE Service – забезпечує з'єднання з пульсометром, отримання даних у реальному часі;
- модуль Data Processor – виконує обробку значень ЧСС, визначає пульсові зони, формує проміжні статистичні значення;
- модуль Database (SQLite) – реалізує зберігання сесій локально на пристрої;
- модуль Storage Manager – відповідає за збереження, видалення, експорт і перейменування сесій.

Також у межах архітектури застосунку важливо описати навігаційну структуру між екранами, яка реалізується засобами React Navigation. На рисунку 3.3 представлено схему переходів між екранами у застосунку.

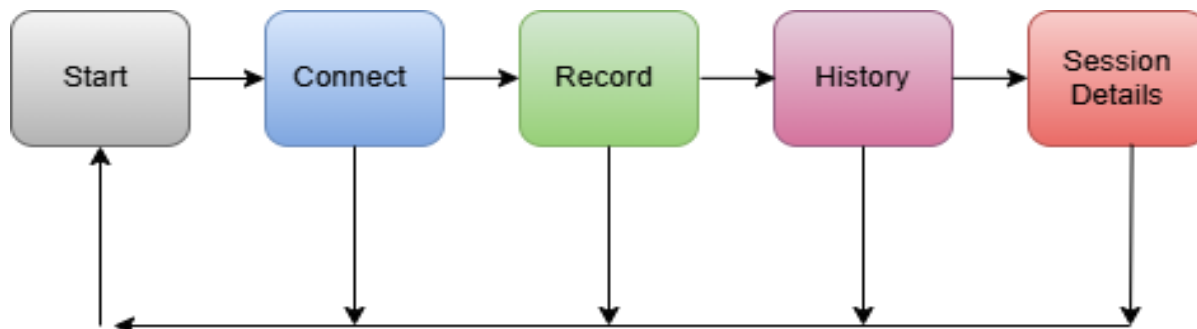


Рисунок 3.3 – Схема навігації меж екранами мобільного застосунку

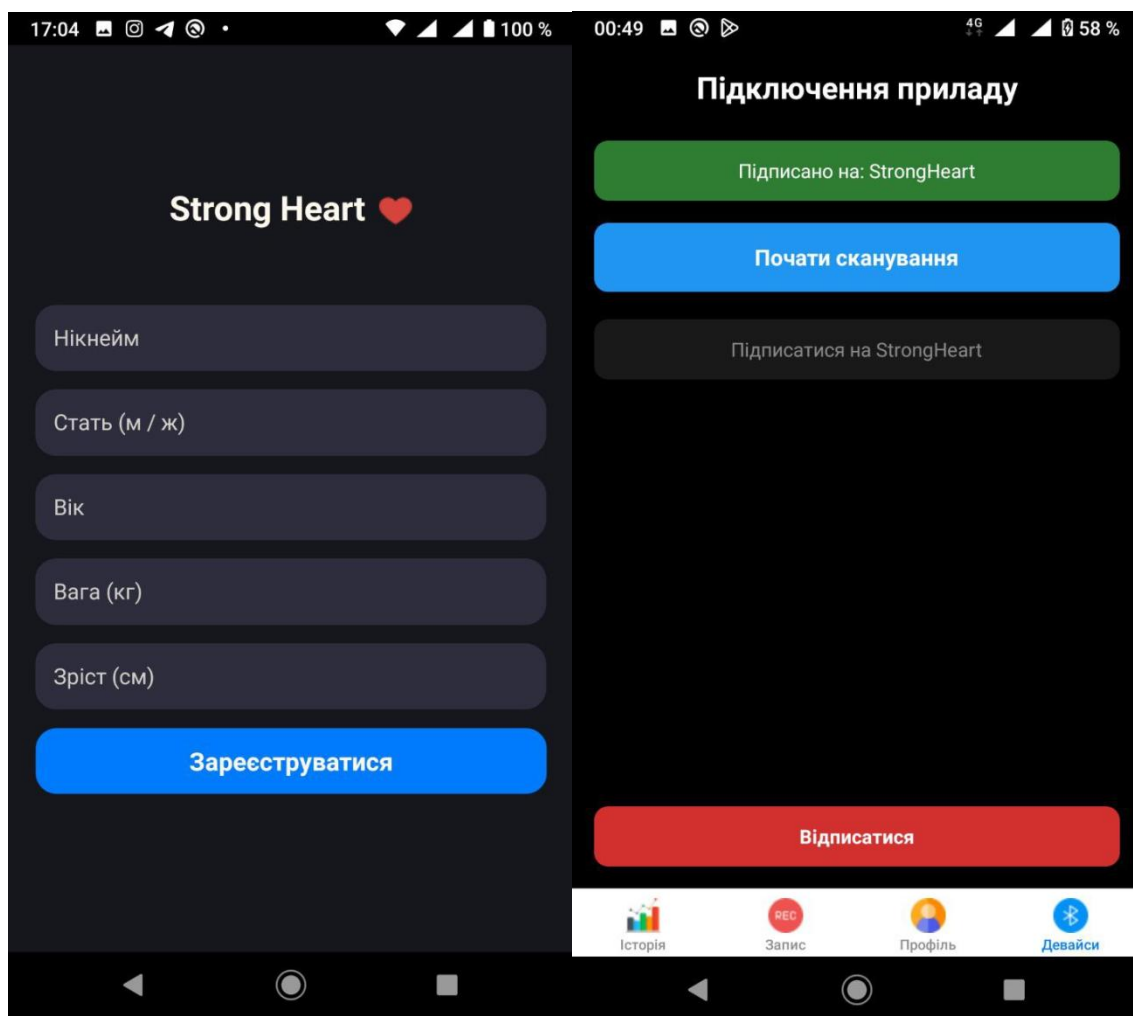
Обидві схеми дозволяють краще зрозуміти внутрішню організацію застосунку, послідовність взаємодії між модулями та логіку навігації в процесі використання.

3.3 Проектування інтерфейсу користувача

Важливою складовою розробки будь-якого мобільного застосунку є створення інтуїтивно зрозумілого, зручного та візуально узгодженого інтерфейсу користувача. Саме зовнішній вигляд і логіка взаємодії значною мірою визначають загальне враження від використання програмного продукту. У контексті пульсометра, що позиціонується як засіб для щоденного моніторингу фізіологічних показників, особливо важливо забезпечити просту навігацію, чітку візуалізацію даних та мінімалістичний стиль.

Для створення дизайну інтерфейсу було обрано інструмент Figma – сучасне веборієнтоване середовище для прототипування та дизайну. Завдяки цьому середовищу було розроблено повний набір макетів для всіх основних екранів мобільного застосунку.

На рисунку 3.4 (а) зображено екран реєстрації користувача, та екран з'єднання із BLE-пристроєм на рисунку 3.4 (б).



(а)

(б)

Рисунок 3.4 – Екран реєстрації (а) та з'єднання із BLE-пристроєм (б)

Під час реєстрації, користувач має обов'язково заповнити всі поля інформацією. За допомогою віку, потім буде обчислено максимальній пульс, та приблизні пульсові зони, тому реєстрація є невід'ємною частиною користування застосунком.

Екран з'єднання із BLE-пристроєм виконує роль стартової точки взаємодії користувача з приладом, де він бачить статус підключення до пульсометра. У центрі розташована кнопка "Підключитись", яка ініціює процес пошуку пристрою через Bluetooth. Інтерфейс реалізований у

мінімалістичному стилі з акцентом на простоту взаємодії. Колірна гама нейтральна, щоб не відволікати користувача від основного функціоналу.

На рисунку 3.5 (а) наведено дизайн екрану запису сесії вимірювання ЧСС, а на рисунку 3.5 (б) – екран історії записаних сесій.

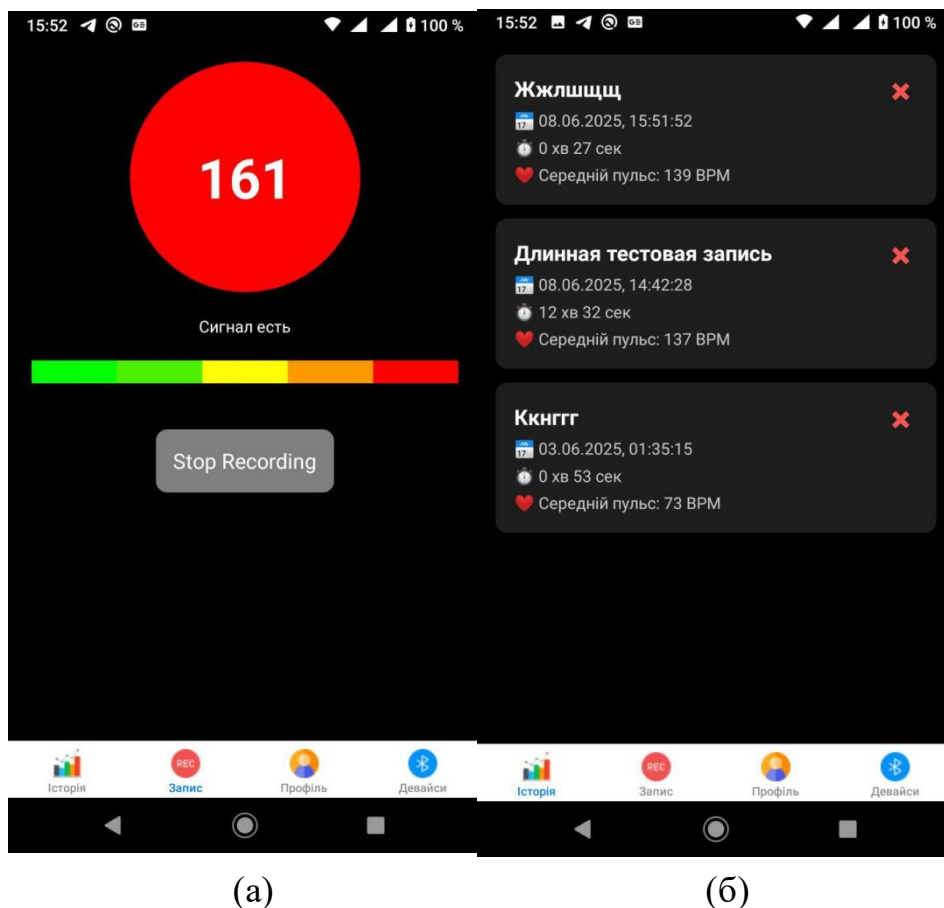


Рисунок 3.5 – Екран запису сесії (а) та екран історії сесій (б)

На екрані запису сесій, користувач має можливість натиснувши на кнопку “Почати запис”, розпочати запис сесії вимірювання ЧСС, якщо він підключений до BLE-пристрою. Як тільки запис розпочнеться – автоматично, у реальному часі, користувач буде бачити поточну ЧСС, у червоному кругу, а також поточну пульсову зону, яка буде змінюватись відповідно до змін ЧСС та позначатися маркером на різнокольоровій прямій в реальному часі, де кожен колір – своя пульсова зона:

- сірий – 1 пульсова зона;
- зелений – 2 пульсова зона;

Користувач має змогу подивитись особисту інформацію свого профілю: ім'я, стать, вік, вагу, зріст, а також індивідуально розраховані, враховуючи власну інформацію, максимальну ЧСС та пульсові зони. При виникненні потреби відредагувати особисту інформацію, користувач натискає на кнопку “Редагувати”, та йому буде доступний інтерфейс редагування, де можна змінити всю інформацію.

На рисунку 3.7 наведено дизайн екрану розширеної інформації про сесію запису ЧСС.



Рисунок 3.7 – Екран розширеної інформації про сесію

При натисканні на плашку сесії з екрану історії, у користувача відкривається екран розширеної інформації про сесію, на якому можна подивитись запис ЧСС у вигляді графіку, на якому можна побачити піки та спади пульсу. Також кількість часу проведеного у конкретній пульсовій зоні під час запису сесії, та інша загальна інформація. Це все дозволяє користувачу всебічно оцінити свій пульс на обраному проміжку часу.

Таким чином, інтерфейс мобільного застосунку спроектовано відповідно до сучасних принципів UX/UI-дизайну, з акцентом на простоту, функціональність та зручність використання. Кожен екран має чітко визначену мету та мінімалістичне візуальне оформлення, що дозволяє користувачу швидко орієнтуватись у програмі навіть без попереднього досвіду. Завдяки використанню Figma вдалося ефективно реалізувати адаптивний та візуально узгоджений інтерфейс, що забезпечує цілісне користувацьке враження від роботи з пульсометром.

3.4 База даних для мобільного застосунку

Для забезпечення локального збереження результатів вимірювання пульсу та інформації про користувача в мобільному застосунку реалізовано базу даних на основі вбудованої СУБД SQLite [6], яка є оптимальним рішенням для мобільних платформ завдяки своїй легкості, стабільності та підтримці SQL-запитів.

База даних має реляційну структуру і містить три основні таблиці:

- `user_profile` – зберігає персональні дані користувача;
- `sessions` – містить інформацію про окремі сесії вимірювання;
- `session_points` – зберігає деталізовані значення пульсу з часовими мітками, пов'язані з конкретною сесією через зовнішній ключ `session_id`.

Для наочного представлення логіки бази даних нижче наведено діаграму зв'язків між таблицями (рис. 3.8), яка відображає структуру зберігання інформації та типи зв'язків.

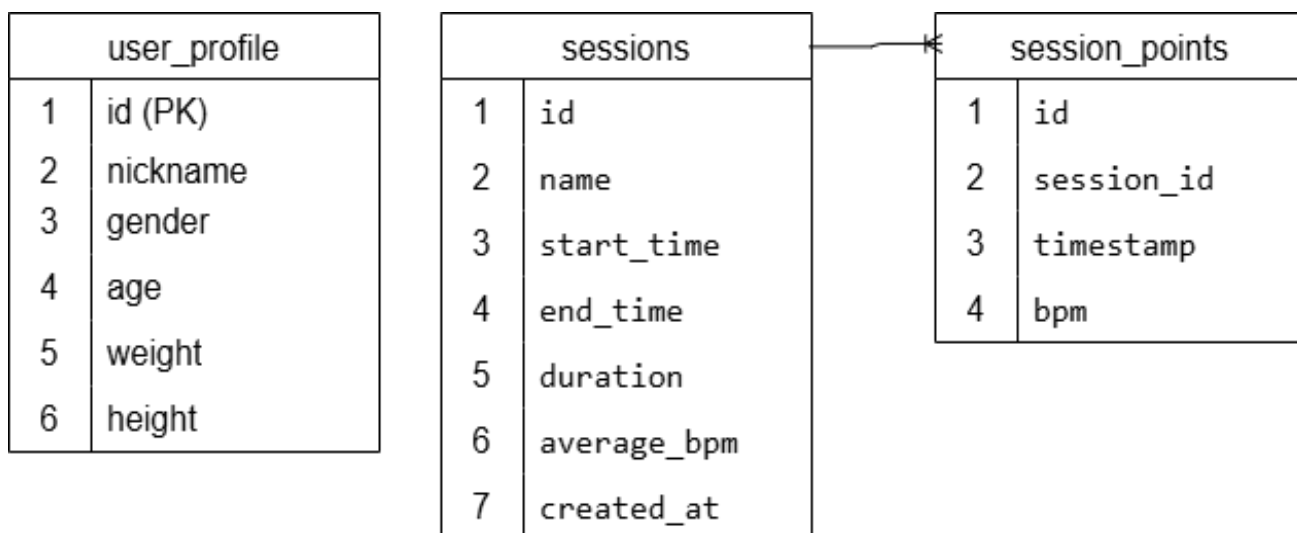


Рисунок 3.8 – ER-діаграма структури бази даних застосунку

Зв'язок між таблицями реалізований відповідно до принципів цілісності: кожен запис у таблиці session_points має посилання на сесію з таблиці sessions, а у випадку видалення сесії – відповідні дані видаляються автоматично завдяки директиві ON DELETE CASCADE, відношення між таблицями: session до session_points як один до багатьох. Так як передбачено, що смартфоном зазвичай користується одна людина – через це немає практичного сенсу реалізовувати прив'язку профілю користувача до конкретного приладу, тому це локальний блок інформації взагалі без зв'язків. Створення таблиць наведено у лістингу А.2.

3.5 Реалізація функції реєстрації профілю користувача

На початковому етапі використання мобільного застосунку користувачу пропонується ввести основні персональні дані, які необхідні для подальшої індивідуалізації процесу вимірювання та аналізу пульсу.

До таких даних належать: нікнейм, стать, вік, вага та зріст. Ці параметри можуть бути використані для подальших розрахунків, зокрема для визначення максимального пульсу, що визначається на формулою:

$$\text{ЧСС}_{\text{max}} = 220 - \text{вік}$$

Маючи вже значення максимального можливого пульсу людини можемо визначити приблизні персональні пульсові зони користувача, що визначаються як:

1) зона 1 (легка активність):

$$\text{ЧСС1} = \text{ЧСС}_{\text{max}} \times 0.50 \text{ до } \text{ЧСС}_{\text{max}} \times 0.60;$$

2) зона 2 (аеробна зона):

$$\text{ЧСС2} = \text{ЧСС}_{\text{max}} \times 0.60 \text{ до } \text{ЧСС}_{\text{max}} \times 0.70;$$

3) зона 3 (інтенсивна аеробна):

$$\text{ЧСС3} = \text{ЧСС}_{\text{max}} \times 0.70 \text{ до } \text{ЧСС}_{\text{max}} \times 0.80;$$

4) зона 4 (анаеробна):

$$\text{ЧСС4} = \text{ЧСС}_{\text{max}} \times 0.80 \text{ до } \text{ЧСС}_{\text{max}} \times 0.90;$$

5) зона 5 (максимальне навантаження):

$$\text{ЧСС5} = \text{ЧСС}_{\text{max}} \times 0.90 \text{ до } \text{ЧСС}_{\text{max}} \times 1.00.$$

Введена користувачем інформація при реєстрації профілю зберігається у таблиці `user_profile` локальної бази даних SQLite. Структура таблиці передбачає унікальний ідентифікатор профілю та відповідні текстові або числові поля для кожного з параметрів. Оскільки застосунок орієнтований на індивідуальне використання, передбачається зберігання лише одного профілю користувача.

Логіка взаємодії реалізована таким чином, що при першому запуску застосунку перевіряється наявність запису у таблиці `user_profile`. У разі його відсутності користувач перенаправляється на екран реєстрації. Валідація введених даних здійснюється на рівні інтерфейсу: усі поля є обов'язковими до заповнення, для числових значень використовується обмеження формату (наприклад, тільки цифри для віку, ваги та зросту). Після успішного збереження даних, застосунок переходить до основного функціоналу.

Такий підхід забезпечує зручну та зрозумілу взаємодію з користувачем, а також дозволяє підтримувати цілісність персоніфікованих даних, необхідних для аналізу фізіологічних показників. Повний код реалізації даного компоненту програми наведено у лістингу А.3.

3.6 Сканування та підключення до BLE-пристрою

Одним із ключових етапів взаємодії мобільного застосунку з апаратною частиною пульсометра є встановлення BLE з'єднання. Взаємодія відбувається за допомогою бібліотеки `react-native-ble-manager`, яка забезпечує повноцінну підтримку функцій сканування, підключення та обміну даними відповідно до специфікацій BLE. Алгоритм сканування реалізовано таким чином, що при переході на відповідний екран застосунок ініціює пошук пристроїв поблизу. Сканування триває певний фіксований час або завершується достроково при виявленні пристрою з назвою, що відповідає ідентифікатору пульсометра – “StrongHeart”. Після виявлення цільового пристрою здійснюється спроба встановлення з'єднання та підписки на характеристику передавання даних про ЧСС, відповідно до стандарту GATT-сервісів (Heart Rate Service). У разі успішного підключення користувачу візуально відображається стан з'єднання, після чого він може переходити до вимірювання. У випадку втрати сигналу або відсутності пристрою в радіусі дії, застосунок інформує користувача про помилку з'єднання та пропонує повторити сканування. З метою зручності та уникнення дублювання коду, логіку сканування, підключення та відписки від пристрою інкапсульовано у сервісний модуль `BleManagerService`, що забезпечує модульність архітектури та спрощує подальше обслуговування або масштабування функціоналу BLE.

Таким чином, реалізація процесу з'єднання з пульсометром забезпечує стабільну передачу даних у реальному часі, контроль стану з'єднання та належну реакцію застосунку на зміну умов підключення. Повний код реалізації даного компоненту програми наведено у лістингу А.4.

3.7 Отримання та обробка даних ЧСС

Після встановлення з'єднання з BLE-пристроєм застосунок переходить до активного отримання даних частоти серцевих скорочень (ЧСС) у

реальному часі. Передача даних здійснюється за допомогою GATT-характеристики з типом Notify, що дозволяє пристрою надсилати оновлення без необхідності запитів з боку клієнта. Типовий інтервал надсилання даних – кожні 2 секунди.

На стороні мобільного застосунку обробка вхідних значень відбувається в кілька етапів:

- валідація отриманого значення: перевірка на нульові або аномальні значення, які можуть свідчити про втрату контакту з пальцем;

- фільтрація: при потребі застосовується згладжування або відкидання одиничних шумових імпульсів;

- агрегація: усереднення даних для подальшої статистики, а також підрахунок поточної пульсової зони користувача відповідно до його віку, визначеного у профілі;

- візуалізація: числове значення ЧСС та відповідна зона відображаються на екрані запису у режимі реального часу. Для візуального сприйняття також реалізовано динамічний індикатор кольору або анімацію;

- технічна обробка: паралельно кожне значення зберігається у масиві, що згодом буде записаний у локальну базу даних разом з часовими мітками.

Завдяки такій структурі обробки застосунок забезпечує не лише достовірне відображення поточного стану, а й накопичення даних для подальшого аналізу. Повний код реалізації даного компоненту програми наведено у лістингу А.5.

3.8 Запис і збереження сесій в локальну базу даних

Після початку вимірювання ЧСС застосунок ініціює створення нової сесії, яка фіксується в таблиці sessions локальної бази даних. На момент запуску зберігається час початку, а при завершенні – час завершення, тривалість та середнє значення ЧСС.

Під час сесії кожне отримане значення ЧСС разом з міткою часу тимчасово зберігається в оперативній пам'яті. Після завершення запису вони групово додаються до таблиці `session_points`, що пов'язана з сесією через зовнішній ключ `session_id`.

Загальна логіка реалізації включає: ініціацію нової сесії користувачем; накопичення даних протягом вимірювання; контроль втрати зв'язку з пристроєм, при якому призупиняється запис; завершення сесії вручну або після тривалого розриву з'єднання; запис усіх даних до бази: метадані – у `sessions`, точки ЧСС – у `session_points`.

Після завершення сесії вона стає доступною для перегляду в історії, а також для аналізу та візуалізації статистики. Повний код реалізації даного компоненту програми наведено у лістингу А.6.

3.9 Відображення графіків, статистики та історії

Для забезпечення зручного аналізу попередніх вимірювань мобільний застосунок реалізує окремий розділ перегляду історії сесій. Усі завершені сесії, збережені в базі даних, виводяться у вигляді списку з коротким описом: назва, дата, тривалість та середній пульс.

Користувач може перейти до детального перегляду будь-якої сесії, де представлено: лінійний графік ЧСС, побудований на основі даних з таблиці `session_points`; середнє значення пульсу за сесію; загальну тривалість та час вимірювання; діаграму розподілу пульсових зон, яка візуалізує, скільки часу користувач перебував у кожній із зон (від зони 1 до зони 5).

Візуалізація реалізована з використанням зовнішньої бібліотеки побудови графіків [11], що підтримує інтерактивні елементи. Це дозволяє користувачеві аналізувати тренди та пікові значення пульсу в межах кожної сесії. Загальна логіка формування інтерфейсу історії забезпечує зручну навігацію між сесіями, а також наочне представлення отриманих даних. Повний код реалізації даного компоненту програми наведено у лістингу А.7.

4 ТЕСТУВАННЯ ТА ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ КОМПЛЕКСУ

Після завершення етапів проектування та реалізації апаратної і програмної частин пульсометра було проведено тестування з метою перевірки його функціональності, точності вимірювань та коректної взаємодії між компонентами системи. Даний розділ описує результати перевірки працездатності комплексу, зокрема, тестування BLE-з'єднання між пристроєм та мобільним застосунком, збереження даних у локальній базі даних, відображення пульсу в реальному часі, а також стабільність запису ЧСС під час тренувань і в стані спокою.

Під час першого запуску застосунку користувач бачить екран реєстрації (рис. 4.1), на якому він має точно заповнити усі поля, що зображено на екрані.

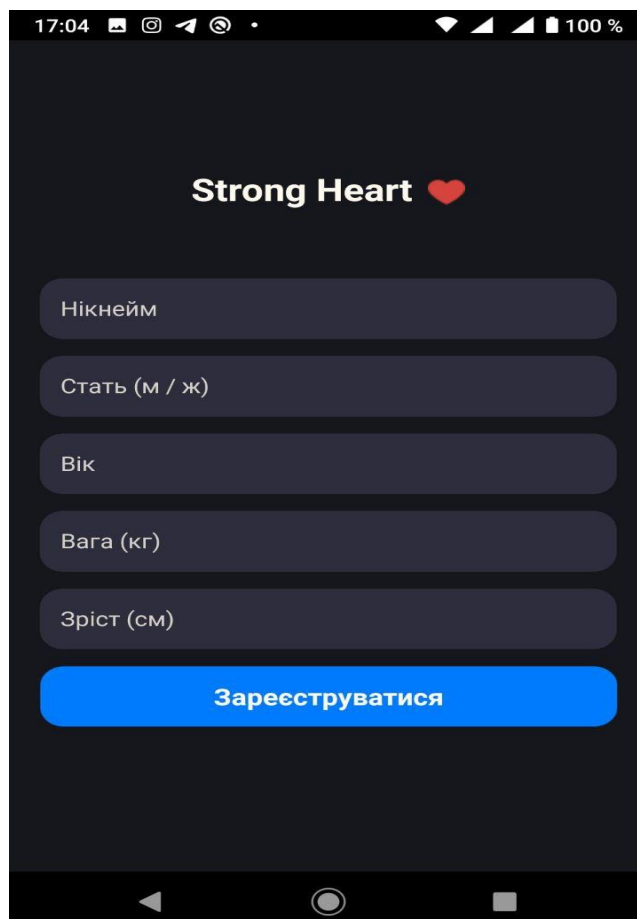


Рисунок 4.1 – Екран реєстрації профілю користувача

Якщо буде введено некоректні для конкретного поля дані, програма буде сигналізувати про це червоною рамкою навколо некоректно введених даних.

Після натискання на кнопку “Зареєструватися” – дані про користувача автоматично будуть занесені до локальної бази даних, для подальшого використання у вимірюванні діапазонів пульсових зон та макисмальної частоти серцевих скорочень. Якщо всі поля введені коректно та реєстрацію успішно завершено – користувач побачить на екрані смартфона головний екран застосунку (рис. 4.2), на якому зображено усі ті особисті дані, що було введено користувачем на етапі реєстрації, а також пораховані індивідуальні діапазони пульсових зон та максимальну ЧСС.

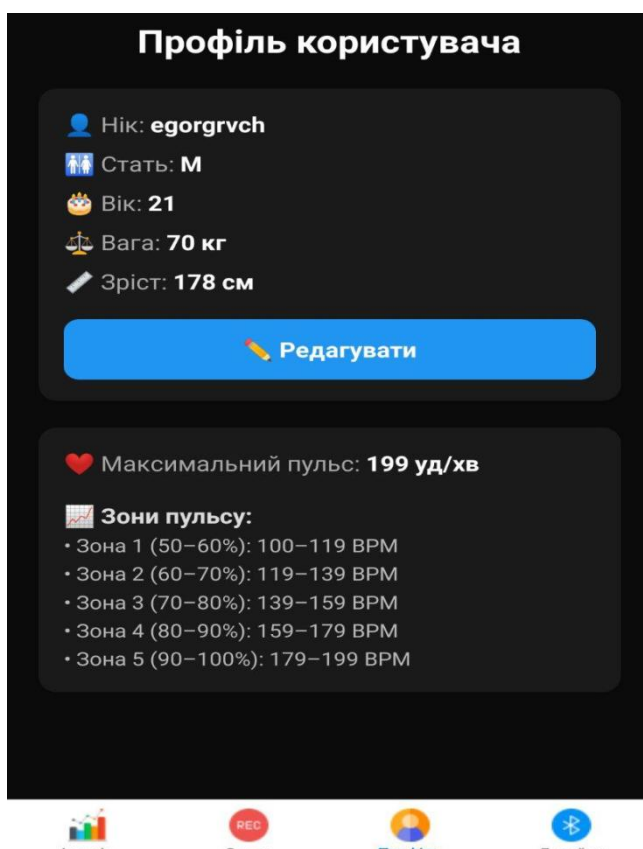


Рисунок 4.2 – Екран профілю користувача

Також користувач, за потреби, може редагувати усі сової особисті дані. Для цього потрібно натиснути на кнопку “Редагувати”, та ввести у відповідних полях нову потрібну інформацію, після чого, якщо дані

впливають на змінення максимальної ЧСС, або пульсових зон – їх буде перераховано автоматично відповідно новим венесеним даним.

Для використання пульсометру його акумулятор має буди зарядженим. Щоб зарядити акумулятор приладу – користувач має підключити його за допомогою Туре-Скабелю до джерела живлення 5В, 2А (рис. 4.3 (а), рис. 4.3 (б)).



(а)

(б)

Рисунок 4.3 – Сигналізація світлодіодом процесу заряджання акумулятора (а) та стану повного заряду 100% (б)

Якщо акумулятор на момент підключення не заряджено на 100%, та відбувається процес його заряджання, користувачу сигналізує про це червоний колір світлодіода, що розташований біля Туре-С кабеля, а коли процес заряджання завершено, та акумулятор має 4.2В на своїх виходах – світлодіод світиться синім коліром. Користувач може не боятися переразряду акумулятора завдяки модулю зарядки акумуляторів TP4056, через це не потрібно постійно перевіряти, чи не горить світлодіод вже синім щоб скоріше від’єднати прилад від джерела живлення.

Після того як пульсометр було заряджено, можна починати користуватися його функціями. За допомогою ремінця, користувач може зручно зафіксувати прилад на своїй руці сенсором до шкіри. Перевівши

перемикач у положення “On” – загоряється зелений сигнальний світлодіод, він індикуює користувачу, що пульсометр успішно запущено, та можна переходити до з’єднання його з мобільним застосунком. Якщо цей сигнальний світлодіод блимає червоним кольором – акумулятор пульсометру розряджений, потрібно поставити прилад на зарядку. Фіксація приладу на руці та робота світлодіодної індикації зображена на рисунках 4.4 (а) та 4.4 (б).



Рисунок 4.4 – Коректна робота приладу (а) та індикація розрядженого акумулятора (б)

Після того як акумулятор заряджен та пристрій ввімкнено, користувач переходить на екран “Девайси”, та бачить інтерфейс за допомогою якого можна здійснити пошук BLE-пристроїв поблизу смартфона, та під’єднатись до них. Користувач має натиснути кнопку “Почати сканування”, після чого нижче цієї кнопки в нього з’явиться список знайдених BLE-пристроїв. Пульсометр завжди буде позначатися в цьому списку як девайс з іменем “StrongHeart”. Коли сканування завершено та наш пристрій знайдено, можемо натискати на його назву, після чого буде виконано з’єднання з пульсометром за допомогою BLE, коли підключення буде встановлено, користувач побачить текст на зеленому фоні ”Підписано на: StrongHeart”, відключитися від пристрою можна натиснувши “Відписатися”(рис. 4.5).

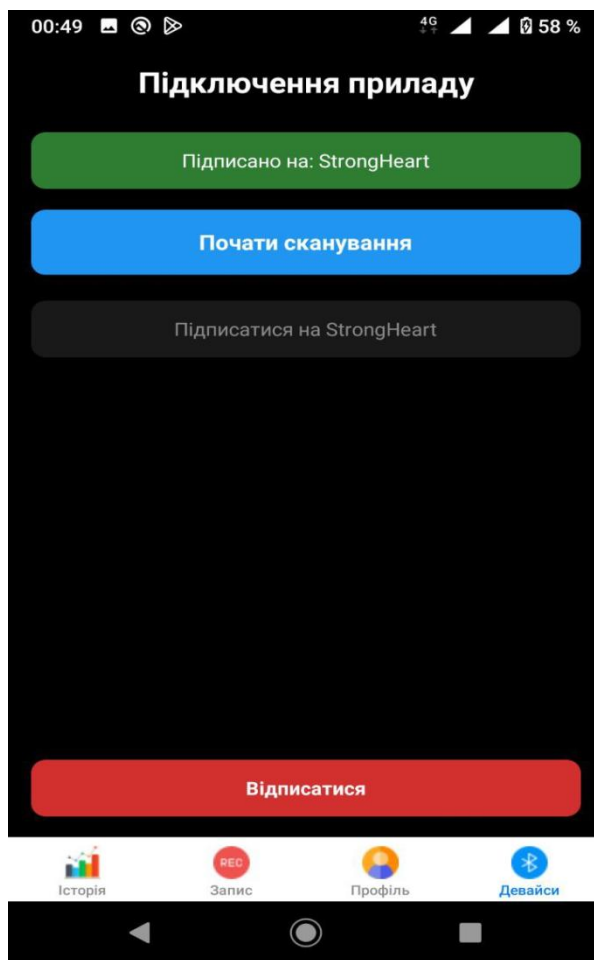


Рисунок 4.5 – Екран підключення до пульсометра

Після того як встановлен зв'язок між приладом та програмним застосунком на смартфоні, користувач може переходити до запису сесії вимірювання ЧСС. Переходимо на екран з назвою “Запис”, на якому натискаємо на кнопку “Почати запис сесії”, та бачимо модальне вікно (рис. 4.6).

У модальному вікні користувач має обов'язково ввести назву сесії, та натиснути кнопку “ОК”, якщо готов розпочати, або “Відміна”, якщо передумав починати запис сесії, ця кнопка закриє модальне вікно. Якщо програмний застосунок не має підключення до пульсометра, та не отримує дані – кнопка “ОК” буде недоступна, та користувачу потрібно буде перевірити підключення до приладу.

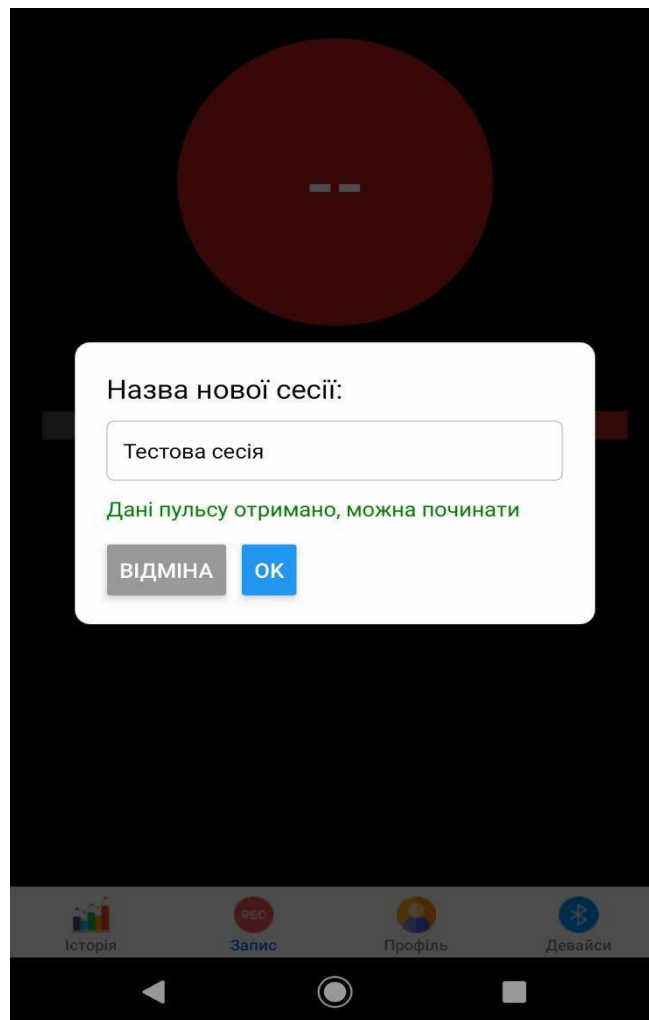


Рисунок 4.6 – Модальне вікно початку запису сесії

Якщо підключення до приладу стабільне та програма дозволяє перейти до запису сесії, модальне вікно закривається, та користувач бачить головне інформаційне вікно осунку, на якому у червоному колі відображається пульс користувача у реальному часі, а також поточна пульсова зона людини. Пульсову зону інтерактивно відображено на різнокольоровій прямій, де:

- сірий – 1 пульсова зона;
- зелений – 2 пульсова зона;
- жовтий – 3 пульсова зона;
- жовтогарачий – 4 пульсова зона;
- червоний – 5 польсова зона.

Для вказівки в якій зоні відповідно поточному пульсу користувач знаходиться, в реальному часі по прямій переміщується білий маркер, який і вказує на відповідну пульсову зону, що зображено на рисунку 4.7.

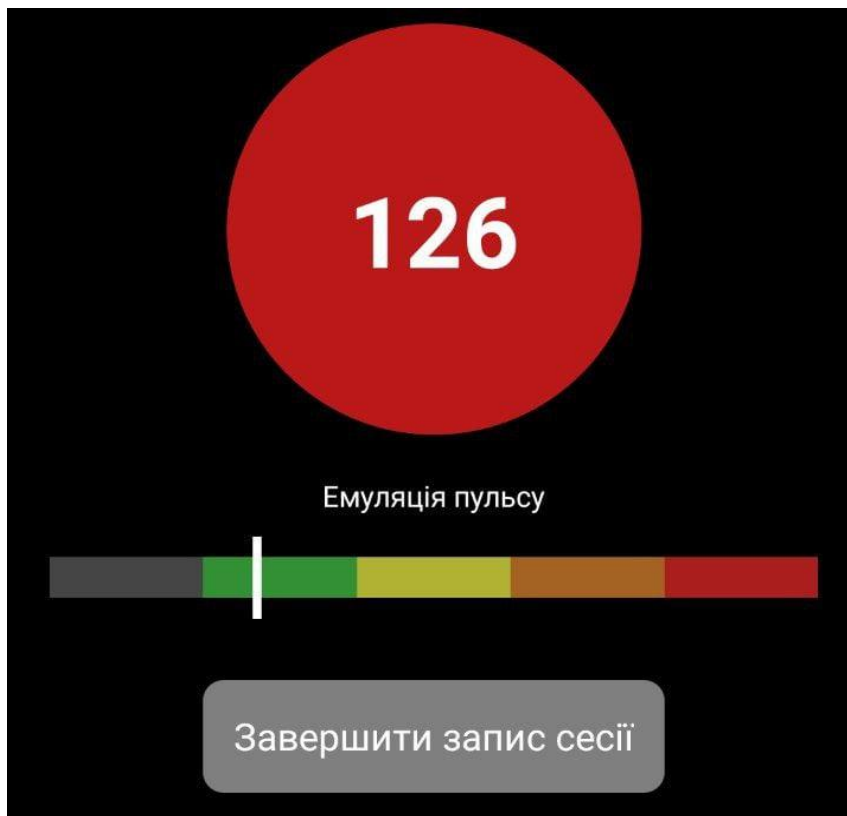


Рисунок 4.7 – Процес запису сесії

Якщо користувач вирішує закінчити запис пульсу він має натиснути на кнопку “Завершити запис сесії”, яка завершить запис та збереже дані у БД.

Коли користувач завершив запис сесії, він має змогу подивитися на інформацію про записану сесію, для цього потрібно перейти на екран “Історія” в нижній навігаційній панелі. Після переходу на цей екран, користувач побачить список усіх записаних сесій у вигляді відокремлених один від одної плиток, зображено на рисунку 4.8.

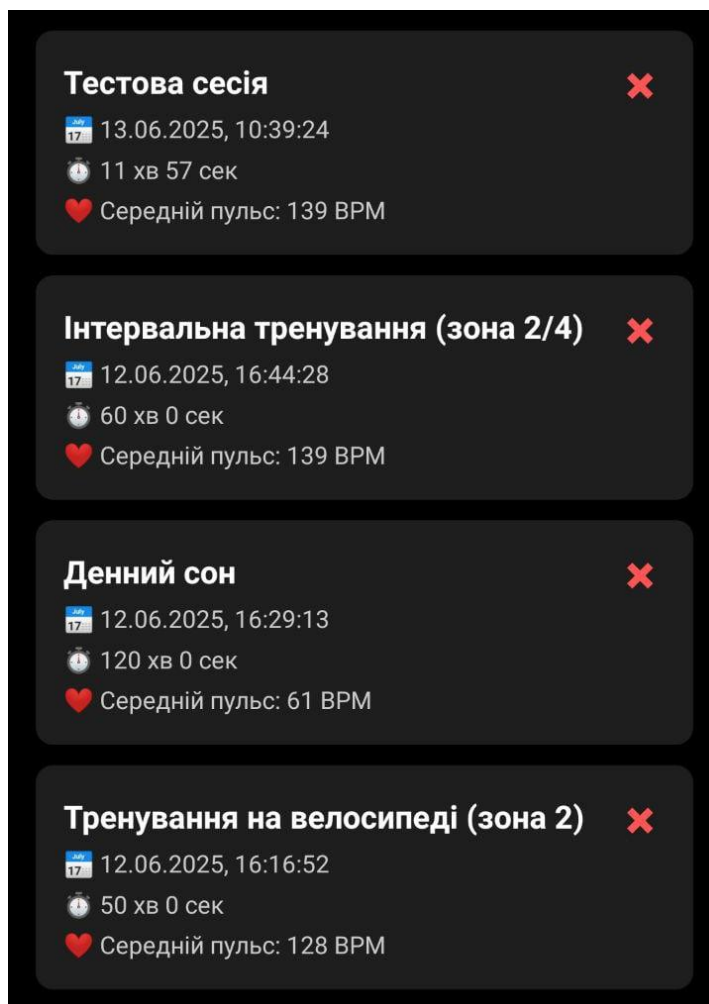


Рисунок 4.8 – Екран історії записаних сесій

Якщо користувач захоче видалити сесію з бази даних – він має натиснути на червоний хрестик, який є на плиточці кожної сесії у правому верхньому куті.

У ході тестування було проведено 1 тестову сесію, при першому перевірочному запуску, яка не несе ніякої корисної інформації, а також 3 повноцінних сесії, кожна з яких тривала більше 50-ти хвилин.

Денний сон – це сесія, яка тривала 2 години, під час якої прилад вимірював ЧСС під час сну, подробиці сесії наведено на рисунку 4.9.

← Деталі сесії

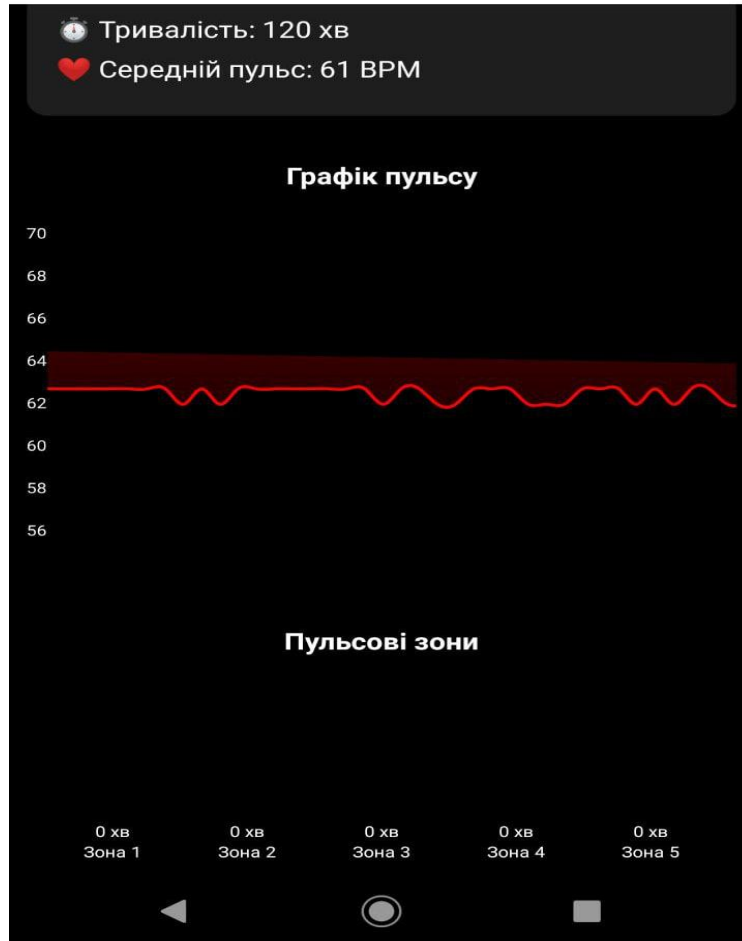


Рисунок 4.9 – Інформація сесії денного сну

На графіку бачимо відносно рівний графік ЧСС, без стрімких піків та спадів, так як під час сну пульс максимально стабільний, та на дистанції поступово знижується від початкового значення під час засинання. Також бачимо, що час проведення у всіх 5-ти пульсових зонах дорівнює нулю хвилин, так як значення пульсу не підвищувалось до значення навіть першої пульсової зони під час сесії.

Тренування на велосипеді(зона 2) – це сесія яка була направлена на максимально довге утримання ЧСС у другій пульсовій зоні, що є практично на 100% аеробним навантаженням без закислення м'язів, що розвиває загальну витривалість, подробиці сесії наведено на рисунку 4.10.

← Деталі сесії

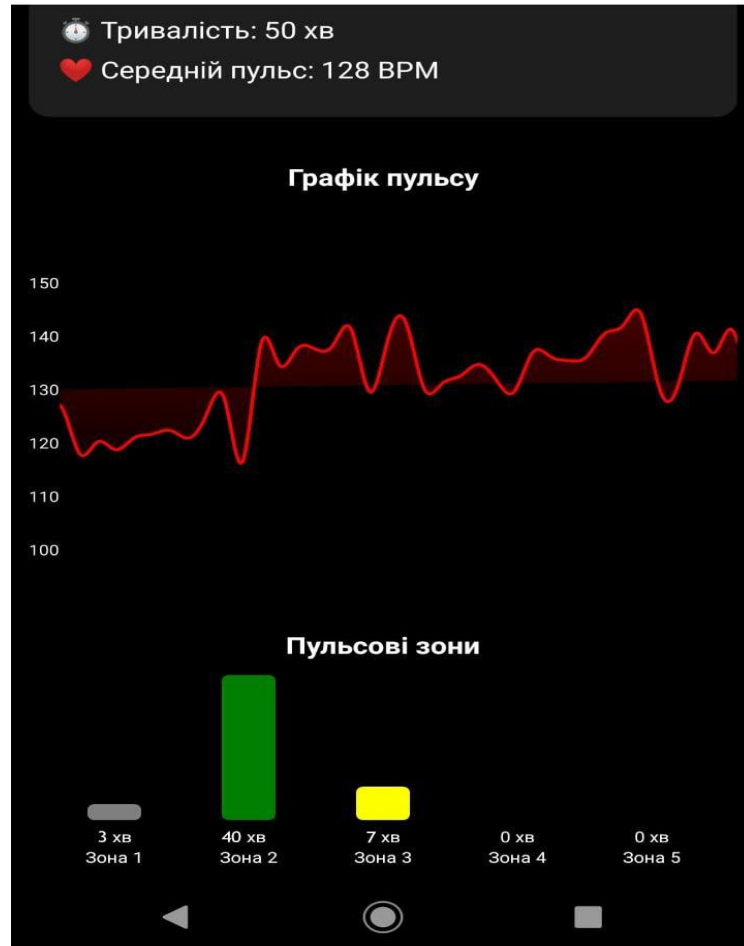


Рисунок 4.10 – Інформація сесії тренування на велосипеді (зона 2)

За інформацією про час проведення у пульсових зонах, можна зробити висновок, що цілі, які були поставлені на тренування виконано. З загального часу тренування, у другій пульсовій зоні було проведено 40 хвилин. 3 хвилини знадобилася на початку тренування, щоб підняти ЧСС з початкового значення до значень 2 пульсової зони, та інколи відбувались підвищення пульсу та вхід у 3 пульсову зону, що загалом склало 7 хвилин часу.

Інтервальне тренування (зона 2 та 4) – це сесія яка була направлена на запис інтервального тренування, що складалось з 20 хвилин зони 2 на початку, а потім інтервали: 4 хвилини максимального навантаження та 4 хвилини відпочинку у зоні 2, подробиці сесії наведено на рисунку 4.11.

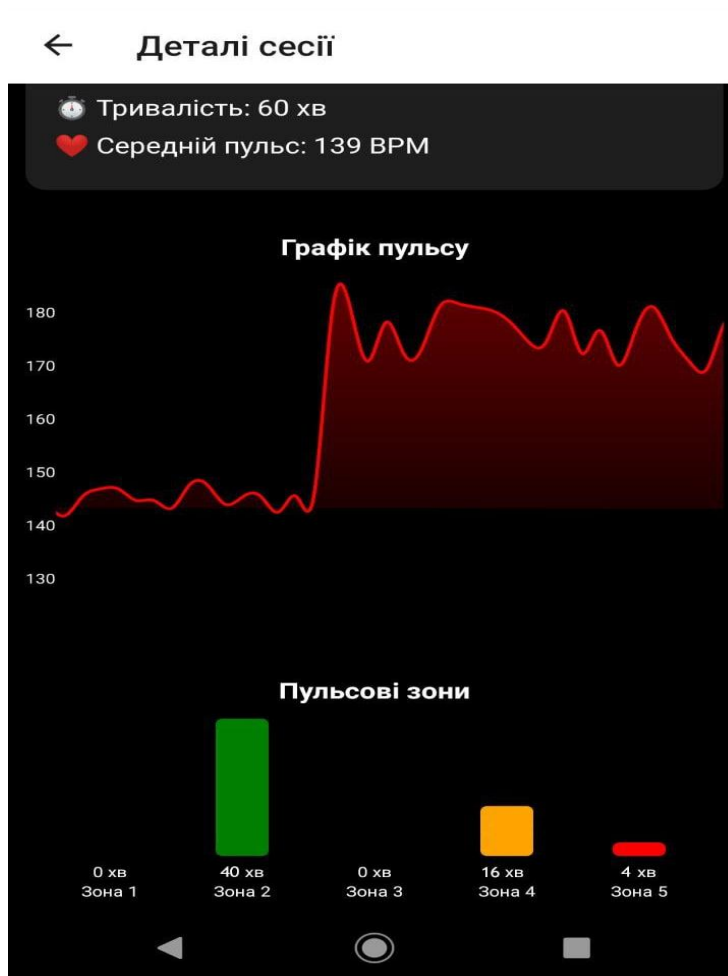


Рисунок 4.11 – Інформація сесії інтервального тренування (зона 2 та 4)

На графіку бачимо момент переходу з відпочинку до інтервалу максимального навантаження, пульс стрімко зростає близько до максимальних показників. Також видно за зонами пульсу, що загалом 40 хвилин ми провели у зоні 2, та сумарно 20 хвилин провели у 4 та 5 пульсових зона, що склали самі інтервали.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розроблено повнофункціональний прототип персонального пульсометра на основі мікроконтролера ESP32-C3 та оптичного сенсора MAX30102, а також створено мобільний застосунок для Android з інтеграцією BLE-зв'язку, локального зберігання та візуалізації даних. Реалізація даного комплексу дозволяє проводити вимірювання частоти серцевих скорочень у реальному часі, здійснювати запис сесій за ініціативою користувача та виконувати подальший аналіз результатів у зручному візуальному форматі.

Основні результати наведено нижче.

1. Було проведено системний аналіз сучасних підходів до моніторингу ЧСС у побутових умовах. Показано, що фотоплетизмографія, попри чутливість до артефактів, є найбільш придатною технологією для вбудованих носимих пристроїв, завдяки простоті реалізації, компактності та енергоефективності.

2. Апаратна частина прототипу пульсометра побудована на базі енергоефективного мікроконтролера ESP32-C3 з вбудованим BLE-модулем та сенсора MAX30102. Для забезпечення автономної роботи реалізовано живлення від літій-полімерного акумулятора з елементами захисту, індикації та стабілізації напруги. За допомогою 3D-принтера було надруковано корпус, адаптований для натільного використання.

3. Реалізовано прошивку мікроконтролера ESP32-C3 у середовищі Arduino IDE, яка забезпечує ініціалізацію інтерфейсів, зчитування IR-сигналу, розрахунок ЧСС, контроль втрати сигналу з пальця та передачу даних через BLE. Забезпечено оновлення ЧСС з інтервалом 2 с, фільтрацію сплесків і реагування на розриви контакту.

4. Створено мобільний застосунок для Android за допомогою React Native CLI з підтримкою BLE-з'єднання, відображення ЧСС у реальному

часі, запуску/завершення сесій, візуалізації пульсових зон, локального зберігання даних у SQLite, історії сесій та графічного аналізу результатів. Інтерфейс оптимізовано для швидкої взаємодії з користувачем та інтуїтивного керування.

5. Розроблений пристрій забезпечує точне та стабільне вимірювання пульсу, надаючи користувачу автономний інструмент для самопостереження. У поєднанні з мобільним застосунком, він дозволяє не лише фіксувати ЧСС, але й здійснювати постсесійний аналіз, що наближає функціональність рішення до комерційних аналогів.

6. Прототип запропонованого пульсометра є недорогим у виробництві, масштабованим та не залежним від хмарних платформ. Це дозволяє використовувати його у дослідницьких цілях, в особистих фітнес-програмах або як основу для розробки подальших розумних натільних пристроїв.

Запропонований у роботі підхід до розробки персонального пульсометра дозволяє об'єднати простоту, автономність, функціональність та адаптивність до користувача. Результати роботи демонструють практичну реалізованість і технічну ефективність рішення, що свідчить про успішне досягнення поставленої мети.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ESP32-C3 Technical Reference Manual. Espressif Systems. 2023. URL: <https://surl.li/qzheeh> (датазвернення: 10.05.2024).
2. MAX30102 Pulse Oximeter and Heart-Rate Sensor. Maxim Integrated. URL: <https://surl.li/bzeicf> (датазвернення: 15.05.2024).
3. SparkFun MAX30105 Particle Sensor Hookup Guide. SparkFun Electronics. URL: <https://surl.li/tzrmxd> (датазвернення: 23.05.2024).
4. Bluetooth Core Specification v5.0. Bluetooth SIG. URL: <https://surl.li/lfaenh> (дата звернення: 23.05.2024).
5. React Native Documentation: Environment Setup. React Native. URL: <https://surl.li/cc/jopxpo> (датазвернення: 24.05.2024).
6. SQLite Documentation. SQLite Consortium. URL: <https://surl.li/cc/dvxfmx> (датазвернення: 24.05.2024).
7. Allen P., Holberg J. Principles of Biomedical Instrumentation. Amsterdam: Academic Press, 2021. 354 p.
8. Panja A. K. Biomedical Sensors and Smart Sensing: A Beginner's Guide. Primers in Biomedical Imaging Devices and Systems. New York: Elsevier, 2022. 298 p.
9. Wang J. (editor). Electrochemical Sensors, Biosensors and Their Biomedical Applications. Amsterdam: Elsevier, 2017. 432 p.
10. Rafiquzzaman M. Fundamentals of Digital Logic and Microcontrollers. Hoboken: John Wiley & Sons, 2014. 528 p.
11. Webb A. G. Principles of Biomedical Instrumentation. Cambridge: Cambridge Univ. Press, 2018. 344 p.
12. IDC Worldwide Quarterly Wearable Device Tracker. International Data Corporation. 2023. URL: <https://surl.li/cuslcz> (датазвернення: 01.06.2024).