

Галицький фаховий коледж імені В'ячеслава Чорновола  
відділення комп'ютерних технологій  
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділення

комп'ютерних технологій

Наталія СТЕФУРАК / \_\_\_\_\_ /

(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2025 р.

### ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

освітньо-професійного ступеня «фаховий молодший бакалавр»

зі спеціальності 123 «Комп'ютерна інженерія»

на тему:

«Система відстеження фізичних показників спортсмена під час тренування»

Студент групи КІ-41

Едуардо САВЧУК

\_\_\_\_\_  
(підпис)

Керівник роботи

Василь ПАВЛЮС

\_\_\_\_\_  
(підпис)

Консультанти:

з техніко-економічного  
обґрунтування

Любов МЕЛЕНЧУК

\_\_\_\_\_  
(підпис)

Нормоконтролер

Оксана СИРОТЮК

\_\_\_\_\_  
(підпис)

Тернопіль – 2025

Галицький фаховий коледж імені В'ячеслава Чорновола  
відділення комп'ютерних технологій  
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділенням  
комп'ютерних технологій

Наталія СТЕФУРАК / \_\_\_\_\_ /

підпис

« \_\_\_ » \_\_\_\_\_ 2024 р.

### ЗАВДАННЯ

на кваліфікаційну роботу

на здобуття освітньо-професійного ступеня «фаховий молодший бакалавр»

студенту Савчуку Едуарду Ігоровичу

---

(прізвище, ім'я та по-батькові студента)

1. Тема роботи: «Система відстеження фізичних показників спортсмена під час тренувань», затверджена наказом по коледжу від «25» листопада 4024р., №253а-н.
2. Термін здачі студентом завершеної роботи: «27» червня 2025 р.
3. Вихідні дані до роботи: актуальні методики та засоби вимірювання частоти серцевих скорочень, наявні рішення, технічне завдання.
4. Перелік питань, які повинні бути розроблені в роботі:
  - а) основна частина дослідження предметної області, формалізація вимог до системи, проектування структури та алгоритму роботи системи, реалізація та тестування системи.
  - б) техніко-економічне обґрунтування аналіз ринку наявних рішень, дослідження кошторису реалізації системи, підрахунок загальної вартості розробки системи.

5. Перелік графічного матеріалу: структурна схема системи, блок-схема алгоритму роботи системи, схема з'єднання компонентів, електрична принципова схема.

6. Консультанти роботи:

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко-економічного обґрунтування	<p>Меленчук Л.І. вчена ступінь, звання</p> <hr/> <p>П.І.П. консультанта</p>		

### КАЛЕНДАРНИЙ ПЛАН кваліфікаційної роботи

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	<i>Вибір теми, ознайомлення з вимогами до кваліфікаційної роботи.</i>	10.11.2024	25.11.24
2.	<i>Дослідження предметної області, огляд типових рішень.</i>	28.11.2024	07.02.2025
3.	<i>Дослідження технологій реалізації.</i>	10.02.2025	19.02.2025
4.	<i>Розробка функціональних вимог до системи та робота над її структурою.</i>	21.02.2025	28.03.2025
5.	<i>Проектування системи та підготовка відповідного графічного матеріалу.</i>	29.02.2025	11.03.2025
6.	<i>Вибір, встановлення та налаштування середовища реалізації.</i>	12.03.2025	13.03.2025
7.	<i>Реалізація системи та написання відповідного розділу кваліфікаційної роботи.</i>	14.03.2025	22.04.2025
8.	<i>Доопрацювання апаратної складової.</i>	23.04.2025	28.04.2025
9.	<i>Опрацювання економічного розділу та написання відповідного розділу кваліфікаційної роботи.</i>	01.05.2025	06.05.2025
10.	<i>Тестування системи та усунення недоліків.</i>	07.05.2025	29.05.2025
11.	<i>Робота над оформленням пояснювальної записки.</i>	01.06.2025	15.06.2025
12.	<i>Попередній захист кваліфікаційної роботи, доопрацювання.</i>	16.06.2025	16.06.2025
13.	<i>Підготовка до захисту кваліфікаційної роботи.</i>	17.06.2025	26.06.2025
14.	<i>Захист кваліфікаційної роботи.</i>	27.06.2025	27.06.2025

Дата видачі «25» листопада 2024р. Керівник \_\_\_\_\_ / Василь ПАВЛЮС

Завдання прийняв до виконання \_\_\_\_\_ / Едуардо САВЧУК

## Реферат

Кваліфікаційна робота. Система відстеження фізичних показників спортсмена під час тренування. 61 сторінок, 22 рисунки, 2 додатки, 6 джерел.

Метою даної роботи є розробка та реалізація системи відстеження фізичних показників спортсмена під час тренувань. Система повинна відстежувати показники серцебиття спортсмена під час тренувань та сповіщати про перевищення рекомендованих значень з метою уникнення потенційно небезпечних станів.

Для реалізації проєкту було використано мікроконтролер ESP32, середовище розробки Arduino IDE для програмування мікроконтролера ESP, бібліотеки для роботи з периферійними пристроями та онлайн сервіс Thingier.io.

У роботі розглянуто процес розробки та реалізації системи відстеження фізичних показників спортсмена під час тренувань, представлено її ключові характеристики та переваги, а також надано рекомендації щодо її можливого вдосконалення.

ФІЗИЧНІ ПОКАЗНИКИ СПОРТСМЕНА, ЧСС, ARDUINO, ESP32, THINGER.IO, ІНТЕРНЕТ РЕЧЕЙ.

## Abstract

Graduation Project. Athlete's Physical Parameter Monitoring System During Training. 61 pages, 22 figures, 2 appendices, 6 sources.

The aim of this work is to develop and implement a system for tracking an athlete's physical parameters during training. The system is designed to monitor the athlete's heart rate during exercise and provide alerts when recommended limits are exceeded in order to prevent potentially dangerous conditions.

The project was implemented using an ESP32 microcontroller, the Arduino IDE development environment for ESP programming, peripheral device libraries, and the online service Thingier.io.

The thesis presents the development and implementation process of the athlete's physical parameter tracking system, highlights its key features and advantages, and provides recommendations for potential improvements.

ATHLETE'S PHYSICAL PARAMETERS, HEART RATE, ARDUINO, ESP32, THINGER.IO, INTERNET OF THINGS.

## ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ.....	8
1.1 Змістовний аналіз предметної області .....	8
1.3 Постанова завдання.....	15
2 ПРОЄКТУВАННЯ СИСТЕМИ.....	16
2.1 Визначення компонентів системи .....	16
2.2 Проєктування структури системи .....	17
2.3 Алгоритм роботи системи.....	19
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ.....	21
3.1 Вибір компонентів системи.....	21
3.2 Реалізація принципової електричної схеми та монтаж пристрою .....	25
3.3 Реалізація програмного коду.....	27
3.4 Підключення та налаштування сервісу thinger.io .....	34
3.5 Тестування роботи системи .....	38
4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ПРОЄКТУ.....	44
4.1 Аналіз ринку .....	44
4.2 Розрахунок витрат на реалізацію.....	44
4.3 Обґрунтування доцільності розробки .....	45
ВИСНОВКИ.....	48
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	50
ДОДАТКИ.....	51

					<i>КР.КІ 25.015.11.000 ПЗ</i>		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Савчук Е.І.			Лім.	Арк.	Акрушів
Перевір.		Павлюс В.П.			5	61	
Рецензент		Кузик В.М.			ГФК. ВКТ. КІ-41		
Н. Контр.		Сиротюк О.Б.					
Затверд.		Стефурак Н.А.					
					<i>Система відстеження фізичних показників спортсмена під час тренування</i>		

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЧСС – Частота серцевих скорочень

BLE – Bluetooth Low Energy

GPIO – General-Purpose Input/Output

I2C – Inter-Integrated Circuit

IDE – Integrated Development Environment

IoT – Internet of Things

OLED – Organic Light-Emitting Diode

ECG – Electrocardiogram

RA – Right Arm

RL – Right Leg

LA – Left Arm

LL – Left Leg

					<i>КР.КІ 25.015.11.000 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		6

## ВСТУП

У сучасному світі спорту та фітнесу точне відстеження фізичних показників спортсмена під час тренувань стає все більш важливим аспектом для досягнення оптимальних результатів. Серцебиття є одним із найважливіших показників, який дозволяє оцінити інтенсивність навантаження та ефективність тренувального процесу.

Завданням даного проєкту є розробка системи моніторингу серцевого ритму спортсмена на базі мікроконтролера ESP32 та модуля ЕКГ з використанням бездротових технологій для передачі даних на смартфон користувача. Ключовою особливістю системи буде можливість відстеження показників у реальному часі, що дозволить спортсмену або тренеру своєчасно коригувати інтенсивність тренування.

Актуальність роботи обумовлена зростаючим попитом на персоналізовані рішення для моніторингу здоров'я та фізичної активності, а також необхідністю в більш доступних альтернативах комерційним продуктам. Створення такої системи дозволяє вирішити практичну задачу та одночасно поглибити розуміння принципів роботи мікроконтролерних систем та засобів бездротової комунікації.

Основна мета проєкту полягає не в створенні революційного чи унікального продукту, а в закріпленні практичних навичок аналізу предметної області, проєктування та реалізації систем на базі мікроконтролерів, а також роботи з бездротовими технологіями передачі даних. Проєкт включатиме всі етапи розробки: від дослідження наявних рішень до створення функціонального прототипу та його тестування.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

## 1.1 Змістовний аналіз предметної області

Моніторинг серцевого ритму є одним із ключових аспектів контролю ефективності та безпеки тренувань як для професійних спортсменів, так і для аматорів. Частота серцевих скорочень (ЧСС) надає об'єктивну інформацію про реакцію організму на фізичне навантаження та допомагає оптимізувати тренувальний процес.

Важливість моніторингу серцебиття під час тренувань обумовлена кількома факторами. По-перше, це дозволяє визначити оптимальні зони тренувань, які зазвичай вираховуються як відсоток від максимальної ЧСС. Наприклад, для жироспалювання ефективним є діапазон 60-70% від максимальної ЧСС, для підвищення витривалості – 70-80%, а для розвитку анаеробних можливостей – 80-90%. По-друге, контроль ЧСС запобігає перетренуванню та допомагає уникнути потенційно небезпечних станів, особливо для людей з серцево-судинними захворюваннями. По-третє, дані про серцевий ритм дозволяють оцінити рівень відновлення організму та коригувати інтенсивність наступних тренувань.

Сучасні методи вимірювання серцебиття під час фізичної активності включають [5]:

1. Оптичні сенсори – найпоширеніший метод у носимих пристроях (фітнес-браслети, смарт-годинники). Принцип роботи базується на фотоплетизмографії – вимірюванні змін відбиття світла від крові, що пульсує в капілярах. Переваги: компактність, зручність. Недоліки: нижча точність при інтенсивних рухах.

2. Електрокардіографічні (ЕКГ, ECG – electrocardiogram) датчики – забезпечують найвищу точність вимірювань. Принцип роботи ґрунтується на реєстрації електричних потенціалів, що виникають під час роботи серця.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

Традиційно використовуються нагрудні ремені, але існують й більш компактні рішення.

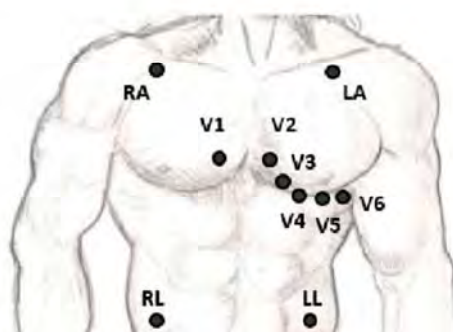
3. Біоімпедансні методи – вимірюють зміни електричного опору тканин під час пульсації крові.

ECG-датчики працюють на основі вимірювання електричних сигналів, які генеруються серцевим м'язом під час його скорочення та розслаблення. Серцевий цикл супроводжується характерною послідовністю електричних потенціалів, які можна зафіксувати за допомогою електродів, розташованих на тілі.

Основний принцип роботи ЕКГ-датчика полягає в наступному:

- Електроди, розташовані на тілі, фіксують різницю потенціалів між різними точками.
- Отриманий сигнал є дуже слабким (мілівольти), тому він підсилюється спеціальними схемами.
- Після підсилення сигнал фільтрується для усунення шумів та артефактів.
- Оброблений сигнал перетворюється в цифрову форму для подальшої обробки мікроконтролером.

На рисунку 1.1 показано стандартне 12-канальне ЕКГ-дослідження, де електроди розміщені на різних частинах тіла. Це дозволяє отримати повну електричну карту серцевої діяльності з різних кутів, що є типовим для медичної діагностики серцево-судинних захворювань.



RA – right arm (права рука), RL – right leg (права нога), LA – left arm (ліва рука), LL – left leg (ліва нога)

Рисунок 1.1 – Розташування електродів в 12-канальній системі ЕКГ

Для використання в портативних системах моніторингу найчастіше застосовують спрощені варіанти ЕКГ-датчиків з меншою кількістю електродів. Такі датчики спеціально розроблені для портативних рішень і можуть легко інтегруватися з мікроконтролерами, такими як ESP32.

Використання мікроконтролерних систем на базі ESP32 для аналізу серцевого ритму має значні переваги: висока обчислювальна потужність, вбудовані модулі Bluetooth і WiFi для бездротової передачі даних, низьке енергоспоживання та компактність. Це робить ESP32 ідеальною платформою для створення носимих пристроїв для моніторингу фізичних показників спортсменів.

## 1.2 Огляд наявних рішень

### 1.2.1 Медичний електрокардіограф BTL-08 LC

BTL-08 LC є професійним медичним пристроєм для електрокардіографічних досліджень, який широко використовується в клінічній практиці. Це 12-канальний електрокардіограф, який забезпечує одночасну реєстрацію ЕКГ з 12 електродів, що дозволяє отримати повну картину електричної активності серця [2].

Пристрій оснащений високочутливими сенсорами, які здатні сприймати мінімальні зміни електричних потенціалів (до 0,01 мВ). Система використовує 10 електродів – шість грудних (V1-V6) та чотири для кінцівок. Електроди розміщуються на точно визначених ділянках тіла згідно зі стандартною схемою розташування для 12-канального ЕКГ.

BTL-08 LC має вбудований кольоровий сенсорний дисплей високої роздільної здатності, на якому відображаються ЕКГ-хвилі в режимі реального часу. Пристрій оснащений термопринтером для друку результатів на термопапері з автоматичною інтерпретацією даних.

Програмне забезпечення електрокардіографа дозволяє:

- автоматично визначати основні параметри ЕКГ;
- виявляти аритмії та інші патологічні стани;

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

- зберігати записи у внутрішній пам'яті;
- передавати дані на комп'ютер через USB або LAN-з'єднання.



Рисунок 1.2 – Медичний електрокардіограф BTL-08 LC

Основні переваги BTL-08 LC включають високу точність вимірювань, надійність, широкий функціонал та можливість інтеграції з іншими медичними системами.

Недоліками є висока вартість (приблизно 3000-4000 євро), необхідність спеціальної підготовки для використання та інтерпретації результатів, а також відносно великі розміри, що робить неможливим його використання під час фізичних тренувань.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

### 1.2.2 Нагрудний датчик серцевого ритму Polar H10

Polar H10 є одним із найпопулярніших та найточніших нагрудних датчиків серцевого ритму, який широко використовується як професійними спортсменами, так і аматорами. На відміну від медичних електрокардіографів, Polar H10 використовує спрощену схему з одноканальною реєстрацією ЕКГ за допомогою двох електродів, вбудованих у нагрудний ремінь (рис. 1.3).

Пристрій складається з ременя з електропровідними зонами та знімного модуля, який містить електроніку для обробки сигналів та передачі даних. Електроди виготовлені з м'якого еластичного матеріалу з вкрапленнями провідних частинок, що забезпечує комфортне носіння та хороший контакт зі шкірою навіть під час інтенсивних тренувань [1].



Рисунок 1.3 – Нагрудний датчик серцевого ритму Polar H10

Polar H10 використовує запатентовану технологію вимірювання електрокардіограми, яка забезпечує точність, наближену до медичних пристроїв (похибка вимірювання ЧСС  $\pm 1$  уд/хв). Датчик здатний реєструвати ЕКГ-сигнал з частотою дискретизації 130 Гц, що достатньо для аналізу варіабельності серцевого ритму.

Обробка даних виконується у реальному часі вбудованим мікропроцесором, після чого інформація передається на зовнішні пристрої (смартфон, спортивний годинник, тренажерне обладнання) через Bluetooth або

власний протокол Polar GymLink. Пристрій також має вбудовану пам'ять на 1 тренування тривалістю до 30 годин.

Основними перевагами Polar H10 є доволі висока точність, тривалий час автономної роботи, водонепроникність та сумісність з широким спектром додатків та пристроїв. До недоліків можна віднести необхідність носіння нагрудного ременя, що може бути менш комфортним порівняно з оптичними датчиками в смарт-годинниках, та обмежений функціонал порівняно з повноцінними ЕКГ-системами.

До недоліків Polar H10 можна віднести наступне:

- висока вартість порівняно з базовими моніторами серцевого ритму (орієнтовна ціна від 100 євро);
- обмежені можливості для самостійної інтерпретації ЕКГ-даних;
- відсутність можливості перегляду повної ЕКГ-кривої в реальному часі;
- необхідність періодичної заміни ременя через зношування еластичного матеріалу та погіршення провідних властивостей електродів.

### 1.2.3 Портативний ЕКГ-монітор AliveCor KardiaMobile

KardiaMobile від компанії AliveCor є інноваційним портативним пристроєм для самостійного моніторингу електрокардіограми, який поєднує медичну точність з компактністю та зручністю використання [6].

KardiaMobile має розміри кредитної картки та вагу менше 20 грам. Пристрій складається з двох металевих електродів, розташованих на верхній поверхні, та електронного модуля з мікропроцесором і передавачем. Для вимірювання ЕКГ користувач розміщує пальці лівої та правої руки на відповідних електродах, створюючи таким чином одне відведення (рис. 1.4).

Пристрій не має дисплею – дані передаються на смартфон, де спеціальний застосунок Kardia App обробляє сигнал та відображає електрокардіограму. Процес запису займає 30 секунд, після чого застосунок автоматично аналізує ЕКГ та класифікує її як нормальну, ненормальну або некласифіковану.

KardiaMobile використовує запатентовану технологію обробки сигналу, яка дозволяє отримувати клінічно значимі результати навіть за умови наявності звичайних побутових електромагнітних перешкод. Частота дискретизації сигналу становить 300 Гц, а смуга пропускання – 0,5-40 Гц, що відповідає медичним стандартам для амбулаторних ЕКГ-пристроїв.

Живлення здійснюється від змінної батареї CR2032, яка забезпечує до 12 місяців роботи при щоденному використанні. Пристрій не є водонепроникним і не призначений для використання під час фізичних тренувань.



Рисунок 1.4 – ЕКГ-монітор AliveCor KardiaMobile

Основними перевагами KardiaMobile є клінічна точність, компактність, простота використання, тривалий час автономної роботи та можливість зберігання і надсилання записів ЕКГ лікарям.

До недоліків належать:

- вартість базової версії становить понад 100 доларів США;
- необхідність знерухомилення рук під час вимірювання;
- неможливість безперервного моніторингу;
- відсутність функціоналу для використання під час спортивних тренувань.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

### 1.3 Постановка завдання

На основі проведеного аналізу предметної області та огляду існуючих рішень можна сформулювати перелік конкретних завдань, які необхідно виконати в рамках дипломного проєкту:

– Розробити пристрій на базі мікроконтролера ESP32 та ЕКГ модуля для вимірювання серцевого ритму спортсмена під час тренувань.

– Реалізувати систему збору та первинної обробки даних з ЕКГ-датчика, що забезпечить точне визначення частоти серцевих скорочень та можливість виявлення аномалій серцевого ритму.

– Реалізувати відображення на пристрої ЧСС в реальному часі.

– Реалізувати функцію попередження користувача при перевищенні встановлених меж ЧСС через звукові сигнали.

– Забезпечити стабільну бездротову передачу даних у «хмару» за допомогою технології WiFi.

– Реалізувати можливість перегляду показників ЧСС на мобільному пристрої на основі інформації переданої в «хмару».

– Провести тестування системи для перевірки точності вимірювань, надійності роботи та зручності використання.

– Сформулювати рекомендації щодо можливих напрямків розвитку системи, включаючи розширення функціоналу, інтеграцію з іншими фітнес-платформами та покращення характеристик.

Виконання зазначених завдань дозволить створити функціональний прототип системи моніторингу фізичних показників спортсмена, що продемонструє практичні навички розробки вбудованих систем на базі мікроконтролерів.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРОЄКТУВАННЯ СИСТЕМИ

В попередньому розділі було визначено загальні принципи роботи систем моніторингу серцевого ритму, проаналізовано сучасні методи вимірювання серцебиття під час фізичної активності, розглянуто основні принципи роботи ЕКГ-датчиків. Крім того, було проаналізовано деякі наявні рішення, кожне з яких має як певні переваги, так і недоліки. Також було сформульовано список завдань, які необхідно виконати в ході реалізації проєкту.

Наступним етапом роботи є проєктування системи, який передбачає визначення її компонентів та способів їх взаємодії, проєктування апаратної та програмної структури системи, а також побудова алгоритму роботи системи. Розглянемо ці кроки детальніше.

### 2.1 Визначення компонентів системи

Для створення ефективної системи відстеження фізичних показників спортсмена під час тренування необхідно визначити ключові компоненти, які забезпечать надійний збір, обробку та передачу даних.

Система складатиметься з наступних компонентів:

Мікроконтролер – центральний елемент системи, який відповідає за обробку вхідних даних від сенсорів, керування периферійними пристроями та організацію бездротової передачі даних. Мікроконтролер має забезпечувати необхідну обчислювальну потужність, підтримку бездротових інтерфейсів та оптимальне енергоспоживання.

Електрокардіографічний модуль – спеціалізований сенсор для зняття електрокардіограми, що дозволяє отримувати дані про серцевий ритм спортсмена. Модуль має забезпечувати точність вимірювань та стабільність роботи в умовах фізичної активності.

Дисплей – компактний модуль для відображення основних показників безпосередньо на пристрої. Дисплей повинен бути енергоефективним, мати

достатню роздільну здатність для відображення числових значень та забезпечувати видимість при різних умовах освітлення.

Модуль бездротового зв'язку – компонент для організації передачі даних на хмарний сервіс та смартфон користувача. Має підтримувати стандарти бездротового зв'язку, необхідні для стабільної передачі даних.

Блок живлення – система елементів живлення (акумулятор). Блок повинен забезпечувати автономну роботу пристрою протягом всього тренування.

## 2.2 Проектування структури системи

Структура системи відстеження фізичних показників спортсмена формується на основі взаємодії визначених вище компонентів та потоків даних між ними. Загальна структура системи включає апаратну частину (пристрій моніторингу) та програмну частину (хмарний сервіс та інтерфейс користувача).

### 2.2.1 Апаратна структура системи

1. Підсистема збору даних – включає електрокардіографічний модуль, який взаємодіє безпосередньо з тілом спортсмена та збирає дані про серцевий ритм. Сигнали з модуля передаються на аналогово-цифровий перетворювач мікроконтролера.

2. Підсистема обробки даних – центральним елементом є мікроконтролер, який отримує дані від сенсорів, здійснює їх первинну обробку та підготовку до передачі.

3. Підсистема відображення – включає мініатюрний дисплей та необхідні інтерфейси для відображення ключових показників безпосередньо на пристрої.

4. Підсистема сповіщення – забезпечує звукові сповіщення при перевищенні допустимих значень ЧСС.

5. Підсистема бездротової передачі даних – відповідає за організацію зв'язку з хмарним сервісом через бездротовий інтерфейс.

6. Підсистема живлення – забезпечує енергією всі компоненти системи, включає акумулятор, схеми захисту та стабілізації напруги.

Структурна схема апаратною частини системи наведена на наступному рисунку.



Рисунок 2.1 – Структурна схема апаратної частини системи

### 2.2.2 Програмна структура системи

1. Вбудоване програмне забезпечення (firmware) – програмний код, що виконується на мікроконтролері та забезпечує:

- збір даних від сенсорів;
- обробку отриманих сигналів;
- виявлення пікових навантажень або аномалій;
- відображення інформації на дисплеї;
- подачу звукових сигналів;
- організацію бездротової передачі даних.

2. Хмарний сервіс – програмна платформа, яка забезпечує:

- прийом даних від пристрою моніторингу;
- формування вебінтерфейсу для доступу з браузера.

Зм.	Арк.	№ докум.	Підпис	Дата

3. Вебінтерфейс користувача – інтерфейс для перегляду даних через браузер мобільного пристрою, що забезпечує відображення поточних показників у реальному часі.

Структурна схема програмної частини системи наведена на наступному рисунку.

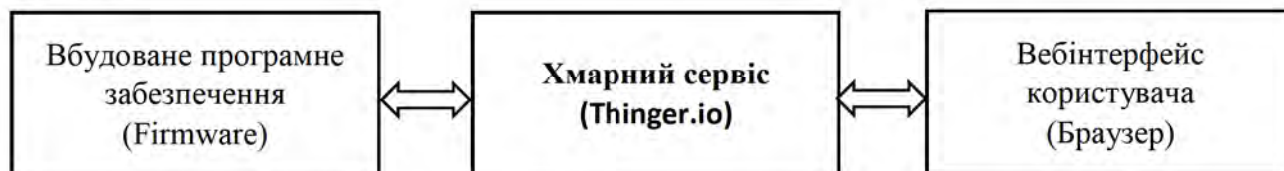


Рисунок 2.2 – Структурна схема програмної частини системи

### 2.2.3 Принцип роботи системи

Взаємодія між компонентами системи організована за наступним принципом:

1. Електрокардіографічний модуль знімає біомедичні сигнали з тіла спортсмена.
2. Отримані аналогові сигнали передаються на мікроконтролер для оцифрування та обробки.
3. Мікроконтролер обробляє сигнали та визначає частоту серцевих скорочень.
4. Поточні показники відображаються на дисплеї пристрою.
5. Одночасно дані передаються через бездротовий інтерфейс на хмарний сервіс.
6. Користувач отримує доступ до даних через браузер мобільного пристрою.

### 2.3 Алгоритм роботи системи

Алгоритм роботи системи відстеження фізичних показників спортсмена (додаток А) розбивається на декілька основних етапів:

1. Ініціалізація системи:
  - Включення пристрою.

- Ініціалізація периферійних модулів мікроконтролера.
  - Встановлення бездротового з'єднання з хмарним сервісом.
2. Основний цикл роботи під час тренування:
- Безперервне зняття даних з електрокардіографічного модуля.
  - Оцифрування аналогового сигналу.
  - Розрахунок миттєвої частоти серцевих скорочень (ЧСС).
  - Відображення поточної ЧСС на дисплеї пристрою.
  - Передача даних на хмарний сервіс через бездротовий інтерфейс.
3. Обробка критичних ситуацій:
- Моніторинг перевищення безпечних порогів ЧСС.
  - Сигналізація про пікові навантаження або аномалії.
4. Взаємодія з хмарним сервісом:
- Регулярна передача даних про ЧСС.
  - Підтримка з'єднання для забезпечення моніторингу в реальному часі
5. Завершення тренування:
- Передача фінальних показників на хмарний сервіс.
  - Виключення пристрою.

Ключовою особливістю системи є робота з оптимальними інтервалами опитування та обробки даних:

- Зняття сигналу з ЕКГ-модуля відбувається з достатньою частотою для забезпечення точності вимірювань.
- Обчислення ЧСС відбувається кожні 1-2 секунди.
- Оновлення дисплею відбувається кожну секунду для забезпечення актуальності відображення.
- Передача даних на хмарний сервіс здійснюється з регулярними інтервалами для забезпечення моніторингу в реальному часі.

Такий підхід дозволяє забезпечити оптимальний баланс між точністю вимірювань, оперативністю відображення даних та ефективністю роботи системи.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

## 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

### 3.1 Вибір компонентів системи

Як було зазначено в попередньому розділі, система складатиметься з мікроконтролера, OLED-дисплея, модуля ЕКГ, бузера для звукових сповіщень та блока живлення. Розглянемо детальніше дані компоненти.

#### 3.1.1 Плата мікроконтролера з вбудованим OLED-дисплеєм

Оскільки система повинна забезпечувати віддалений доступ через Інтернет, найзручнішим способом для цього є використання бездротової технології WiFi. З огляду на це, а також потребу у достатній кількості портів вводу-виводу для підключення зовнішніх модулів та дисплея, було прийнято рішення використати в ролі центрального контролера плату ESP32 Wemos LoLin32 з вбудованим OLED-дисплеєм (рис. 3.1) [4].

Ця плата побудована на потужному мікроконтролері ESP32, який має вбудовані модулі WiFi та Bluetooth, а також розширені можливості цифрового та аналогового вводу-виводу. Завдяки інтегрованому OLED-дисплею можна виводити ключову інформацію без потреби у зовнішніх екранах. Короткі технічні характеристики мікроконтролера наведено в таблиці 3.1, а характеристика вбудованого екрана – в таблиці 3.2.

Таблиця 3.1 – Характеристики мікроконтролера ESP32

Мікроконтролер	ESP32 (ESP32-D0WDQ6, Xtensa LX6 Dual-Core)
Частота процесора	До 240 МГц
Флеш-пам'ять	4 МБ
Оперативна пам'ять	520 КБ
Вбудований WiFi	802.11 b/g/n
Вбудований Bluetooth	BLE + Classic
Інтерфейси	GPIO, I2C, SPI, UART, ADC, DAC, PWM

Таблиця 3.2 – Характеристики вбудованого екрана

Тип	OLED (органічний світлодіодний)
Діагональ	0,96 дюйма
Роздільна здатність	128×64 пікселів
Контролер	SSD1306
Інтерфейс підключення	I2C
Напруга живлення	3,3 В

Завдяки високій обчислювальній здатності, енергоефективності, широкому набору інтерфейсів та вбудованому дисплею плата ESP32 Wemos LoLin32 забезпечує ефективну роботу системи збору, обробки, відображення та передавання даних у режимі реального часу.

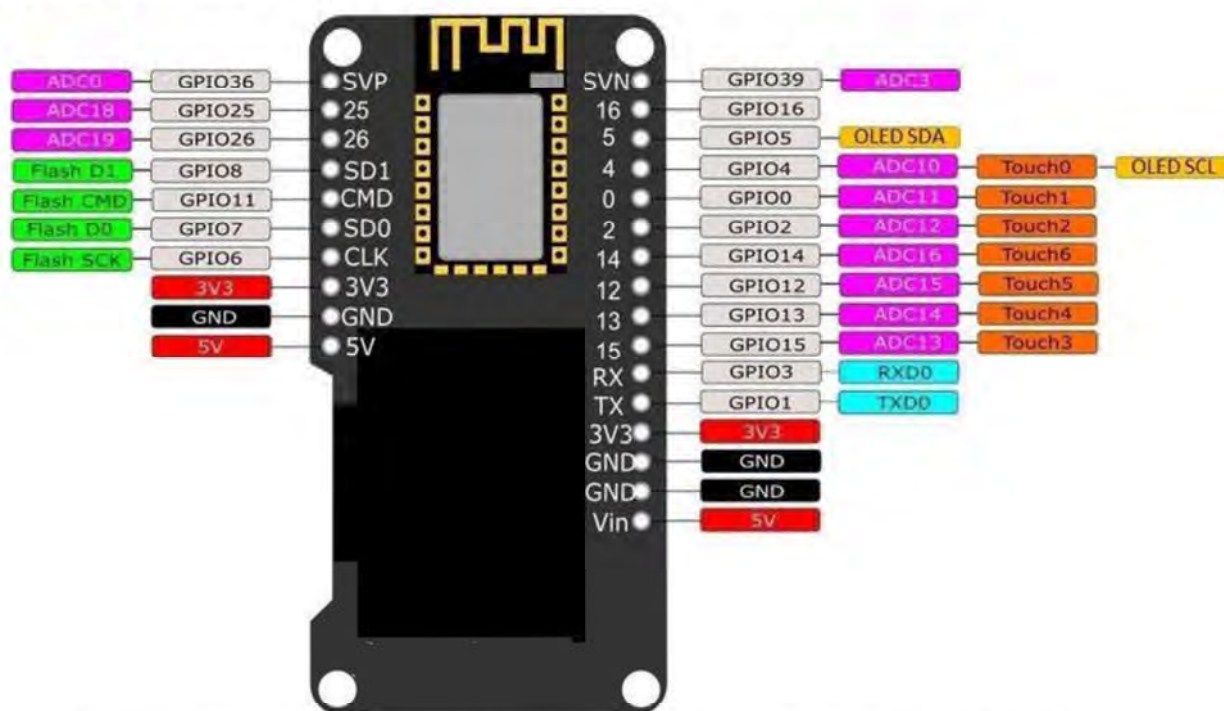


Рисунок 3.1 – Плата розробки ESP32 Wemos LoLin32 з OLED-екраном

### 3.1.2 Модуль ЕКГ

Для реалізації надійного зчитування електрокардіографічного сигналу в умовах мобільного середовища було обрано модуль AD8232 (рис. 3.2), який виявився оптимальним рішенням для проєкту завдяки своїй високій чутливості, простоті підключення, наявності вбудованої фільтрації шумів та низькому

енергоспоживанню. Інші варіанти (наприклад, багатоканальні медичні АЦП або саморобні підсилювачі) вимагають складнішого налаштування, більшої кількості компонентів та споживають більше ресурсів мікроконтролера для обробки сигналу.

Модуль AD8232 побудований на базі однойменного мікроконтролера від Analog Devices, призначеного для зняття біоелектричних сигналів із тіла людини. Він має вбудовані фільтри високих і низьких частот, що дозволяє отримати чистий ЕКГ-сигнал без складної цифрової обробки. До модуля підключаються три електроди: правий (RA), лівий (LA) та земля (RL). Аналоговий вихід сигналу передається на АЦП мікроконтролера через відповідний пін (для мікроконтролера ESP32 зазвичай це пін GPIO36), що дозволяє обробляти дані в реальному часі.

Основні технічні характеристики модуля AD8232 наведено в таблиці 3.3.

Таблиця 3.3 – Характеристики модуля ЕКГ AD8232

Контролер	AD8232
Частота дискретизації	До 100 Гц
Інтерфейс виводу даних	Аналоговий
Напруга живлення	3,3 В
Додаткові виводи	LO+ / LO– (виявлення від’єднання електродів)
Особливості	Вбудовані фільтри шумів, низьке енергоспоживання

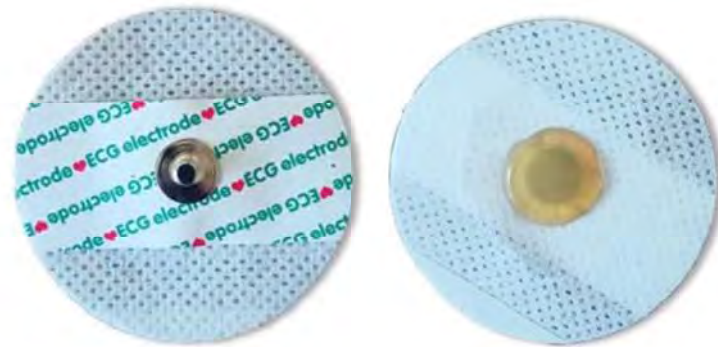
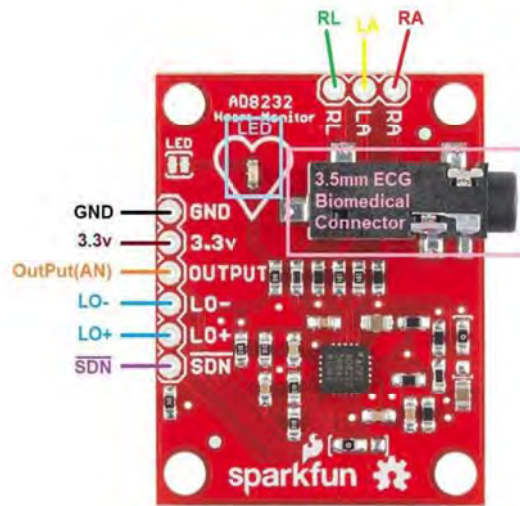


Рисунок 3.2 – Модуль AD8232 з трьома електродами

Зм.	Арк.	№ докум.	Підпис	Дата

КР.КІ 25.015.11.000 ПЗ

Арк.

24

### 3.1.3 Бузер

Бузер (або як ще його іноді називають зумер чи «пищалка») – найпростіший модуль для отримання звуку (рис. 3.3). Зазвичай використовується як сигнальний пристрій, який може бути електромеханічним, електронним або п'єзоелектричним. Бузер може бути активним або пасивним. Активний зумер має власний генератор звуку, тоді як пасивний зумер потребує зовнішнього генератора звуку.

Бузер часто використовується для звукової сигналізації, наприклад, при роботі з мікроконтролерами. Якщо нема потреби генерувати звукові сигнали різної тональності, то доцільніше використовувати активний бузер.

Типові технічні характеристики активного бузера наведені в таблиці 3.4.



Рисунок 3.3 – Активний бузер

Таблиця 3.4 – Типові характеристики активного бузера

Живлення	3 – 5 В
Струм	не більше 30 мА
Частота звуку	2000 – 2300 Гц
Розмір	12 x 9,5 мм

### 3.2 Реалізація принципової електричної схеми та монтаж пристрою

Ознайомившись із технічними параметрами та принципом дії ключових елементів системи, можемо перейти до створення її електричної схеми.

Для зручного проектування такої схеми доцільно використати середовище Fritzing – інструмент, що дозволяє розробляти електронні схеми й макети прототипів. За допомогою Fritzing користувач може компоувати схеми з віртуальних елементів, візуалізувати їхні з'єднання та створювати макети друкованих плат. Інтерфейс програми інтуїтивно зрозумілий, що робить її придатною для початківців і використання в навчальному процесі. Крім того,

Fritzing підтримує можливість експорту готових схем для подальшого виготовлення плат.

На рисунку 3.4 зображено принципову електричну схему системи, на рисунку 3.5 – її монтажну схему, а на рисунку 3.6 – готовий пристрій, змонтований на монтажній платі та з під'єднаними електродами.

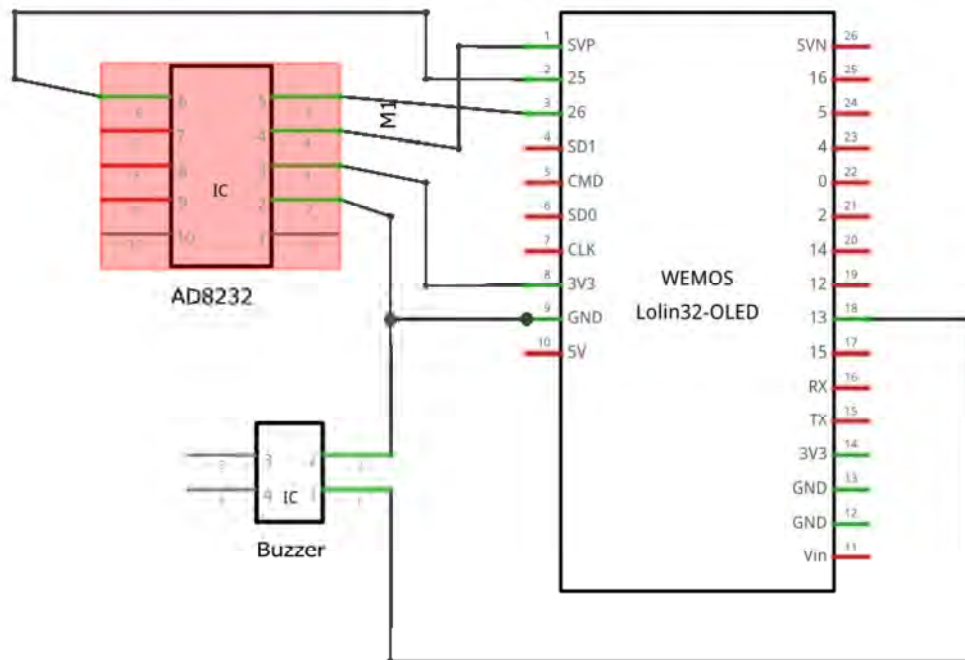


Рисунок 3.4 – Принципова електрична схема системи

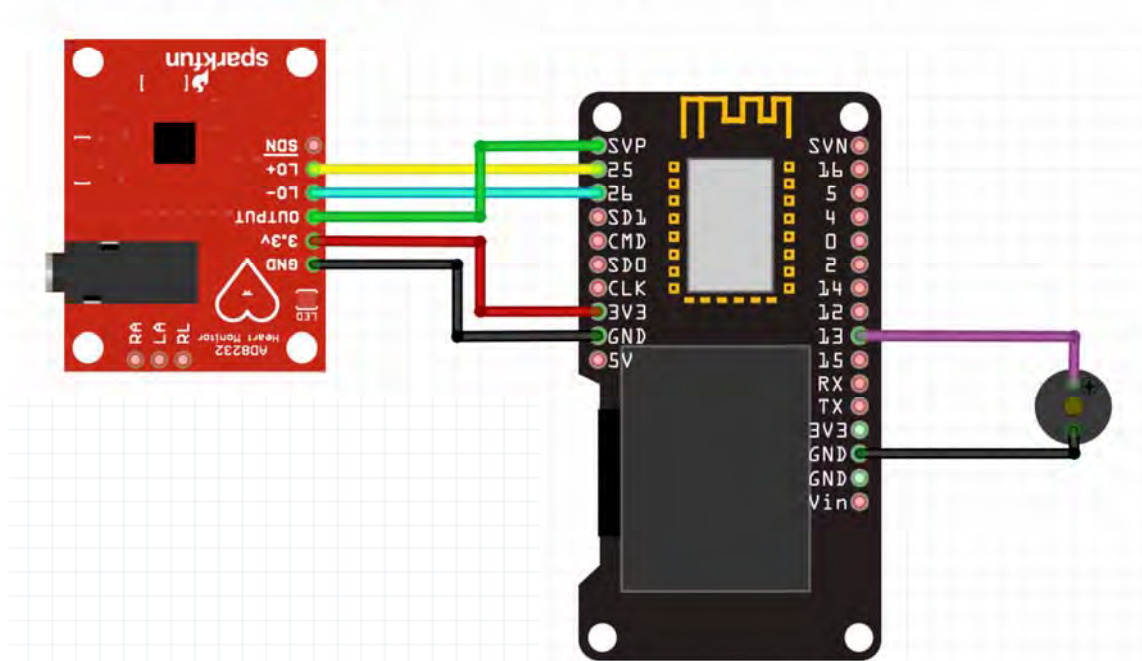


Рисунок 3.5 – Монтажна схема системи

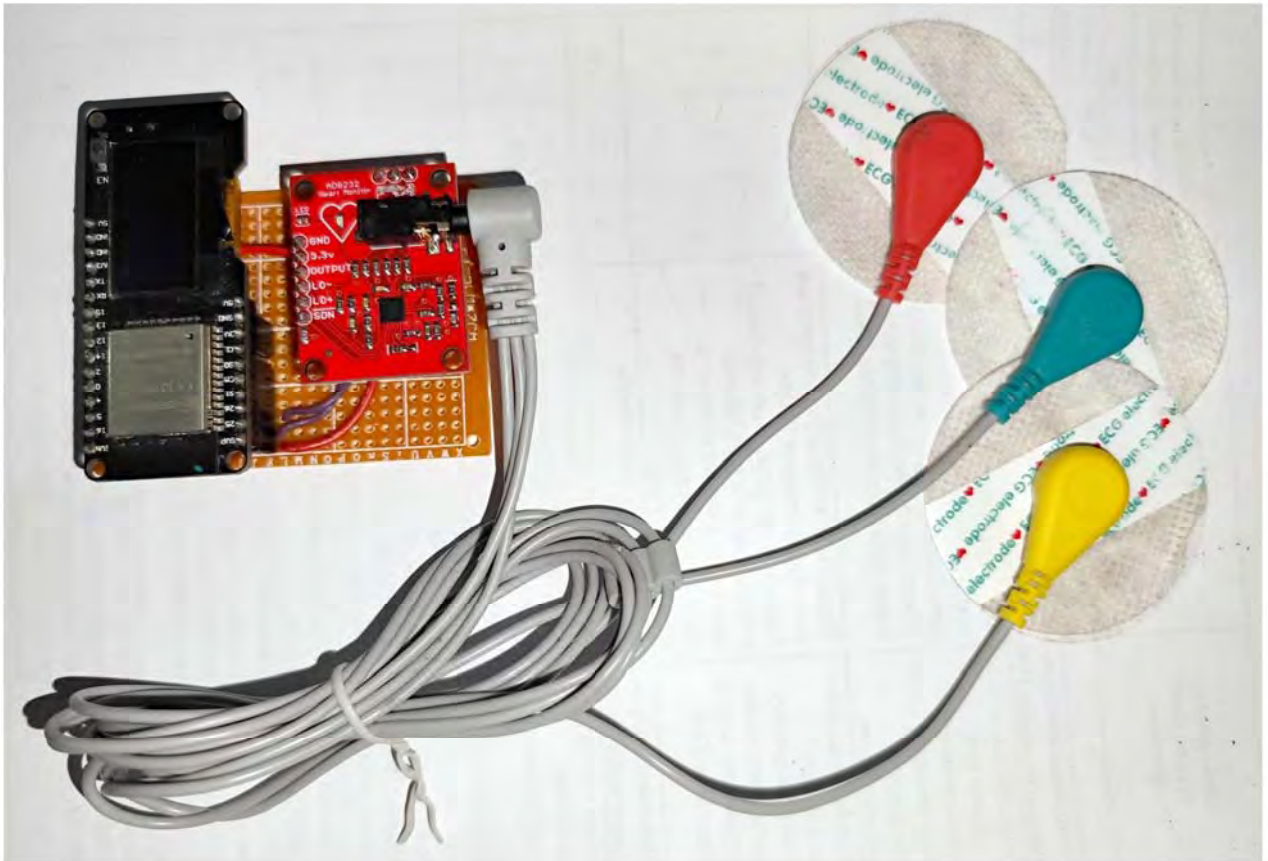


Рисунок 3.6 – Змонтований пристрій з під'єднаними електродами

### 3.3 Реалізація програмного коду

#### 3.3.1 Аналіз етапів роботи системи

Перш ніж приступити до реалізації програмного коду системи варто ще раз визначити основні етапи роботи системи:

##### 1) Управління мережевим з'єднанням

Система спочатку перевіряє стан WiFi підключення. Якщо з'єднання активне, відбувається відправка поточних показників у «хмару». При відсутності мережевого з'єднання цей етап пропускається без впливу на основну функціональність.

##### 2) Діагностика апаратного забезпечення

Виконується перевірка чи під'єднані до тіла контактні електроди. Система аналізує цифрові сигнали з двох контрольних входів, які індикують якість контакту правого та лівого електродів з тілом людини. Результат цієї перевірки визначає подальший сценарій роботи системи.

Зм.	Арк.	№ докум.	Підпис	Дата

КР.КІ 25.015.11.000 ПЗ

Арк.

27

### 3) Обробка аварійних ситуацій

У випадку від'єднання електродів від тіла людини всі змінні обнуляються, графік очищується, а на дисплеї з'являється діагностичне повідомлення. Це запобігає відображенню недостовірних даних та інформує користувача про необхідність усунення технічної проблеми.

### 4) Збір та первинна обробка біосигналу

При наявності якісного електричного контакту система зчитує аналоговий сигнал з електродів. Отримане значення проходить обробку для відображення на дисплеї та передачі через мережу у «хмару».

### 5) Аналіз серцевого ритму

Система виконує пошук характерних піків у біосигналі, порівнюючи поточне значення з встановленим порогом. При виявленні піка перевіряється часовий інтервал від попереднього для запобігання хибним спрацьовуванням. Кожен пік реєструється з точною міткою часу та додається до буфера останніх десяти подій. Коли накопичується достатня кількість піків, система обчислює середній інтервал між ними та конвертує результат у стандартні одиниці вимірювання ударів за хвилину (ЧСС).

### 6) Моніторинг критичних станів

Розрахована частота порівнюється з встановленим пороговим значенням. При перевищенні критичного рівня ЧСС активується система попереджень, яка включає звуковий сигнал та відповідне повідомлення на екрані.

### 3.3.2 Обґрунтування технологій та підходів до реалізації програмного коду

Завдяки поєднанню простоти розробки та високої продуктивності оптимальним рішенням для реалізації даного проекту була обрана платформа Arduino. Дана платформа надає готову екосистему з величезною кількістю бібліотек для роботи з периферійними пристроями, що значно прискорює розробку та зменшує ймовірність помилок. Особливо цінною є можливість прямого доступу до апаратних ресурсів через низькорівневі функції, що дозволяє точно контролювати часові характеристики системи. Водночас високорівневі

абстракції спрощують інтеграцію з мережевими сервісами та графічними компонентами. Arduino IDE забезпечує швидкий цикл розробки «код-компіляція-завантаження-тестування», що особливо важливо для прототипування та налагодження складних систем.

Для кращого структурування програмного коду було прийнято рішення реалізувати його через користувацькі функції, кожна з яких виконуватиме чіткі задачі. Такий підхід створює модульну архітектуру, де зміна алгоритму в одній частині системи не впливає на роботу інших компонентів. Модульність значно спрощує процес налагодження, оскільки кожен функцію можна тестувати незалежно від решти системи. Крім того, структурований код набагато легше розуміти і підтримувати, а також дозволяє повторно використовувати його в інших проєктах.

### 3.3.3 Опис користувацьких функцій

`connectToWiFiAndThinger` – управляє процесом встановлення мережевого з'єднання з автоматичним повторенням спроб та обробкою помилок. Забезпечує інтеграцію з хмарною платформою Thingier.io та підтримує резервний автономний режим роботи.

```
void connectToWiFiAndThinger() {
    // Додаємо WiFi налаштування до Thinger ПЕРЕД підключенням
    thing.add_wifi(SSID, SSID_PASSWORD);
    // Додаємо ресурси Thingier.io
    thing["bpm"] >> outputValue(bpm);
    thing["ecg_signal"] >> outputValue(currentEcgValue);
    thing["bpm_threshold"] << inputValue(bpmAlarmThreshold);
    const int maxAttempts = 3; // збільшуємо кількість спроб
    int attempt = 0;
    while (attempt < maxAttempts) {
        attempt++;
        Serial.print("WiFi connection attempt ");
        Serial.print(attempt);
        Serial.print("/");
        Serial.println(maxAttempts);
        // Відключаємо WiFi перед новою спробою
```

```

WiFi.disconnect();
delay(1000);
// Спроба підключення до WiFi
WiFi.begin(SSID, SSID_PASSWORD);
int wifiTimeout = 0;
while (WiFi.status() != WL_CONNECTED && wifiTimeout < 30) {
    delay(500);
    Serial.print(".");
    wifiTimeout++;
    // Показуємо прогрес на екрані
    if (wifiTimeout % 4 == 0) {
        display.setCursor(0, 30);
        display.print("Progress: ");
        display.print((wifiTimeout * 100) / 30);
        display.print("%   ");
        display.display();
    }
}
if (WiFi.status() == WL_CONNECTED) {
    wifiConnected = true;
    // Показуємо успішне підключення WiFi
    display.clearDisplay();
    display.setCursor(0, 0);
    display.print("WiFi OK");
    // Тестуємо підключення до Thingier.io
    bool thingerConnected = false;
    for (int i = 0; i < 10; i++) {
        thing.handle();
        delay(1000);
    }
    Serial.println("Connected to Thingier.io!");
    Serial.println("Starting ECG monitoring...");
    delay(2000);
    return; // успішне підключення
} else {
    // Показуємо причину помилки
    display.clearDisplay();
    display.setCursor(0, 0);
    display.print("WiFi attempt ");

```

```

    display.print(attempt);
    display.setCursor(0, 15);
    display.print("FAILED");
    display.setCursor(0, 30);
    display.print(errorMessage);
    display.display();
    Serial.println();
    Serial.print("WiFi connection attempt ");
    Serial.print(attempt);
    Serial.print(" failed");
    delay(3000);
}
}
// Всі спроби невдали
wifiConnected = false;
display.clearDisplay();
display.setCursor(0, 0);
display.print("WiFi connection");
display.setCursor(0, 15);
display.print("FAILED");
Serial.println("Starting ECG monitoring in offline mode");
delay(3000);
}

```

checkElectrodes – діагностує якість електричного контакту електродів з тілом через аналіз стану контрольних входів. Повертає логічний результат про готовність системи до збору біомедичних даних.

```

bool checkElectrodes() {
    bool loPlusStatus = digitalRead(loPlusPin); // HIGH = відключено
    bool loMinusStatus = digitalRead(loMinusPin); // HIGH =
відключено
    // Електроди підключені коли обидва сигнали LOW
    bool connected = (!loPlusStatus && !loMinusStatus);
    if (!connected) {
        Serial.println("Electrodes disconnected");
    }
    return connected;
}

```

checkBpmAlarm – моніторить критичні значення ЧСС та забезпечує інформування користувача через звукові та текстові повідомлення при виявленні критичних показників.

```
void checkBpmAlarm() {
    unsigned long now = millis();
    // Якщо пульс вище порогу і пройшло достатньо часу
    if (bpm >= bpmAlarmThreshold && bpm > 0 && (now - lastBuzzerTime > buzzerCooldown)) {
        // Подаємо звуковий сигнал
        tone(buzzerPin, buzzerTone, buzzerDuration);
        lastBuzzerTime = now;
        // Виводимо повідомлення в Serial Monitor
        Serial.print("HIGH BPM ALARM: ");
        Serial.print(bpm);
        Serial.println(" bpm");
    }
}
```

detectPeak – виконує аналіз вхідного сигналу для виявлення характерних піків серцебиття.

```
void detectPeak(int val) {
    unsigned long now = millis();
    if (val > threshold && !peakDetected && (now - lastPeakTime > minInterval)) {
        peakDetected = true;
        lastPeakTime = now;
        // зберігаємо мітку часу піка
        peakTimestamps[peakIndex] = now;
        peakIndex = (peakIndex + 1) % 10;
        if (peakIndex == 0) {
            // обчислення BPM за 10 піків
            long totalInterval = 0;
            for (int i = 1; i < 10; i++) {
                long interval = peakTimestamps[i] - peakTimestamps[i - 1];
                totalInterval += interval;
            }
            float avgInterval = totalInterval / 9.0;
            bpm = 60000.0 / avgInterval;
        }
    }
    if (val < threshold - 100) {
        peakDetected = false; // чекаємо падіння сигналу для нової детекції
    }
}
```

drawGraph – відповідає за формування повного графічного інтерфейсу на дисплеї. Малює безперервну лінію біосигналу, відображає числові показники ЧСС та показує попереджувальні повідомлення.

```
void drawGraph() {
    display.clearDisplay();
    // Перевіряємо стан електродів
    bool electrodesConnected = checkElectrodes();
    if (!electrodesConnected) {
        // Показуємо повідомлення про відключені електроди
        display.setTextSize(2);
        display.setCursor(0, 10);
        display.print("Check");
        display.setCursor(0, 30);
        display.print("electrodes!");
        display.setTextSize(1); // повертаємо розмір тексту
    } else {
        // Малюємо лінію графіка
        for (int x = 1; x < SCREEN_WIDTH; x++) {
            display.drawLine(x - 1, ecgGraph[x - 1], x, ecgGraph[x],
SSD1306_WHITE);
        }
        // Виводимо BPM
        display.setCursor(0, 0);
        display.print("BPM: ");
        display.print(bpm);
        // Показуємо попередження при високому пульсі
        if (bpm >= bpmAlarmThreshold && bpm > 0) {
            display.setCursor(70, 0);
            display.print("HIGH!");
        }
    }
    display.display();
}
```

ecgGraphShift – реалізує алгоритм графічного відображення сигналу. Збирає біосигнали в масив з одночасним масштабуванням діапазону вхідних даних до розмірів екрану.

```
void ecgGraphShift(int val) {
    for (int i = 0; i < SCREEN_WIDTH - 1; i++) {
        ecgGraph[i] = ecgGraph[i + 1];
    }
    ecgGraph[SCREEN_WIDTH - 1] = map(val, 0, 4095, SCREEN_HEIGHT-1,
0);
}
```

Повний лістинг програмного коду наведений в додатку Б.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

### 3.4 Підключення та налаштування сервісу thinger.io

#### 3.4.1 Опис сервісу thinger.io

Відповідно до поставленого завдання, розроблювана система має надавати можливість дистанційного контролю параметрів пульсу через мережу Інтернет. Це дасть змогу користувачеві отримувати інформацію про серцевий ритм, використовуючи веббраузер на будь-якому пристрої з Інтернет-підключенням. Для втілення цієї мети планується застосування хмарної платформи thinger.io.

Thingier.io являє собою IoT-платформу, яка пропонує засоби для інтеграції пристроїв у мережу Інтернет та забезпечує їх спостереження і контроль через вебінтерфейс або мобільні застосунки.

Ключові функції Thingier.io:

- інтеграція пристроїв: сервіс забезпечує простий спосіб підключення різних IoT-пристроїв за допомогою загальноприйнятих протоколів зв'язку;
- спостереження у реальному режимі: платформа дає можливість відстежувати інформацію з сенсорів та контролювати стан обладнання в режимі реального часу;
- дистанційне управління: забезпечує віддалений контроль над підключеними пристроями;
- аналіз даних та звітність: пропонує засоби для накопичення та обробки інформації, що сприяє прийняттю зважених рішень;
- сумісність з зовнішніми сервісами: легко поєднується з іншими онлайн-платформами та програмами для розширення можливостей.

Thingier.io полегшує розробку IoT-рішень, пропонуючи готову технічну базу для з'єднання, спостереження та керування пристроями, що дозволяє заощадити час і ресурси програмістів.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

### 3.4.2 Реєстрація в сервісі Thinger.io

Реєстрація облікового запису в сервісі Thinger.io відбувається швидко та легко. Спочатку потрібно перейти на офіційний сайт [thinger.io](http://thinger.io), після чого вказати активну електронну пошту, створити унікальне ім'я користувача та встановити пароль.

Для даного проєкту нам знадобиться створити один пристрій. Для цього потрібно перейти на сторінку Devices, натиснути кнопку Create a Device, заповнити необхідні параметри та натиснути кнопку Add Device (рис. 3.7).

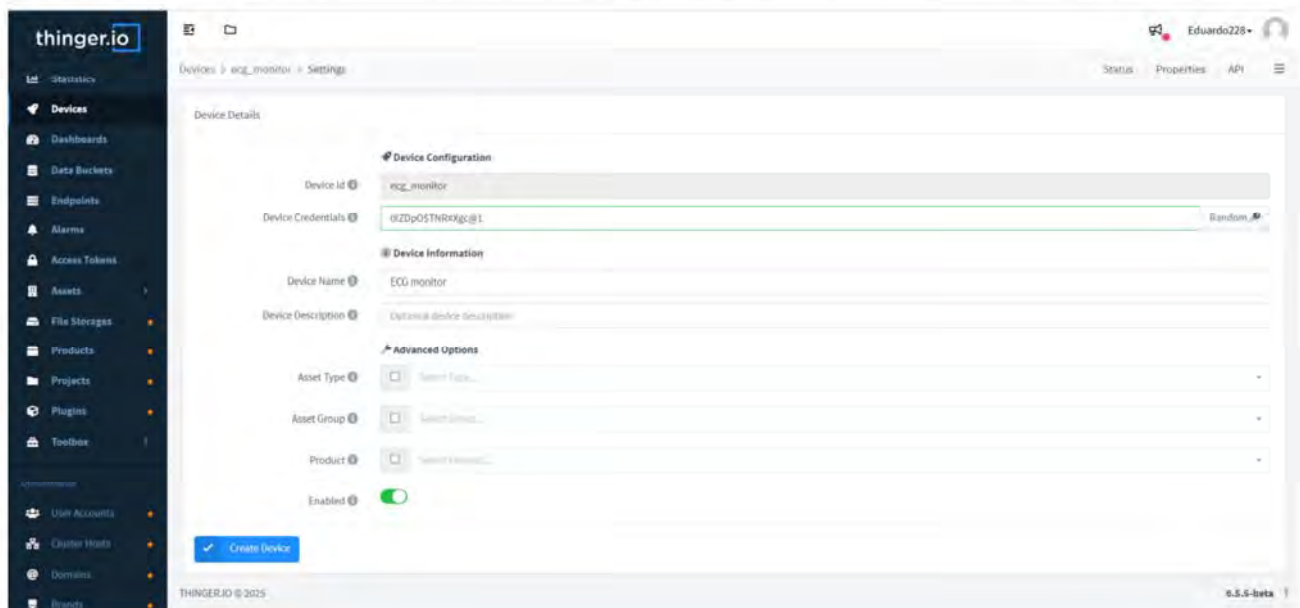


Рисунок 3.7 – Створення пристрою в Thinger.io

Використовуючи отримані дані Device Id та Device Credentials разом з іменем користувача, налаштуємо програмний код системи для інтеграції з платформою Thinger.io.

```
// Thinger.io налаштування
#define USERNAME "Eduardo228"
#define DEVICE_ID "ecg_monitor"
#define DEVICE_CREDENTIAL "0IZDpO$TNR#Xgc@1"
```

В функції `connectToWiFiAndThinger()` додамо два ресурси (Resources) для використання в хмарному сервісі: `bpm` – для передачі поточного значення ЧСС, та `ecg_signal` – для передачі поточного значення ЕКГ-сигналу.

```
thing["bpm"] >> outputValue(bpm);
thing["ecg_signal"] >> outputValue(currentEcgValue);
```

В функції loop() реалізуємо циклічне надсилення оновлених значень цих ресурсів при підключенні до thinger.io:

```
if (wifiConnected) {  
  thing.handle();  
}
```

Після внесення змін у програмний код і завантаження його в мікроконтролер необхідно з'єднати пристрій з WiFi мережею та протестувати його функціонування через платформу Thingier.io. За умови коректного виконання всіх дій, новостворений пристрій має з'явитися із зеленим індикатором, що підтверджує успішне встановлення з'єднання (рис. 3.8).



Рисунок 3.8 – Пристрій активний після успішного підключення

### 3.4.3 Створення та налаштування панелі керування в сервісі Thingier.io

Тепер переходимо до створення панелі керування (Dashboard). На сторінці Dashboards натискаємо кнопку Create Dashboard, заповнюємо необхідні параметри та натискаємо кнопку Add Dashboard (рис. 3.9).

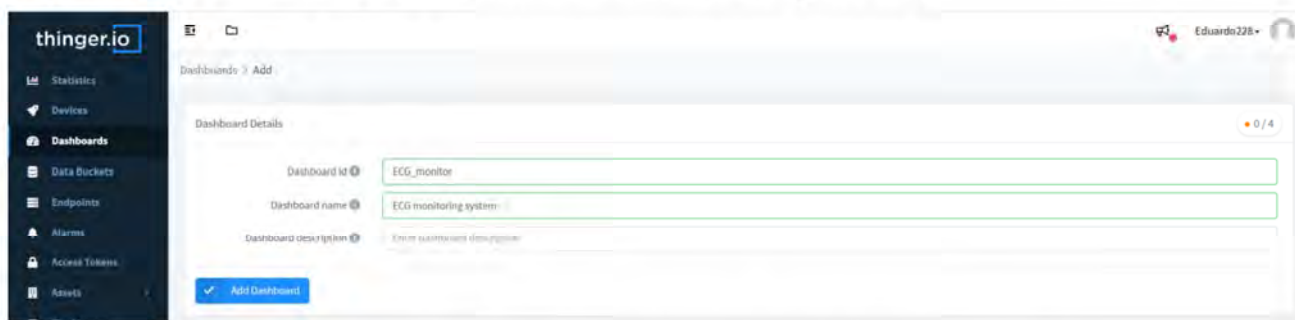


Рисунок 3.9 – Створення нової панелі керування

Після цього потрібно наповнити панель керування відповідними віджетами. Додамо на 2 віджети:

1. Віджет типу Gauge, який відобразить поточне значення ЧСС.
2. Віджет типу Time Series Chart, який відобразить графік значень ЕКГ.

Вигляд панелі керування після встановлення обох віджетів представлений на рисунку 3.12.

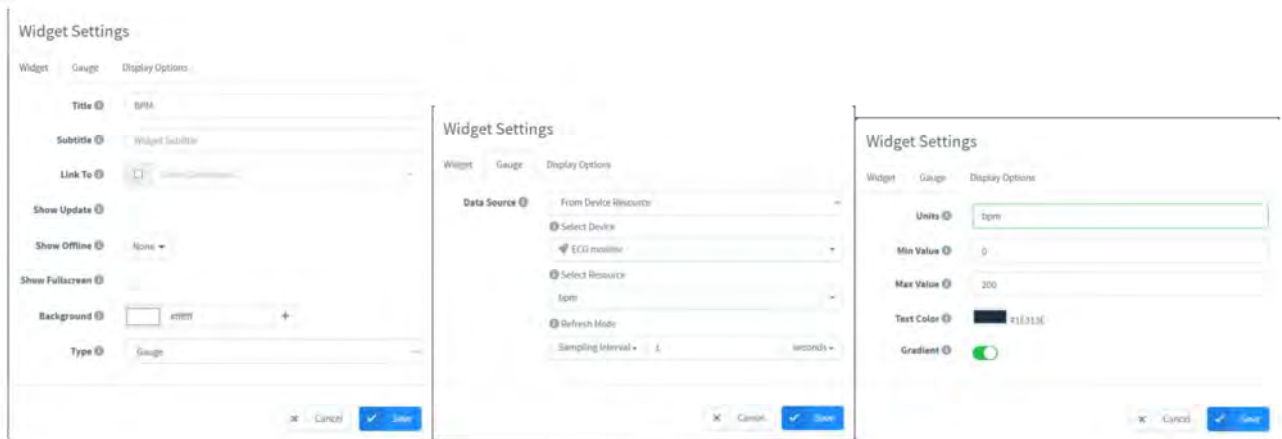


Рисунок 3.10 – Налаштування віджета ВРМ для відображення ЧСС

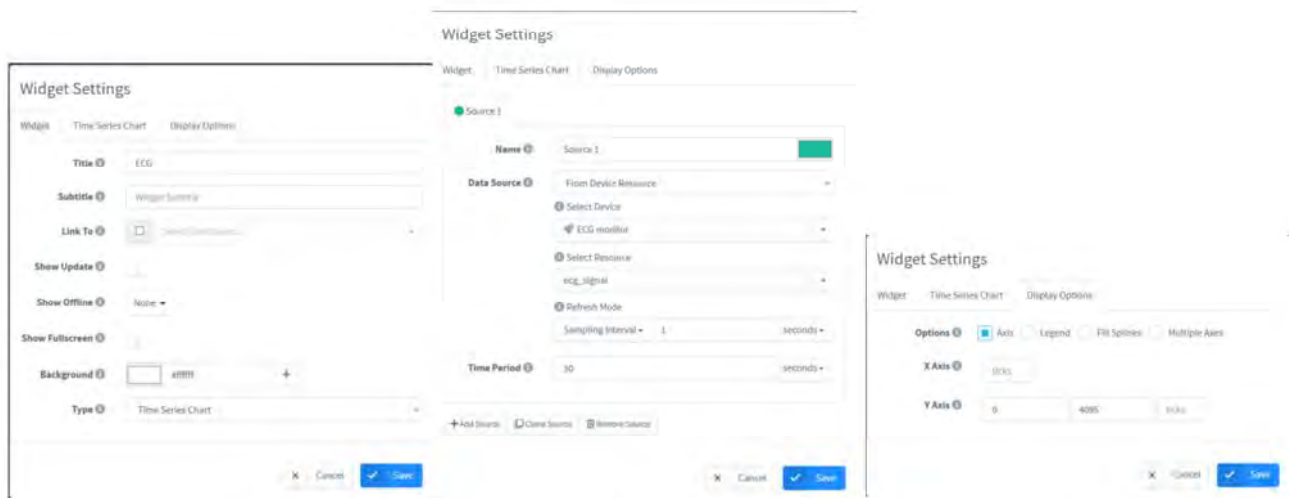


Рисунок 3.11 – Налаштування віджета ЕКГ-сигналу



Рисунок 3.12 – Панель керування після встановлення двох віджетів

### 3.5 Тестування роботи системи

Для тестування роботи системи складемо короткий план тестування та перевіримо роботу системи відповідно до нього.

План тестування:

1. Перевірити підключення пристрою до мережі WiFi.
2. Перевірити підключення пристрою до сервісу Thinger.io.
3. Перевірити роботу системи при від'єднаній електродах.
4. Перевірити роботу системи при нормальному пульсі (менше 120 ударів серця за секунду).
5. Перевірити роботу системи при підвищеному пульсі (понад 120 ударів серця за секунду).
6. Перевірити відображення результатів вимірювань в Thinger.io.

Результати тестування відображені в таблиці 3.1 та на рисунках 3.13-3.16.

Таблиця 3.1 – Результати тестування роботи системи

№	Дія	Очікуваний результат	Фактичний результат
1.	Підключення пристрою до мережі WiFi	Пристрій підключився до WiFi. На екрані вивелося відповідне сповіщення.	Пристрій підключився до WiFi. На екрані вивелося відповідне сповіщення.
2.	Підключення пристрою до сервісу Thinger.io	Пристрій підключився до сервісу Thinger.io. На екрані вивелося відповідне сповіщення.	Пристрій підключився до сервісу Thinger.io. На екрані вивелося відповідне сповіщення.
3.	Робота пристрою при від'єднанні електродів	На екрані виводиться повідомлення «Check electrodes». В thinger.io дані не оновлюються.	На екрані виводиться повідомлення «Check electrodes». В thinger.io дані не оновлюються.
4.	Робота пристрою при нормальному пульсі	На екрані виводиться поточне значення ЧСС (BPM) та малюється кардіограма. В thinger.io відображається поточне значення ЧСС та малюється кардіограма.	На екрані виводиться поточне значення ЧСС (BPM) та малюється кардіограма. В thinger.io відображається поточне значення ЧСС та малюється кардіограма.
5.	Робота пристрою при підвищеному пульсі (понад 120 ударів за секунду)	На екрані виводиться поточне значення ЧСС (BPM) та малюється кардіограма. Періодично виводиться звукову сповіщення. В thinger.io відображається поточне значення ЧСС та малюється кардіограма.	На екрані виводиться поточне значення ЧСС (BPM) та малюється кардіограма. Періодично виводиться звукову сповіщення. В thinger.io відображається поточне значення ЧСС та малюється кардіограма.
6.	Робота пристрою при відсутньому підключенні до WiFi	Після трьох невдалих спроб підключення на екрані виводиться повідомлення «WiFi connection FAILED. Starting offline» і починається відображення показників ЧСС та малюється кардіограма. В thinger.io дані не оновлюються.	Після трьох невдалих спроб підключення на екрані виводиться повідомлення «WiFi connection FAILED. Starting offline» і починається відображення показників ЧСС та малюється кардіограма. В thinger.io дані не оновлюються.

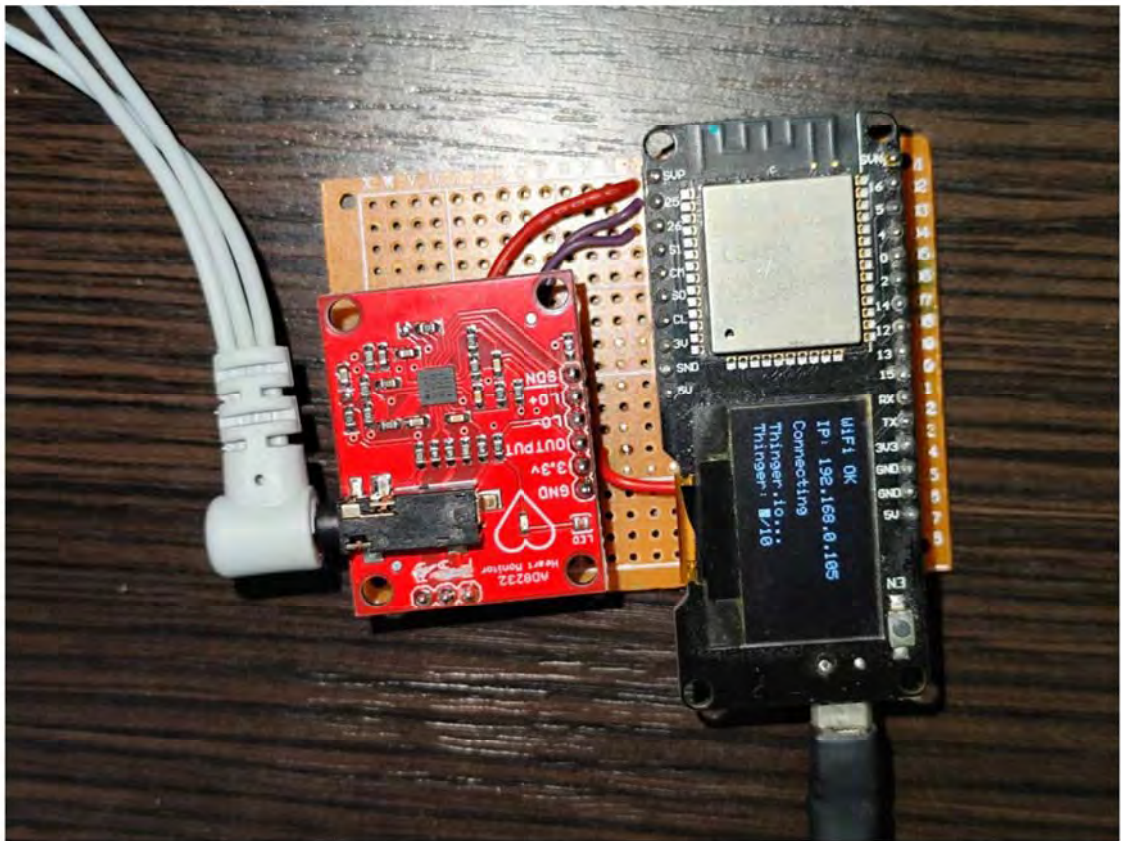


Рисунок 3.13 – Підключення пристрою до Thingier.io

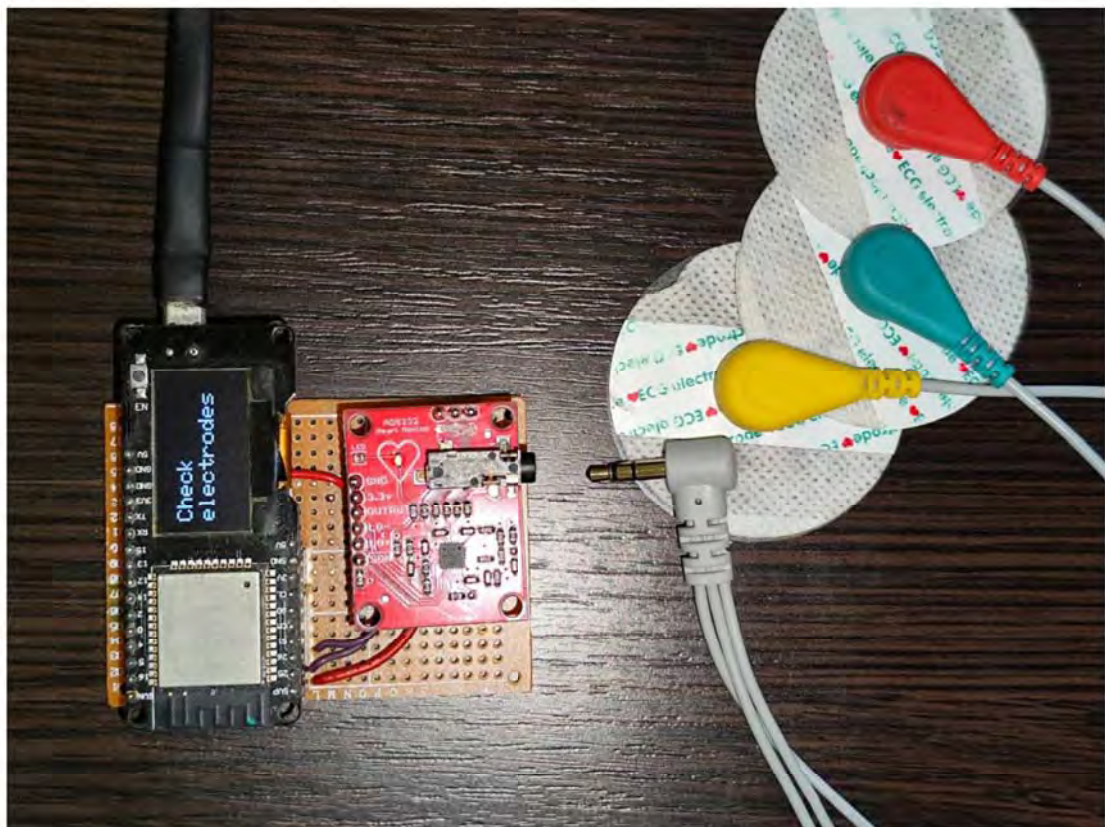


Рисунок 3.14 – Робота пристрою при від'єднаних електродах

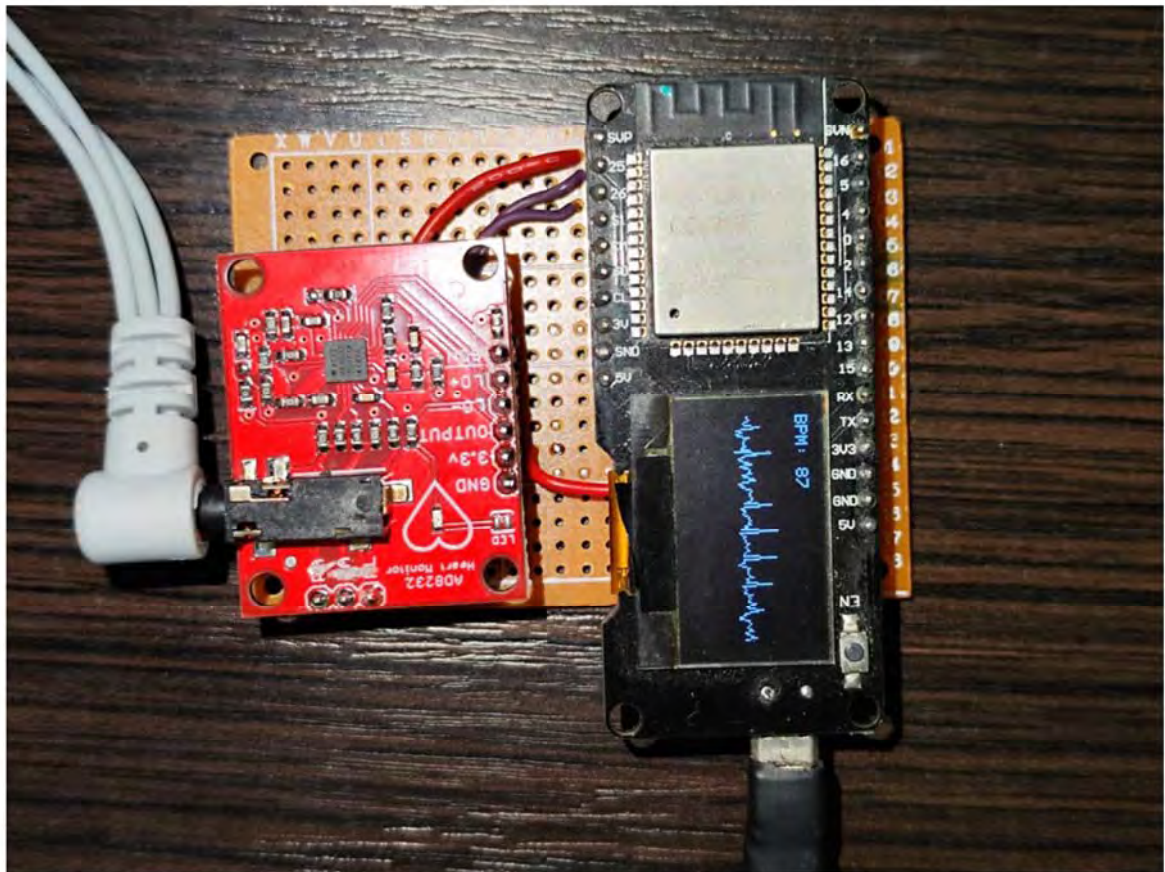


Рисунок 3.15 – Робота пристрою при нормальному пульсі

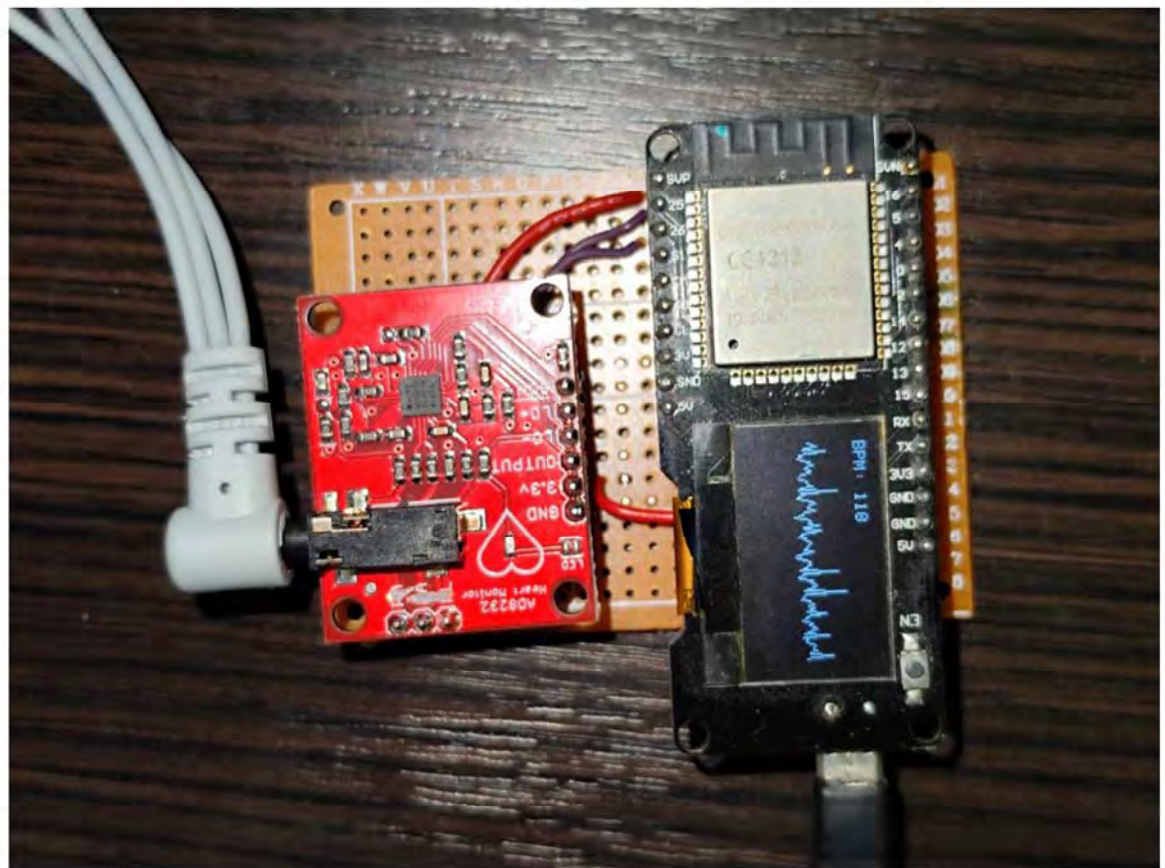


Рисунок 3.16 – Робота пристрою при підвищеному пульсі

Зм.	Арк.	№ докум.	Підпис	Дата

КР.КІ 25.015.11.000 ПЗ

Арк.

41

Хоча розроблена система ЕКГ моніторингу і демонструє заявлені функціональні можливості, однак в процес тестування було виявлено кілька суттєвих обмежень, які впливають на її практичне застосування.

Найбільш критичним недоліком є непридатність для динамічного моніторингу. Традиційні електроди, які використовуються в системі, створюють значні незручності під час фізичної активності. Для спортсменів, які потребують постійного контролю серцевого ритму під час тренувань, такі електроди стають серйозною перешкодою – при тривалому використанні вони відклеюються та зміщуються при рухах.

Висока чутливість до зовнішніх завад є ще однією проблемою, яка значно знижує надійність системи. Навіть незначне погіршення контакту електродів призводить до появи артефактів і хибних показників, що може спричинити помилкові тривоги або, навпаки, пропуск справжніх аномалій.

Технічні обмеження платформи Thinger.io створюють додаткові проблеми з візуалізацією. Обмеження частоти оновлення даних до одного разу на секунду кардинально спотворює графік ЕКГ, роблячи неможливим якісний аналіз форми хвилі. Для медичного моніторингу, де важливі навіть найменші деталі сигналу, така частота дискретизації є недостатньою.

Враховуючи вищезазначене, у подальшому доцільно розглянути кілька напрямків оптимізації системи. По-перше, варто дослідити можливість заміни поточних електродів на більш практичні аналоги, які забезпечуватимуть стабільніше з'єднання та комфортніше використання під час тренувань.

По-друге, необхідно модифікувати програмний код для врахування можливих артефактів, що виникають під час вимірювань, та мінімізувати їх вплив на загальні показники. Це дозволить підвищити точність отриманих даних та знизити кількість хибних спрацьовувань.

По-третє, доцільно відмовитися від побудови графіка ЕКГ у платформі Thinger.io на користь більш функціонального рішення. Замість цього рекомендується інтегрувати систему автоматичних сповіщень на мобільний

телефон або електронну пошту, яка спрацюватиме при виявленні підвищених показників частоти серцевих скорочень під час тренувань. Такий підхід забезпечить оперативне реагування на критичні зміни стану організму спортсмена.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		43

## 4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ПРОЄКТУ

### 4.1 Аналіз ринку

Сучасний ринок пристроїв для моніторингу фізичних показників спортсменів характеризується широким спектром рішень різного рівня складності та вартості. Аналіз існуючих продуктів показує значну диференціацію за ціновими категоріями та функціональними можливостями.

У сегменті професійного медичного обладнання домінують високоточні системи, такі як медичний електрокардіограф BTL-08 LC, вартість якого становить 3000-4000 євро. Такі пристрої забезпечують найвищу точність діагностики та широкий функціонал, однак їх висока вартість робить їх недоступними для більшості індивідуальних користувачів та невеликих спортивних організацій.

Середній ціновий сегмент представлений спеціалізованими спортивними датчиками, зокрема нагрудним датчиком серцевого ритму Polar H10, ціна якого починається від 100 євро. Це обладнання орієнтоване на серйозних спортсменів та тренерів, пропонуючи компроміс між точністю вимірювань та доступністю.

Бюджетний сегмент включає портативні пристрої, такі як ЕКГ-монітор AliveCor KardiaMobile, вартістю понад 100 доларів США. Такі рішення надають базові можливості моніторингу, але часто обмежені в функціональності та точності.

Аналіз ринку показує існування значної ніші між професійним медичним обладнанням та споживчими пристроями. Запропонована система відстеження фізичних показників спортсмена займає унікальну позицію, пропонуючи можливості моніторингу за доступною ціною.

### 4.2 Розрахунок витрат на реалізацію

Економічна ефективність розробленої системи забезпечується використанням сучасних мікроконтролерних платформ та доступних електронних компонентів. Детальний розрахунок витрат включає дві основні

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

категорії: витрати на апаратні компоненти та витрати на оплату праці розробників.

#### 4.2.1 Витрати на апаратні компоненти

- плата розробки ESP32 Wemos LoLin32 з OLED-екраном – орієнтовна вартість 400 грн.;
- модуль електрокардіограми AD8232 – орієнтовна вартість 350 грн.;
- активний буюер для звукових сповіщень – орієнтовна вартість 20 грн.

Таким чином загальна вартість компонентів системи становить приблизно 770 грн.

#### 3.2.2 Розрахунок витрат на оплату праці

Для реалізації проекту оптимальним є залучення інженера-програміста, який може виконувати функції проєктувальника, розробника програмного забезпечення та налаштування системи. Середня місячна заробітна плата такого спеціаліста становить 30 000 грн.

Розрахунок податкових відрахувань:

- Податок на доходи фізичних осіб (18%):  $30\,000 \times 0,18 = 5\,400$  грн.
- Військовий збір (5%):  $30\,000 \times 0,05 = 1\,500$  грн.
- Єдиний соціальний внесок (22%):  $30\,000 \times 0,22 = 6\,600$  грн.
- Загальні відрахування з зарплати працівника:  $5\,400 + 1\,500 = 6\,900$  грн.
- Чиста зарплата працівника:  $30\,000 - 6\,900 = 23\,100$  грн.

Загальна вартість праці (включаючи всі відрахування): 30 000 грн

Загальна вартість реалізації системи: 30 000 грн (оплата праці) + 770 грн (компоненти) = 30 770 грн

#### 4.3 Обґрунтування доцільності розробки

Економічна доцільність розробки системи відстеження фізичних показників спортсмена обґрунтовується кількома ключовими факторами, що впливають з аналізу ринку та розрахунку витрат на реалізацію.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

Важливо відзначити, що розрахована вартість 30 770 грн відображає витрати на створення прототипу, а реальна економічна доцільність виготовлення проявляється при переході до масового виробництва.

При масовому виробництві структура витрат кардинально змінюється. Витрати на оплату праці розробників, які становлять левову частку вартості прототипу (97,5%), розподіляються на всю партію виробів, що значно знижує їх частку в собівартості одиниці продукції. Наприклад, при виробництві партії з 100 одиниць витрати на розробку становитимуть лише 300 грн на один пристрій.

Головним компонентом ціни при масовому виробництві стає собівартість апаратних компонентів. Оптові закупівлі електронних компонентів дозволяють отримати значні знижки порівняно з роздрібними цінами. Зокрема, при замовленні партій від 100 одиниць можна очікувати зниження вартості компонентів на 30-50%. Це означає, що собівартість апаратної частини може зменшитися з 770 грн до 400-500 грн за одиницю.

Навіть з урахуванням додаткових витрат на виробництво, упаковку та логістику, собівартість одного пристрою при масовому виробництві не перевищуватиме 800-1000 грн. При ринковій ціні 3000-5000 грн система забезпечуватиме рентабельність 70-80%, що є надзвичайно привабливим показником. Це дозволить пропонувати продукт за ціною в 6-10 разів нижчою за професійні аналоги.

Використання сучасної мікроконтролерної платформи ESP32 забезпечує можливість подальшого розвитку системи без кардинальної зміни апаратної основи. Відкрита архітектура дозволяє легко додавати нові датчики, покращувати алгоритми обробки даних та розширювати функціональність системи.

Аналіз ринку показує існування значного попиту на доступні професійні рішення моніторингу фізичних показників. Цільова аудиторія включає індивідуальних спортсменів, невеликі спортивні клуби, фітнес-тренери та медичні установи з обмеженим бюджетом. Низька собівартість при масовому

					<i>КР.КІ 25.015.11.000 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		46

виробництві робить систему доступною для надзвичайно широкого кола користувачів.

Таким чином, техніко-економічний аналіз підтверджує доцільність розробки системи відстеження фізичних показників спортсмена. Хоча створення прототипу потребує значних початкових інвестицій, перехід до масового виробництва забезпечує винятково привабливі економічні показники та широкі можливості комерціалізації продукту.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

## ВИСНОВКИ

У даній роботі було досліджено, спроектовано, реалізовано та протестовано систему відстеження фізичних показників спортсмена під час тренування. В процесі аналізу предметної області було визначено, що контроль частоти серцевих скорочень є критично важливим аспектом у спортивній медицині, фітнесі та загальному моніторингу здоров'я. Метою роботи було визначено створення автоматизованої системи, яка здатна в режимі реального часу відстежувати показники ЧСС та забезпечувати оперативне реагування на критичні зміни.

На етапі проектування було розроблено алгоритм роботи системи та її структурну схему, що включає модуль обробки біомедичних даних, дисплей для локального відображення інформації та мікроконтролер ESP32 для обробки даних і зв'язку з хмарним сервісом Thingier.io. Було обрано апаратні компоненти та побудовано електричну схему системи. Система була спроектована таким чином, щоб надати можливість віддаленого моніторингу показників через Інтернет.

На етапі реалізації було розроблено програмне забезпечення для мікроконтролера ESP32, яке включає алгоритми розрахунку ЧСС. Інтеграція з платформою Thingier.io дозволила забезпечити віддалений моніторинг та візуалізацію даних у режимі реального часу.

Однак у процесі тестування було виявлено ряд технічних обмежень як самого пристрою, так і платформи Thingier.io. Найбільш критичними недоліками виявилися непридатність традиційних електродів для динамічного моніторингу під час фізичної активності, висока чутливість системи до зовнішніх завад та технічні обмеження платформи Thingier.io щодо частоти оновлення даних. Ці недоліки обмежують практичне застосування системи в умовах підвищеної фізичної активності та вимагають додаткового доопрацювання.

Економічний аналіз показав, що розробка та впровадження системи відстеження фізичних показників спортсмена є економічно доцільною з точки

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

зору вартості компонентів порівняно з комерційними аналогами. Використання доступних апаратних компонентів та відкритих платформ дозволить значно знизити витрати на розробку при збереженні базової функціональності. Особливо економічний ефект проявиться при масовому виробництві даного пристрою. Однак для масового виробництва необхідно попередньо усунути виявлені технічні недоліки та врахувати запропоновані кроки для подальшого вдосконалення системи.

					<i>КР.КІ 25.015.11.000 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		49

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Polar H10 – нагрудний пульсометр для бігу, плавання, велотренувань // Polar Ukraine. – 2023. – 20 листоп. URL: <https://www.polar-ukraine.com/shop/h10/> (дата звернення: 29.11.2024).

2. ЕКГ апарат BTL-08 LC [Електронний ресурс] // Меділаб. URL: <https://medilab.com.ua/uk/btl-08-lc> (дата звернення: 29.11.2024).

3. Методичні рекомендації до виконання кваліфікаційної роботи для студентів спеціальності 123 «Комп’ютерна інженерія» освітньої програми «Інженерія Інтернету речей» / Павлюс В.П., Посвятовська О.Б., Кульчинська Н.З. – Галицький фаховий коледж імені В’ячеслава Чорновола, Тернопіль, 2023. 52с.

4. Мікроконтролер ESP32: веб-сайт. URL: <https://uk.wikipedia.org/wiki/ESP32> (дата звернення: 15.03.2025).

5. Основи електрокардіографії.. Compendium. <https://compendium.com.ua/uk/clinical-guidelines-uk/cardiology-uk/section-5-uk/glava-1-osnovi-elektrokardiografiji/> (дата звернення: 28.11.2024).

6. Портативний електрокардіограф AliveCor KardiaMobile 6L. URL: <https://prom.ua/ua/p1550828868-portativnij-elektrokardiograf-alivecor.html> (дата звернення: 29.11.2024).

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

# ДОДАТКИ

## Додаток А

### Блок-схема алгоритму роботи системи



Зм.	Арк.	№ докум.	Підпис	Дата

## Додаток Б

### Лістинг програмного коду

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <ThingerESP32.h>
#include <WiFi.h>

// Thinger.io налаштування
#define USERNAME "Eduardo228"
#define DEVICE_ID "ecg_monitor"
#define DEVICE_CREDENTIAL "0IZDpO$TNR#Xgc@1"

// WiFi налаштування
#define SSID "edik_wifi"
#define SSID_PASSWORD "big_secret"

ThingerESP32 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
#define LCD_SDA 5 // I2C SDA
#define LCD_SCL 4 // I2C SCL

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

const int ecgPin = 36; // GPIO36 (ADC1_CH0)
const int buzzerPin = 13; // GPIO13 для бузера
const int loPlusPin = 25; // GPIO25 для LO+ (правий електрод)
const int loMinusPin = 26; // GPIO26 для LO- (лівий електрод)
const int threshold = 2100; // поріг для пікової детекції (підібрати
вручну)
const int minInterval = 300; // мінімальний інтервал між піками (мс)
// Налаштування звукових сигналів
int bpmAlarmThreshold = 120; // поріг ЧСС для звукового сигналу
const int buzzerTone = 1000; // частота звуку (Гц)
const int buzzerDuration = 200; // тривалість сигналу (мс)
uint8_t ecgGraph[SCREEN_WIDTH]; // масив для зберігання сигналу
unsigned long lastPeakTime = 0;
int bpm = 0;
unsigned long peakTimestamps[10]; // масив останніх 10 піків
int peakIndex = 0;
bool peakDetected = false;
```

```

unsigned long lastBuzzerTime = 0; // час останнього звукового сигналу
const int buzzerCooldown = 5000; // пауза між звуковими сигналами
// Змінні для Thingier.io
int currentEcgValue = 0; // поточне значення ЕКГ сигналу
bool wifiConnected = false; // статус WiFi підключення

void setup() {
  Serial.begin(115200);
  // Підключення до WiFi та Thingier.io
  thing.add_wifi(SSID, SSID_PASSWORD);
  // Налаштування ресурсів Thingier.io
  thing["bpm"] >> outputValue(bpm);
  thing["ecg_signal"] >> outputValue(currentEcgValue);
  // Ресурс для отримання налаштувань порогу BPM з dashboard
  thing["bpm_threshold"] << inputValue(bpmAlarmThreshold);
  // Ініціалізація екрану
  Wire.begin(LCD_SDA, LCD_SCL);
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    while (true);
  }
  display.clearDisplay();
  display.setTextColor(SSD1306_WHITE);
  display.setTextSize(1);
  pinMode(ecgPin, INPUT);
  pinMode(buzzerPin, OUTPUT); // налаштування бузера як вихід
  pinMode(loPlusPin, INPUT); // налаштування LO+ як вхід
  pinMode(loMinusPin, INPUT); // налаштування LO- як вхід
}

void loop() {
  if (wifiConnected) {
    thing.handle(); // обробка Thingier.io тільки при підключенні
  }
  // Перевіряємо стан електродів
  bool electrodesConnected = checkElectrodes();
  if (electrodesConnected) {
    int raw = analogRead(ecgPin);
    currentEcgValue = raw; // оновлюємо для Thingier.io
  }
}

```

```

    ecgGraphShift(raw);
    detectPeak(raw);
    checkBpmAlarm(); // перевірка на високий пульс
} else {
    // При відключених електродах обнулюємо дані
    bpm = 0;
    peakIndex = 0;
    currentEcgValue = 0;
}
drawGraph();
delay(20); // ~50 Гц частота вибірки
}
// Зміщує масив вліво та додає нове значення
void ecgGraphShift(int val) {
    for (int i = 0; i < SCREEN_WIDTH - 1; i++) {
        ecgGraph[i] = ecgGraph[i + 1];
    }
    // масштабування значення до екрану (0-4095 -> 0-63)
    ecgGraph[SCREEN_WIDTH - 1] = map(val, 0, 4095, SCREEN_HEIGHT - 1, 0);
}
// Виводить графік та BPM на екран
void drawGraph() {
    display.clearDisplay();
    // Перевіряємо стан електродів
    bool electrodesConnected = checkElectrodes();
    if (!electrodesConnected) {
        // Показуємо повідомлення про відключені електроди
        display.setTextSize(2);
        display.setCursor(0, 10);
        display.print("Check");
        display.setCursor(0, 30);
        display.print("electrodes!");
        display.setTextSize(1); // повертаємо розмір тексту
    } else {
        // Малюємо лінію графіка
        for (int x = 1; x < SCREEN_WIDTH; x++) {
            display.drawLine(x - 1, ecgGraph[x - 1], x, ecgGraph[x],
SSD1306_WHITE);
        }
    }
}

```

```

// Виводимо BPM
display.setCursor(0, 0);
display.print("BPM: ");
display.print(bpm);
// Показуємо попередження при високому пульсі
if (bpm >= bpmAlarmThreshold && bpm > 0) {
    display.setCursor(70, 0);
    display.print("HIGH!");
}
}
display.display();
}
// Детекція піків та обчислення BPM
void detectPeak(int val) {
    unsigned long now = millis();
    if (val > threshold && !peakDetected && (now - lastPeakTime >
minInterval)) {
        peakDetected = true;
        lastPeakTime = now;
        // зберігаємо мітку часу піка
        peakTimestamps[peakIndex] = now;
        peakIndex = (peakIndex + 1) % 10;
        if (peakIndex == 0) {
            // обчислення BPM за 10 піків
            long totalInterval = 0;
            for (int i = 1; i < 10; i++) {
                long interval = peakTimestamps[i] - peakTimestamps[i - 1];
                totalInterval += interval;
            }
            float avgInterval = totalInterval / 9.0;
            bpm = 60000.0 / avgInterval;
        }
    }
    if (val < threshold - 100) {
        peakDetected = false; // чекаємо падіння сигналу }
    }
// Перевірка на високий пульс та подача звукового сигналу
void checkBpmAlarm() {
    unsigned long now = millis();

```

```

// Якщо пульс вище порогу і пройшло достатньо часу
if (bpm >= bpmAlarmThreshold && bpm > 0 && (now - lastBuzzerTime >
buzzerCooldown)) {
    // Подаємо звуковий сигнал
    tone(buzzerPin, buzzerTone, buzzerDuration);
    lastBuzzerTime = now;
    // Виводимо повідомлення в Serial Monitor
    Serial.print("HIGH BPM ALARM: ");
    Serial.print(bpm);
    Serial.println(" bpm");
}
}

// Перевірка стану електродів
bool checkElectrodes() {
    bool loPlusStatus = digitalRead(loPlusPin); // HIGH = відключено
    bool loMinusStatus = digitalRead(loMinusPin); // HIGH = відключено
    // Електроди підключені коли обидва сигнали LOW
    bool connected = (!loPlusStatus && !loMinusStatus);
    if (!connected) {
        Serial.println("Electrodes disconnected!");
    }
    return connected;
}

// Підключення до WiFi та Thingier.io
void connectToWiFiAndThingier() {
    // Додаємо WiFi налаштування до Thingier ПЕРЕД підключенням
    thing.add_wifi(SSID, SSID_PASSWORD);
    // Додаємо ресурси Thingier.io
    thing["bpm"] >> outputValue(bpm);
    thing["ecg_signal"] >> outputValue(currentEcgValue);
    thing["bpm_threshold"] << inputValue(bpmAlarmThreshold);
    const int maxAttempts = 3; // збільшуємо кількість спроб
    int attempt = 0;
    while (attempt < maxAttempts) {
        attempt++;
        // Показуємо спробу підключення
        display.clearDisplay();
        display.setCursor(0, 0);
    }
}

```

```

display.print("WiFi attempt ");
display.print(attempt);
display.print("/");
display.print(maxAttempts);
display.setCursor(0, 15);
display.print("Connecting...");
display.display();
Serial.print("WiFi connection attempt ");
Serial.print(attempt);
Serial.print("/");
Serial.println(maxAttempts);
// Відключаємо WiFi перед новою спробою
WiFi.disconnect();
delay(1000);
// Спроба підключення до WiFi
WiFi.begin(SSID, SSID_PASSWORD);
int wifiTimeout = 0;
while (WiFi.status() != WL_CONNECTED && wifiTimeout < 30) {
    delay(500);
    Serial.print(".");
    wifiTimeout++;
    // Показуємо прогрес на екрані
    if (wifiTimeout % 4 == 0) {
        display.setCursor(0, 30);
        display.print("Progress: ");
        display.print((wifiTimeout * 100) / 30);
        display.print("%  ");
        display.display();
    }
}
if (WiFi.status() == WL_CONNECTED) {
    wifiConnected = true;
    // Показуємо успішне підключення WiFi
    display.clearDisplay();
    display.setCursor(0, 0);
    display.print("WiFi OK");
    display.setCursor(0, 15);
    display.print("IP: ");
    String ip = WiFi.localIP().toString();

```

```

if (ip.length() > 15) ip = ip.substring(0, 15);
display.print(ip);
display.setCursor(0, 30);
display.print("Connecting");
display.setCursor(0, 45);
display.print("Thingier.io...");
display.display();
Serial.println();
Serial.print("WiFi connected! IP: ");
Serial.println(WiFi.localIP());
Serial.print("Signal strength: ");
Serial.print(WiFi.RSSI());
Serial.println(" dBm");
Serial.println("Connecting to Thingier.io...");
// Тестуємо підключення до Thingier.io
bool thingierConnected = false;
for (int i = 0; i < 10; i++) {
    thing.handle();
    delay(1000);
    // Показуємо прогрес підключення до Thingier
    display.setCursor(0, 55);
    display.print("Thingier: ");
    display.print(i + 1);
    display.print("/10");
    display.display();
    // Перевіряємо, чи є інтернет-з'єднання
    if (i == 5) {
        Serial.println("Testing internet connection...");
        // Можна додати ping тест тут
    }
}
// Показуємо результат
display.clearDisplay();
display.setCursor(0, 0);
display.print("WiFi Connected");
display.setCursor(0, 15);
display.print("Thingier.io OK");
display.setCursor(0, 35);
display.print("Starting ECG");

```

```

display.setCursor(0, 50);
display.print("monitoring...");
display.display();
Serial.println("Connected to Thingier.io!");
Serial.println("Starting ECG monitoring...");
delay(2000);
return; // успішне підключення
} else {
// Показуємо причину помилки
String errorMsg = "";
switch (WiFi.status()) {
case WL_NO_SSID_AVAIL:
errorMsg = "SSID not found";
break;
case WL_CONNECT_FAILED:
errorMsg = "Wrong password";
break;
case WL_CONNECTION_LOST:
errorMsg = "Connection lost";
break;
case WL_DISCONNECTED:
errorMsg = "Disconnected";
break;
default:
errorMsg = "Unknown error";
break;
}
display.clearDisplay();
display.setCursor(0, 0);
display.print("WiFi attempt ");
display.print(attempt);
display.setCursor(0, 15);
display.print("FAILED");
display.setCursor(0, 30);
display.print(errorMsg);
display.display();
Serial.println();
Serial.print("WiFi connection attempt ");
Serial.print(attempt);

```

```

    Serial.print(" failed: ");
    Serial.println(errorMsg);
    delay(3000);
  }
}
// Всі спроби невдали
wifiConnected = false;
display.clearDisplay();
display.setCursor(0, 0);
display.print("WiFi connection");
display.setCursor(0, 15);
display.print("FAILED");
display.setCursor(0, 30);
display.print("Check network");
display.setCursor(0, 45);
display.print("Starting offline");
display.display();
Serial.println("All WiFi connection attempts failed");
Serial.println("Starting ECG monitoring in offline mode");
delay(3000);
}

```

					<i>КР.КІ 25.015.11.000 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60