

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-апаратний засіб для вимірювання серцевого ритму та аналізу отриманих даних на основі ESP32 з підтримкою бездротового зв'язку
Назва теми

КВРКІ 210243.21.02.58 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»
Назва

Виконав: студент IV курсу, група KI2-21-2


Підпис

Богдан МАЗУР
Ініціали, прізвище

Керівник


Підпис, дата

Богдан САВЕНКО
Ініціали, прізвище

Нормоконтролер


Підпис, дата

Тетяна КИСІЛЬ
Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Підпис

Ольга ПАВЛОВА
Ініціали, прізвище

«2» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Мазуру БОГДАНУ Володимировичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-апаратний засіб для вимірювання серцевого ритму та аналізу отриманих даних на основі ESP32 з підтримкою бездротового зв'язку

Керівник проекту (роботи) Богдан Савенко, д.ф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Аналіз програмно-апаратних засобів кіберфізичних систем та постановка задач її розробки та функціонування

Проектування програмно-апаратного засобу для вимірювання серцевого ритму

Практична частина розробки програмно-апаратного засобу

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Блок-схема роботи ПЗ

Схема з'єднання компонентів ПЗ

Монтажна схема

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КІС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КІС		

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2025	виконано
4	Робота над розділом 2 – вибір компонентів для проектування програмно-апаратного засобу для вимірювання серцевого ритму	01.04.2025	виконано
5	Робота над розділом 3 – проектування програмно-апаратного засобу для вимірювання серцевого ритму	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Підпис

Богдан МАЗУР
Ініціали, прізвище

Керівник роботи

Підпис

Богдан САВЕНКО
Ініціали, прізвище

№ р я д к а	Ф о р м а т	Позначення	Найменування	К і л - л и с т і в	№ ек з	П р и м і т к а
			<u>Текстові документи</u>			
1		КвРКІ 210243.21.02.58 ПЗ	Пояснювальна записка	60		
			<u>Графічні матеріали</u>			
2		КвРКІ 210243.21.02.58 Е1	Блок-схема роботи ПЗ	1		
3		КвРКІ 210243.21.02.58 Е2	Схема з'єднання компонентів ПЗ	1		
4		КвРКІ 190186.19.02.58 Е3	Монтажна схема	1		

КвРКІ 210243.21.02.58 ВП

Зм	Арк	№ докум	Підпис	Дата	Відомість проекту	Літера	Аркуш	Аркушів
Розробив		Мазур		02.06.25		У	1	1
Перевір.		Савенко		02.06.25		ХНУ, КІ2-21-2		
Н. контр.		Кисіль		02.06.25				
Затв.		Павлова		02.06.25				

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-апаратний засіб для вимірювання серцевого ритму та аналізу отриманих даних на основі ESP32 з підтримкою бездротового зв'язку».

Автор роботи: Богдан МАЗУР.

Керівник роботи: Савенко Богдан Олегович.

Пояснювальна записка: 60 с., 33 рис., 1 табл., 3 дод., 43 джерел.

Графічна частина: 3 креслення.

ПРОГРАМНО-АПАРАТНИЙ ЗАСІБ, АРХІТЕКТУРА, МОНІТОРИНГ, СЕНСОРИ, БЕЗДРОТОВИЙ ЗВ'ЯЗОК.

Метою даної статті є розробка програмно-апаратного засобу для вимірювання частоти серцевих скорочень та дослідження методів аналізу отриманих біомедичних даних за допомогою мікроконтролера ESP32 з підтримкою бездротового зв'язку. Основна увага приділяється забезпеченню надійності збору даних, їх попередньої обробки та подальшої передачі в режимі реального часу для моніторингу стану користувача.

Об'єктом дослідження є функціонування програмно-апаратного засобу для вимірювання серцевого ритму.

Предметом дослідження є методи збору, обробки та передачі біомедичних даних за допомогою мікроконтролерів ESP32.

Під час проведення даного дослідження був проведений метод систематичного огляду літератури для аналізу предметної області принципів роботи датчиків серцевого ритму, сучасних методів обробки та передачі біомедичної інформації в бездротових мережах.






Підпис студента

30.05.2025

Дата

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ПРОГРАМНО-АПАРАТНИХ ЗАСОБІВ КІБЕРФІЗИЧНИХ СИСТЕМ ТА ПОСТАНОВКА ЗАДАЧ ЇЇ РОЗРОБКИ ТА ФУНКЦІОНУВАННЯ	6
1.1 Аналіз структурних і функціональних особливостей програмного-забезпечення	6
1.2 Аналіз апаратних засобів отримання фізичної інформації про стан здоров'я людини	10
1.3 Аналіз програмно-апаратного забезпечення обробки інформації в кіберфізичній системі	16
1.4 Висновки до першого розділу	17
2 ПРОЕКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ ДЛЯ ВИМІРЮВАННЯ СЕРЦЕВОГО РИТМУ	18
2.1 Розробка структури програмно-апаратного засобу вимірювання серцевого ритму	18
2.2 Апаратне забезпечення програмно-апаратного засобу вимірювання серцевого ритму	19
2.3 Програмне забезпечення програмно-апаратного засобу вимірювання серцевого ритму	27
2.4 Опис принципової схеми пристрою	33
2.5 Висновки до другого розділу	36
3 ПРАКТИЧНА ЧАСТИНА РОЗРОБКИ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ	38
3.1 Опис реалізації модулів апаратного забезпечення програмно-апаратного засобу	38
3.2 Налаштування середовища розробки ПЗ	43
3.2.1 Опис середовища розробки ПЗ	43
3.2.2 Налаштування Arduino IDE для роботи з модулем ESP32	43

КвРКІ 210243.21.02.58 ПЗ								
Зм.	Арк.	№ док.ум.	Підпис	Дата	Програмно-апаратний засіб для вимірювання серцевого ритму	Літера	Аркуш	Аркушів
Виконав		Богдан МАЗУР		20.06.25		у		
Перевір.		Богдан САВЕНКО		22.06.25			2	72
Н.контр.		Тетяна КИСЛЬ		22.06.25		ХНУ КІ2-21-2		
Затвер.		Ольга ПАВЛОВА		22.06.25				

3.2.3 Встановлення бібліотек	46
3.3 Візуалізація даних за допомогою MQTT брокера	47
3.4 Візуальний зворотній зв'язок апаратно-програмного засобу.....	53
3.5 Висновки до третього розділу.....	58
ВИСНОВКИ	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	61
ДОДАТОК А	66
ДОДАТОК Б	67
ДОДАТОК В	68

ВСТУП

Моніторинг фізіологічних параметрів людини, наприклад серцевого ритму, стає більш важливим значенням в сучасному житті. Він допомагає людині у багатьох сферах, таких як охорона здоров'я, спорт та фітнес, а також самому стежити за показниками свого здоров'я. Технологічний прогрес уможливив широке використання портативних пристроїв для вимірювання частоти серцевого ритму в повсякденному житті без необхідності традиційних клінічних методів.

В наш час поширено, що люди стали більше турбуватися про своє здоров'я і тому з'явився великий попит на пристрої для відстежування свого здоров'я. Традиційні методи моніторингу пульсу, які використовуються медичними установами, складні та дорогі, тому перехід від клінічного до портативного моніторингу серцевого ритму відображає демократизацію даних про здоров'я, що дає людям можливість більш активно керувати та слідкувати за своїм здоров'ям. З появою портативних комерційних пристроїв ця можливість стала доступною для більш широкого кола людей, для яких моніторинг свого серцевого ритму став не заміним.

Об'єктом дослідження виступають процеси збирання та обробки інформації про фізіологічний стан людини, їх аналіз та подача користувачу в зручній формі. Предметом дослідження виступає апаратно-програмний комплекс, що містить засоби отримання інформації, її обробку та передачі даних до користувача каналами зв'язку. Методи дослідження використовують інженерні технології розробки систем на базі мікроконтролерів, проектування інтерфейсу та електронних схем. Для аналізу точності вимірювання датчиків буде застосовано ряд тестів у реальних умовах роботи.

Метою дипломної роботи є розробка портативного моніторингу серцевого ритму з використанням мікроконтролера ESP32, сенсором для точного вимірювання серцевого ритму та надійними протоколами бездротового зв'язку. Для цього я проаналізую наявні на ринку сенсори та зробивши аналіз виберу

					КВРКІ 210243.21.02.58 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

найкращий з них, і розроблю алгоритм аналізу серцевого ритму, для виявлення проблем зі здоров'ям.

Практичне значення роботи полягає у розробці та впровадження ефективної системи отримання та аналізу даних засобами мікроелектроніки, що активно розвивається та потребує створення нових алгоритмів обробки інформації про здоров'я людини в умовах реального часу яка буде корисна в медичних установах та в домашніх умовах. Створена система має потенціал розвитку для подальшої інтеграції з медичними пристроями та системами, що здійснюють моніторинг стану здоров'я людини.

					КВРКІ 210243.21.02.58 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

1 АНАЛІЗ ПРОГРАМНО-АПАРАТНИХ ЗАСОБІВ КІБЕРФІЗИЧНИХ СИСТЕМ ТА ПОСТАНОВКА ЗАДАЧ ЇЇ РОЗРОБКИ ТА ФУНКЦІОНУВАННЯ

1.1 Аналіз структурних і функціональних особливостей програмно-забезпечення

Предметна область моніторингу серцевого ритму є надзвичайно актуальною у сучасному світі. Вимірювання та аналіз серцевого ритму відіграють важливу роль у різних сферах, включаючи клінічну медицину для діагностики та лікування серцево-судинних захворювань, а також у сфері фітнесу для оптимізації тренувань та підвищення загальної обізнаності про стан здоров'я.

В сфері медицині пристрої для вимірювання серцевого ритму відіграють надзвичайно важливу роль у сучасній медицині, надаючи цінну інформацію для діагностики та лікування різноманітних серцево-судинних захворювань. Монітори серцевого ритму, такі як кардіологічні монітори подій, є важливим інструментом для фіксації нечастих симптомів та виявлення аритмій, які можуть не проявлятися під час звичайного лікарського огляду.

Ці пристрої здатні відстежувати ритм серця, силу електричних сигналів, їхню часову послідовність та частоту скорочень. Аналогія з автомобілем, який видає тривожний звук лише час від часу, влучно ілюструє потребу в тривалому моніторингу в медицині. Саме такий підхід дозволяє зафіксувати тимчасові серцеві події, які можуть залишитися непоміченими під час рутинних обстежень.

Регулярний кардіологічний моніторинг є невід'ємною частиною профілактики та лікування серцево-судинних захворювань, що є провідною причиною смерті у світі. Раннє виявлення, контроль прогресування захворювання та оцінка ефективності лікування є ключовими перевагами такого моніторингу. Моніторинг серцевого ритму використовується для оцінки ефективності лікування, зокрема медикаментів, призначених для корекції тахікардії або брадикардії, а також для перевірки функціонування імплантованих пристроїв, таких як кардіостимулятори та імплантовані кардіовертери-дефібрилятори.

					КВРКІ 210243.21.02.58 ПЗ	Арк. 6
Зм.	Арк.	№ докум.	Підпис	Дата		

Безперервне вимірювання серцевого ритму може надавати ранні попередження про потенційні проблеми зі здоров'ям та допомагати людям краще розуміти поведінку свого серця під час щоденної діяльності та сну. Ця тенденція до інтеграції моніторингу серцевого ритму в повсякденне життя для загального контролю стану здоров'я, а не лише для конкретних медичних епізодів або фізичних вправ, стає все більш поширеною. Постійний збір даних дозволяє виявляти ледь помітні зміни та закономірності серцевого ритму, які можуть свідчити про приховані проблеми зі здоров'ям або вплив способу життя на серцево-судинну систему [1-5].

У сфері спорту пристрої для вимірювання серцевого ритму широко використовуються для визначення інтенсивності вправ, оптимізації тренувань та запобігання перенапруженню. Тренування в персоналізованих зонах серцевого ритму допомагає спортсменам досягати конкретних цілей, таких як спалювання жиру, розвиток витривалості або оптимізація продуктивності. Персоналізація тренувального процесу в спорті вказує на те, що моніторинг серцевого ритму дозволяє спортсменам адаптувати свої тренування до індивідуальних фізіологічних реакцій та рівня фізичної підготовки. Перебуваючи в межах певних діапазонів серцевого ритму, спортсмени можуть бути впевнені, що вони працюють з потрібною інтенсивністю для досягнення бажаних результатів тренувань без ризику травм або вигорання.

Дані про серцевий ритм у реальному часі дозволяють тренерам контролювати рівень навантаження спортсменів та приймати обґрунтовані рішення щодо коригування тренувань та потреб у відновленні. Застосування моніторингу серцевого ритму поширюється й на командні види спорту, де тренери можуть використовувати ці дані для управління тренувальним навантаженням багатьох спортсменів одночасно.

Об'єктивні дані про серцевий ритм під час тренувальних сесій надають цінний зворотний зв'язок щодо фізіологічної реакції кожного спортсмена, що дозволяє тренерам оптимізувати плани тренувань для всієї команди та окремих

					КВРКІ 210243.21.02.58 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

гравців. Безперервний моніторинг серцевого ритму допомагає спортсменам відстежувати свій прогрес з часом, оскільки більш ефективно серце б'ється рідше для досягнення тієї ж інтенсивності тренувань. Це підкреслює використання серцевого ритму як довготривалого показника покращення фізичної форми. У міру того як спортсмен стає фізично підготовленим, його серцевий ритм у спокої має тенденцію до зниження, а реакція серцевого ритму на фізичні вправи стає ефективнішою, що відображає покращення серцево-судинного здоров'я.

Частота серцевих скорочень у відновлювальному періоді (ЧССВ), тобто швидкість, з якою серцевий ритм знижується після вправ, є показником рівня фізичної підготовки та може надавати інформацію про ризик серцевих подій. Моніторинг серцевого ритму не лише під час вправ, але й у фазі відновлення надає додаткову цінну інформацію про серцево-судинне здоров'я та фізичну форму. Швидше повернення до серцевого ритму в спокої після вправ зазвичай вказує на кращу серцево-судинну підготовку та нижчий ризик певних серцевих захворювань. Варіабельність серцевого ритму (ВСР) все частіше використовується в спорті для контролю тренувального стресу, адаптації, відновлення та навіть для прогнозування результатів. Зростання інтересу до ВСР у спорті свідчить про перехід до більш глибокого розуміння спортивних результатів, яке враховує роль автономної нервової системи. ВСР може надавати інформацію про фізіологічну готовність спортсмена до тренувань та змагань, допомагаючи оптимізувати тренувальні навантаження та запобігати перетренованості [6-8].

У контексті фітнесу та велнесу моніторинг серцевого ритму допомагає людям займатися спортом у межах своїх цільових зон серцевого ритму, щоб максимізувати користь та забезпечити безпеку. Це дозволяє оптимізувати витрати калорій та покращити серцево-судинне здоров'я. Концепція цільових зон серцевого ритму робить вправи більш орієнтованими на результат та ефективними для людей з різними фітнес-цілями. Розуміючи свої цільові діапазони серцевого ритму для різних рівнів інтенсивності активності, люди можуть адаптувати свої тренування для спалювання більшої кількості жиру, покращення витривалості або підвищення

					КВРКІ 210243.21.02.58 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

серцево-судинної підготовки. Безперервний моніторинг серцевого ритму може допомогти людям відстежувати рівень щоденної активності, стрес та якість сну, що сприяє загальному добробуту. Інтеграція моніторингу серцевого ритму в щоденні фітнес-трекери та смарт-годинники робить його доступним для ширшого кола людей для проактивного управління здоров'ям. Постійний збір даних та аналіз можуть надати інформацію про різні аспекти їхнього добробуту, що дозволяє людям приймати більш обґрунтовані рішення щодо здорового способу життя [9-11].

Моніторинг серцевого ритму в спокої може надати інформацію про загальний стан серця та рівень фізичної підготовки. Нижчий серцевий ритм у спокої зазвичай вказує на краще серцево-судинне здоров'я. Серцевий ритм у спокої є простим, але потужним показником для оцінки базового рівня серцево-судинної підготовки та відстеження змін з часом. У міру того як люди стають більш фізично активними та покращують своє серцево-судинне здоров'я, їхній серцевий ритм у спокої має тенденцію до зниження.

Носимі технології з моніторингом серцевого ритму можуть підвищити мотивацію до фізичних вправ, забезпечуючи постійний зворотний зв'язок та відстеження прогресу. Функції гейміфікації та відстеження у фітнес-браслетах можуть значно підвищити залученість користувачів та їхню прихильність до тренувальних програм. Відображення даних у реальному часі та відстеження прогресу у досягненні фітнес-цілей може мотивувати людей залишатися активними та підтримувати здорові звички. Аналіз варіабельності серцевого ритму (ВСР) у контексті фітнесу та велнесу може надати інформацію про рівень стресу, відновлення та загальний добробут. ВСР стає стандартною функцією багатьох фітнес-трекерів, що відображає її зростаюче визнання як цінного показника для розуміння реакції організму на різні стресори. Відстежуючи ВСР, люди можуть краще усвідомлювати свій рівень стресу, контролювати своє відновлення після фізичної активності та приймати обґрунтовані рішення для покращення якості сну та психічного здоров'я [12-16].

					КВРКІ 210243.21.02.58 ПЗ	Арк. 9
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз апаратних засобів отримання фізичної інформації про стан здоров'я людини

У предметній області розробки пристроїв для вимірювання серцевого ритму існує широкий спектр програмно-апаратних рішень. До апаратних засобів належать різноманітні датчики серцевого ритму, такі як фотоплетизмографічні (PPG) датчики (наприклад, MAX30102, KY-039), електрокардіографічні (ECG) датчики (наприклад, AD8232), а також комбіновані датчики, що вимірюють кілька параметрів (наприклад, частоту серцевих скорочень та рівень кисню в крові). Як мікроконтролери часто використовуються ESP32 та Arduino для розробки вбудованих систем.

Порівняння датчиків:

MAX30102: Інтегрований датчик пульсоксиметрії та серцевого ритму, інтерфейс I2C, низьке енергоспоживання, вимірює SpO₂. MAX30102 працює за принципом фотоплетизмографії (ФПГ), оптичного методу, що вимірює пульсуючі зміни об'єму крові. Датчик випромінює червоне світло з довжиною хвилі 660 нм та інфрачервоне світло з довжиною хвилі 880 нм у тканину, після чого фотодетектор вимірює кількість відбитого світла. Різниця в поглинанні світла цих двох довжин хвиль оксигенованим та дезоксигенованим гемоглобіном дозволяє обчислити як частоту серцевих скорочень, так і рівень насичення крові киснем (SpO₂). Можливість MAX30102 вимірювати як частоту серцевих скорочень, так і SpO₂ в одному компактному модулі робить його універсальним варіантом для комплексного моніторингу здоров'я. Така інтеграція зменшує потребу у використанні кількох датчиків, що спрощує конструкцію апаратного забезпечення. Датчик оснащений вбудованим буфером FIFO, який може зберігати до 32 зразків SpO₂ та частоти серцевих скорочень, що зменшує навантаження на мікроконтролер [17].

					КВРКІ 210243.21.02.58 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

КУ-039 (рис. 1.1): Простий датчик серцебиття, що використовує інфрачервоний світлодіод та фототранзистор, аналоговий вихід. КУ-039 використовує інфрачервоний (ІЧ) світлодіод для пропускання світла через кінчик пальця або мочку вуха та фототранзистор для виявлення кількості світла, яке пройшло або відбилося. Зміни об'єму крові під час кожного серцебиття впливають на кількість світла, що потрапляє на фототранзистор, що призводить до пульсуючого аналогового вихідного сигналу. Працює за простішим принципом порівняно з МАХ30102, покладаючись на базове пропускання/відбиття світла для виявлення серцебиття. Він забезпечує прямий аналоговий сигнал, який потребує подальшої обробки [18].

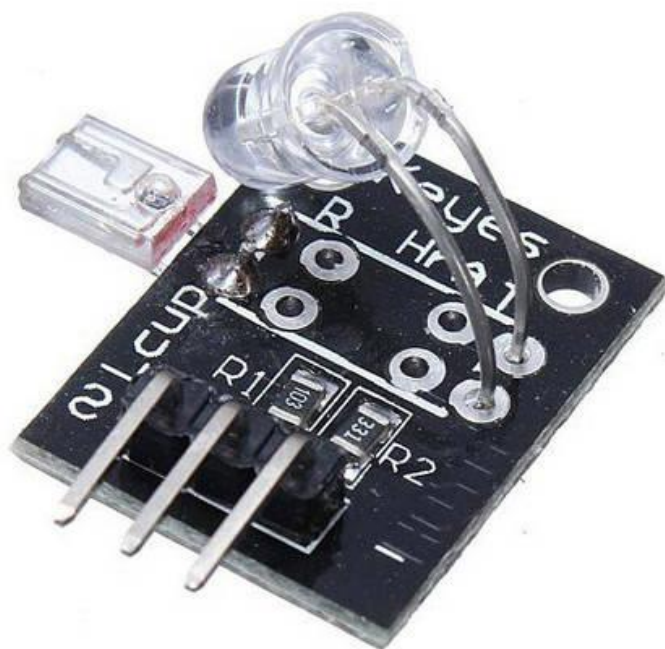


Рисунок 1.1 – Зовнішній вигляд КУ-039 [42]

AD8232: Одноканальний монітор серцевого ритму, розробленим для носіння на тілі з метою моніторингу здоров'я. Він використовує електрокардіографію (ЕКГ) для вимірювання електричної активності серця. AD8232 використовує електрокардіографію (ЕКГ) для вимірювання електричної активності серця за допомогою електродів, розміщених на шкірі (зазвичай у одноканальній

конфігурації). AD8232 є інтегрованим блоком кондиціонування сигналу, який витягує, підсилює та фільтрує ці невеликі біопотенціальні сигнали, навіть за наявності шуму від руху або розміщення електродів. Він діє як операційний підсилювач для отримання чистого сигналу ЕКГ, виділяючи інтервали PR та QT. Забезпечує більш пряме та детальне вимірювання серцевої діяльності порівняно з датчиками на основі ФПГ, фіксуючи електричні сигнали, що генеруються серцем. Його інтегрована схема обробляє посилення та фільтрацію, спрощуючи вимоги до обробки сигналу на ESP32 [19].

Вибір датчика залежить від конкретних вимог проєкту. MAX30102 пропонує більше функціональності, включаючи вимірювання SpO₂, але вимагає інтерфейсу I2C та має дещо вищу складність підключення та програмування порівняно з KY-039. KY-039 є найпростішим та найменш дорогим варіантом, але надає лише аналоговий імпульсний сигнал серцебиття, точність якого може бути нижчою. AD8232 забезпечує найдетальнішу інформацію про серцеву діяльність завдяки вимірюванню ЕКГ, що дозволяє проводити більш глибокий аналіз, але вимагає правильного розміщення електродів та обробки сигналу ЕКГ.

Також розглянемо типи комерційних пристроїв для вимірювання серцевого ритму, які відрізняються за принципом дії, точністю, зручністю використання та сферами застосування:

1. Нагрудні ремені: Використовують електроди для вимірювання електричної активності серця (ЕКГ). ЕКГ є фундаментальним методом дослідження серцевої діяльності, що вимірює біопотенціали, які контролюють скорочення серцевих камер. Ці пристрої мають щонайменше два електроди, які відчують електричну активність серця. Електроди, часто сухі, виготовляються з еластичного та провідного матеріалу і підключаються до невеликої електронної схеми. ЕКГ-монітори серцевого ритму вимірюють електричний потенціал на поверхні шкіри, який може бути збережений як повна ЕКГ-крива або оброблений для виявлення серцевих скорочень, зокрема комплексу QRS. Для більшості нагрудних ременів необхідний хороший контакт зі шкірою, який іноді

забезпечується використанням провідного гелю або змочуванням датчиків водою для поліпшення електричної провідності. Сухі електроди можуть спотворювати дані, тому для їх зволоження використовується водопровідна вода або спеціальні розчини. Після виявлення серцевого скорочення радіосигнал передається на приймач. Сучасні нагрудні ремені використовують Bluetooth або ANT+ для передачі даних на смартфони та інші розумні пристрої. Пряме вимірювання електричної активності серця робить нагрудні ремені за своєю суттю точнішими за пристрої, що покладаються на непрямі вимірювання, такі як кровотік. Для надійних показань необхідний хороший контакт електродів зі шкірою та вологість для забезпечення провідності. Ці пристрої безпосередньо вимірюють частоту серцевих скорочень, на відміну від оптичних пристроїв, які вимірюють пульс. Нагрудні ремені надійні та менш схильні до впливу артефактів руху порівняно з оптичними датчиками. Вони ідеально підходять для високоінтенсивних тренувань та видів спорту зі значними рухами рук, таких як біг, їзда на велосипеді та веслування. Деякі моделі можуть надавати додаткові дані, такі як варіабельність серцевого ритму (ВСР) та динаміка бігу [20-23].

2. Наручні пристрої та смарт-кільця: (рис. 1.2): Наручні пристрої та смарт-кільця використовують технологію фотоплетизмографії (ФПГ) для вимірювання серцевого ритму. Ця технологія передбачає використання світлодіодів (зазвичай зелених або інфрачервоних) для просвічування шкіри та фотодетекторів для вимірювання відбитого світла. ФПГ виявляє зміни об'єму крові в мікросудинному руслі тканини з кожним серцевим скороченням. Ці зміни перетворюються на вимірювання серцевого ритму (частоти пульсу). Смарт-кільця працюють за тим самим принципом ФПГ, часто використовуючи інфрачервоні світлодіоди та стратегічно розміщені датчики на пальці. Пальці мають більш розвинену мережу кровоносних судин, розташованих ближче до поверхні шкіри, що потенційно може призвести до точніших показань. Обидва типи пристроїв покладаються на ФПГ, оптичний метод, що вимірює зміни кровотоку для оцінки серцевого ритму. Розміщення смарт-кілець на пальці може забезпечити кращу точність завдяки

щільності судин та близькості датчиків. Точність ФПГ-пристроїв залежить від рівня активності: зазвичай хороша у стані спокою, але знижується під час фізичних вправ через рухи. Деякі бренди, такі як Apple Watch, показують кращі результати, ніж інші. Смарт-кільця мають більше проблем з точністю під час фізичних вправ порівняно з наручними пристроями [24-28].



Рисунок 1.2 – Смарт-кільце Samsung Galaxy Ring [43]

3. Пульсоксиметри: Зазвичай використовуються в медичних установах для вимірювання частоти пульсу та рівня кисню в крові за допомогою оптичного методу. Точність вимірювання серцевого ритму за допомогою пульсоксиметрів, як правило, є хорошою у стані спокою та під час низькоінтенсивної активності. Однак точність може знижуватися під час інтенсивних вправ через нестабільність датчика, артефакти руху та потовиділення. На точність показань SpO₂ та серцевого ритму можуть впливати такі фактори, як поганий кровообіг, пігментація шкіри, її товщина, температура, а також лак для нігтів. Пульсоксиметри можуть недооцінювати частоту серцевих скорочень при високих значеннях (понад 155 ударів на хвилину) і зазвичай не підходять для безперервного моніторингу серцевого ритму під час фізичних вправ. Хоча пульсоксиметри надають показник серцевого ритму, вони в першу чергу призначені для вимірювання насичення крові

					КВРКІ 210243.21.02.58 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

киснем. Їхня точність для серцевого ритму може бути обмежена, особливо під час фізичної активності та в осіб з певними фізіологічними особливостями [29].

4. Мобільні додатки: Деякі програми для смартфонів використовують камеру та спалах телефону для вимірювання серцевого ритму за допомогою оптичного методу. Точність мобільних додатків для вимірювання серцевого ритму значно варіюється між різними програмами, навіть тими, що використовують одну й ту саму технологію. Контактні додатки ФПГ (палець на камері), як правило, точніші за безконтактні додатки ФПГ (камера, спрямована на обличчя). У деяких контактних додатках точність може бути порівнянною з пульсоксиметрією. Однак вони менш точні, ніж спеціалізовані монітори серцевого ритму та медичне обладнання. На точність можуть впливати тиск, що чиниться на палець, інтенсивність освітлення, рух та тон шкіри. Безконтактні додатки можуть завищувати частоту серцевих скорочень при високих значеннях. Вони не є надійними для моніторингу серцевого ритму під час фізичних вправ. Якість камери та модель смартфона також можуть впливати на точність, як і умови навколишнього освітлення. Точність мобільних додатків для вимірювання серцевого ритму дуже мінлива і, як правило, нижча, ніж у спеціалізованих пристроїв, особливо під час активності. Контактні додатки точніші за безконтактні [30, 31].

5. Медичні монітори (пластирі, реєстратори подій, петльові монітори, імплантовані реєстратори): Використовуються для діагностики серцевих захворювань, забезпечують тривалий моніторинг та можуть виявляти різні аритмії. ЕКГ-пластирі забезпечують розширений безперервний моніторинг у зручному бездротовому форматі, покращуючи виявлення рідкісних серцевих подій. Реєстратори подій мають вирішальне значення для фіксації нечастих серцевих подій, що дозволяє співвіднести симптоми з основними порушеннями ритму протягом тривалих періодів моніторингу. Петльова пам'ять підвищує діагностичну цінність завдяки фіксації даних до появи симптомів. Імплантовані петльові реєстратори пропонують найтриваліший період моніторингу, що є критично

важливим для діагностики дуже рідкісних серцевих подій та надання довгострокових даних для лікування таких станів, як фібриляція передсердь у пацієнтів високого ризику [32].

1.3 Аналіз програмно-апаратного забезпечення обробки інформації в кіберфізичній системі

Серед основних вимог до розроблюваного програмно-технічного засобу можна виділити такі : точне вимірювання серцевого ритму за допомогою обраного датчика (ФПГ або ЕКГ), обробка та аналіз отриманих даних для визначення середнього значення серцевого ритму та, можливо, інших параметрів, таких як HRV або виявлення аномалій, підтримка бездротового зв'язку (Wi-Fi та/або Bluetooth) для передачі даних на зовнішні пристрої або платформи [33].

Розробка інтерфейсу для медичних пристроїв повинна поєднувати зручність використання з дотриманням нормативних вимог, зокрема стандарту ІЕС 62366, який гарантує пріоритет безпеки, ефективності та доступності. Інтерфейс має бути інтуїтивно зрозумілим і мінімізувати ризик помилок користувача, враховуючи потреби людей з різними фізичними та когнітивними можливостями. Оскільки інтерфейс є ключовим каналом взаємодії між медичними працівниками та пристроями, його якісний дизайн підвищує продуктивність, зменшує помилки та покращує результати лікування. Критично важливо проводити тестування зручності використання та регулярно отримувати зворотний зв'язок від користувачів. Прикладом ефективного інтерфейсу є візуалізація даних серцевого ритму у зручному форматі, через веб-інтерфейс або мобільний додаток. Енергоефективність для забезпечення тривалої роботи пристрою від акумулятора [34-38].

Безпека передачі медичних даних та захист даних про здоров'я є критично важливим аспектом, що охоплює як збережену інформацію, так і дані, що передаються. У США Закон про переносимість і підзвітність медичного

страхування (HIPAA) визначає загальні рекомендації щодо найкращих практик зберігання та захисту захищеної медичної інформації (PHI). Відповідність вимогам HIPAA гарантує конфіденційність, цілісність та доступність медичних даних, захищаючи їх від несанкціонованого доступу та крадіжки. В Україні персональні дані, пов'язані зі здоров'ям, вважаються чутливими і обробка таких даних, як правило, заборонена без чіткої згоди особи або наявності інших законних підстав. Україна ратифікувала Конвенцію про захист осіб у зв'язку з автоматизованою обробкою персональних даних та її Додатковий протокол, що покладає на державу зобов'язання з належного регулювання цієї сфери [39-41].

1.4 Висновки до першого розділу

У першому розділі було проведено змістовний аналіз предметної області розробки пристрою для вимірювання серцевого ритму, розглянуто її структурні та функціональні особливості. Також було здійснено аналіз наявного програмно-апаратного забезпечення у даній сфері та визначено основні вимоги до майбутньої системи. На основі проведеного аналізу буде розроблено технічне завдання для подальшої розробки пристрою.

					КВРКІ 210243.21.02.58 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ ДЛЯ ВИМІРЮВАННЯ СЕРЦЕВОГО РИТМУ

2.1 Розробка структури програмно-апаратного засобу вимірювання серцевого ритму

Для реалізації програмно-апаратного засобу вимірювання серцевого ритму розглянемо його функціональну схему що представлена на рис. 2.1. Вона включає наступні компоненти:

1) Мікроконтролер відповідає за збір та обробку інформації від давачів, здійснює аналіз отриманих даних та приймає рішення щодо генерації сповіщень і надсилання повідомлень.

2) Блок давачів, який включає давачі рівня серцевого ритму та рівня кисню в крові.

3) Блок сповіщення, який призначений для відображення сповіщень та необхідних параметрів.

4) Модуль зв'язку, який забезпечує обмін інформацією між системою та віддаленим сервером, що дозволяє відправляти дані для реалізації дистанційного моніторингу.

5) Віддалений сервер для зберігання вимірюваних величин, що дозволяє аналізувати стан здоров'я людини з часом.

6) Інтерфейс користувача реалізований у вигляді мобільного додатку або веб-інтерфейсу, що дозволяє налаштовувати параметри системи, відслідковувати стан зміни контрольованих параметрів, отримувати сповіщення та інформацію в реальному часі.

Ця функціональна схема забезпечує виконання поставлених завдань для здійснення моніторингу та контролю стану серцевого ритму користувача з можливістю отримання даних не лише на засобах відображення приладу а і дистанційно з використанням ресурсу мережі Інтернет на смартфоні або персональному комп'ютері.

					КВРКІ 210243.21.02.58 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

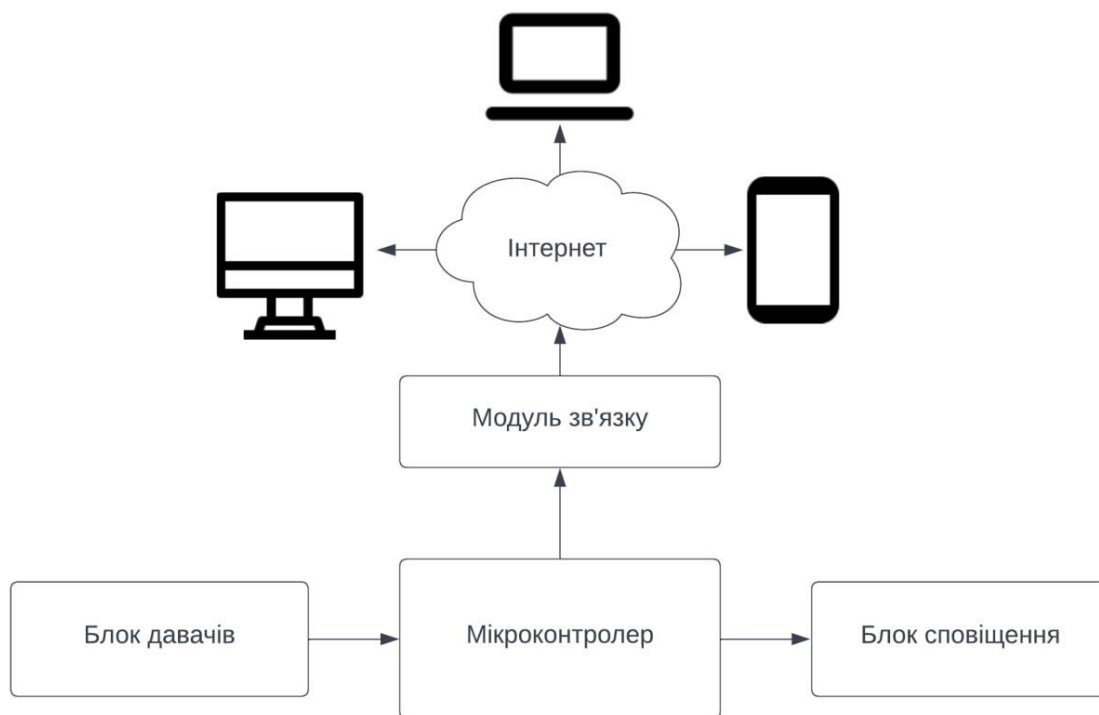


Рисунок 2.1 – Функціональна схема програмно-апаратного засобу вимірювання серцевого ритму

Для реалізації вибраної схеми проаналізуємо та здійснимо вибір основних апаратних та програмних компонентів.

2.2 Апаратне забезпечення програмно-апаратного засобу вимірювання серцевого ритму

Мікроконтролер – це інтегральна схема, яка об'єднує центральний процесор, пам'ять і периферійні пристрої введення/виведення, призначена для автоматичного керування електронними системами. Він виконує попередньо запрограмовані інструкції, зчитує дані з датчиків, обробляє інформацію та керує виконавчими механізмами, забезпечуючи автономну роботу обладнання.

Звичайні маленькі мікроконтролери, які широко використовуються в портативних і вбудованих пристроях, включають ATtiny (включаючи ATtiny85),

STM32F0/G0 (такі як STM32F030), ESP32-C3 і ESP8285, а також PIC10/12/16 від Microchip. Ці мікроконтролери відрізняються компактним корпусом, низьким енергоспоживанням і достатнім набором функцій для управління датчиками, модулями зв'язку і виконавчими механізмами. Крім того, мікроконтролери nRF52 (такі як nRF52832), які підтримують Bluetooth Low Energy, також часто використовуються в пристроях з бездротовою передачею даних. Завдяки невеликому розміру та універсальності ці мікроконтролери ідеально підходять для використання в переносній електроніці, сенсорних системах, Інтернеті речей (IoT) та інших пристроях, де простір обмежений.

ESP32, ESP8266 і Arduino UNO – це три популярні мікроконтролери, кожен з яких має свої унікальні характеристики (рис. 2.2), які роблять їх придатними для певних застосувань.

SPECS/BOARD	ESP32	ESP8266	ARDUINO UNO
Number of Cores	2	1	1
Architecture	32 Bit	32 Bit	8 Bit
CPU Frequency	160 MHz	80 MHz	16 MHz
WiFi	YES	YES	NO
BLUETOOTH	YES	NO	NO
RAM	512 KB	160 KB	2 KB
FLASH	16 MB	16 MB	32 KB
GPIO PINS	36	17	14
Busses	SPI, I2C, UART, I2S, CAN	SPI, I2C, UART, I2S	SPI, I2C, UART
ADC Pins	18	1	6
DAC Pins	2	0	0

Рисунок 2.2 – Порівняльна таблиця ESP32, ESP8266 та Arduino UNO

ESP32 (рис. 2.3) має два ядра, що робить його потужнішим у порівнянні з однопроцесорним ESP8266 і Arduino UNO. Багатоядерні можливості дозволяють ESP32 ефективніше виконувати паралельні завдання. І ESP8266, і Arduino UNO оснащені одним ядром. Обидва використовують різні архітектури: ESP8266 – 32-

розрядний, а Arduino UNO – 8-розрядний. 32-розрядна архітектура ESP8266 забезпечує більш високу продуктивність і здатність виконувати більш складні завдання. ESP32 працює на частоті 160 МГц, найвищій з трьох мікроконтролерів. Це забезпечує йому високу продуктивність і швидку обробку даних. ESP8266 має частоту 80 МГц, що також досить високо, але все ще не так добре, як ESP32. Arduino UNO з частотою 16 МГц значно поступається ESP32 і ESP8266, що обмежує його застосування в проектах, які вимагають високої швидкості обробки.

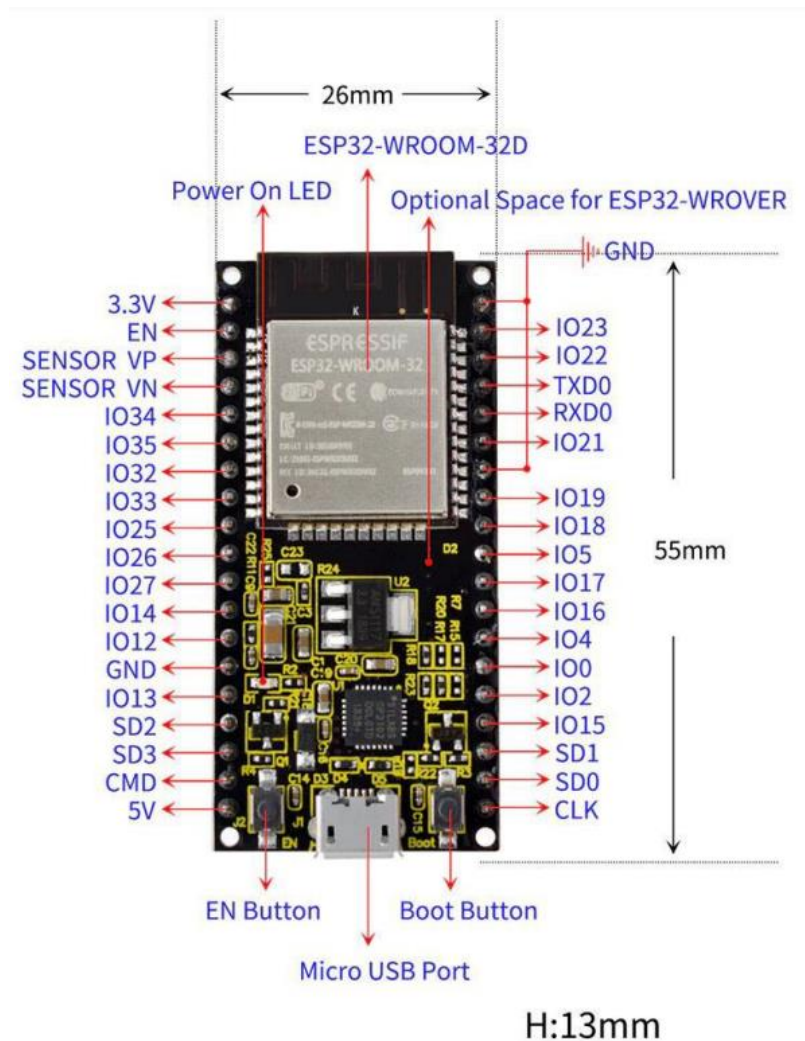


Рисунок 2.3 – Розташування та призначення пінів на ESP32

ESP32 має 512 КБ оперативної пам'яті, що суттєво перевищує обсяг у ESP8266 (160 КБ) та Arduino UNO (2 КБ), завдяки чому він здатен ефективніше обробляти великі обсяги даних і виконувати складніші програми. Обсяг FLASH-

пам'яті в ESP32 та ESP8266 однаковий і становить 16 МБ, що значно більше порівняно з 32 КБ у Arduino UNO, що забезпечує можливість зберігати більші за обсягом програми та додаткові дані.

Кількість GPIO-пінів є важливим параметром при виборі мікроконтролера. ESP32 має 36 таких пінів, що значно перевищує кількість у ESP8266 (17) та Arduino UNO (14), що забезпечує більшу гнучкість у підключенні датчиків і виконавчих пристроїв.

За підтримкою інтерфейсів ESP32 також вирізняється універсальністю, підтримуючи SPI, I2C, UART, I2S та CAN. Для порівняння, ESP8266 підтримує SPI, I2C, UART і I2S, що робить його досить функціональним, але з меншими можливостями, ніж у ESP32. Arduino UNO підтримує лише SPI, I2C і UART, що обмежує варіанти підключення. За кількістю аналогових входів (ADC), ESP32 також має перевагу - 18 пінів, у той час як ESP8266 має лише 1, а Arduino UNO - 6.

Додатково ESP32 оснащений двома цифровими аналоговими перетворювачами (DAC), чого немає в ESP8266 та Arduino UNO. Таким чином, ESP32 є найпотужнішим і найфункціональнішим мікроконтролером серед розглянутих варіантів.

Таблиця 2.2 – Технічні характеристики ESP32

Параметр	Опис
Архітектура	Xtensa 32-bit LX6 dual-core
Тактова частота	240 МГц
Оперативна пам'ять (RAM)	520 КБ SRAM
Вбудована пам'ять (ROM)	448 КБ ROM
Флеш-пам'ять	4 МБ
GPIO	34 програмованих входів/виходів
Інтерфейси	SPI, I2C, I2S, UART, SDIO, PWM, ADC, DAC
Робоча напруга	2.2–3.6 В

Кінець таблиці 2.2

Споживана потужність	Споживання в режимі сну: < 5 μ A, при активній роботі: до 500 мА
Wi-Fi	802.11 b/g/n
Bluetooth	Bluetooth 4.2 та BLE
Температурний діапазон	-40°C до +125°C
Підтримувані середовища розробки	Arduino IDE, ESP-IDF, MicroPython, PlatformIO
Додаткові функції	Вбудований датчик Холла, датчик температури, підтримка апаратного шифрування, акселератор криптографії
Роз'єм для антени	Вбудована антена та роз'єм для зовнішньої антени (u.FL)
Сертифікація	CE, FCC, RoHS
Розміри модуля	25.5 мм x 18 мм

Використання ESP32 завдяки його високій продуктивності, багатоядерності, широким можливостям бездротового зв'язку (Wi-Fi та Bluetooth), великій кількості GPIO пінів та підтримці різних інтерфейсів, дозволить здійснити розробку необхідного апаратно-програмного комплексу з можливістю подальшого розвитку та удосконалення.

Датчик серцевого ритму MAX30102 (рис 2.4) є інтегрованим модулем для вимірювання пульсу та рівня насичення крові киснем (SpO_2). У його складі об'єднані світлодіоди, фотодетектор, високоточний аналоговий підсилювач, АЦП, цифровий обробник сигналу та модуль інтерфейсу. Для підключення до мікроконтролера використовується послідовний інтерфейс I2C.

Модуль оснащено двома світлодіодами — червоного та інфрачервоного діапазонів. Є можливість регулювання струму світіння (від 0 до 50 мА) та тривалості імпульсів (від 200 мкс до 1,6 мс), що дозволяє точно налаштувати режим

роботи для отримання більш достовірних результатів. Крім того, у пристрої вбудований температурний датчик, який забезпечує можливість калібрування.



Рисунок 2.4 – Розташування пінів на MAX30102

Основні характеристики MAX30102:

- вимірювані параметри: частота пульсу та SpO_2 ;
- живлення: 3,3 В (з вбудованим стабілізатором на 1,8 В);
- споживання струму під час вимірювання: 1,2 мА;
- споживання струму в режимі сну: до 10 мкА;
- інтерфейс: I2C (з напругою 3,3 В та частотою до 400 кГц);
- габарити: 18,5 × 14,5 × 3 мм.

Завдяки компактності, низькому енергоспоживанню та високій точності, цей сенсор ідеально підходить для використання в портативних і переносних медичних пристроях.

Модуль MPU-6050 (рис 2.5) є компактним сенсором, який об'єднує в собі 3-осьовий акселерометр і 3-осьовий гіроскоп, що дозволяє точно визначати положення та рух об'єкта у просторі. Керування модулем здійснюється через інтерфейс I2C (TWI), що забезпечує простоту підключення до мікроконтролерів.

У модулі використовується чіп MPU-6050, який містить вбудований 16 бітний аналогово-цифровий перетворювач (АЦП) для кожного з каналів, що забезпечує високу точність вимірювань. Доступні різні діапазони вимірювання для гіроскопа (± 250 , ± 500 , ± 1000 , ± 2000 °/с) та акселерометра (± 2 , ± 4 , ± 8 , ± 16 g), що дозволяє адаптувати модуль до різних умов роботи.

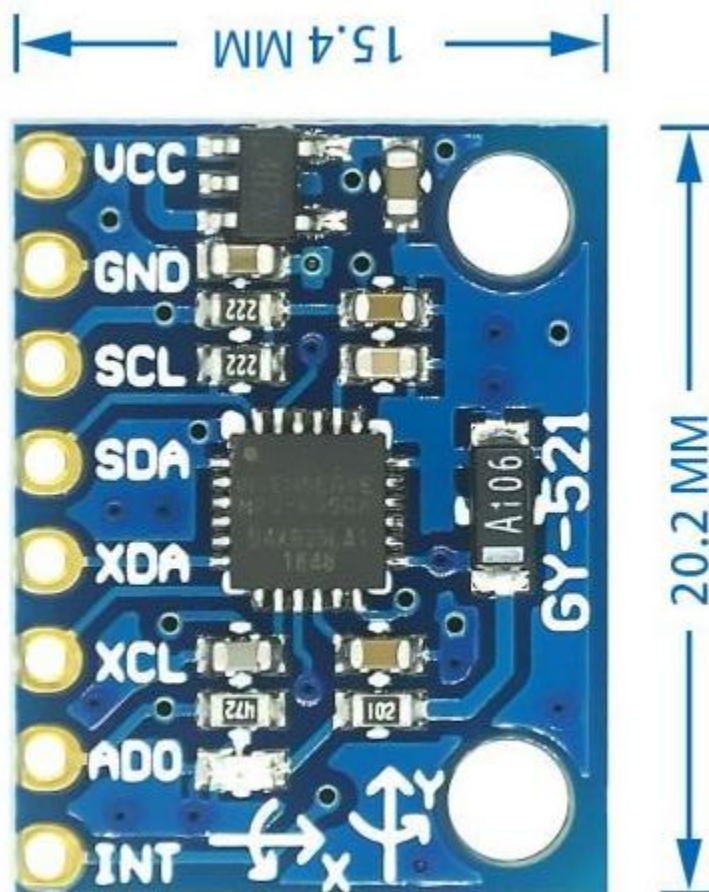


Рисунок 2.5 – Розташування пінів на MPU-6050

Основні технічні характеристики MPU-6050;

- напруга живлення: 3,3–5,0 В;
- інтерфейс: I2C (TWI);
- чіп: MPU-6050;
- вбудований 16-бітний АЦП;
- діапазон гіроскопа: ± 250 , ± 500 , ± 1000 , ± 2000 град/с;

Зм.	Арк.	№ докум.	Підпис	Дата

КВРКІ 210243.21.02.58 ПЗ

Арк.
25

- діапазон акселерометра: $\pm 2, \pm 4, \pm 8, \pm 16$ g;
- розміри модуля: 20×16 мм.

Завдяки своїй малій вазі, габаритам і сумісності з широким діапазоном напруги, модуль ідеально підходить для застосування в мобільній робототехніці, переносних пристроях, ігрових контролерах та системах стабілізації.

SPI TFT-дисплей (рис. 2.6) із роздільною здатністю 240×320 пікселів на базі контролера ILI9341 є оптимальним рішенням для проектів, де важливе зручне відображення та введення інформації при мінімальних апаратних витратах. Завдяки використанню послідовного інтерфейсу SPI, дисплей легко підключається до широкого спектра мікроконтролерів, включаючи Arduino, Raspberry Pi та інші платформи, які підтримують даний інтерфейс. Крім того, на платі передбачено SD слот, що дозволяє використовувати карту пам'яті для зберігання великих обсягів даних, зокрема графічної інформації.

Дисплей може працювати від джерела живлення як 3,3 В, так і 5 В, що робить його універсальним у застосуванні.

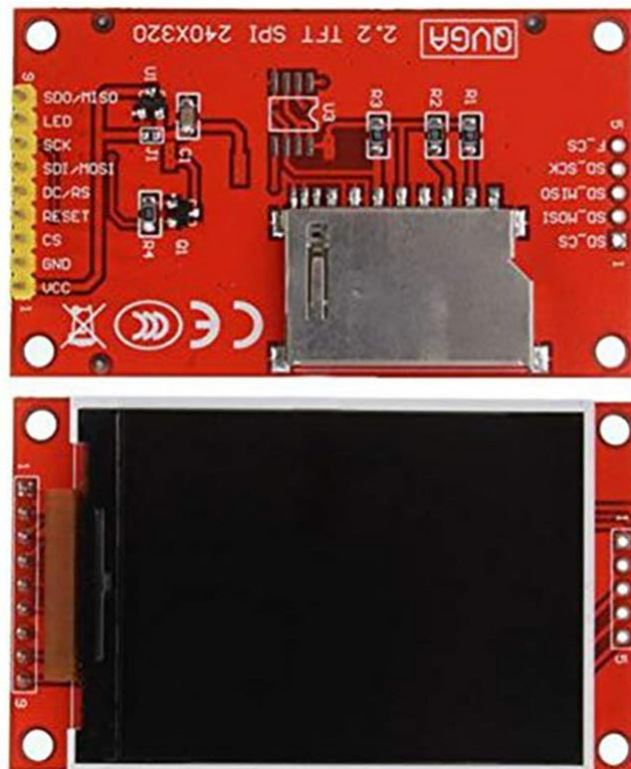


Рисунок 2.6 – Розташування пінів на ILI9341

Основні технічні характеристики:

- тип дисплея: LCD TFT (без сенсорної панелі);
- розмір екрана: 2,8 дюйма;
- інтерфейс: SPI;
- напруга логічних рівнів: 3,3 В;
- розміри активної області дисплея: 43,2 × 57,6 мм;
- розміри плати: 50,0 × 86,0 мм;
- розміщення пікселів: RGB, вертикальне;
- контролер: ILI9341;
- підсвічування: біле світлодіодне;
- кут огляду: 70°;
- глибина кольору: 65 тисяч кольорів;
- роздільна здатність: 240 × 320 пікселів;
- живлення: 3,3–5 В;
- кількість керувальних ліній: 4.

Цей дисплей є надійним та зручним компонентом для візуалізації даних у вбудованих системах, портативних пристроях, приладах керування тощо.

2.3 Програмне забезпечення програмно-апаратного засобу вимірювання серцевого ритму

Для реалізації взаємодії пристрою з користувачем буде застосовано протокол обміну MQTT.

MQTT (Message Queuing Telemetry Transport) – це спеціалізований протокол, розроблений для потреб інтернету речей (IoT). Його головними перевагами є відкритість, простота реалізації та ефективність обміну даними між пристроями й модулями. MQTT забезпечує надійне та оперативне з'єднання між елементами

					КВРКІ 210243.21.02.58 ПЗ	Арк. 27
Зм.	Арк.	№ докум.	Підпис	Дата		

системи, спрощуючи комунікацію та оптимізуючи передачу даних у реальному часі.

Протокол відповідає за безпечне встановлення з'єднання, стабільну роботу мережі, а також високошвидкісну передачу інформації. Він стійкий до збоїв і технічних зупинок, завдяки чому забезпечує безперебійне функціонування програмних та апаратних компонентів. MQTT дозволяє ефективно організувати взаємодію як між окремими пристроями, так і між цілими локальними мережами, забезпечуючи гнучку масштабованість у межах глобального інтернету.

MQTT – це відкритий та легкий протокол, спеціально створений для пристроїв із низьким енергоспоживанням. Його основна функція — обмін повідомленнями між пристроями через посередника, званого MQTT-брокером. Комунікація організована за моделлю «видавець–передплатник», де одні пристрої (видавці) надсилають дані, а інші (передплатники) отримують їх за відповідними темами. Брокер забезпечує маршрутизацію повідомлень, зберігаючи стабільність і точність передавання інформації.

Протокол MQTT демонструє високу ефективність у роботі навіть у мережах із низькою пропускнуою здатністю та не потребує значних обчислювальних ресурсів, що робить його ідеальним для вбудованих систем. Завдяки мінімальній кількості службових даних, він значно перевершує традиційні протоколи, зокрема HTTP, в плані пропускнуої здатності та швидкості - згідно з дослідженнями, MQTT у 93 рази ефективніший за REST у 3G-мережах.

Окрема перевага MQTT – це простота налаштування та використання. Виконавчі агенти (видавці або передплатники) взаємодіють з брокером, надсилаючи або отримуючи повідомлення в межах конкретних тем, що забезпечує гнучку і надійну структуру комунікації в системах IoT. Завдяки цьому MQTT набуває широкого поширення у розумних будинках, системах моніторингу, промисловій автоматизації та інших галузях.

У типовій клієнт-серверній архітектурі передбачено наявність постачальника (тобто пристрою, що надсилає дані з датчиків на сервер) та споживача (користувача

або системи, яка звертається до сервера для отримання цих даних). Протокол MQTT (рис. 2.7) працює за схожим принципом, однак має свої особливості. Це легкий і ефективний протокол, що забезпечує взаємодію між пристроями у мережі.

У системі MQTT пристрої "публікують" дані на спеціальний сервер – MQTT-брокер, який виступає посередником у передаванні інформації. Після публікації дані стають доступними для інших пристроїв або веб-застосунків, які можуть "підписуватись" на ці теми, тобто отримувати відповідні повідомлення від брокера. Такий підхід дозволяє ефективно організувати обмін даними між великою кількістю учасників системи.

На відміну від звичайного веб-сервера, MQTT має дві ключові переваги:

1. Використання надзвичайно легких пакетів даних, що підвищує швидкість передачі, знижує навантаження на мережу і зменшує енергоспоживання.
2. Можливість роботи не лише через Інтернет, а й через альтернативні канали зв'язку, наприклад Bluetooth, що розширює сферу його застосування у пристроях з обмеженим доступом до традиційних мереж.

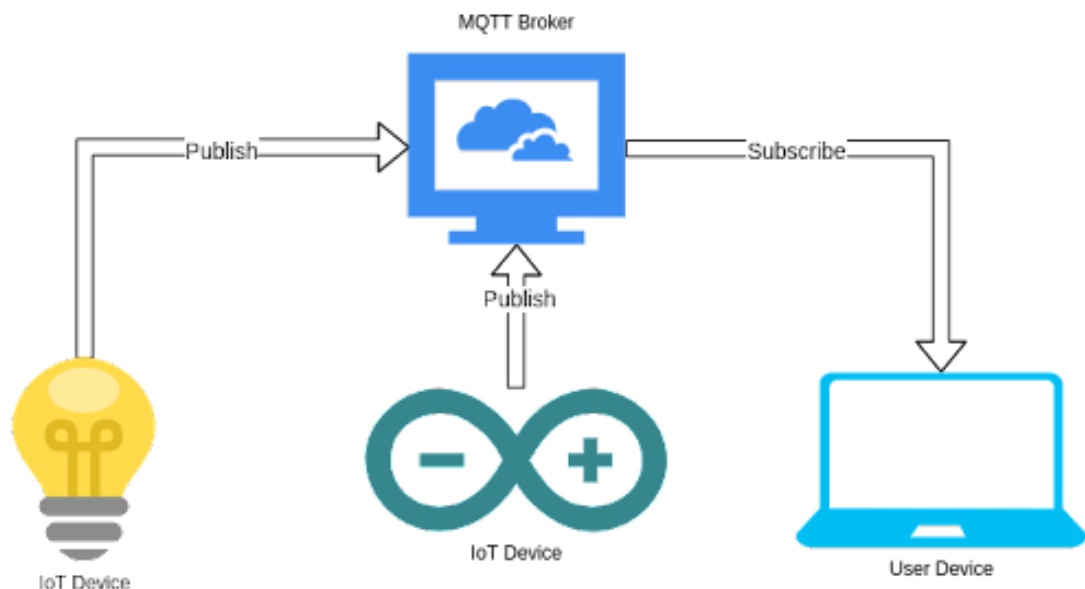


Рисунок 2.7 – MQTT брокер з пристроями

Завдяки вищезгаданим перевагам, MQTT здобув широку популярність як протокол зв'язку серед IoT-пристроїв. Раніше для взаємодії між пристроями та

сервером переважно використовували HTTP-запити типу POST або GET, однак такий підхід обмежував можливість застосування альтернативних засобів зв'язку, зокрема Bluetooth чи ZigBee. MQTT, будучи незалежним від типу транспортного середовища, усуває ці обмеження, що дозволяє реалізовувати передачу повідомлень у межах локальних mesh-мереж навіть без доступу до інтернету.

Розглянемо принцип застосування MQTT при проектуванні пристроїв в середовищі Arduino IDE. Існує безліч способів розпочати роботу з MQTT, включаючи розробку повного, кінцевого рішення. Щоб спростити завдання, ми будемо використовувати безкоштовний сервіс брокера MQTT, який надається Adafruit.

Перш за все необхідно встановити потрібні бібліотеки. Відкриваємо Arduino IDE і перейдемо на лівій панелі в пункт «Керування бібліотеками» (рис.2.8).

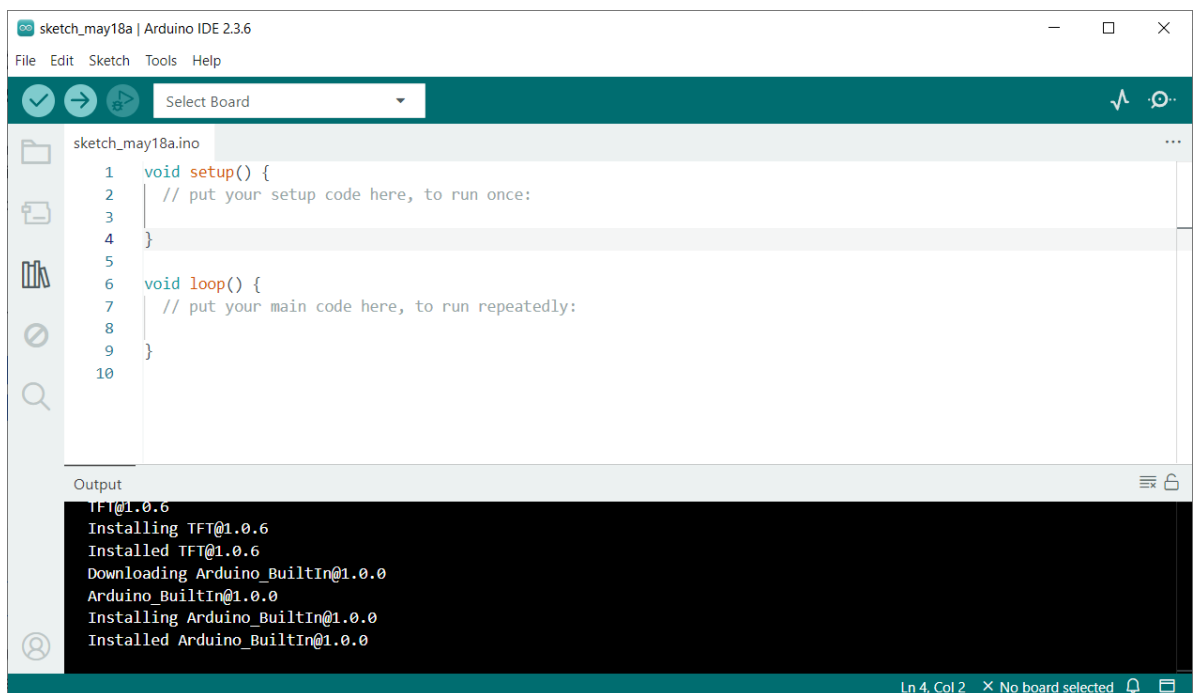


Рисунок 2.8 – Керування бібліотеками в Arduino IDE

Знаходимо бібліотеку Adafruit MQTT та встановлюємо її (рис. 2.9)

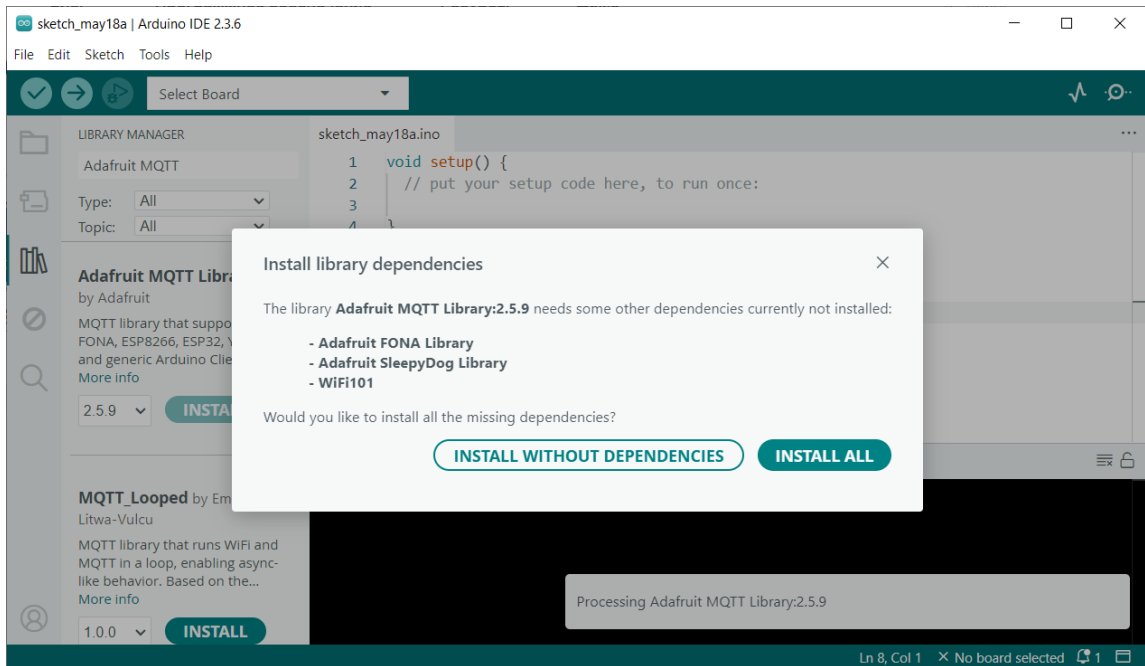


Рисунок 2.9 – Встановлення бібліотеки Adafruit MQTT в Arduino IDE

Після встановлення бібліотек ми можемо перейти до написання коду, але попередньо нам потрібно зробити паузу та отримати наші облікові дані з облікового запису Adafruit IO. Переходим на сайт «adafruit.com». Входимо у Adafruit IO і натискаємо "Мій ключ" у верхній частині (рис.2.10).

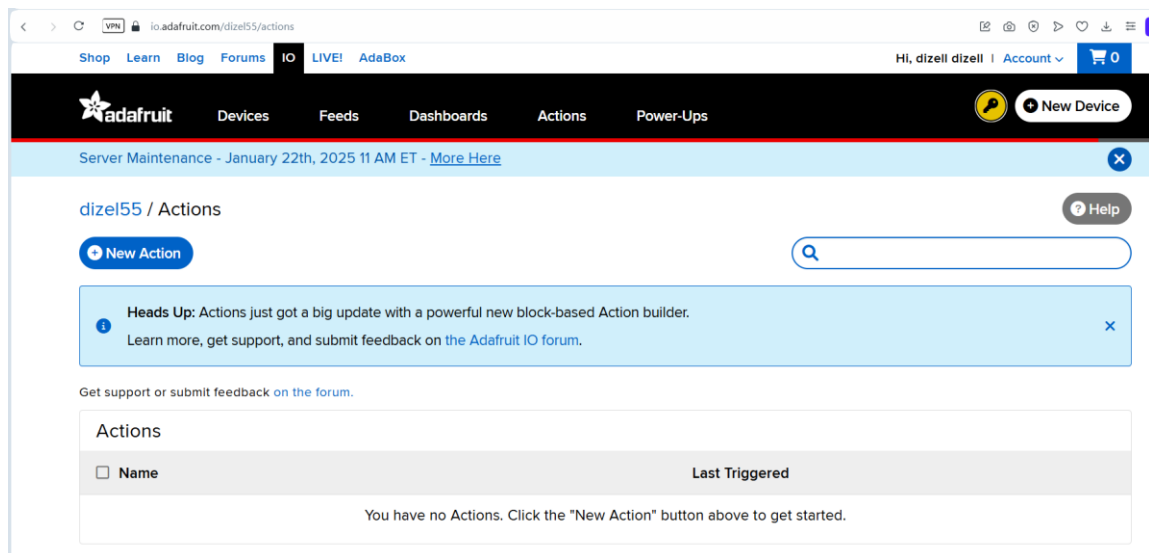
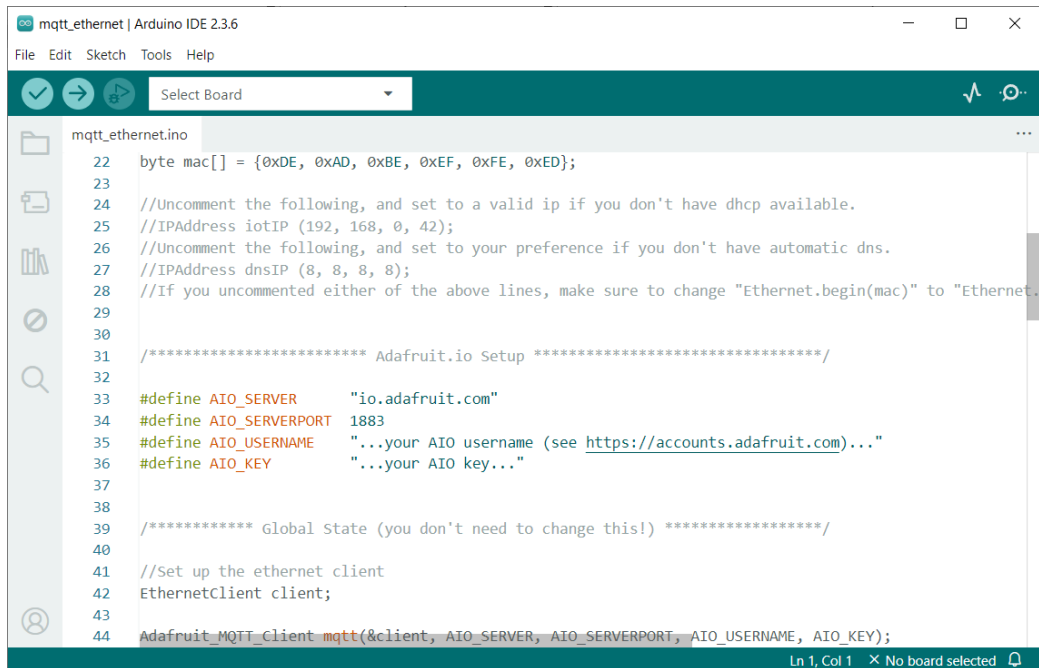


Рисунок 2.10 – Отримання ключа доступу на сайті Adafruit.com

Після отримання параметрів аутенфікації, вводимо їх у відповідні поля AIO_USERNAME и AIO_KEY в кодї програми та завантажуюмо програму в пристрій (рис 2.11).



```
mqtt_ethernet | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Select Board
mqtt_ethernet.ino
22 byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
23
24 // Uncomment the following, and set to a valid ip if you don't have dhcp available.
25 // IPAddress iotIP (192, 168, 0, 42);
26 // Uncomment the following, and set to your preference if you don't have automatic dns.
27 // IPAddress dnsIP (8, 8, 8, 8);
28 // If you uncommented either of the above lines, make sure to change "Ethernet.begin(mac)" to "Ethernet.begin(mac, dnsIP)".
29
30
31 /***** Adafruit.io Setup *****/
32
33 #define AIO_SERVER      "io.adafruit.com"
34 #define AIO_SERVERPORT 1883
35 #define AIO_USERNAME    "...your AIO username (see https://accounts.adafruit.com)..."
36 #define AIO_KEY         "...your AIO key..."
37
38
39 /***** Global State (you don't need to change this!) *****/
40
41 // Set up the ethernet client
42 EthernetClient client;
43
44 Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
Ln 1, Col 1 × No board selected
```

Рисунок 2.11 – Введення параметрів аутенфікації брокера в програму

Щоб подивитися актуальні дані на сервері, необхідно повернутися до нашого облікового запису Adafruit IO і натиснути на "Feeds", потім натиснути на назву нашого з'єднання з платою (рис 2.12).



Рисунок 2.12 – Канали Adafruit IO

Тепер ми отримуємо живу передачу даних, які наш Arduino надсилає на брокер MQTT. Тут ми також можемо завантажити попередні данні для аналізу інформації (рис 2.13).

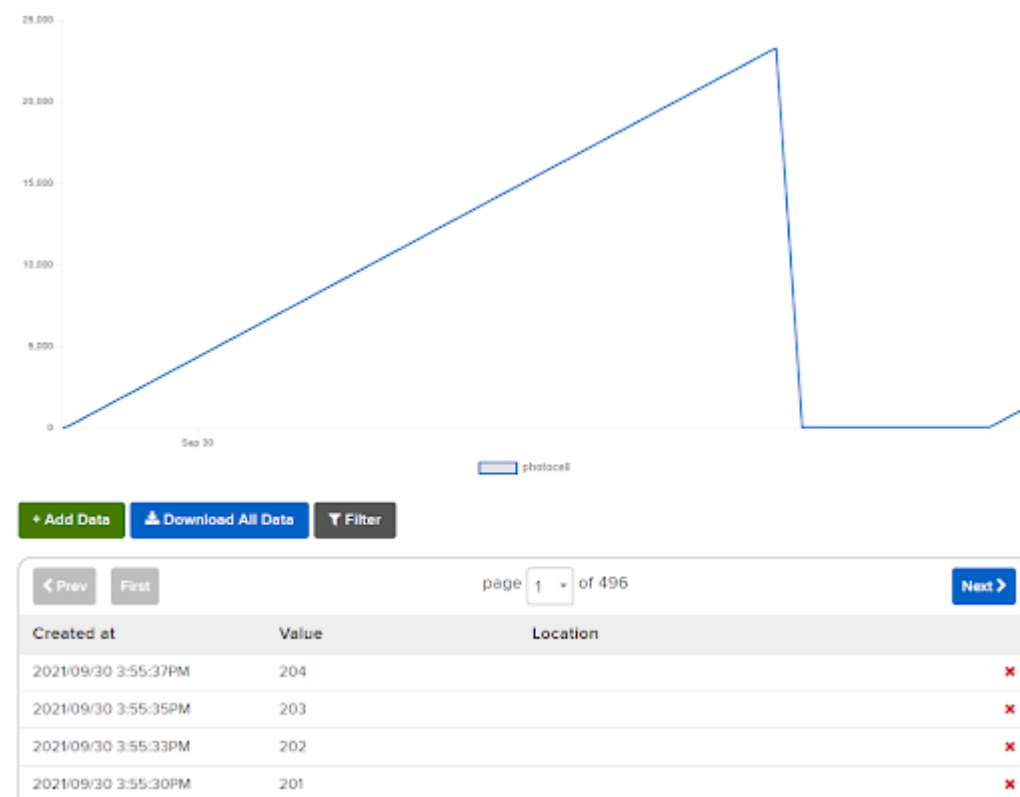


Рисунок 2.13 – Актуальні та попередні данні Adafruit IO

2.4 Опис принципової схеми пристрою

Для ефективного функціонування програмно-апаратного засобу для вимірювання серцевого ритму та аналізу отриманих даних необхідно налагодити коректну взаємодію між її апаратними та програмними компонентами. У цьому розділі розглядаються методи комунікації, які застосовані в проекті для об'єднання окремих елементів системи в єдине ціле.

У даній реалізації (рис 2.14) мікроконтролер ESP32 взаємодіє з датчиком пульсу та насичення крові киснем MAX30102 та датчиком положення в просторі MPU-6050 за допомогою інтерфейсу UART (Universal Asynchronous Receiver-Transmitter). Цей протокол забезпечує послідовну асинхронну передачу даних, при якій інформація надходить по одному біту через одну лінію зв'язку. Такий підхід є

простим у впровадженні й зручним для роботи з різноманітними сенсорними пристроями.

MAX30100 надсилає дані, що містять значення частоти пульсу та рівня кисню в крові, у вигляді серійних пакетів. Кожен пакет формується на основі даних, зібраних протягом певного проміжку часу, та передається до ESP32 для оперативної обробки в реальному часі.

Аналогічно MPU-6050 надсилає пакети даних про зміну положення датчика, що буде відповідати зміну положення тіла людини. Дані про положення будуть використані для аналізу інформації про рух об'єкта та відповідно обрахування кількості витрачених калорій.

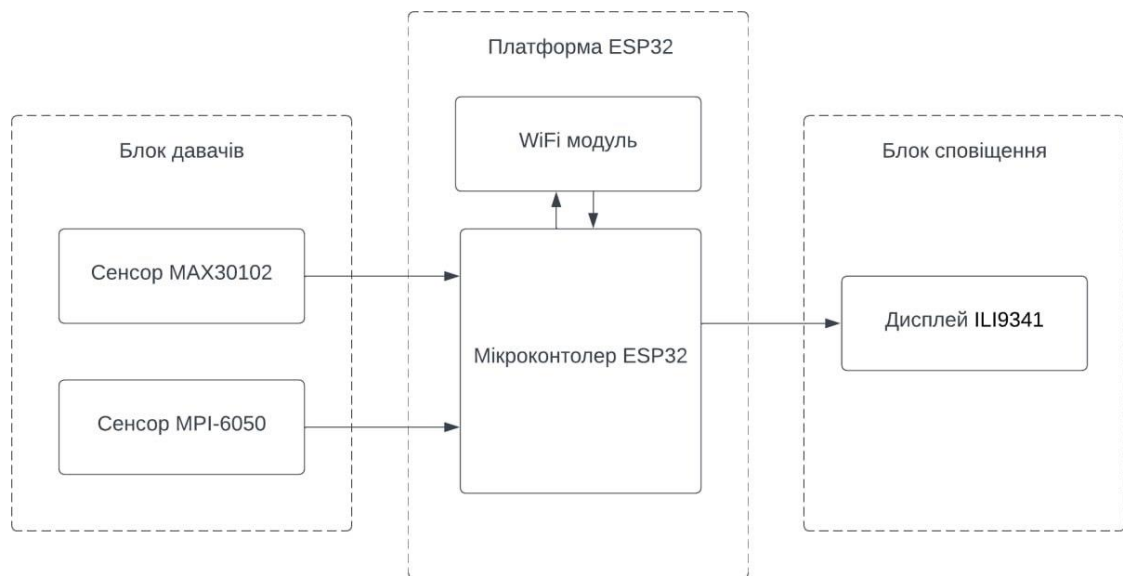


Рисунок 2.14 – Структурна схема програмно-апаратного засобу

Контролер ESP32 обробляє отриманні дані згідно складеного алгоритму та надсилає необхідну інформацію на SPI TFT-дисплей із роздільною здатністю 240×320 пікселів на базі контролера ILI9341 та за допомогою вбудованого Wi-Fi модуля встановлює з'єднання з безпроводною мережею і передає дані на MQTT-брокер (віддалений сервер або хмару) за моделлю «підписка/публікація». Згідно з цим, після аутентифікації ESP32 публікує вимірювання в певний «топік» брокера.

Мобільний пристрій, підключений до того ж брокера, підписується на відповідний «топик» і отримує дані в реальному часі. Таким чином, дані вимірювань передаються через інтернет по Wi-Fi, а отримані значення можуть виводитися у вигляді графіків на мобільному екрані.

Протокол MQTT був обраний як стандартизований простий протокол обміну повідомленнями для IoT: він розроблений як дуже спрощений механізм «publish/subscribe», оптимізований для з'єднання віддалених пристроїв з невеликим кодовим і мережевим навантаженням.

Алгоритм роботи програмно-апаратного засобу для вимірювання серцевого ритму (рис 2.15) показує принцип роботи апаратної складової під ініціалізацією та виконанням приладом опрацювання отриманих даних.

Алгоритм починається з натискання кнопки, а потім послідовно ініціалізує два датчики: MAX30102 (для вимірювання пульсу) і MPU-6050 (для виявлення руху). На цьому етапі перевіряється їх наявність: якщо хоча б одного не знайдено, на екран виводиться повідомлення про помилку і виконання програми припиняється. Якщо обидва датчики успішно виявлені, користувачеві відкривається екран завантаження, а потім починається основний цикл читання даних.

Під час основного циклу спочатку виконується зчитування з MAX30102. Якщо серцебиття не виявлено, алгоритм повернеться до повторного читання. Після виявлення серцебиття дані зчитуються з MPU-6050. Отримана інформація обробляється (фільтрація шумів, розрахунок параметрів), зберігається та відображається на екрані пристрою. Потім він перевіряє наявність бездротового з'єднання: якщо з'єднання існує, дані надсилаються на зовнішній пристрій; якщо він відсутній, програма перестає працювати.

До переваг даного алгоритму можна віднести автоматичну перевірку працездатності пристрою перед початком роботи, умовне зчитування даних (активація MPU-6050 тільки при серцебитті), чітку обробку помилок і підтримку бездротової передачі інформації. Для покращення функціональності варто

обмежити кількість спроб зчитування серцевого ритму, щоб уникнути нескінченних циклів, додати сповіщення про їх відсутність (з тайм-аутом) і реалізувати періодичні повторні перевірки бездротового з'єднання для урахування короткочасних втрат мережі.

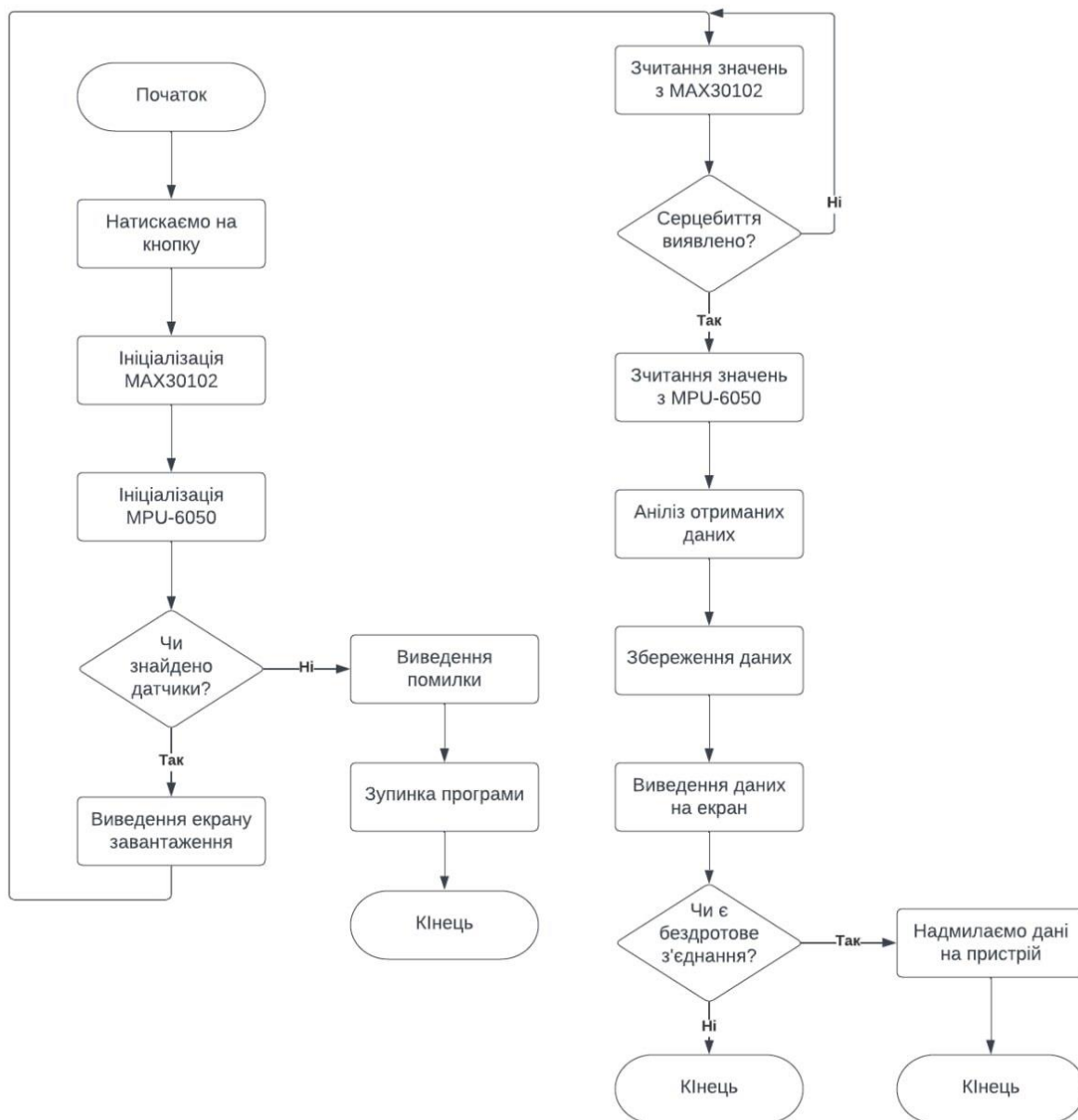


Рисунок 2.15 – Алгоритм роботи програмно-апаратного засобу

2.5 Висновки до другого розділу

У межах розділу 2 проведено аналіз апаратних та програмних підсистем програмно-технічного засобу, способи взаємодії між підсистемами та описано функціональне призначення основних модулів та інформаційних ресурсів програмно-технічного засобу. Розглянута система включає такі апаратні підсистеми: ESP32 зі вбудованими модулями Wi-Fi та Bluetooth, датчик MAX30100, датчик MPU-6050 та LCD-дисплей.

Функціонування системи ґрунтується на взаємодії контролера ESP 32 з датчиками пульсу та положення людини, інформація з яких буде аналізуватися та відображатися на екрані дисплея, паралельно буде здійснюватися передача даних через MQTT протокол на цифровий пристрій користувача.

Використання MQTT протоколу дозволить не тільки отримувати інформацію в реальному часі але й переглядати певну історичну інформацію, яка передавалася за допомогою даного протоколу раніше.

					КВРКІ 210243.21.02.58 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ПРАКТИЧНА ЧАСТИНА РОЗРОБКИ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ

3.1 Опис реалізації модулів апаратного забезпечення програмно-апаратного засобу

Створення програмно-апаратного засобу розпочинається з розробки електричної схеми пристрою та її налагодження за допомогою емулятора електронних схем. Складання схеми за допомогою емулятора дозволить перевірити її працездатність та виявити несправності та порушення реалізації схеми підключення елементів між собою.

Електронну схему розробляємо за допомогою емулятора роботи ESP32 - <https://wokwi.com/>. Даний емулятор є досить простим у використанні, працює за допомогою веб-інтерфейсу та дозволяє створювати емуляції безкоштовно.

Для початку виконання емуляції, створюємо новий проект в якому вибираємо як основний елемент мікроконтролер ESP32. Далі здійснюємо розміщення та підключення необхідних нам датчиків та елементів керування.

Серед доступних елементів для виконання емуляції присутній модуль MPU-6050 та дисплей ILI9341. Модуля MAX30102 немає у списку доступних компонентів, тому його роботу змоделюємо самостійно, використовуючи об'єкт «Custom Chip». Даний компонент дозволить нам змоделювати роботу датчика вимірювання пульсу та рівню кисню для використання наданої ним інформації в процесі налагодження програми роботи створеного засобу. Для моделювання створимо дві величини: пульс та рівень кисню, які будуть передаватися в контролер відповідно до математичної функції, що відповідатиме природньому ритму зміни цих величин.

Вибравши необхідні компоненти, здійснюємо їх з'єднання згідно розпінування контролера та відповідних модулів. З'єднання здійснюємо відповідно до креслення «Схема з'єднання компонентів ПЗ» (див. Додаток Б). Живлення усіх додаткових модулів автоматично з'єднується з піном +3,3 В та «Gnd» контролера

					КвРКІ 210243.21.02.58 ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

ESP32. В результаті отримаємо схематичне зображення нашої схеми з відображенням усіх з'єднувальних провідників (рис. 3.1).

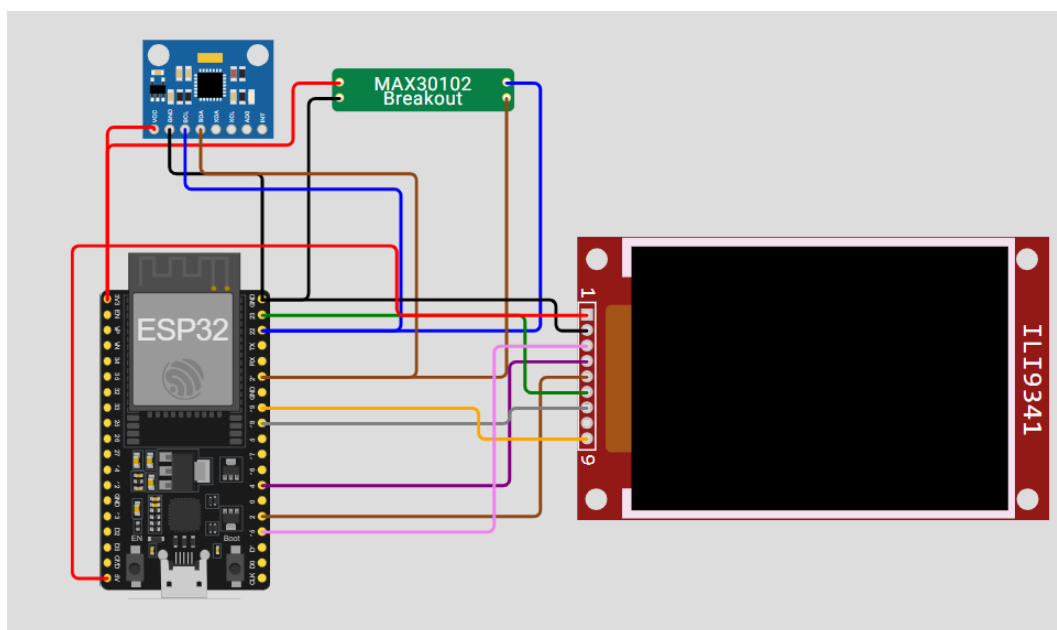


Рисунок 3.1 –Розміщення елементів в симуляторі Wokwi

Для створення програми роботи програмно-апаратного засобу складемо алгоритм його роботи, що буде дещо спрощений (рис. 3.2), враховуючи принципи роботи контролера ESP32, що застосовний в емуляторі.

Процес починається з ініціалізації та перевірки справності апаратної частини системи, включаючи підключені сенсори та модулі. Якщо під час перевірки виявляється, що все функціонує коректно, пристрій переходить до етапу підключення до мережі WiFi та спроби встановлення з'єднання з MQTT брокером. У разі успішного з'єднання відбувається зчитування даних з датчиків, після чого зібрана інформація узагальнюється для побудов відповідного графіка та передачі у мережу. Далі дані передаються на MQTT брокер для подальшого використання іншими сервісами або пристроями. Якщо на будь-якому з етапів виявляється помилка - наприклад, відсутнє з'єднання або несправна апаратна частина - пристрій повертається до відповідного кроку для повторної спроби, забезпечуючи надійність та безперервність процесу.

Зм.	Арк.	№ докум.	Підпис	Дата



Рисунок 3.2 – Алгоритм роботи програмно-апаратного засобу для емулятора

Аналіз отриманих даних буде включати в себе збір інформації у масив для побудови графіка зміни величин на екрані пристрою.

У симуляторі Wokwi модуль MPU6050 дозволяє емулювати роботу тривісного акселерометра та гіроскопа, що є важливими елементами для визначення руху та положення пристрою в просторі. Акселерометр вимірює прискорення по трьох осях: X, Y та Z, яке подається у значеннях від -2g до +2g, де 1g - це прискорення вільного падіння. Користувач має можливість вручну змінювати ці значення за допомогою повзунків у вікні налаштувань, імітуючи різні

сценарії руху, наприклад, нахил пристрою вперед, у бік або вгору. У наведеному прикладі прискорення становить 0g по осі X, що означає відсутність руху вбік, 2g по осі Y, що вказує на сильний рух або нахил вперед, і 1g по осі Z - рух або зміщення вгору. Таке налаштування може бути використано для тестування алгоритмів розпізнавання положення або виявлення падіння.

Гіроскопічна частина MPU6050 вимірює кутову швидкість обертання також по трьох осях - X, Y та Z - і відображає її в градусах за секунду (°/сек). Ці значення можна задавати вручну в межах діапазону ± 250 °/сек, що дозволяє моделювати обертання або повороти пристрою навколо кожної з осей. Наприклад, при збільшенні значення по осі Z можна змоделювати обертання навколо вертикальної осі, що часто використовується в проектах зі стабілізацією або орієнтацією. У показаній конфігурації значення кутової швидкості для всіх осей дорівнює 0 °/сек, тобто обертання не відбувається, і пристрій умовно перебуває в спокої. Це корисно для початкової калібровки або тестування стану “без руху” в логіці програми. Змінюючи параметри, можна перевірити, як реагує мікроконтролер на різні сценарії руху.

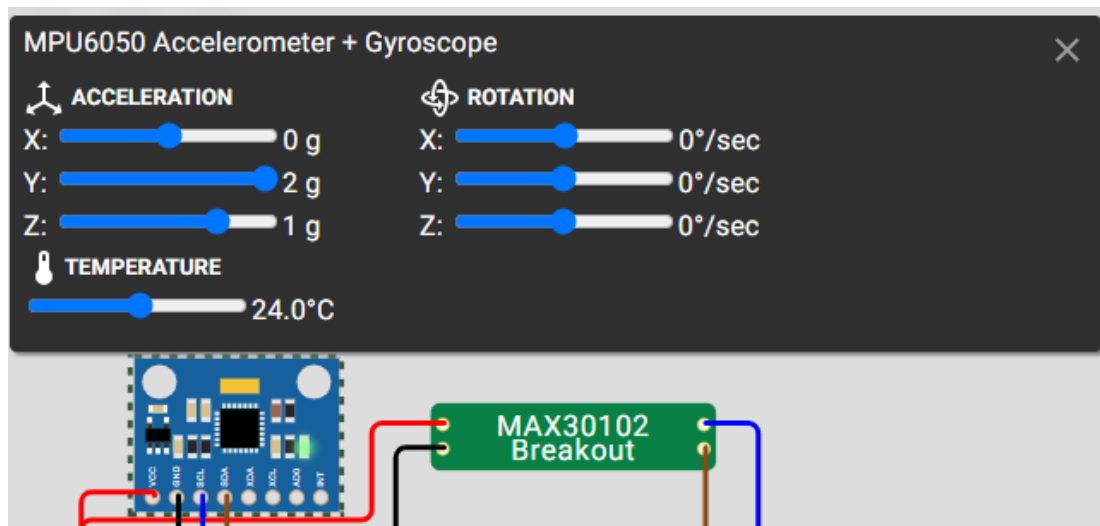


Рисунок 3.3 – Налаштування параметрів модуля MPU6050

Додатково доступний датчик температури, який за замовчуванням встановлений на 24.0°C. Це значення також можна змінювати вручну, що корисно

при перевірці реакції програми на зміну температури, оскільки MPU6050 має вбудований температурний сенсор. Усі ці параметри налаштовуються через вікно інтерфейсу, що з'являється при натисканні на сам модуль в середовищі Wokwi (рис. 3.3).

Даний модуль буде використано для емуляції здійснення руху, та відповідно підрахування кількості кроків людиною. Змінюючи параметри модуля можна встановити значення руху та спокою об'єкта та здійснивши певні перетворення значень X, Y та Z рахувати кількість кроків.

Однією з ключових переваг модуля ESP32 є вбудований модуль Wi-Fi, який дозволяє мікроконтролеру безпосередньо підключатися до бездротових мереж без потреби в додатковому обладнанні. Це значно спрощує реалізацію IoT-проектів, у яких необхідно передавати дані на сервер, отримувати інформацію з Інтернету або взаємодіяти з хмарними платформами, такими як Adafruit IO, ThingSpeak, Blynk тощо. Наявність Wi-Fi забезпечує зручне та гнучке підключення до локальних мереж або Інтернету, що робить ESP32 ідеальним рішенням для створення розумних пристроїв, які потребують віддаленого моніторингу або керування.

Крім цього, Wi-Fi-модуль відкриває можливості для створення локальних веб-серверів прямо на ESP32. Це означає, що мікроконтролер може обслуговувати веб-інтерфейси для перегляду даних з датчиків у браузері або навіть для керування пристроєм у режимі реального часу. Такий підхід дозволяє створювати повноцінні системи збору, обробки та візуалізації інформації без необхідності у складній інфраструктурі. Завдяки вбудованому Wi-Fi ESP32 дозволяє реалізовувати як прості, так і складні бездротові проекти з мінімальними затратами часу, коштів і ресурсів.

Підключення до мережі WiFi в симуляторі Wokwi є досить простим, адже мережа буде внутрішня, хоча параметри для її налаштування в коді програми сприймаються в повній мірі.

3.2 Налаштування середовища розробки ПЗ

3.2.1 Опис середовища розробки ПЗ

Arduino IDE (Integrated Development Environment) – це інтегроване середовище розробки, призначене для створення, редагування, компіляції та завантаження програмного коду до сімейства мікроконтролерів Arduino та сумісних платформ, таких як ESP32. Це зручне середовище, яке забезпечує користувачам взаємодію з апаратним забезпеченням через простий інтерфейс, який включає редактор коду, послідовний монітор і інструмент для завантаження мікропрограми на плату (рис. 3.4).



Рисунок 3.4 – Платформа Arduino IDE

3.2.2 Налаштування Arduino IDE для роботи з модулем ESP32

Щоб налаштувати Arduino IDE для роботи з мікроконтролером ESP32, потрібно виконати кілька простих кроків. Першим кроком є додавання посилання на спеціальний конфігураційний файл, що містить інформацію про платформу

					КВРКІ 210243.21.02.58 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

ESP32. Це посилання дозволяє Arduino IDE отримати доступ до пакета підтримки платформи ESP32, включаючи всі доступні плати від Espressif (рис. 3.5).

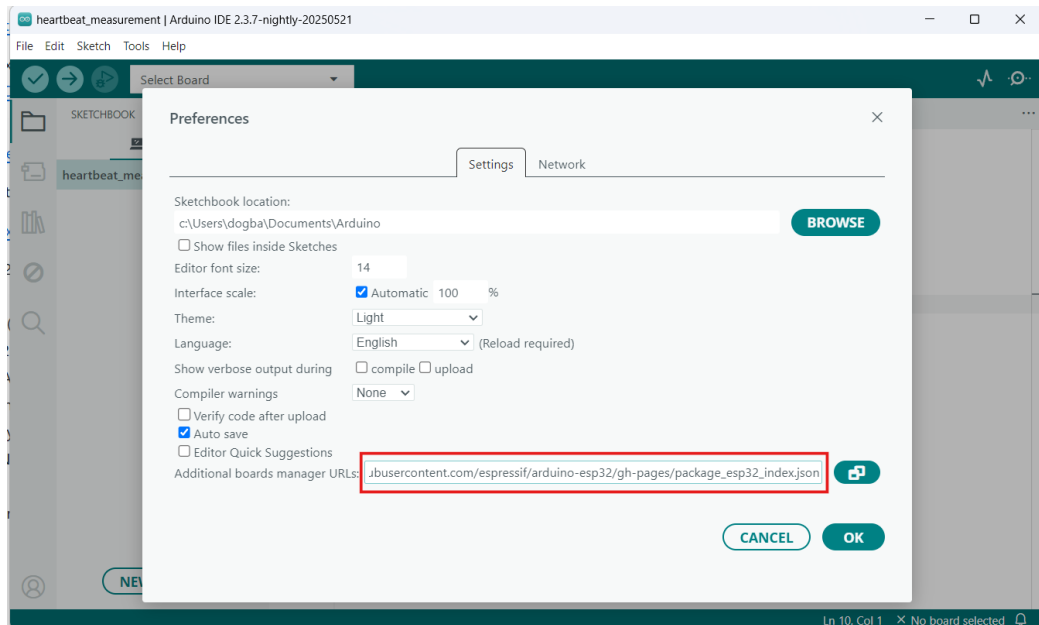


Рисунок 3.5 – Налаштування у вікні Preferences в Arduino IDE для роботи з модулем ESP32

Після додавання посилання на файл у налаштуваннях Arduino IDE наступним кроком є встановлення пакета підтримки для мікроконтролера ESP32 через диспетчер плат.

Для цього відкрийте меню "Диспетчер плат" і в полі пошуку введіть запит "ESP32". Результати пошуку відобразять "esp32 від Espressif Systems". Потрібно вибрати його та натиснути кнопку «Встановити», щоб встановити необхідний пакет підтримки (рис. 3.6).

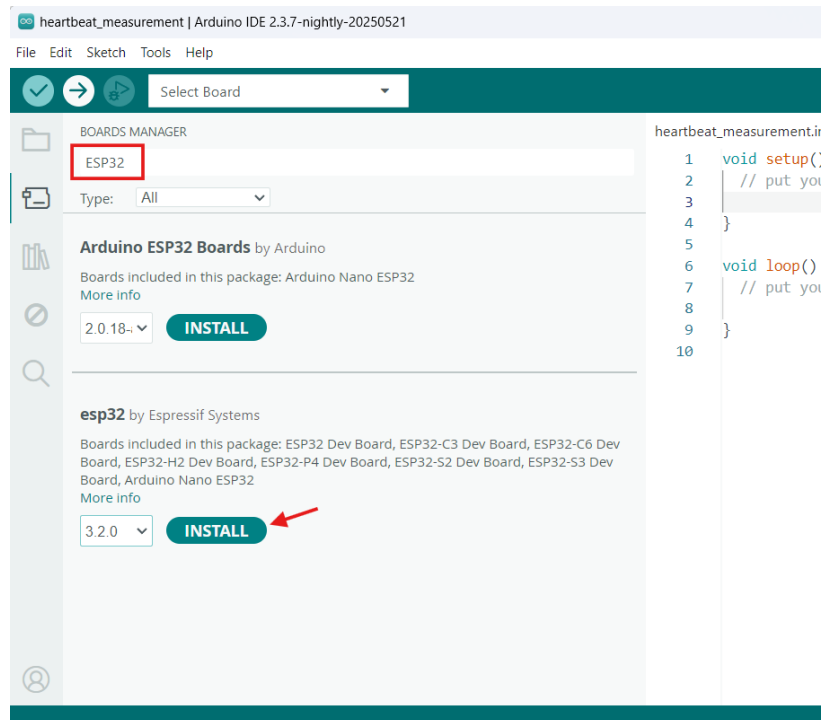


Рисунок 3.6 – Встановлення додатку ESP32

Після завершення інсталяції додатку для підтримки плати вам слід зайти в меню "Tools", далі "Board" і знайти в списку плат розділ «esp32», де потрібно вибрати «ESP32 Dev Module» – це одна з найпопулярніших і широко використовуваних плат для початку роботи з ESP32 (рис. 3.7).

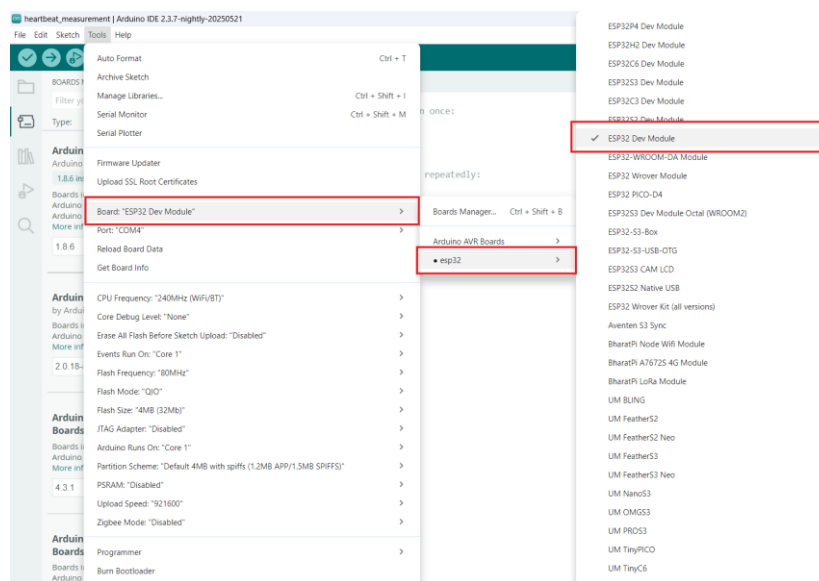


Рисунок 3.7 – Вибір платформи ESP в Arduino IDE

Після цього налаштування Arduino IDE завершено, і модуль ESP32 повністю готовий до завантаження вашого коду.

3.2.3 Встановлення бібліотек

Щоб інсталювати бібліотеку Adafruit MQTT до Arduino IDE, спочатку потрібно перейти до розділу керування бібліотекою, для цього відкрийте меню "Include Library" і виберіть там пункт "Manage Libraries", коли відкриється вікно менеджера бібліотеки, введіть назву бібліотеки - Adafruit MQTT - у верхньому рядку пошуку. Натиснувши кнопку "Install", користувач починає процес завантаження та інсталяції бібліотеки. Після встановлення бібліотеки вона буде автоматично додано до середовища, та доступна для використання в будь-яких нових або існуючих проектах (рис. 3.8).

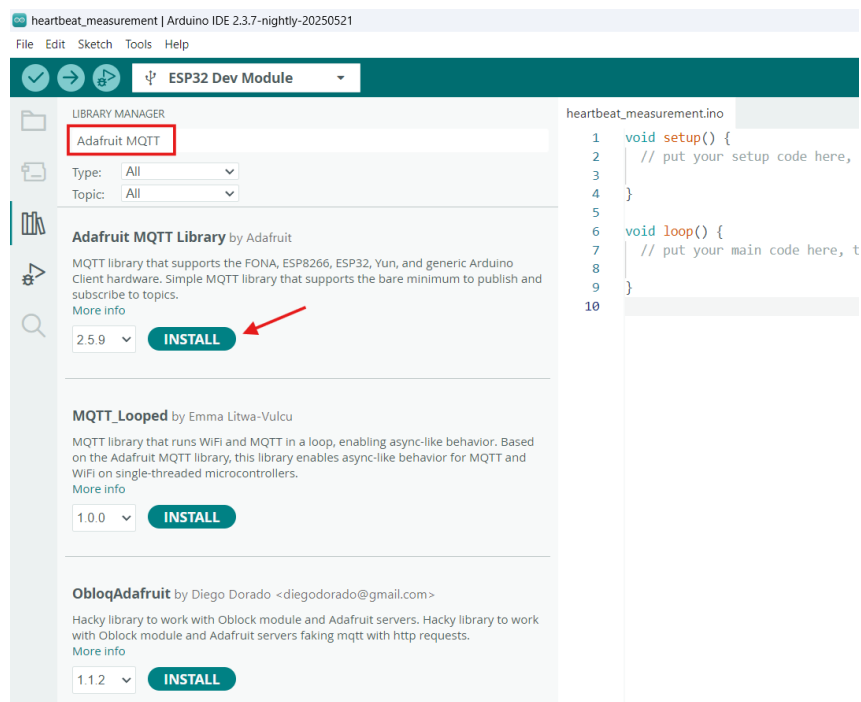


Рисунок 3.8 – Встановлення бібліотеки Adafruit MQTT

Бібліотека Adafruit MQTT в Arduino IDE призначена для організації обміну даними між мікроконтролером (в нашому випадку ESP32) та MQTT-брокером.

Вона дозволяє легко підключатися до сервісу Adafruit IO або будь-якого іншого MQTT-брокера через Wi-Fi, використовуючи простий інтерфейс для публікації (publish) і підписки (subscribe) на топіки. Завдяки бібліотеці користувачі можуть надсилати дані з сенсорів у хмару, керувати пристроями дистанційно або візуалізувати дані на веб-інтерфейсі без написання великої кількості коду низького рівня.

Однією з ключових особливостей є інтеграція з бібліотекою Adafruit MQTT Client, яка підтримує автоматичне повторне з'єднання з Wi-Fi та MQTT, обробку збоїв мережі й підтримку QoS 0/1. Крім того, бібліотека надає об'єкти типу Adafruit_MQTT_Publish і Adafruit_MQTT_Subscribe, які значно спрощують структуру програми. Передбачено можливість просто створити об'єкт для публікації в конкретний топик, наприклад feeds/pulse, і викликати метод .publish(value), щоб надіслати дані. Так само легко налаштовується обробка вхідних повідомлень з Adafruit IO або іншого сервера.

У використанні з Arduino IDE ця бібліотека добре працює з ESP32, але вимагає попереднього налаштування Wi-Fi та MQTT-параметрів (ім'я користувача, ключ API, URL брокера). Важливо правильно налаштувати циклічну перевірку з'єднання в loop(), а також виклик методу mqtt.read() для обробки вхідних повідомлень. Оскільки бібліотека не використовує окремий потік або переривання, вона потребує постійної уваги в основному циклі. Завдяки цьому вона залишається простою, сумісною з більшістю плат і легко адаптується до різних IoT-завдань.

3.3 Візуалізація даних за допомогою MQTT брокера

MQTT (Message Queuing Telemetry Transport) – це легкий мережевий протокол обміну повідомленнями, розроблений спеціально для пристроїв з обмеженими ресурсами та нестабільними мережевими з'єднаннями. Його загальний принцип роботи базується на архітектурі "видавець-підписник" (publish-subscribe), де центральним елементом є MQTT брокер. Брокер виконує роль

посередника між пристроями: він приймає повідомлення від видавців (publishers) і доставляє їх усім підписникам (subscribers), які зацікавлені у відповідній темі (topic). Видавець надсилає дані на певну тему, а всі пристрої, що підписані на цю тему, автоматично отримують повідомлення через брокера.

Цей підхід дозволяє розділити джерела даних та їх споживачів, що забезпечує високу гнучкість і масштабованість систем.

Для використання MQTT брокера Adafruit IO виконуємо вхід на сайт, створивши там акаунт, отримуємо параметри для доступу, натиснувши кнопку «API key». Далі створюємо три канали у вкладці «Feeds» і дамо їм назви pulse, spo2 та steps, вазавши відповідні коментарі про призначення каналу.

MQTT брокер працює досить просто, отримуючи данні та відображаючи їх на відповідних вкладках, згідно закладеного алгоритму (рис. 3.9).



Рисунок 3.9 – Алгоритм роботи MQTT брокера

Ініціалізація модуля Adafruit потребує наявності параметрів доступу до брокера, які можна отримати на сайті Adafruit після проходження процедури реєстрації. Активація параметрів відбувається командою:

```
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
```

Параметри запуску брокера задають попередньо відповідних константах:

AIO_SERVER "io.adafruit.com" – веб-адреса брокера;

AIO_SERVERPORT 1883 – порт доступу брокера;

AIO_USERNAME, AIO_KEY – акаунт та пароль доступу до брокера.

Для організації зв'язку нашого апаратно-програмного комплексу та MQTT брокера в програмі застосовано три величини pulseFeed, spo2Feed, stepsFeed, які привязані до величин, розміщених на каналах pulse, spo2, steps створеного брокера відповідно.

Для здійснення прив'язки скористаємося командами модуля Adafruit MQTT Library:

```
Adafruit_MQTT_Publish pulseFeed = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/pulse");
```

```
Adafruit_MQTT_Publish spo2Feed = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/spo2");
```

```
Adafruit_MQTT_Publish stepsFeed = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/steps");
```

Для безпосередньої передачі величин отриманих від датчиків використаємо відповідні команди модуля Adafruit MQTT Library:

```
pulseFeed.publish((int32_t)simulatedPulse);
```

```
spo2Feed.publish((int32_t)simulatedSpO2);
```

```
stepsFeed.publish((int32_t)stepCount);
```

При запуску емуляції, інформація одразу починає надходити у відповідні канали нашого брокера, з отриманням значення та дати та часу її запису на канал (рис 3.10).

					КВРКІ 210243.21.02.58 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

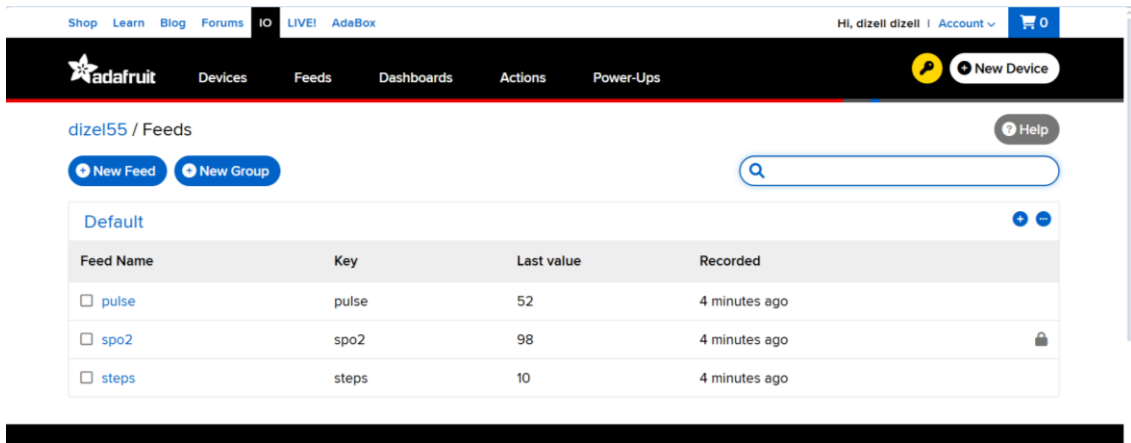


Рисунок 3.10 – Канали даних MQTT брокера

Для отримання аналізу значень каналу, слід клікнути по необхідному каналу. Після цього відкриється вкладка з історичними значеннями даної величини, які відображаються у формі графіка, та таблиці значень (рис 3.11).

В налаштуваннях каналу доступні наступні параметри: встановлення приватності чи публічності транслявання значень каналу, обмеження зберігання значень, що залежать від наявної ліцензії, можливість завантаження історичної інформації та встановлення фільтрування даних для виведення певних часових проміжків отриманих даних. Skorиставшись додатковими налаштуваннями ми можемо обмежити передачу даних на мобільні пристрої, або заборонити її взагалі.

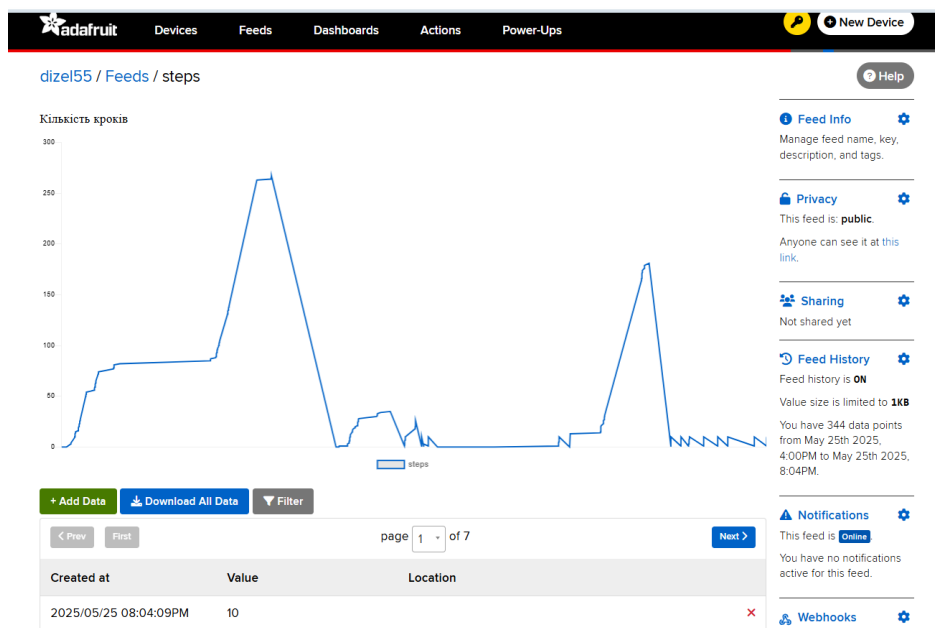


Рисунок 3.11 – Канал даних MQTT брокера

Dashboards в Adafruit IO – це інтерактивні інформаційні панелі, які дозволяють візуально відображати, керувати та взаємодіяти з даними з каналів (feeds) у зручному графічному інтерфейсі. Вони створюються користувачем для моніторингу стану пристроїв, перегляду сенсорних даних у реальному часі та навіть для віддаленого керування обладнанням. Кожна панель складається з віджетів (widgets), які прив'язуються до конкретного каналу та можуть показувати дані у вигляді графіка, гістограми, цифрового індикатора, повзунка, перемикача тощо.

Dashboards є особливо корисними в проектах IoT, де потрібно наочно відображати, наприклад, рівень пульсу, SpO2 або кількість кроків, отриманих з ESP32 через MQTT. Користувач може налаштувати кольори, розміри, діапазони значень, період оновлення та навіть створювати елементи керування (наприклад, кнопки для вмикання реле або світлодіодів). Панелі можна відкривати з будь-якого пристрою з доступом до Інтернету, що дозволяє в реальному часі стежити за показниками з будь-якої точки світу.

dizel55 / Dashboards / Bogdan

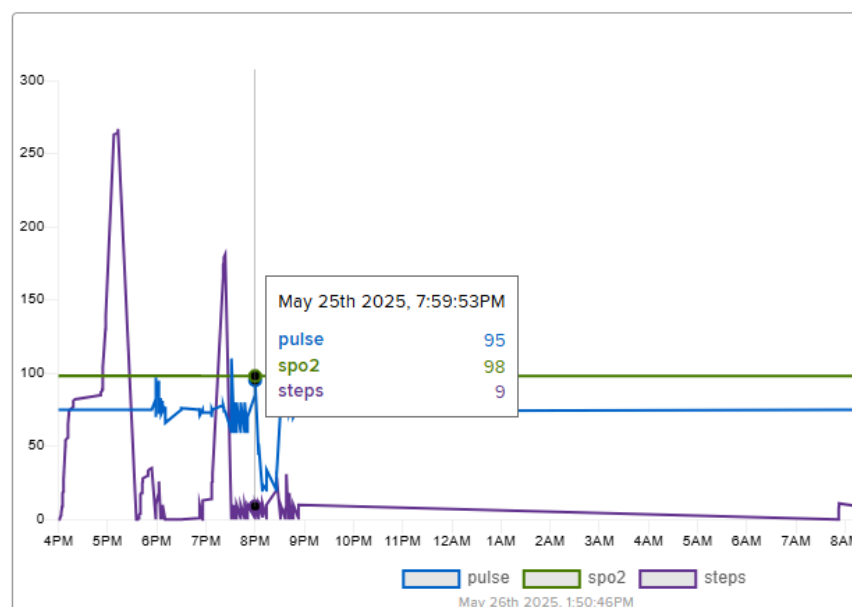


Рисунок 3.12 – Панель з даними MQTT брокера

Згенерована панель для нашого проекту відображає отримані параметри у зручній формі та надає можливість аналізу отриманої інформації про залежність пульсу та рівня кисню в залежності від кількості кроків (рис 3.12).

Для налаштування параметрів відображення скористаємося кнопкою «Параметри». Тут можна вибрати термін відображення на дошці отримуваних величин в межах від 1 години до 60 днів або ж актуальні данні онлайн. Також в даному підменю можна налаштувати додаткові параметри відображення та увімкнути блокування дошки для особистого використання. (рис 3.13)

Використання інтерактивної дошки (dashboard) для визначення пульсу та рівня кисню (SpO_2) в залежності від кількості пройдених кроків має кілька важливих переваг як для розробників, так і для користувачів системи моніторингу здоров'я. По-перше, така дошка дозволяє візуалізувати зв'язок між фізичним навантаженням і станом організму в реальному часі. Коли користувач бачить, як змінюються пульс і SpO_2 при збільшенні кількості кроків, це дає змогу оперативної оцінки реакції організму на активність, а також виявити потенційні відхилення - наприклад, занадто сильне підвищення пульсу або зниження кисню в крові після незначного навантаження.

По-друге, така система надає можливість довготривалого аналізу та самоспостереження, що особливо важливо для людей з серцево-судинними або дихальними захворюваннями. Дані, зібрані на дошці протягом дня або тижня, можуть показати загальні тенденції — як часто підвищується пульс, скільки кроків в середньому проходить користувач, чи відновлюється рівень кисню після фізичних вправ. Це дозволяє приймати обґрунтовані рішення щодо зміни режиму дня, рівня активності або навіть звернення до лікаря. Крім того, такі панелі можуть бути корисні для навчання, досліджень або спортивних програм, де потрібне точне і наочне порівняння навантаження та фізіологічної відповіді.

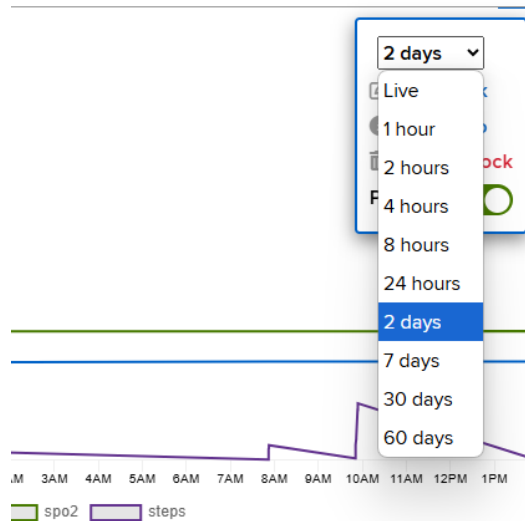


Рисунок 3.13 – Налаштування параметрів відображення для дошки

Панель чудово адаптується до відображення на мобільних пристроях, таких як смартфони або планшети, що дозволяє користувачу контролювати стан свого здоров'я будь-де і будь-коли. Інтерфейс оптимізований для мобільних браузерів: всі віджети автоматично підлаштовуються під розмір екрана, зберігаючи функціональність та читабельність. Користувач може швидко переглядати показники пульсу, рівня кисню в крові (SpO₂) та кількості кроків у вигляді графіків, цифрових дисплеїв чи кольорових індикаторів прямо зі свого телефону.

3.4 Візуальний зворотній зв'язок апаратно-програмного засобу

Для здійснення керуванням апаратно-програмним засобом передбачена кнопка запуску та зупинки роботи пристрою. При ввімкненні пристрою, для візуального контролю його роботи, передбачено виведення інформації значень датчиків на дисплей (рис 3.14). Для інформування буде відображатися поточне значення пульсу, рівень кисню та кількість кроків, що пройшов користувач з моменту ввімкнення пристрою.

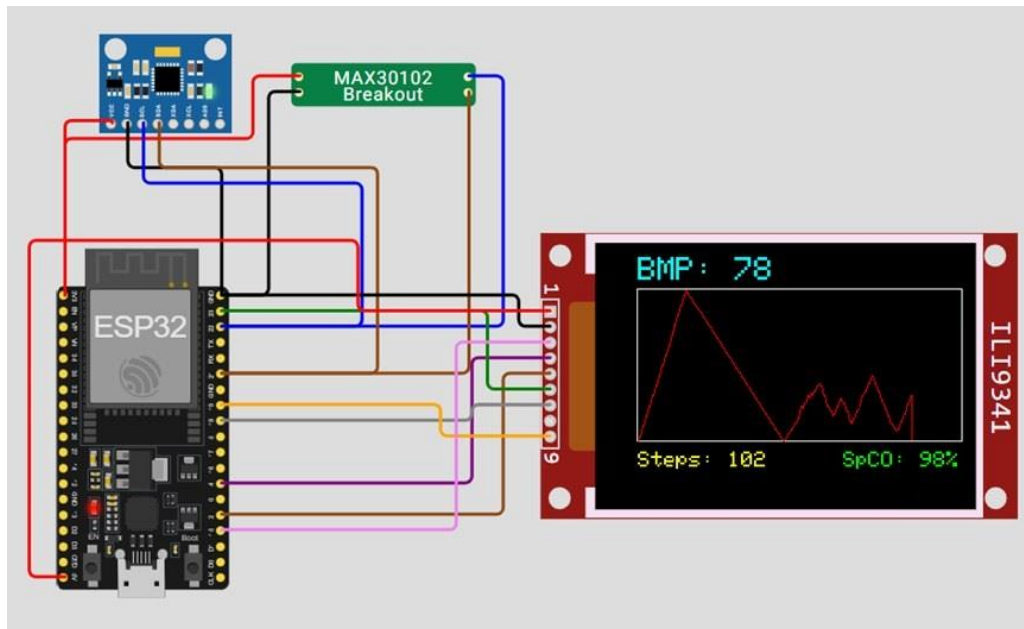


Рисунок 3.14 – Результат відображення на екрані апаратно-програмного засобу

Щоб покращити візуальне сприйняття інформації, на дисплеї будується графік зміни серцевого ритму, який відображає історичні данні збережені попередньо в зарезервованому масиві даних. Графік будується у відносних координатах без збереження пропорційності, але дозволяє спостерігати фактичну різку зміну пульсу дослідного об'єкта та реагувати на різкі зміни значення.

Для реалізації засобу візуального відображення, складемо алгоритм роботи цієї системи, врахувавши її незалежність від основної системи передачі даних через мережу WiFi використовуючи протокол MQTT (рис. 3.15).

Алгоритм, використовує для своєї роботи масив, у якому зберігаються значення пульсу, отримані від сенсора. Починається все з ініціалізації, тобто створення масиву та встановлення початкових значень, у нашому випадку це буде значення 100. Це потрібно для того, щоб при побудові графіка для початкових значень, після увімкнення приладу, графік мав передбачувану будову, зокрема пряма лінія посередині дисплея. Після ініціалізації програма переходить до основного циклу, де постійно перевіряє: чи досягнута межа заповнення масиву. Наприклад, якщо масив розрахований на 30 значень, то на початку заповнюються всі елементи по черзі, без зсувів.

Коли масив ще не заповнений, нове значення пульсу додається у наступну порожню позицію. Але як тільки останній елемент заповнений, алгоритм змінює поведінку: тепер перед тим, як додати нове значення, всі наявні дані в масиві зсуваються на одну позицію вліво. Таким чином, найстаріше значення (яке стоїть на початку масиву) видаляється, і в кінці масиву звільняється місце для нового пульсового значення. Це реалізація принципу циклічного буфера, що особливо корисно для аналізу змін у часі або побудови графіків в реальному часі, не перевантажуючи пам'ять новими і новими даними.



Рисунок 3.15 – Алгоритм роботи системи візуалізації даних

Після кожного додавання нового значення в масив, відбувається виведення всього масиву на екран. Така схема дозволяє візуально контролювати зміни в пульсі користувача у реальному часі, а також забезпечує постійну актуалізацію даних, зберігаючи останні 30 вимірів. Після цього процес повторюється: система знову перевіряє, чи заповнений масив, і діє відповідно - або просто додає нове значення, або спочатку зсуває масив.

Для збереження значень вимірюного пульсу, використаємо масив цілих чисел `pulseData`. Ініціалізація масив передбачає занесення в масив початкових значень, для чого можна скористатися простим циклом:

```
for (int i = 0; i < 30; i++) {pulseData[i] = 100;}.
```

Але при значній кількості елементів масиву, робота такого циклу значно сповільнює роботу контролера. Тому для виконання цієї роботи ми скористаємося наступною функцією:

```
std::fill (pulseData, pulseData + 30, 100);
```

`std::fill` - це стандартна функція з бібліотеки `<algorithm>` у мові C++, яка дозволяє швидко заповнити частину або весь масив (чи інший контейнер) одним і тим самим значенням. Вона має три аргументи: початок діапазону, кінець діапазону (не включно) та значення, яке потрібно присвоїти всім елементам. Наприклад, у нашому випадку, вона встановить усі 30 елементів масиву `pulseData` рівними 100. Ця команда є зручною, компактною та ефективною альтернативою ручному циклу `for`, особливо при ініціалізації або скиданні даних у масиві. Перевагою застосування цієї функції є апаратна підтримка її контролером ESP32.

Для визначення заповнення масиву елементами, що вимірюються, скористаємося простим методом підрахунку кількості вимірювань, що здійснено, використавши змінну `count`. Заповнення масиву визнаємо тоді, коли кількість вимірів стане рівною кількості елементів масиву. Такий принцип, дозволяє відмовитися від додаткових функцій визначення заповнення масиву елементами та пришвидшить роботу контролера.

```
if (count < (29)) count ++;
else
```

```

{
    //Зсув елементів масиву
};

```

Для виконання команди зсуву значень масиву на одиницю, скористаємося функцією:

```
memmove(pulseData, pulseData + 1, (29) * sizeof(int));
```

Дана функція являється стандартною функцією мови C/C++, яка копіює блок пам'яті з одного місця в інше. Її синтаксис та вигляд: `memmove(void* dest, const void* src, size_t n);`, де `dest` — адреса призначення, `src` — джерело, а `n` — кількість байтів для копіювання. На відміну від `memcpy`, `memmove` безпечно працює навіть тоді, коли області пам'яті перекриваються, тому вона ідеальна для задач зі зсувами в масиві, коли старі значення зміщуються, а нові додаються.

На ESP32 ця функція дуже корисна в задачах обробки потоків даних, наприклад, при роботі з сенсорами або буферами. Завдяки апаратній підтримці копіювання пам'яті, `memmove` працює швидше, ніж ручний цикл `for`, особливо на великих масивах. Вона знижує навантаження на процесор і робить код компактним і легким для читання. Саме тому вона часто використовується в проектах, пов'язаних із реальним часом - як-от аналіз пульсу, прискорення або температури, де потрібно зберігати лише останні `N` значень.

У нашому випадку відбувається зсув масиву `pulseData` на одну позицію вліво: 29 значень, починаючи з другого елемента (`pulseData + 1`), копіюються на позиції з нуля (`pulseData`). Таким чином, найстаріше значення з масиву "випадає", і звільняється місце в кінці масиву для запису нового значення. Такий підхід широко застосовується для реалізації зміщення даних, зберігання останніх вимірів або створення історії зміни сигналу.

Побудову графіка реалізуємо в використанні циклу, в якому вісь `X` є параметром циклу, і обраховується автоматично, а значення по осі `Y` береться безпосередньо з масиву.

```

for (int i = 0; i < 29; i++) {
    int x0 = graphX + i*10;

```

```
int y0 = pulseData[i];  
int x1 = graphX + 10 + i*10;  
int y1 = pulseData[i+1];  
tft.drawLine(x0, y0, x1, y1, ILI9341_RED)
```

Застосування масиву дає можливість отримати ефект плаваючого графіка, дані на якому зміщуються вліво в процесі старіння інформації. Такий формат дозволяє моніторити стан серцевого ритму протягом певного відрізка часу.

Даний графік не відображає повної картини зміни пульсу людини, а генерується лише з моменту ввімкнення живлення приладу, та використовується для поточного аналізу стану серцебиття. Для більш глибокого аналізу слід користуватися MQTT брокером, який має здатність зберігати отримані данні протягом 60 днів.

3.5 Висновки до третього розділу

У межах третього розділу здійснено реалізацію апаратної та програмної складової апаратно-програмного засобу для вимірювання серцевого ритму та аналізу отриманих даних на основі ESP32 з підтримкою бездротового зв'язку.

В процесі реалізації створено модель апаратної складової апаратно-програмного засобу, яка дозволила проаналізувати роботу програми та апаратних частин в умовах передачі даних про стан здоров'я користувача за допомогою засобів бездротового зв'язку. Моделювання дозволило передбачити можливі недоліки апаратної схеми пристрою та узагальнити візуалізацію процесів за допомогою дисплею.

При реалізації системи дистанційного відображення даних, які вимірює пристрій, було використано протокол MQTT. У розділі докладно описано принципи використання MQTT брокера для реалізації відображення параметрів, які отримує апаратно-програмний засіб. Описано порядок створення каналів та описано можливі параметри відображення інформації з використанням дошки приладів.

Описано також технологію підключення брокера від Adafruit для використання його ресурсів при створенні моделі на платформі Wokwi та при складанні коду на платформі Arduino IDE.

Отже, використання Adafruit IO у поєднанні з ESP32, датчиком пульсу (MAX30102) і акселерометром (MPU6050) дає змогу створити зручну та функціональну систему для моніторингу здоров'я в реальному часі. Завдяки MQTT-зв'язку дані про пульс, SpO2 та кількість кроків передаються на хмарні канали, де можуть бути легко візуалізовані за допомогою інтерактивної панелі (dashboard). Користувач має змогу швидко оцінювати свій фізичний стан, аналізувати динаміку змін і виявляти важливі залежності між активністю та фізіологічними показниками.

Особливою перевагою є адаптивність панелі до мобільних пристроїв, що забезпечує постійний доступ до важливої інформації без прив'язки до стаціонарного комп'ютера. Такий підхід підвищує зручність використання та дозволяє контролювати стан здоров'я навіть під час руху або фізичних навантажень. У цілому, ця система є ефективним інструментом для особистого моніторингу, спортивного аналізу або навіть медичних застосунків, де важлива простота, доступність і наочність представлення даних.

Система візуального відображення інформації за допомогою екрану працює в режимі швидкого відображення найбільш актуальних фізіологічних даних людини. Вона дозволяє користувачу орієнтуватися на стан серцебиття при виконанні вправ або занятті спортом.

					КвРКІ 210243.21.02.58 ПЗ	Арк. 59
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Виконавши теоретичні та практичні дослідження, було розроблено апаратно-програмний засіб для вимірювання серцевого ритму та аналізу отриманих даних на основі ESP32 з підтримкою бездротового зв'язку. В процесі роботи вдалося розробити практичну схему пристрою контролю серцевого ритму людини та визначити систему реалізації дистанційного аналізу отриманих даних.

У першому розділі проведено аналіз наявних кіберфізичних систем, встановлені основні напрямки розвитку систем вимірювання фізіологічних параметрів людини та основних принципів для аналізу та відображення цієї інформації. В даному розділі вироблено технічні завдання для реалізації завдань дослідження.

У другому розділі виконано вибір апаратних засобів застосованих для реалізації апаратно-програмного засобу, визначено принципи отримання даних з використанням мережі WiFi та протоколу MQTT. Також розроблено алгоритм та структурну схему програмно-апаратного засобу. Визначено основні етапи та позитивні сторони виконання алгоритму з використанням вибраних програмних засобів.

У третьому розділі розглянуто алгоритм та принципи розробки моделі у симуляторі Wokwi. Розроблено алгоритм роботи програмно-апаратного засобу в умовах та обмеженнях симулятора. Описано практичне створення каналів відображення та візуалізації інформації отриманої з програмно-апаратного засобу за допомогою MQTT брокера Adafruit. Вироблено принципи використання платформи Adafruit при створенні програмного коду в середовищі розробки Arduino IDE. Для кращої візуалізації інформації, що надається апаратною частиною системи, виконано подання параметрів на дисплеї у формі графіка.

					КВРКІ 210243.21.02.58 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Інтернет медичних речей (ІоМТ) для ортопедії в умовах пандемії COVID-19: ролі, проблеми та застосування. URL: <https://stfalcon.com/uk/blog/post/iot-in-healthcare-benefits-challengesWearable> (дата звернення: 28.04.2025)
2. Домрачева К. О., Довженко Н. М., Дмитренко В. В. Аналіз технологій та стандартів зв'язку для мережі ІоТ Наук. зап. Укр. наук.–дослід. ін–ту зв'язку. 2019. с. 54–62
3. Charlton P. H., Kyriacou P. A., Mant J., Marozas V., Chowienczyk P. and Alastruey J. Wearable Photoplethysmography for Cardiovascular Monitoring. *Proceedings of the IEEE*, vol. 110, no. 3, P. 355-381, March 2022, URL: doi: 10.1109/JPROC.2022.3149785
4. Alsheikhy A., Said Y., Shawly T., Lahza H., A Model to Predict Heartbeat Rate Using Deep Learning Algorithms – PMC. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC9914604/> (дата звернення: 28.04.2025)
5. Amat S., Busquier S., Algorithm-Based Real-Time Analysis of Heart Rate Measures in HIIT Training: An Automated Approach. URL: <https://www.mdpi.com/2076-3417/15/9/4749> (дата звернення: 28.04.2025)
6. Решетов В., Пристрої для відстеження серцевого ритму: Баланс між точністю та комфортом. URL: <https://bimedis.net/latest-news/browse/754/pristroyi-dlya-vidstezhennya-serceвого-ritmu-balans-mizh-tochnistyu-ta-komfortom> (дата звернення: 28.04.2025)
7. Heart Rate Reserve: How to Calculate It & What It Means - Cleveland Clinic. URL: <https://my.clevelandclinic.org/health/articles/24649-heart-rate-reserve> (дата звернення: 28.04.2025)
8. Koperska M., Bowater J., Heart Rate Calculator. URL: <https://www.omnicalculator.com/sports/heart-rate> (дата звернення: 28.04.2025)
9. Gajda R., Gajda J., Czuba M., Knechtle B., Drygas W., Sports Heart Monitors as Reliable Diagnostic Tools for Training Control and Detecting Arrhythmias in Professional and Leisure-Time Endurance Athletes: An Expert Consensus Statement.

					КВРКІ 210243.21.02.58 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10799155/> (дата звернення: 28.04.2025)

10. Wu M., Luo J., Wearable Technology Applications in Healthcare: A Literature Review – HIMSS. URL: <https://legacy.himss.org/resources/wearable-technology-applications-healthcare-literature-review> (дата звернення: 28.04.2025)

11. Yates D., Wearable Technology Continuously Monitors Heart-rate Recovery to Predict Risk. URL: <https://medicine.illinois.edu/news/wearable-technology-continuously-monitors-heart-rate-recovery-to-predict-risk> (дата звернення: 28.04.2025)

12. Heart Rate Variability Analysis on Electrocardiograms, Seismocardiograms and Gyrocardiograms on Healthy Volunteers – PMC. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC7472094/> (дата звернення: 28.04.2025)

13. Farnsworth B., Heart Rate Variability - How to Analyze ECG Data – iMotions. URL: <https://imotions.com/blog/learning/best-practice/heart-rate-variability/> (дата звернення: 28.04.2025)

14. Cennini G., Arguel J., Akşit K., Heart rate monitoring via remote photoplethysmography with motion artifacts reduction. URL: <https://opg.optica.org/oe/abstract.cfm?uri=oe-18-5-4867>

15. Heart Rate Monitoring in Team Sports <https://pmc.ncbi.nlm.nih.gov/articles/PMC5990631/> URL: (дата звернення: 28.04.2025)

16. Accuracy of Wearable Technology & Smart Watches. URL: <https://www.aim7.com/blog/smartwatch-wearable-technology-accuracy> (дата звернення: 28.04.2025)

17. MAX30102 Data Sheet. URL: <https://www.analog.com/en/products/max30102.html> (дата звернення: 28.04.2025)

18. KY-039 Heartbeat sensor. URL: <https://sensorkit.joy-it.net/en/sensors/ky-039> (дата звернення: 28.04.2025)

19. AD8232 Data Sheet. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf> (дата звернення: 28.04.2025)

					КВРКІ 210243.21.02.58 ПЗ	Арк. 62
Зм.	Арк.	№ докум.	Підпис	Дата		

20. ECG vs. Optical Heart Rate URL: <https://www.wefitter.com/blog/ecg-vs-optical-heart-rate/> (дата звернення: 28.04.2025)

21. Heart Rate Monitor Choose Chest Strap Or Armband – COOSPO URL: <https://www.coospo.com/blogs/knowledge/chest-strap-or-armband-heart-rate-monitor> (дата звернення: 28.04.2025)

22. Nowak M., Why a Chest Strap Is Better than a Wrist Heart Rate Monitor for Your Training. URL: <https://www.aidlab.com/blog/Why-a-Chest-Strap-Is-Better-than-a-Wrist-Heart-Rate-Monitor-for-Your-Training> (дата звернення: 28.04.2025)

23. ALL ABOUT: Chest strap heart rate monitors. URL: <https://www.hodinky-365.com/blog/chest-strap-heart-rate-monitors> (дата звернення: 28.04.2025)

24. The 7 Best Fitness Trackers of 2025 | Tested & Rated - Tech Gear Lab URL: <https://www.techgearlab.com/topics/health-fitness/best-fitness-tracker> (дата звернення: 28.04.2025)

25. Bhatt D. L., Low heart rate warnings via smart watch URL: <https://www.health.harvard.edu/heart-health/low-heart-rate-warnings-via-smart-watch#> (дата звернення: 28.04.2025)

26. Pros and Cons of Smart Rings for Personal Use - Vertu URL: <https://vertu.com/lifestyle/pros-cons-smart-rings-personal-use/> (дата звернення: 28.04.2025)

27. Heart Rate Monitors: How They Work and Accuracy - Cleveland Clinic URL: <https://my.clevelandclinic.org/health/diagnostics/23429-heart-rate-monitor> (дата звернення: 28.04.2025)

28. Remote Cardiac Rhythm Monitoring in the Era of Smart Wearables: Present Assets and Future Perspectives URL: <https://www.frontiersin.org/journals/cardiovascular-medicine/articles/10.3389/fcvm.2022.853614/full> (дата звернення: 28.04.2025)

29. Tamura T., Current progress of photoplethysmography and SPO2 for health monitoring URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC6431353/> (дата звернення: 28.04.2025)

					КВРКІ 210243.21.02.58 ПЗ	Арк. 63
Зм.	Арк.	№ докум.	Підпис	Дата		

30. Smartphone Apps Using Photoplethysmography for Heart Rate Monitoring: Meta-Analysis URL: <https://cardio.jmir.org/2018/1/e4> (дата звернення: 28.04.2025)
31. Accuracy of heart rate apps varies. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC5558137/> (дата звернення: 28.04.2025)
32. Mubarik A., Arshad M, Holter Monitor. URL: <https://www.ncbi.nlm.nih.gov/books/NBK538203/> (дата звернення: 28.04.2025)
33. Schor J., MedTech UX Design: Best Practices and Expert Tips. URL: <https://www.webstacks.com/blog/medtech-ux-design> (дата звернення: 28.04.2025)
34. The Vital Role of User Interface and User Experience in Medical Device Development. URL: <https://nectarpd.com/user-interface-what-is-it-and-why-is-it-important-for-medical-device-development/> (дата звернення: 28.04.2025)
35. Faisal F., Best Strategies to Enhance Medical Device UX Designs. URL: <https://www.aufaitux.com/blog/ux-for-medical-devices/> (дата звернення: 28.04.2025)
36. User Experience Study of the Patient Monitoring Systems Based on Usability Testing and Eye Tracking – MDPI. URL: <https://www.mdpi.com/2227-9032/12/24/2573> (дата звернення: 28.04.2025)
37. Shears A., Keeping Your Medical Device GUIs Healthy – Key Tech. URL: <https://www.keytechinc.com/2023/02/keeping-your-medical-device-guis-healthy/> (дата звернення: 28.04.2025)
38. Choosing displays for medical devices – Riverdi. URL: <https://riverdi.com/blog/choosing-displays-for-medical-devices> (дата звернення: 28.04.2025)
39. Your Guide to Healthcare Data Security – Pacific Health Group. URL: <https://mypacifichealth.com/healthcare-data-protection-and-security-policy/> (дата звернення: 28.04.2025)
40. Karthik S., What is HIPAA Compliant Data Storage & Requirements – Sprinto. URL: <https://sprinto.com/blog/hipaa-compliant-data-storage/> (дата звернення: 28.04.2025)

					КВРКІ 210243.21.02.58 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

41. Data Security For Healthcare: Essential Measures - Haven Health Management. URL: <https://havenhealthmgmt.org/data-security-for-healthcare-essential-measures/> (дата звернення: 28.04.2025)

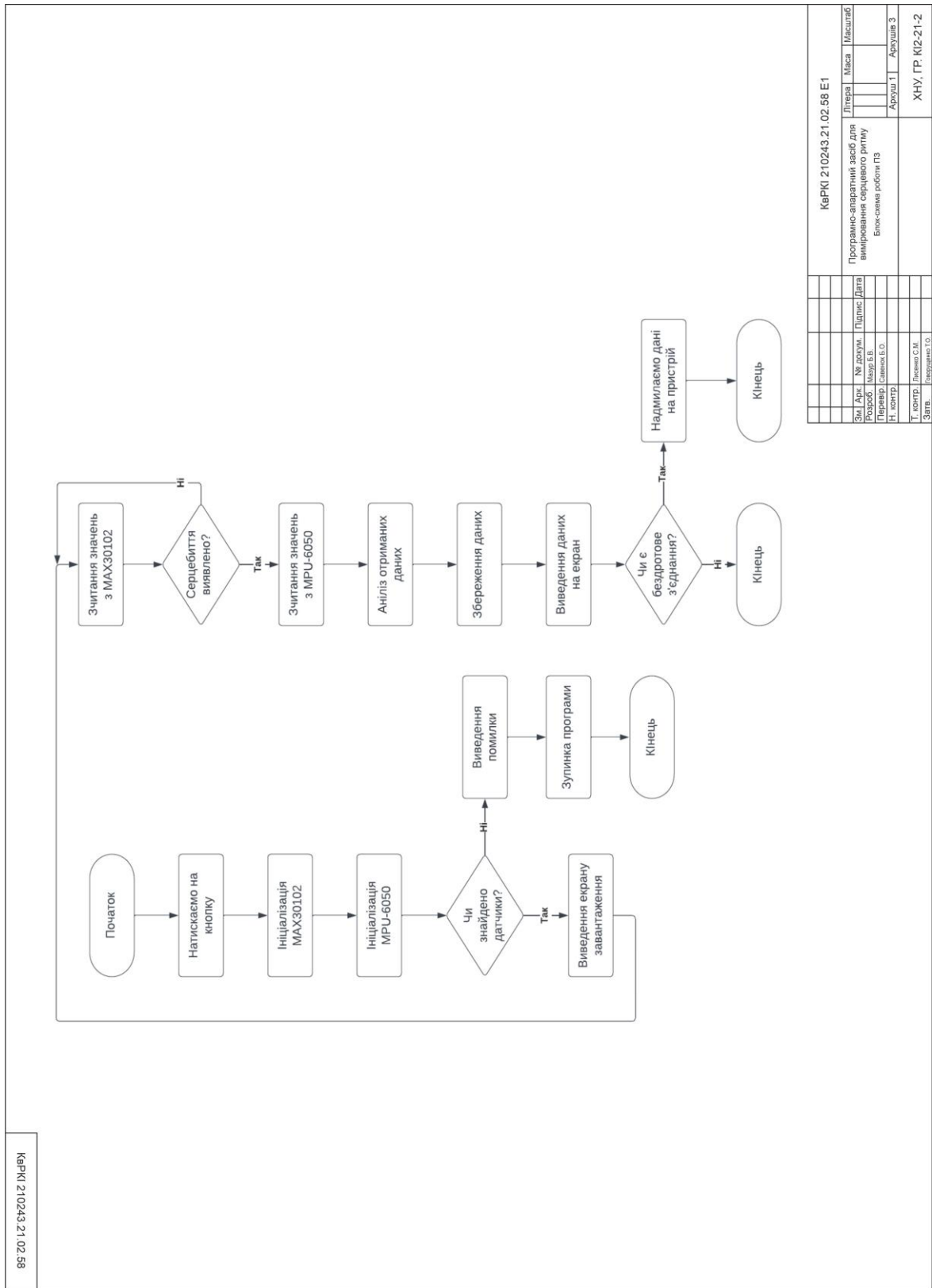
42. Sensor Wiki: KY-039 Heartbeat Sensor. URL: <https://www.thegeekpub.com/wiki/sensor-wiki-ky-039-heartbeat-sensor/>

43. Samsung Galaxy Ring. URL: <https://www.samsung.com/us/rings/galaxy-ring/>

					КВРКІ 210243.21.02.58 ПЗ	Арк. 65
Зм.	Арк.	№ докум.	Підпис	Дата		

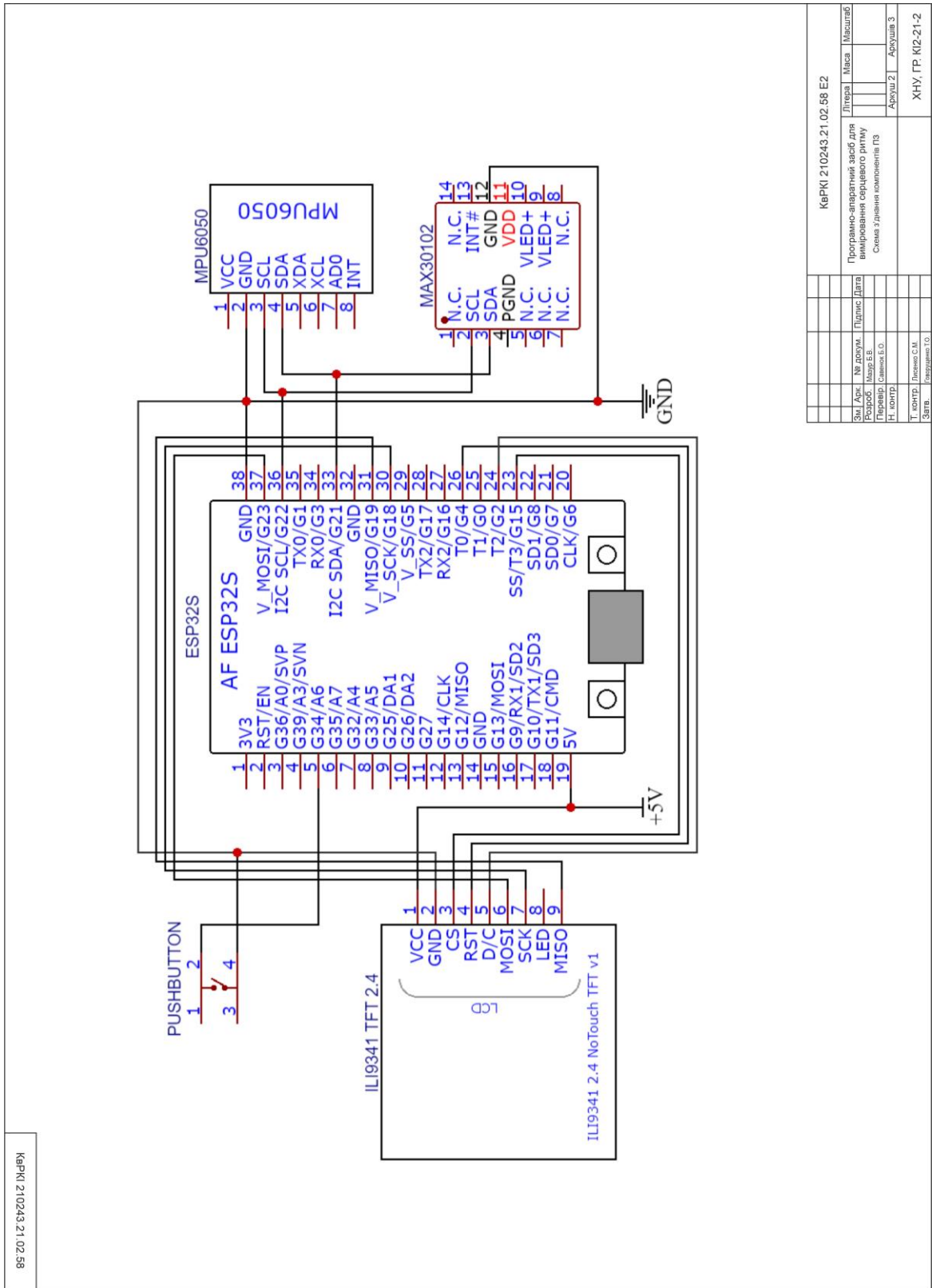
Додаток А (обов'язковий)

Копія креслення «Блок-схема роботи ПЗ»



Додаток Б (обов'язковий)

Копія креслення «Схема з'єднання компонентів ПЗ»

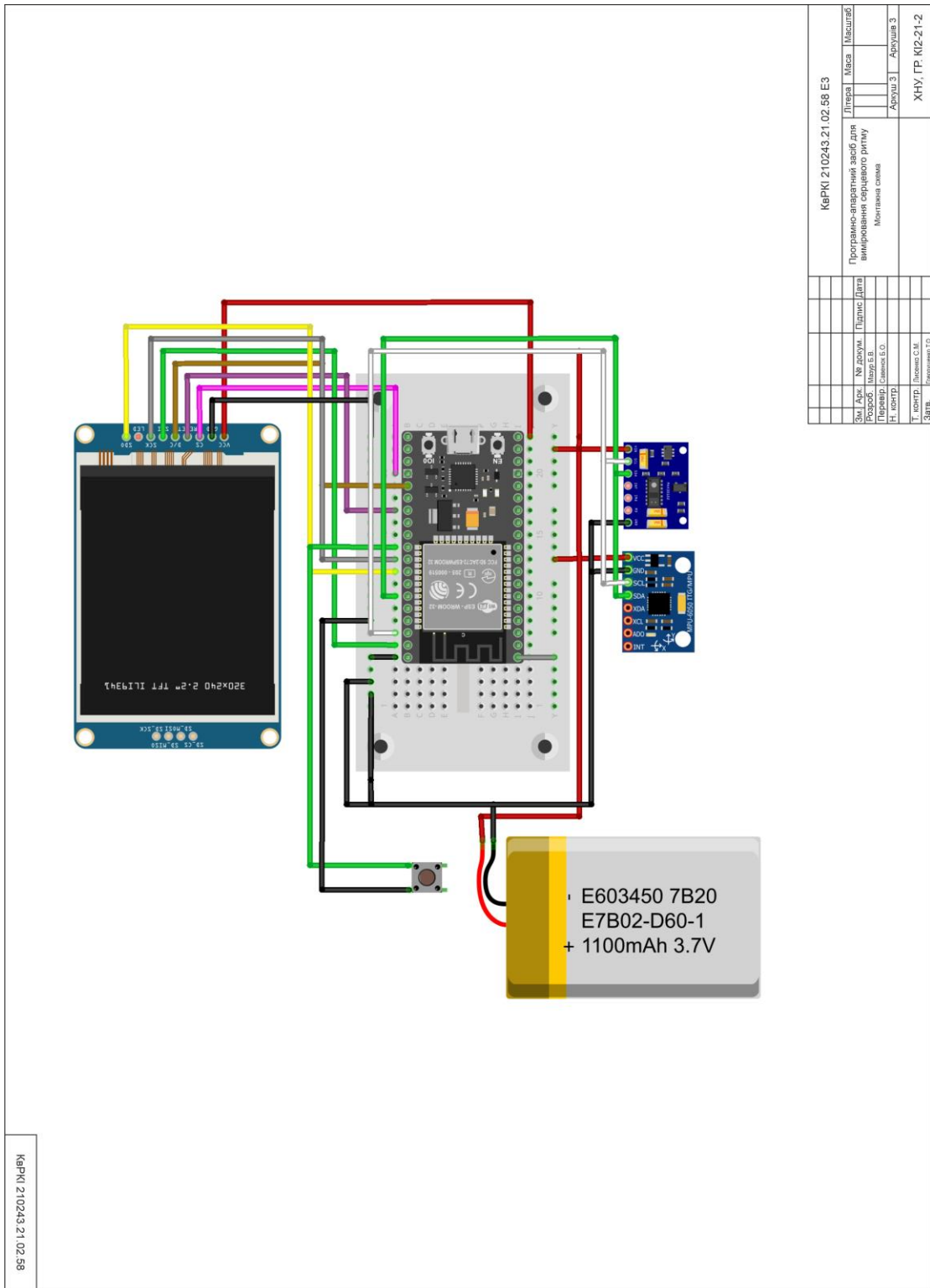


КвРКІ 210243.21.02.58

КвРКІ 210243.21.02.58 Е2			
Літера	Мапа	Масштаб	
Програмно-апаратний засіб для вимірювання серцевого ритму			
Схема з'єднання компонентів ПЗ			
Аркуш 2			Аркуш 3
Т. констр. Штук.			Ліценз. С.М. Підприємство ІО
ХНУ, ГР КІ2-21-2			

Додаток В (обов'язковий)

Копія креслення «Монтажна схема»



Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Богдан МАЗУР

Співавтор:

Назва: Мазур_Програмно-апаратний засіб для вимірювання серцевого ритму та аналізу отриманих даних на основі ESP32 з підтримкою бездротового зв'язку

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 7.4%

Коефіцієнт подібності 2: 1.7%

Мікропробіли: 38

Заміна букв: 1

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-05-29 07:54:45.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-05-29

Дата



Доцент Андрій Нічепорук

експерт

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 25.0%

Dictionaries check: en_US, ru_RU, ua_UA. Errors in the documents: 12%

ID: 242399 Title: БКР Програмно-апаратний засіб для вимірювання серцевого ритму та аналізу отриманих даних на основі ESP32 з підтримкою бездротового зв'язку Added in a DB: 2025-05-29 Authors: Богдан МАЗУР Heads: Богдан САВЕНКО Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	77525	571	19729 (25%)	141 (25%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes
240799	Title: Звіт з ПДП Розробка пристрою для вимірювання серцевого ритму та аналізу отриманих даних на основі ESP32 з підтримкою бездротового зв'язку Added in a DB: 2025-05-04 Authors: Мазур Б.В. Heads: Савенко О.С. Consultants: Opponents:	19385 (25.0%)	136 (24.0%)

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Мазур Богдан

Тема: Програмно-апаратний засіб для вимірювання серцевого ритму та аналізу отриманих даних на основі ESP32 з підтримкою бездротового зв'язку

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 60

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка програмно-апаратного засобу для вимірювання серцевого ритму
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи виконано всебічний аналіз предметної області: розглянуто сучасні методи вимірювання серцевого ритму, типові підходи до обробки біосигналів та особливості використання різних сенсорів. Проаналізовано функціональні можливості мікроконтролера ESP32, зокрема щодо обробки даних у реальному часі та бездротової передачі інформації. У другому розділі обґрунтовано вибір основних апаратних компонентів пристрою. Враховуючи вимоги до точності вимірювань, компактності та енергоефективності, обрано мікроконтролер ESP32, сенсор MAX30102 та дисплей ILI9341. Досліджено актуальні технології з галузі IoT, зокрема протокол MQTT як надійний і легкий спосіб передавання даних. На основі проведеного аналізу виконано постановку задачі та сформульовано технічні вимоги до програмно-апаратного засобу. Розроблено алгоритм роботи програмно-апаратного засобу. У третьому розділі виконано тестування розробленої системи в програмному емуляторі. Побудовано структурну та принципову електричну схему пристрою. Розроблено прошивку для мікроконтролера з використанням сучасних бібліотек мовою C++, яка забезпечує зчитування даних із сенсора, їх попередню фільтрацію,

виявлення піків серцевого ритму, підрахунок пульсу та передавання результатів через бездротовий інтерфейс на MQTT-брокер. Перевірено коректність роботи прошивки, зчитування та обробки даних, а також функціональність передачі результатів через бездротове з'єднання. Оцінено точність алгоритмів виявлення серцевого ритму, стабільність MQTT-з'єднання та правильність відображення результатів на віртуальному дисплеї. Відсутність фізичної плати компенсовано детальним моделюванням поведінки системи в симульованому середовищі.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатньо глибокий аналіз фільтрації та обробки біосигналів.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.


8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Безрукова Т. П.

з.д.и.к., професор, завідувач кафедри ІІЗ

"02" 06 2025 р.

 (підпис)

Завідувачу кафедри КПС
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Богдана МАЗУРА

ГІБ здобувача вищої освіти

ФІТ, 4 курсу, групи КІ2-21-2

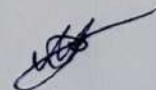
ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

28 05 2025 року



РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Програмно-апаратний засіб для вимірювання серцевого ритму та аналізу отриманих даних на основі ESP32 з підтримкою бездротового зв'язку

Автор Богдан МАЗУР

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський) рівень

Спеціальність 123– Комп'ютерна інженерія

Науковий керівник: д.ф., Богдан Савенко

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	Відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	Не виявлено

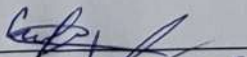
Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) запозичення, знайдені системою Anti-Plagiarism, складають звіт з передипломної практики Мазура Богдана.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 7.45% і адресується до 41 першоджерела; та системою Anti-Plagiarism складає 25%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи



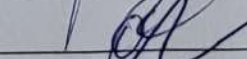
Богдан САВЕНКО

Гарант ОП



Андрій НІЧЕПОРУК

Завідувач кафедри КІС



Ольга ПАВЛОВА