

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

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Кіберфізична система моніторингу стану здоров'я
Назва теми

КвРКІ 2101067.21.01.17 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 III курсу, група КІ2с-21-1


Підпис


М.Р. Якимішин
Ініціали, прізвище

Керівник


Підпис, дата

Д.М.Медзатий
Ініціали, прізвище

Нормоконтролер


Підпис, дата

С.М.Лисенко
Ініціали, прізвище

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


Підпис

Т.О. Говорущенко
Ініціали, прізвище

«24» червня 2024 р.

Хмельницький 2024

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 Т.О. Говорушенко

“ 10 ” 01 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Якимишину Максиму Романовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Кіберфізична система моніторингу стану здоров'я

Керівник проекту (роботи) Медзатий Д.М., к.т.н., доц.

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

Затверджена наказом ректора університету від 01.03.2024 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2024 р.

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

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

Кіберфізична система моніторингу стану здоров'я та постановка задачі щодо її розробки

Проектування кіберфізичної системи моніторингу стану здоров'я

Програмно-апаратна реалізація кіберфізичної системи моніторингу стану здоров'я

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

Архітектура ПЗ проєкту

Архітектура ПЗ для кіберфізичної системи

Апаратне забезпечення проєкту

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., д.т.н., професор		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2024	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2024	виконано
4	Робота над розділом 2 – вибір компонентів для проектування системи адаптивного застосування моніторингових елементів розвідувального БПЛА	01.04.2024	виконано
5	Робота над розділом 3 – проектування системи адаптивного застосування моніторингових елементів розвідувального БПЛА	29.04.2024	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2024	виконано
7	Попередній захист ВКР	30.05.2024	виконано
8	Захист ВКР на засіданні ЕК	Червень 2024 року	

Студент


Підпис

М.Р. Якимішин
Ініціали, прізвище

Керівник роботи


Підпис

Д.М. Медзатий
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 2101067.21.01.17 ПЗ	Пояснювальна записка	55		
			<u>Графічні матеріали</u>			
2		КвРКІ 2101067.21.01.17 Е8	Схема з'єднань мікроконтролеру	1		
3		КвРКІ 2101067.21.01.17 Е8	Загальна блок-схема роботи алгоритму	1		
4		КвРКІ 2101067.21.01.17 Е8	Блок-схема роботи Flask додатку	1		

					КвРКІ 2101067.21.01.17 ВП			
Зм	Арк	№ докум	Підпис	Дата	Кіберфізична система моніторингу стану здоров'я	Літера	Аркуш	Аркушів
Розробив		Яковичин		14.06		У	1	1
Перевір.		Медзатий		14.06		ХНУ, КІ2с-21-1		
Н. контр.		Лисенко		14.06				
Затв.		Говорушечко		14.06				

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Кіберфізична система моніторингу стану здоров'я».

Автор роботи: Якимишин Максим Романович.

Керівник роботи: Медзатий Дмитро Миколайович.

Пояснювальна записка: 55 с., 10 рис., 4 дод., 41 джерело.

Графічна частина: 3 креслення.

Моніторингові сенсори, КІБЕРФІЗИЧНА СИСТЕМА, Архітектура, МОНІТОРИНГ, БАЗА ДАНИХ.

Метою дипломної роботи є визначення умов та особливостей застосування обладнання для моніторингу стану здоров'я, а також оцінка механізмів обробки інформації у кіберфізичній системі адаптивного застосування моніторингових елементів для забезпечення достатньої ефективності виявлення змін у стані здоров'я.

Об'єктом дослідження є функціонування моніторингових елементів у кіберфізичній системі здоров'я.

Предметом дослідження є оцінка режимів застосування моніторингових елементів у кіберфізичній системі здоров'я.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.


Підпис студента

14.06.2024

Дата

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1 КІБЕРФІЗИЧНА СИСТЕМА МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я ТА ПОСТАНОВКА ЗАДАЧІ ЩОДО ЇЇ РОЗРОБКИ	6
1.1 Аналіз структурних і функціональних особливостей кіберфізичної системи	6
1.2 Аналіз програмно-апаратного забезпечення обробки інформації в кіберфізичній системі	121
1.3 Постановка задачі.....	18
Висновки до розділу 1	19
РОЗДІЛ 2 ПРОЕКТУВАННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я	21
2.1 Визначення апаратних і програмних підсистем	21
2.2 Визначення методу взаємодії підсистем.....	29
2.3 Опис основних модулів	333
Висновки до розділу 2	39
РОЗДІЛ 3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я	42
3.1 Опис реалізації основних модулів.....	42
3.2 Опис процесу створення БД.....	49
3.3 Опис схем апаратної частини.....	50
3.4 Інструкція користувача.....	53
Висновки до розділу 3	545
ВИСНОВКИ	557
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	59
ДОДАТОК А Схема з'єднань мікроконтролеру	67
ДОДАТОК Б Загальна блок-схема роботи алгоритму	68
ДОДАТОК В Блок-схема роботи flask додатку	69

КвРКІ.190130.19.01.19 ПЗ

Вм.	Арк.	Нелокум.	Підпис	Дата	Кіберфізична система моніторингу стану здоров'я Пояснювальна записка	Літера	Аркш	Аркшів
Виконав		Якимович М.Р.		14.06		y		2
Перевір.		Медзатія Д.М.		14.06				
Н.контр.		Лисенко С.М.		14.06				
Затвер.		Говорушенко Т.О.		14.06				

ХНУ КІ2с-21-1

					КвРКІ. 2101067.21.01.17 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Сучасний світ характеризується стрімким розвитком технологій, які все більше інтегруються в повсякденне життя людей, сприяючи підвищенню якості та комфорту життя. Однією з ключових сфер, де це особливо актуально, є медицина і, зокрема, моніторинг здоров'я. Потреба у надійних, доступних і зручних методах моніторингу стану здоров'я на дому зростає, оскільки це дозволяє вчасно реагувати на зміни стану здоров'я і, за необхідності, коригувати лікування та поведінку. Відповідно, актуальність розробки системи для моніторингу життєво важливих показників, таких як пульс та рівень кисню в крові, є важливим елементом сучасної медицини. Метою даної дипломної роботи є розробка та впровадження портативної системи моніторингу стану здоров'я на базі мікроконтролера Arduino, яка дозволить користувачам самостійно відстежувати свої показники здоров'я в реальному часі.

Об'єктом дослідження є процеси збору, обробки та передачі фізіологічних даних користувача. Предметом дослідження є апаратно-програмний комплекс системи моніторингу, який включає в себе датчики для вимірювання пульсу та рівня кисню в крові, а також модулі для передачі даних через Bluetooth. Методи дослідження базуються на інженерному підході до розробки системи з використанням програмування мікроконтролерів, проектування електронних схем та розробки користувацького інтерфейсу. Для вимірювання точності датчиків використовуються методи статистичного аналізу, а результати реалізації системи перевіряються через ряд тестів у реальних умовах.

Практичне значення роботи полягає у створенні ефективної та доступної системи моніторингу стану здоров'я, яка може бути використана як в медичних установах, так і в домашніх умовах для регулярного контролю життєво важливих показників. Очікувані результати проекту включають забезпечення користувачів зручним інструментом для самостійного здоров'я моніторингу, що допоможе знизити ризики пов'язані зі здоров'ям, покращити своєчасність діагностики та

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 4
Зм.	Арк.	№ докум.	Підпис	Дата		

оптимізувати процеси лікування. Запропонована система також має потенціал для подальшого розвитку та інтеграції з іншими медичними пристроями та системами здоров'я, що робить її важливим внеском у розвиток домашньої медицини.

					КвРКІ. 2101067.21.01.17 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

1 КІБЕРФІЗИЧНА СИСТЕМА МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я ТА ПОСТАНОВКА ЗАДАЧІ ЩОДО ЇЇ РОЗРОБКИ

1.1 Аналіз структурних і функціональних особливостей кіберфізичної системи

Інтернет речей (IoT) відноситься до пристроїв з датчиками, обчислювальною потужністю, програмним забезпеченням та іншими технологіями, які підключаються та спілкуються з іншими пристроями та системами через Інтернет або інші мережі зв'язку [1-5].

Інтернет речей охоплює електроніку, комунікації та інформатику. "Інтернет речей" вважався неправильним, оскільки пристрій не вимагає підключення до загальнодоступного Інтернету і повинен бути підключений лише до мережі [6] і мати окремі адреси [7-8].

Індустрія розвивалася завдяки поєднанню різноманітних технологій, включаючи всюдисущі обчислення, звичайні датчики, все більш потужні Вбудовані системи та машинне навчання [9]. Застаріла галузь вбудованих систем забезпечує бездротові сенсорні мережі, системи управління, автоматизацію (включаючи автоматизацію будинку та будівництва), Інтернет речей незалежно та спільно [10].

На споживчому неділю технологія Інтернету речей широко використовується в Побутова техніка і побутова техніка (освітлювальні прилади, термостати, системи домашньої безпеки, він підтримує екосистему з 1 або більше поширених продуктів розумного дому, включаючи (наприклад, камери та іншу побутову техніку), а також підтримує елементи керування, які використовують пристрої, пов'язані з цією екосистемою. як смартфони і розумні колонки.

Ви можете це зробити. Інтернет речей також використовується в системах охорони здоров'я [11].

Існує велика стурбованість щодо ризиків, пов'язаних із зростанням технологій та продуктів Інтернету речей, особливо в галузі конфіденційності та

					КвРКІ. 2101067.21.01.17 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

безпеки, і Промисловість та уряд вживають заходів для вирішення цих проблем, включаючи розробку міжнародних та регіональних стандартів. керівні принципи та нормативно-правова база [12].

Оскільки пристрої Інтернету речей взаємопов'язані, вони вразливі до порушень безпеки та конфіденційності. У той же час методи бездротового зв'язку цих пристроїв створюють нормативну невизначеність і ускладнюють юрисдикційні обмеження передачі даних [13].

Описуючи Інтернет речей як "момент, коли "речі і речі" підключаються до Інтернету, а не до людей", Cisco Systems підрахувала, що IoT "народився" в період з 2008 по 2009 рік., і що співвідношення речей і людей збільшилося. З 2003 року - 0,08, з 2010 року-1,84 [30].

Широке застосування пристроїв Інтернету речей [31] зазвичай поділяється на споживчу, комерційну, промислову та інфраструктурну сфери [32-33]:

З точки зору споживача можна помітити, що кількість пристроїв Інтернету речей зростають, включаючи підключені гаджети, домашню автоматизацію, носимі технології, підключену охорону здоров'я та пристрої з функціями віддаленого моніторингу [34].

Розглянемо пристрої Інтернету речей з точки зору домашньої автоматизації. Пристрої IoT є частиною більш широкої концепції домашньої автоматизації, включаючи системи освітлення, опалення, кондиціонування повітря, мультимедіа та безпеки, а також системи камер [35-36]. довгострокові вигоди можуть включати економію енергії за рахунок автоматичного відключення освітлення та електроніки або повідомлення мешканців будинку про використання [37].

Розумний будинок або автоматизований будинок можуть базуватися на платформі або концентраторі, який керує розумними пристроями та технологіями [38]. Наприклад, виробники, які використовують Homekit від Apple, можуть використовувати програми на своїх пристроях iOS, таких як iPhone та Apple Watch, для управління своїми продуктами та аксесуарами вдома.

Це може бути спеціальний додаток або ваш власний додаток для iOS, наприклад Siri [41]. Це можна проілюструвати на прикладі Lenovo Smart Home Essentials, набору розумних домашніх пристроїв, якими можна керувати за допомогою програми Apple Home або Siri без використання мосту Wi-Fi [41].

Також, Існує також виділений центр розумного будинку, який пропонується як автономна платформа для підключення різних продуктів розумного будинку. До них відносяться Amazon Echo, Google Home, Homepod від Apple і Smart Деки від Samsung hub.In на додаток до комерційних систем існує безліч невиключних екосистем з відкритим вихідним кодом, включаючи Home Assistant, Openhouse і Domoticz.

Наступна сфера використання є догляд за людьми похилого віку. Одним з важливих застосувань розумного будинку є допомога людям похилого віку та людям з обмеженими можливостями.

Ці домашні системи використовують допоміжні технології для адаптації до конкретних недоліків власника. Голосове управління може допомогти користувачам з візуальними обмеженнями та обмеженнями рухливості, а система оповіщення може бути безпосередньо підключена до кохлеарних імплантатів, які носять глухі люди .

Він може бути оснащений додатковими функціями безпеки, включаючи датчики, які відстежують невідкладну медичну допомогу, таку як падіння та судоми. Технологія розумного дому, що використовується таким чином, може надати користувачам більше свободи та вищу якість життя .

Термін «Enterprise IoT» стосується пристроїв, які використовуються в бізнесі та корпоративних налаштуваннях. Очікується, що до 2019 року EIoT налічуватиме 9,1 мільярда пристроїв[32].

Розглянемо сферу медицина та охорони здоров'я. Медичний Інтернет речей (IoMT) - це програма Інтернету речей для медичних та медичних цілей, яка збирає та аналізує дані для досліджень та моніторингу. IoMT згадується як технологія

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

створення оцифрованої системи охорони здоров'я, яка поєднує наявні медичні ресурси з медичними послугами.

Пристрої Інтернету речей можна використовувати для забезпечення віддаленого моніторингу працездатності та систем аварійної сигналізації. Ці пристрої для моніторингу здоров'я варіюються від моніторів артеріального тиску та серцевого ритму до сучасних пристроїв, які можуть контролювати спеціальні імплантати, такі як кардіостимулятори, електронні браслети Fitbit та сучасні слухові апарати.

Деякі лікарні почали впроваджувати "розумні ліжка", які можуть визначати, коли вони зайняті або коли пацієнт збирається прокинутися. Його також можна регулювати самостійно, щоб забезпечити належний тиск і підтримку пацієнта без ручної взаємодії медсестер .

Згідно зі звітом Goldman Sachs за 2015 рік, медичні пристрої IoT можуть "щорічно економити понад 3000 мільярдів фунтів стерлінгів на витратах на охорону здоров'я, збільшуючи доходи та зменшуючи витрати. Крім того, використання мобільних пристроїв для підтримки медичного моніторингу призвело до створення m-health за допомогою аналізованої статистики охорони здоров'я."

Житлові райони також можуть бути оснащені спеціальними датчиками для моніторингу здоров'я та загального самопочуття людей похилого віку, забезпечення належного лікування та допомоги людям у відновленні рухливості, втраченої в результаті лікування.

Ці датчики утворюють мережу розумних датчиків, які можуть збирати, обробляти та передавати цінну інформацію в різних середовищах, таких як підключення пристроїв домашнього спостереження до лікарняних систем .

Інтернет речей також дозволяє використовувати інші споживчі пристрої, які підтримують здоровий спосіб життя, такі як підключені ваги та портативні Монітори серця. Платформи Інтернету речей також доступні для комплексного

					КВРКІ. 2101067.21.01.17 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

моніторингу стану здоров'я пренатальних та хронічних пацієнтів, допомагаючи керувати основними показниками здоров'я та нормальними потребами в ліках.

Вдосконалений метод виготовлення електроніки з пластику або тканини дозволив створити наддешеві датчики ІоМТ, які можна використовувати і утилізувати.

Ці датчики разом з базовою RFID-електронікою можуть бути виготовлені з паперу або електронного текстилю для бездротових одноразових сенсорних пристроїв. Медичні діагностичні програми були розроблені на місцях, і важлива портативність та низька складність системи.

Станом на 2018 рік ІоМТ використовується не тільки в галузі клінічних лабораторій, але також у секторах охорони здоров'я та медичного страхування. У сфері охорони здоров'я ІоМТ тепер надає послуги лікарям, пацієнтам та іншим особам, таким як доглядачі пацієнтів, медсестри та члени сім'ї. Станьте частиною системи, де записи пацієнтів зберігаються в базі даних, а лікарі та решта медичного персоналу мають доступ до інформації про пацієнтів.

ІоМТ у страховій галузі забезпечує доступ до найкращих та нових видів динамічної інформації. Сюди входять сенсорні рішення, такі як біосенсори, портативні пристрої, підключені медичні пристрої та мобільні додатки для моніторингу поведінки клієнтів. Це може призвести до більш точного андеррайтингу та нової моделі ціноутворення.

Використання Інтернету речей в охороні здоров'я відіграє фундаментальну роль у лікуванні хронічних захворювань, а також у профілактиці та лікуванні захворювань.

Віддалений моніторинг можливий завдяки підключенню потужного бездротового рішення. Цей зв'язок дозволяє медичним працівникам збирати дані про пацієнтів і застосовувати передові алгоритми для аналізу даних про стан здоров'я.

Одноплатний мікроконтролер - це мікроконтролер, вбудований в одну друковану плату. Ця плата містить усі схеми, необхідні для корисної роботи

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

управління: мікропроцесор, схему вводу / виводу, генератор тактової частоти, оперативну пам'ять, пам'ять та пам'ять для збережених програм. Мета полягає в тому, щоб карта відразу ж стала в нагоді розробникам додатків, не витрачаючи час і зусилля на розробку апаратного забезпечення контролера.

Одноплатний мікроконтролер відрізняється від одноплатного комп'ютера тим, що йому не вистачає інтерфейсу користувача загального призначення та інтерфейсів накопичувача, які були б у комп'ютері більш загального призначення. Порівняно з мікропроцесорною платою розробки, плата мікроконтролера наголошує на взаємозв'язках цифрового та аналогового керування деякою керованою системою, тоді як плата розробки може мати лише кілька дискретних або аналогових пристроїв введення/виведення або взагалі не мати їх.

Плата розробки існує для демонстрації або навчання на певному сімействі процесорів, тому внутрішня реалізація важливіша за зовнішню функцію.

Мікроконтролерні системи забезпечують різні форми вхідних і вихідних сигналів, щоб дозволити прикладному програмному забезпеченню керувати зовнішньою системою «реального світу».

Дискретний цифровий вхід/вивід забезпечує один біт даних (увімкнено або вимкнено). Аналогові сигнали, що представляють безперервний діапазон змінних, наприклад температуру або тиск, також можуть бути входами та виходами для мікроконтролерів.

Дискретні цифрові входи та виходи можуть бути буферизовані від шини даних мікропроцесора лише за допомогою адресної засувки або можуть керуватися спеціалізованою мікросхемою вводу/виводу, такою як адаптер паралельного вводу/виводу Intel 8255 або Motorola 6821.

Пізніші однокристальні мікроконтролери мають вхідні та вихідні контакти. Ці схеми вводу/виводу зазвичай не забезпечують достатнього струму для безпосереднього керування такими пристроями, як лампи чи двигуни, тому твердотільні реле керуються цифровими виходами мікроконтролера, а входи ізольовані схемами зміни рівня кондиціонування сигналу та захисними схемами.

Інтерфейси зв'язку відрізняються залежно від віку системи мікроконтролера. Ранні системи могли використовувати послідовний порт для забезпечення RS-232 або струмової петлі.

Послідовний порт може використовуватися прикладною програмою або може використовуватися в поєднанні з ПЗУ монітора для передачі програм у пам'ять мікроконтролера. Сучасні мікроконтролери можуть підтримувати USB, бездротові мережі (Wi-Fi, Zigbee тощо) або забезпечувати з'єднання Ethernet. Крім того, вони можуть підтримувати стек протоколів TCP/IP.

Деякі пристрої мають мікропрограму, доступну для реалізації веб-сервера, що дозволяє розробнику додатків швидко створювати веб-інструмент або систему.

Багато ранніх систем не мали внутрішніх засобів для програмування і покладалися на окрему «господарську» систему для цього завдання. Це програмування, як правило, виконувалося на мові асемблера або іноді на C або PL/M, а потім крос-асемблювався або крос-компілювався на хості.

Деякі одноплатні мікроконтролери підтримують систему мови BASIC, що дозволяє розробляти програми на цільовому обладнанні.

Хостингова розробка дозволяє використовувати всі накопичувачі та периферійні пристрої настільного комп'ютера, забезпечуючи більш потужне середовище розробки.

1.2 Аналіз програмно-апаратного забезпечення обробки інформації в кіберфізичній системі

Arduino - це проект, що надає прості та доступні засоби для створення електронних пристроїв, які можуть взаємодіяти з оточуючим середовищем за допомогою датчиків та приводів. Основні характеристики плат Arduino включають:

– різноманітність плат:

Arduino пропонує різноманітні моделі плат з різними мікроконтролерами та наборами цифрових та аналогових входів/виходів (I/O).

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

Ці плати можна придбати на офіційному веб-сайті компанії або через авторизованих дистриб'юторів;

– інтерфейси зв'язку:

плати Arduino мають різні інтерфейси послідовного зв'язку, включаючи універсальну послідовну шину (USB) на деяких моделях. Ці інтерфейси використовуються для завантаження програм;

– мови програмування:

мікроконтролери Arduino можна програмувати мовами C і C++ з використанням стандартного API, відомого як мова програмування Arduino.

Для програмування використовується інтегроване середовище розробки (IDE) або інструмент командного рядка;

– відкрите використання:

обладнання та програмне забезпечення Arduino є вільно доступними за ліцензіями з обмеженням авторського права, але назва "Arduino" є ексклюзивною для офіційного продукту.

Проект Arduino є важливим інструментом для електронних та програмних розробок, дозволяючи як початківцям, так і фахівцям створювати інноваційні пристрої та розширювати їх можливості.

Плати Arduino надають більшість контактів вводу/виводу мікроконтролера для використання іншими схемами.

Деякі плати, такі як Diecimila, Duemilanove і поточний Uno, забезпечують 14 цифрових контактів вводу/виводу, шість з яких можуть створювати сигнали з широтно-імпульсною модуляцією, і шість аналогових входів, які також можна використовувати як шість цифрових I/O шпильок. Ці штифти розташовані у верхній частині плати через роз'єми розміром 0,1 дюйма (2,54 мм).

Є також багато інших Arduino-сумісних плат, які можуть мати різну кількість контактів та функцій.

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

Проект Arduino є важливим інструментом для електронних та програмних розробок, дозволяючи як початківцям, так і фахівцям створювати інноваційні пристрої та розширювати їх можливості.

Arduino IDE підтримує мови C і C++ за допомогою спеціальних правил структурування коду. Вона постачається з бібліотекою програмного забезпечення з проекту Wiring, яка містить багато загальних процедур введення та виведення.

Написаний користувачем код вимагає лише двох основних функцій: для запуску ескізу та основного циклу програми. Ці функції компілюються та зв'язуються за допомогою заглушки програми main() у виконувану циклічну виконавчу програму з інструментальним набором GNU, який також включений у дистрибутив IDE.

Існує порт Xinu OS для ATmega328P (Arduino Uno та інші з таким самим чіпом), який включає більшість основних функцій. Вихідний код цієї версії є у вільному доступі.

Arduino Uno - це мікроконтролерна плата з відкритим вихідним кодом, яка базується на мікроконтролері Microchip ATmega328P (MCU). Вона була розроблена компанією Arduino.cc і була вперше представлена в 2010 році. Ця плата оснащена наборами контактів цифрового та аналогового вводу/виводу (I/O), які можна підключати до різних плат розширення та інших схем.

Основні характеристики плати Arduino Uno включають:

- 14 цифрових контактів вводу/виводу, шість з яких можуть генерувати сигнали з широтно-імпульсною модуляцією (ШІМ);
- 6 аналогових контактів вводу/виводу;
- програмування здійснюється за допомогою Arduino IDE (інтегрованого середовища розробки) через кабель USB типу B;
- живиться за допомогою USB-кабелю або роз'єму, який приймає напругу від 7 до 20 вольт;
- використовує мікроконтролер ATmega328P, такий же, як на платі Arduino Nano, та має ті самі роз'єми, що й плата Leonardo.

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

Arduino Uno (рисунок 1.1) відрізняється від попередніх плат тим, що використовує мікроконтролер Atmega16U2 (Atmega8U2 до версії R2), запрограмований як USB-последовний перетворювач, замість послідовного чіпа FTDI USB-to-UART.

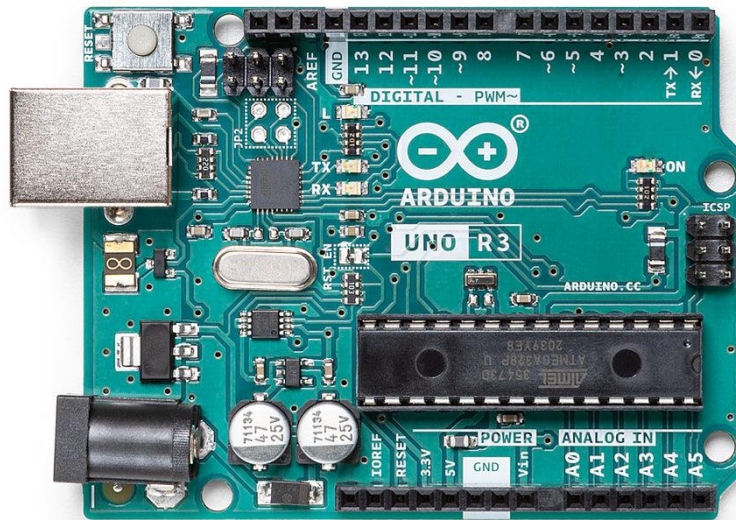


Рисунок 1.1 — Зовнішній вигляд Arduino UNO

Загальні функції плати Arduino Uno:

- є вбудований світлодіод, що керується цифровим виводом 13. Коли на виводі високе значення, світлодіод світиться, коли на виводі низьке значення, він вимкнений;
- VIN (Вхідна напруга) це контакт на платі Arduino/Genuino, який призначений для введення зовнішнього джерела живлення, коли використовується не USB-з'єднання. Напругу можна подавати через цей контакт або отримувати доступ до неї, якщо напруга подається через гніздо живлення. Діапазон напруги, який можна подати через цей контакт, зазвичай від 7 до 20 вольт;
- 5V (5 Вольт) цей вивід надає регульовану напругу 5 вольт від регулятора на платі. Живлення плати може здійснюватися від різних джерел, таких як роз'єм постійного струму (7-20 В), роз'єм USB (5 В) або контакт VIN плати (7-20 В);

- 3V3 (3,3 Вольта) це джерело живлення 3,3 вольта, що створюється вбудованим регулятором на платі. Максимальний споживаний струм цим джерелом становить 50 мА;
- GND це контакти заземлення, які використовуються для підключення до землі;
- IOREF контакт на платі Arduino/Genuino забезпечує опорну напругу для мікроконтролера. Він дозволяє правильно налаштувати екрани, які можуть зчитувати напругу на контакті IOREF і вибирати відповідне джерело живлення або вмикають транслятори напруги на виходах для роботи з 5 В або 3,3 В;
- контакт скидання зазвичай використовується для додавання кнопки скидання до щитів, які блокують кнопку на платі Arduino/Genuino.

Кожна з 14 цифрових контактів і 6 аналогових контактів на платі Arduino Uno може використовуватися як вхід або вихід під керуванням програмного забезпечення, і це можливо завдяки вбудованим функціям, таким як `pinMode()`, `digitalWrite()` і `digitalRead()`. Ось деякі з особливих функцій окремих контактів:

- послідовний / UART (Universal Asynchronous Receiver-Transmitter) контакти 0 (RX) і 1 (TX) призначені для отримання (RX) і передачі (TX) TTL послідовних даних. Вони зазвичай використовуються для комунікації з іншими пристроями або мікроконтролерами через послідовний порт;
- Зовнішні переривання контакти 2 і 3 можуть бути налаштовані на ініціювання переривання при певних подіях, таких як низький рівень сигналу, наростаючий або спадаючий фронт, або зміна значення. Це дозволяє мікроконтролеру реагувати на зовнішні події в реальному часі;
- ШІМ (широко-імпульсна модуляція) контакти 3, 5, 6, 9, 10 і 11 підтримують генерацію сигналів ШІМ, що дозволяє керувати яскравістю світлодіодів, швидкістю моторів та іншими аналоговими пристроями;
- SPI (послідовний периферійний інтерфейс) контакти 10 (SS), 11 (MOSI), 12 (MISO) і 13 (SCK) призначені для зв'язку за допомогою протоколу SPI з іншими пристроями, такими як сенсори, дисплеї або інші мікроконтролери;

- TWI (двопровідний інтерфейс) / I²C контакти SDA (A4) і SCL (A5) призначені для зв'язку за допомогою протоколу I²C, що дозволяє підключати різні пристрої до мікроконтролера, такі як сенсори температури, акселерометри тощо;
- AREF (аналогове опорне значення) контакт AREF використовується як опорна напруга для аналогових входів, і його можна налаштувати для зміни верхньої межі діапазону напруги для аналогових вимірювань.

Arduino/Genuino Uno має різні засоби для спілкування з комп'ютером та іншими пристроями. Ось деякі з них:

- послідовний зв'язок UART ATmega328 забезпечує TTL послідовний зв'язок, доступний через цифрові контакти 0 (RX) і 1 (TX). ATmega16U2 відправляє цей зв'язок через USB і відображається як віртуальний COM-порт на комп'ютері;
- програмне скидання (автоматичне скидання) Arduino Uno розроблено так, що його можна скинути програмно з комп'ютера, що підключений до плати. Це можливо завдяки апаратній конфігурації, яка забезпечує скидання мікроконтролера через програмні команди.
- світлодіоди RX і TX на платі Arduino вмикаються/вимикаються під час передачі даних через USB-послідовний інтерфейс між платою та комп'ютером. Вони не світяться, коли використовуються контакти 0 і 1 для послідовного зв'язку;
- якщо використання основних послідовних портів недоступне, бібліотека SoftwareSerial дозволяє створювати додаткові послідовні порти на будь-яких інших цифрових контактах Uno.

Ці засоби комунікації дозволяють Arduino Uno взаємодіяти з іншими пристроями та програмним забезпеченням, що робить його дуже універсальним і зручним для багатьох застосувань.

Python - це мова програмування загального призначення високого рівня, яка працює на принципах, спрямованих на забезпечення читабельності коду через використання значних відступів.

Її мовна структура та об'єктно-орієнтований підхід призначені для допомоги розробникам у написанні послідовних та логічних кодів як для невеликих, так і для великих проектів.

Python володіє динамічним збором сміття та підтримує кілька парадигм програмування, включаючи структурне (включаючи процедурне), об'єктно-орієнтоване та функціональне програмування.

Велику користь приносить розширена стандартна бібліотека Python, яку часто називають "мовою з вбудованими батареями".

Python є багатопарадигмовою мовою програмування, повністю підтримує об'єктно-орієнтоване і структурне програмування, а також надає можливості функціонального і аспектно-орієнтованого програмування, включаючи метапрограмування та метаоб'єктний підхід.

Python також включає динамічний збір сміття і підтримує динамічне розділення імен для зв'язування методів і змінних під час виконання програми.

Зокрема, Python розроблений з метою підтримки функціонального програмування, запозичуючи функції, такі як фільтрація, відображення, скорочення і ітератори.

Стандартна бібліотека містить модулі `itertools` і `functools`, які реалізують функціональні інструменти, вдосконалені за запозиченням з мов Haskell і Standard ML.

Python має розширюваність завдяки своєму модульному підходу, хоча не всі його функції вбудовані в ядро мови. Ця модульність дозволяє легко додавати програмовані інтерфейси до існуючих програм, роблячи його популярним серед розробників для розширення функціональності.

Мова Python також використовується для розробки експлойтів та інформаційної безпеки.

Багато програм для навчання, такі як Sugar Laptop XO і проект Raspberry Pi, використовують Python для програмування користувачів.

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 18
Зм.	Арк.	№ докум.	Підпис	Дата		

Python є складовою багатьох операційних систем, включаючи Linux, AmigaOS 4, FreeBSD, NetBSD, OpenBSD та macOS. Він широко використовується в інсталяторах для дистрибутивів Linux, таких як Ubuntu та Red Hat Linux, і в системі керування пакетами Portage для Gentoo Linux.

З урахуванням широкого спектру застосувань та великих можливостей мови Python, яку було описано вище, можна визначити, що ця мова програмування відзначається універсальністю. Саме через це вона обрана для виконання цієї роботи.

З урахуванням широкого спектру застосувань та великих можливостей мови Python, яку було описано вище, можна визначити, що ця мова програмування відзначається універсальністю. Саме через це вона обрана для виконання цієї роботи.

PyCharm представляє собою інтегроване середовище розробки (IDE), спеціально призначене для програмування мовою Python. Це програмне забезпечення було розроблене чеською компанією JetBrains і володіє широким функціоналом, включаючи аналіз коду, графічний налагоджувач, вбудований модуль тестування та можливість інтеграції з системами контролю версій (VCS). PyCharm також підтримує використання фреймворку Django для розробки веб-додатків і платформи Anaconda для аналізу даних.

Це інтегроване середовище розробки доступне для користувачів Windows, macOS та Linux, завдяки чому вона є кросплатформовою. Крім безкоштовної версії PyCharm Community Edition з відкритим вихідним кодом, існує також професійна версія, яка надає додатковий функціонал і розповсюджується за власною ліцензією.

1.3 Постановка задачі

Тема: розробка кіберфізичної системи для моніторингу пульсу та рівня кисню в крові з використанням Arduino

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

Загальний опис: цей проект передбачає створення кіберфізичної системи, що дозволяє моніторити життєво важливі показники здоров'я, такі як пульс та рівень кисню в крові, в реальному часі.

Система використовуватиме Arduino UNO як основу для збору даних від пульсоксиметричного датчика, обробки цих даних та їх відправки на сервер, де користувачі можуть переглядати результати через веб-інтерфейс.

Основні завдання:

- проектування та виготовлення апаратної частини системи, що включає Arduino UNO та датчики для вимірювання пульсу та рівня кисню в крові;
- розробка програмного забезпечення для зчитування, обробки та збереження даних з датчиків;
- створення API для передачі даних з Arduino до бази даних;
- розробка веб-інтерфейсу для візуалізації даних у вигляді таблиці, з можливістю відображення рядків з аномальними показниками в червоному кольорі;
- тестування системи та аналіз її точності та надійності.

Технічні специфікації:

- Arduino UNO: використовується як мікроконтролер для збору та передачі даних;
- датчик пульсу та кисню (пульсоксиметр): здійснює моніторинг пульсу та насичення крові киснем;
- веб-сервер на Flask: для обробки запитів до API та відображення даних користувачам;
- SQLite: як система управління базами даних для зберігання історичних даних моніторингу.

Очікувані результати: розроблена система надасть можливість для реального моніторингу пульсу та рівня кисню в крові, що може використовуватися в домашніх умовах для осіб з потребою у постійному нагляді за цими показниками. Система має виявити підвищену точність та надійність у порівнянні з комерційно

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

доступними пульсоксиметрами та забезпечити користувачам доступ до історичних даних через зручний веб-інтерфейс.

1.4 Висновки до розділу 1

Перший розділ дипломної роботи охоплює всебічний аналіз структурних і функціональних особливостей кіберфізичної системи моніторингу здоров'я.

Цей розділ включає вивчення програмно-апаратного забезпечення, необхідного для реалізації системи, а також визначення конкретних завдань, які потрібно виконати для успішної розробки системи.

У цьому розділі було детально проаналізовано різні компоненти, що складають систему, зокрема мікроконтролер Arduino UNO, різноманітні датчики (датчик пульсу та пульсоксиметр), та модуль Bluetooth для бездротової передачі даних.

Огляд почався з визначення вимог до апаратного забезпечення, включаючи вибір відповідних датчиків, здатних точно вимірювати життєво важливі показники, такі як пульс і рівень кисню в крові.

Було виявлено, що інтеграція апаратних та програмних компонентів є критично важливою для досягнення високої точності та надійності системи моніторингу.

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я

2.1 Визначення апаратних і програмних підсистем

Розробка кіберфізичної системи моніторингу здоров'я з використанням пульсоксиметра на базі Arduino вимагає ретельного підходу до визначення апаратних та програмних компонентів. Цей підрозділ має за мету описати ключові апаратні та програмні підсистеми, які складають основу проекту.

Arduino UNO є центральним елементом кіберфізичної системи моніторингу стану здоров'я, відповідальним за керування всіма процесами збору, обробки та передачі даних. Цей мікроконтролер базується на мікрочипі ATmega328P, який забезпечує високу продуктивність і надійність. Arduino UNO оснащений 14 цифровими входами/виходами, що дозволяють підключати різноманітні сенсори і виконавчі пристрої, а також 6 аналоговими входами для зчитування сигналів з аналогових сенсорів, що можуть вимірювати параметри здоров'я, такі як пульс, температура, рівень кисню в крові тощо.

Крім того, Arduino UNO підтримує можливість підключення до Інтернету через Ethernet або Wi-Fi модулі, що дозволяє передавати зібрані дані на віддалені сервери або хмарні платформи для подальшого аналізу і зберігання. Це розширює можливості системи, дозволяючи здійснювати віддалений моніторинг стану здоров'я пацієнтів в режимі реального часу.

Arduino UNO вибрано для цієї системи завдяки його надійності, легкості використання та широкій підтримці спільнотою розробників і ентузіастів.

Величезна кількість готових бібліотек, прикладів коду і детальна документація дозволяють швидко налаштувати систему і вирішувати різноманітні технічні задачі. Така підтримка значно полегшує розробку і впровадження кіберфізичної системи моніторингу стану здоров'я, забезпечуючи її ефективну і безперебійну роботу.

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

МАХ30100 є інтегрованим датчиком для вимірювання пульсу та рівня кисню в крові, який використовує метод пульсоксиметрії. Цей метод базується на використанні світлодіодів і фотодетектора для аналізу зміни поглинання світла кров'ю під час серцевих скорочень.

МАХ30100 оснащений двома світлодіодами – червоним і інфрачервоним – та високочутливим фотодетектором, що дозволяє точно вимірювати ключові параметри серцевої діяльності та оксигенації крові.

Коли серце скорочується, кровеносні судини наповнюються кров'ю, змінюючи поглинання світла на різних довжинах хвиль. МАХ30100 генерує імпульси світла через світлодіоди, які проходять через тканини пацієнта, а фотодетектор реєструє кількість світла, що було поглинуто або відбите кров'ю. Ці дані обробляються для отримання точних показників частоти серцевих скорочень (ЧСС) та насиченості крові киснем (SpO₂). Зображення плати можна побачити на рисунку нижче(рисунок 2.1).

Вимірювання частоти серцевих скорочень є важливим показником для моніторингу стану серцево-судинної системи. Відстеження ЧСС дозволяє вчасно виявляти порушення ритму серця, стресові стани або фізичну перевтому. Насиченість крові киснем (SpO₂) є критичним показником для оцінки дихальної функції та загального рівня оксигенації організму. Зниження рівня кисню в крові може свідчити про проблеми з дихальною системою або іншими медичними станами, що вимагають негайної уваги.

МАХ30100 є компактним і енергоефективним рішенням для інтеграції в портативні та носимі пристрої для моніторингу здоров'я. Його висока точність і надійність роблять його ідеальним вибором для використання в кіберфізичних системах моніторингу стану здоров'я, де необхідно постійно відстежувати важливі фізіологічні параметри пацієнта в режимі реального часу.

Для забезпечення зв'язку між Arduino та іншими компонентами системи, такими як датчики та сервер, використовуються різноманітні інтерфейсні модулі.

Наприклад, для підключення до мережі Internet може використовуватися Ethernet-щит або Wi-Fi модуль, який підключається через SPI або UART інтерфейс.



Рисунок 2.1 – Зовнішній вигляд пульсоксиметра МАХ30100

Програмне забезпечення для Arduino розробляється на мові C++ із використанням інтегрованого середовища розробки Arduino IDE. Це середовище забезпечує зручний інтерфейс для написання, компіляції та завантаження коду на мікроконтролер. Програмне забезпечення включає кілька ключових компонентів, які забезпечують функціональність системи моніторингу стану здоров'я.

Перш за все, код містить алгоритми для зчитування даних з датчика МАХ30100. Ці алгоритми ініціалізують датчик, налаштовують його режими роботи та періодично зчитують значення частоти серцевих скорочень (ЧСС) та рівня насиченості крові киснем (SpO_2). Зчитані дані обробляються для усунення шумів та артефактів, що можуть виникати через рух або зовнішнє освітлення.

Крім того, програмне забезпечення містить драйвери для керування датчиком МАХ30100. Драйвери забезпечують взаємодію з апаратним забезпеченням на низькому рівні, що дозволяє точно контролювати роботу світлодіодів та фотодетектора, а також зчитувати дані з внутрішніх регістрів датчика.

Окрім керування датчиком, код містить модулі для обробки та аналізу отриманих даних. Ці модулі включають функції для розрахунку середніх значень,

фільтрації даних та виявлення аномалій. Наприклад, вони можуть виявляти нерегулярні серцеві ритми або критичні зміни рівня кисню в крові, що можуть свідчити про серйозні медичні стани.

Програмне забезпечення також включає компоненти для відправлення даних через мережеві інтерфейси, такі як Ethernet або Wi-Fi модулі. Ці компоненти дозволяють передавати оброблені дані на віддалені сервери або хмарні платформи для подальшого аналізу та зберігання. Мережеві модулі забезпечують безпечне та надійне з'єднання, що дозволяє здійснювати віддалений моніторинг стану здоров'я в режимі реального часу.

Загалом, програмне забезпечення для Arduino складається з кількох взаємопов'язаних компонентів, які разом забезпечують повний цикл збору, обробки, аналізу та передачі даних з датчика MAX30100. Це дозволяє створювати ефективні та надійні системи моніторингу стану здоров'я, які можуть працювати автономно або в складі більш складних кіберфізичних систем.

Для забезпечення віддаленого доступу до даних та їх візуалізації використовується веб-сервер на базі Flask. Flask є мінімалістичним веб-фреймворком для Python, що дозволяє легко створювати веб-додатки та API. У цьому проєкті Flask виконує кілька ключових функцій, забезпечуючи ефективну роботу системи моніторингу стану здоров'я.

Основні функції Flask у проєкті:

1. Flask-сервер отримує дані від мікроконтролера Arduino через мережеві інтерфейси. Це може бути HTTP-запити, що відправляються Arduino через Ethernet або Wi-Fi модулі. Дані включають показники частоти серцевих скорочень (ЧСС) та рівня насиченості крові киснем (SpO2), які були зчитані та оброблені мікроконтролером;

2. отримані від Arduino дані зберігаються у базі даних для подальшого аналізу та візуалізації. У цьому проєкті використовується реляційна база даних, така як SQLite або MySQL, завдяки її простоті інтеграції з Flask. Flask ORM (Object-

Relational Mapping) або SQLAlchemy може бути використано для взаємодії з базою даних, забезпечуючи зручне управління даними;

3. Flask надає зручний веб-інтерфейс, через який користувачі можуть переглядати збережені дані. Цей інтерфейс може бути реалізований за допомогою HTML, CSS та JavaScript, що дозволяє створити інтерактивні і зручні для користувача сторінки. Веб-інтерфейс може включати графіки, діаграми та інші візуалізації, які показують зміни ЧСС та SpO2 у часі.

Додаткові можливості Flask у проєкті:

1. Flask може забезпечувати реєстрацію та автентифікацію користувачів, що дозволяє контролювати доступ до даних. Це особливо важливо у контексті медичних даних, де необхідно забезпечити конфіденційність та безпеку інформації;

2. Flask може бути налаштований для відправлення сповіщень у разі виявлення критичних змін у стані здоров'я пацієнта. Наприклад, якщо рівень кисню у крові падає нижче безпечного рівня, система може автоматично відправляти повідомлення лікарю або самому пацієнту;

3. Flask може надавати RESTful API, що дозволяє інтегрувати систему моніторингу з іншими медичними або аналітичними системами. Це розширює можливості проєкту, дозволяючи використовувати зібрані дані в більш широкому контексті.

SQLite використовується для збереження історичних даних моніторингу в системі кіберфізичного моніторингу стану здоров'я. Це легка та надійна система управління базами даних (СУБД), яка не потребує налаштування та обслуговування окремого сервера баз даних, що робить її ідеальним вибором для вбудованих або автономних застосунків.

Основні переваги SQLite:

1. SQLite є однією з найлегших СУБД, займаючи мінімум місця на диску. Весь двигун бази даних упакований в єдину бібліотеку, яка додає лише кілька сотень кілобайт до загального розміру програми. Це робить SQLite ідеальним для

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

застосування в обмежених апаратних середовищах, таких як мікроконтролери та вбудовані системи;

2. SQLite забезпечує високу надійність та стійкість до помилок. Вона підтримує атомарні транзакції, що гарантують цілісність даних навіть у разі збою живлення або інших непередбачених обставин. Це важливо для медичних додатків, де втрата або пошкодження даних є неприйнятними;

3. SQLite не вимагає налаштування або обслуговування сервера баз даних. Всі дані зберігаються в одному файлі, який може бути легко перенесений або скопійований при необхідності. Це спрощує процес розгортання та обслуговування системи, дозволяючи зосередитися на розробці основної функціональності;

4. SQLite демонструє високу продуктивність при роботі з невеликими та середніми обсягами даних, що робить її ідеальною для додатків, які не потребують надмірних ресурсів або складної архітектури бази даних. Це особливо важливо для системи моніторингу стану здоров'я, де швидкість доступу до даних та їх обробка мають критичне значення.

За допомогою SQLite у проєкті реалізуються наступні функції:

1. SQLite використовується для зберігання всіх історичних даних, зібраних системою моніторингу, таких як показники частоти серцевих скорочень (ЧСС) та рівня насиченості крові киснем (SpO₂). Це дозволяє зберігати та аналізувати дані за тривалий період, що є важливим для виявлення довгострокових тенденцій та аномалій у стані здоров'я пацієнта;

2. SQLite легко інтегрується з Flask за допомогою бібліотеки SQLAlchemy або інших ORM (Object-Relational Mapping) інструментів. Це дозволяє швидко та ефективно реалізувати взаємодію між веб-сервером та базою даних, забезпечуючи зручний доступ до збережених даних через веб-інтерфейс;

3. SQLite не потребує окремого сервера баз даних, загальна архітектура системи стає значно простішою та менш вимогливою до апаратних ресурсів. Це знижує витрати на інфраструктуру та спрощує обслуговування системи;

4. Всі дані зберігаються в одному файлі бази даних, що дозволяє легко переносити базу даних між різними пристроями та платформами. Це забезпечує високу гнучкість у використанні системи та можливість її масштабування або перенесення при необхідності.

Інтерфейс користувача для системи моніторингу стану здоров'я розробляється з використанням HTML, CSS та JavaScript. Ці технології забезпечують створення динамічних та інтерактивних веб-сторінок, що надають користувачам зручний та інтуїтивно зрозумілий спосіб візуалізації даних. HTML є основною мовою розмітки, яка використовується для створення структури веб-сторінок, визначаючи основні елементи інтерфейсу, такі як заголовки, параграфи, таблиці, форми та інші компоненти, що відображають дані моніторингу.

CSS використовується для стилізації веб-сторінок, визначення їх зовнішнього вигляду та оформлення. Він дозволяє змінювати кольори, шрифти, розміри елементів, відступи, вирівнювання та інші візуальні аспекти сторінки. Це допомагає створити естетично привабливий та зручний для користувачів інтерфейс, що робить роботу з даними комфортною та інтуїтивно зрозумілою.

JavaScript використовується для додавання динамічності та інтерактивності веб-сторінок. Він виконує важливі функції, такі як обробка подій (наприклад, кліки користувачів на кнопки), динамічне оновлення даних без перезавантаження сторінки (через AJAX-запити) та створення інтерактивних графіків і діаграм для візуалізації даних моніторингу. JavaScript дозволяє створювати більш реалістичні та зручні для користувача веб-додатки.

Для забезпечення адаптивності та зручності інтерфейсу використовується фреймворк Bootstrap. Це популярний фреймворк, який включає готові компоненти та стилі, що дозволяють швидко створювати привабливі та функціональні інтерфейси. Bootstrap забезпечує коректне відображення інтерфейсу на різних пристроях та екранах, дозволяючи легко адаптувати дизайн під настільні комп'ютери та мобільні пристрої. Він включає безліч готових компонентів, таких як навігаційні панелі, кнопки, форми, модальні вікна та таблиці, що значно

спрощують розробку інтерфейсу. Всі ці компоненти легко налаштовуються та інтегруються у проект з мінімальними зусиллями, дозволяючи зосередитися на логіці та функціональності додатку.

Веб-інтерфейс використовує бібліотеки JavaScript, такі як Chart.js або D3.js, для створення інтерактивних графіків та діаграм, що візуалізують дані. Ці візуалізації дозволяють користувачам легко аналізувати зміни у показниках стану здоров'я (ЧСС та SpO2) та виявляти тенденції або аномалії. Інтерфейс також може включати інтерактивні елементи, такі як фільтри, випадаючі списки та кнопки, що дозволяють користувачам налаштовувати відображення даних відповідно до їхніх потреб.

Розробка інтерфейсу користувача з використанням HTML, CSS, JavaScript та Bootstrap забезпечує створення зручного, естетично привабливого та функціонального веб-додатку для моніторингу стану здоров'я. Такий підхід дозволяє користувачам легко взаємодіяти з системою, аналізувати дані та приймати обґрунтовані рішення на основі отриманих показників, що є критично важливим для ефективного управління здоров'ям.

Правильне визначення апаратних та програмних підсистем є критично важливим для успішної реалізації проекту. Вибір Arduino UNO, датчика MAX30100, інтерфейсних модулів, а також розробка відповідного програмного забезпечення та веб-інтерфейсу забезпечує надійність, доступність та легкість використання кінцевої системи моніторингу стану здоров'я.

2.2 Визначення методу взаємодії підсистем

Для ефективної роботи кіберфізичної системи моніторингу здоров'я важливо забезпечити правильну взаємодію між її апаратними та програмними підсистемами. Цей розділ описує методи взаємодії, які використовуються в проекті для інтеграції різних компонентів системи.

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 29
Зм.	Арк.	№ докум.	Підпис	Дата		

Arduino UNO та датчик пульсу та кисню MAX30100 взаємодіють через протокол UART (Universal Asynchronous Receiver-Transmitter). UART забезпечує послідовну асинхронну передачу даних між мікроконтролером і датчиком.

Це дозволяє передавати інформацію по одному біту за раз через одну лінію, що робить цю технологію простою у використанні та ідеальною для інтеграції з різними сенсорами.

Дані від датчика MAX30100, що включають інформацію про пульс та рівень кисню в крові, передаються у вигляді серійних пакетів. Кожен пакет даних містить інформацію, зібрану датчиком протягом певного часового інтервалу, та надсилається до Arduino для подальшої обробки в режимі реального часу.

Для забезпечення підключення до Інтернету Arduino UNO може використовувати Wi-Fi модуль, наприклад ESP8266 або ESP32, який підключається до мікроконтролера через SPI (Serial Peripheral Interface).

SPI є синхронним протоколом передачі даних, що забезпечує швидку та ефективну взаємодію між пристроями.

Використання SPI забезпечує високу швидкість передачі даних, що є важливим для стабільного та надійного з'єднання між Arduino та Wi-Fi модулем (рисунок 2.2).



Рисунок 2.2 – Зовнішній вигляд модуля Wi-Fi ESP32

Коли Wi-Fi модуль отримує дані від Arduino через SPI, він передає їх на віддалений сервер через Інтернет-з'єднання.

Це дозволяє системі моніторингу працювати в реальному часі, забезпечуючи постійний доступ до актуальної інформації про стан здоров'я користувача.

Дані можуть бути збережені на сервері, де вони доступні для подальшого аналізу та візуалізації через веб-інтерфейс.

Такий підхід дозволяє створити надійну систему моніторингу, яка оперативно реагує на зміни в стані здоров'я та забезпечує користувачів і медичних фахівців точними та своєчасними даними.

Інтеграція технічних засобів і чіткі інструкції користувачам дозволяють легко впровадити систему у повсякденне життя, забезпечуючи ефективний моніторинг та аналіз особистого здоров'я.

Використання HTTP REST API дозволяє веб-серверу на Flask приймати та обробляти HTTP запити від Arduino, забезпечуючи ефективну взаємодію між апаратною та програмною частинами системи. REST (Representational State Transfer) є архітектурним стилем, що використовує стандартні HTTP методи, такі як GET, POST, PUT та DELETE для створення API, які можуть легко взаємодіяти з різними клієнтами.

Arduino відправляє дані до веб-сервера у форматі JSON (JavaScript Object Notation), який є легким для читання та запису форматом обміну даними. Використання формату JSON дозволяє передавати структуровану інформацію, таку як показники пульсу та рівня кисню в крові, у вигляді простого тексту. Це спрощує обробку даних на стороні сервера.

Коли Arduino має нові дані для відправки, він використовує HTTP POST запити для передачі цієї інформації до REST API, розгорнутого на Flask сервері. POST запити дозволяють клієнту (в даному випадку, Arduino) відправляти дані до сервера, де вони можуть бути оброблені та збережені. Flask сервер приймає ці запити, декодує JSON дані та передає їх на обробку.

На стороні сервера Flask API обробляє отримані дані. Коли сервер отримує POST запит з даними, він витягує інформацію, перевіряє її на валідність і потім зберігає в базі даних SQLite. SQLite є легкою та надійною системою управління базами даних, яка не вимагає налаштування окремого сервера, що робить її ідеальною для таких застосунків. Дані зберігаються у вигляді таблиць, що дозволяє ефективно управляти та швидко доступитися до історичних показників.

REST API забезпечує універсальний інтерфейс для взаємодії між апаратною та програмною частинами системи, що дозволяє легко інтегрувати додаткові функції або змінювати існуючі компоненти без суттєвих змін у коді. Ця універсальність і гнучкість є одними з головних переваг використання REST API, оскільки вони дозволяють легко адаптувати систему до нових вимог або умов експлуатації.

Завдяки використанню HTTP REST API веб-сервер на Flask здатний приймати та обробляти дані, що надходять від Arduino, забезпечуючи надійну та ефективну взаємодію між різними компонентами системи моніторингу стану здоров'я. Це створює міцну основу для подальшого розвитку та масштабування системи, забезпечуючи точність та своєчасність обробки та зберігання даних.

Веб-інтерфейс, розроблений на Flask, слугує основним точком доступу для користувачів до системи моніторингу стану здоров'я. Цей інтерфейс не тільки відображає дані, але й забезпечує інтерактивний інструмент для їх аналізу. Завдяки використанню сучасних веб-технологій, таких як HTML, CSS та JavaScript, інтерфейс забезпечує зручне та інтуїтивно зрозуміле середовище для користувачів. Вони можуть переглядати показники здоров'я в реальному часі, такі як пульс та рівень кисню в крові, а також аналізувати історичні дані. Це дозволяє користувачам отримувати повну картину свого стану здоров'я та вчасно реагувати на будь-які зміни.

Однією з ключових особливостей веб-інтерфейсу є можливість відображення даних у вигляді графіків та діаграм. Це значно полегшує сприйняття інформації та дозволяє користувачам швидко ідентифікувати тенденції або аномалії у своїх

показниках здоров'я. Інтерактивні елементи, такі як фільтри, пошукові поля та випадючі списки, надають додаткові можливості для аналізу даних, дозволяючи користувачам налаштовувати відображення інформації відповідно до своїх потреб.

Взаємодія між підсистемами кіберфізичної системи моніторингу здоров'я відіграє ключову роль у її ефективності та надійності. Використання стандартизованих комунікаційних протоколів та технологій, таких як UART, SPI, HTTP REST API та SQL, забезпечує високу інтегрованість та сумісність компонентів системи. Протокол UART дозволяє ефективно передавати дані між датчиками та мікроконтролером, забезпечуючи точність та своєчасність збору інформації. Протокол SPI забезпечує швидку та надійну передачу даних між мікроконтролером та мережевими модулями, такими як Wi-Fi.

HTTP REST API дозволяє веб-серверу на Flask приймати дані від Arduino та інших пристроїв, обробляти їх та зберігати у базі даних SQLite. Це створює гнучкий та масштабований інтерфейс для взаємодії між апаратною та програмною частинами системи, забезпечуючи стабільну роботу в різних умовах експлуатації. Використання SQL для управління базою даних забезпечує надійне зберігання та швидкий доступ до історичних даних, що є критично важливим для аналізу та прийняття рішень.

Завдяки такій інтегрованій архітектурі система моніторингу здоров'я може надійно функціонувати у різних умовах, забезпечуючи точність та своєчасність обробки даних. Це дозволяє створити ефективну та зручну у використанні систему, яка відповідає сучасним вимогам та потребам користувачів. та бути доступною для користувачів з різним рівнем технічної підготовки.

2.3 Опис основних модулів

Кіберфізична система моніторингу здоров'я складається з декількох ключових модулів, кожен з яких відіграє важливу роль у загальній архітектурі системи. Нижче наведено детальний опис основних модулів: апаратного модуля,

програмного забезпечення Arduino, серверного програмного забезпечення, веб-інтерфейсу та бази даних.

Апаратний модуль включає в себе мікроконтролер Arduino UNO, датчик пульсу та кисню MAX30100, Wi-Fi модуль для зв'язку, а також додаткове обладнання для інтерфейсу та живлення. Arduino UNO слугує як основний контролер системи, відповідальний за обробку сигналів з датчиків і передачу зібраних даних. Мікроконтролер Arduino UNO базується на мікрочіпі ATmega328P, який має достатню потужність для обробки вхідних сигналів і виконання необхідних обчислень в реальному часі.

Датчик MAX30100 спеціалізується на вимірюванні пульсу та рівня кисню в крові, використовуючи методику черезшкірного відбиття світла. Цей метод передбачає використання світлодіодів, які випромінюють світло через шкіру, і фотодетектора, який вимірює кількість відбитого світла. Залежно від кількості крові, що проходить через судини під шкірою, інтенсивність відбитого світла змінюється. Датчик аналізує ці зміни, щоб обчислити пульс і рівень кисню в крові.

Дані, зібрані датчиком MAX30100, передаються до Arduino UNO, де вони обробляються. Обробка даних включає фільтрацію шумів, обчислення середніх значень та підготовку даних для передачі. Цей процес забезпечує точність і надійність вимірювань, що є критично важливим для моніторингу фізіологічних параметрів пацієнта.

Wi-Fi модуль, підключений до Arduino UNO, забезпечує зв'язок з мережею, дозволяючи передавати оброблені дані на віддалений сервер або інший пристрій для подальшого аналізу та зберігання. Цей модуль може бути підключений через SPI (Serial Peripheral Interface), що забезпечує швидку та ефективну передачу даних. Завдяки Wi-Fi модулю система може працювати бездротово, що значно спрощує її використання та встановлення.

Додаткове обладнання для інтерфейсу включає кнопки, світлодіоди та інші компоненти, які дозволяють користувачам взаємодіяти з системою. Наприклад, світлодіоди можуть використовуватися для індикації стану системи або

сигналізації про певні події, такі як успішна передача даних або виявлення аномальних фізіологічних показників.

Живлення апаратного модуля забезпечується через зовнішній блок живлення або батареї, що дозволяє системі працювати автономно. Стабілізатори напруги та інші компоненти забезпечують належну подачу електроенергії до всіх елементів системи, запобігаючи перевантаженням та забезпечуючи стабільну роботу.

Програмне забезпечення для Arduino розробляється з використанням середовища розробки Arduino IDE та мови програмування C++. Arduino IDE надає інтуїтивно зрозумілий інтерфейс для написання, компіляції та завантаження коду на мікроконтролер Arduino UNO. Мова програмування C++ використовується для створення ефективних та швидких алгоритмів, що керують всіма процесами в системі. Інтерфейс користувача для використання Arduino IDE можна побачити нижче (рисунок 2.3)



Рисунок 2.3 – Зовнішній вигляд середовища Arduino IDE

Основним компонентом програмного забезпечення є драйвери, які забезпечують управління датчиком МАХ30100 та Wi-Fi модулем. Драйвер для датчика МАХ30100 відповідає за ініціалізацію датчика, налаштування його параметрів, таких як інтенсивність світла світлодіодів, та зчитування сирих даних. Драйвер забезпечує стабільне та надійне з'єднання з датчиком, дозволяючи отримувати точні вимірювання пульсу та рівня кисню в крові.

Драйвер для Wi-Fi модуля відповідає за налаштування мережевого з'єднання, підключення до Wi-Fi мережі, та управління передачею даних. Цей драйвер реалізує функції для відправлення даних через інтернет за допомогою протоколу HTTP. Він також може обробляти відповіді від сервера, що дозволяє реалізувати двосторонній зв'язок між мікроконтролером та сервером.

Програмне забезпечення включає алгоритми для обробки даних, зібраних датчиком МАХ30100. Після зчитування сирих даних з датчика, ці алгоритми виконують обробку сигналів для фільтрації шумів та артефактів. Використовуючи методи цифрової обробки сигналів, програмний код визначає ключові параметри, такі як частота серцевих скорочень та рівень насичення крові киснем. Ці параметри обчислюються на основі змін інтенсивності світла, що відбивається від кровоносних судин, і надають важливу інформацію про фізіологічний стан користувача.

Для передачі оброблених даних на сервер, програмне забезпечення використовує HTTP запити. Коли нові дані готові до відправлення, мікроконтролер формує HTTP POST запит, який включає дані у форматі JSON. Цей запит відправляється на сервер, де дані можуть бути збережені та проаналізовані. Передача даних через HTTP забезпечує гнучкість та сумісність, дозволяючи інтегрувати систему з різними веб-сервісами та додатками.

Програмний код також містить функції для моніторингу стану з'єднання та повторної спроби відправки даних у разі виникнення помилок. Це забезпечує надійність системи та мінімізує втрати даних у випадку тимчасових проблем з мережею.

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		

Серверна частина системи базується на веб-фреймворку Flask, який є легким і гнучким інструментом для створення веб-додатків на мові програмування Python. Flask надає можливості для швидкої розробки RESTful API, які забезпечують взаємодію між апаратним модулем та сервером. RESTful API дотримується принципів REST (Representational State Transfer) і використовує стандартні HTTP методи, такі як GET, POST, PUT та DELETE, для обробки запитів.

API, розроблене на Flask, приймає запити від апаратного модуля, тобто від мікроконтролера Arduino UNO, який передає дані через HTTP POST запити. Ці запити містять дані у форматі JSON, що включають показники пульсу та рівня кисню в крові, зібрані датчиком MAX30100. Flask обробляє ці запити, розпаковує JSON дані та перевіряє їх на коректність і повноту. Після перевірки дані передаються на подальшу обробку і зберігання.

Отримані дані зберігаються у базі даних SQLite, яка є легкою і надійною системою управління базами даних, що не вимагає окремого сервера для функціонування. SQLite чудово підходить для невеликих до середніх проектів, де важлива простота налаштування та використання. Сервер створює структуру бази даних з відповідними таблицями для зберігання даних про пульс, рівень кисню та інші необхідні метадані.

Кожен запис у базі даних містить час і дату вимірювання, що дозволяє відстежувати динаміку фізіологічних показників користувачів.

Окрім прийому даних від апаратного модуля, сервер також обробляє запити від веб-інтерфейсу.

Веб-інтерфейс, розроблений з використанням HTML, CSS та JavaScript, дозволяє користувачам взаємодіяти з системою через веб-браузер. Flask відповідає за обробку HTTP GET запитів від веб-інтерфейсу, надаючи доступ до зібраних даних та історичної інформації, що зберігається у базі даних.

Користувачі можуть переглядати свої фізіологічні показники у реальному часі, а також аналізувати історичні дані за допомогою графіків і діаграм.

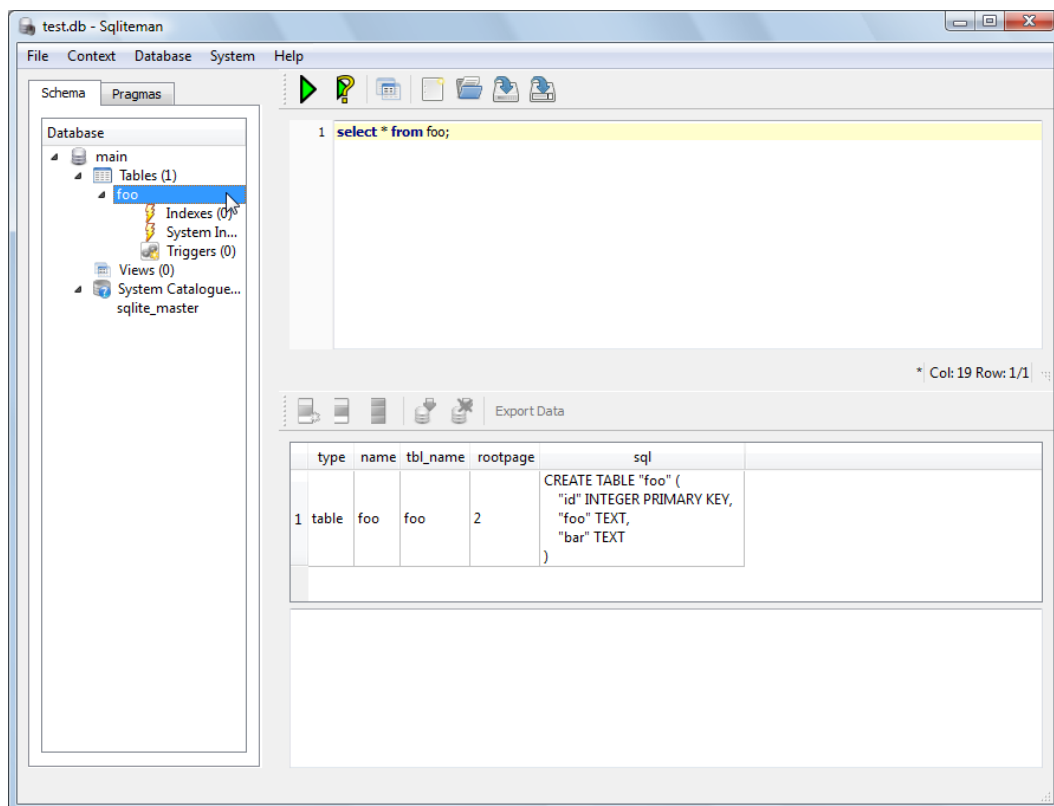
					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

Flask надає функціонал для запиту і відображення цих даних у зручному форматі.

Запити можуть фільтруватися за певними критеріями, такими як діапазон дат або конкретні показники здоров'я, що дозволяє користувачам налаштовувати відображення даних відповідно до своїх потреб.

Таким чином, серверна частина системи, побудована на Flask, виконує критичні функції для прийому, обробки і зберігання даних, а також забезпечує доступ користувачів до інформації через веб-інтерфейс. Це дозволяє створити інтегровану і ефективну систему моніторингу стану здоров'я, яка відповідає сучасним вимогам та потребам користувачів.

SQLite — це компактна вбудована реляційна база даних з відкритим кодом. Вона одна з найпопулярніших у світі, має нагороду Google-O'Reilly Open Source Awards і широко використовується в додатках та системах, де потрібно організувати зберігання даних.



Рисунк 2.3 – Зовнішній вигляд середовища SQLite

База даних SQLite використовується для збереження всіх даних, які генеруються системою моніторингу стану здоров'я. Ця база даних є надійною і легкою системою управління базами даних, що дозволяє зберігати значні обсяги інформації без необхідності в окремому сервері.

SQLite ідеально підходить для невеликих і середніх проєктів завдяки своїй простоті у налаштуванні та використанні, а також високій продуктивності.

У базі даних зберігаються не тільки поточні вимірювання пульсу та рівня кисню, але й історичні дані.

Кожен запис містить часові мітки, що дозволяють відстежувати зміни фізіологічних параметрів у динаміці. Це надає можливість для детального аналізу стану здоров'я користувача за певний період часу. Наприклад, можна визначити тенденції, виявити аномалії або порівнювати показники за різні дні чи тижні.

Використання SQL запитів забезпечує швидкий і зручний доступ до збережених даних.

Запити можуть бути простими, такими як вибірка останніх записів, або складними, включаючи агрегування даних і фільтрацію за різними критеріями. Це дозволяє швидко знаходити необхідну інформацію і проводити ефективний аналіз. SQLite підтримує стандартний SQL синтаксис, що полегшує розробку і інтеграцію з іншими системами.

Описані модулі складають основу кіберфізичної системи моніторингу стану здоров'я. Взаємодія між апаратними і програмними компонентами забезпечує комплексне функціонування системи.

Апаратний модуль, включаючи Arduino UNO, датчик пульсу і кисню MAX30100, та Wi-Fi модуль, відповідає за збір і передачу даних. Програмне забезпечення, розроблене на мові C++ з використанням Arduino IDE, обробляє ці дані і передає їх на сервер через HTTP запити.

На серверній стороні веб-фреймворк Flask реалізує RESTful API для прийому даних, їх обробки і зберігання у базі даних SQLite.

Flask також обслуговує запити від веб-інтерфейсу, надаючи користувачам доступ до зібраної інформації.

Інтерактивний веб-інтерфейс дозволяє користувачам переглядати дані у реальному часі і аналізувати історичні показники за допомогою графіків та діаграм.

Ця інтегрована система забезпечує ефективну обробку даних, надійне зберігання і зручний доступ до інформації. Вона створює високу функціональність та користувацьку цінність, дозволяючи користувачам отримувати детальну інформацію про свій стан здоров'я та своєчасно реагувати на будь-які зміни. Завдяки модульному підходу та використанню стандартизованих технологій система є масштабованою і може бути легко адаптована для різних потреб та умов.

2.4 Висновки до розділу 2

Другий розділ дипломної роботи присвячений детальному проектуванню кіберфізичної системи моніторингу стану здоров'я, яка об'єднує як апаратні, так і програмні підсистеми, методи їх взаємодії та детальний опис основних модулів, що становлять цю систему.

У цьому розділі детально розглядаються апаратні компоненти, такі як Arduino UNO, різноманітні датчики, включаючи датчики пульсу та рівня кисню в крові, а також програмне забезпечення, що забезпечує збір, обробку та аналіз зібраних даних.

Arduino UNO виступає центральним контролером цієї системи, який виконує збір даних з датчиків і керує процесом їх обробки. Він взаємодіє з датчиками через свої аналогові та цифрові входи/виходи, забезпечуючи збір даних про пульс та рівень кисню в крові.

Датчик пульсу відіграє важливу роль у системі, здійснюючи моніторинг серцебиття та передаючи відповідні дані на аналогові входи Arduino для подальшої обробки та аналізу.

Пульсоксиметр, що є ще одним важливим компонентом, вимірює рівень кисню в крові і пульс, зчитуючи свої дані через аналоговий вхід Arduino. Ці дані є

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

надзвичайно важливими для оцінки загального стану здоров'я користувача, зокрема його фізичного стану та потреб у кисні, що особливо критично у випадках моніторингу пацієнтів з респіраторними захворюваннями.

Bluetooth модуль, що є частиною системи, відповідає за бездротовий зв'язок між системою Arduino та зовнішніми пристроями, такими як смартфони або комп'ютери. Це забезпечує можливість передачі зібраних медичних даних для подальшого аналізу або відображення на зовнішніх пристроях, що робить систему більш доступною та корисною для кінцевих користувачів, надаючи їм можливість легко отримувати інформацію про свій стан здоров'я.

Особлива увага в розділі приділяється координації між підсистемами, що є ключовим аспектом у ефективному функціонуванні системи моніторингу стану здоров'я.

Взаємодія між центральним контролером Arduino, датчиками та Bluetooth модулем є критично важливою для забезпечення ефективного збору, передачі та обробки даних. Точність, надійність та інтеграція всіх компонентів є вирішальними аспектами, які дозволяють оптимізувати процес моніторингу та забезпечувати високу якість зібраних даних.

Кожен з компонентів системи відіграє свою унікальну роль, і їх правильна інтеграція є вирішальною для загальної ефективності системи.

Окрім технічних аспектів, у розділі розглядаються методи забезпечення точності та надійності системи, включаючи заходи щодо мінімізації помилок у зборі та передачі даних. Це включає як апаратні рішення, так і алгоритми програмного забезпечення, що використовуються для фільтрації та аналізу даних, зібраних з датчиків.

Таким чином, другий розділ дипломної роботи не лише описує технічні аспекти створення кіберфізичної системи моніторингу стану здоров'я, але й надає детальний аналіз її компонентів, методів їх взаємодії та забезпечення ефективності системи в цілому.

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я

3.1 Опис реалізації основних модулів

Детальний опис коду, який забезпечує функціональність зчитування та обробки даних з датчика пульсу KY-039 та пульсоксиметра MAX30105 для вимірювання частоти серцевих скорочень. Кожен фрагмент коду пояснено з акцентом на його призначення та функціональність.

```
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"
```

У цьому фрагменті коду підключаються необхідні бібліотеки:

- Wire.h: Бібліотека для роботи з протоколом I2C, який використовується для комунікації між мікроконтролером та іншими пристроями;
- MAX30105.h: Бібліотека для роботи з пульсоксиметром MAX30105;
- heartRate.h: Бібліотека для обчислення частоти серцевих скорочень на основі даних з пульсоксиметра.

```
MAX30105 particleSensor;
```

Ініціалізація об'єкта particleSensor для роботи з пульсоксиметром MAX30105. Цей об'єкт забезпечує доступ до функцій і методів, необхідних для налаштування та зчитування даних з пульсоксиметра.

```
const int pulsePin = A0;
```

Оголошення константи pulsePin, яка вказує, що датчик пульсу KY-039 підключено до аналогового піна A0 мікроконтролера. Це дозволяє програмі зчитувати аналогові сигнали з цього піна.

```
long lastBeat = 0;
```

```
float beatAvg = 0;
```

Оголошення змінних для зберігання даних:

- lastBeat: Змінна для зберігання часу останнього серцевого скорочення. Використовується для обчислення інтервалів між ударами серця;
- beatAvg: Змінна для зберігання середньої частоти серцевих скорочень, вираженої у ударах на хвилину.

```
void setup() {
  Serial.begin(9600);
  while (!Serial);
```

Функція setup виконується один раз при запуску програми. Вона починається з ініціалізації послідовного порту на швидкості 9600 біт/с, що використовується для передачі даних на комп'ютер або інший пристрій через Bluetooth або USB. Команда while (!Serial); забезпечує затримку до відкриття серійного порту, що зручно для налаштувань, які можуть вимагати присутності серійного монітора для виведення даних.

```
    if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) {
      Serial.println("MAX30105 не знайдено. Перевірте
підключення.");
      while (1);
    }
```

Цей блок коду ініціалізує пульсоксиметр MAX30105 через інтерфейс I2C. Якщо датчик не знайдено, на послідовний порт виводиться повідомлення про помилку, і програма зупиняється в нескінченному циклі while (1);, що фактично зупиняє подальше виконання програми.

```
    particleSensor.setup();
    particleSensor.setPulseAmplitudeRed(0x0A);
    particleSensor.setPulseAmplitudeGreen(0);
  }
```

Цей блок коду налаштовує пульсоксиметр MAX30105 з параметрами за замовчуванням. Функція particleSensor.setup() виконує початкове налаштування датчика, тоді як setPulseAmplitudeRed(0x0A) встановлює амплітуду червоного

світлодіода для сигналізації роботи, а `setPulseAmplitudeGreen(0)` вимикає зелений світлодіод.

```
void loop() {  
    int pulseValue = analogRead(pulsePin);
```

Функція `loop` виконується безперервно після завершення `setup`. Спочатку зчитуються дані з аналогового піну, до якого підключено датчик пульсу KY-039. Значення, зчитане з цього піну, зберігається у змінній `pulseValue`.

```
    long irValue = particleSensor.getIR();
```

Цей рядок зчитує інфрачервоне (IR) значення з пульсоксиметра MAX30105. Інфрачервоне значення використовується для визначення присутності серцевих скорочень.

```
    if (checkForBeat(irValue) == true) {  
        long delta = millis() - lastBeat;  
        lastBeat = millis();  
        beatAvg = 60 / (delta / 1000.0);  
    }
```

Цей блок коду перевіряє, чи виявлено серцевий ритм за допомогою функції `checkForBeat`, яка аналізує IR дані. Якщо ритм виявлено, обчислюється час між поточним та останнім серцевим скороченням (`delta`). Значення `millis()` повертає кількість мілісекунд, що минули з моменту запуску програми. Цей час використовується для обчислення середньої частоти серцевих скорочень (`beatAvg`) в ударах на хвилину.

```
    Serial.print("Pulse Sensor Value: ");  
    Serial.print(pulseValue);  
    Serial.print(", Heart Rate: ");  
    Serial.print(beatAvg);  
    Serial.print(", IR Value: ");  
    Serial.println(irValue);  
  
    delay(1000);
```

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

}

Останній блок коду відповідає за відправку зчитаних даних на телефон через Bluetooth. Команди `Serial.print` і `Serial.println` використовуються для виведення значень змінних на послідовний порт. Значення `pulseValue`, `beatAvg` і `irValue` виводяться для моніторингу і подальшого аналізу. Використання `delay(1000);` додає затримку в 1 секунду між кожним циклом зчитування і відправки даних.

Таким чином, цей код забезпечує базову функціональність для зчитування та обробки даних з датчиків пульсу KY-039 і пульсоксиметра MAX30105, що дозволяє вимірювати частоту серцевих скорочень та відправляти ці дані на зовнішні пристрої для подальшого аналізу або відображення.

Нижче представлено блок-схеми розроблених алгоритмів(рисунок 3.1, рисунок 3.2).

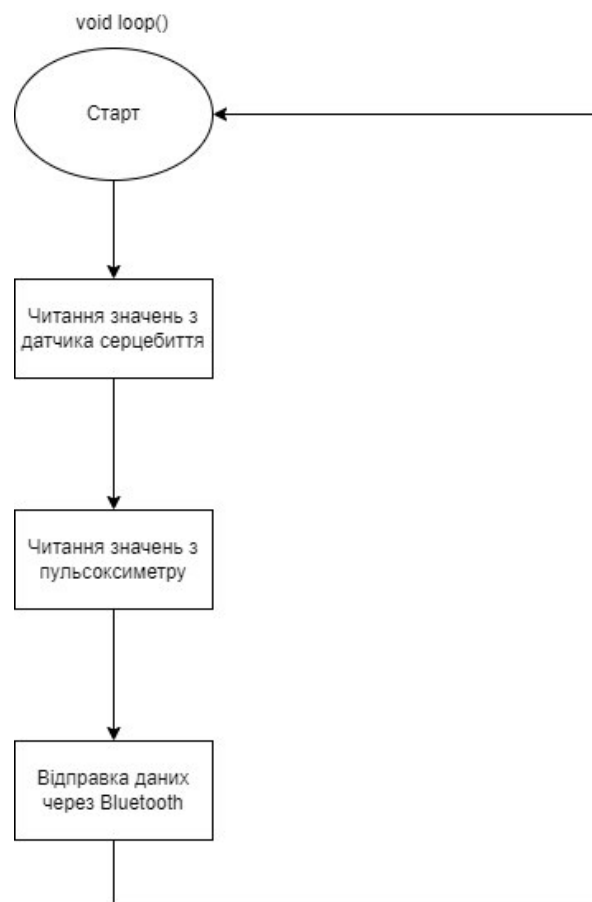


Рисунок 3.1 — Блок-схема основного циклу

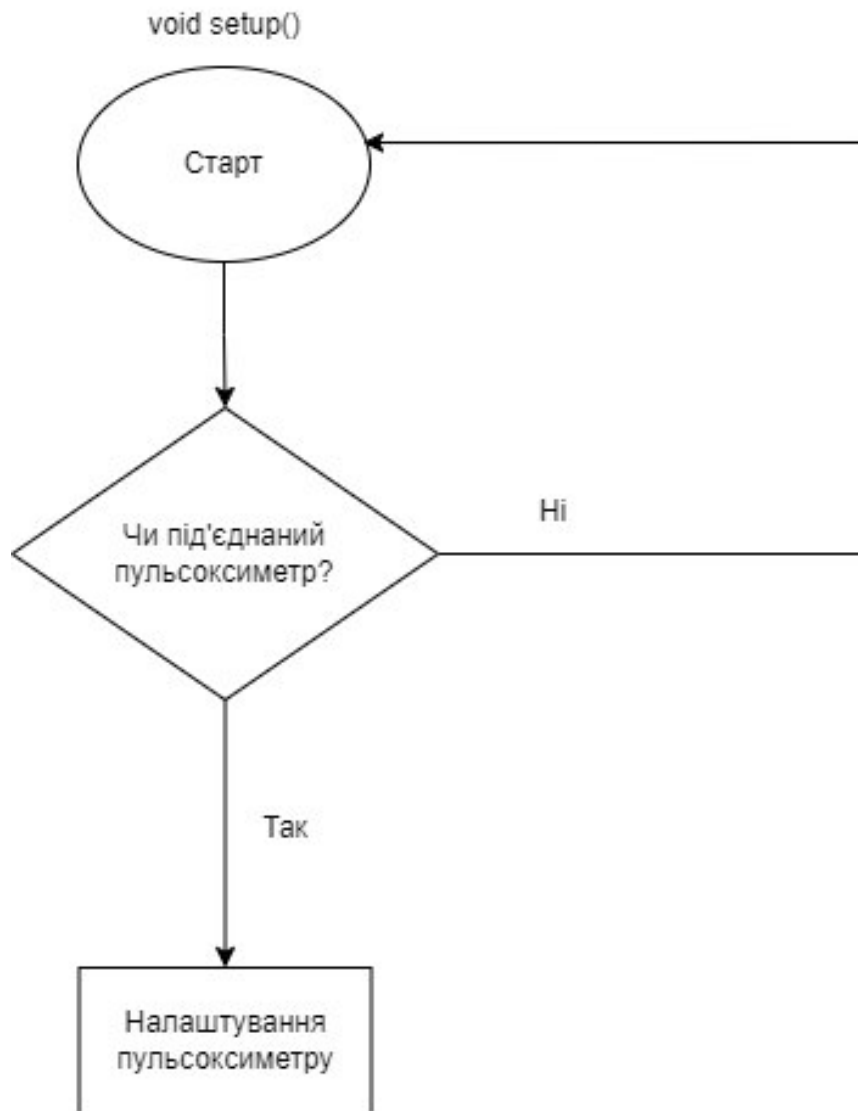


Рисунок 3.2 — Блок-схема ініціалізації

Також було розроблено програмну частину для візуалізації.

```

from flask import Flask, request, render_template
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime
  
```

Імпортуються необхідні бібліотеки: Flask для веб-додатку, SQLAlchemy для роботи з базою даних, і datetime для роботи з датами та часом.

```

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] =
'sqlite:///data.db'
  
```

```
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)
```

Ініціалізація Flask-додатку та налаштування бази даних:

- `app = Flask(__name__)` створює об'єкт додатку Flask.
- `app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///data.db'` вказує шлях до бази даних SQLite.
- `app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False` вимикає функцію відстеження змін SQLAlchemy для зменшення навантаження.
- `db = SQLAlchemy(app)` ініціалізує об'єкт бази даних SQLAlchemy, пов'язаний з додатком Flask.

```
class HealthData(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    pulse = db.Column(db.Integer, nullable=False)
    oxygen = db.Column(db.Integer, nullable=False)
    timestamp = db.Column(db.DateTime,
default=datetime.now)
    def __repr__(self):
        return f'<HealthData {self.pulse}, {self.oxygen},
{self.timestamp}>'
```

Оголошення моделі HealthData, яка представляє таблицю в базі даних:

- `id = db.Column(db.Integer, primary_key=True)` визначає первинний ключ.
- `pulse = db.Column(db.Integer, nullable=False)` зберігає значення пульсу.
- `oxygen = db.Column(db.Integer, nullable=False)` зберігає рівень кисню в крові.
- `timestamp = db.Column(db.DateTime, default=datetime.now)` зберігає час запису даних з автоматичним встановленням поточного часу.
- Метод `__repr__` забезпечує зручне текстове представлення об'єкта для відлагодження.

```
@app.route('/submit', methods=['POST'])
def submit_data():
```

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

```

pulse = request.json['pulse']
oxygen = request.json['oxygen']
data = HealthData(pulse=pulse, oxygen=oxygen)
db.session.add(data)
db.session.commit()
return {'status': 'success'}, 200

```

Маршрут /submit обробляє POST-запити для додавання нових даних:

- pulse = request.json['pulse'] і oxygen = request.json['oxygen'] зчитують дані з тіла запиту.
- data = HealthData(pulse=pulse, oxygen=oxygen) створює новий запис у базі даних.
- db.session.add(data) додає новий запис у сесію бази даних.
- db.session.commit() зберігає зміни у базі даних.
- Повертається JSON-відповідь із статусом успіху.

```
@app.route('/')
```

```
def index():
```

```
    data = HealthData.query.all()
```

```
    return render_template('index.html', data=data)
```

Маршрут / обробляє GET-запити для відображення всіх записів:

- data = HealthData.query.all() зчитує всі записи з таблиці HealthData.
- return render_template('index.html', data=data) повертає HTML-шаблон index.html з переданими даними.

```
if __name__ == '__main__':
```

```
    db.create_all()
```

```
    app.run(debug=True)
```

Цей блок коду запускає додаток:

- db.create_all() створює всі таблиці в базі даних, якщо вони ще не існують.
- app.run(debug=True) запускає веб-сервер Flask у режимі налагодження, що дозволяє бачити помилки та автоматично перезавантажувати сервер при зміні коду.

3.2 Опис процесу створення БД

Цей код визначає модель HealthData, яка представляє структуру таблиці в базі даних для зберігання даних про стан здоров'я, таких як пульс і рівень кисню в крові. Він використовує SQLAlchemy, об'єктно-реляційний мапер (ORM) для Python, який спрощує роботу з базами даних шляхом відображення об'єктів Python на записи в базі даних.

```
class HealthData(db.Model):
```

```
class HealthData(db.Model):
```

Клас HealthData є підкласом db.Model, що робить його моделлю SQLAlchemy. Це означає, що SQLAlchemy буде використовувати цей клас для створення та управління відповідною таблицею в базі даних.

```
id = db.Column(db.Integer, primary_key=True)
```

Поле id є первинним ключем таблиці. Воно має тип Integer (ціле число) і автоматично унікальне для кожного запису завдяки атрибуту primary_key=True. Первинний ключ використовується для унікальної ідентифікації кожного запису в таблиці.

```
pulse = db.Column(db.Integer, nullable=False)
```

Поле pulse зберігає значення пульсу. Воно також має тип Integer і не може бути порожнім (nullable=False). Це означає, що при створенні нового запису в таблиці обов'язково потрібно вказати значення для цього поля.

```
oxygen = db.Column(db.Integer, nullable=False)
```

Поле oxygen зберігає рівень кисню в крові. Воно також має тип Integer і не може бути порожнім (nullable=False), що гарантує, що кожен запис містить значення рівня кисню.

```
timestamp = db.Column(db.DateTime, default=datetime.now)
```

Поле timestamp зберігає дату та час створення запису. Воно має тип DateTime і за замовчуванням встановлюється на поточний час (default=datetime.now). Це поле корисне для відстеження, коли саме були зібрані дані.

```
def __repr__(self):
```

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

```
return f'<HealthData {self.pulse}, {self.oxygen},  
{self.timestamp}>'
```

Метод `__repr__` визначає текстове представлення об'єкта моделі. Це особливо корисно для відлагодження, оскільки дозволяє легко бачити значення полів об'єкта при його виведенні на екран. У цьому випадку метод повертає рядок, який містить значення пульсу, рівня кисню та часу запису. Нижче зображено схему бази даних (рисунок 3.3).

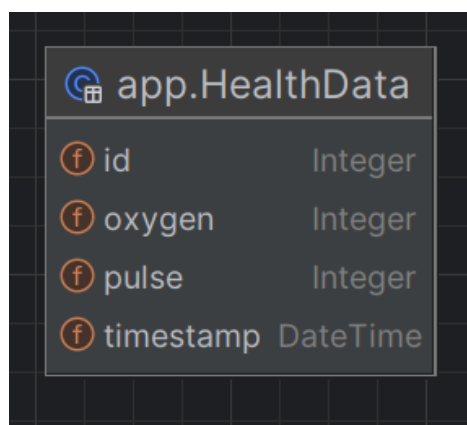


Рисунок 3.3 — Схема бази даних

3.3 Опис схем апаратної частини

Нижче зображено схему зв'язків між компонентами (рисунок 3.4).

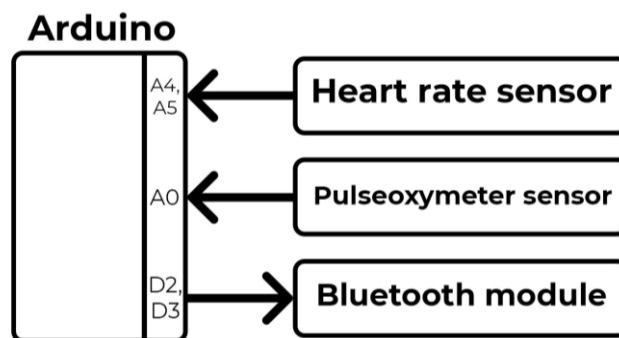


Рисунок 3.4 — Схема електрична функціональна

Ця схема ілюструє високорівневі зв'язки між основними компонентами системи моніторингу стану здоров'я, яка використовує Arduino UNO, датчики пульсу та пульсоксиметр, а також Bluetooth модуль для бездротового зв'язку з іншими пристроями.

Arduino виступає як центральний контролер системи. Він забезпечує збір даних з датчиків і керує комунікацією через Bluetooth модуль.

Аналоговий вхід A0 використовується для зчитування даних з пульсоксиметра, який міряє рівень кисню в крові та пульс.

Цифрові входи/виходи D2 та D3 використовуються для зв'язку з Bluetooth модулем для передачі даних.

Датчик пульсу (Heart rate sensor) моніторить частоту серцебиття. Дані з датчика передаються на аналогові входи A4 та A5 Arduino для обробки.

Пульсоксиметр (Pulseoxymeter sensor) вимірює рівень кисню в крові та пульс людини. Він підключений до аналогового входу A0 Arduino, щоб передавати зібрані дані контролеру.

Bluetooth модуль забезпечує бездротовий зв'язок між Arduino та іншими зовнішніми пристроями, такими як смартфони або комп'ютери. Це дозволяє передавати зібрані дані для подальшого аналізу або відображення.

Ця схема демонструє, як Arduino виступає як центральний вузол, що збирає інформацію з датчиків пульсу та пульсоксиметра і передає цю інформацію через Bluetooth для подальшого аналізу або відображення на зовнішніх пристроях.

Така система може бути використана для моніторингу стану здоров'я пацієнтів або спортсменів у реальному часі.

схема допомагає зрозуміти як підключені і налаштовані всі апаратні компоненти для правильної роботи системи.

3.4 Інструкції користувача

Ця інструкція описує процес налаштування та використання системи моніторингу стану здоров'я з пульсоксиметром та датчиком серцевого ритму, підключеними до Arduino UNO.

Система здатна збирати дані про пульс та рівень кисню в крові, а також передавати ці дані через Bluetooth.

Переконайтеся, що в комплекті є наступні компоненти:

- Arduino UNO;
- Bluetooth модуль HC-06;
- пульсоксиметр MH-ET LIVE;
- датчик серцевого ритму;
- кабелі та з'єднувачі.

Збірка системи:

1. підключити датчик пульсу та пульсоксиметр до відповідних портів на Arduino UNO згідно схеми підключення;
2. підключити Bluetooth модуль до відповідних TX і RX пінів на Arduino;
3. підключити живлення до Arduino;
4. перевірка з'єднань;
5. всі кабелі мають міцно з'єднані контакти;
6. немає пошкоджень кабелів або компонентів;
7. увімкнення Arduino, система автоматично розпочне збір даних з датчиків.
8. Надягнути датчик пульсу на палець або іншу частину тіла, де можна виміряти пульс.
9. Дані будуть відображені на підключеному до системи пристрої через Bluetooth.

10. Відкрийте додаток на смартфоні або програму на комп'ютері, яка підтримує Bluetooth модуль HC-06.
11. Переконайтеся, що ваш смартфон або комп'ютер підключений до Bluetooth модуля системи.
12. Можна бачити поточні показники пульсу та рівня кисню в крові на екрані підключеного пристрою.
13. Перевірте, чи увімкнений Bluetooth на вашому смартфоні чи комп'ютері.
14. Перезавантаження Arduino:
15. Переконайтеся, що датчик правильно закріплений на тілі.
16. Уникайте руху:

3.5 Висновки до розділу 3

Третій розділ дипломної роботи зосереджений на детальному описі процесу програмно-апаратної реалізації кіберфізичної системи моніторингу стану здоров'я. В цьому розділі розглядаються такі аспекти, як реалізація основних модулів системи, створення бази даних для збереження та аналізу зібраних даних, а також апаратні схеми, що забезпечують функціонування системи. Особлива увага приділяється інструкціям для користувачів, що дозволяють ефективно використовувати систему для моніторингу особистого здоров'я.

У розділі ретельно розглянуто реалізацію основних модулів системи, таких як збір та обробка даних з датчиків, керування комунікацією між компонентами системи, а також збереження та аналіз зібраних даних. Висвітлено методи та алгоритми, що використовуються для опрацювання та інтерпретації даних з датчиків, а також розглянуто технології, які використовуються для забезпечення безперервного збереження та доступу до зібраних медичних даних.

Подальше викладення присвячено апаратним аспектам системи, зокрема апаратним схемам, які забезпечують функціонування системи. Описуються зв'язки та взаємодія між різними апаратними компонентами, їх роль у забезпеченні роботи системи моніторингу стану здоров'я.

Завершальна частина розділу включає інструкції для користувачів. Ці інструкції надають детальний опис процесу використання системи, включаючи підготовку до роботи, запуск системи, способи моніторингу та аналізу зібраних даних, а також рекомендації щодо усунення можливих проблем.

У результаті інтеграції різноманітних компонентів та чітких інструкцій користувач може легко впровадити систему у своє повсякденне життя. Ця система дозволяє ефективно моніторити та аналізувати особисте здоров'я, забезпечуючи користувачеві зручний та доступний інструмент для підтримки здорового способу життя.

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 55
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У рамках виконання дипломної роботи було розроблено та впроваджено кіберфізичну систему моніторингу стану здоров'я на базі мікроконтролера Arduino, яка дозволяє користувачам вимірювати життєво важливі показники, такі як пульс та рівень кисню в крові. Система складається з апаратної частини, включаючи Arduino UNO, датчик пульсу, пульсоксиметр та Bluetooth модуль для бездротової передачі даних, а також програмного забезпечення, яке забезпечує обробку та відображення зібраних даних.

Основні досягнення проекту включають:

1) Ефективна інтеграція апаратних компонентів

Вдало підібрані та налаштовані датчики та модулі дозволили точно збирати фізіологічні дані. Arduino UNO виступає центральним контролером, який обробляє дані з датчиків. Датчик пульсу фіксує серцевий ритм, передаючи аналогові сигнали на Arduino, де вони обробляються. Пульсоксиметр, що вимірює рівень кисню в крові, також підключений до аналогових входів мікроконтролера, забезпечуючи точність зчитуваних показників.

2) Розробка надійного програмного забезпечення

Програмне забезпечення для Arduino було створене з урахуванням необхідності стабільної та безперебійної роботи системи. Збір даних з датчиків і їх обробка здійснюється в реальному часі. Супутній серверний софт на базі Flask забезпечує зручний доступ до зібраних даних через веб-інтерфейс, дозволяючи користувачам переглядати свої показники здоров'я у браузері.

3) Забезпечення мобільності та доступності системи

Використання Bluetooth модуля для передачі даних дозволило користувачам легко отримувати інформацію на свої портативні пристрої, такі як смартфони або планшети. Це забезпечує високу мобільність і зручність використання системи, оскільки користувачі можуть моніторити свій стан здоров'я у будь-який час і в будь-якому місці.

					КвРКІ. 2101067.21.01.17 ПЗ	Арк. 56
Зм.	Арк.	№ докум.	Підпис	Дата		

Практичне значення роботи демонструється через здатність системи надавати користувачам інструмент для регулярного моніторингу власного здоров'я, що може сприяти ранньому виявленню потенційних загроз здоров'ю. Така система особливо корисна для людей із хронічними захворюваннями, які потребують постійного нагляду за критичними показниками здоров'я. Регулярний моніторинг дозволяє вчасно реагувати на зміни у стані здоров'я і, за необхідності, коригувати лікування та поведінку.

Використання цієї системи може бути розширене до включення додаткових датчиків та функціональних можливостей, таких як моніторинг температури тіла, вимірювання артеріального тиску, що зробить її ще більш універсальною та корисною. Також перспективним є розвиток інтеграції системи з медичними базами даних і клінічними системами для забезпечення кращої координації лікування пацієнтів, що дозволить лікарям оперативніше отримувати інформацію про стан здоров'я пацієнтів і приймати більш обґрунтовані рішення щодо їх лікування.

У подальшому, можливе розширення проекту з включенням штучного інтелекту для аналізу зібраних даних. Використання алгоритмів машинного навчання дозволить не тільки моніторити, але й прогнозувати можливі зміни у стані здоров'я користувачів на основі накопиченої інформації. Це значно підвищить вартість системи та її ефективність у плані допомоги користувачам у підтримці та управлінні їх здоров'ям. Завдяки цьому, система зможе надавати рекомендації щодо здорового способу життя, попереджати про можливі загрози та допомагати у своєчасному прийнятті рішень щодо медичного втручання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Інтернет речей: що це таке та де застосовують? URL: <https://cybercalm.org/novyny/internet-rechej-shho-tse-take-ta-de-zastosovuyut/> (дата звернення: 15.06.2024)
2. IoT, або Інтернет речей URL: <https://hub.kyivstar.ua/articles/iot-abo-internet-rechej> (дата звернення: 15.06.2024)
3. Основи Інтернету Речей: Датчики та живлення URL: <http://edu.asu.in.ua/mod/book/view.php> (дата звернення: 15.06.2024)
4. Інтернет речей (IoT) – суть, технології і приклади URL: <https://termin.in.ua/internet-rechej-iot/> (дата звернення: 15.06.2024)
5. К. О. Домрачева, Н. М. Довженко, В. В. Дмитренко Аналіз технологій та стандартів зв'язку для мережі IoT Наук. зап. Укр. наук.–дослід. ін-ту зв'язку. 2019. С. 54–62
6. Проектування Інтернет речей (IoT) URL: <https://www.slideshare.net/slideshow/iot-79608563/79608563> (дата звернення: 15.06.2024)
7. IoT у медицині: від теорії до реальних кейсів URL: <https://hub.kyivstar.ua/articles/iot-u-medyczyni-vid-teoriyi-do-realnyh-kejsiv> (дата звернення: 15.06.2024)
8. «Надійне керування формуванням для мережевих роботизованих систем, що використовують негативну уявну динаміку» 2022. <https://www.sciencedirect.com/science/article/pii/S0005109822000802> (дата звернення 15.06.2024)
9. Що являє собою інтернет речей та які перспективи розвитку має цей напрям URL: <https://gsminfo.com.ua/120474-shho-yavlyaye-soboyu-internet-rechej-ta-yaki-perspektyvu-rozvytku-maye-czej-napryam.html> (дата звернення 15.06.2024)
10. IoT, або Інтернет речей URL: <https://hub.kyivstar.ua/articles/iot-abo-internet-rechej> (дата звернення: 16.05.2024)

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

11. Інтернет речей: що це таке та де застосовують? URL: <https://cybercalm.org/novyny/internet-rechej-shho-tse-take-ta-de-zastosovuyut/> (дата звернення: 16.05.2024)

12. Основи Інтернету Речей: Датчики та живлення URL: <http://edu.asu.in.ua/mod/book/view.php> (дата звернення: 16.05.2024)

13. Передача даних в архітектурі IoT: MQTT URL: <http://edu.asu.in.ua/mod/book/tool/print/index.php>. (дата звернення: 16.05.2024)

14. Столлінгс, Вільям (2016). Основи сучасної мережі: SDN, NFV, QoE, IoT та Cloud URL: <https://www.coursesidekick.com/information-systems/4386230> (дата звернення: 16.05.2024)

15. "Шлях стека" URL: www.industryweek.com (дата звернення: 16.05.2024)

16. «Ринок корпоративного Інтернету речей». Business Insider URL: <https://www.telesphera.net/news/majbutne-internetu-rechej.html> (дата звернення: 16.05.2024)

17. «Розширена система безпеки для побутової техніки в розумному домі». Людиноорієнтовані обчислювальні та інформаційні науки. URL: <https://hcis-journal.springeropen.com/articles/10.1186/s13673-017-0087-4> (дата звернення: 16.05.2024)

18. «Соціально інтелектуальні інтерфейси для підвищення енергоінформованості вдома». URL: <https://www.sciencedirect.com/science/article/pii/S0306261916303579> (дата звернення: 16.05.2024)

19. «HomeKit – розробник Apple». URL: <https://www.developer.apple.com> (дата звернення: 16.05.2024)

20. «Ось все, що вам потрібно знати про Apple HomeKit» URL: <https://journal.ilounge.ua/ua/review/apple-homekit> (дата звернення: 17.05.2024)

21. «Пристрої HomeKit стають доступнішими, оскільки Lenovo анонсує лінійку Smart Home Essentials». URL: <https://www.actualidadiphone.com/uk/lenovo->

se-suma-al-carro-de-homekit-con-sus-smart-home-essentials/ (дата звернення: 17.05.2024)

22. «Інтернет здоров'я: до інтелектуального моніторингу життєво важливих показників у лікарняних палатах.» URL: <https://www.dusuniot.com/uk/resources/technical-brief/the-future-of-bluetooth-roaming-to-make-staying-connected-easier-than-ever/> (дата звернення: 17.05.2024)

23. «Інтернет медичних речей: огляд останніх внесків, що стосуються кіберфізичних систем у медицині» URL: <https://stfalcon.com/uk/blog/post/how-the-internet-of-medical-things-revolutionizes-healthcare> (дата звернення: 17.05.2024)

24. Тополь, Ерік (2016). Пацієнт побачить вас зараз: майбутнє медицини у ваших руках. Основні книги. ISBN 978-0465040025.

25. «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-challenges> (дата звернення: 17.05.2024)

26. «Делойт Центр рішень для охорони здоров'я». URL: <https://www2.deloitte.com/ua/uk/pages/life-sciences-and-healthcare/solutions/health-care.html> (дата звернення: 17.05.2024)

27. Управління мережами з обмеженими пристроями: випадки використання". URL: <https://fiberroad.com/uk/resources/glossary/serial-communication-in-iot/> (дата звернення: 17.05.2024)

28. «mHealth. Нові горизонти здоров'я через мобільні технології» URL: <https://mediacom.com.ua/virtualna-realnist-u-mobilnix-dodatках-novij-svit-mozhливостej/> (дата звернення: 18.05.2024)

29. «Потенціал Інтернету речей мобільного здоров'я «m-ІоТ» для неінвазивного визначення рівня глюкози». URL: <https://cases.media/article/cikava-statistika-ta-fakti-pro-internet-rechei-iot-rozmir-rinku-vikoristannya-ta-proгнози> (дата звернення: 18.05.2024)

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

30. «Мультиплексне тестування на місці надання медичної допомоги – хРОСТ». URL: <https://www.siemens-healthineers.com/ua/point-of-care-testing> (дата звернення: 18.05.2024)

31. «Що таке НІЕ? | HealthIT.gov». URL: <https://www.healthit.gov>. (дата звернення: 18.05.2024)

32. Олійник В. В. Arduino: від простого до складного. Київ: Ентузіаст, 2017. 320 с.

33. Шмаков М. А. Arduino: основи програмування та роботи з датчиками. Харків: Фоліо, 2016. 272 с.

34. Шмаков М. А. Arduino: інтерфейси та підключення. Харків: Фоліо, 2016. 240 с.

35. Ковальчук І. М.. Arduino: практичний курс для початківців. Київ: ВHV, 2019. 208 с.

36. Петренко О. В. Arduino: основи програмування та роботи з датчиками. Харків: Фоліо, 2018. 272 с.

37. Іванов С. М. Arduino: від простого до складного. Київ: Ентузіаст, 2020. 320 с.

38. Белецкий С. В. Arduino: від теорії до практики. Київ: ДМК Пресс, 2018. 320 с.

39. Бойко А. В.. Arduino: все, що потрібно знати. Київ: ВHV, 2017. 400 с.

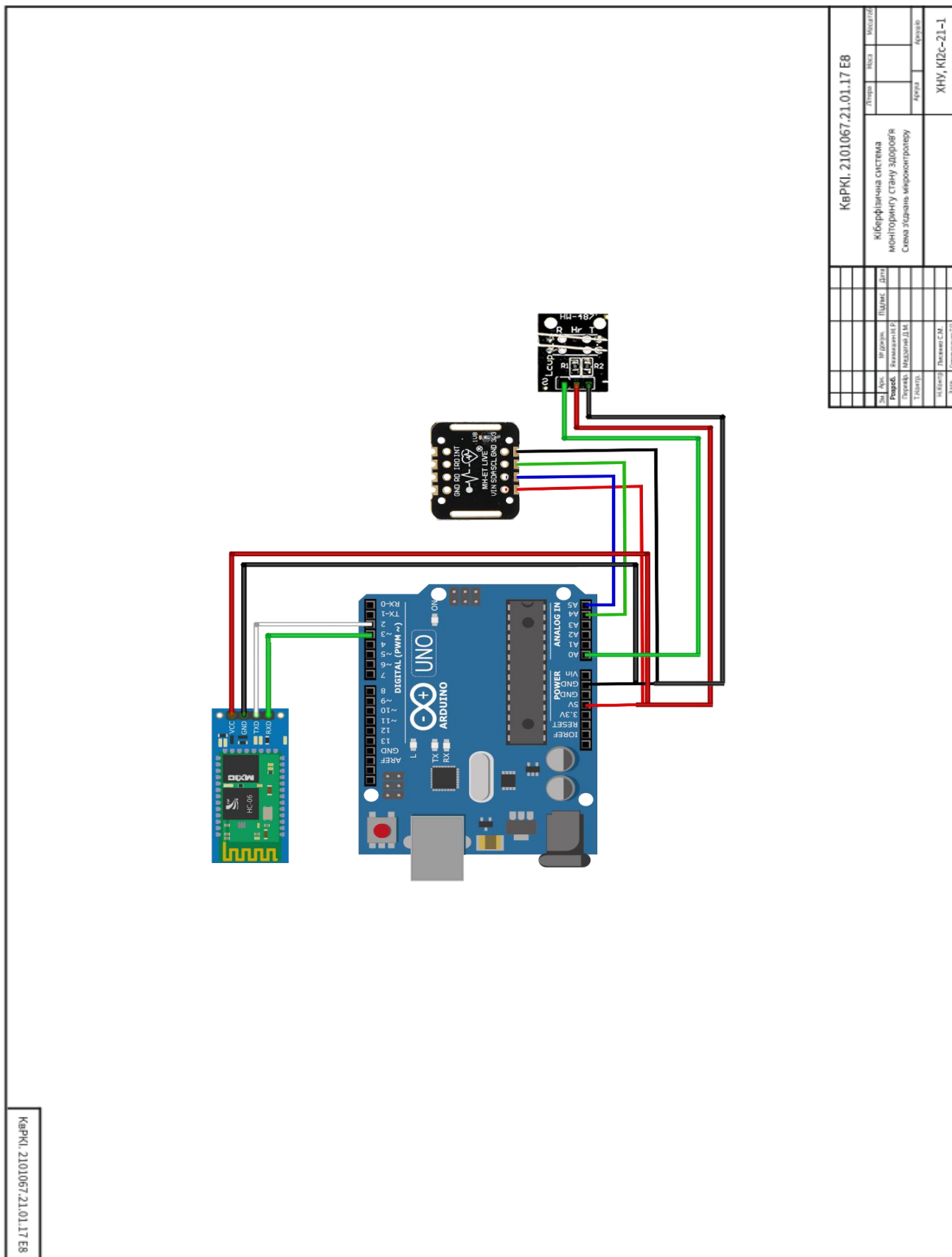
40. Сидоренко А. А. Arduino: секрети майстерності. Київ: ВHV, 2021. 352 с.

41. «Інтернет речей. Руйнування традиційних бізнес-моделей» URL: <https://metinvest.digital/ua/page/internet-veshchej-chem-on-mozhet-byt-polezen-dlya-biznesa?culture=uk> (дата звернення: 19.05.2024)

					КВРКІ. 2101067.21.01.17 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток А (обов'язковий)

Копія креслення «Схема з'єднань мікроконтролеру»

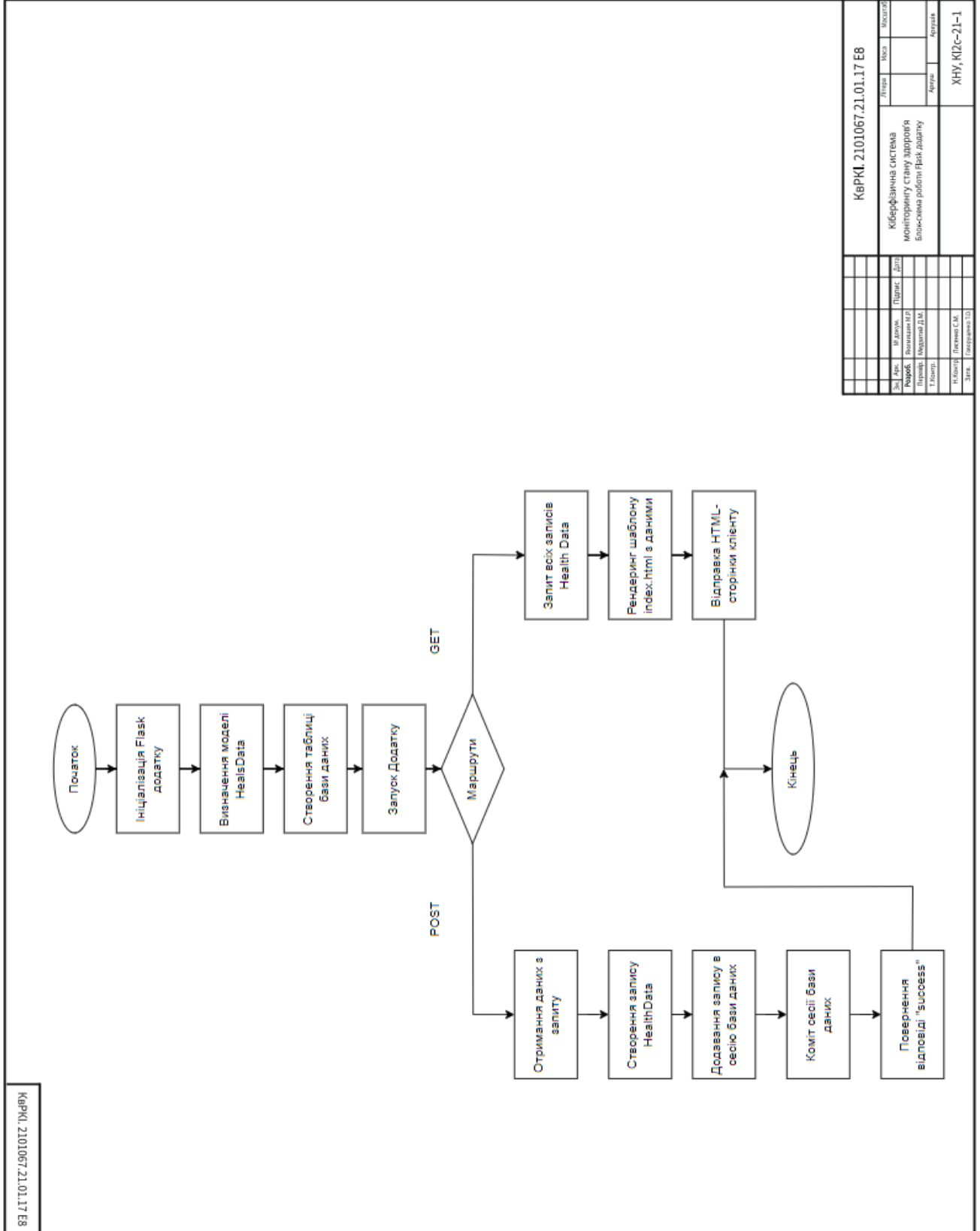


КВРКІ. 2101067.21.01.17.Е8

КВРКІ. 2101067.21.01.17.Е8		Лист	№02	Модифікація
Кибернічна система моніторингу стану здоров'я		Вид		
Схема з'єднань мікроконтролеру		Код		архів
№	Код	Вірність	Підпис	Дата
1	001	Висхідна		
2	002	Висхідна		
3	003	Висхідна		
4	004	Висхідна		
5	005	Висхідна		
6	006	Висхідна		
7	007	Висхідна		
8	008	Висхідна		
9	009	Висхідна		
10	010	Висхідна		
11	011	Висхідна		
12	012	Висхідна		
13	013	Висхідна		
14	014	Висхідна		
15	015	Висхідна		
16	016	Висхідна		
17	017	Висхідна		
18	018	Висхідна		
19	019	Висхідна		
20	020	Висхідна		
21	021	Висхідна		
22	022	Висхідна		
23	023	Висхідна		
24	024	Висхідна		
25	025	Висхідна		
26	026	Висхідна		
27	027	Висхідна		
28	028	Висхідна		
29	029	Висхідна		
30	030	Висхідна		
31	031	Висхідна		
32	032	Висхідна		
33	033	Висхідна		
34	034	Висхідна		
35	035	Висхідна		
36	036	Висхідна		
37	037	Висхідна		
38	038	Висхідна		
39	039	Висхідна		
40	040	Висхідна		
41	041	Висхідна		
42	042	Висхідна		
43	043	Висхідна		
44	044	Висхідна		
45	045	Висхідна		
46	046	Висхідна		
47	047	Висхідна		
48	048	Висхідна		
49	049	Висхідна		
50	050	Висхідна		
51	051	Висхідна		
52	052	Висхідна		
53	053	Висхідна		
54	054	Висхідна		
55	055	Висхідна		
56	056	Висхідна		
57	057	Висхідна		
58	058	Висхідна		
59	059	Висхідна		
60	060	Висхідна		
61	061	Висхідна		
62	062	Висхідна		
63	063	Висхідна		
64	064	Висхідна		
65	065	Висхідна		
66	066	Висхідна		
67	067	Висхідна		
68	068	Висхідна		
69	069	Висхідна		
70	070	Висхідна		
71	071	Висхідна		
72	072	Висхідна		
73	073	Висхідна		
74	074	Висхідна		
75	075	Висхідна		
76	076	Висхідна		
77	077	Висхідна		
78	078	Висхідна		
79	079	Висхідна		
80	080	Висхідна		
81	081	Висхідна		
82	082	Висхідна		
83	083	Висхідна		
84	084	Висхідна		
85	085	Висхідна		
86	086	Висхідна		
87	087	Висхідна		
88	088	Висхідна		
89	089	Висхідна		
90	090	Висхідна		
91	091	Висхідна		
92	092	Висхідна		
93	093	Висхідна		
94	094	Висхідна		
95	095	Висхідна		
96	096	Висхідна		
97	097	Висхідна		
98	098	Висхідна		
99	099	Висхідна		
100	100	Висхідна		
101	101	Висхідна		
102	102	Висхідна		
103	103	Висхідна		
104	104	Висхідна		
105	105	Висхідна		
106	106	Висхідна		
107	107	Висхідна		
108	108	Висхідна		
109	109	Висхідна		
110	110	Висхідна		
111	111	Висхідна		
112	112	Висхідна		
113	113	Висхідна		
114	114	Висхідна		
115	115	Висхідна		
116	116	Висхідна		
117	117	Висхідна		
118	118	Висхідна		
119	119	Висхідна		
120	120	Висхідна		
121	121	Висхідна		
122	122	Висхідна		
123	123	Висхідна		
124	124	Висхідна		
125	125	Висхідна		
126	126	Висхідна		
127	127	Висхідна		
128	128	Висхідна		
129	129	Висхідна		
130	130	Висхідна		
131	131	Висхідна		
132	132	Висхідна		
133	133	Висхідна		
134	134	Висхідна		
135	135	Висхідна		
136	136	Висхідна		
137	137	Висхідна		
138	138	Висхідна		
139	139	Висхідна		
140	140	Висхідна		
141	141	Висхідна		
142	142	Висхідна		
143	143	Висхідна		
144	144	Висхідна		
145	145	Висхідна		
146	146	Висхідна		
147	147	Висхідна		
148	148	Висхідна		
149	149	Висхідна		
150	150	Висхідна		
151	151	Висхідна		
152	152	Висхідна		
153	153	Висхідна		
154	154	Висхідна		
155	155	Висхідна		
156	156	Висхідна		
157	157	Висхідна		
158	158	Висхідна		
159	159	Висхідна		
160	160	Висхідна		
161	161	Висхідна		
162	162	Висхідна		
163	163	Висхідна		
164	164	Висхідна		
165	165	Висхідна		
166	166	Висхідна		
167	167	Висхідна		
168	168	Висхідна		
169	169	Висхідна		
170	170	Висхідна		
171	171	Висхідна		
172	172	Висхідна		
173	173	Висхідна		
174	174	Висхідна		
175	175	Висхідна		
176	176	Висхідна		
177	177	Висхідна		
178	178	Висхідна		
179	179	Висхідна		
180	180	Висхідна		
181	181	Висхідна		
182	182	Висхідна		
183	183	Висхідна		
184	184	Висхідна		
185	185	Висхідна		
186	186	Висхідна		
187	187	Висхідна		
188	188	Висхідна		
189	189	Висхідна		
190	190	Висхідна		
191	191	Висхідна		
192	192	Висхідна		
193	193	Висхідна		
194	194	Висхідна		
195	195	Висхідна		
196	196	Висхідна		
197	197	Висхідна		
198	198	Висхідна		
199	199	Висхідна		
200	200	Висхідна		
201	201	Висхідна		
202	202	Висхідна		
203	203	Висхідна		
204	204	Висхідна		
205	205	Висхідна		
206	206	Висхідна		
207	207	Висхідна		
208	208	Висхідна		
209	209	Висхідна		
210	210	Висхідна		
211	211	Висхідна		
212	212	Висхідна		
213	213	Висхідна		
214	214	Висхідна		
215	215	Висхідна		
216	216	Висхідна		
217	217	Висхідна		
218	218	Висхідна		
219	219	Висхідна		
220	220	Висхідна		
221	221	Висхідна		
222	222	Висхідна		
223	223	Висхідна		
224	224	Висхідна		
225	225	Висхідна		
226	226	Висхідна		
227	227	Висхідна		
228	228	Висхідна		
229	229	Висхідна		
230	230	Висхідна		
231	231	Висхідна		
232	232	Висхідна		
233	233	Висхідна		
234	234	Висхідна		
235	235	Висхідна		
236	236	Висхідна		
237	237	Висхідна		
238	238	Висхідна		
239	239	Висхідна		
240	240	Висхідна		
241	241	Висхідна		
242	242	Висхідна		
243	243	Висхідна		
244	244	Висхідна		
245	245	Висхідна		
246	246	Висхідна		
247	247	Висхідна		
248	248	Висхідна		
249	249	Висхідна		
250	250	Висхідна		
251	251	Висхідна		
252	252	Висхідна		
253	253	Висхідна		
254	254	Висхідна		
255	255	Висхідна		
256	256	Висхідна		
257	257</			

Додаток В (обов'язковий)

Копія креслення блок-схеми роботи Flask додатку



Додаток Г

Лістинг коду апаратної частини

```
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"

MAX30105 particleSensor;

// Пини підключення датчика пульсу KY-039
const int pulsePin = A0;

// Змінні для зберігання даних
long lastBeat = 0;
float beatAvg = 0;

void setup() {
  Serial.begin(9600); // Ініціалізація послідовного порту для Bluetooth
  while (!Serial);

  // Ініціалізація пульсоксиметра
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) {
    Serial.println("MAX30105 не знайдено. Перевірте підключення.");
    while (1);
  }

  particleSensor.setup(); // Налаштування пульсоксиметра з параметрами по
  // замовчуванню
  particleSensor.setPulseAmplitudeRed(0x0A); // Легке підсвічування червоного
  // світлодіода для сигналізації роботи
  particleSensor.setPulseAmplitudeGreen(0); // Вимкнути зелений світлодіод
}

void loop() {
  // Зчитування даних з датчика пульса KY-039
  int pulseValue = analogRead(pulsePin);

  // Зчитування даних з пульсоксиметра MAX30105
  long irValue = particleSensor.getIR();

  if (checkForBeat(irValue) == true) {
    long delta = millis() - lastBeat;
    lastBeat = millis();
  }
}
```

```
    beatAvg = 60 / (delta / 1000.0);  
  }  
  
  // Відправка даних на телефон через Bluetooth  
  Serial.print("Pulse Sensor Value: ");  
  Serial.print(pulseValue);  
  Serial.print(", Heart Rate: ");  
  Serial.print(beatAvg);  
  Serial.print(", IR Value: ");  
  Serial.println(irValue);  
  
  delay(1000);  
}
```

Ім'я користувача:
Кафедра КІ

ID перевірки:
1016361031

Дата перевірки:
14.06.2024 17:57:38 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
15.06.2024 09:58:01 EEST

ID користувача:
100005591

Назва документа: Якимишин_Кіберфізична система моніторингу стану здоров'я

Кількість сторінок: 64 Кількість слів: 12890 Кількість символів: 98392 Розмір файлу: 773.58 KB ID файлу: 1016165906

6.48% Схожість

Найбільша схожість: 1.18% з Інтернет-джерелом (<https://er.nau.edu.ua/bitstream/NAU/63366/1/%d0%9d%d0%9d%d0%9d...>)

5.96% Джерела з Інтернету

192

Сторінка 66

2% Джерела з Бібліотеки

125

Сторінка 68

1.17% Цитат

Цитати

8

Сторінка 69

Посилання

1

Сторінка 69

0% Вилучень

Немає вилучених джерел

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 11%

ID: 130583 Назва: БКР Кіберфізична система моніторингу стану здоров'я Додано в БД: 2024-06-14 Автора: М.Р. Якимішин Керівники: Д.М.Медзатий Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	82374	677	541 (1%)	9 (1%)

Джерело плагиату

ID	Опис	Наявність плагиату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Якимишин Максим Романович

Тема: Кіберфізична система моніторингу стану здоров'я

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 55

1. Короткий зміст роботи та прийнятих рішень: метою кваліфікаційної роботи є визначення умов та особливостей застосування обладнання для моніторингу стану здоров'я.
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: Перший розділ дипломної роботи охоплює всебічний аналіз структурних і функціональних особливостей кіберфізичної системи моніторингу здоров'я Другий розділ дипломної роботи присвячений детальному проектуванню кіберфізичної системи моніторингу стану здоров'я, яка об'єднує як апаратні, так і програмні підсистеми, методи їх взаємодії та детальний опис основних модулів, що становлять цю систему. У цьому розділі детально розглядаються апаратні компоненти, такі як Arduino UNO, різноманітні датчики, включаючи датчики пульсу та рівня кисню в крові, а також програмне забезпечення, що забезпечує збір, обробку та аналіз зібраних даних. В третьому розділі кваліфікаційної роботи розглядаються такі аспекти, як реалізація основних модулів системи, створення бази даних для збереження та аналізу зібраних даних, а також апаратні схеми, що забезпечують функціонування системи.
4. Позитивні сторони роботи: висока практична цінність роботи.
5. Негативні сторони роботи: обмежена кількість вимірюваних показників.

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.

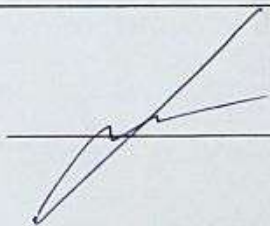
8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно (3.5) D

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) Гішов

Володимир Юрійович, доцент кафедри кібербезпеки

“24” 06 2024 р.

 (підпис)

Завідувачу кафедри КІС
д-р.техн.наук, проф. Говорушенко Т. О.

Якимишина Максима Романовича

ІІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-21-1

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 (Unicheck 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.

22 квітня 2024 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Кіберфізична система моніторингу стану здоров'я

Автор: Якимишин Максим Романович

Спеціальність: 123- Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Медзатий Дмитро Миколайович, к.т.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

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

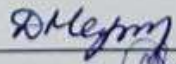
- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-20 джерелами на один фрагмент речення;

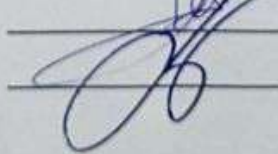
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 6.48% і адресується до 317 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС





Д. М. Медзатий

С. М. Лисенко

Т. О. Говорущенко