Міністерство освіти і науки України
Державний університет «Одеська політехніка»

Інститут штучного інтелекту та робототехніки
Кафедра «Комп'ютерні системи»

Бойко Віктор Олександрович,
студент групи УК-161

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Дослідження комп'ютерної системи діагностики людини на смарт датчиках

Спеціальність 123 Комп'ютерна інженерія
Спеціалізація: Спеціалізовані к омп'ютерні системи

Керівник:
Ступень Павло В'ячеславович ,
кандидат техн. наук, доцент

Одеса – 2021

ЗМІСТ

ВСТУП

Дослідження та розробка комп’ютерних систем діагностики людини на сьогоднішній день є важливою проблемою. Весь світ намагається знаходити нові методи виявлення та діагностики, для встановлення діагнозів та попередження важкого перебігу хвороб. На разі чи мало держав, намагаються стримати коронавірусну інфекцію COVID-19, для цього витрачаються чимало коштів, та проводиться велика кількість різноманітних досліджень, які спрямовані на розробку вакцин та систем діагностики.

Навіть після розробки вакцин, існує проблема яка пов’язана з хвилями хвороби, коли медична система не завжди може впоратись з великою кількістю хворих людей, в такій ситуації доводиться робити вибір на користь важко хворих, яких потрібно забезпечити кисневими концентраторами або ж апаратами ШВЛ (штучної вентиляції легень). Зараз для виявлення інфікованих використовується ПЛР тест (полімеразна ланцюгова реакція), на разі цей метод є одним з найефективніших при діагностиці інфекційних захворювань, але при цьому, метод застосовується зазвичай вже до контактних осіб для виявлення потенційних носіїв вірусу з метою запобігання подальшого розповсюдження.

Одним із шляхів зменшення смертності є постійний моніторинг стану хворого який має легку форму прояву корона вірусної хвороби оскільки в такій ситуації, якщо стан людини погіршиться дуже часто буває вже запізно саме тому важливо контролювати рівень насиченості крові киснем адже це саме перше, що вражає корона вірусна хвороба. А для зменшення ризику інфікування осіб які перебувають вдома з хворим, важливо забезпечити дистанційний моніторинг стану, і сповіщення в разі потенційно небезпечної ситуації.

Саме тому **тема** кваліфікаційної роботи: «Дослідження комп’ютерних систем діагностики людини на смарт датчиках», є важливою адже дозволить

попереджати випадки пізнього звернення до лікаря а також зменшить ризики інфікування людей, що перебувають з хворим в одному середовищі.

Метою кваліфікаційної роботи є дослідження комп'ютерної системи діагностики людини за допомогою смарт датчиків.

Об'єкт дослідження – процес діагностики стану людини.

Предмет дослідження – комп'ютерна система діагностики стану людини та перевірка її ефективності.

Комп'ютерні системи діагностики мають на меті, встановлення вірного діагнозу людини, або ж виявлення хвороб, моніторинг стану здоров'я. Методи діагностики стану здоров'я людини бувають різними починаючи від звичайної пальпації (дотиків за допомогою пальців або ж долонь) до магнітно-резонансної томографії. Методи дослідження комп'ютерної системи полягають в постійному спостереженні за хворим та порівнянні показників насиченості крові та серцебиття. Це основні клінічні симптоми хвороби, але оскільки кожна людина індивідуальна то ці параметри можуть відрізнятись тому потрібно мати можливість налаштувати систему під кожну окрему людину після чого перевіряти вже відкалібровані дані. Для дослідження та покращення параметрів потрібно розробити комп'ютерну систему яка здатна вимірювати визначенні параметри та порівнювати їх. До таких систем можна віднести пульсоксиметри вони вимірюють насиченість крові киснем та серцебиття, при цьому зазвичай ці виміри відбуваються одноразово, а не впродовж тривалого часу. Стан здоров'я людини це сукупність різних факторів як зовнішніх так і внутрішніх які впливають на нього. Всі ці фактори можуть по різному впливати на нього звичайні фізичні навантаження можуть прискорити серцебиття. У звичайної людини норма сатурації (насиченості крові киснем) починається від 95% і більше, сатурація нижче 94% вважається заниженою, а 92% є приводом для госпіталізації, але при цьому є винятки так наприклад люди з захворюваннями легень або серцево судинної системи можуть мати показник від 88% до 92% і для них це буде вважатись нормальним показником. Потрібно постійно контролювати показники

сатурації та серцебиття для запобігання критичних станів або ж не своєчасного звернення до лікаря.

Технології сьогодення дозволяють використовувати невеликі за габаритами датчики, для вимірювання різноманітних параметрів тіла. Всі ці технології за умови правильно використання дозволяють проводити моніторинг стану людини в реальному часі.

Важливими параметрами для систем діагностики людини є їх точність, стабільність роботи, автономність та можливість постійного моніторингу.

1 АНАЛІЗ ПРОБЛЕМИ МОНІТОРИНГУ КОРОНАВІРУСУ

Система, що розробляється покращує можливості моніторингу за станом хворого вдома, за допомогою постійного моніторингу завдяки високій автономності пристрою та дистанційній передачі даних за допомогою технології Wi-Fi. Після аналізу запропонованих сучасних систем можна виділити певні потреби а саме:

- Покращити автономність, для підвищення можливості постійного моніторингу стану без додаткової потреби постійної підзарядки.
- Забезпечити можливість дистанційного контролю за станом хворого за допомогою смартфону. Відображати його показники на екрані за допомого додатку.
- Покращити систему сигналізації про критичні стани, несправності пристрою, або некоректні дані. В системі пропонується відправляти звукові та текстові сповіщення на екран смартфону, що дозволить відразу розуміти яка саме проблема потребує вирішення.
- Можливість калібрування пристрою за допомогою смартфону в залежності від індивідуальних особливостей людини. Тобто дати змогу користувачу власноруч налаштовувати кількість вимірів, дальність вимірювання.

1.1 Проблема моніторингу стану людини

Під час корона вірусної хвороби є дуже важливою проблемою сьогодення оскільки, на даний момент в Україні зафіксовано 3,6 млн захворювання та 94 тис з них летальні[1]. Медична система не завжди може впоратись з проблемою корона вірусної хвороби оскільки, кількість лікарів, місць в лікарнях та кисню є обмеженими. Саме тому не завжди є можливість забезпечити всіх хворих необхідним доглядом та увагою. Виникає потреба в виділенні особливо важких хворих та осіб які мають легку форму перебігу

хвороби. При цьому виникає наступна проблема у вигляді погіршення стану хворого з подальшими ускладненнями через несвоєчасне звернення до лікаря у зв'язку з відсутністю системи довготривалого моніторингу стану хворого. Саме тому важливим є дослідження та вдосконалення систем діагностики людини які здатні спостерігати за станом хворого, надати можливість без близького контакту контролювати дистанційно стан хворого, що зменшує ризики інфікування тих хто доглядає за хворим, та попереджувати про небезпечні станів. На рисунку 1.1 зображено поточну статистику хворих на корона вірус.

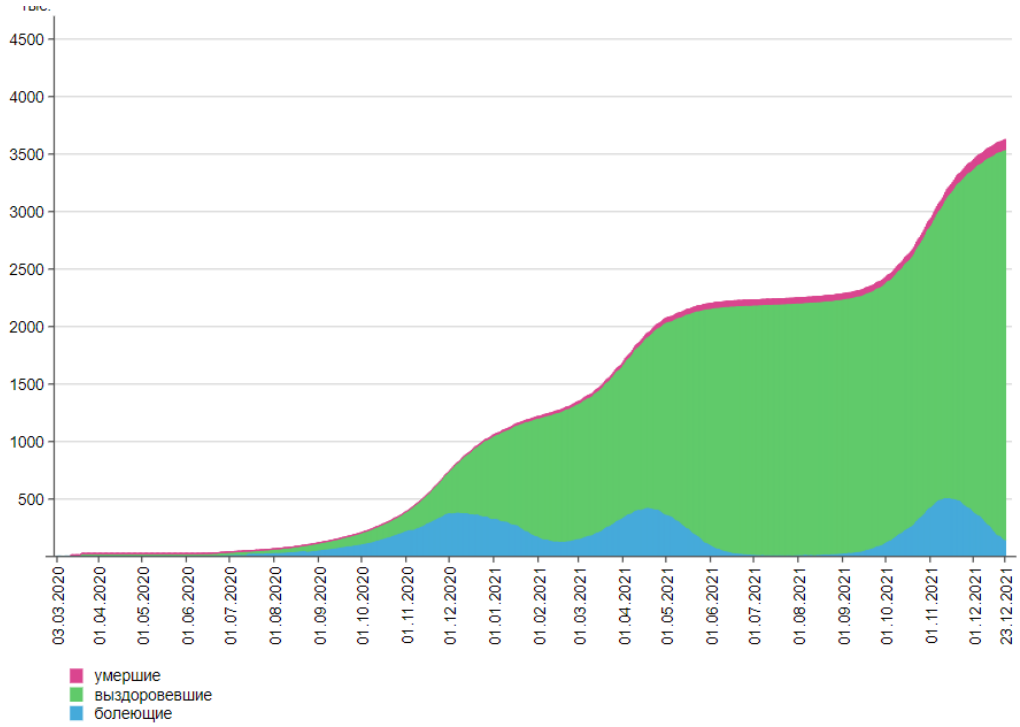

Рисунок 1.1 – Статистика інфікованих корона вірусною інфекцією в Україні

На сьогоднішній день існує чимало систем які здатні визначати сатурацію крові зазвичай для цього використовуються різноманітні пульсоксиметри, оскільки такі апарати як ІВЛ мають високу ціну не є зручними у використані в дома для хворих з легким перебігом хвороби саме пульсоксиметри є оптимальним варіантом для постійного моніторингу стану вдома. Якщо брати до уваги професійні пульсоксиметри які використовуються

на швидких для контролю стану пацієнта, то при діагностиці вдома ці системи є не дуже ефективними оскільки мають досить високу ціну, великі розміри та не включають в себе функцію передачі даних на відстані, при цьому розглядаючи більш компактні версії пульсоксиметрів можна виділити наступний недолік який полягає в відсутності можливості автономної роботи оскільки в середньому такі пульоксиметри при роботі витрачають приблизно 40 мА, врахувавши те, що в них використовуються елементи живлення які не можливо підзарядити а їхня ємність сягає 750 мАг виходить, що такий пристрій може пропрацювати на двох таких елементах живлення не більше 38 годин. Тому можна зробити висновок, що професійні медичні пульоксиметри хоч і є автономними та надійними, але при цьому мають досить великі розміри та ціну, якщо ж розглядати пульсоксиметри невеликих розмірів то вони більше підходять для одноразових вимірювань оскільки їх елементи живлення не розраховані на постійне використання. Не менш важливими при цьому є можливість передачі даних дистанційно, так деякі пальцеві пульсоксиметри включають в себе таку можливість, але при цьому це знову буде впливати на їх автономність та зменшувати можливість постійної роботи з 38 годин до 18, що не є прийнятним враховуючи, що система повинна працювати постійно і мати можливість підзарядки. Для професійних медичних пульсоксиметрів які наприклад використовуються в швидких дистанційна передача даних не є доречною. До того ж радіус дії таких пристроїв є досить обмеженим і сягає 15 метрів. Тому виникає потреба в підвищенні автономності таких пристроїв, зменшенні розмірів та ціни і покращенні можливостей дистанційного зв’язку для передачі даних.

1.2 Порівняння аналогів

Серед аналогів розглянемо дві конкретні моделі пристроїв один з яких представляє з себе професійний тобто його можна використовувати в швидких та лікарнях назва моделі «Окситест-1»[2], другим пульсоксиметром який буде розглядатись є побутова модель яка має назву «Fingertip BLS-1102B»

Якщо розглядати першу модель пульсоксиметра вона призначена для професійного використання та моніторингу стану хворого в автомобілях швидкої він має хорошу автономність: На рисунку 1.2 зображено медичний пульсоксиметр «Окситест-1».

Переваги:

- Можливість довготривалого спостереження
- Вбудований акумулятор
- Зарядка як від звичайної мережі так і від автомобільної
- Налаштування верхніх та нижніх граничних параметрів частоти пульсу та рівня кисню в крові
- Система звукової сигналізації
- Час автономної роботи 50 годин

Недоліки:

- Висока ціна
- Габарити
- Відсутність можливості передавати дані дистанційно

Рисунок 1.2 – Професійний пульсоксиметр «Окситест-1»

Якщо розглядати другу модель пульсоксиметра яка використовується в побуті, вона має досить компактні розміри, може використовуватись як засіб для одноразових вимірювань, оскільки не може працювати в режимі довготривалої діагностики через відсутність вбудованої батареї, має низьке енергоспоживання. На рисунку 1.3 можна побачити побутовий пульсоксиметр «Fingertip BLS-1102B».

Переваги:

- Низька ціна
- Мобільність
- Компактність
- Автоматичне відключення після вимірювань

Недоліки:

- Відсутність звукової сигналізації
- Не можливість постійного моніторингу
- Час автономної роботи 30 годин без можливості підзарядки

Рисунок 1.3 – Побутовий пульсоксиметр «Fingertip BLS-1102B»

Всі ці пристрої для вимірювань в своїй основі мають принцип фотоплетизмографії, цей метод є точним та безпечним. Завдяки використанню декількох світлодіодів з різною довжиною хвилі та фото резистору ми отримуємо якісні параметри та пристрої для визначення рівня насичення крові киснем. Що у побутових, що у професійних пульсоксиметрів похибка складає ± 2 % в діапазоні рівня сатурації 85-100%. Та ± 3 % при вимірюваннях нижче 84%. Точність вимірювання частоти пульсу теж однакова та складає ±1% або ж ± 1-2 уд/хв. В цих пристроях не враховано можливості передачі даних на відстань, що може бути дуже корисним при догляді за хворим.

1.3 Методи моніторингу функцій дихання.

Респіраторний моніторинг широко використовується при хірургічних операціях для контролю за станом пацієнта і попередження гіпоксемії або гіпоксії.

Контроль за диханням може проводиться по різному, це можуть бути спірометричні датчики в апаратах ІВЛ які слідкують за вентиляцією, концентрацією газів в дихальній системі, об’ємними та динамічними

параметрами. Контроль дихання пацієнта за допомогою апарату ІВЛ зображено на рисунку 1.4.

Рисунок 1.4 – Контроль стану пацієнт за допомогою апарату ІВЛ

До методів контролю можна віднести спостереження за станом шкіри та слизових оболонок, який має на меті попередження ціанозу, але він є не дуже точним і має лише 60% ефективність.

1.3.1 Метод пульсоксиметрії

Метод використовує принципи фотоплетизмографії, які дозволяють виділити артеріальну складову поглинання світла для визначення оксигенації артеріальної крові. Таке вимірювання дозволяє використовувати спектрофотометрію для вимірювання рівня сатурації. Для отримання результатів використовується методика фотоплетизмографії в якій ділянка тканин які досліджуються розміщується між джерелом світла та світлоприймачем[3].

Рівень абсорбції світла визначається двома параметрами розміром судини або об’ємом крові який проходить через досліджувану ділянку. Скорочення і

розширення під час пульсацій викликають зміни амплітуди сигналу отримуваного на фотоприймачу. Принцип роботи пульсоксиметра показано на рисунку 1.5.

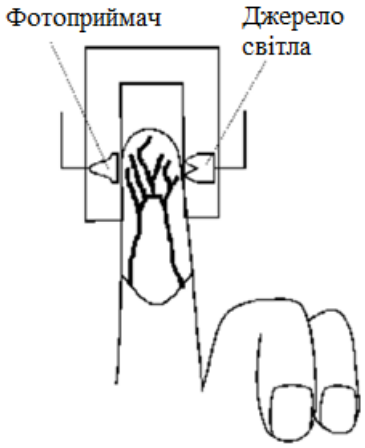

Рисунок 1.5 – Принцип роботи пульсоксиметра

Для підвищення ефективності вимірювань використовуються два джерела світла які мають різні спектральні характеристики. Оскільки при довжині хвилі в 660 нм гемоглобін поглинає приблизно в 10 разів більше світла ні ж оксигемоглобін, а на довжині хвилі в 940 нм поглинання оксигемоглобіну вище. На рисунку 1.6 показано залежність поглинання світла гемоглобіном від довжини хвилі.

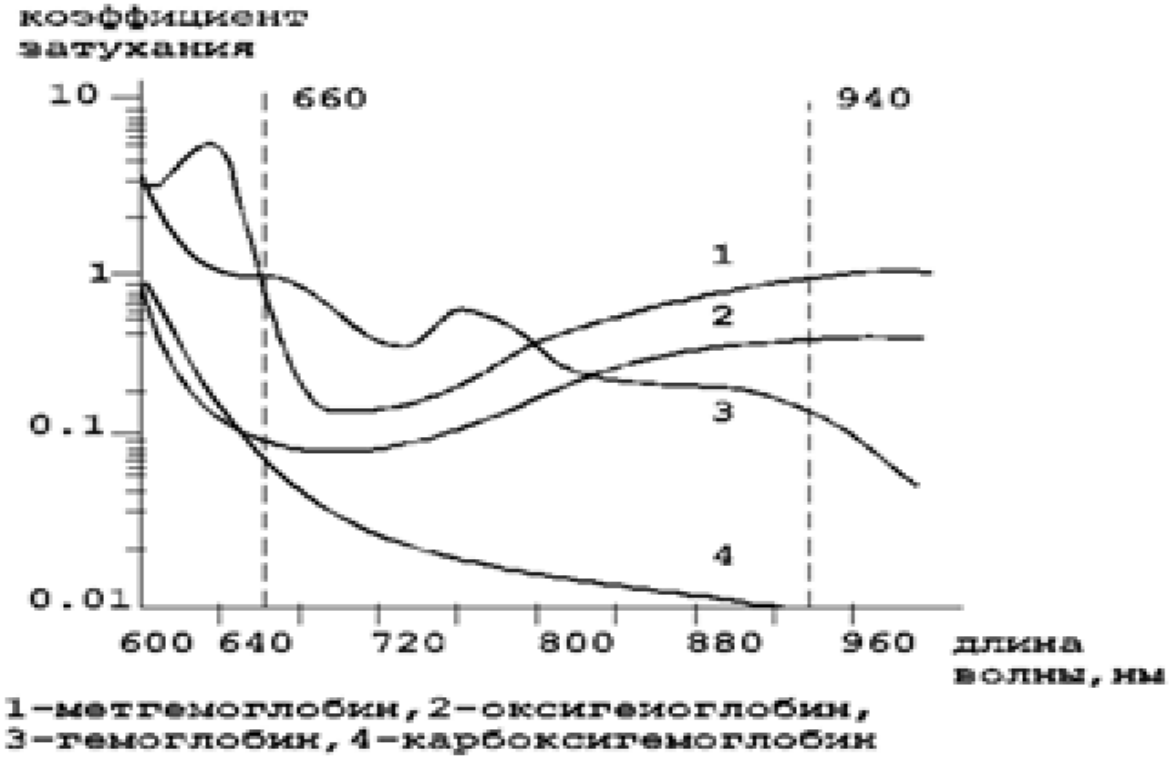

Рисунок 1.6 – Залежність поглинання світла від довжини хвилі для різних форм гемоглобіну

Зазвичай при такому методі вимірювання датчик закріпляється на кінчику пальця руки або ноги, мочці вуха. Похибка при вимірюванні сатурації в діапазоні від 80-99: складає +-2%, а для сатурації 50-79% +-3% час необхідний для отримання заміру складає не більше 10 секунд. При вимірюванні в поле зору датчика потрапляють артеріальні судини. В такому випаду сигнал від датчика ділиться на два компонента пульсуючу та постійну складові як показано на рисунку 1.7. Пульсуюча складова обумовлена змінами об’єма артеріальної крові під час серцевих скорочень тим самим в постійній складовій виділяються окремо тканини, артеріальну та венозну кров. Завдяки таким викидами можна найбільш точно визначити насиченість крові киснем.

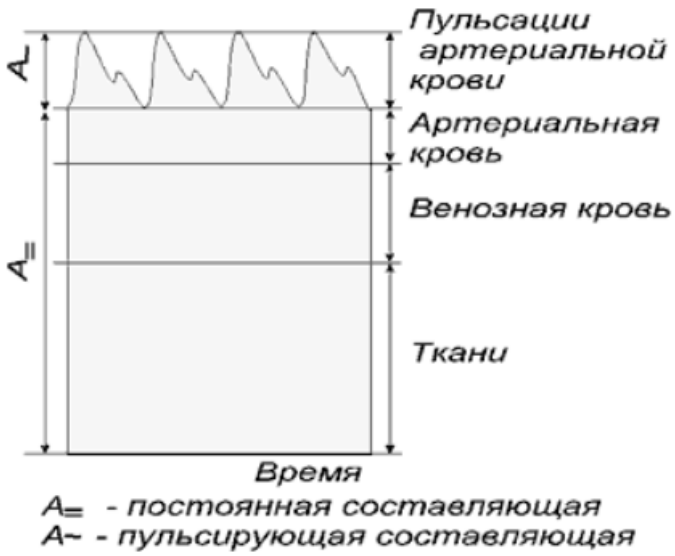

Рисунок 1.7 – Розподіл поглинання світла в тканинах

1.3.2 Метод капнометрії

Цей метод дозволяє оцінити газообмін в легенях шляхом вимірювання концентрації вуглекислого газу. При використанні цього методу відбувається безперервний запис дихання людини з вимірюванням рівня вуглекислого газу в повітрі яке вдихається і видихається. Зазвичай виміри відбуваються в стані спокою.[4] Більшість капнометрів використовується принцип заснований на поглинанні вуглекислим газом інфрачервоних променів як працює метод показано на рисунку 1.8. Недоліком таких систем є необхідність постійного фіксованого положення тіла впродовж часу заміру та висока ціна пристроїв для капнометрії.

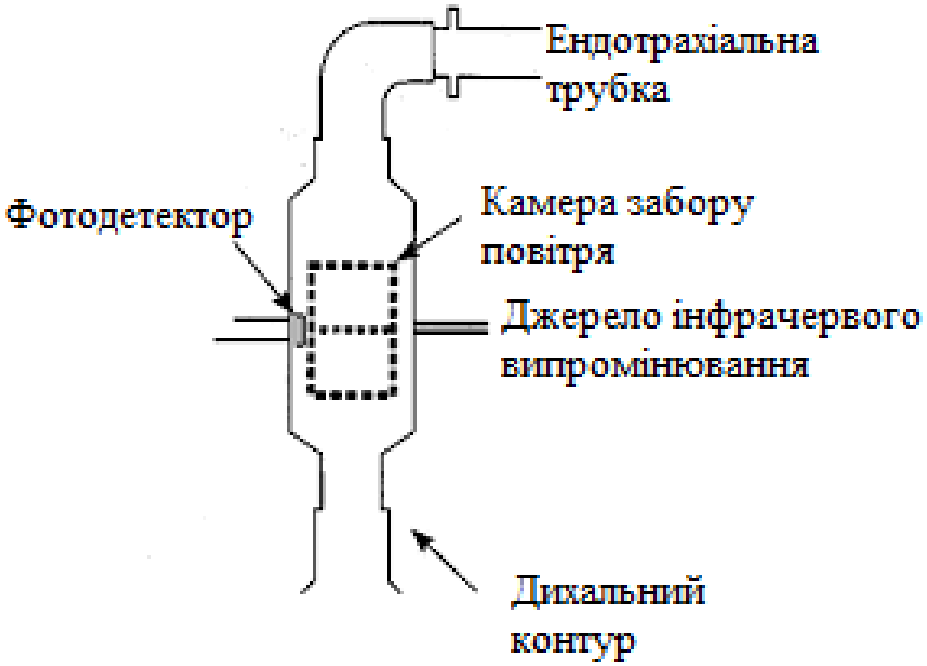

Рисунок 1.8 – Капнометрія прямого потоку

1.4 Датчики для вимірювання сатурації та серцебиття

Для вимірювання рівня сатурації та частоти серцевих скорочень, можна використовувати спеціальні смарт датчики. Які побудовані на різних принципах вимірювань найбільш оптимальним неінвазивним методом визначення сатурації, є пульсоксиметрія, цей метод дозволить без зайвого дискомфорту вимірювати кількість ударів серця за хвилину та рівень насиченості крові киснем. Тому розглянемо два датчики які побудовані на цьому принципі. Датчик сатурації та серцебиття Nellcor OximaxTM дає наступні можливості для вимірювань:

- Метод вимірювання: два світлодіоди
- Діапазон сатурації: 20-100%
- Діапазон пульсу: 20-250 (уд./хв.)
- Точність вимірювання сатурації: 90-100% ± 2%, 70-89% ± 3%
- Довжина кабелю: 3 м

Ще один датчик який розглядається це MAX30102 він має наступні параметри:

- Метод вимірювання: два світлодіоди
- Діапазон сатурації: 20-100%
- Діапазон пульсу: 20-250 (уд./хв.)
- Точність вимірювання сатурації: 90-100% ± 2%, 70-89% ± 3%
- Довжика кабелю: відсутній

Порівнявши два датчики можна зробити висновок, що їх параметри ідентичні при цьому, в датчика MAX30102 відсутній кабель для підключення, але це відкриває можливості для визначення довжини самостійно шляхом пайки кабелю необхідної довжини. Ціна цих датчиків відрізняється досить сильно, MAX30102 дешевше в 6-7 разів, що робить його більш вигідним для використання. Датчик MAX30102 зображено на рисунку 1.9.

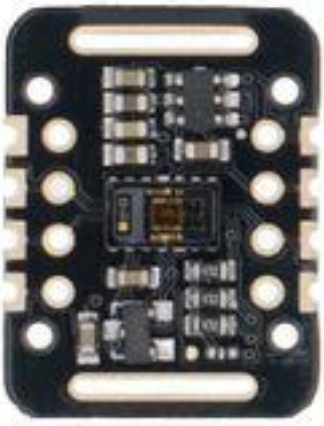

Рисунок 1.9 – Датчик сатурації та пульсу MAX30102

1.5 Методи вимірювання пульсу

Розглянемо два методи вимірювання пульсу: Оптичний(HR) та нагрудний кардіо-пульсометр(HRM).

В цих методах використовуються два абсолютно різних підходи до визначення частоти пульсу.

Оптичні датчики зазвичай використовуються в різноманітних фітнес браслетах та пульсоксиметрах, для контролю під час тренувань, вони дають змогу оцінити частоту пульсу, вони використовуються принципи які використовуються і для визначення рівня сатурації, шляхом проходження світла через та поглинання його гемоглобіном, за рахунок чого визначається

сатурація, світловий сигнал проходячи через тканини людини, набуває пульсуючий характер, що впливає на об’єм артеріального русла таким чином ми і вимірюємо частоту пульсу. На рисунку 1.10 показано процес вимірювання пульсу за допомогою пульсоксиметра.

Рисунок 1.10– Вимірювання частоти пульсу за допомогою пульсоксиметра

HRM в своїй основі використовує систему з двох датчиків наприклад нагрудний ремінь з електродом який зчитує електричні імпульси та система яка отримує ці данні наприклад розумний годинник, цей метод має більшу високу точність і його можна порівняти з процесом ЕКГ[5]. Комплект такого пристрою можна побачити на рисунку 1.11.

Рисунок 1.11 – комплектація кардіо-пульсометра

1.6 Одно платні системи на базі мікроконтролерів

До важливих питань при розробці та досліджені комп'ютерних систем можна віднести вибір мікроконтролера для розробки, саме від нього залежить більшість параметрів та можливостей майбутньої системи.

В даній роботі буде розглянуто дві одно платних платформи на базі мікроконтролерів.

1.6.1 Платформа Arduino Uno

Плата мікроконтролера на базі процесору atmega 328.

Має 14 цифрових входів/виходів 6 з яких можуть працювати в режимі широтно імпульсної модуляції(ШІМ), 6 аналогових входів для підключення аналогових датчиків. Програмування плати відбувається за допомогою USB порту та середовища розробки Arduino IDE крім цього можна використати SPI або UART інтерфейс для завантаження програми. Плата сумісна з величезною кількістю датчик, має широку спільноту, а відповідно і велику кількість бібліотек для спрощення роботи з датчиками. Тому плата є чудовим варіантом для розробки різних простих прототипів пристроїв, але не дуже підходить для розробки складних пристроїв через досить обмежену пам'ять.

Характеристики:

Робоча напруга: 5В

Рекомендована вхідна напруга: 7-11 В

Струм на кожен вхід/вихід: 40 мА

Флеш пам'ять: 32 КБ (2 з них зайняті завантажувачем).

SRAM: 2 КБ

EEPROM: 1 КБ

Тактова частота: 16 МГц

Розміри: 69 х 54 мм

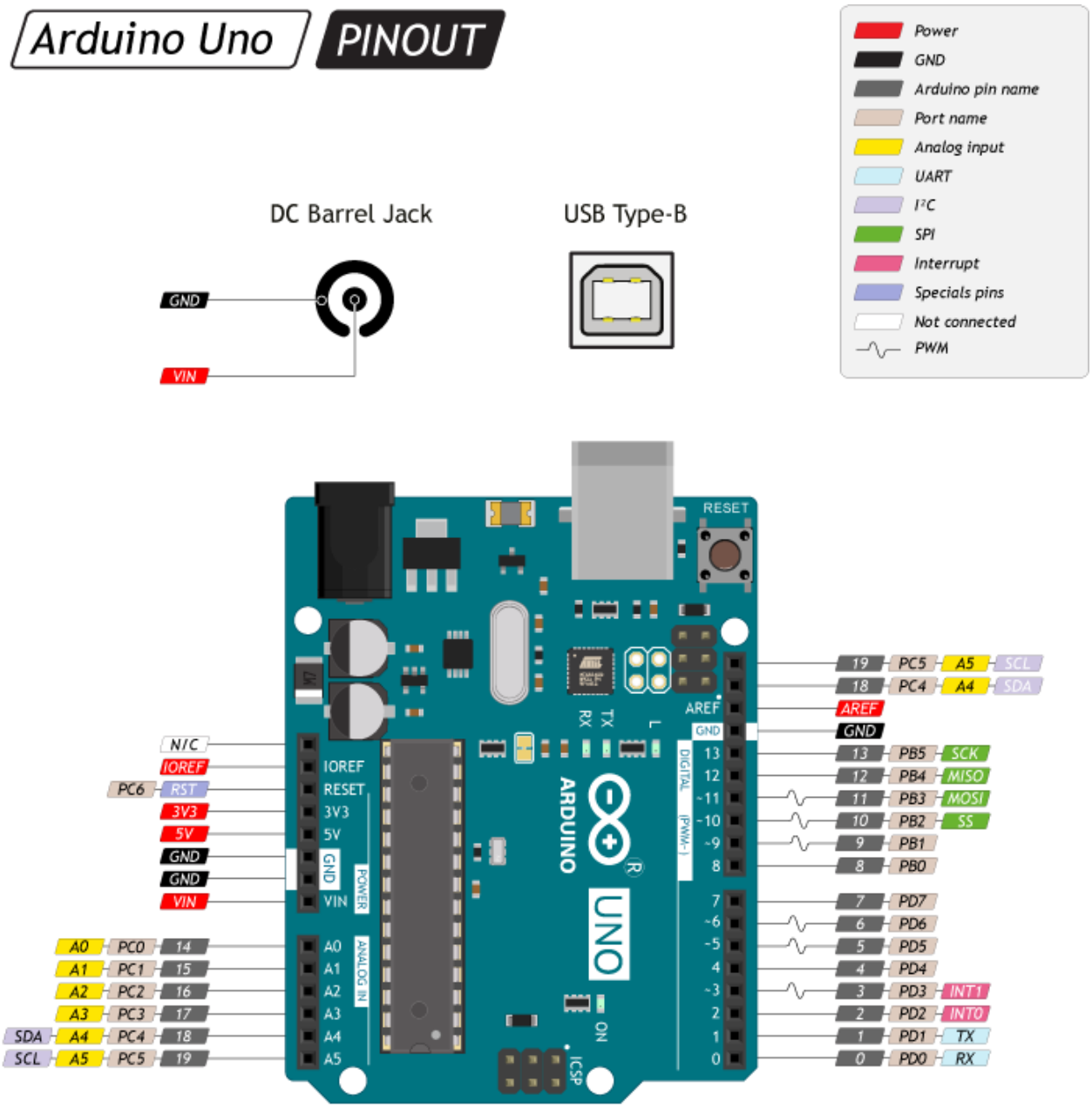

Рисунок 1.12 - Призначення виводів Arduino Uno

1.6.2 Плата ESP32 WROOM DEVKIT V1

Плата мікроконтролера на базі процесору Tensilica Xtensa LX6. Має 16 портів для ШІМ, вбудовані WI-FI та Bluetooth. Програмується за допомогою порту micro usb та середовища розробки Arduino IDE при цьому потребує лише встановлення додаткового програмного забезпечення. Всього має 30 повноцінних портів вводу/виводу. Має хороші можливості розробки за

рахунок великої кількості портів, інтерфейсів підключення, вбудовані засоби дистанційного зв’язку, деякі бібліотеки є сумісними з Arduino, тобто можна використовувати бібліотеки написані для Arduino, що є ще однією перевагою. При цьому розміри цієї плати є незначними, це дозволяє створювати компактні пристрої з дистанційним керуванням та передачею даних[7].

Характеристики:

Напруга живлення: 5 В

Максимальна струм стабілізатора напруги: 800 мА

Wi-Fi стандарти: FCC/CE/IC/TELEC/KCC/SRRC/NCC

Протоколи: 802.11 b/g/n/d/e/i /k/r (802.11n до 150 Мбіт/с)

Частотний діапазон: 2.4-2.5 ГГц

Bluetooth протоколи: Bluetooth v4.2 BR/EDR і BLE specification

Аудіо: CVSD и SBC

Апаратні засоби та інтерфейси: SD, UART, SPI, SDIO, I²C, LED PWM, Motor PWM, I²S, I²C, IR GPIO, сенсорний датчик, ADC, DAC, LNA

Вбудовані датчики: датчик Холла, температурний датчик

Генератори: кварцові 26 МГц і 32 кГц

Живлення мікромодуля: 2.2-3.6 В

Середній робочий струм: 80 мА

Піковий робочий струм: 500 мА

Діапазон робочих температур: -40°C ~ 85°C

Flash-память: 448 КБ

Зовнішня Flash-память: 4 МБ

SRAM-память: 520 КБ

Розміри: 51×28 мм

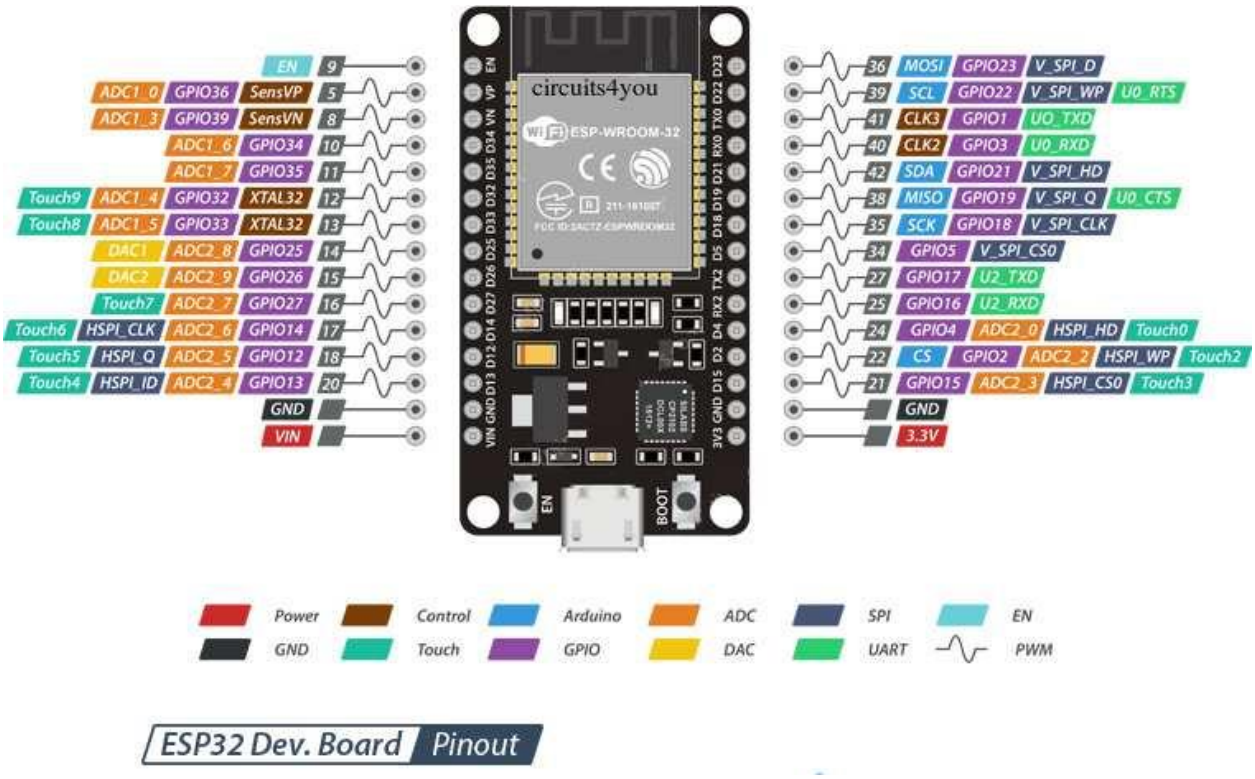

Рисунок 1.13 - Призначення виводів esp32

Розглянувши дві цих платформи можна виділити їх позитивні та негативні сторони.

Arduino Uno

Переваги:

- Можливість напряму подавати за допомогою цифрових портів живлення на датчики оскільки робоча напруга 5 В.
- Достатня кількість портів вводу/виводу для підключення необхідних датчиків.
- Низька ціна
- Високий рівень сумісності з датчиками
- Високий струм виходу у цифрових портів
- Наявність 10 бітного АЦП
- Порти з АЦП 6
- Інтерфейси I²C,SPI та UART

Недоліки:

- Високе енергоспоживання 24 мА в активному режимі та 5 мА в режимі сну
- Відсутність вбудованих засобів бездротового звязку
- Розрядність ШІМ всього 8 біт
- Кількість портів з можливістю переривань 2

Плата esp32

Переваги:

- Низький рівень енергоспоживання близько 20 мА та 10 мкА в режимі сну
- 12 бітний АЦП забезпечує більшу точність в порівнянні з Arduino
- по два інтерфейси I²C та I²S, по три SPI та UART
- Порти з перериваннями 25
- 15 портів з АЦП
- Великий розмір пам’яті
- Невеликі габарити
- Вбудовані Wi-Fi та Bluetooth
- Можливість роботи як веб-сервер, точка доступу.

Недоліки:

- Логічний рівень 3.3В потребує обережності при роботі з модулями які базуються на напрузі в 5 В
- Потреба у додатковому джерелі живленні для більшості датчиків

1.7 Проблема пульсоксиметрів.

Найчастіше проблемою медичних пульсоксиметрів які використовуються в каретах швидкої допомоги для швидкої діагностики є їх ціна та неможливість передавати дані дистанційно, зазвичай вони виводяться на вбудований дисплей або ж через usb порт на персональний комп’ютер таким чином унеможливлює використання їх як засіб дистанційного контролю

за станом хворого, як наприклад догляд хворого вдома при цьому, не потрібно постійно сидіти поруч з хворим ризикуючи захворіти, або ж дозволяти хворому самому вільно рухатись при цьому контролюючи стан свого організму.

Побутові ж пульсоксиметри хоч і мають невеликі розміри та низьку ціну проте не дають змоги постійно контролювати стан хворого через свою не автономність.

1.8 Засоби дистанційного звязку

Для дистанційного зв’язку можна використовувати різні технології основними для зв’язку між телефоном та платформою з датчиками є Bluetooth та Wi-Fi ці технології є простими та надійними, але між ними є деякі відмінності, перша з них стосується енергоспоживання, Bluetooth потребує в 3-4 рази менше живлення при передачі даних в той час як Wi-Fi є більш енерговитратним. Тому за цим параметром виграє Bluetooth, але в нього є суттєвий недолік який стосується радіусу дії, а саме до 15м до того ж на максимальній відстані дані можуть зникати або ж спотворюватись. В цей же час Wi-Fi має радіус дії до 100м і цієї відстані цілком достатньо для дистанційного контролю за станом хворого. Таким чином Wi-Fi хоч і витрачає більше енергії, але дозволяє слідкувати за хворим на великій відстані. Тому технологія Wi-Fi є оптимальним рішенням, схема зв’язку пристроїв зображена на рисунку 1.14.

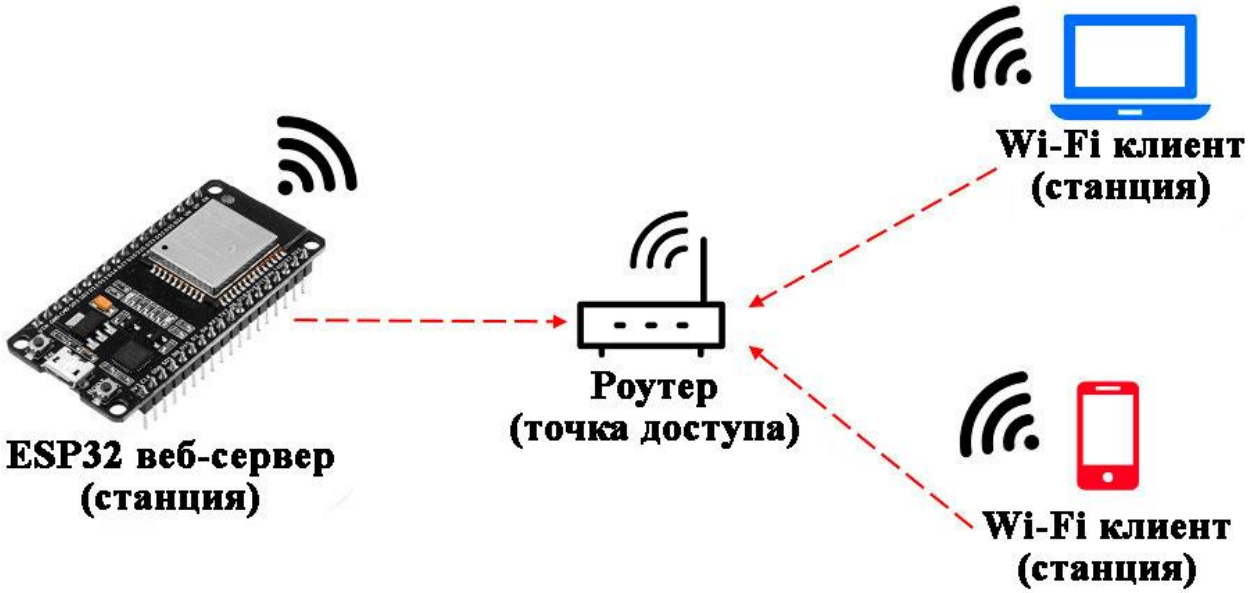

Рисунок 1.14 - Зв'язок пристроїв за допомогою технології Wi-Fi

1.9 Способи живлення

Важливим моментом для забезпечення автономності будь-якої системи є елементи живлення тому, важливо підбирати елементи які можуть забезпечити потреби пристрою.

1.9.1 Універсальні мобільні батареї

Зазвичай мають досить високу ємність акумулятора, розглянемо одну з універсальних батарей, а саме «Xiaomi Mi Power Bank 3» зображено на рисунку 1.15

Ця батарея має наступні характеристики:

- Ємність акумулятора: 10000 мАг
- Максимальний струм: 3 А
- Параметри входу: 5В = 2А
- Параметри виходу: 5В = 2.4 А
- Розміри: 90 х 63.9 х 24.4 мм
- Вага: 200г

Проаналізувавши характеристики можна зробити наступні висновки, такі батареї мають високу ємність акумулятора, але при цьому досить не зручні розміри для живлення портативних пристроїв. Напруга виходу дозволяє підключати та живити різні енерговитратні датчики, при цьому при підключені мікроконтролерів у яких напруга живлення 3.3 В виникає потреба знижувати напругу з 5 В до 3.3 В для цього в схему потрібно додавати понижуючий стабілізатор.

Рисунок 1.15 – Універсальна мобільна батарея

1.9.2 AA Ni-MH(нікель металогідридний) акумулятори

Мають чудові характеристики для живлення різних мікроконтролерів номінальна напруга одного такого акумулятора має 1.2 В це означає, що 3 таких батареї чудово підходять для живлення ESP32, а 4 батареї будуть чудовий варіантом для живлення плат Arduino. При цьому ємність таких акумуляторів сягає 2000 мАг довжина такого елемента сягає 50 мм, а діаметр 13.5 – 14.5 мм. Розміри одного такого елементу досить не великі, але оскільки для живлення мікроконтролеру таких елементів потрібно декілька можна зробити висновок, що габарити будуть близькі до універсальної мобільної батареї. Тому як варіант живлення такий варіант є менш вигідним оскільки має в 5 разів меншу ємність в порівнянні з УМБ і при цьому ті ж самі розміри, єдиним параметром за яким виграють дані батареї є вага оскільки чотири таких батареї будуть мати вагу від 56 до 120 грам, а вага стандартних універсальних

мобільних батарей коливається в районі 200 грам. Також до переваг цього типу батареї можна віднести те, що їх початкова ємність майже незмінна від початку та впродовж всього часу експлуатації. На рисунку 1.16 показано чотири батареї типу Ni-MH.

Рисунок 1.16 – акумуляторні батареї типу Ni-MH

1.9.3 Батареї типу АА лужні

Номінальна напруга такої батареї сягає 1.5 В , тому 2 таких батареї можуть живити ESP32, а 3 батареї можуть живити плату Arduino. Вага та розміри такої батареї ідентичні розмірам Ni-MH акумулятора ємність коливається від 1000 до 2800 мАг при цьому такий тип батареї має значний недолік неможливість перезарядки. Тому такий тип батареї не підходить для автономних пристроїв які повинні працювати в постійному режимі.

1.9.4 Літієві акумулятори

Вирізняються високою реальною ємність, часто використовуються для досить енергоємних пристроїв таких як ехолот або металошукач ємність одного такого акумулятора коливається в районі від 2000 до 4000 мАг в залежності від виробника. Напруга від 3.6 В до 4.2 В, що дозволяє живити як

ESP32 так и Arduino впродовж тривалого часу. При цьому літієві акумулятори дозволяють використовувати їх навіть для таких енерговитратних завдань як передача даних через Wi-Fi та мають чудове енергозбереження тому є найкращим варіантом для живлення в тому числі і для забезпечення автономності портативних пристроїв[6]. При цьому вага одного такого акумулятора коливається в районі 45 грам тому при паралельному зєднані двох таких акумуляторів вже можна отримати ємність від 4000 до 8000 мАг при цьому вага та габарити цієї батареї є досить високою. Габарити літієвих акумуляторів в порівнянні з AA та AAA акумуляторами зображено на рисунку 1.17

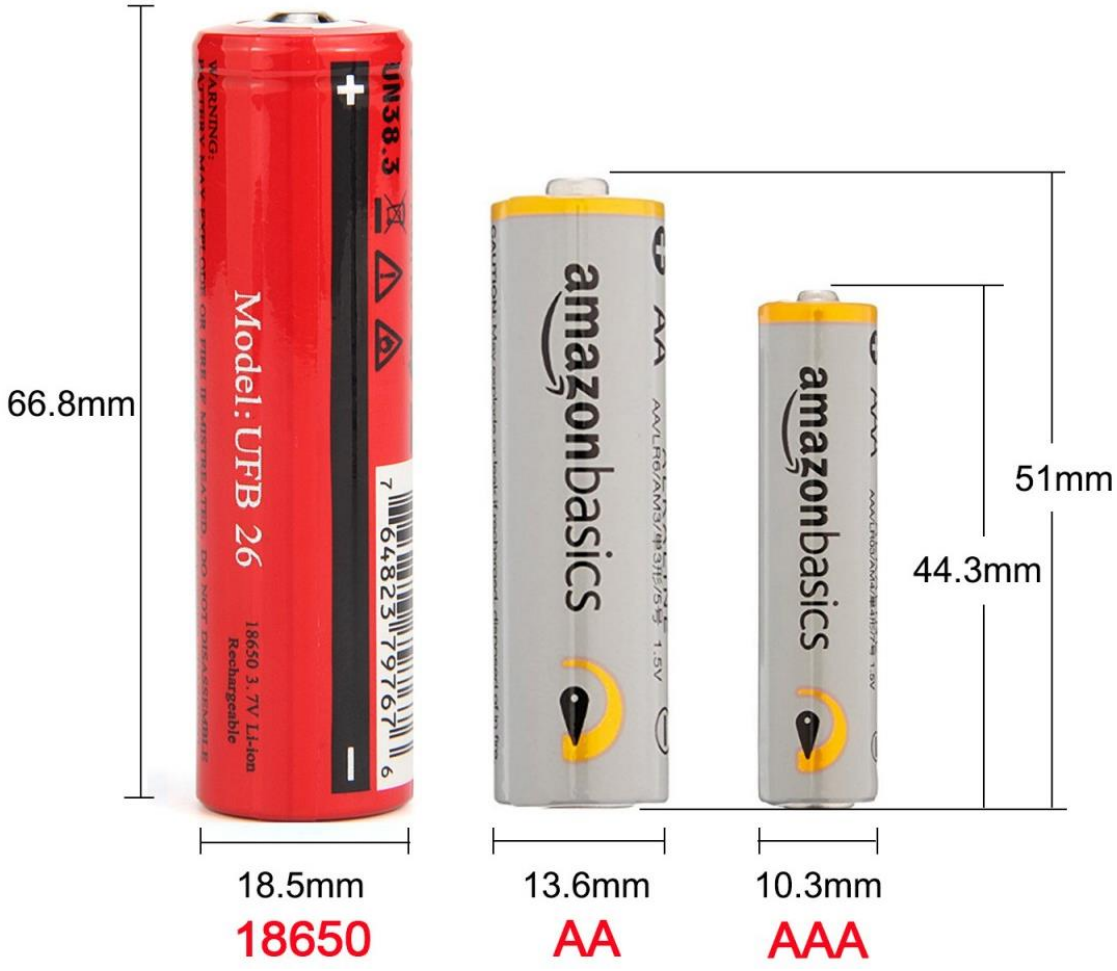

Рисунок 1.17 – Габарити акумуляторів типу 18650, AA та AAA

Таблиця 1.1 – Типи батарей та їх ємність

Тип батареї	Ємність	Розміри	Вага	Напруга
Літієві 18650	2000-3800 мАг	18.3х69 мм	48 г	3.6 В
УМБ	5000-20000 мАг	90х63 мм	200-390 г	5 В
Лужні батареї	1500-2500 мАг	14.5х55 мм	25 г	1.5 В
Ni-MH	1000-2500 мАг	14.5х55 мм	56 – 120 г	1.5 В

З табл 1.17 видно, що УМБ мають найбільшу ємність, за що вони платять своїми розмірами та вагою, крім цього їх напруга не підходить для живлення мікроконтролерів з 3.3 В логікою, лужні та нікель-металогідридні мають однакову ємність та розміри, хоч лужні і виграють за рахунок меншої ваги, їх напруга є однаковою, для роботи пристрою потрібно дві послідовно з’єднаних батареї, що не є вигідним. Літієві батареї мають високу ємність, зручну напругу живлення та досить невеликі розміри та вагу.

1.10 Висновки до розділу

Проаналізовано методи та способи моніторингу стану людини. Виявлено основні проблеми сучасних пристроїв та методів моніторингу. Визначено основні методи контролю насиченості крові киснем, та визначення частоти серцевих скорочень людини. Розглянуто технології та одно платні платформи які забезпечують можливості для розробки та вдосконалення системи діагностики. В сучасних системах є необхідність покращення їхньої автономності для постійного моніторингу стану хворого, а також можливість контролювати стан та показання датчиків дистанційно. Крім цього більшість систем подібного типу мають систему сповіщення про несправність яка найчастіше складається тільки зі звукового сигналу, або ж виведення даних на екран. Для вдосконалення системи сповіщень окрім звукового сигналу, потрібно забезпечити можливість передавати попередження на пристрій або ж пристрої дистанційного моніторингу.

2 ВДОСКОНАЛЕННЯ СИСТЕМИ ДІАГНОСТИКИ

Проаналізувавши існуючі аналоги та потреби сьогодення можна виділити наступні завдання які потрібно вирішити при розробці системи, а саме автономність, довготривалий моніторинг, дистанційний контроль стану, компактність. Вирішення цих завдань потребує, визначення компонентів системи, методів вимірювання, способів передачі даних.

2.1 Компоненти системи

Для системи яка розробляється важливим є вибір компонентів, а саме мікроконтролеру який буде контролювати та обробляти дані датчиків, і передавати інформацію за допомогою однієї з технологій Wi-Fi чи Bluetooth. Оскільки ESP32 вже має в собі Wi-Fi та Bluetooth цей контролер є оптимальний і дозволяє зменшити розміри отриманого пристрою завдяки чому збільшити його компактність та зручність у користуванні. Тому вибір ESP32 в якості мікроконтролера для системи є найкращим, Bluetooth та Wi-Fi мають свої переваги та недоліки, розроблюваний пристрій повинен забезпечувати безперебійний зв'язок на досить великі відстані і саме ці потреби може забезпечити Wi-Fi хоч він і має досить високе енергоспоживання. Отже обравши мікроконтролер та спосіб передачі даних потрібно визначити який з датчиків для визначення сатурації є підходящим для даної системи. Розібравши характеристики датчиків сатурації можна зробити висновок, що

хорошим вибором для системи буде датчик MAX30102 який має невеликі розміри, низьку ціну і точність яка нічим не відрізняється від професійних датчиків. Для живлення оптимальним варіантом буде одна або дві літієвих батареї типу 18650 яка забезпечать високу ємність при невеликих розмірах. Прототип системи можна побачити на рисунку 2.1.

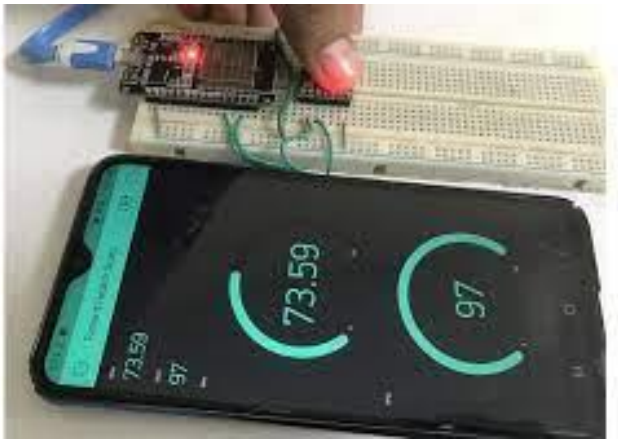

Рисунок 2.1 – Прототип розроблюваної системи

2.2 Автономність системи

Забезпечення автономності системи потребує аналізу наступних параметрів споживання струму, ємність елементів живлення, швидкість виконання енерговитратних завдань, перехід в режим очікування. Режими роботи показано на рисунку 2.2.

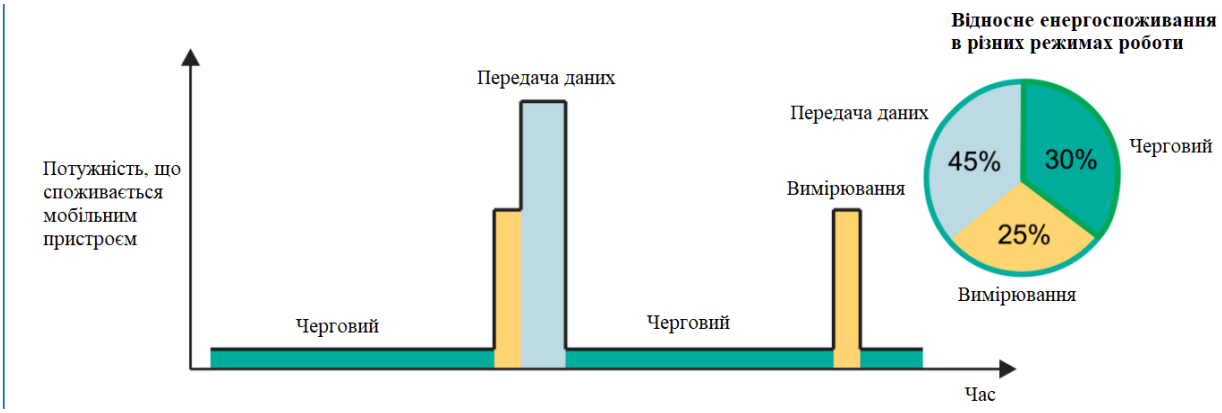

Рисунок 2.2 - Енергоспоживання пристрою в різних режимах роботи

Таблиця 2.1 – Режими роботи ESP32

Режими роботи ESP32	Споживання струму
Глибокий сон	10 мкА
Легкий сон	0.8 мА
Нормальний режим (240 МГц)	50 мА
Модем в режимі сну	30-68 мА

З рисунку 2.2 можна побачити, що найбільше енергії витрачається під час передачі даних та вимірювань. Тому потрібно зробити так щоб процес передачі даних займав якомога менше часу. Для цього потрібно розібрати режими роботи мікроконтролера при яких, вимикаються зайві периферійні пристрої і вмикаються лише, датчики які необхідні для вимірювань та Wi-Fi для передачі даних з подальшим переходом в сплячий режим модулю Wi-Fi який є основним споживачем енергії в режимі активної роботи.

Нормальний режим – всі периферійні пристрої увімкненні, мікроконтролер працює в штатному режимі, може отримувати, передавати та слухати.

Режим сну модему – мікроконтролер знаходиться в робочу стані таймери можна налаштовувати, основна смуга Wi-Fi, Bluetooth та радіо вимкненні.

Режим легкий сон – мікроконтролер призупинений. Вмикається процесор ультра низького споживання. Пам'ять годинника реального часу та його периферія увімкненні, будь-які події можуть вивести мікроконтролер з режиму сну, такі як зовнішні переривання, ведучий пристрій, таймер реального часу.

Режим глибокого сну – працює лише годинник реального часу та його периферія. Дані для з’єднання Wi-Fi та Bluetooth зберігаються в пам’яті годинника реального часу. Процесор ультра низького споживання функціонує.[7]

При живлені від акумуляторної батареї необхідно враховувати напругу живлення мікроконтролера так наприклад, при використанні ESP32 потрібна напруга 3.0 В – 3.6 В , при виході з цього діапазону можливе пошкодження мікроконтролеру або його нестабільна робота. Для того, щоб обрати

правильний акумулятор потрібно розібратись з кількістю струму який потребується для роботи датчиків та модулю Wi-Fi для подальшої стабільної автономної роботи пристрою тому визначимо скільки потребує датчик пульсу та сатурації на базі MAX30102, в режимі сну датчик споживає 1.2 мкА, а в режимі вимірювань цей показник сягає 1.2 мА, але при цьому також потрібно враховувати енергоспоживання двох світлодіодів вбудованих в датчик яке буде коливатись від 0 до 50 мА та частоту вимірювань, якщо брати до уваги скільки потребує Wi-Fi для передачі даних то цей датчик має досить низьке споживання енергії не враховуючи світлодіоди. Тому для підвищення автономності роботи пристрою потрібно звернути увагу на зменшення часу роботи Wi-Fi та підвищення його ефективності за потреби, адже його енергоспоживання коливається в діапазоні від 80 до 240 мА, що є дуже високим показником для автономного пристрою. Для зменшення енергоспоживання Wi-Fi можна використати наступні методи, а саме режим сну який дозволяє зменшити споживання всіх модулів плати шляхом їх відключення, при цьому згідно документації мінімальний час необхідний на увімкнення та перезавантаження мікроконтролеру складає 50 мкС і зображено на рисунку 2.3 та табл 2.24[7]

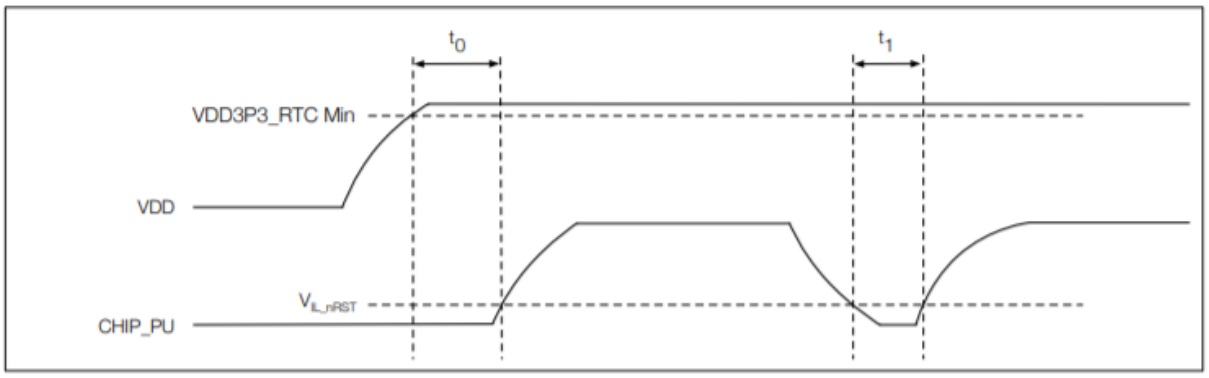

Рисунок 2.3 - ESP32 час увімкнення та перезавантаження

Таблиця 2.2 – Час увімкнення та перезавантаження мікроконтролеру

Параметри	Описання	Мінімальне	Одиниці
t_0	Час між підключенням 3.3 В і увімкненням мікроконтролеру	50	мкс
t_1	Час між отриманням сигналу перезавантаження та перезавантаженням	50	мкс

Таким чином можна зробити висновок, що час необхідний на увімкнення мікроконтролеру є досить низьким і одним із способів зменшення енергоспоживання можна використати, додатковий зовнішній таймер або переривання які будуть вмикати мікроконтролер на час вимірювань і після відправки даних вимикати, робити це все потрібно циклічно для отримання достовірних даних. Цей спосіб може збільшувати похибку вимірювань тому він не є підходящим. Саме тому потрібно використовувати спеціальні режими мікроконтролеру які зображено в таблиці . серед підходящих режимів для роботи пристрою є режим сну модему, коли мікроконтролер проводить вимірювання в штатному режимі при цьому Wi-Fi є вимкненим, що дозволяє не витрачати енергію на передачу та отримання даних, після вимірювань з певною тривалістю можна визначати середні значення та відправляти їх на пристрій який з’єднано дистанційно через Wi-Fi після чого знову переходити в режим сну модему[8].

Щоб реалізувати зменшення енергоспоживання потрібно переводити ESP32 в режим роботи модему, тобто вимикати Wi-Fi модуль зі збереженням поточного підключення. Реалізація переходу в режим модему і назад відбувається наступним чином, для цього потрібно перевести частоту мікроконтролера на 40 МГц, а для переходу до звичайного режиму роботи частота мікроконтролера повинна дорівнювати 240 МГц. Створимо два методи які будуть реалізувати необхідні задачі.

```
void setModemSleep();
void wakeModemSleep();
```

роботи в 40 МГц, другий метод вмикає модуль та встановлює частоту 240 МГц. Таким чином відбувається зменшення витрат енергії на постійну підтримку Wi-Fi в активному стані весь час, це поліпшує автономність пристрою і дозволяє працювати в декілька разів довше.

Необхідно визначити з яким інтервалом потрібно передавати отримані значення через Wi-Fi для цього розглянем декілька варіантів та знайдем оптимальний який дозволить отримати прийнятну автономність та при цьому не вплине на якість отриманих даних.

Якщо розглядати пульсоксиметр то в середньому їх час вимірювань коливається в діапазоні від 10 до 20 секунд. Тому розглянемо п’ять варіантів на 5,10,15, 20 та 30 секунд відповідно. В таблиці . представлено енергоспоживання кожного з окремих датчиків та мікроконтролеру.

Таблиця 2.3 – Енергоспоживання модулів системи

Пристрій	Енергоспоживання
Мікроконтролер (в режимі модему)	30-68 мА
Датчик пульсу MAX30102	1.2 мА
Wi-Fi	80-240 мА
Світлодіод датчика пульсу	0-50 мА

Датчик пульсу має особливість у вигляді двох світлодіодів які вмикаються з певною частотою (200 мкс - 1.6 мс) для проведення вимірювань і мають енергоспоживання від 0 до 50 мА тому потрібно врахувати і цей показний. Показники світлодіодів для розрахунків візьмемо середні тобто 25 мА з частотою 800 вимірів за секунду та довжиною імпульсів в 118 мкс, інші параметри візьмемо максимальні. Також потрібно врахувати, що модуль Wi-Fi працює не постійно а лише передає данні після завершення вимірюваннь. Таким чином спробуємо розрахувати скільки енергії буде споживати датчик MAX30102 під час постійної роботи.

Для цього потрібно кількість вимірів помножити на їх довжину та їх струм споживання. В результаті отримаємо рівняння:

$$I_{споживання} = T_{вимірювання} * t_{імпульса} * I_{пристрою} \quad (2.1)$$

де $I_{споживання}$ та $I_{пристрою}$ – це струм споживання датчика пульсу та струм який витрачається світлодіодами відповідно.

З рівняння 2.1 виходить, що датчик використовує струм силою 4.72 мА. За годину роботи пристрою мікроконтролер буде потребувати 68 мА, а датчик пульсу 1.2 мА без врахування світлодіодів, таким чином залишається розрахувати скільки саме буде потребувати струму Wi-Fi для передачі даних. Для цього потрібно розрахувати скільки всього часу Wi-Fi перебуває в активному режимі, якщо час передачі даних дорівнює 1 секунді[9].
Скористаємося формулою (2.1):

Таблиця 2.4 - Споживання енергії Wi-Fi модулем

Час передачі	1 с	1 с	1 с	1 с	1 с
Період	5 с	10 с	15 с	20 с	30 с
Споживання	240 мА	240 мА	240 мА	240 мА	240 мА
Споживання за годину	48 мА	24 мА	16 мА	12 мА	8 мА

Знаючи скільки потребує кожен з модулів пристрою можна розрахувати загальний час роботи.

Для цього використаємо формулу:

$$t = C_{ак} / I_{н} \quad (2.2)$$

де,

t – час роботи акумуляторної батареї в годинах

$C_{ак}$ – ємність акумулятора в мАг

$I_{н}$ – струм навантаження (мА)

Таблиця 2.5 - Загальний час роботи пристрою

Струм живлення	Акумулятор	Період передачі	Час роботи
121.92 мА	3000 мАг	5 с	24 год
97.92 мА	3000 мАг	10 с	30 год
89.92 мА	3000 мАг	15 с	33 год
85.92 мА	3000 мАг	20 с	34 год
81.92 мА	3000 мАг	30 с	36 год
121.92 мА	6000 мАг	5 с	49 год
97.92 мА	6000 мАг	10 с	61 год
89.92 мА	6000 мАг	15 с	66 год
85.92 мА	6000 мАг	20 с	69 год
81.92 мА	6000 мАг	30 с	73 год

Залежність часу роботи від струму живлення показано на рисунку 2.4

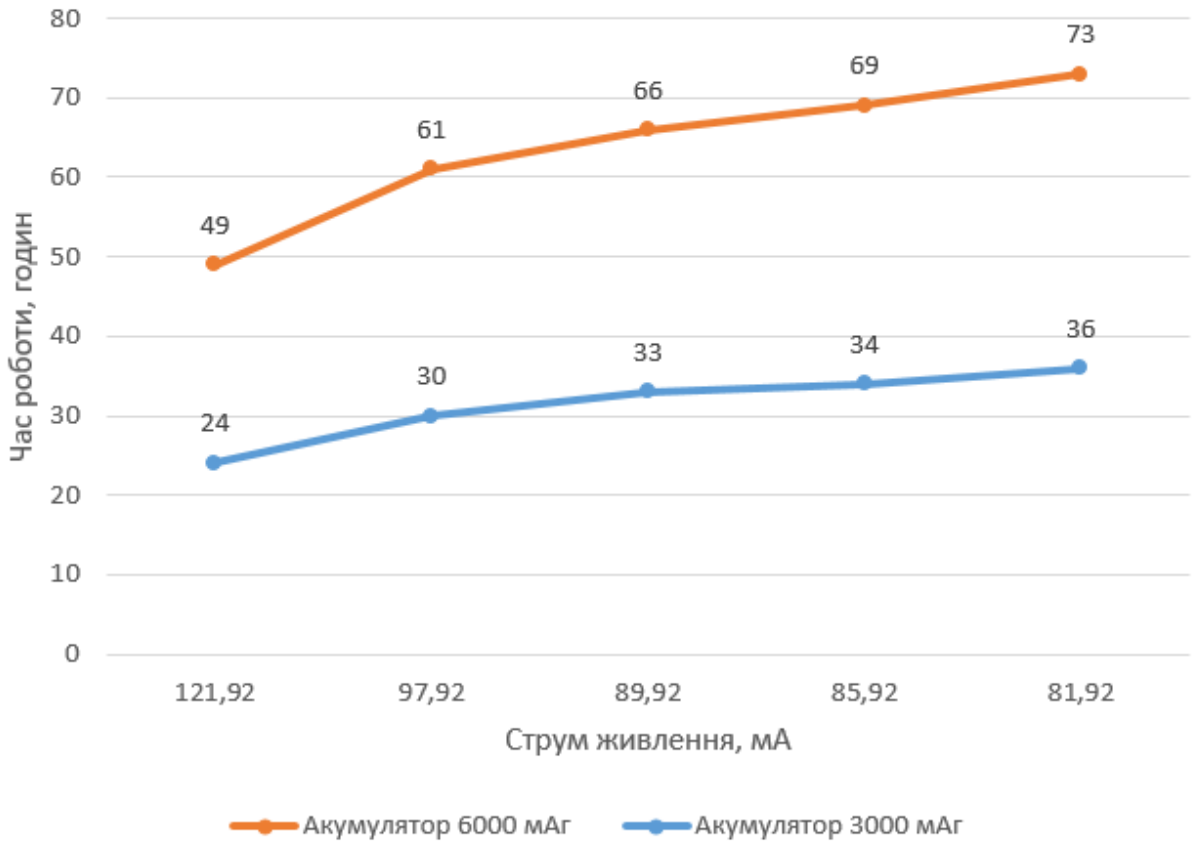

Рисунок 2.4 – Графік залежності часу роботи від струму живлення

З табл 2.5 можна побачити, що при ємності акумулятора і періоді передачі в 5 секунд час роботи від акумулятора ємністю 6000 мАг складає приблизно 49 годин, що є досить хорошим результатом для такого пристрою крім цього помітно, що період передачі в 5 та 30 секунд має різницю в часі роботи на цілих 24 години, тобто якщо потрібно збільшити автономність пристрою, то період між передачею отриманих від датчиків даних повинен бути якомога більшим, при цьому це матиме негативний вплив в якості передачі даних, що стосується пульсу оскільки при швидкій змінні пульсу дізнатись про це можна буде з затримкою в результаті періодичності передачі даних на пристрій моніторингу. Оптимальним варіантом між передачею даних та часом роботи пристрою можна вважати час в 10 секунд, це дозволить правильно реагувати на зміни в стані здоров'я людини, та забезпечить час роботи в 61 годину при ємності акумулятора в 6000 мАг та 30 годин при ємності в 3000 мАг, для пристроїв такого типу краще забезпечити час роботи в 61 годину оскільки таким чином можна впродовж більш тривалого часу контролювати стан людини без необхідності підзарядки.

Довготривалий моніторинг завдяки високій автономності дозволяє, дивитись за динамікою параметрів і на основі даних робити висновки, що до потреби консультації з лікарем. Якщо порівняти автономність отриманого пристрою то можна сказати, що вона є високою в порівнянні з аналогами, звичайні пальцеві пульсоксиметри працюють лише 30 годин після чого потребують повну заміну батареї тому, їх використання для постійного моніторингу не є доречним. Медичні пульсоксиметри мають більш високі показники автономності, що досягають 50 годин, а деякі до 72 годин.

2.3 Дистанційний контроль стану

Дистанційний контроль стану використовує технологію Wi-Fi для зв'язку між пристроями тому, важливо правильно пов'язати між собою пристрої для коректної передачі даних та стабільної роботи системи. Оскільки

ESP32 може працювати як веб-сервер одним з варіантів є створення веб-сайту який буде знаходиться на стороні ESP32 з подальшим підключенням через телефон або ж персональний комп’ютер, але даний спосіб має суттєвий недолік, а саме збільшення навантаження на Wi-Fi чим власне збільшується енергоспоживання, а оскільки використовується режим сну модему це може привести до нестабільності роботи системи. Крім цього буде потреба у виведені адреси веб-серверу на екран, який відсутній в системі, і теж збільшує навантаження на акумулятор системи, тому даний спосіб дистанційного контролю стану не є оптимальним.

Для дистанційного контролю стану можна скористатись спеціальною технологією Blynk. Це набір програмного забезпечення та інструментів для створення програми дистанційного керування пристроєм. Програми та додатки створені за допомогою Blynk від разу готові до застосування, технологія має можливості налаштування користувачів які мають доступ до пристрою. Платформа дозволяє налаштовувати вигляд зображуваних даних, доступна для таких популярних операційних систем як Android та iOS тобто, для більшості мобільних пристроїв, а також має веб-версію тобто дає можливість проводити моніторинг стану і через персональний комп’ютер. Зовнішній вигляд створюваного додатку можна налаштовувати за допомогою веб-сайту або ж мобільного додатку. Технологія автоматично генерує інструменти, кнопки, мітки, текст. Кожен з цих елементів має свій унікальний номер, що дозволяє за допомогою середовища розробки Arduino IDE та бібліотеки Blynk зв’язати між собою мобільний додаток та пристрій на основі ESP32, все, що для цього потрібно це згенерувати спеціальний ключ, який є унікальним для кожного додатку та вказати його в скетчі (програмі) після чого автоматично відбувається підключення. Також технологія дозволяє налаштувати вхідні та вихідні дані, керувати портами мікроконтролера, вказувати діапазони цифрових даних для відображення інформації від датчиків, а також одиниці вимірювання[10]. Вікно розробки додатку зображено на рисунку 2.5

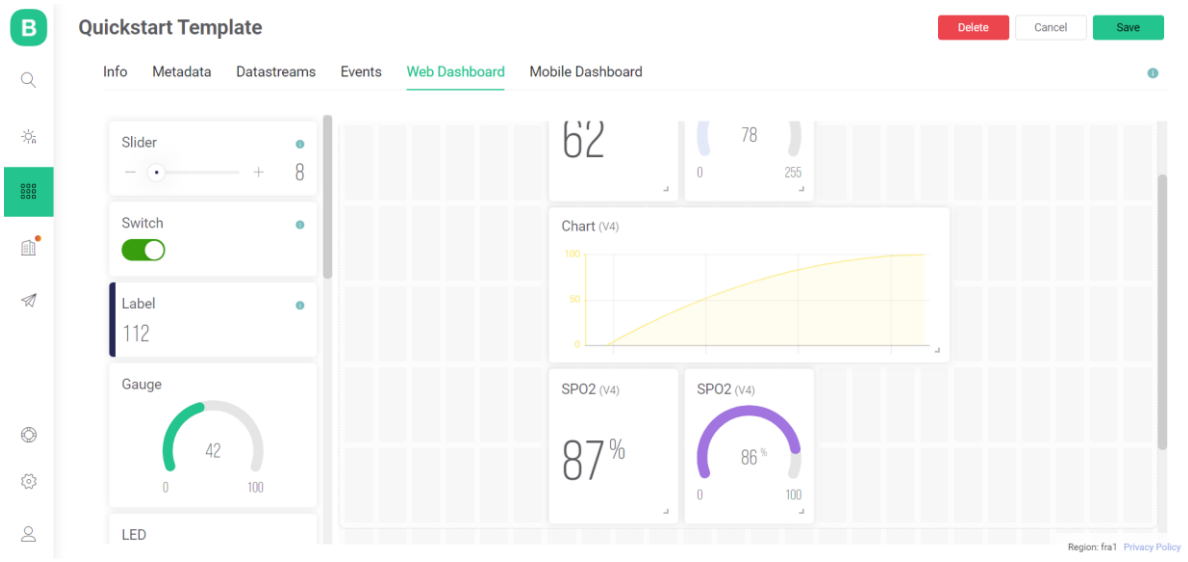

Рисунок 2.5 – Вікно розробки додатку веб версія

Крім цього не менш важливою перевагою цієї технології є можливість налаштування сповіщень в отриманому мобільному додатку. Тобто якщо якийсь з параметрів людини таких як сатурація або пульс досягає критичного рівня, на мобільний телефон буде приходити сповіщення і відповідно людина буде реагувати на критичний стан. Також можна налаштувати період відправлення таких сповіщень до моменту вирішення проблеми. Даний сервіс повністю забезпечує потреби пристрою та дозволяє отримувати дані дистанційно. Якщо порівнювати отриманий пристрій з аналогами то можна зробити висновок, що результат повністю забезпечує потреби оскільки, зв'язок відбувається за допомогою Wi-Fi це дозволяє отримати високу якість та радіус зв’язку між пристроєм та мобільним телефоном, за допомогою якого і відбувається дистанційний моніторинг стану, можливість передавати тривожні сповіщення дозволяє не дивитись в телефон постійно, а займатись своїми справами. Налаштування передачі даних від мікроконтролера до додатку і навпаки відбувається за наступним принципом. Технологія Blynk дозволяє використовувати віртуальні порти – це засіб ідентифікації додатком даних та зв’язку з кодом програми, і є унікальним для кожного поля даних. Віртуальні порти можуть передавати одні і ті ж дані одразу на декілька

компонентів додатку. Вікно налаштувань віртуального порту зображено на рисунку 2.6.

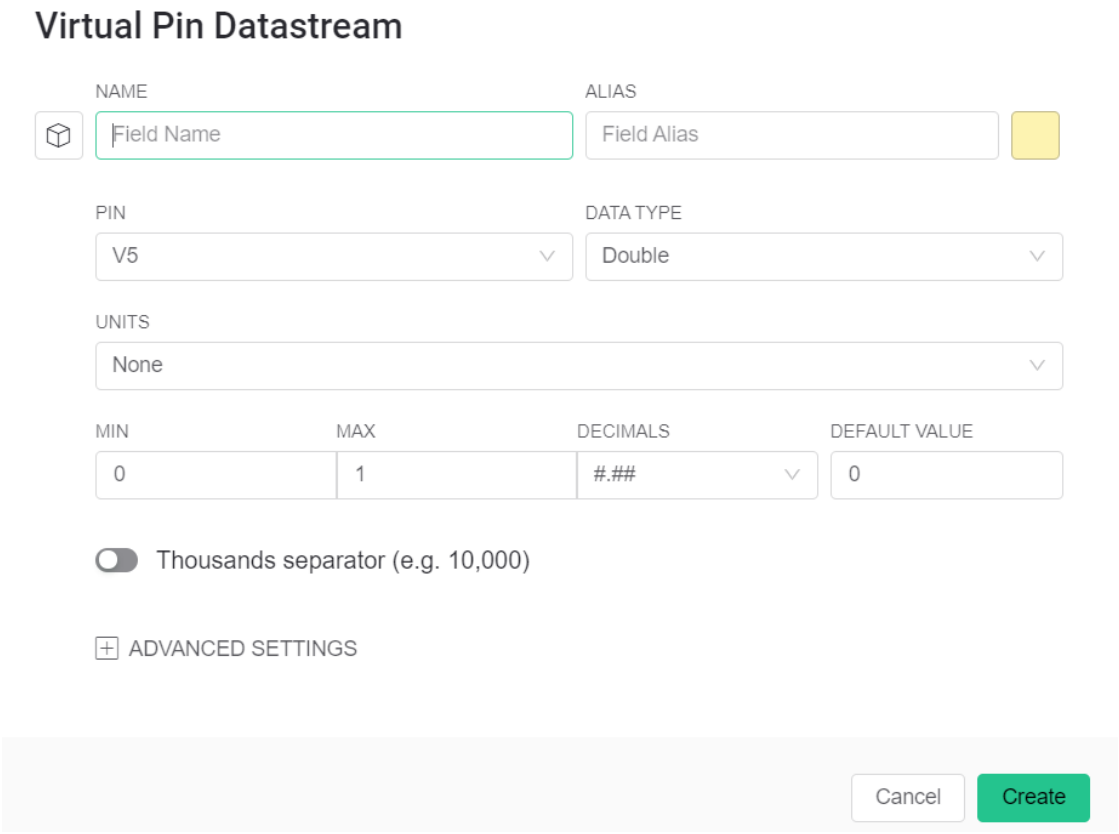

Рисунок 2.6 – вікно налаштувань віртуального порту

Віртуальний порт повинен обов’язково мати назву, номер порту, тип даних, також є можливість обрати одиниці вимірювання, мінімальне, максимальне значення та стандартне значення яке ставиться при ініціалізації поля. Для отримання або внесення даних в віртуальний порт з метою їх подальшого відображення використовуються наступні методи.

Blynk.virtualWrite(V0, 50);

Цей метод передає за допомогою Wi-Fi на віртуальний порт під номером 0 число 50. Таким чином відбувається передача даних від мікроконтролера на мобільний додаток, завдяки цьому можна реалізувати метод який буде передавати дані з певною періодичністю.

void sendDataToBlynk()

```
{
  wakeModemSleep();
     Blynk.virtualWrite(V3, beatAvg);
     Blynk.virtualWrite(V4, 50);
     Blynk.virtualWrite(V0, 50);
     tsLastReport = millis();
     if (millis() - reportTime > WIFI_REPORT_PERIOD_MS)
     {
       reportTime = millis();
       setModemSleep();
       }
  }
```

Метод sendDataToBlynk запускається з основного циклу програми з періодом в 5 секунд, після чого вмикає модуль Wi-Fi передає необхідні дані в додаток Blynk, вимикає Wi-Fi.

2.4 Система сповіщень

Система потребує реалізації сповіщень, для того щоб користувач весь час мав змогу дізнатись про критичні ситуації. В системі реалізовано два види сповіщень, перший це відправлень сповіщень на телефон, другий звуковий за допомогою динаміка в пристрої. Відправлення сповіщень відбувається за допомогою Blynk та програмного коду. Метод який реалізовано в бібліотеці Blynk дозволяє відправити сповіщення в додаток, з можливістю налаштувати конкретний текст сповіщення і момент коли він спрацює. Наприклад для низького рівня сатурації створено умову в якій викликається метод сповіщення.

```
 if(sp02Avg < 90)
  {
    Blynk.logEvent("lowsaturationlevel");
```

```
tone(3, 800);
}
```

Цей метод перевіряє рівень сатурації і вразі його заниженого показника відправляє сповіщення в додаток та вмикається звукове сповіщення за допомогою динаміка. Таким же чином реалізовано перевірку та сповіщення інших типів таких як відсутність пальця в зоні дії датчика або ж завищений або занижений рівень серцебиття у людини. Приклад системи сповіщень в мобільному додатку відображено на рисунку 2.7.

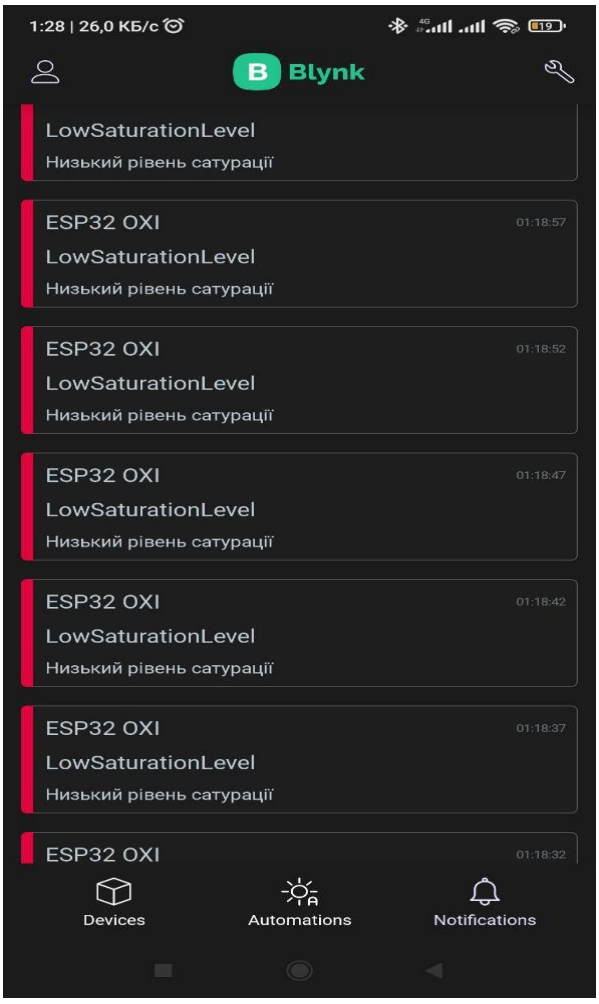

Рисунок 2.7 - вигляд системи сповіщень в мобільному додатку

Завдяки розробленій системі сповіщень, користувач який доглядає за хворим постійно буде знати його стан і вразі потреби може миттєво відреагувати. Крім цього система дозволить відразу чітко зрозуміти яка саме несправність чи ситуація відбулась.

2.5 Метод визначення кількості серцевих скорочень

Кількість серцевих скорочень визначається алгоритмом який розраховує відстань між сусідніми піками сигналу. Алгоритм ділиться на декілька частин. Диференціація сигналу. Це дає змогу наблизити сигнал до нуля та зменшити коливання під час зміни освітлення при вимірюваннях. Після диференціювання сигналу можна порівнювати висоту піків сигналу. Для цього сигнал потрібно відфільтрувати за допомогою фільтру низьких частот його реалізація показана в наступному методі.

```
int16_t lowPassFIRFilter(int16_t din)
{
  cbuf[offset] = din;
  int32_t z = mul16(FIRCoeffs[11], cbuf[(offset - 11) & 0x1F]);
  for (uint8_t i = 0 ; i < 11 ; i++)
  {
    z += mul16(FIRCoeffs[i], cbuf[(offset - i) & 0x1F] + cbuf[(offset - 22 + i) & 0x1F]);
  }
  offset++;
  offset %= 32; //Wrap condition
  return(z >> 15);
}
```

Сигнал після проходження через фільтр низьких частот зображено на рисунку 2.8

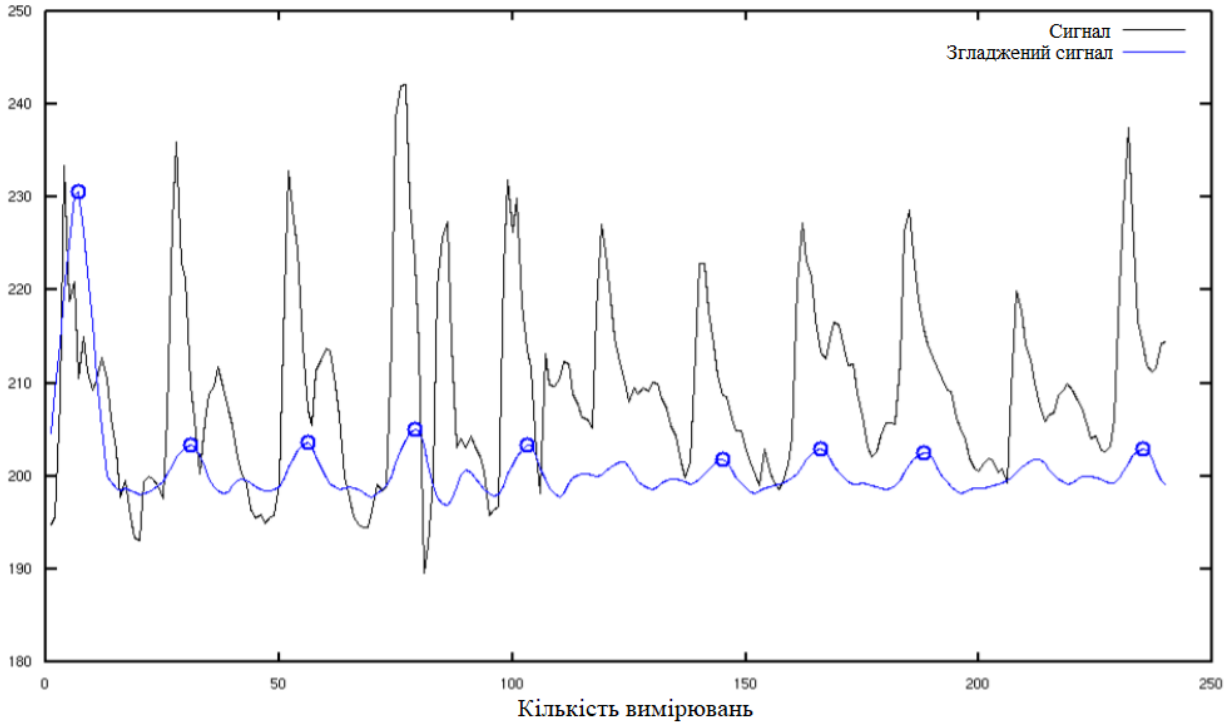

Рисунок 2.8 – Сигнал після проходження фільтру низьких частот

Після проведення вимірювань впродовж 10 секунд та визначення кількості пікових сигналів яка дорівнює 20 розраховуємо дистанції між піками для кожного набору піків з попереднього вимірювання. Далі відбувається вибір наборів піків дисперсія значень яких є мінімальною серед інших варіантів.

Розраховуємо середнє значення відстаней між сусідніми вершинами для набору вершин, обраного попередньо. Частота вимірювань помножена на 60 і розділена на середнє значення відстаней між сусідніми вершинами дозволяє отримати кількість серцевих скорочень в удар на хвилину[11].

2.6 Розробка програмного забезпечення для системи

Під час розробки системи не менш важливою частиною, після вибору компонентів є розробка алгоритмів та програмного коду роботи програми, оскільки без цього будь-які датчики та пристрої є неефективними. Тому потрібно розробити методи для обробки даних від датчиків та подальшої їх

відправки на пристрій дистанційного моніторингу. Програмне забезпечення займає велику роль в системи, від його алгоритмів обробки залежить точність, та ефективність проведених вимірів, енергоспоживання пристрою. На рисунку 2.9 зображено вікно розробки середовища Arduino IDE.

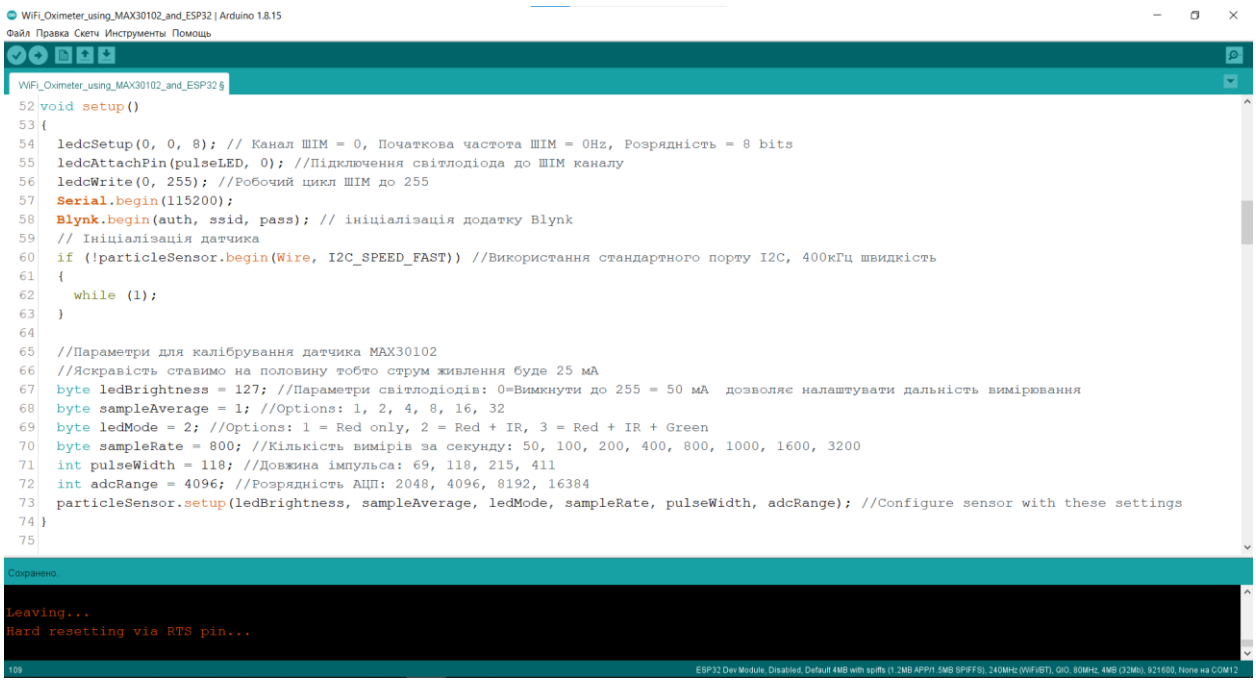

Рисунок 2.9 – середовище розробки Arduino IDE

Завдяки середовищам розробки можна скоротити час розробки програмного забезпечення в декілька разів. Наступним етапом є алгоритм роботи програми. При роботі з мікроконтролером початковим етапом є його ініціалізація, тобто налаштування тактування процесору, портів та режимів їх роботи, початкове налаштування периферійних пристроїв та датчиків[12].

Основним етапом роботи мікроконтролеру є нескінчений цикл в якому весь час виконуються перевірки та реакції на зовнішні та внутрішні події в системі. Після чого в мобільному додатку відбувається їх аналіз, та реакція на них, яка може бути як у вигляді попереджень про певні неполадки так і тривожних повідомлень при небезпечних

ситуаціях. Блок-схема алгоритму роботи програми показана на рисунку 2.10.

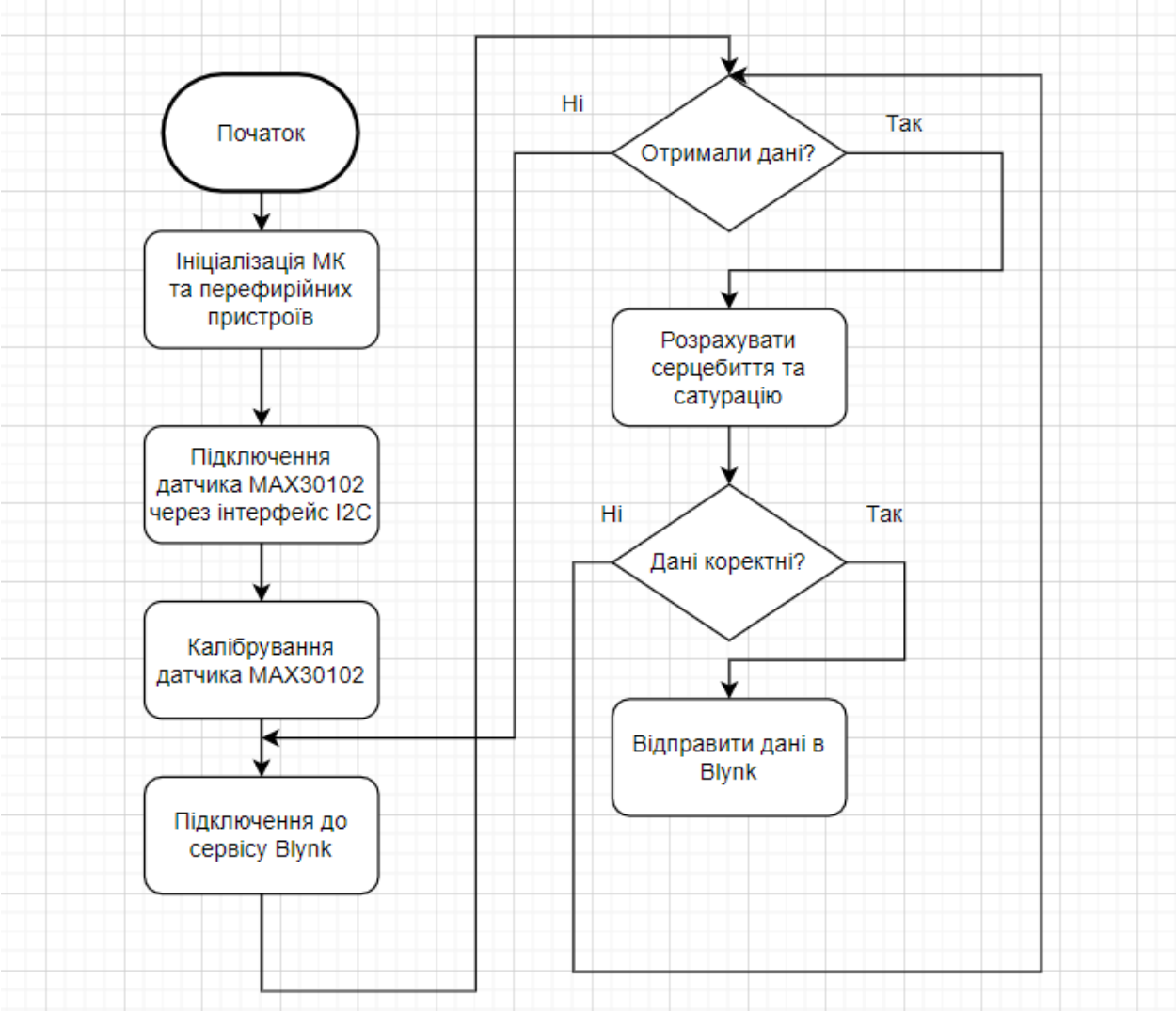

Рисунок 2.10 – Блок схема роботи програми мікроконтролера

В результаті розробки програмного забезпечення було реалізовано обмін даними між мікроконтролером та пристроєм дистанційного моніторингу таким як смартфон. Та покращено автономність системи за рахунок реалізації алгоритму передачі даних який передбачає увімкнення та вимкнення Wi-Fi для збереження енергії.

2.7 Висновки до розділу

В цьому розділі було розглянуто компоненти системи, основні способи покращення автономності, та спосіб реалізації та моніторингу за станом хворого, а також розглянуто алгоритм визначення кількості серцевих скорочень на хвилину за допомогою датчика MAX30102, та блок-схему алгоритму роботи програми. В результаті чого отримали результати які дозволять вдосконалити систему, моніторингу стану людини.

3 ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

Проведення тестування та аналізу отриманих результатів є однією з найважливіших складових під час розробки будь-якої системи. Тому удосконалена система підлягає перевірці.

3.1 Моделювання та тестування системи

Для цих цілей використовуватиметься програмне забезпечення від розробника датчика MAX30102 яке має назву MAX30102 Evaluation Kit, це програмне забезпечення дозволяє чітко регулювати параметри датчика та аналізувати його показники та ефективність проведених вимірів. Програма включає в себе вікно для налаштування датчика яке зображено на рисунку 3.1.

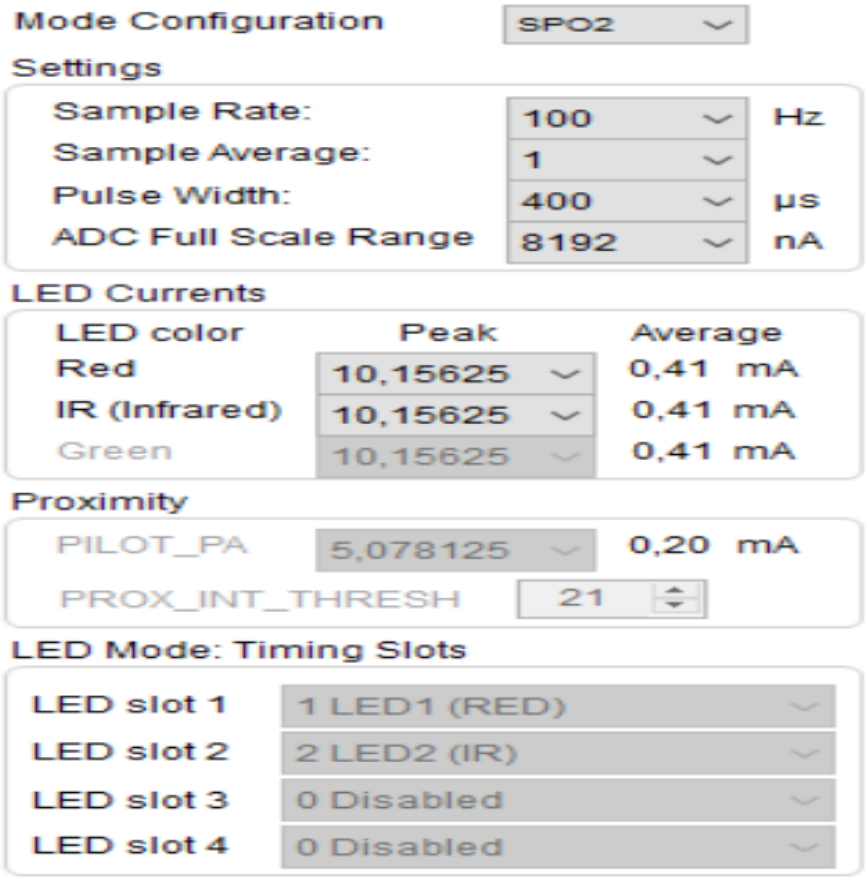

Рисунок 3.1 – Вікно налаштувань середовища MAX30102 Evaluation Kit

З цього вікна видно, що програма забезпечує вибір режиму вимірювання, а саме частота серцебиття або ж рівень насиченості киснем та налаштування таких параметрів як частота вимірювань, середнє значення, ширина імпульсів, розрядність аналого-цифрового перетворювача, а також струм який подається на червоний та інфрачервоний світлодіоди. Програма дозволяє автоматично визначати середній рівень споживання струму світлодіодами при будь-яких налаштуваннях кількості вимірювання та ширини імпульсу. Завдяки програмі можна зручно змоделювати, та протестувати роботу системи в реальному часі. Це дасть змогу перевірити отримані результати та відкалібрувати роботу пристрою для знаходження оптимального варіанту для стандартних вимірювань, а можливість налаштувань за допомогою смартфону, дозволить користувачу підлаштувати роботу пристрою саме під себе. Також програма дає можливість записувати дані в файл з метою їх подальшого аналізу. Тому саме ця програма ідеально підходить для тестування роботи системи та точності її вимірювань[13].

Калібрування системи, та її налаштування залежить від багатьох параметрів середовища, довжина хвилі та інтенсивність світлодіоду, реакція фотоприймача, положення частини тіла і навколишній рівень освітлення.

Для зручного калібрування програма включає в себе вікно з вхідними та вихідними даними датчика, це вікно зображено на рисунку 3.2.

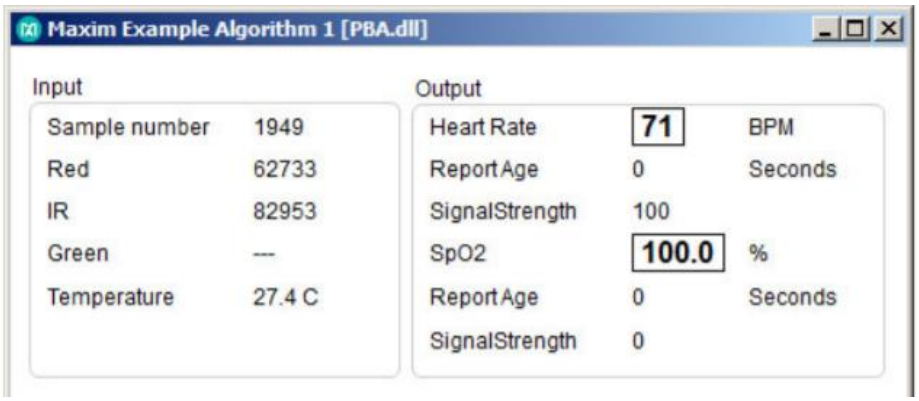

Рисунок 3.2 – Вікно вхідних та вихідних даних в програмі MAX30102 EV KIT

Графіки які теж відображаються в програмі дозволяють проаналізувати зміни сигналів від датчиків. На цих графіках чудово видно як може змінитись сигнал в залежності від положення тканин де проводяться виміри, та вплив світла на сигнал. Вікно графіків зображено на рисунку 3.3.

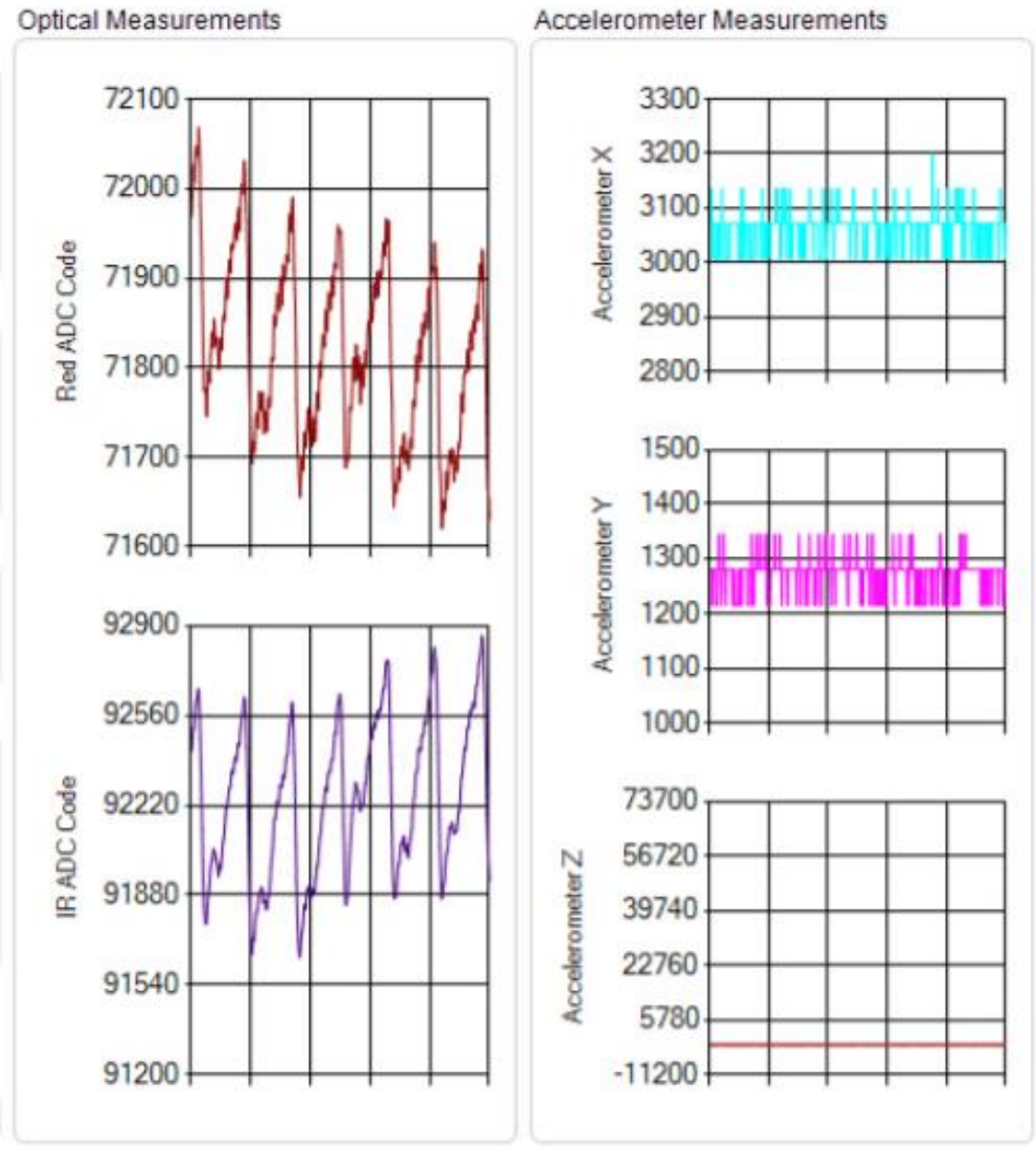

Рисунок 3.3 – Вікно графіків програми MAX30102 EV KIT

Алгоритм її роботи можна виразити за допомогою наступної блок-схеми зображеної на рисунку 3.4.

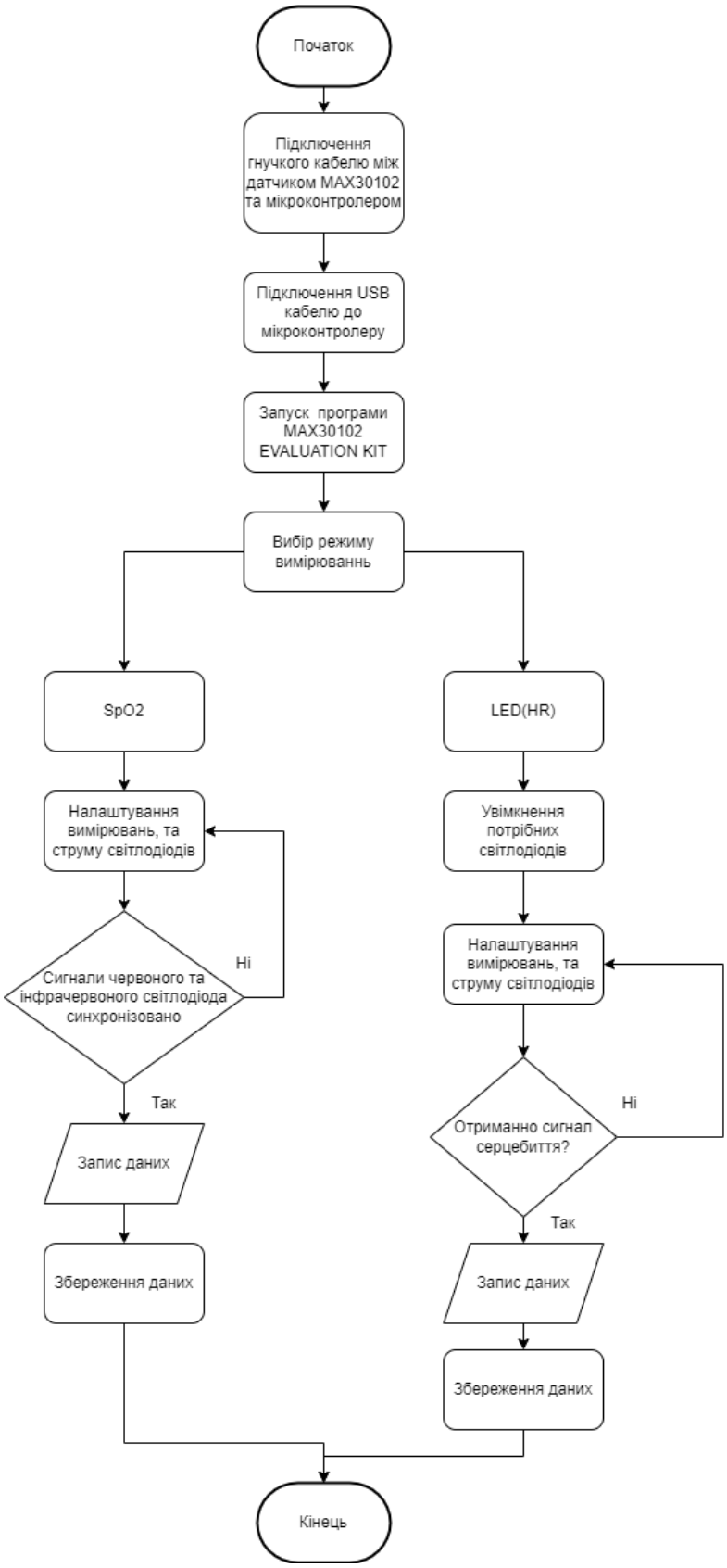

Рисунок 3.4 – Блок-схема роботи програми MAX30102 EV KIT

3.2 Тестування оптимальної яскравості світлодіодів

Для тестування детектування об’єктів перед собою, була розроблена наступна програма яка завантажується в мікроконтролер, після чого проводиться калібрування. Робота цієї програми має наступний принцип, спочатку відбувається ініціалізація та налаштування яскравості світлодіодів, вводиться змінна яка рахує кількість вимірів, та змінна яка рахує час виміру. Налаштування потребує лише інфрачервоного світлодіоду, тому в залежності від датчика всі інші вимикаються. Отримані дані роботи програми виводяться на екран комп’ютера[14]. Результатом роботи програми є оптимальне значення яскравості світлодіоду для датчика. Алгоритм роботи показано на рисунку 3.5

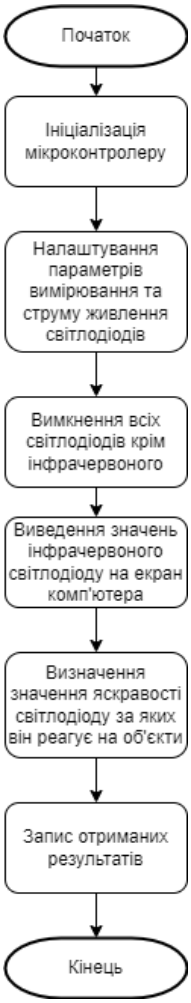

Рисунок 3.5 – Алгоритм роботи програми з визначення оптимальної яскравості світлодіоду.

Під час розрахунків струму живлення для світлодіоду використовувалось середнє значення у вигляді 25 мА, але завдяки тестуванню та визначенню оптимальної яскравості світлодіоду, виявилось, що дальність спрацювання світлодіоду при струму живлення в 7 мА не відрізняється від ефективності струму живлення в 25 мА тобто завдяки цьому можна збільшити автономність роботи пристрою. Розрахунок часу роботи представлено в табл. 3.1

Таблиця 3.1 – Час роботи вдосконаленого пристрою за різних режимі передачі даних

Струм живлення	Акумулятор	Період передачі	Час роботи
118,5216 мА	3000 мАг	5 с	25.5 год
94,5216 мА	3000 мАг	10 с	31.5 год
86,5216 мА	3000 мАг	15 с	34.5 год
82,5216 мА	3000 мАг	20 с	36.5 год
78,5216 мА	3000 мАг	30 с	38.5 год
118,5216 мА	6000 мАг	5 с	50.5 год
94,5216 мА	6000 мАг	10 с	63.5 год
86,5216 мА	6000 мАг	15 с	69.5 год
82,5216 мА	6000 мАг	20 с	72.5 год
78,5216 мА	6000 мАг	30 с	76.5 год

Виходячи з даних таблиці, завдяки тестуванню вдалось збільшити час роботи пристрою на 1.5-2 години завдяки зменшенню струму живлення світлодіодів датчика MAX30102. Якщо порівняти дану систему з побутовим пульсоксиметром, який зазвичай живиться двома акумуляторами типу АА, що з’єднанні послідовно, а тому їх ємність дорівнює 1500 мАг. Струм живлення таких пульсоксиметрів в середньому становить 40 мАг під час вимірювань, звідси можна зробити висновок, що час роботи такого пульсоксиметра в режимі постійного моніторингу становитиме 37.5 годин, без можливості

підзарядки. Медичний в свою чергу має ємність акумулятора близьку до 3000 мАг і струм живлення 125 мА тому термін роботи такого пристрою в режимі моніторингу стану людини становить 24 години, після чого потрібна підзарядка. Отримана система при ємності акумулятора в 3000 мАг та частоті передачі в 10 секунд має автономність на 6 годин меншу ніж у побутового пульсоксиметр, але при цьому при ємності в 6000 мАг на 26 годин більшу, що видно з рисунку 3.6.

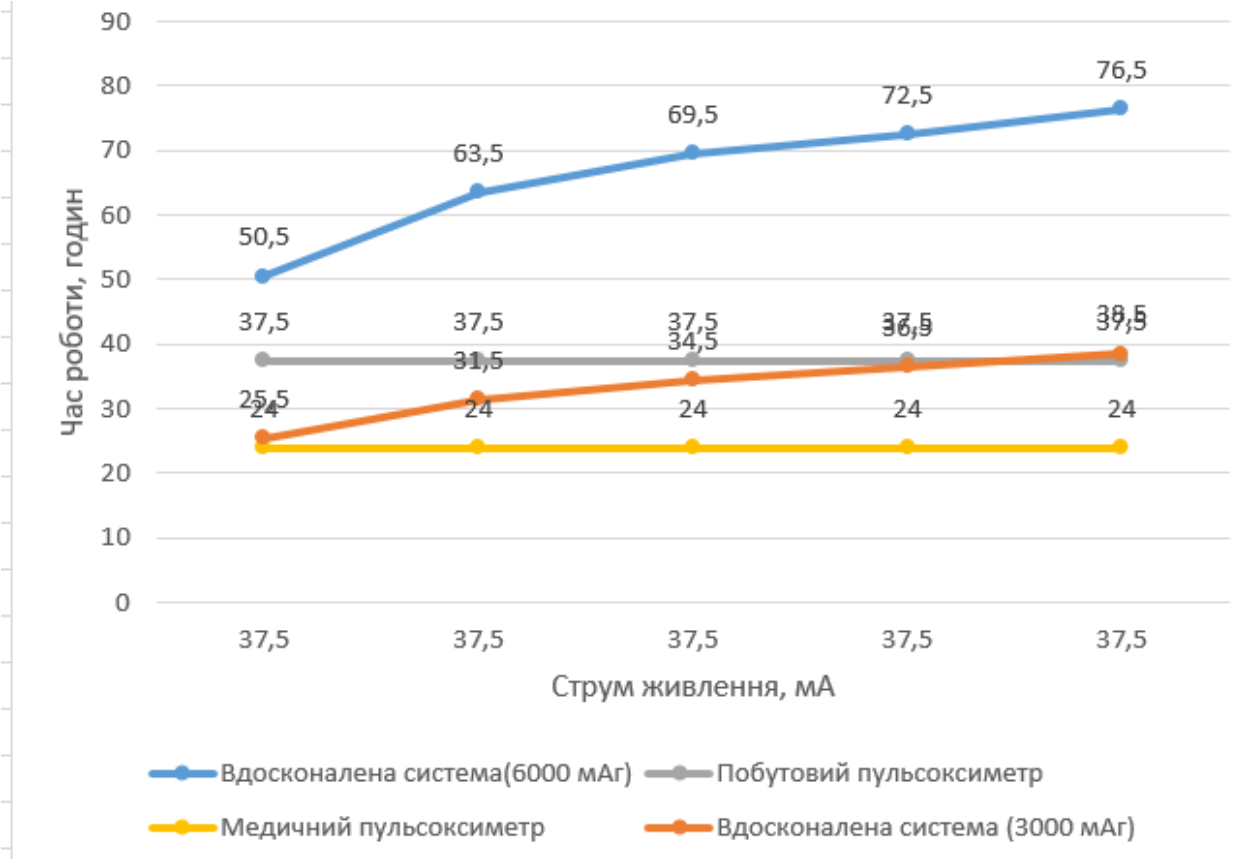

Рисунок 3.6 – Графік часу роботи пульсоксиметрів.

Автономність системи виходить досить високою враховуючи дистанційну передачу даних через Wi-Fi, що створює додаткове навантаження на елементи живлення.

3.3 Висновки до розділу

В цьому розділі було проведено калібрування та тестування розробленої системи, під час тестування розроблено та в подальшому реалізовано алгоритм програми для оптимізації електроспоживання за рахунок вибору оптимального режиму та дальності дії світлодіодів датчика пульсу та насиченості крові, вдалось покращити автономність системи завдяки зменшенню яскравості світлодіодів без втрати при цьому точності та ефективності роботи.

ВИСНОВКИ

У кваліфікаційній роботі була досліджена та вдосконалена система моніторингу стану людини, за принципом пульсоксиметра. Проаналізовано системи аналоги, виділено основні потреби та проблеми які в них є. Досліджено оптимальні алгоритми та методи для визначення насиченості крові киснем та кількості серцевих скорочень. В створено систему вбудовано технологію для дистанційного моніторингу стану, та калібрування пристрою. Результатом кваліфікаційної роботи є система моніторингу яка дозволяє в залежності від модифікації забезпечити від 31 до 63 годин автономної роботи без необхідності підзарядки, що дозволить ефективно контролювати стан людини хворої на корона вірус. Визначено компоненти та датчики які використовуватимуться в системі, проаналізовано їх характеристики, можливості та принципи роботи. Розроблено систему сповіщень яка відправляє повідомлення про несправності та критичні ситуації на мобільний додаток. Порівняно можливості автономної роботи побутових, медичних та розробленого пульсоксиметрів. В результаті розробки вдалося зменшити енергоспоживання модуля Wi-Fi, що дозволило збільшити час роботи системи. Змодельовано роботу системи з датчиком MAX30102 за допомогою програми MAX30102 EV KIT, під час моделювання виявлено, що яскравість в діапазоні від 7 мА до 25 мА не впливає на точність та дистанцію вимірів, це дало змогу покращити автономність системи ще на декілька годин.

СПИСОК ЛІТЕРАТУРИ

[1]. Корона вірус в Україні [Електронний ресурс] – Режим доступу до ресурсу: https://index.minfin.com.ua/ua/reference/coronavirus/ukraine/

[2]. Пульсоксиметр [Електронний ресурс] – Режим доступу до ресурсу: https://www.pulsoksimetr.biz/pulsoksimetr-oksitest-1-s-datchikami-dlya-vzroslyh-i-detey.php

[3]. Керівництво ВОЗ з пульсоксиметрії [Електронний ресурс] – Режим доступу до ресурсу: https://csma.org.ua/wp-content/uploads/2018/06/%D0%9F%D0%A3%D0%9B%D0%AC%D0%A1%D0%9E%D0%9A%D0%A1%D0%98%D0%9C%D0%95%D0%A2%D0%A0%D0%86%D0%AF.pdf

[4]. Капнографія та капнометрія [Електронний ресурс] – Режим доступу до ресурсу: http://gmt-med.ru/?education=kapnografija-i-kapnometrija-2

[5]. Автономний кардіоремінь [Електронний ресурс] – Режим доступу до ресурсу: https://prostobzor.com/garmin-hrm-pro/

[6]. Принцип роботи літієвих акумуляторів [Електронний ресурс] – Режим доступу до ресурсу: https://habr.com/ru/post/532616/

[7]. ESP32 datasheet [Електронний ресурс] – Режим доступу до ресурсу: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[8]. ESP32 збереження енергії [Електронний ресурс] – Режим доступу до ресурсу: https://www.mischianti.org/2021/03/10/esp32-power-saving-modem-and-light-sleep-2/

[9]. Ємність акумулятора [Електронний ресурс] – Режим доступу до ресурсу https://220volt.com.ua/news/useful/akkumulyatornie-batarei/chto-takoe-emkost-akkumulyatora-metodika-ee-rascheta.html

[10]. Технологія дистанційного зв’язку Blynk [Електронний ресурс] – Режим доступу до ресурсу: https://www.kdnuggets.com/2014/06/dlib-library-machine-learning.html

[11]. Методика визначення серцебиття [Електронний ресурс] – Режим доступу до ресурсу: https://www.maximintegrated.com/en/design/technical-documents/app-notes/6/6845.html

[12]. Arduino IDE [Електронний ресурс] – Режим доступу до ресурсу: https://www.arduino.cc/

[13]. Методика тестування комп’ютерних систем - Euclidean group [Електронний ресурс] – Режим доступу до ресурсу: https://www.ixbt.com/platform/cpu-method-2020.html

[14]. SparkFun_MAX30102_Sensor_Library [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library

Додаток А

Лістинг програми для визначення рівня насиченості крові та пульсу

Код програми:

```
// Template ID, Device Name та Auth Token
#define BLYNK_TEMPLATE_ID "TMPLk03zgBlw"
#define BLYNK_DEVICE_NAME "Quickstart Template"
#define
BLYNK_AUTH_TOKEN"MfCc4M5f5R2buzFrM03_dCxjXw33lOCn"

#include <Wire.h> // бібліотека для інтерфейсу звязку I2C
#include <Blynk.h> // бібліотека для роботи з технологією Blynk
#include <BlynkSimpleEsp32.h>
#include <WiFi.h>  // бібліотека для роботи з модулем Wi-Fi

#include "MAX30105.h"// бібліотеки для роботи з датчиком пульсу
#include "spo2_algorithm.h"
#include "heartRate.h"

MAX30105 particleSensor; // створення обєкту типу датчик пульсу
#define MAX_BRIGHTNESS 255 // максимальна яскравість світлодіодів

uint32_t irBuffer[100]; // масив для збереження даних інфрачервоного
uint32_t redBuffer[100];  //масив для збереження даних червоного

#define REPORTING_PERIOD_MS 5000 // частота відправки даних
#define WIFI_REPORT_PERIOD_MS 1000 // час роботи Wi-Fi
```

```
char auth[] = BLYNK_AUTH_TOKEN; // Ідентифікатор в додатку Blynk

// Дані Wi-Fi до якого підєднується пристрій
// "" якщо пароль відсутній
char ssid[] = "Redmi"; // Назва мережі
char pass[] = "1230123vik";// Пароль мережі

uint32_t tsLastReport = 0;  //час останьої відправки даних в додаток Blynk
uint32_t reportTime = 0;

int32_t bufferLength; //довжина даних, що відправляються
int32_t spo2; //Значення сатурації
int8_t validSPO2; //індикатор правильності даних сатурації (якщо датчик
int32_t heartRate; //Значення серцебиття
int8_t validHeartRate; // індикатор правильності даних серцебиття

byte pulseLED = 2; //блимати світлодіодом при фіксації серцебиття
byte readLED = 19; //Червоний світлодіод блимати якщо дані зчитано

long lastBeat = 0; //Час коли прийшов останій біт

float beatsPerMinute; //ударів на хвилину
int beatAvg = 0, sp02Avg = 0; //Середнє серцебиття та сатурація
float ledBlinkFreq; //частота блимання світлодіода

void setModemSleep(); // метод для переходу мікроконтролера в режим
void wakeModemSleep(); // метод для увімкнення звичайного режиму для
void setup()
{
```

```
    ledcSetup(0, 0, 8); // Канал ШІМ = 0, Початкова частота ШІМ = 0Hz, Розрядність = 8 bits
    ledcAttachPin(pulseLED, 0); //Підключення світлодіода до ШІМ каналу
    ledcWrite(0, 255); //Робочий цикл ШІМ до 255
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass); // ініціалізація додатку Blynk
    // Ініціалізація датчика
    if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Використання I2C
    {
      while (1);
    }

    //Параметри для калібрування датчика MAX30102
    //Яскравість ставимо на половину тобто струм живлення буде 25 мА
    byte ledBrightness = 127; //Параметри світлодіодів: 0 до 255
    byte sampleAverage = 1; //Options: 1, 2, 4, 8, 16, 32
    byte ledMode = 2; //Options: 1 = Red only, 2 = Red + IR
    byte sampleRate = 800; // вимірів за секунду: 50, 100, 200, 400, 800, 1000,
    int pulseWidth = 118; //Довжина імпульса: 69, 118, 215, 411
    int adcRange = 4096; //Розрядність АЦП: 2048, 4096, 8192, 16384
    particleSensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth, adcRange); //Ініціалізація датчика
  }

  void loop()
  {
    bufferLength = 100; //Довжина буфера  100 зберігає виміри 4 секунди

    //зчитати перших 100 вимірів та визначити діапазон сигналу
    for (byte i = 0 ; i < bufferLength ; i++)
```

```
{
  while (particleSensor.available() == false) //отримали нові дані?
    particleSensor.check(); //Перевірка нових даних

  redBuffer[i] = particleSensor.getIR();// інфрачервоний світлодіод
  irBuffer[i] = particleSensor.getRed();// Записуємо значення червоного св
  particleSensor.nextSample(); //Наступний вимір

}

//Розрахувати частоту серцебиття та сатурації після перших 100 вимірів
maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer,
&spo2, &validSPO2, &heartRate, &validHeartRate);

//Безперервне вимірювання серцебиття та сатурації, оновлення даних
while (1)
{
  Blynk.run(); // зєднуємось з додатком Blynk
  //Запис перших 25 вимірів в память та зміщення 75 вимірів на гору
  for (byte i = 25; i < 100; i++)
  {
    redBuffer[i - 25] = redBuffer[i];
    irBuffer[i - 25] = irBuffer[i];
  }
  Blynk.virtualWrite(V4, 50);
  //Беремо 25 вимірів перед розрахунком частоти пульсу
  for (byte i = 75; i < 100; i++)
  {
    while (particleSensor.available() == false) //отримали нові дані?
      particleSensor.check(); //Перевірка сенсора на нові дані
```

```
digitalWrite(readLED, !digitalRead(readLED)); //Блимаємо світлодіодом при зчитуванні даних

redBuffer[i] = particleSensor.getRed();
irBuffer[i] = particleSensor.getIR();
particleSensor.nextSample(); //Перехід до нових вимірів

long irValue = irBuffer[i];

//Розрахунок ударів на хвилину
if (checkForBeat(irValue) == true)
{
  //Відчули серцебиття
  long delta = millis() - lastBeat;
  lastBeat = millis();

  beatsPerMinute = 60 / (delta / 1000.0);
  beatAvg = (beatAvg + beatsPerMinute) / 2;

  if (beatAvg != 0)
    ledBlinkFreq = (float)(60.0 / beatAvg);
  else
    ledBlinkFreq = 0;
  ledcWriteTone(0, ledBlinkFreq);
}
if (millis() - lastBeat > 10000)
{
  beatsPerMinute = 0;
  beatAvg = (beatAvg + beatsPerMinute) / 2;
```

```
      if (beatAvg != 0)
        ledBlinkFreq = (float)(60.0 / beatAvg);
      else
        ledBlinkFreq = 0;
      ledcWriteTone(0, ledBlinkFreq);
    }
  }
```

maxim_heart_rate_and_oxygen_saturation(irBuffer,bufferLength, redBuffer, &spo2, &validSPO2, &heartRate, &validHeartRate);

```
  //Визначення середнього значення рівня сатурації
  if (validSPO2 == 1 && spo2 < 100 && spo2 > 0)
  {
    sp02Avg = (sp02Avg + spo2) / 2;
  }
  else
  {
    spo2 = 0;
    sp02Avg = (sp02Avg + spo2) / 2;;
  }

  //Налаштування періодичності відправки даних в додаток Blynk
  if (millis() - tsLastReport > REPORTING_PERIOD_MS)
  {
    sendDataToBlynk();
  }
 }
```

```
}
BLYNK_WRITE(V1)
{
 int powerLevel = param.asInt(); // Значення з додатку Blynk
particleSensor.setPulseAmplitudeRed(powerLevel);
 particleSensor.setPulseAmplitudeIR(powerLevel);
 particleSensor.setPulseAmplitudeProximity(powerLevel);
 Serial.println(powerLevel);
}
BLYNK_WRITE(V2)
{
 int sampleRate = param.asInt();
 Serial.println(sampleRate);
 if (sampleRate == 0)
 { particleSensor.setSampleRate(50); //Вимірюваннь на секунду
  Blynk.virtualWrite(V0, 50);
 }
 else if (sampleRate == 1)
 { particleSensor.setSampleRate(100);
  Blynk.virtualWrite(V0, 100);
 }
 else if (sampleRate == 2)
 { particleSensor.setSampleRate(200);
  Blynk.virtualWrite(V0, 200);
 }
 else if (sampleRate == 3)
 { particleSensor.setSampleRate(400);
  Blynk.virtualWrite(V0, 400);
 }
 else if (sampleRate == 4)
```

```
  {
   particleSensor.setSampleRate(800);
   Blynk.virtualWrite(V0, 800);
  }
  else if (sampleRate == 5)
  { particleSensor.setSampleRate(1000);
   Blynk.virtualWrite(V0, 1000);
  }
 }

 void sendDataToBlynk() // відправлення даних в додаток Blynk
 {
  wakeModemSleep();
  if (sp02Avg < 90)// перевірка рівня сатурації
  {
   Blynk.logEvent("lowsaturationlevel");
   tone(3, 800);
  }
  Blynk.virtualWrite(V3, beatAvg);
  Blynk.virtualWrite(V4, sp02Avg);
  tsLastReport = millis();
  if (millis() - reportTime > WIFI_REPORT_PERIOD_MS)
  {
   reportTime = millis();
   setModemSleep();
  }
 }
 void setModemSleep() { // увімкнення та вимкнення режиму модему
WiFi.setSleep(true);
  if (!setCpuFrequencyMhz(40)) {
```

```
  }
}
void wakeModemSleep() {
  setCpuFrequencyMhz(240);
}
```

Додаток Б

Лістинг програми для налаштування оптимальної яскравості світлодіодів датчика пульсу

Код програми:

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

MAX30105 particleSensor;

long samplesTaken = 0; //Counter for calculating the Hz or read rate
long unblockedValue; //Average IR at power up
long startTime; //Used to calculate measurement rate

void setup()
{
  Serial.begin(9600);
  Serial.println("MAX30105 Presence Sensing Example");

  // Initialize sensor
  if (particleSensor.begin(Wire, I2C_SPEED_FAST) == false) //Use default I2C port, 400kHz speed
  {
    Serial.println("MAX30105 was not found. Please check wiring/power. ");
    while (1);
  }
```

```
  //Setup to sense up to 18 inches, max LED brightness
  byte ledBrightness = 0xFF; //Options: 0=Off to 255=50mA
  byte sampleAverage = 4; //Options: 1, 2, 4, 8, 16, 32
  byte ledMode = 2; //Options: 1 = Red only, 2 = Red + IR, 3 = Red + IR + Green
  int sampleRate = 400; //Options: 50, 100, 200, 400, 800, 1000, 1600, 3200
  int pulseWidth = 411; //Options: 69, 118, 215, 411
  int adcRange = 2048; //Options: 2048, 4096, 8192, 16384

 particleSensor.setup(ledBrightness, sampleAverage, ledMode, sampleRate, pulseWidth, adcRange);

  particleSensor.setPulseAmplitudeRed(0); //Turn off Red LED
  particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED

  //Take an average of IR readings at power up
  unblockedValue = 0;
  for (byte x = 0 ; x < 32 ; x++)
  {
    unblockedValue += particleSensor.getIR(); //Read the IR value
  }
  unblockedValue /= 32;

  startTime = millis();
 }

 void loop()
 {
  samplesTaken++;
```

```
  Serial.print("IR[");
  Serial.print(particleSensor.getIR());
  Serial.print("] Hz[");
  Serial.print((float)samplesTaken / ((millis() - startTime) / 1000.0), 2);
  Serial.print("]");

  long currentDelta = particleSensor.getIR() - unblockedValue;

  Serial.print(" delta[");
  Serial.print(currentDelta);
  Serial.print("]");

  if (currentDelta > (long)100)
  {
    Serial.print(" Something is there!");
  }

  Serial.println();
}
```