

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

## КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка інформаційної системи дистанційного моніторингу  
температури і сатурації пацієнтів сімейного лікаря

Виконав: студент IV курсу, групи СТс-41  
спеціальності 126 Інформаційні системи та технології  
(шифр і назва спеціальності)

Ткачик Д.О.  
(підпис) (прізвище та ініціали)

Керівник Дудкін П.Д.  
(підпис) (прізвище та ініціали)

Нормоконтроль Марценко С.В.  
(підпис) (прізвище та ініціали)

Завідувач кафедри Боднарчук І.О.  
(підпис) (прізвище та ініціали)

Рецензент Цуприк Г.Б.  
(підпис) (прізвище та ініціали)

Тернопіль  
2023

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
Боднарчук І.О.  
(підпис) (прізвище та ініціали)  
« 23 » січня 2023 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 126 Інформаційні системи та технології  
(шифр і назва спеціальності)

Студенту Ткачику Денису Олеговичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інформаційної системи дистанційного моніторингу температури і сатурації пацієнтів сімейного лікаря

Керівник роботи Дудкін Павло Дмитрович, к.е.н., доцент кафедри МП  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 7 » лютого 2023 року № 4/7-134

2. Термін подання студентом завершеної роботи 14 червня 2023р.

3. Вихідні дані до роботи Перелік літературних та інтернет джерел

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Постановка задачі та формування вимог до інформаційної системи температури і сатурації пацієнтів сімейного лікаря. 1.1 Аналітичний огляд існуючих рішень. 1.2 Аналіз технічного завдання. 1.3 Опис і обґрунтування вибору елементної бази. Висновок до першого розділу. 2. Проектування інформаційної системи дистанційного моніторингу температури і сатурації пацієнтів сімейного лікаря. 2.1 Розробка функціональної схеми пристрою. 2.2 Розробка веб серверу. 2.3 Розробка алгоритму системи. 2.4 Написання текстів програми. 2.5 Розробка інструкції з експлуатації електронного пристрою. 2.5 Розробка методики перевірки, функціонування (контролю, випробування) електронного пристрою. 2.6 Висновок до другого розділу. 3. Безпека життєдіяльності, основи охорони праці. 3.1 Долікарська допомога при ураженні електричним струмом. 3.2 Розробка конкретних заходів щодо боротьби із статичною електрикою. 3.3 Особливості заходів електробезпеки на підприємствах Висновок до третього розділу. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)



## АНОТАЦІЯ

Розробка інформаційної системи дистанційного моніторингу температури і сатурації пацієнтів сімейного лікаря // Кваліфікаційна робота освітнього рівня «Бакалавр» // Ткачик Денис Олегович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СТс-41 // Тернопіль, 2023 // С. 62 , рис. 25, табл. 6, додат. 1, бібліогр. 30.

**Ключові слова:** бази даних, інформаційна система, Arduino, веб сервери.

Кваліфікаційна робота присвячена розробці інформаційної системи із використанням сервера ThingSpeak.

Мета даної роботи полягає в створенні інформаційної системи дистанційного моніторингу температури і сатурації пацієнтів сімейного лікаря.

В першому розділі кваліфікаційної роботи здійснено постановку задачі та формування вимог до інформаційної системи температури і сатурації пацієнтів сімейного лікаря, оглянуто існуючі рішення, описано і обґрунтовано вибір елементної бази.

В другому розділі кваліфікаційної роботи зроблено проектування інформаційної системи, проведено розробку функціональної схеми пристрою, веб серверу, алгоритму системи, написано тексти програми, а також розроблено інструкцію з експлуатації.

Об'єктом дослідження є інформаційна система моніторингу пацієнтів сімейного лікаря.

Предметом дослідження є методи і засоби розробки приладів на базі платформи Arduino .

## ANNOTATION

Development of an information system for remote monitoring of temperature and saturation of patients of a family doctor // Qualification work of educational level "Bachelor" / Tkachyk Denys // Ternopil Ivan Pul'uj National Technical University, Department of Computer Information Systems and Software Engineering, Department of Computer Science,, group STs-41 // Ternopil, 2023 // P. 62, fig. 25, Table 6, Annexes. 1, References. 30.

Keywords: databases, information system, Arduino, web servers.

The qualification work is devoted to the development of an information system using the ThingSpeak server.

The purpose of this work is to create an information system for remote monitoring of temperature and saturation of patients of a family doctor.

In the first chapter of the qualification work, the task is set and the requirements for the information system of temperature and saturation of patients of a family doctor are formed, existing solutions are analyzed, the choice of the element base is described and justified

The second section of the qualification work describes the design of the information system, the development of the functional diagram of the device, the web server, the system algorithm, the program texts, and the development of the operating instructions.

The object of research is an information system for monitoring patients of a family doctor.

The subject of research is the means and methods of developing devices based on the Arduino platform.

## ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

IOS (англ. Iphone Operating System) – власницька мобільна операційна система від Apple.

SQL (англ. Structured Query Language) – структурована мова запитів.

БД – база даних.

ПК – персональний комп'ютер.

ТЗ – технічне завдання.

## ЗМІСТ

ВСТУП.....	8
<b>ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.</b>	
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ТА ФОРМУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ ТЕМПЕРАТУРИ І САТУРАЦІЇ ПАЦІЄНТІВ СІМЕЙНОГО ЛІКАРЯ.....	8
1.1 Аналітичний огляд існуючих рішень.....	8
1.2 Аналіз технічного завдання.....	5
	<b>Ошибка! Закладка не определена.</b>
1.3 Опис і обґрунтування вибору елементної бази.....	6
	<b>Ошибка! Закладка не определена.</b>
1.4 Висновок до першого розділу.....	27
РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДИСТАНЦІЙНОГО МОНІТОРИНГУ ТЕМПЕРАТУРИ І САТУРАЦІЇ ПАЦІЄНТІВ СІМЕЙНОГО ЛІКАРЯ.....	28
	<b>ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА...</b>
2.1 Розробка функціональної схеми пристрою.....	28
2.2 Розробка веб серверу.....	29
2.3 Розробка алгоритму системи.....	36
	<b>Ошибка! Закладка не определена.</b>
2.4 Написання текстів програми.....	38
2.5 Розробка інструкції з експлуатації електронного пристрою.....	39
	<b>Закладка не определена.</b>
2.6 Розробка методики перевірки, функціонування (контролю, випробування) електронного пристрою.....	40
2.7 Висновок до другого розділу.....	41

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ.....	42
3.1 Долікарська допомога при ураженні електричним струмом.....	42
3.2 Розробка конкретних заходів щодо боротьби із статичною електрикою.....	43
3.3 Особливості заходів електробезпеки на підприємствах. <b>Ошибка!</b>	
<b>Закладка не определена.....</b>	<b>44</b>
<b>ВИСНОВКИ</b> <b>ОШИБКА!</b>	
<b>ЗАКЛАДКА</b>	
<b>НЕ</b>	
<b>ОПРЕДЕЛЕНА.....</b>	<b>46</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ.....</b>	<b>47</b>
<b>ДОДАТКИ.....</b>	<b>51</b>



## ВСТУП

**Актуальність теми.** Визначення показників функції зовнішнього дихання та контроль процесу газообміну є важливими в медицині. Цей процес включає легеневу вентиляцію, транспорт газів кров'ю, обмін газів в тканинах та клітинне дихання. Оцінка параметрів фізіологічних систем, що здійснюють ці функції, залежно від виду клінічного моніторингу, має різну діагностичну цінність.

Одним з важливих показників є температура тіла, саме оцінка функцій дихальної системи та вимірювання температури тіла є важливими елементами медичного моніторингу та діагностики, що дозволяють оцінити стан пацієнта та прийняти відповідні медичні заходи.

**Мета і задачі дослідження.** Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка інформаційної системи дистанційного моніторингу температури і сатурації пацієнтів сімейного лікаря, яка полегшить роботу лікарям і дозволить слідкувати за показниками пацієнтів з любої точки світу.

Для успішного імплементації поставленої мети, слід виконати наступні задачі:

- провести аналіз технічного завдання проекту;
- спроектувати функціональну інформаційну систему згідно до поставлених вимог;
- провести повне тестування усіх функціональних можливостей інформаційної системи.

**Практичне значення одержаних результатів.** Спроектowana та розроблена інформаційна система буде ергономічною у використанні та дозволить полегшити роботу лікарів.

# РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ТА ФОРМУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

## 1.1 Аналітичний огляд існуючих рішень

Існують різноманітні види пульсоксиметрії, які варіюються за розташуванням датчиків, можливістю пересування приладу, віком пацієнтів та місцем розташування джерела світла та датчика.

1. За розташуванням датчиків відносно крові пацієнта:

– інвазивний: передбачає введення датчиків прямо в кровоносні судини або тканини;

– неінвазивний: датчики розташовуються зовнішньо і не потребують введення в організм.

2. За можливістю пересувати прилад:

– стаціонарний: розрахований на використання в стаціонарних умовах, наприклад, у медичних закладах;

– переносний (мобільний): здатний бути перенесеним і використовуватися в рухомих умовах, має компактний розмір і може працювати на батареях.

3. За віком пацієнтів:

– дорослий: призначений для вимірювання пульсу та насиченості киснем у дорослих пацієнтів;

– дитячий: спеціально розроблений для використання у дітей, з урахуванням їхніх особливостей та розмірів.

4. За місцем розташування джерела світла та датчика:

– на палець: датчик розташовується на пальці, зазвичай на пальці руки, і використовується для непрямого вимірювання;

– на вухо: датчик розміщений на вусі і забезпечує пряме вимірювання пульсу та насиченості киснем;

– на руку, ногу: датчик може бути розташований на руці або нозі, і використовується для непрямого вимірювання.

Неінвазивна пульсоксиметрія набула широкого розповсюдження в медицині завдяки своїй простоті, швидкості і зручності у використанні.

Виконуючи пульсоксиметрію, важливо дотримуватись технічних вимог і медичних стандартів, а також правил асептики та антисептики. Особливо важливо, щоб датчики були одноразовими або використовувалися для різних пацієнтів або постраждалих з дотриманням необхідної стерилізації.

За принципом дії пульсоксиметри можуть бути:

– фотометричні ("пропускаючі", "трансмисивні"): ці типи пульсоксиметрів використовують світло для просвітлення тканин та вимірювання поглинання або прохідності світла через них. Датчик розміщується на пальці або іншій частині тіла, і світло проходить через тканини для вимірювання пульсу та насиченості киснем.

– "відбиваючі": ці типи пульсоксиметрів використовують принцип відбиття світла від тканин. Датчик розміщується на шкірі, і світло відбивається від тканин і захоплюється датчиком для вимірювання пульсу та насиченості киснем.

Важливо враховувати, що правильне розміщення датчика на тілі пацієнта та дотримання інших технічних і медичних вимог є необхідними для отримання точних і надійних результатів вимірювання.

Неінвазивна відбиваюча пульсоксиметрія, хоч і має свої обмеження, є менш поширеною через свою технічну складність та вартість порівняно з іншими методами. На відміну від неінвазивної пропускаючої пульсоксиметрії, де світло пропускається через тканини, у відбиваючій методі світло відбивається від тканин, і це може створювати технічні труднощі.

Інвазивна пульсоксиметрія, яка включає інвазивний моніторинг пацієнта, зазвичай здійснюється шляхом введення датчика в судинне русло. Цей метод дозволяє безпосередньо аналізувати газовий склад крові, а також вимірювати і реєструвати інші показники, такі як артеріальний тиск, системний судинний опір, серцевий викид, серцевий індекс та ударний об'єм. Інвазивна пульсоксиметрія застосовується в більш складних клінічних умовах, наприклад, в інтенсивній терапії.

У непрямій фотометричній пульсоксиметрії, світло, яке проходить через тканини, аналізується для визначення насиченості крові киснем. Вона не вимагає безпосереднього вимірювання газового складу крові, а заснована на фізичних властивостях крові, зокрема на здатності крові пропускати світло. Зміна кольору крові відбивається фотодетектором, що дозволяє оцінити насиченість крові киснем

Периферійний датчик пристрою зазвичай кріпиться на палець, найчастіше на вказівний, обтискаючи його з двох сторін. В інших випадках, наприклад, при використанні на зовнішньому вусі, необхідний додатковий датчик з двох сторін.

У портативних пристроях датчик і сам пристрій розташовані в одному корпусі, і на невеликому екрані відображаються показники (доки є заряд батареї для автономної роботи). У стаціонарних пристроях датчик після фіксації на пацієнті потрібно підключити за допомогою спеціального провідного з'єднання та налаштувати режим роботи.

Мобільні та особливо домашні мобільні пульсоксиметри мають деякий діапазон похибок, який залежить від частоти серцевих скорочень та рівня заряду батареї, що живить пристрій.

Вушний датчик швидше виявляє зміни у порівнянні із датчиком на пальці [1].

Серед пристроїв вимірювання сатурації крові можна виділити наступні:

1. Пульсоксиметр Gamma Oxy Scan. Цей портативний пристрій призначений для зручного та неінвазивного вимірювання рівня сатурації киснем в капілярній крові, моніторингу серцевого пульсу і індексу перфузії. Він забезпечує надійний затиск, не спричиняючи незручностей під час використання. Цей високоякісний пристрій ідеально підходить для осіб будь-якого віку та розміру пальця.

Завдяки неінвазивному методу вимірювання, пристрій здатний точно визначати рівень насичення крові киснем ( $SpO_2$ ) та частоту пульсу (PR), просто застосовуючи його на палець. Це особливо корисно для осіб з хронічними захворюваннями дихальної та серцево-судинної систем, а також для пацієнтів, які отримують кисневу терапію або займаються спортом.

Пульсоксиметр Gamma Oxy Scan забезпечує компактні розміри, що дозволяє його зручно носити з собою та використовувати в будь-який час. Він знадобиться багатьом людям з респіраторними або легневими проблемами, серцево-судинними захворюваннями, а також може бути використаний для профілактичних цілей.

Пристрій Gamma Oxy Scan широко застосовується в лікарнях, відділеннях швидкої допомоги, а також може бути використаний в домашніх умовах, кабінетах лікарів, спортивних центрах, реабілітаційно-оздоровчих центрах і багатьох інших місцях. Він простий у використанні і працює від батарейок, що робить його зручним та доступним для використання в будь-який час і місці.



Рисунок 1.1 – Пульсоксиметр Gamma Oxy Scan

Таблиця 1.1 – Технічні характеристики Пульсоксиметр Gamma Oxy Scan

Діапазон вимірювань SpO <sub>2</sub>	0% ~ 100%
Точність	70% ~ 100%: ± 2%
PR Діапазон вимірювань	30 ~ 250 уд / хв
Точність	± 2 уд / хв або ± 2%
Крок (SpO <sub>2</sub> )	1%
Крок (PR)	1 уд / хв
Робочий струм	≤25 мА
Живлення	2 батарейки типу ААА
Розмір	57 x 34 x 31 мм

Вартість пристрою: 600 грн. Початковий етап використання пульсоксиметра Gamma Oxy Scan, полягає у встановленні батарейок, вставивши їх у відповідні отвори. Потім помістити палець у спеціальний отвір пристрою і надійно прикріпити його. Натиснувши кнопку управління, пульсоксиметр перейде в робочий режим. Завдяки передовим технологіям, які використовуються у пульсоксиметрі Gamma Oxy Scan, результати вимірювань відобразяться на екрані дуже швидко - менше ніж за 5 секунд. Горизонтальний дизайн екрану полегшує спостереження за відображеними значеннями і робить його більш зручним для користувача. Якщо палець буде вийнятий з пристрою протягом 10-15 секунд, пульсоксиметр автоматично перейде в режим очікування або сну.

#### Особливості користування:

– при вимірюванні показників кисню у дітей, пульсоксиметр Gamma Oxy Scan зазвичай дає точні результати, але може бути деяка похибка в вимірюванні пульсу;

– для чоловіків рекомендується використовувати мізинець або безіменний палець для отримання більш точних показників. Використання інших пальців може призвести до похибки в вимірюваннях;

– у жінок, які мають нанесений гель-лак на нігтях, пульсоксиметр може давати невірні показники. Рекомендується перед вимірюванням зняти гель-лак з пальців для отримання точних результатів;

– важливо мати на увазі, що якщо у споживача холодні руки, то пульсоксиметр може показувати неточні показники або ж не надавати жодних значень. Рекомендується нагріти руки перед вимірюванням, наприклад, тримаючи їх під теплою водою або обтираючи рушником.

#### Особливості пульсоксиметра Gamma Oxy Scan включають:

– висока точність вимірювання частоти пульсу (PR) та рівня кисню в крові (SpO<sub>2</sub>), що дозволяє отримувати достовірні результати;

– інформація про значення SpO<sub>2</sub> і частоти пульсу зручно відображається на дисплеї пристрою;

- пульсоксиметр має низький рівень енергоспоживання, що дозволяє економно використовувати батареї та продовжувати тривалість його роботи;
- для збереження заряду батареї, екран автоматично вимикається на протязі 15 секунд бездіяльності;
- компактні розміри пульсоксиметра роблять його зручним для перенесення та використання в будь-який час.

2. Пульсоксиметр Gamma Oxy Scan. Пульсоксиметр є потужним і універсальним пристроєм для вимірювання пульсу і кисневого насичення крові. Він широко застосовується в спостереженні за пацієнтами зі складними медичними станами. Пристрій працює за допомогою неінвазивного датчика, який надівається на палець. Завдяки передовій технології вимірювання, результати отримуються швидко і ефективно всього за кілька секунд.

Пульсоксиметр знаходить широке застосування у багатьох галузях медицини. Він є незамінним при оцінці функції дихальної системи та діагностиці різних захворювань, таких як туберкульоз і саркоїдоз. Також він використовується в анестезіології для контролю пульсу і насичення крові під час операцій. Пульсоксиметр знайшов своє застосування і в лікуванні хронічної обструктивної хвороби легень, де вимірювання пульсу і кисневого насичення допомагають у контролі стану хворого. Застосування пульсоксиметра дозволяє забезпечити швидку і надійну інформацію про стан пацієнта, сприяє ранньому виявленню проблем і допомагає медичному персоналу прийняти належні рішення щодо діагностики і лікування.

На рисунку 1.2 зображено зовнішній вигляд пульсоксиметра.



Рисунок 1.2 – Пульсоксиметр Gamma Oxy Scan

Пульсоксиметри G1B, як вимірювачі пульсу, мають додаткові функціональні можливості, які підвищують їх зручність та користувацьку цінність. Зокрема, вони оснащені звуковим сигналом, який відображає рівень сатурації крові киснем, що дозволяє отримувати аудіальний звуковий сигнал, коли значення спадає або підвищується поза задані межі. Це корисно для контролю рівня насиченості крові та своєчасного сповіщення про можливі проблеми.

Крім того, пульсоксиметри G1B мають систему тривоги з налаштуванням. Це означає, що користувач може налаштувати певні параметри тривоги, такі як мінімальний або максимальний рівень сатурації киснем, при яких спрацюватиме тривога. Це дозволяє контролювати стан пацієнта та отримувати сповіщення у разі виникнення відхилень від норми.

Незважаючи на те, що пульсоксиметри можуть бути чутливими до яскравого зовнішнього світла, тремтіння і рухів, вони залишаються найточнішими і практичними засобами вимірювання рівня насиченості крові киснем порівняно з альтернативними методами. Вони забезпечують швидкі, надійні та неінвазивні вимірювання, що робить їх цінними інструментами для медичного спостереження та контролю.

Таблиця 1.2 – Технічні характеристики пульсоксиметр Gamma Oxy Scan

Пацієнт	новонароджений, дитячий, дорослий
Діапазон вимірювання	0-100%
Дозвіл	1%
Точність виміру	± 2удара в хвилину
Параметри	SpO2. PR
Діапазон пульсу	30-250 ударів на хвилину
Дозвіл	1 удар в хвилину
Точність вимірювань пульсу	± 2удара в хвилину
Температура при експлуатації	0 ° C ~ 50 ° C
Температура при зберіганні	-40 ° C ~ 55 ° C
Вологість при експлуатації	0% ~ 93% (без конденсації)
Вологість при зберіганні	0% ~ 95%



Вартість пристрою: 8 990 грн. Пульсоксиметр G1B може працювати як від акумулятора, так і від зарядної підставки. Історія вимірювань зберігається протягом 24 годин і може бути перенесена на комп'ютер за допомогою прилагодженого з'єднувального кабелю. Вимірювані значення відображаються на яскравому 1,6-дюймовому кольоровому екрані у формі числових значень і графіків трендів. Для встановлення конкретних межів значень можна налаштувати звуковий сигнал, який буде активуватися при їх досягненні [3].

## 1.2 Формування вимог до інформаційної системи

Система призначена для автоматичного моніторингу температури і сатурації пацієнтів сімейного лікаря. Для моніторингу показників температури і сатурації використовуються такі елементи: плата Arduino, датчик температури, вай фай модуль, та пульсоксиметр .

Система дистанційного моніторингу температури і сатурації містить:

- датчик температури - призначений для зчитування температури об'єкту;
- вай фай модуль - призначений для передачі даних з датчиків на сервер;
- пульсоксиметр - для зчитування показників пульсу і кисню в крові;
- веб сервер – для забезпечення дистанційного зв'язку з сімейним лікарем.

Структура системи на МК зображена на рисунку 1.3.

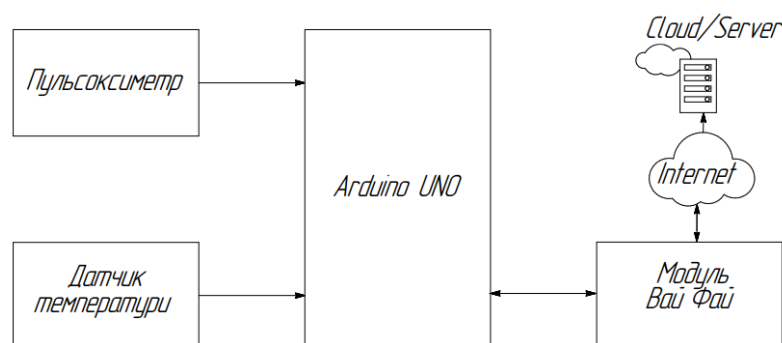


Рисунок 1.3 – Структурна схема пристрою

Для роботи з пристроєм, необхідно прикласти палець до датчика сатурації після чого він почне зчитувати дані такі як пульс та насиченість крові киснем.

Далі потрібно поставити палець на датчик температури щоб він виміряв температуру тіло. Після того як дані будуть зчитані з датчиків вони надсилаються на сервер де вони заносяться в таблицю.

Структурна схема є початковою моделлю електронного пристрою, яка відображає загальний принцип його роботи. Основною перевагою структурної схеми є можливість швидко отримати уявлення про склад, структуру та функції пристрою, які він виконує [4].

### **1.3 Опис і обґрунтування вибору елементної бази**

Система створена на основі двох плат Arduino Uno. У обидвох системах використовується 8-розрядний мікроконтролер ATMEL - Atmega328P в якості мікропроцесора.

Arduino Uno базується на цьому контролері. Відповідна платформа містить 14 цифрових входів/виходів (6 з яких можуть використовуватися як ШІМ-виходи), 6 аналогових входів, 16 МГц кварцовий генератор, USB-роз'єм, силовий роз'єм, ICSP-роз'єм та кнопку перезавантаження. Для функціонування необхідно підключити платформний сервіс до особистого комп'ютера за допомогою USB-кабелю або ж блоку живлення можна забезпечити за допомогою адаптера AC/DC або батареї. На рисунку 3.1 зображено умовне графічне позначення Arduino Uno [5].

На відміну від попередніх плат, які використовували FTDI USB-мікроконтролер для зв'язку по USB, Arduino Uno використовує мікроконтролер ATmega16U2 [6].

Програмне забезпечення для розробки на платформі Arduino - це кроссплатформний додаток Java, який включає компілятор, редактор коду і модуль для передачі прошивки на плату. Це інтегроване середовище розробки (ICP), яке надає зручний інтерфейс для написання, збирання і завантаження програмного коду на плату Arduino. Завдяки цьому середовищу розробки розробники можуть створювати програми для Arduino на різних операційних системах, включаючи Windows, macOS і Linux.

Середовище розробки Arduino засноване на мові програмування Processing і призначене для новачків, які не мають значного досвіду у розробці програмного забезпечення. Строго кажучи, це спеціальна версія мови C++, яка доповнена деякими бібліотеками. Програми, написані в середовищі розробки, спочатку обробляються процесором, а потім компілюються за допомогою AVR-GCC - компілятора для мікроконтролерів AVR, який використовується в платі Arduino. Це дозволяє розробникам створювати програми для Arduino, використовуючи знайомий синтаксис та функціональні можливості мови C++ з додатковими можливостями, наданими бібліотеками Arduino.

На рисунку 1.4 зображено зовнішній вигляд плати Arduino Uno. Виводи живлення Arduino Uno:  $V_{cc}$ , GND; Цифрові входи / виходи: 1,2,3,4-16; Аналогові входи / виходи: 19-24; Виводи для взаємодії з LCD-екраном: 27,28.

На рисунку 1.4 зображено зовнішній вигляд плати Arduino UNOї

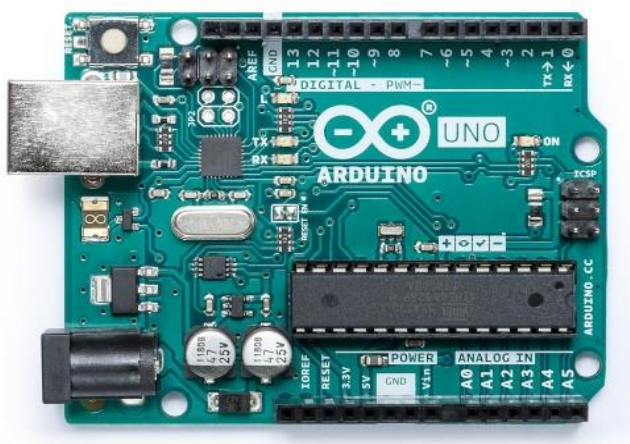


Рисунок 1.4 – Зовнішній вигляд плати Arduino Uno

Платформа Arduino може працювати при зовнішньому живленні в діапазоні від 6 В до 20 В [5]. Проте, варто зазначити, що при напрузі живлення нижче 7 В вивід 5V може видавати менше 5 В, що може призвести до нестабільної роботи платформи. З іншого боку, використання напруги вище 12 В може спричинити перегрів регулятора та пошкодження плати [8]. Рекомендований діапазон живлення для Arduino Uno знаходиться від 7 В до 12 В.

Arduino Uno має 14 цифрових виходів, кожен з яких може бути налаштований як вхід або вихід за допомогою функцій `pinMode()`, `digitalWrite()` і

digitalRead(). Виводи працюють при напрузі 5 В. Кожен вивід має завантажувальний резистор зі значенням 20-50 кОм (за замовчуванням відключений) і може пропускати до 40 мА.

Фізичні розміри друкованої плати Arduino Uno складають 6.9 см в довжину та 5.3 см в ширину.

Таблиця 1.3 – Характеристики Arduino UNO

Мікроконтролер	Atmega328P
Робоча напруга	5 В
Вхідна напруга	7-12 В
Вхідна напруга	6-20 В
Цифрові Входи / Виходи	14
Аналогові входи	6
Постійний струм через вхід / вихід	40 мА
Постійний струм для виведення 3.3В	50 мА [22]
Флеш-пам'ять	32 Кб (ATmega328P)
ОЗУ	2 Кб(ATmega328P)
EEPROM	1 Кб (ATmega328P)
Тактова частота	16 МГц [10]

Роз'єм USB і силовий роз'єм виходять за межі основних розмірів платформи Arduino Uno. Це дозволяє підключати плату до комп'ютера через USB-порт для програмування та передачі даних, а також живити її від зовнішнього джерела живлення.

На платі Arduino Uno також є чотири отвори, які дозволяють закріпити плату на поверхні або встановити її у корпусі апаратного пристрою.

Щодо відстані між цифровими виводами, між 7-м та 8-м виводами вона становить 0.4 см, тоді як між іншими виводами вона складає 0.25 см. Це може бути корисною інформацією при розміщенні компонентів чи підключенні дротів до виводів плати.

Технічний опис контролера Atmega328P [11]:

- висока продуктивність, низька потужність;
- прогресивна PDIP архітектура;

- 131 Потужна Інструкція;
- 32 x 8 загального призначення робочі регістри;
- повністю статичні операції;
- до 20 MISP пропускна здатність у 20 МГц;
- високо витривалі енергонезалежні сегменти пам'яті;
- 4/8/16/32К байт система самопрограмованої флеш пам'яті (ATmega48P/88P/168P/328P);
- 256/512/512/1К байт EEPROM (ATmega48P/88P/168P/328P);
- 512/1К/1К/2К байт внутрішня SRAM (ATmega48P/88P/168P/328P);
- цикли запису / стирання: 10000 Flash/100,000 EEPROM;
- зберігання даних: 20 років при 85 ° C/100 років при 25 ° C (1);
- необов'язковий розділ завантажувального коду з незалежними бітами блокування;
- периферійні особливості;
- два 8-розрядних таймер/лічильники з роздільним прескалером і режимом порівняння;
- один 16-розрядний таймер/лічильник з окремим попереднім дільником, режимом порівняння;

#### Режими:

- шести канальний ШІМ;
- 8-канальний 10-бітний АЦП в TQFP і QFN / МФ пакет;
- 6-канальний 10-бітний АЦП в PDIP Пакеті;
- програмований послідовний USART;
- master / Slave SPI послідовний інтерфейс;
- програмований сторожовий таймер з окремим вбудованим генератором;
- вбудований аналоговий компаратор;
- спеціальні особливості мікроконтролера:
- скидання по включенню живлення;
- внутрішній калібрований генератор;
- зовнішні і внутрішні джерела переривань;

– режим сну: скорочення АЦП, енергозбереження, при відключенні живлення, в режимі очікування.

I / O і пакети: 23 програмованих входів / виходів.

Робоча напруга: 1,8 – 5,5 для ATmega48P/88P/168PV; 2,7 – 5,5 для ATmega48P/88P/168P; 1,8 – 5,5 для ATMEGA328P;

Діапазон робочих температур: 40 ° С до 85 ° С; активний режим: 0,3 мА; при відключенні живлення режим: 0,1; режим економії енергії: 0,8 мкА (у тому числі 32 кГц RTC) [12].

На рисунку 1.5 зображено позначення виводів контролера ATMEGA328P.

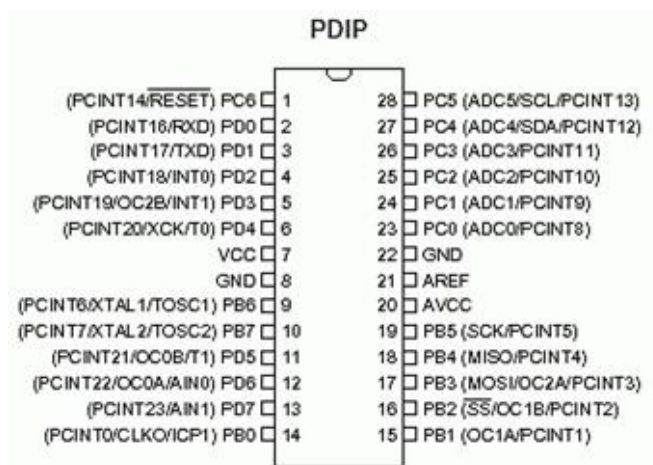


Рисунок 1.5 – Позначення виводів контролера

Датчик температури. LM35 є прецизійним інтегральним датчиком температури, який має широкий діапазон вимірювання температури, високу точність і калібрований вихід по напрузі. Ці характеристики зробили його досить популярним серед користувачів.

Датчик LM35 належить до серії прецизійних інтегральних датчиків температури, де вихідна напруга пропорційна температурі за шкалою Цельсія. Це має перевагу порівняно з датчиками, що використовують вихідну напругу за шкалою Кельвіна, оскільки не потрібно виконувати додаткові обчислення для перетворення вихідної напруги на шкалу Цельсія. LM35 забезпечує вимір температури з точністю  $\pm 0.25$  °С в кімнатних умовах і з точністю  $\pm 0.75$  °С в повному діапазоні робочих температур від -55 °С до +150 °С, без необхідності зовнішньої калібрування або налаштування вихідної напруги. Низька ціна



- працюють в широкому діапазоні напруги живлення від 4 до 30 В;
- споживають струм менше 60 мкА;
- низький рівень власного нагріву - всього 0.08 °С;
- нелінійність датчиків становить лише  $\pm 0.25$  °С;
- мають низький вихідний опір - 0.1 Ом при струмі навантаження 1 мА.

Ці особливості роблять датчики LM35 досить привабливими для багатьох застосувань, де потрібна точна вимірювання температури в широкому діапазоні. Позначення датчика LM35 на схемі зображене на рисунку 1.7.

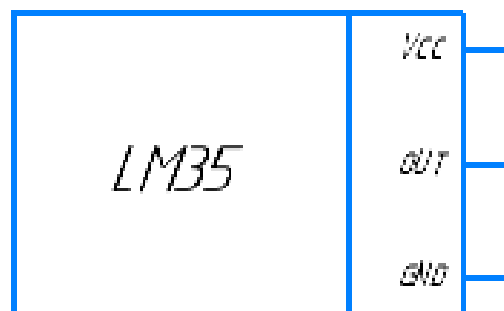


Рисунок 1.7 – Зображення датчика LM35 на схемі

Даний датчик є аналоговим, що означає, що на його виході ми отримуємо безперервну змінну напругу в діапазоні від 0 до 5 вольт. Тому, для підключення датчика LM35 до платформи Arduino, необхідно підключити його до аналогових входів A0-A5 за схемою, зображеної далі. Після складання схеми, потрібно завантажити скетч на платформу Arduino, який дозволить зчитувати значення з аналогових датчиків і виводити дані через апаратний послідовний порт.

Варто зазначити, що перед підключенням датчика до Arduino важливо переконатись, що використовується необхідні резистори або підсилювачі, які забезпечують правильне зчитування з датчика і захищають вхід аналогового порту від можливого пошкодження через надмірну напругу.

Підключення датчика LM35 до Arduino (див. рис. 1.8).



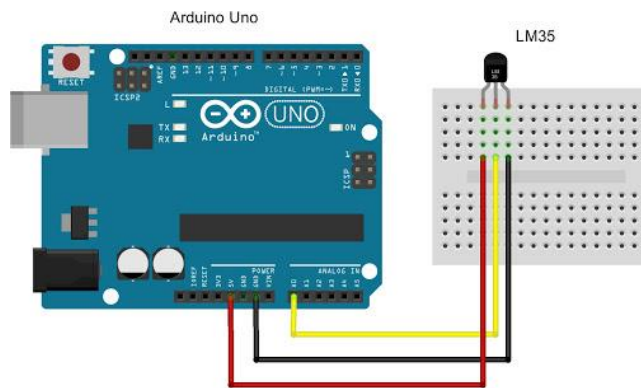


Рисунок 1.8 – Підключення датчика LM35 до Arduino

Модуль вай фай. Модуль призначений для надсилання і отримання даних через Wi-Fi (рис. 1.9). Стандартні налаштування модуля наведені у таблиці 1.4.

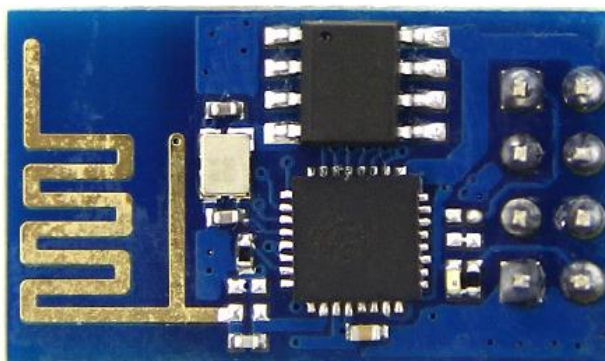


Рисунок 1.9 – Модуль Wi-Fi ESP8266 ESP-01

Таблиця 1.4 – Стандартні налаштування модуля

Швидкість передачі даних	100-460800 б/с.
Ім'я модуля	Wi-Fi ESP8266 ESP-01

Позначення виводів модулю Wi-FiESP8266 ESP-01 наведено в таблиці 1.5.

Таблиця 1.5 – Позначення виводів модулю Wi-FiESP8266 ESP-01

Wi-Fi ESP8266 ESP-01	ArduinoUno
GND	Підключення на +5V
Vcc	Підключення до землі -5V
TX	Підключаємо до RX
RX	Підключаємо до TX

Основні параметри модуля вай фай ESP8266 ESP-01 наведено в таблиці 1.6.

Таблиця 1.6 – Основні параметри модуля Wi-Fi ESP8266 ESP-01

Швидкість передачі даних	100-460800 б/с.
Вихідна напруга логічної одиниці	3v
Максимальний струм споживання	220mA
Напруга живлення	3-3.6v
Максимальний вхід напруги лог.од	3.6v

Позначення модулю Wi-Fi ESP8266 на схемі зображене на рисунку 1.10.

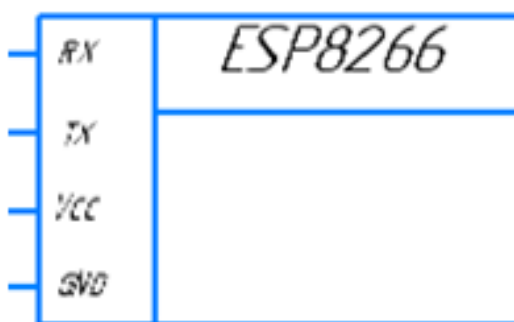


Рисунок 1.10 – Зображення модулю Wi-FiESP8266 на схемі

- Налаштовуємо частоту процесора "CPU Frequency: "80 MHz"", швидкість "Upload Speed: " 115200 "" і вибираємо "Порт".
- Завантажити скетч, який змусить ESP8266 включити світлодіод.
- Якщо все правильно підключили і налаштували, світлодіод на ESP8266 почне світити з інтервалом в 1 секунду.

Підключення Wi-Fi ESP8266 ESP-01 до Arduino наведено на Рисунку1.11.

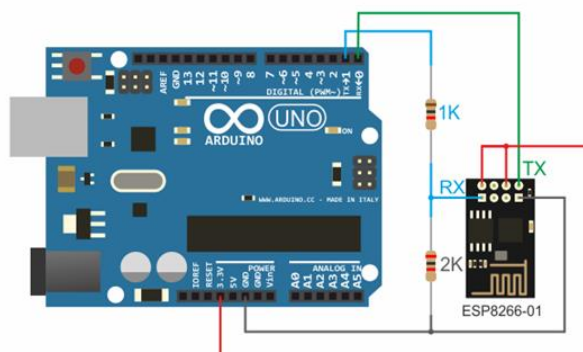


Рисунок 1.11 – Підключення Wi-Fi ESP8266 ESP-01 до Arduino

- Встановити з сайту [arduino.cc](http://arduino.cc) програму IDE Arduino.
- Далі необхідно встановити ESP плату в IDE Arduino, для цього запускаємо програму IDE Arduino, відкриваємо: Файл -> Налаштування.
- В новому відкритому вікні, в полі "Додаткові посилання для Менеджера плат": додаємо необхідне посилання для цього модулю.
- Натиснути кнопку "ОК", далі відкриваємо: "Інструменти -> Плати: -> Менеджер плат ..".
- В відкритому вікні, слід знайти "esp8266 by ESP8266 Community" і натискаємо "Встановити". Установка кілька хвилин, з'явиться напис "Installed".
- Прейти в "Інструменти -> Плати -> Generic ESP8266 Module".
- Підключити модуль ESP-01 до комп'ютера через спеціальний адаптер USB на чіпі CH340G.

Модуль пульсоксиметр. Є датчиком серцевого ритму MAX30100, який призначений для вимірювання частоти серцевих скорочень чи пульсової оксиметрії. Широко використовуваний у диверсифікованих медичних приладах. Має плату спроектовану на основі датчика MAX30100 та інших допоміжних компонентів, які сприяють його правильній роботі. Мікросхема MAX30100 спроектована та містить два світлодіоди (червоного і інфрачервоного), аналогового підсилювача, фотоприймача, цифрового процесора і інтерфейсного модуля.

Особливості даного датчика включають наявність низького рівня частот власних шумів і нівелювання зовнішнього засвічення. Він також має високу стійкість до вібрацій при вимірюванні показань, відносно стійку частоту дискретизації, і надійний процес вимірювання.

Управління мікросхемою MAX30100 здійснюється за допомогою програмних регістрів. Дані з датчика зберігаються в буфері FIFO. При вимірюванні пульсу використовуються канали червоного і інфрачервоного світіння. Датчик також може змінювати температуру залежно від вимірювань SpO<sub>2</sub>. Роздільна здатність датчика температури становить 0,0625 °C.

Технічні характеристики модуля: мікросхема - MAX30100; напруга живлення, В - 5 (внутрішній стабілізатор); струм в режимі вимірювання, мА - 1,2;

струм в режимі sleep, мкА - до 10; інтерфейс - I2C; максимальна частота інтерфейсу, кГц - 400; розміри модуля, мм - 8,5 x 14,4 x 3 [15].

На рисунку 1.12 зображений зовнішній вигляд модуля.



Рисунок 1.12 – Модуль MAX30100

Модуль підключається до платформи Arduino або іншим мікроконтролерним пристроїв за допомогою послідовного інтерфейсу I2C.

Призначення виходів: GND: "Земля"; RD: драйвер червоного світлодіода; IRD: драйвер ІК світлодіода; INT: переривання; SDA: лінія даних; SCL: лінія тактування; VIN: напруга живлення.

Для візуалізації параметрів датчика можна використовувати LCD дисплей. Підключається екран за допомогою висновків: SCL і SDA.

Для роботи з модулем в середовищі розробки ArduinoIDE необхідно завантажити бібліотеку, а потім самостійно її встановити.

Підключення Модуля MAX30100 до Arduino наведено на рисунку 1.13.

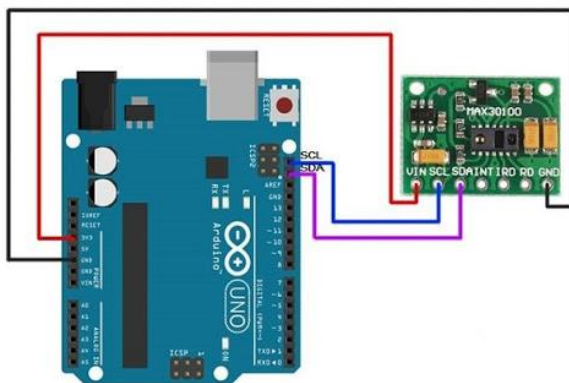


Рисунок 1.13 – Підключення Модуль MAX30100 до Arduino

Підключений до джерела живлення від зовнішнього джерела живлення, платформи Arduino або іншого мікроконтролерного пристрою. Напруга живлення модуля становить 5 В.

Позначення датчика MAX30100 на схемі зображене на рисунку 1.14.

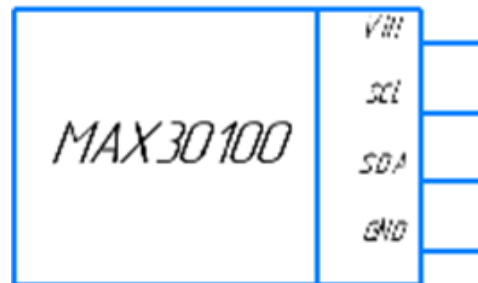


Рисунок 1.14 – Зображення датчика MAX30100 на схемі

#### 1.4 Висновок до першого розділу

В першому розділі був проведений аналітичний огляд існуючих рішень, розроблено технічний та робочий проект, а також проведено аналіз технічного завдання і обґрунтовано вибір елементної бази.

Аналітичний огляд існуючих рішень дозволив оцінити наявні на ринку системи моніторингу пацієнтів і визначити їх переваги та недоліки. Було виявлено, що деякі системи мають обмежені функціональні можливості, не забезпечують достатню безпеку даних або не забезпечують зручного інтерфейсу користувача. Враховуючи ці обмеження, було вирішено розробити власну інформаційну систему, яка відповідала би потребам сімейних лікарів та їх пацієнтів.

Під час розробки технічного та робочого проекту були визначені вимоги до системи, спроектована архітектура, розроблені базові модулі і функціональні можливості. Вирішені питання щодо збереження та безпеки медичних даних, забезпечення зручного інтерфейсу користувача, а також інтеграції з іншими системами, наприклад, Інтернетом речей (IoT) для отримання даних з медичних пристроїв. Аналіз ТЗ дозволив уточнити вимоги до системи, визначити потреби користувачів та їх очікування.

## РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1 Розробка і опис роботи функціональної схеми

Функціональна схема є ключовим проектним документом, який визначає функціональну структуру системи або пристрою, що проектується, а також окремих його модулів. Головна мета цієї схеми полягає у візуалізації детальної структури пристрою, включаючи його основні блоки, вузли, компоненти та зв'язки між ними. Функціональна схема дозволяє зрозуміти, як компоненти пристрою підключаються та взаємодіють між собою.

У функціональній схемі повинні бути чітко видно всі компоненти пристрою та їхні взаємозв'язки. Вона допомагає уявити, як різні частини пристрою працюють разом для досягнення поставленої мети. Завдяки функціональній схемі розробники можуть зрозуміти, які блоки чи модулі необхідні для реалізації певних функцій, як вони пов'язані між собою та які дані або сигнали передаються між ними.

Загальна структура функціональної схеми залежить від конкретного проекту, але вона має бути логічною, зрозумілою та відображати всі необхідні компоненти та їх взаємодію. Така схема є важливим інструментом для комунікації між розробниками, інженерами та іншими зацікавленими сторонами, що дозволяє узгодити розуміння системи та забезпечити її правильну реалізацію.

DM1 – датчик температури LM35 призначений для зчитування температури пацієнта;

DM2 – пульсоксиметр MAX30100 призначений для зчитування пульсу і насиченості крові киснем;

DM3 – мікропроцесор Arduino Uno, призначений для управління роботою системи;

DM4 – вай-фай модуль призначений для зв'язку пристрою з веб-сервером;

Функціональну схему зображено на рисунку 2.1.

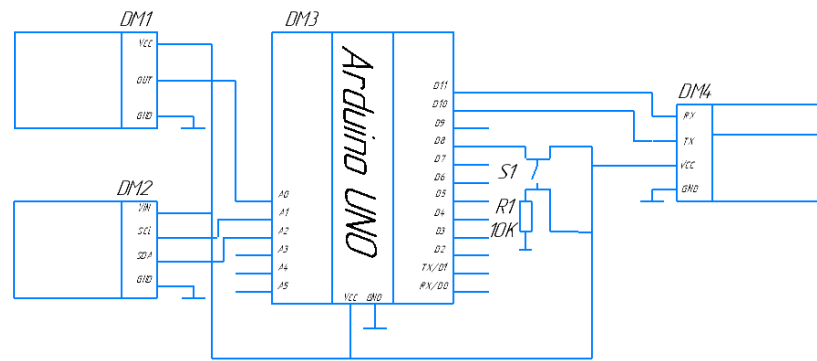


Рисунок 2.1 – Функціональна схема пристрою

## 2.2 Розробка веб-сервера

Сервіс ThingSpeak є дуже зручним для реалізації різних проєктів. За допомогою каналів і веб-сторінок, які забезпечуються даним сервісом, можна моніторити і управляти системою з будь-якої точки де є мережа інтернет. ThingSpeak збирає ('Collects') дані від датчиків, аналізує і візуалізує їх ('Analyze and Visualize') і виконує ('Acts') на підставі цього аналізу різні дії.

На рисунку 2.2 зображено зовнішній вигляд головної сторінки сервісу ThingSpeak.com.

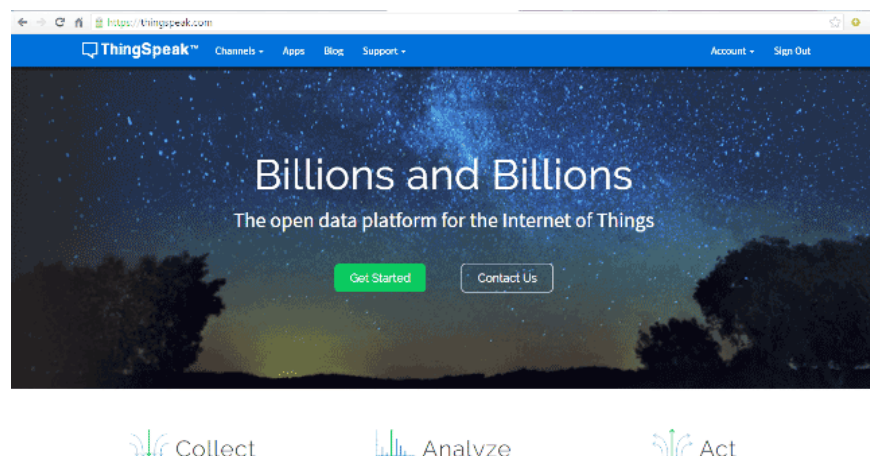


Рисунок 2.2 – Зовнішній вигляд головної сторінки сервісу ThingSpeak.com

У цьому проєкті потрібно використовувати ThingSpeak для моніторингу пульсу і температури пацієнта через мережу інтернет. Також будемо використовувати платформу IFTTT щоб з'єднати ThingSpeak з сервісами email /

message, щоб можна було передавати повідомлення тривоги коли пацієнт перебуває в критичному стані.

ThingSpeak - це платформа, спрямована на реалізацію проєктів, пов'язаних з "Інтернетом речей" (IoT). Вона надає зручні можливості для збору, обробки та візуалізації даних, зібраних з різних датчиків.

Основними функціями ThingSpeak є збір даних у реальному часі, обробка цих даних і їх графічна візуалізація. Завдяки відкритому API ThingSpeak, користувачі можуть легко відправляти, зберігати та отримувати доступ до своїх даних. Крім того, платформа надає різноманітні статистичні методи для обробки цих даних.

ThingSpeak є потужним інструментом для створення IoT-проєктів, де збираються дані з різних датчиків і використовуються для аналізу, моніторингу та прийняття рішень. Вона дозволяє зручно працювати з даними та створювати різноманітні застосунки на основі цих даних.

З використанням ThingSpeak ви можете легко створити власні IoT-проєкти, контролювати датчики, отримувати дані в реальному часі і виконувати аналіз для виявлення корисних залежностей або трендів. Ця платформа забезпечує широкі можливості для розробників і дослідників, які працюють у сфері "Інтернету речей".

Конфігурація сервісу ThingSpeak для запису даних пацієнта. Для початку роботи з сервісом треба створити аккаунт на сервісі ThingSpeak.com, авторизуватись в ньому і потім натиснути Get Started.

Тепер потрібно перейти в пункт меню 'Channels' (канали), де слід створити новий канал (New Channel).

Після цього відкриється форма для створення каналу, потрібно заповнити в ній ім'я каналу (Name) і його опис (Description). Також необхідно заповнити 'Saturation', 'Temperature' і 'Panic' в полях Field 1, Field 2 and Field 3. Також потрібно поставити галочку в чек-боксі (check box) 'Make Public' нижче в формі і збереження каналу.

Тепер потрібно використати додаток ThingHTTP щоб запускати аплет IFTTT для передачі даних в Google sheets і передачі email / sms. Додаток



ThingHTTP дозволяє здійснювати взаємозв'язок між пристроями, веб-сайтами та веб-сервісами не використовуючи протоколи зв'язку на рівні пристроїв. В ThingHTTP можна детально викласти дії, які можна ініціювати в додатку ThingSpeak, наприклад React.

Після чого буде створено новий канал і три графіка які мають виглядати так як зображено на рисунку 2.3

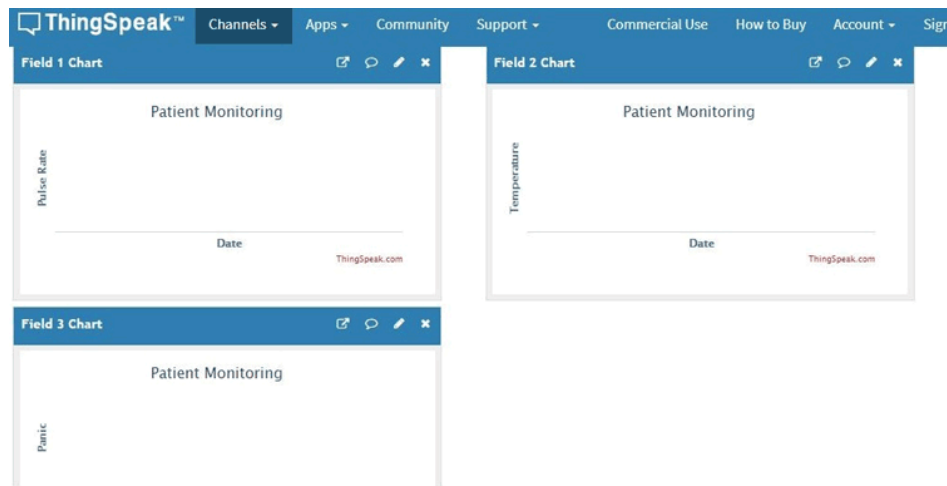


Рисунок 2.3 – Зовнішній вигляд графіків

Тепер потрібно використати додаток ThingHTTP щоб запускати аплет IFTTT для передачі даних в Google sheets і передачі email / sms. Додаток ThingHTTP дозволяє здійснювати взаємозв'язок між пристроями, веб-сайтами та веб-сервісами не використовуючи протоколи зв'язку на рівні пристроїв. В ThingHTTP можна детально викласти дії, які можна ініціювати в додатку ThingSpeak, наприклад React.

Щоб створити нову дію в ThingHTTP потрібно мати адресу URL для ініціювання дії. Цю адресу можна отримати в сервісі IFTTT.

Конфігурація сервісу IFTTT для передачі Mail / SMS

IFTTT (If This Then That) - це веб-сервіс, який надає можливість автоматизувати взаємодії між різними соціальними мережами, додатками та системами розумного будинку, шляхом встановлення простих правил. Його принцип роботи базується на умовному операторі "Якщо трапилося ситуація А, то виконай дію Б".

IFTTT дозволяє створювати зв'язки між різними платформами та сервісами, щоб автоматично реагувати на події і виконувати певні дії. Наприклад, ви можете налаштувати автоматичне включення світла, коли отримуєте новий лист в Gmail, або отримувати сповіщення на свій телефон, коли вашій авто запарковано в певному місці. Це лише кілька прикладів з безлічі можливостей, які надає IFTTT.

IFTTT працює на основі каналів і рецептів. Канали представляють різні платформи та сервіси, з якими можна взаємодіяти, наприклад, Facebook, Twitter, Gmail, Philips Hue тощо. Рецепти визначають умови і дії, які потрібно виконати, якщо відбувається певна подія.

IFTTT є потужним інструментом для автоматизації рутинних задач і покращення продуктивності. Він дозволяє зручно зв'язувати різні сервіси і додатки, щоб створювати персоналізовані сценарії, що працюють на основі вашого власного набору умов і дій.

На рисунку 2.4 зображено зовнішній вигляд діалогового вікна Documentation.



Тепер потрібно вибрати дію (trigger) "Receive a web request". ім'я події – точно в діалоговому вікні зображеному на рисунку 2.5. Натискаємо на Create Trigger.

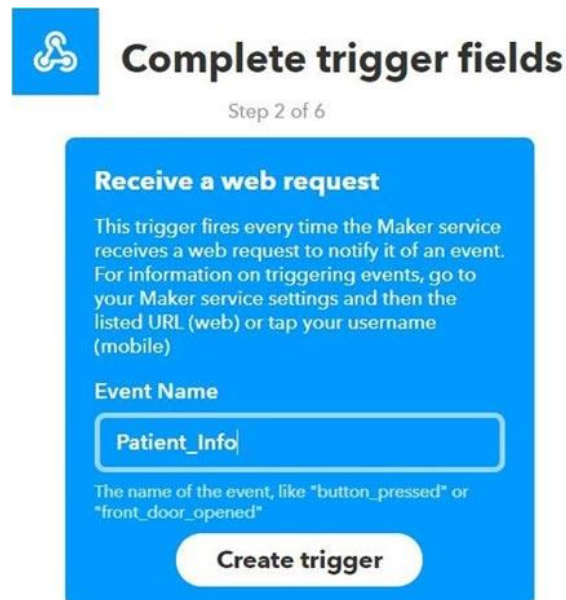


Рисунок 2.5 – Зовнішній вигляд діалогового вікна Complete trigger fields

Після цього необхідно створити Applet щоб з'єднати ThingHTTP з Google sheet і щоб передавати email / sms. Переходимо в ThingHTTP щоб завершити там створення нової події. Переходимо на вкладку My Applets. і клікаєм на New Applet. Натискаємо на "+ this", вводим в пошуку Webhooks і потім натискаємо на нього.

Тепер потрібно натиснути на "+ that" і ввести в пошуку Google Sheets і натисніть на нього. Переходимо на сторінку Add row to spreadsheet.

Аналогічним чином можна створити аплет для передачі email коли буде відбуватися подія Panic.

Тепер необхідно натиснути на "+ this", і вибрати Webhooks, потім в найменуванні події ввести "Panic". Переходимо по "+ that", запускаємо пошук Gmail і переходим на нього. І на сторінці потрібно вибрати Send an email.

У наступному вікні слід ввести адресу email, на який будуть приходити листи якщо пацієнту стане погано це може бути email сімейного лікаря або близького родича.

Тепер необхідно написати листа, що буде пересилатися, і клацаємо на create action. Перевіряємо створений аплет і натискаємо на finish.

Ввід ім'я в таблиці. У відформатованому рядку слід записати вираз для дати, часу, імені подій, значень пульсу та температур тіла, як показано на рисунку 2.6 [17].



Рисунок 2.6 – Параметри таблиці

ThingHTTP для з'єднання ThingSpeak з IFTTT. Натискаємо на New ThingHTTP. Даємо йому ім'я і копіюємо URL, який отримали в webhooks documentation.

У тілі (Body) створеного ThingHTTP потрібно написати інформацію яку має передаватися з додатком (аплету) IFTTT. Для цієї системи необхідно передавати значення сатурації і температури тіла.

Після заповнення цієї інформації потрібно натиснути на Save ThingHTTP.

Аналогічним чином потрібно зробити новий ThingHTTP для події "Panic". В строці для URL необхідно записати Panic замість Patient\_Info. Тіло (Body) залишається порожнім, а вся інша інформація та ж сама, як і в попередньому ThingHTTP.

Заповнимо поля які показано на наступному рисунку 2.7.

The image shows the configuration form for a new React in ThingSpeak. The fields are as follows:

- Name:** Patient\_Monitor
- API Key:** [Redacted]
- URL:** https://maker.ifttt.com/trigger/Patient\_Info/with/key/hwZoxX
- HTTP Auth Username:** [Empty]
- HTTP Auth Password:** [Empty]
- Method:** POST
- Content Type:** application/json
- HTTP Version:** 1.1
- Host:** [Empty]
- Headers:** [Table with one row: Name | [Empty]]

Рисунок 2.7 – Діалогове вікно ThingSpeak

Тепер слід створити React (реакцію) щоб ініціювати (запустити) URL. React працює з додатком ThingHTTP щоб виконувати певні дії при настанні певних умов на каналах

Щоб створити React треба клацнути на Apps -> React, а в ньому натиснути на New React (нова реакція, нове дію).

Далі даємо ім'я для React (дії). В якості типу умови (Condition type) вибираємо Numeric, а в якості частоти перевірок (Test Frequency) - Data Insertion.

Аналогічним чином необхідно змінити React (дія) на подію Panic як показано на рисунку 2.8.

Після цього слід вибрати умову (Condition) при настанні якої буде формуватися необхідний URL.

Вибираєм канал з меню, що випадає "If Channel".

Потім вибираєм у полі (field) "1 Saturation" (насиченість крові киснем) і створюєм умову "is greater than" - тобто більше ніж.

Тепер вибираєм ThingHTTP з меню дій. Потім виберіть пункт "Run action each time condition is met" (виконувати дію кожного разу коли виконається умова) і клацаєм на Save React (зберегти).

React Name	<input type="text" value="Panic"/>
Condition Type	<input type="text" value="Numeric"/>
Test Frequency	<input type="text" value="On Data Insertion"/>
Condition	<input type="text" value="If channel"/> <input type="text" value="Patient Monitoring (523997)"/>
	<input type="text" value="field"/> <input type="text" value="3 (Panic)"/>
	<input type="text" value="is equal to"/>
	<input type="text" value="1"/>
Action	<input type="text" value="ThingHTTP"/>
	<input type="text" value="then perform ThingHTTP"/> <input type="text" value="Panic"/>

Рисунок 2.8 – Приклад змін в події Panic

Після чого потрібно перейти в пункт меню "Run action each time condition is met" і натиснути на Save React [18].

### 2.3 Розробка алгоритму системи

Для роботи з WI-FI модулем використовується віртуальний UART (mySerial) - бібліотека "SoftwareSerial.h". Також потрібно підключити бібліотеки для роботи з датчиками температури і сатурації.

Алгоритм роботи складається з наступних функціональних модулів:

Ініціалізація змінних.

Налаштування виводів і швидкості обміну по uart і myserial.

Перевірка даних по myserial.

Якщо дані отримано, то здійснюється:

1.1 Опрацювання вхідних даних; 1.2. Зчитування даних з датчиків; 1.3. Передача даних по вай-фай модулю;

2. Опрацювання тривожної кнопки.

3. Якщо натиснута кнопка:

3.1 Надсилання сповіщення лікарю.

4. Повернення до пункту 3.

Алгоритм системи відображено в додатку А.

## 2.4 Написання текстів програми

У програмі потрібно буде використовувати спеціальну бібліотеку для роботи з датчиком сатурації. Також необхідно використовувати таймер для установки тимчасового інтервалу між зчитування даних.

Насамперед в програмі слід підключити всі використовувані бібліотеки. Для взаємодії з esp8266 буде використовуватися бібліотека послідовного зв'язку (software serial).

```
#include <SoftwareSerial.h>
```

```
#include "Timer.h"
```

```
#include <MAX30100_SpO2Calculator.h> //бібліотека для роботи з датчиком сатурації.
```

Тепер необхідно створити об'єкти таймера, послідовного порту і датчика сатурації щоб надалі їх можна було використовувати в програмі.

```
Timer t;
```

```
PulseSensorPlayground pulseSensor;
```

```
SoftwareSerial esp8266(10, 11);
```

Далі потрібно ініціалізувати можливість використання переривань низького рівня для підвищення точності вимірювань і задіяти DEBUG щоб показувати надходять команди у вікні монітора послідовної зв'язку (serial monitor) [19].

```
#define USE_ARDUINO_INTERRUPTS true
```

```
#define DEBUG true
```

Тепер в програмі необхідно ввести ім'я свого WiFi підключення, пароль для підключення до нього і IP сервісу thingspeak.com.

```
#define SSID "DeltaFrost" // "назва wifi "
```

```
#define PASS "Denys2002" // "парол від wifi "
```

```
#define IP "184.106.153.149" // thingspeak.com ip
```

Після цього потрібно оголосити строкову змінну щоб оновлювати інформацію на каналі ThingSpeak. Для цього знадобиться API ключ, який можна знайти в сервісі ThingSpeak в наступному місці: ThingSpeak channel-> API key. Скопіювати його звідти і вставити в наступний рядок:

```
Stringmsg = "GET /update?key=YourApiKey";
```

У функції `setup` слід встановити бодову швидкість для послідовної зв'язку між монітором послідовної зв'язку Arduino і модулем `esp8266`. Запустимо в роботу модуль `esp8266` шляхом подачі на нього відповідної команди і з'єднаємо її з мережею WiFi за допомогою виклику функції `connectWiFi ()`. Після цього ініціалізуємо таймери за допомогою виклику функції `t.every (time_interval, do_this)`; За допомогою таймера буде задаватися часовий інтервал для зчитування даних з датчика.

Також в програмі потрібно запрограмувати функції `connectWiFi ()`, `panic_button ()`, `update_info ()` і `getReadings ()`.

Функція `connectWiFi ()` буде повертати значення `True` або `False` залежно від того чи буде підключений модуль до мережі WiFi або немає. Команда `AT + CWMODE = 1` переведе модуль `ESP8266` в роботу в режимі станції. За допомогою команди `AT + CWJAP = \` будемо здійснене підключення до точки доступу WiFi.

Функція `getReadings ()`; буде зчитувати дані з датчика сатурації і датчика температури `LM35` і конвертувати їх в рядок за допомогою функції `dtostrf ()`;

Тепер необхідно ініціалізувати символний масив для зберігання `BPM` (числа ударів пульсу в хвилину) і температури і конвертуємо значення з виходу датчиків в рядок за допомогою функції `dtostrf ()`.

```
char buffer1[10];
char buffer2[10];
BPM = dtostrf(myBPM, 4, 1, buffer1);
temp = dtostrf(myTemp, 4, 1, buffer2);
}
```

Після цього слід створити функцію для оновлення інформації від датчиків на каналі `ThingSpeak`. За допомогою команди `"AT + CIPSTART = \" TCP \", \"` встановлюється зв'язок по протоколу TCP на порту 80.

Необхідно включити в рядок для передачі на сервер `ThingSpeak` дані про пульс і температуру за допомогою рядків `"& field1 ="`; (Для пульсу) і `"& field2 ="`; (Для температури). Цю інформацію можна пердати використовуючи команду `"AT + CIPSEND ="`.



Аналогічним чином потрібно створити функцію для тривожної кнопки (`panic_button`). Коли на контакті, до якого підключена кнопка, буде напруга високого рівня (HIGH), модуль `esp8266` буде передавати інформацію на сервер використовуючи команди `AT + CIPSTART` і `AT + CIPSEND`.

Слід прикріпити цю інформацію до переданої на сервер рядку за допомогою `"& field3 ="` [21].

```
cmd = msg;
    cmd += "&field3=";
```

У функції `loop` потрібно викликати функції `panic_button ()` і `t.update ()`.

```
void loop()
{
    panic_button();

    while (!t.shouldIRun()) { // Зачекати, поки заплановані
завдання виконуються
    }
    t.update();
```

Повний текст програми розміщено у додатку Б.

## 2.5 Розробка інструкції з експлуатації електронного пристрою

Для того, щоб система виконувала свої функції, перш за все до `Arduino` потрібно приєднати елементи: такі як датчик температури, модуль вай-фай, датчик сатурації. Підключення потрібно проводити відповідно до функціональної схеми. .

Для роботи з пристроєм, необхідно прикласти палець до датчика сатурації після чого він почне зчитувати дані такі як пульс та насиченість крові киснем. Далі потрібно поставити палець на датчик температури щоб він виміряв температуру тіла. Після того як дані будуть зчитані з датчиків вони надсилаються на сервер де вони заносяться в таблицю. Приклад таблиці зображено на рисунку 2.10.

	A	B	C	D	E	F	G
1							
2	Date and Time		Pulse Rate (BPM)	Body Temp. (F)			
3	June 25, 2018 at 03:11PM	Patient_Info	217	51			
4	June 25, 2018 at 03:12PM	Patient_Info	127	53			
5	June 25, 2018 at 03:18PM	Patient_Info	220	50			
6	June 25, 2018 at 04:21PM	Patient_Info	222	54			
7	June 25, 2018 at 04:25PM	Patient_Info	209	62			
8	June 25, 2018 at 04:45PM	Patient_Info	121	86.5			
9	June 25, 2018 at 04:46PM	Patient_Info	103	83.9			
10	June 25, 2018 at 05:01PM	Patient_Info	209	99.7			
11	June 25, 2018 at 05:01PM	Patient_Info	212	83			
12	June 25, 2018 at 05:02PM	Patient_Info	209	79.5			
13							
14							
15							

Рисунок 2.10 – Дані пацієнта

Також в пристрої є функція тривожно кнопки натиснувши на яку і затримавши на 4-5 секунд сімейний лікар отримаєте email з текстом що у пацієнта проблеми. Приклад email зображено на рисунку 2.11 [22].

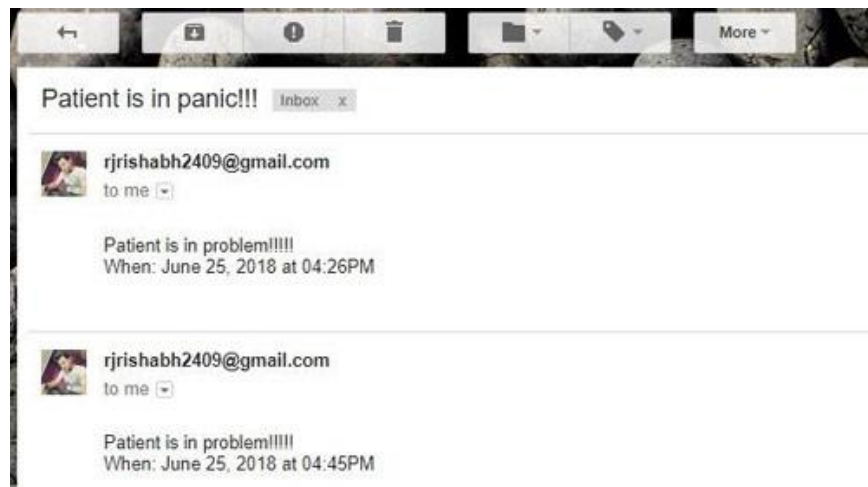


Рисунок 2.11 – Приклад роботи тривожно кнопки

## 2.6 Розробка методики перевірки, функціонування (контролю, випробування) електронного пристрою

Для тестування і налагодження системи використовуються віртуальний термінал вбудований в середовище Arduino. Значення температури і сатурвції з датчиків виводяться у вікні терміналу а також у електроній таблиці на сервері.

У автоматизованій системі дистанційного моніторингу температури і сатурації можливі такі види несправностей:

1. Не виводяться дані і не вносяться в таблицю: перевірити правильність підключення датчика до Arduino; перевірити чи подається живлення на плату; перевірити наявність підключення вай-фай модуля до серверу; перевірити під'єднання вай-фай модуля до Arduino;

2. Некоректно відображається температура пацієнта: запустити термінал і перевірити дані датчика. Якщо дані відображаються некоректно, необхідно перевірити підключення датчика температури;

3. Некоректно відображається сатурація і пульс пацієнта: запустити термінал і перевірити дані датчика. Якщо дані відображаються некоректно, необхідно перевірити підключення датчика температури; необхідно відкалібрувати датчик, завантаживши скетч (додаток А) в мікроконтролер;

4. Не можливо з'єднатись з модулем WI-FI: звернути увагу на підключення виводів модуля до плати Arduino; перевірити роботу плати Arduino, модуля WI-FI і подачу живлення.

## **2.7 Висновок до другого розділу**

У другому розділі кваліфікаційної роботи [23] була розроблена функціональна схема пристрою, яка враховує всі необхідні функції та забезпечує ефективну взаємодію з системою. Це дозволяє зручно та ефективно збирати, обробляти та аналізувати дані про пацієнтів, що допомагає лікарю приймати обґрунтовані рішення та забезпечує підтримку його роботи.

Розроблений алгоритм системи дозволяє ефективно обробляти та аналізувати дані про пацієнтів, забезпечуючи швидкий доступ до необхідної інформації та виявлення потенційних проблем. Це сприяє покращенню якості медичної діагностики та наданню своєчасної допомоги пацієнтам.

## РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 3.1 Долікарська допомога при ураженні електричним струмом

Долікарська допомога при ураженні електричним струмом є важливим аспектом надання першої медичної допомоги в екстрених ситуаціях. При ураженні електричним струмом виникає потенційна загроза для життя потерпілого, тому необхідно діяти швидко та правильно для забезпечення безпеки та мінімізації можливих ускладнень.

Перш за все, важливо відключити джерело електричного струму, якщо це можливо, наприклад, виключити лінію електропостачання або відключити прилад, що спричинив ураження. При цьому слід дотримуватися всіх безпечних процедур та не вступати в контакт з електричними джерелами без належного захисту.

Наступним кроком є негайне викликання медичної допомоги, зателефонувавши на екстрену лінію. Інформація про ураження електричним струмом повинна бути надана оператору, а також повідомлено про потенційні пошкодження, що можуть виникнути [24].

Поки медична допомога не надійде, слід негайно оцінити стан потерпілого та надати першу допомогу [25]. Важливо забезпечити безпеку для себе та оточуючих, перед тим як наближатися до ураженої особи. Необхідно перевірити наявність дихання та пульсу. Якщо серцева діяльність зупинилася, слід негайно розпочати кардіопульмональну реанімацію (кпр) або надати штучне дихання та виконати непрямий масаж серця.

У разі наявності дихання та пульсу, необхідно забезпечити стабільне положення потерпілого та уникати будь-яких рухів, які можуть призвести до подальшого ушкодження. Варто з'ясувати наявність інших травм або пошкоджень та надати першу допомогу залежно від ситуації, наприклад, зупинити кровотечу, застосувати холод на опікні ділянки тощо.

Важливо пам'ятати, що надання долікарської допомоги при ураженні електричним струмом повинно здійснюватися фахівцями з медичної сфери.

Інструкції, наведені вище, є загальними рекомендаціями та не замінюють професійного медичного консультування та допомоги. Лише медичні працівники можуть оцінити стан постраждалого та прийняти відповідні рішення щодо надання допомоги з урахуванням конкретних обставин.

### **3.2 Розробка конкретних заходів щодо боротьби із статичною електрикою**

Боротьба зі статичною електрикою вимагає розуміння її природи та прийняття певних заходів для зменшення її впливу. Основна мета полягає в усуненні неприємних ефектів статичної електрики, таких як розряди, неприємні відчуття та можливі пошкодження електронної апаратури [26]. Ось кілька конкретних заходів, які можна вжити для боротьби зі статичною електрикою.

1. Застосування зволожувачів повітря: Встановлення зволожувачів в приміщеннях допомагає підвищити рівень вологості, що зменшує накопичення статичної електрики на поверхнях і в повітрі.

2. Земляні контакти: Важливо забезпечити належну заземленість для пристроїв, які легко накопичують статичний заряд, наприклад, комп'ютери та електронна апаратура. Заземлення допомагає розподілити надлишковий заряд у безпечний спосіб.

3. Використання антистатичних матеріалів: Виробники одягу та меблів пропонують антистатичні матеріали, які допомагають уникнути накопичення статичного заряду. Носіння такого одягу або використання покриттів на меблі та підлогу може допомогти зменшити ризик статичних розрядів.

4. Використання антистатичних засобів: Для електронної апаратури, такої як комп'ютери, існують спеціальні антистатичні засоби, які допомагають зменшити накопичення статичної електрики та захистити обладнання від можливих пошкоджень.

5. Використання пристроїв з контролем статички: Деякі пристрої, наприклад, антистатичні підлогові мати або спеціальні розетки з захистом від статичної електрики, можуть бути використані для зменшення ризику

накопичення заряду.

6. Уникання тертя: Уникайте тертя одягу або матеріалів, яке може створювати статичну електрику. Наприклад, носіть натуральні тканини та уникайте синтетичних матеріалів, які більше схильні до накопичення заряду [27].

Ці заходи спрямовані на зменшення негативного впливу статичної електрики і створення комфортного та безпечного середовища. Варто пам'ятати, що ефективність заходів може варіюватися залежно від конкретних умов та ситуацій, тому рекомендується вивчати та впроваджувати відповідні рішення, враховуючи специфіку кожного випадку.

### **3.3 Особливості заходів електробезпеки на підприємствах**

Заходи електробезпеки на підприємствах є невід'ємною частиною діяльності, яка має на меті запобігання нещасним випадкам та забезпечення безпеки працівників, а також захисту від можливих пошкоджень електроустаткування. Особливості таких заходів включають наступне [28]:

1. Розробка та виконання електробезпекових інструкцій: Підприємства повинні розробити докладні інструкції з електробезпеки, які включають процедури роботи з електрообладнанням, заземлення та використання захисного обладнання. Ці інструкції повинні бути оновлюваними і доступними для всіх працівників.

2. Проведення періодичних навчань та тренінгів: Працівники повинні бути навчені правильним методам роботи з електричним обладнанням, використання захисного обладнання, виявлення небезпечних ситуацій та надання першої допомоги. Проведення регулярних тренінгів допомагає підтримувати свідомість працівників щодо електробезпеки.

3. Регулярна перевірка та обслуговування електроустаткування: Підприємства повинні регулярно перевіряти та обслуговувати електрообладнання згідно з встановленими стандартами. Це включає перевірку проводки, заземлення, ізоляції та інших параметрів для виявлення потенційних проблем та усунення їх до виникнення аварій [29].

4. Встановлення захисних пристроїв: Підприємства повинні встановити відповідні захисні пристрої, такі як автоматичні вимикачі, розетки з захистом від струму, грозозахисні системи тощо. Це допомагає запобігти несправностям, коротким замиканням та можливим пошкодженням.

5. Система контролю та аудиту: Підприємства повинні мати систему контролю та аудиту, яка дозволяє періодично оцінювати ефективність заходів електробезпеки, виявляти потенційні ризики та вносити відповідні зміни до процедур та політик.

Враховуючи особливості підприємства, його розмір, характеристики та види діяльності, заходи електробезпеки можуть відрізнятися [30]. Однак, загальні принципи безпеки, навчання та перевірки є важливими елементами для забезпечення безпеки працівників та уникнення можливих нещасних випадків, пов'язаних з електричним струмом.

## ВИСНОВКИ

В процесі виконання дипломного проекту було розроблено інформаційну систему дистанційного моніторингу температури і сатурації пацієнтів сімейного лікаря.

В першому розділі даного дипломного проекту було розглянуто обґрунтування актуальності теми дипломного проекту та аналітичний огляд існуючих рішень.

У другому розділі описується аналіз технічного завдання, розробка структурної і функціональної схеми з вибором елементної бази, написання програми управління для платформи *Arduino*, та взаємодія з платформою *ThingSpeak*.

Після впровадження певного функціоналу, проводилося тестування усіх сценаріїв використання програми.

У розділі "Безпека життєдіяльності, основи охорони праці" описано небезпеку яку несе електричний струм і яку долікарську допомогу необхідно надавати у різних ситуаціях. Також розглянуто основні заходи щодо боротьби із статичною електрикою та електробезпекою на підприємствах.



**ПЕРЕЛІК ДЖЕРЕЛ**

1. Пульсоксиметр [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki?curid=829763>
2. Портативний пульсоксиметр Gamma Oxy Scan [Електронний ресурс] – Режим доступу до ресурсу: <https://medtehnikalife.com.ua/ua/pulsoksimetr-gamma-oxy-scanc>
3. Пульсоксиметр G1B [Електронний ресурс] – Режим доступу до ресурсу: <http://heaco.kiev.ua/pulsoksimetr-g1b-uk>
4. Studfile [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/9154031/page:10/>
5. ЗБІРНИК НАУКОВИХ ПРАЦЬ МІЖНАРОДНОЇ НАУКОВО-ТЕХНІЧНОЇ INTERNETКОНФЕРЕНЦІЇ СТУДЕНТІВ І МОЛОДИХ ВЧЕНИХ [Електронний ресурс] – Режим доступу до ресурсу: [http://www.dgma.donetsk.ua/docs/kafedry/tiup/konf/МОЛОДА\\_НАУКА\\_2017.pdf](http://www.dgma.donetsk.ua/docs/kafedry/tiup/konf/МОЛОДА_НАУКА_2017.pdf)
6. Home protector [Електронний ресурс] – Режим доступу до ресурсу: <https://andriy1024.github.io/home-protector/menu/elementbase.html>
7. elar.khnu.km.ua [Електронний ресурс] – Режим доступу до ресурсу: <http://elar.khmnu.edu.ua/jspui/bitstream/123456789/10117/1/Part%205-32-36.pdf>
8. ela.kpi.ua [Електронний ресурс] – Режим доступу до ресурсу: [https://my.plag.com.ua/file/fLr4RiG2-2023-KRB-ST-41-Tkachuk-D-antiplagiat-1-docx?page\\_no=3](https://my.plag.com.ua/file/fLr4RiG2-2023-KRB-ST-41-Tkachuk-D-antiplagiat-1-docx?page_no=3)
9. lpnu.ua [Електронний ресурс] – Режим доступу до ресурсу: <https://lpnu.ua/sites/default/files/2020/dissertation/1637/diser.pdf>
10. ela.kpi.ua [Електронний ресурс] – Режим доступу до ресурсу: [https://ela.kpi.ua/bitstream/123456789/34956/2/Krot\\_bakalavr.pdf](https://ela.kpi.ua/bitstream/123456789/34956/2/Krot_bakalavr.pdf)
11. Arduino Duemilanove [Електронний ресурс] – Режим доступу до ресурсу: <https://arduino-school.blogspot.com/2012/08/arduino-duemilanove.html>
12. Установка для декоративного освітлення на основі RGBсвітлодіода з функцією керування кольором [Електронний ресурс] – Режим доступу до ресурсу: [https://ela.kpi.ua/bitstream/123456789/28865/1/Dran\\_bakalavr.pdf](https://ela.kpi.ua/bitstream/123456789/28865/1/Dran_bakalavr.pdf)

13. Давач температури LM35 Arduino (11741) [Електронний ресурс] – Режим доступу до ресурсу: <https://bigl.ua/ua/p896870750-datchik-temperature-lm35>
14. Вивчення властивостей мікроконтролерів і електронних систем на базі платформи Ардуіно [Електронний ресурс] – Режим доступу до ресурсу: <https://openarchive.nure.ua/server/api/core/bitstreams/7349b613-9f68-4edf-9f1d-c35baac25c76/content>
15. ТЕЛЕМЕДИЦИНА. КОМП'ЮТЕРНІ СИСТЕМИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ [Електронний ресурс] – Режим доступу до ресурсу: <file:///C:/Users/dtkac/Downloads/Телемедицина.%20Монографія.pdf>
16. Розробка інформаційної системи GPS-контролю перевезень багажу [Електронний ресурс] – Режим доступу до ресурсу: [https://elartu.tntu.edu.ua/bitstream/lib/35804/1/КР\\_Церковний\\_В.О\\_2021.pdf](https://elartu.tntu.edu.ua/bitstream/lib/35804/1/КР_Церковний_В.О_2021.pdf)
17. Моделі управління "розумної" мінітеплиці / "Smart" mini greenhouses control models [Електронний ресурс] – Режим доступу до ресурсу: [http://dspace.wunu.edu.ua/bitstream/316497/40534/1/Гросуляк\\_КР\\_2020.pdf](http://dspace.wunu.edu.ua/bitstream/316497/40534/1/Гросуляк_КР_2020.pdf)
18. IoT Based Patient Health Monitoring System using ESP8266 and Arduino [Електронний ресурс] – Режим доступу до ресурсу: <https://circuitdigest.com/microcontroller-projects/iot-based-patient-monitoring-system-using-esp8266-and-arduino>
19. MATLAB Answers [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mathworks.com/matlabcentral/answers/792122-sensor-data-doesn-t-gets-uploaded-to-thingspeak-channel-but-it-show-on-the-serial-monitor>
20. Удосконалення управління інноваційною активністю підприємства ресторанного господарства (на прикладі ресторану “Галич”) [Електронний ресурс] - Режим доступу до ресурсу: [https://elartu.tntu.edu.ua/bitstream/lib/36811/1/Dyplom\\_Dobrutaska\\_Yu\\_M\\_2021.pdf](https://elartu.tntu.edu.ua/bitstream/lib/36811/1/Dyplom_Dobrutaska_Yu_M_2021.pdf)
21. IoT Based Patient Monitoring System using ESP8266 and Arduino [Електронний ресурс] - Режим доступу до ресурсу: <https://duino4projects.com/iot-based-patient-monitoring-system-using-esp8266-and-arduino/>

22. Розроблення високочутливих оптичних сенсорів шкідливих газів (So<sub>2</sub>, No<sub>2</sub>, Та Co<sub>2</sub>) на основі рідкокристалічних речовин, допованих вуглецевими нанотрубками [Електронний ресурс] - Режим доступу до ресурсу: <https://ena.lpnu.ua:8443/server/api/core/bitstreams/aba994f6-6254-4cd1-a633-c325d6f2a6cb/content>

23. Розробка соціальної мережі засобами MongoDB та Node.js [Електронний ресурс] - Режим доступу до ресурсу: [https://elartu.tntu.edu.ua/bitstream/lib/38351/1/2022\\_KRB\\_SN-41\\_HALIUK\\_v1.8.pdf](https://elartu.tntu.edu.ua/bitstream/lib/38351/1/2022_KRB_SN-41_HALIUK_v1.8.pdf)

24. Пам'ятка «Перша допомога при ураженні електричним струмом» [Електронний ресурс] - Режим доступу до ресурсу: <https://bozhedarivskaselrada.gov.ua/news/1576497483/>

25. Перша допомога при ураженні електричним струмом. [Електронний ресурс] - Режим доступу до ресурсу: [http://www.yu.mk.ua/news/show/persha\\_dopomoga\\_pri\\_urazhenni\\_elektrichni\\_m\\_strumom?](http://www.yu.mk.ua/news/show/persha_dopomoga_pri_urazhenni_elektrichni_m_strumom?)

26. Гарьковець А. М. Методичні вказівки до виконання розділу «Охорона праці та безпека в надзвичайних ситуаціях» в дипломних проектах та магістерських роботах (для студентів усіх форм навчання за спеціальностями 7.05070202, 8.05070202 «Електричні системи і комплекси транспортних засобів», 7.05070203, 8.05070203 «Електричний транспорт», 7.05070204, 8.05070204 «Електромеханічні системи автоматизації та електропривод») / Харк. нац. ун-т міськ. госп-ва; уклад.: А. М. Гарьковець. – Х. : ХНУМГ, 2013. – 15 с. Режим доступу до ресурсу: <https://core.ac.uk/download/pdf/16423754.pdf>

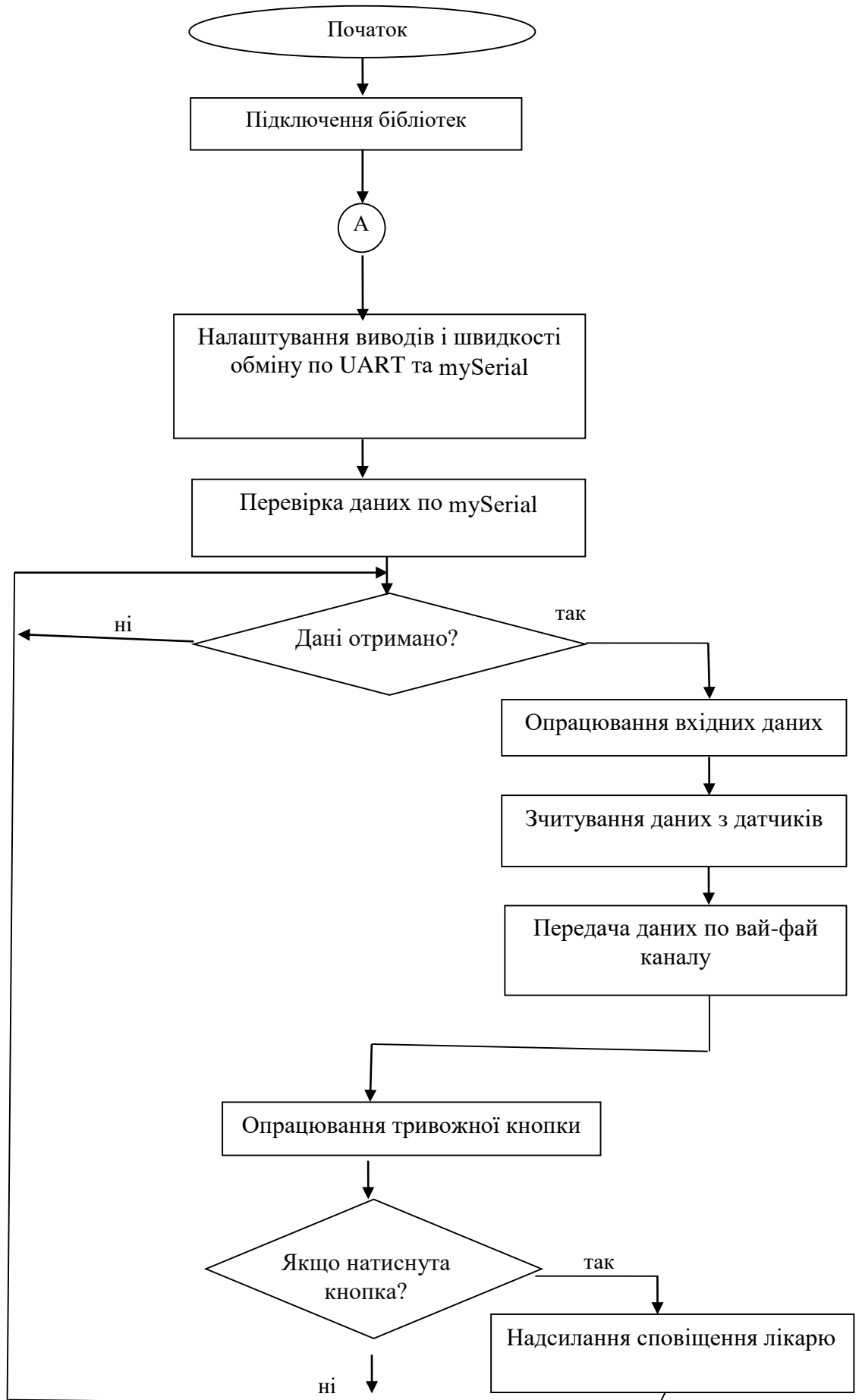
27. Запорожець О. І., Протоєрейський О. С., Франчук Г. М., Боровик І. М. Основи охорони праці. Підручник. – К.: Центр учбової літератури, 2009. – 264 с. Режим доступу до ресурсу: <https://studies.in.ua/bjd-zaporojec/1252-133-sposobizasobi-zahistu-vd-statichnoyi-elektriki.html>

28. Ліньов Анатолій. Електробезпека. Режим доступу до ресурсу: <https://pro-op.com.ua/article/745-elektrobezpeka>

29. Електробезпека на підприємстві та в офісі. Режим доступу до ресурсу:  
<https://profiteh.ua/elektrobezpeka-na-pidpryemstvi-ta-ofisi/>
30. Про заходи з електробезпеки для працівників офісу. Режим доступу до ресурсу:  
<https://oppb.com.ua/articles/pro-zahody-z-elektrobezpeky-dlya-pracivnykiv-ofisu>

# ДОДАТКИ

## Алгоритм системи



## Лістинг коду програми

```

#define USE_ARDUINO_INTERRUPTS true
#define DEBUG true
#define SSID "*****" // "SSID-WiFiname"
#define PASS "*****" // "password"
#define IP "184.106.153.149" // thingspeak.com ip

#include <SoftwareSerial.h>
#include "Timer.h"
#include <MAX30100_SpO2Calculator.h>

Timer t;
PulseSensorPlayground pulseSensor;

const int PulseWire = A0; // Підключення фіолетового
дроту PulseSensor до аналогового піна 0
const int LED13 = 13; // Вбудований світлодіод
Arduino, поблизу піна 13
int Threshold = 550; // для датчика пульсу

float myTemp;
int myBPM;
char BPM[10];
char temp[10];
int panic;
int raw_myTemp;
float Voltage;
float tempC;

SoftwareSerial esp8266(10, 11);
void setup()
{
  Serial.begin(9600);
  esp8266.begin(115200);
  pulseSensor.analogInput(PulseWire);
  pulseSensor.blinkOnPulse(LED13);
  pulseSensor.setThreshold(Threshold);

  if (pulseSensor.begin()) {
    Serial.println("Створено об'єкт PulseSensor!");
  }
  connectWiFi();
  t.every(10000, getReadings);
  t.every(10000, updateInfo);
}
void loop()
{
  panic_button();
  while (!t.shouldIRun()) {
    // Зачекайте, поки заплановані завдання виконуються
  }
  t.update();
  delay(4000);
}

```

```

}
void updateInfo()
{
    String cmd = "AT+CIPSTART=\"TCP\", \"";
    cmd += IP;
    cmd += "\",80";
    Serial.println(cmd);
    esp8266.println(cmd);
    delay(2000);
    if (esp8266.find("Error"))
    {
        return;
    }
    cmd = msg;
    cmd += "&field1=";    // поле 1 для пульсу (BPM)
    cmd += BPM;
    cmd += "&field2=";  // поле 2 для температури
    cmd += temp;
    cmd += "\r\n";
    Serial.print("AT+CIPSEND=");
    esp8266.print("AT+CIPSEND=");
    Serial.println(cmd.length());
    esp8266.println(cmd.length());
    if (esp8266.find(">"))
    {
        Serial.print(cmd);
        esp8266.print(cmd);
    }
    else
    {
        Serial.println("AT+CIPCLOSE");
        esp8266.println("AT+CIPCLOSE");
        // Повторна відправка...
        error = 1;
    }
}
}
boolean connectWiFi()
{
    Serial.println("AT+CWMODE=1");
    esp8266.println("AT+CWMODE=1");
    delay(2000);
    String cmd = "AT+CWJAP=\"";
    cmd += SSID;
    cmd += "\", \"";
    cmd += PASS;
    cmd += "\"";
    Serial.println(cmd);
    esp8266.println(cmd);
    delay(5000);
    if (esp8266.find("OK"))
    {
        return true;
    }
    else
    {
        return false;
    }
}

```



```

    }
}
void getReadings()
{
    raw_myTemp = analogRead(A1);
    Voltage = (raw_myTemp / 1023.0) * 5000; // 5000 для
отримання мілівольт
    tempC = Voltage * 0.1;
    myTemp = (tempC * 1.8) + 32; // перетворення в градуси
Фаренгейта
    Serial.println(myTemp);
    int myBPM = pulseSensor.getBeatsPerMinute();
    if (pulseSensor.sawStartOfBeat()) {
        Serial.println(myBPM);
    }
    delay(20);
    dtostrf(myBPM, 4, 1, BPM);
    dtostrf(myTemp, 4, 1, temp);
}
void panic_button()
{
    panic = digitalRead(8);
    if (panic == HIGH)
    {
        Serial.println(panic);
        String cmd = "AT+CIPSTART=\"TCP\", \"";
        cmd += IP;
        cmd += "\",80";
        Serial.println(cmd);
        esp8266.println(cmd);
        delay(2000);
        if (esp8266.find("Error"))
        {
            return;
        }
        cmd = msg;
        cmd += "&field3=";
        cmd += panic;
        cmd += "\r\n";
        Serial.print("AT+CIPSEND=");
        esp8266.print("AT+CIPSEND=");
        Serial.println(cmd.length());
        esp8266.println(cmd.length());
        if (esp8266.find(">"))
        {
            Serial.print(cmd);
            esp8266.print(cmd);
        }
        else
        {
            Serial.println("AT+CIPCLOSE");
            esp8266.println("AT+CIPCLOSE");// Повторна відправка...
            error = 1;
        }
    }
}
}

```