

Міністерство освіти і науки України
Чернівецький національний університету імені Юрія Федьковича»
Відокремлений структурний підрозділ «Фаховий коледж Чернівецького
національного університету імені Юрія Федьковича»

Природниче відділення
Циклова комісія комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітньо-професійного ступеня «фаховий молодший бакалавр»
зі спеціальності 123 Комп'ютерна інженерія
підготовки за освітньо-професійною програмою «Комп'ютерна інженерія»
на тему: «Система моніторингу здоров'я пацієнтів на основі IoT
з використанням ESP8266 і Arduino.»

Виконав (ла):

студент (ка) 4-го курсу Скрипник Юрій Васильович
(прізвище, ім'я та по батькові) (підпис)

Керівник:

к. ф.-м. н., Тащук О.Ю.
(науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент:

Мельничук А. Ю.
(науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

До захисту допущено:

Протокол засідання циклової комісії № _

від „_” _____ 2024 р.

Голова циклової комісії _____ Олександр ТАЩУК

Чернівці, 2024

Завдання та календарний план роботи

Міністерство освіти і науки України
Чернівецький національний університету імені Юрія Федьковича»
Відокремлений структурний підрозділ «Фаховий коледж
Чернівецького національного університету імені Юрія Федьковича»

Затверджую
Голова циклової комісії,
Олександр ТАЩУК.

«___» _____ 2023 р.

Завдання

На кваліфікаційну роботу	Скрипник Юрій Васильович
Тема кваліфікаційної роботи	Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino
Постановка завдання в короткій формі	Розробка системи моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino..
Вихідні дані (2-3 адреси сайтів, матеріали яких рекомендує керівник кваліфікаційної роботи)	

Календарний план роботи

Дата отримання версії звіту керівником	Етап виконання роботи	Виконання (зазначає керівник)
10.10.23	Отримання завдання до кваліфікаційної роботи.	
08.11.23	Огляд літератури, присвяченої розробці систем моніторингу стану здоров'я на основі IoT.	
21.12.23	Аналіз існуючих конструкторських рішень. Оформлення розділу 1.	
27.01.24	Вибір компонентної бази, узгодження модулів системи.	
29.02.24	Написання теоретичного матеріалу по використовуваних модулях.	
30.03.24	Проектування та розробка апаратної частини, підключення Оформлення розділу 2.	
28.04.24	Проектування та розробка програмної частини, підключення. Оформлення розділу 3.	

17.05.24	Оформлення та редагування кваліфікаційної роботи.	
31.05.24	Подання роботи на перевірку Інтернет-сервісом Unichesk.	
03.06.24	Подання роботи на рецензію	
06.06.24	Представлення роботи на засіданні Циклової комісії	
<i>за графіком</i>	Захист кваліфікаційної роботи	

Дата видачі завдання _____.

Студент

Юрій СКРИПНИК.

Керівник (П.І.Б)

Олександр ТАЩУК

АНОТАЦІЯ

Скрипник Юрій Васильович Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino. Кваліфікаційна робота.

В роботі виконано моделювання та розробку Системи моніторингу здоров'я пацієнтів на базі мікроконтролерів ESP8266 і Arduino. Розроблена система базується на концепції IoT, яка дозволяє дистанційно стежити за здоров'ям людини

Актуальність роботи пояснюється тим, що у даний момент часу Інтернет речей (IoT) здійснює революцію у сфері охорони здоров'я, роблячи моніторинг стану пацієнтів більш зручним, точним та доступним. Завдяки таким технологіям медичні заклади можуть забезпечити безперервний моніторинг пацієнтів, своєчасну діагностику, здійснювати персоналізоване лікування. Системи моніторингу здоров'я на основі IoT збирають дані про фізіологічні показники пацієнтів за допомогою датчиків, а потім передають їх на хмарні платформи для аналізу та візуалізації. Це дозволяє медичним працівникам відстежувати стан пацієнтів у режимі реального часу, своєчасно виявляти ознаки погіршення самопочуття та надавати їм активну допомогу.

Робота складається з 3 розділів, в яких проаналізовано роботу, змодельовано, описано та виконано розробку Системи моніторингу здоров'я пацієнтів на базі мікроконтролерів ESP8266 і Arduino..

Кваліфікаційна робота містить 52 сторінок, 13 рисунків та 20 посилань на літературні джерела.

Ключові слова: *Інтернет речей IoT, система моніторингу здоров'я, , Arduino Uno, ESP 8266, датчик температури, хмарне сховище, датчик пульсу.*

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
1. ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Поняття та будова Інтернету речей (IoT)	9
1.2 Роль IoT в сучасній медицині.....	11
1.3 Огляд існуючих систем моніторингу здоров'я.....	12
2. ПРОЕКТУВАННЯ АПАРАТНОЇ СКЛАДОВОЇ СИСТЕМИ МОНІТОРИНГУ ЗДОРОВ'Я.....	20
2.1 Вибір компонент.....	20
2.1.1 Вибір апаратної платформи ESP8266 та Arduino Uno.....	20
2.1.2. Датчики та інші апаратні компоненти системи моніторингу.....	23
2.2. Схеми підключення систем моніторингу здоров'я.....	26
3. ПРОЕКТУВАННЯ ПРОГРАМНОЇ СКЛАДОВОЇ СИСТЕМИ МОНІТОРИНГУ ЗДОРОВ'Я.....	30
3.1 Структура програмного забезпечення та хмарного сховища для системи IoT.....	30
3.2 Програмування ESP8266 з використанням Arduino IDE.....	31
ВИСНОВКИ.....	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	39
ДОДАТКИ.....	22
Додаток А. Лістинг програми.....	41
Додаток Б. Схема системи моніторингу здоров'я на основі IoT.....	49
Додаток В. Структурна схема підключення датчиків до плати Arduino Uno..	50
Додаток Г. Алгоритм роботи пристрою.....	51
Додаток Д. Принципова схема системи моніторингу здоров'я пацієнтів на основі ІОТ	52
Додаток Ж. Друкована плата системи моніторингу здоров'я пацієнтів на основі ІОТ.....	53

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

IoT – Інтернет речей.

ОС – операційна система.

САПР – система автоматизованого проектування.

СП – система приводів.

Wi-Fi – технологія бездротової локальної мережі.

GPIO – інтерфейс для зв'язку між компонентами в системі комп'ютерів.

PCB – друкована плата

LCD- дисплей (Liquid Crystal) – рідкокристалічний дисплей.

ВСТУП

Інтернет речей (IoT) є однією з найбільших революційних технологій останнього десятиліття, яка відкрила нові горизонти для підключення та взаємодії між фізичними об'єктами та інформаційними системами. IoT об'єднує безліч пристроїв у мережі, де вони можуть збирати, обробляти та обмінюватися даними, що дозволяє створювати інтелектуальні системи та додатки в різних сферах, включаючи охорону здоров'я, промисловість, транспорт, політику та багато інших.

Інтернет речей виконує ключову роль у трансформації медичної галузі, пропонуючи інноваційні рішення для підвищення якості медичних послуг, підвищення ефективності та зниження витрат. Завдяки IoT, медичні заклади можуть забезпечити безперервний моніторинг та вчасну діагностику стану пацієнта, та надавати персоналізоване лікування.

Концепція Smart систем продовжує розвиватися з неймовірною швидкістю, і їх потенціал ще далеко не вичерпано. З розвитком технологій, таких як 5G, штучний інтелект (AI) та машинне навчання, IoT відкриває нові можливості для створення інтелектуальних систем, здатних автоматично адаптуватися до змін у навколишньому середовищі та приймати складні рішення в режимі реального часу.

Метою роботи є розробка системи моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino. У зв'язку з зростанням зацікавленості до сфери e-здоров'я важливим є розробка технологічних рішень, які здійснюють постійний контроль стану здоров'я пацієнтів, для цього було виконано дослідження технічних можливостей ESP8266 та Arduino для збору та передачі даних, а також змодельовано і розроблено системи для моніторингу здоров'я на базі мікроконтролерів ESP8266 і Arduino.

Об'єктом дослідження система моніторингу здоров'я пацієнтів, які апаратною частиною якого є ESP8266 і Arduino.

Предметом дослідження є розробка апаратного та програмного

забезпечення для збору та відображення даних, а також забезпечення надійності та ефективності розробленої системи.

Мета роботи: розробка та тестування системи моніторингу здоров'я пацієнтів з використанням ESP8266 і Arduino.

Для досягнення цієї мети необхідно вирішити такі завдання:

- Вибір технічних засобів;
- Розробка програмного забезпечення для збору та відображення даних.
- Забезпечення надійності та ефективності розробленої системи.

Методи дослідження. Методи і технології створення систем автоматичного управління (САУ) для різних процесів і комплексів які включають в себе інструментальні засоби для моделювання, планування, а також розробки алгоритмів і програмного забезпечення, необхідного для аналізу та синтезу складних розподілених у просторі гнучких комп'ютерно інтегрованих систем.

Практична цінність полягає в тому що, отримані в роботі результати можна застосувати для подальшої розробки систем моніторингу здоров'я на основі IoT технологій що дадуть змогу відстежувати стан здоров'я в реальному часі.

Актуальність теми полягає в необхідності розвитку інноваційних підходів до моніторингу здоров'я населення через використання IoT технологій. Підвищення якості медичних послуг та зменшення ризику захворювань це основні аспекти, які роблять цю тему актуальною. У сучасних умовах коли поширені різні захворювання та віруси наш світ потребує ефективних та новаторських підходів до медичної діагностики, які забезпечуються системи інтернету речей. Застосування мікроконтролерів ESP8266 та Arduino дозволяє створювати доступні та ефективні системи збору та аналізу медичних даних.

ПОСТАНОВКА ЗАДАЧІ

1.1. Поняття та сутність Інтернет речей (IoT)

Інтернет речей (IoT) — це система взаємопов'язаних об'єктів, які мають унікальні ідентифікатори та здатні передавати дані через мережу без потреби у втручанні людини. Згідно з дослідженнями, проведеними Кевіном Ештоном, IoT включає в себе різні пристрої, від простих сенсорів до складних інтелектуальних систем, які можуть взаємодіяти одна з одною та із зовнішнім середовищем [1].

До основних компонентів IoT належать:

Сенсори та датчики, здійснюють збір інформації з навколишнього середовища.

Пристрої аналізу та перетворення виконують аналіз та обробку зібраної інформації.

Комунікаційні мережі забезпечують передачу даних між пристроями.

Інтерфейси користувача надають можливість користувачу взаємодіяти з системою.

Архітектура IoT складається з кількох рівнів, кожен з яких має свої функції та завдання [3] (Gubbi et al., 2013):

Фізичний рівень: Включає сенсори, датчики та виконавчі механізми, які збирають та вимірюють дані.

Рівень передачі даних: Включає комунікаційні протоколи та мережі (Wi-Fi, Bluetooth, Zigbee, LoRaWAN), що забезпечують передачу даних від пристроїв до центральних систем обробки.

Рівень обробки даних: Містить платформи та сервери, які здійснюють аналіз та обробку зібраної інформації.

Рівень програм: Містить інтерфейси користувача та прикладні програми, які виконують обробку даних які слугують для прийняття рішень або виконання певних дій.

Архітектура IoT може бути реалізована для нових моделей, включаючи хмарну обробку даних до якої належать, периферійна та граничну обробки,

які в свою чергу виконують туманні а також граничні обчислення. Кожна з цих моделей володіє рядом своїх переваг та недоліків, які залежать від конкретних продуктів та вимог системи [4].

IoT має широкий спектр застосування у різних галузях, таких як промисловість, де Індустріальний Інтернет-речей (IIoT) здійснює моніторинг обладнання та управління енергоспоживанням що дозволяє оптимізувати виробничі процеси. У транспортній промисловості на основі IoT створюються інтелектуальні транспортні системи, а також виконується моніторинг, управління рухом та логістикою. Така мережева концепція фізичних об'єктів, які оснащені вбудованими датчиками, відповідним ПЗ чи технологіями, які можуть обмінюватися даними через інтернет це дає можливість до створення цілих Smart систем від розумного будинку до розумного міста, де на основі концепції IoT здійснюється управління та моніторинг інфраструктури, екологічних умов, користувацьких задач чи забезпечення безпеки. Також інтернет речей володіє значним впливом на сферу медицини, відкриваючи нові можливості для покращення якості надання медичних послуг та забезпечення потреб.

Хоча IoT має великий потенціал, однак він також володіє низкою проблем, які потребують рішення адже при підключенні великої кількості пристроїв до мережі виникають ризики для безпеки та конфіденційності, які можуть бути пов'язані з кібератаками та втратою особистих даних, тому розробка ефективних заходів безпеки є одним із головних завдань для впровадження IoT [5]

1.2. Роль IoT в сучасній медицині

Одним із ключових напрямків застосування IoT є медична галузь, завдяки чому така співпраця дозволяє застосовувати інноваційні рішення для підвищення ефективності та якості надання медичних послуг. Системи моніторингу здоров'я на основі інтернету речей збирають дані про фізіологічні показники пацієнтів за допомогою датчиків, а потім передають

їх на хмарні платформи для аналізу та візуалізації. Це дозволяє медичним працівникам відстежувати стан пацієнтів у режимі реального часу, своєчасно здійснювати діагностику, надавати проактивну допомогу, здійснювати персоналізоване лікування пацієнтів, Дистанційний моніторинг пацієнтів (RPM дозволяє знизити кількість госпіталізацій та покращити якість життя та обслуговування пацієнтів.

IoT також сприяє розвитку персоналізованої медицини. Збір даних з різних пристроїв та їх аналіз за допомогою алгоритмів машинного навчання дозволяє створити індивідуальні плани лікування, які враховують особливості кожного пацієнта [5, 6].

Концепція розумних лікарень передбачає використання IoT для оптимізації роботи медичних закладів. Це включає автоматизацію рутинних процесів, управління медичним обладнанням, моніторинг стану потреби та забезпечення безпеки [7]. Розумні лікарні можуть підвищити ефективність роботи медичного персоналу та забезпечити кращу якість медичних послуг. Завдяки IoT лікарі можуть надавати медичну консультацію та лікування на відстані, використовуючи відеоконференції та дистанційний моніторинг [8]. Це особливо важливо для допомоги у віддалених районах, де доступ до медичних послуг обмежений, також така концепція дозволяє ефективніше стежити за хронічними захворюваннями, такими як діабет, гіпертонія та серцево-судинні захворювання.

Системи моніторингу здоров'я на основі IoT можна класифікувати за кількома ознаками, включаючи тип даних, що збираються, місце розташування пацієнта та спосіб передачі даних.

Пристрої для моніторингу рівня глюкози, артеріального тиску та інших параметрів дозволяють пацієнтам самостійно контролювати стан та тимчасово отримувати консультації лікарів [9].

Інтеграція пристроїв IoT з електронними медичними записами (EMR) дозволяє автоматизувати процес збору та зберігання даних, забезпечуючи їх точність і доступність для медичного персоналу. [10].

У даний час системи моніторингу здоров'я, персоналізована медицина, розумні лікарні, безпека лікування, телемедицина, управління хронічними захворюваннями та інтеграція з електронними медичними записами є основними аспектами застосування IoT в медицині. Не зважаючи на численні виклики, IoT має великий потенціал для трансформації медичної галузі та забезпечення кращої якості життя людей у цілому.

1.3 Огляд існуючих систем моніторингу здоров'я

Smart технології дозволяють відстежувати стан здоров'я в реальному часі, що може значно полегшити діагностику та лікування різних хвороб.

В умовах поширення різних захворювань та вірусів наш світ потребує ефективних та новаторських підходів до медичної діагностики, які можуть забезпечити системи моніторингу здоров'я IoT. такі системи дозволяють неперервно контролювати стан здоров'я, вчасно виявляти хвороби та забезпечувати персоналізовану лікарську допомогу.

Системи моніторингу здоров'я на основі IoT можна класифікувати за кількома ознаками, включаючи тип даних, що збираються, місце розташування пацієнта та спосіб передачі даних.

За типом даних:

Системи моніторингу життєво важливих органів. Такі системи використовують:

Датчики серцевого ритму являються одними з найпоширеніших пристроїв у системах моніторингу здоров'я на основі IoT. Вони використовуються для вимірювання серцевої активності та виявлення можливих проблем з серцем. Прикладом таких пристроїв є датчики PPG (Photoplethysmography) Рис 1.1., які використовуються в фітнес-трекерах та розумних годинниках [11].

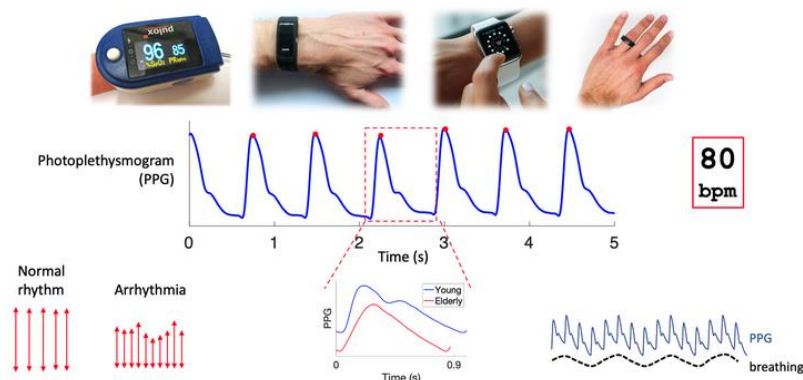


Рисунок 1.1. Датчик PPG (Photoplethysmography)

Датчики артеріального тиску дозволяють моніторити кров'яний тиск пацієнтів у режимі реального часу. Вони можуть бути інтегровані у носимі пристрої або окремі медичні прилади. Це є Omron HeartGuide, Рис 1.2., який використовує технологію надувного манжету для точного вимірювання артеріального тиску[11].



Рисунок 1.2. Датчик Omron HeartGuide

Глюкометри застосовуються при відслідковуванні за рівнем глюкози в крові, Прикладом є системи Dexcom Рис.1.3., вони використовують IoT для збору даних та передачі їх на смартфони або інші пристрої. Це дозволяє пацієнтам з діабетом контролювати свій стан у режимі реального часу [13].



Рисунок 1.3. Система Dexcom

Пульсоксиметри вимірюють рівень насичення крові киснем та використовуються для моніторингу пацієнтів з респіраторними захворюваннями. Вони можуть бути інтегровані у фітнес-трекери або використовуватися як окремі пристрої. Наприклад, Fitbit Sense Рис.1.4. володіє вбудованим датчиком SpO₂ для вимірювання рівня кисню в крові [14].



Рисунок 1.4. Система Fitbit Sense

Датчики температури тіла дозволяють відстежувати температуру пацієнтів та виявляти лихоманку. Прикладом є iHealth Thermometer RT3 Рис.1.5., де використовуються інфрачервоні датчики для безконтактного вимірювання температури [15].



Рисунок 1.5. Система iHealth Thermometer PT3

Системи моніторингу активності: відстежують рівень фізичної активності пацієнта, такі як кроки, пройдені за день, спалені калорії та якість сну [3,4] до них належать фітнес-трекери, такі як Fitbit та Apple Watch Рис.1.6., які використовують IoT технології для моніторингу фізичної активності, сну, серцевого ритму та інших параметрів здоров'я користувачів та передають їх на мобільні додатки, де користувачі можуть аналізувати свої фізичні показники [12].



Рисунок 1.6. Система Fitbit та Apple Watch

Системи моніторингу психічного здоров'я: збирають дані про психічний стан пацієнта, такі як настрій, рівень тривоги та якість сну [5,6].

За місцем розташування пацієнта:

Системи моніторингу вдома: використовуються для дистанційного моніторингу пацієнтів (RPM) що дозволяє лікарям стежити за їх станом на відстані, використовуючи IoT пристрої. Наприклад, система HealthPatch MD

Рис.1.7. від VitalConnect включає носимий датчик, який відстежує серцевий ритм, дихання, температуру тіла та інші параметри, передаючи дані на медичні платформи для подальшого аналізу та прийняття рішень. [4].



Рисунок 1.7. HealthPatch MD

Інтелектуальні інгалятори, такі як Propeller Health Рис.1.8., використовують IoT для моніторингу використання інгаляторів пацієнтами з астмою або хронічною обструктивною хворобою легень (ХОЗЛ). Дані про використання інгалятора передаються на мобільний додаток, що дозволяє пацієнтам та лікарям відстежувати ефективність лікування та виявляти тригери симптомів.



Рисунок 1.8. Система Propeller Health

Системи моніторингу в стаціонарі: використовуються в лікарнях та інших медичних установах для забезпечення безперервного моніторингу стану пацієнтів, управління медичним обладнанням та покращення комфорту пацієнтів. Системи моніторингу життєвих показників, такі як Philips

IntelliVue Рис.1.9. інтегруються з різними датчиками та медичним обладнанням, забезпечуючи лікарів та медсестер актуальною інформацією про стан пацієнтів [16].



Рисунок 1.9. Система Philips IntelliVue

За способом передачі даних:

За способом передачі даних системи можуть бути двох типів: дротові які базуються на дротовому зв'язку передаючи дані за допомогою кабелів, та без провідні, в основі якої лежить бездротовий зв'язок, такі системи передають дані за допомогою бездротових технологій, таких як Bluetooth, Wi-Fi або супутниковий чи мобільний зв'язок.

Компоненти систем моніторингу здоров'я на основі IoT

Системи моніторингу здоров'я на основі IoT зазвичай складаються з наступних компонент:

Датчики: збирають дані про фізіологічні показники пацієнта. Датчики можуть бути носимими, імплантованими або розташованими в навколишньому середовищі пацієнта [14,15].

Пристрій зв'язку: передає дані з датчиків на хмарну платформу. Пристрій зв'язку може бути інтегрований з датчиком або бути окремим пристроєм [16].

Хмарна платформа: зберігає, аналізує та візуалізує дані з датчиків. Хмарна платформа також може генерувати сигнали тривоги та повідомлення для медичних працівників у разі виявлення проблем зі станом здоров'я пацієнта [17,18].

Інтерфейс користувача: дозволяє медичним працівникам та пацієнтам переглядати дані про стан здоров'я та керувати системою. Інтерфейс користувача може бути веб-додатком, мобільним додатком або настільним програмним забезпеченням [19,20].

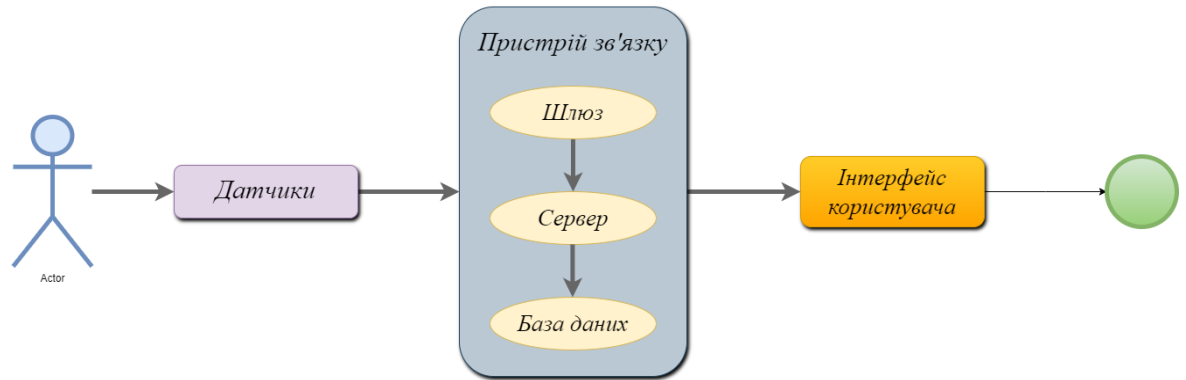


Рисунок 1.10. Система моніторингу здоров'я на основі IoT

Існуючі системи моніторингу здоров'я володіють великою низкою переваг, це надає сучасній медицині можливості до виявлення захворювань на ранніх стадіях. Здійснення постійного моніторингу організму дозволяє виявляти аномалії та зміни у стані здоров'я пацієнтів на ранніх стадіях через що з'являється можливість у своєчасному лікуванні.

На основі персоналізованого лікування проводиться збір та аналіз великих обсягів даних що дозволяє створювати індивідуальні плани лікування, які враховують особливості кожного пацієнта.

Підвищення ефективності медичних послуг за рахунок автоматизації процесів моніторингу та обробки даних знижує навантаження на медичний персонал, дозволяючи їм зосередитись на наданні якісної медичної допомоги.

Впровадження IoT технологій дозволяє знизити витрати на госпіталізацію та тривале лікування завдяки ранньому виявленню захворювань та дистанційному моніторингу.

Системи моніторингу здоров'я на основі IoT відіграють важливу роль у сучасній медицині, забезпечуючи безперервний моніторинг, раннє виявлення захворювань, персоналізоване лікування та підвищення ефективності медичних послуг. Незважаючи на численні виклики, такі як безпека даних, інтероперабельність та масштабованість, IoT має великий потенціал для

подальшого розвитку та впровадження в медичну практику. Важливою складовою успішного впровадження IoT у медицину є розробка надійних технологій та стандартів, що забезпечують безпеку, конфіденційність та сумісність систем.

ПРОЕКТУВАННЯ АПАРАТНОЇ СКЛАДОВОЇ СИСТЕМИ МОНІТОРИНГУ ЗДОРОВ'Я

2.1 Вибір компонент

У сучасному світі моніторинг стану здоров'я є важливою проблемою. Пацієнти страждають від серйозних проблем зі здоров'ям через відсутність належного контролю за здоров'ям. Зараз доступно багато пристроїв IoT для моніторингу здоров'я пацієнтів через Інтернет. Медичні працівники також використовують ці розумні пристрої, щоб стежити за своїми пацієнтами. IoT стрімко революціонізує галузь охорони здоров'я, завдяки сотням нових стартапів у цій галузі.

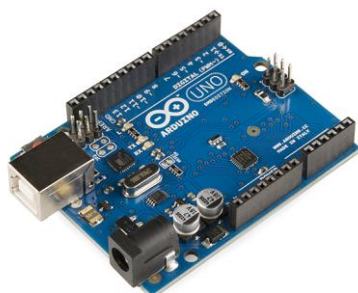
У роботі змодельовано та розроблено систему моніторингу стану здоров'я на основі IOT, яка фіксує частоту пульсу пацієнта та температуру навколишнього середовища. Ця інноваційна технологія має потенціал кардинально змінити підхід до охорони здоров'я, зробивши його доступнішим і зручнішим для людей у всьому світі.

2.1.1 Вибір апаратної платформи ESP8266 та Arduino Uno

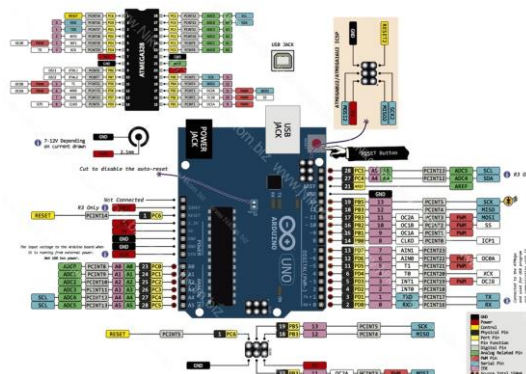
Плата *Arduino Uno* являється одним з найпопулярніших одноплатних мікроконтролерів з відкритим вихідним кодом, на базі яких можна здійснювати розробку цифрових пристроїв та інтерактивних об'єктів, які можуть сприймати та контролювати об'єкти у фізичному світі.

Конструкції плат *Arduino* використовують різні мікропроцесори та елементи керування. Плати оснащені наборами контактів цифрового аналогового входу/виводу (I/O), які можна підключати до різних плат розширення або макетних плат (екранів) та інших схем. Плата має інтерфейси послідовного зв'язку, включаючи Universal_Serial Bus (USB) на деяких моделях, які також використовуються для завантаження програм з персональних комп'ютерів. Рис.2.1. Мікроконтролери зазвичай програмуються з використанням діалекту функцій мов програмування C і C++. На додаток до використання традиційних ланцюжків інструментів

компілятора, проект Arduino забезпечує інтегроване середовище розробки (IDE) на основі проекту мови обробки.



а)



б)

Рисунок. 2.1. а) Плата Мікроконтролера Arduino Uno R3; б) Розпіновка Arduino UNO

Таблиця 2.1 – Технічні характеристики Arduino UNO

№	Параметр	Значення
1.	Мікроконтролер	ATmega328
2.	Робоча напруга	5 В
3.	Вхідна напруга (рекомендована)	7-12 В
4.	Вхідна напруга (гранична)	6-20 В
5.	Цифрові входи/виходи	14 (6 з яких використовуються як виходи ШІМ)
6.	Аналогові входи	6
7.	Максимальний струм навантаження вхід/вихід	40 мА
8.	Максимальний струм навантаження стабілізатора	50 мА
9.	Флеш пам'ять	32 Кб (ATmega328) з яких 0.5 Кб для завантажувача
10.	ОЗУ	2 Кб (ATmega328)
11.	Енергонезалежна пам'ять	1 Кб (ATmega328)
12.	Тактова частота	16 МГц
13.	Порти введення/виведення (I/O)	14 цифрових портів, які можуть бути налаштовані як входи або виходи, і 6 аналогових входів.
14.	Інтерфейси зв'язку	UART (послідовний порт), I2C, SPI
15.	Порт USB	Використовується для програмування та обміну даними з комп'ютером
16.	Роз'єм живлення	Дозволяє живити плату від зовнішнього джерела
17.	Reset Button	Дозволяє перезапустити програму, завантажену в мікроконтролер
16.	Довжина	68.6 мм
14.	Ширина	53.4 мм
15.	Вага	25 грам

Модуль *Wi-Fi ESP 8266* є ідеальним вибором для розробки IoT проектів завдяки своїй низькій вартості, високій продуктивності та багатим комунікаційним можливостям. Він широко використовується для створення різноманітних пристроїв з підключенням до Інтернету, цей мікроконтролер розроблений компанією Espressif Systems. ESP 8266 автономний SOC із вбудованим стеком протоколів TCP/IP, який може надати будь-якому мікроконтролеру доступ до мережі Wi-Fi.

ESP 8266 здатний або розмістити програму, або розвантажити всі функції мережі Wi-Fi з іншого процесора програми.

ESP 8266 володіє вбудованими потужними можливостями для обробки та зберігання даних, які дозволяють інтегрувати його з датчиками та іншими пристроями, призначеними для певних програм, через GPIO з мінімальною попередньою розробкою та мінімальним завантаженням під час виконання. Його високий ступінь вбудованої інтеграції забезпечує мінімальну кількість зовнішніх схем, а все рішення, включаючи зовнішній модуль, розроблено таким чином, щоб займати мінімальну площу друкованої плати.

Цей модуль оснащений вбудованим USB-роз'ємом і багатим набором контактів. За допомогою кабелю micro USB підключається NodeMCU devkit до ПК за рахунок чого здійснюється його прошивка аналогічно до Arduino.

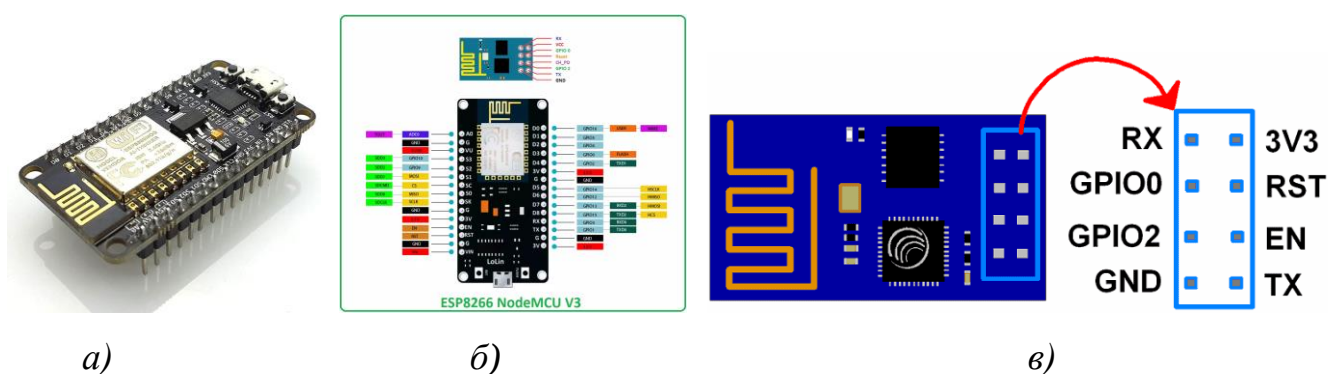


Рисунок. 2.2. а) Плата Модуля Wi-Fi ESP 8266; б) Розпіновка ESP 8266; в) Контактна схема ESP8266

Таблиця 2.2 – Технічні характеристики ESP8266

№	Параметр	Значення
Технічні характеристики		
1.	Мікроконтролер	Tensilica L106 RISC 32-бітний
2.	Тактова частота	80 або 160 MHz.
3.	Оперативна пам'ять (RAM)	64 KB інструкційного кешу, 96 KB даних.
4.	Флеш-пам'ять	Від 512 KB до 4 MB залежно від версії.
5.	Напруга живлення	3.0-3.6V.
Комунікаційні можливості		
6.	Wi-Fi	Підтримує стандарти 802.11 b/g/n, працює на частоті 2.4 GHz.
7.	Режими роботи Wi-Fi	Station (STA), Access Point (AP), та Station+AP.
8.	Безпека	WPA/WPA2.
9.	Протоколи	TCP/IP, UDP, HTTP, FTP, MQTT.
Інтерфейси введення/виведення		
10.	GPIO (General Purpose Input/Output)	17 доступних пінів.
11.	SPI (Serial Peripheral Interface)	Використовується для високошвидкісного зв'язку з периферійними пристроями.
12.	I2C	Для підключення датчиків і периферійних пристроїв.
13.	UART	Для серійного зв'язку з комп'ютером або іншими мікроконтролерами.
14.	ADC (Analog-to-Digital Converter):	10-бітний ADC для вимірювання аналогових сигналів.
15.	Довжина	24.8 мм
16.	Ширина	14.3 мм

2.1.2. Датчики та інші апаратні компоненти системи моніторингу

У даному проекті для створення системи моніторингу стану здоров'я на основі ІОТ. Використовувались наступні компоненти:

Датчик вимірювання артеріального тиску: Рис. 2.3.



а)



б)

Рисунок. 2.3. а) Датчик для вимірювання артеріального тиску; б) Вигляд датчика пульсу зверху та ззаду

Датчик пульсу сумісний з *Arduino* та поєднує в собі базовий оптичний датчик частоти серцевих скорочень із схемою підсилення та шумозаглушення, що дозволяє швидко та легко зчитувати пульс. У центрі сенсорного модуля також є світлодіод, який допомагає визначити серцебиття. Під світлодіодом розташована схема усунення шуму, яка має запобігати впливу шуму на показання.



LM35

1 4-20V
2 OUT
3 GND

LM 35 – це датчик температури, який видає аналоговий сигнал, пропорційний миттєвій температурі. Вихідну напругу можна легко інтерпретувати для отримання показників температури в градусах Цельсія.

РК-дисплей для відображення даних у реальному часі

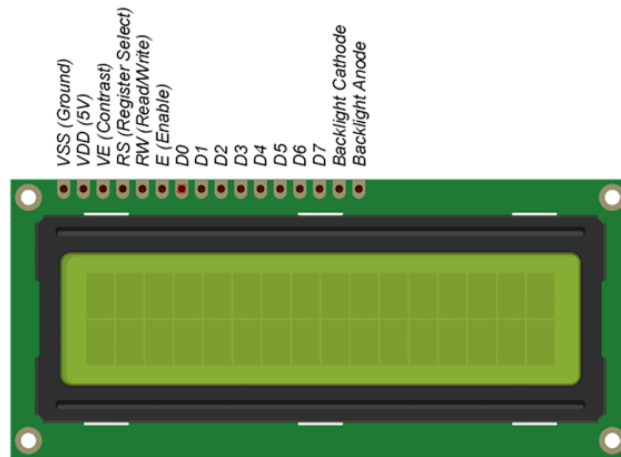


Рисунок. 2.5. РК-дисплей

LCD (рідкокристалічний дисплей) — це технологія, яка використовується для відображення в ноутбуках та інших менших комп'ютерах. Подібно до світлодіодних (LED) і газоплазмових технологій, РК-дисплеї дозволяють робити дисплеї набагато тоншими, ніж світлодіодні та газові дисплеї, оскільки вони працюють за принципом блокування світла, а не його випромінювання. РК-дисплей виготовляється або з пасивною матрицею, або з активною матрицею. РК-дисплей з активною матрицею також відомий як тонкоплівковий транзисторний (TFT) дисплей. Пасивний матричний LCD має сітку провідників з пікселями, розташованими на кожному перетині сітки. Струм надсилається через два провідники в сітці, щоб контролювати світло будь-якого пікселя.

Активна матриця має транзистор, розташований на перетині кожного пікселя, що потребує менше струму для керування яскравістю пікселя. З цієї причини струм на дисплеї з активною матрицею можна часто вмикати та вимикати, покращуючи час оновлення екрана.

Пасивні елементи:

Потенціометри - це змінні резистори, опір яких можна змінювати за допомогою ручки або повзунка Рис.2.6.. Потенціометри використовуються для керування багатьма речами, включно з тим, наскільки яскравим чи тьмяним є світло у вашому домі та регуляторами гучності на вашому телевізорі. Деякі з них використовуються в подільниках напруги.

Потенціометри є пасивними електронними компонент. Вони працюють, змінюючи положення ковзного контакту через рівномірний опір. У потенціометрі вся вхідна напруга прикладається по всій довжині резистора, а вихідна напруга є результатом падіння напруги між фіксованим і ковзним

Резистори є пасивними елемент електричних кіл, вони можуть мати постійні чи змінні значення опору, основним їх призначенням являється лінійне перетворення сили струму в напругу і навпаки, також резистори здійснюють обмеження струму чи поглинання електричної енергії.

При виборі резисторів звертається увага на їх характеристики а саме: габаритні розміри, опір, надійність, стійкість до зовнішніх факторів, ціна.

У роботі використані вуглецеві плівкові резистори 360R серії CFR, які виготовляються шляхом нанесення однорідної плівки з чистого вуглецю на стержні з високоякісної кераміки.

Перемички– це з'єднувальні дроти, у яких на кожному кінці є штирі з'єднувача, що дозволяє використовувати їх для з'єднання двох точок одна з одною без пайки. Вони використовуються з макетними платами та іншими інструментами для створення прототипів, за для полегшення зміни схеми при потребі.

2.2. Схеми підключення системи моніторингу здоров'я

Основним ядром даного проєкту є мікроконтролер Arduino Uno (платформа електронного прототипу з відкритим кодом). Він виконує всі арифметичні та логічні операції та керує всіма підключеними до нього периферійними пристроями відповідно до написаного коду. Головна мета роботи є розробка системи моніторингу здоров'я пацієнтів. Для цього ми використовуємо датчик температури LM 35, та пульсоксиметр MAX 30100.

Датчик температури LM 35 підключається до аналогового входу A0 мікроконтролера Arduino Uno, щоб подати зміну напруги в датчику..

Датчик пульсу підключається до контакту A2, щоб подати вихідний сигнал на *Arduino*. Щоб виміряти температуру пацієнта, ми використали мікросхему *LM 35*,

Щоб відобразити повідомлення відповідно до жесту та стану здоров'я пацієнта, використовується РК-дисплей 16*2. 16-бітні дані, які на ньому відображаються, інформація надсилається через контакт (4,5,6,7) від *Arduino*. Крім того, щоб повідомити про стан здоров'я пацієнтів членів їхніх родин та інших близьких людей, ми надсилаємо дані на інтернет-сервер. Для цього ми використали модуль *Wi-Fi ESP 8266* який підключається до Інтернету та надсилає дані на сервер пристрою IoT.

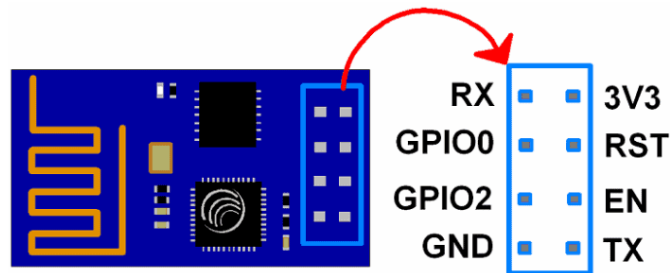


Рисунок. 2.6. Контактна схема ESP8266

Вивід 1: Заземлення – підключено до заземлення схеми.

Вивід 2: Tx/GPIO – 1 підключено до виводу Rx програматора/uC для завантаження програми.

Вивід 3: GPIO – 2 вхід/вихід загального призначення.

Вивід 4: CH_EN Увімкнути мікросхему/активний високий

контакт 5: Flash/GPIO – 0 вхідний/вихідний контакт загального призначення.

Контакт 6: скинути модуль.

Вивід 7: RX/GPIO – 3 вхідний/вихідний контакт загального призначення. Тільки +3,3 В

Як IoT-сервер використовується *Thing Speak*.

Схема підключення та взаємодії датчиків до мікроконтролера Arduino Uno, з використанням Wi-Fi модуля ESP8266 для передачі даних на платформу ThingSpeak, має вигляд Рис. 2.7.

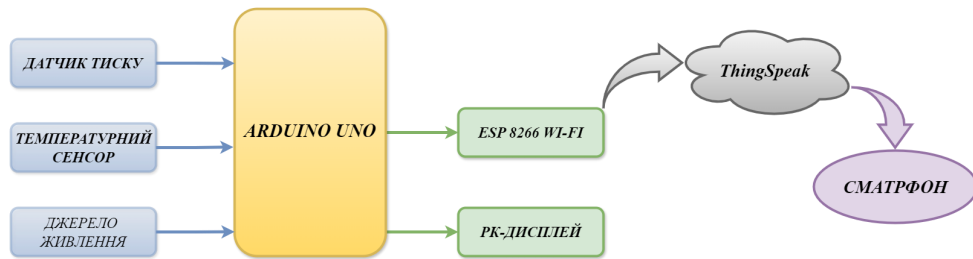


Рисунок. 2.7. Структурна схема підключення датчиків до плати Arduino Uno
Принципова схема моніторингу здоров'я пацієнта на основі ІОТ Рис. 2.8..

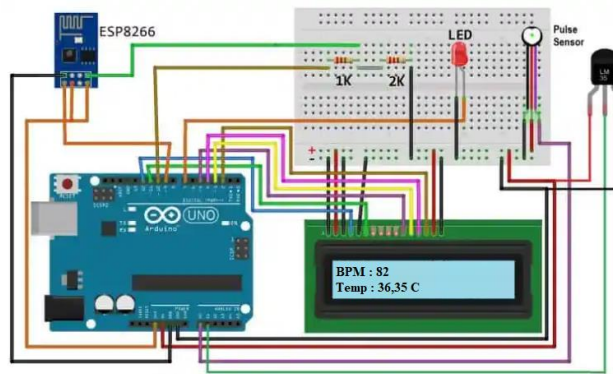


Рисунок. 2.8. Макетна схема моніторингу здоров'я пацієнта на основі ІОТ

На електричній схемі показано, як з'єднати компоненти. Ми підключимо датчик пульсу і датчик температури LM 35 до плати Arduino, а модуль Wi-Fi ESP 8266 до плати. Ми також додамо РК-дисплей для відображення даних у реальному часі та світлодіод для індикації.

Як видно з Рис. 2.8. Вихідний контакт датчика імпульсу підключається до A0 Arduino, а інші два контакти – до VCC і GND. Вихідний контакт датчика температури LM 35 під'єднується до A1 мікроконтролера Arduino, а інші два контакти – до VCC і GND. Світлодіод підключається до цифрового контакту 7 Arduino через резистор на 220 Ом, контакти 1,3,5,16 РК-дисплея до GND, контакти 2,15 LCD до VCC, контакти 4,6,11,12,13,14 РК-дисплея до цифрових контактів 12,11,5,4,3,2 Arduino. Вивід RX ESP 8266 працює на 3,3 В, і він не під'єднується до Arduino, якщо підключати його безпосередньо до Arduino. Тому для нього створено дільник напруги, який перетворить 5 В на

ПРОЕКТУВАННЯ ПРОГРАМНОЇ СКЛАДОВОЇ СИСТЕМИ МОНІТОРИНГУ ЗДОРОВ'Я

3.1 Структура програмного забезпечення та хмарного сховища для системи IoT

ThingSpeak – це платформа для збереження і обробки даних, отриманих від пристроїв Інтернету речей (IoT). Вона дозволяє збирати, візуалізувати та аналізувати дані в режимі реального часу.

Основними функціями ThingSpeak є:

Збір даних: Дозволяє отримувати дані з різних датчиків та пристроїв IoT за допомогою HTTP-протоколу або MQTT.

Збереження даних: Платформа надає можливість зберігати дані у каналах, кожен з яких має до восьми полів для різних типів даних.

Візуалізація: Користувачі можуть створювати графіки та діаграми для візуалізації даних у режимі реального часу.

Аналіз: ThingSpeak підтримує вбудований аналіз даних за допомогою MATLAB, що дозволяє користувачам виконувати складні обчислення та алгоритми прямо на платформі.

Інтеграція з іншими сервісами: Може бути інтегрована з іншими хмарними сервісами, такими як Twitter, Twilio та IFTTT, для створення комплексних IoT-рішень.

Архітектура ThingSpeak

Архітектура ThingSpeak, орієнтована на канали. Кожен користувач може створювати один або кілька каналів, де кожен канал зберігає дані від окремого пристрою або набору пристроїв. Кожен канал має унікальний ключ API, який використовується для аутентифікації запитів на запис або читання даних.

ThingSpeak можна застосовувати для моніторингу довкілля здійснювати збір даних про температуру, вологість, рівень забруднення та інші параметри з датчиків навколишнього середовища. У системах розумний дім для управління і моніторингу системами освітлення, опалення, чи

охоронними системами. Промисловому IoT як систем відстеження стану обладнання, збору даних про продуктивність та запобігання несправностей.

ThingSpeak є потужною та гнучкою платформою для розробників і інженерів IoT, які потребують інструментів для збору, зберігання, візуалізації та аналізу даних. Тому дана платформа була вибрана у якості хмарного сховища.

3.2 Програмування ESP8266 з використанням Arduino IDE

Програмування системи передбачає збір, відображення та передачу даних про тиск і температуру пацієнтів вимірюваних за допомогою мікроконтролера Arduino Uno, дисплея LCD, сенсорів та Wi-Fi модуля ESP 8266.

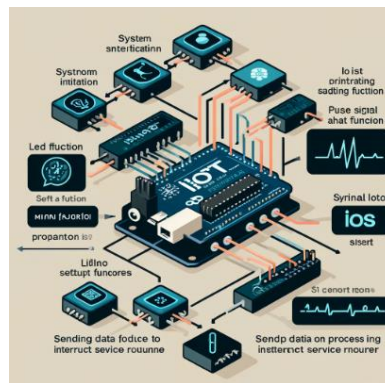


Рисунок. 3.1. Блок-схема основних функціональних блоків та їх взаємодії у програмі.

Основні компоненти коду:

1. Імпорт бібліотек та ініціалізація об'єктів:

Підключаємо бібліотеки для роботи з LCD дисплеєм та програмним послідовним портом, а також ініціалізуємо об'єкти для цих компонентів.

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

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
SoftwareSerial ser(9, 10);
```

2. *Оголошення змінних*

Здійснюється зберігання значень температури, тиску, а також налаштування пінів.

```
float temp = 0;
volatile float pulse = 0;
String apiKey = "00707TGA1BLUNN12";

int pulsePin = A0;
int blinkPin = 7;
int fadePin = 13;
int fadeRate = 0;
```

3. *Ініціалізація пульсу та інших змінних:*

Оголошуються змінні для обробки сигналу пульсу, що використовуються в ISR (Interrupt Service Routine).

```
volatile int BPM;
volatile int Signal;
volatile int IBI = 600;
volatile boolean Pulse = false;
volatile boolean QS = false;

static boolean serialVisual = true;
volatile int rate[10];
volatile unsigned long sampleCounter = 0;
volatile unsigned long lastBeatTime = 0;
volatile int P = 512;
volatile int T = 512;
volatile int thresh = 525;
volatile int amp = 100;
volatile boolean firstBeat = true;
volatile boolean secondBeat = false;
```

4. *Функція setup():*

У функції `setup()` ініціалізуються LCD-дисплей, піни для LED, послідовний порт, а також модуль ESP8266 для підключення до Wi-Fi.

```
void setup() {
  lcd.begin(16, 2);
  pinMode(blinkPin, OUTPUT);
  pinMode(fadePin, OUTPUT);
  Serial.begin(115200);
  interruptSetup();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" Patient Health");
  lcd.setCursor(0, 1);
  lcd.print(" Monitoring ");
}
```

```

delay(4000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Initializing....");
delay(5000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Getting Data....");
ser.begin(9600);
ser.println("AT");
delay(1000);
ser.println("AT+GMR");
delay(1000);
ser.println("AT+CWMODE=3");
delay(1000);
ser.println("AT+RST");
delay(5000);
ser.println("AT+CIPMUX=1");
delay(1000);
String cmd = "AT+CWJAP=\"Home\", \"98765432\"";
ser.println(cmd);
delay(1000);
ser.println("AT+CIFSR");
delay(1000);
}

```

5. *Основний цикл loop() :*

В основному циклі відбувається вивід даних у серійний порт, керування LED, зчитування температури та відправка даних на сервер ThingSpeak.

```

void loop() {
  serialOutput();
  if (QS == true) {
    fadeRate = 255;
    serialOutputWhenBeatHappens();
    QS = false;
  }
  ledFadeToBeat();
  delay(20);
  read_temp();
  esp_8266();
}

```

6. *Функція керування LED:*

Функція ledFadeToBeat() керує поступовим затуханням LED, що імітує биття серця.

```

void ledFadeToBeat() {
  fadeRate -= 15;
  fadeRate = constrain(fadeRate, 0, 255);
}

```

```

    analogWrite(fadePin, fadeRate);
}

```

7. *Налаштування переривань:*

Функція `interruptSetup()` налаштовує таймер для генерації переривань кожні 2 мс для зчитування даних з сенсора пульсу.

```

void interruptSetup() {
    TCCR2A = 0x02;
    TCCR2B = 0x06;
    OCR2A = 0x7C;
    TIMSK2 = 0x02;
    sei();
}

```

8. *Обробка серійного виводу:*

Ці функції керують виводом даних у серійний порт у випадку виявлення серцевого ритму.

```

void serialOutput() {
    if (serialVisual == true) {
        arduinoSerialMonitorVisual('-', Signal);
    } else {
        sendDataToSerial('S', Signal);
    }
}

void serialOutputWhenBeatHappens() {
    if (serialVisual == true) {
        Serial.print("*** Heart-Beat Happened *** ");
        Serial.print("BPM: ");
        Serial.println(BPM);
    } else {
        sendDataToSerial('B', BPM);
        sendDataToSerial('Q', IBI);
    }
}

```

9. *Обробка сигналу сенсора:*

Цей обробник переривань зчитує значення з сенсора пульсу та обробляє їх для визначення серцевого ритму (BPM) та інтервалу між ударами (IBI).

```

ISR(TIMER2_COMPA_vect) {
    cli();
    Signal = analogRead(pulsePin);
    sampleCounter += 2;
    int N = sampleCounter - lastBeatTime;

    if (Signal < thresh && N > (IBI / 5) * 3) {
        if (Signal < T) {

```

```

    T = Signal;
}
}

if (Signal > thresh && Signal > P) {
    P = Signal;
}

if (N > 250) {
    if ((Signal > thresh) && (Pulse == false) && (N > (IBI / 5)
* 3)) {
        Pulse = true;
        digitalWrite(blinkPin, HIGH);
        IBI = sampleCounter - lastBeatTime;
        lastBeatTime = sampleCounter;

        if (secondBeat) {
            secondBeat = false;
            for (int i = 0; i <= 9; i++) {
                rate[i] = IBI;
            }
        }

        if (firstBeat) {
            firstBeat = false;
            secondBeat = true;
            sei();
            return;
        }

        word runningTotal = 0;
        for (int i = 0; i <= 8; i++) {
            rate[i] = rate[i + 1];
            runningTotal += rate[i];
        }

        rate[9] = IBI;
        runningTotal += rate[9];
        runningTotal /= 10;
        BPM = 60000 / runningTotal;
        QS = true;
        pulse = BPM;
    }
}

if (Signal < thresh && Pulse == true) {
    digitalWrite(blinkPin, LOW);
    Pulse = false;
    amp = P - T;
    thresh = amp / 2 + T;
    P = thresh;
    T = thresh;
}

```

```

    if (N > 2500) {
        thresh = 512;
        P = 512;
        T = 512;
        lastBeatTime = sampleCounter;
        firstBeat = true;
        secondBeat = false;
    }

    sei();
}

```

10. Відправка даних на ThingSpeak:

Ця функція встановлює TCP-з'єднання з сервером ThingSpeak та відправляє дані про температуру та пульс.

```

void esp_8266() {
    String cmd = "AT+CIPSTART=4,\"TCP\", \"\"";
    cmd += "184.106.153.149";
    cmd += "\",80";
    ser.println(cmd);
    Serial.println(cmd);

    if (ser.find("Error")) {
        Serial.println("AT+CIPSTART error");
        return;
    }

    String getStr = "GET /update?api_key=";
    getStr += apiKey;
    getStr += "&field1=";
    getStr += String(temp);
    getStr += "&field2=";
    getStr += String(pulse);
    getStr += "\r\n\r\n";

    cmd = "AT+CIPSEND=4, ";
    cmd += String(getStr.length());
    ser.println(cmd);
    Serial.println(cmd);
    delay(1000);
    ser.print(getStr);
    Serial.println(getStr);
    delay(3000);
}

```

11. Зчитування температури:

Функція `read_temp()` зчитує значення з температурного сенсора, конвертує його у градуси Цельсія, виводить на серійний порт та відображає на LCD-дисплеї.

```

void read_temp() {
    int temp_val = analogRead(A1);
    float mv = (temp_val / 1024.0) * 5000;
    float cel = mv / 10;
    temp = cel; // Температура у градусах Цельсія
    Serial.print("Temperature:");
    Serial.println(temp);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("BPM :");
    lcd.setCursor(7, 0);
    lcd.print(BPM);
    lcd.setCursor(0, 1);
    lcd.print("Temp.");
    lcd.setCursor(7, 1);
    lcd.print(temp);
    lcd.setCursor(13, 1);
    lcd.print("C");
}

```

Цей код написаний на Arduino і створює систему моніторингу здоров'я пацієнта, яка зчитує та обробляє дані про пульс і температуру, відображає їх на LCD-дисплеї та передає на сервер ThingSpeak для подальшого аналізу та зберігання.

Після завантаження коду на плату Arduino з'являються дані на РК-екрані, щоб надсилати дані через Інтернет, створюється канал на ThingSpeak. Після входу в обліковий запис, у новому каналі додаються віджети для частоти пульсу та температури тіла. Здійснюється налаштування віджетів, встановивши мінімальні та максимальні значення, одиниці вимірювання та інтервали оновлення. Наприклад, ми можемо налаштувати частоту пульсу на оновлення кожні 15 секунд. Після того як налаштували віджети, копіюється ключ API і вставляється в програму Arduino, потім завантажуються код на плату Arduino, а дані буде надіслано до ThingSpeak. де можна побачити частоту пульсу та температуру тіла, які відображаються на графіках, оновлюючись у реальному часі. Доступ до цих даних можна отримати з будь-якої точки світу, що робить їх потужним інструментом для віддаленого моніторингу охорони здоров'я.

ВИСНОВКИ

Інтернет речей має великий потенціал для подальшого розвитку в медичній галузі. Зокрема, розвиток технологій 5G, штучного інтелекту (AI) та машинного навчання відкриває нові можливості для створення інтелектуальних систем моніторингу здоров'я. Однією з ключових тенденцій є розвиток периферійних обчислень (edge computing), які дозволяють обробляти дані безпосередньо на пристроях або у їхньому безпосередньому оточенні, зменшуючи затримки та підвищуючи ефективність систем.

У сучасному світі моніторинг стану здоров'я є важливою проблемою. Пацієнти страждають від серйозних проблем зі здоров'ям через відсутність належного контролю за здоров'ям. Зараз доступно багато пристроїв IoT для моніторингу здоров'я пацієнтів через Інтернет. Медичні працівники також використовують ці розумні пристрої, щоб стежити за своїми пацієнтами. IoT стрімко революціонізує галузь охорони здоров'я, завдяки сотням нових стартапів у галузі охорони здоров'я, які використовують систему моніторингу здоров'я пацієнтів

У роботі створено систему моніторингу стану здоров'я на основі IOT, яка реєструє частоту пульсу пацієнта та температуру навколишнього середовища. Ця система не тільки записує ці дані, але й оновлює їх на платформі IOT.

Платформою IOT вибрано Thing Speak. Thing Speak — це програма з відкритим кодом Інтернету речей (IoT) і API для зберігання та отримання даних із речей за допомогою протоколу HTTP через Інтернет або через локальну мережу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ashton, K. That 'Internet of Things' Thing. RFID Journal. (2009).
2. Islam, S. M. The Internet of Things for Health Care: A Comprehensive Survey. R., Kwak, D., Kabir, M. H., Hossain, M., & Kwak, K.-S. (2015). IEEE Access, 3, 678–708.
3. Gubbi J., Buyya R, Marusic S, Internet of Things (IoT): A vision, architectural elements, and future directions V. 29, 7, 2013, P. 1645-1660
4. Kumar, S., & Lee, H.-J. Security Issues in Healthcare Applications Using Wireless Medical Sensor Networks: A Survey. Sensors, (2012). 12(1), 55–91.
5. Topol, E. Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again. Basic Books. (2019).
6. Dimitrov, D. V. Medical Internet of Things and Big Data in Healthcare. Healthcare Informatics Research, (2016). 22(3), 156–163.
7. Li, X., Da Xu, L., & Zhao, S. 5G Internet of Things: A Survey. Journal of Industrial Information Integration, (2017). 10, 1–9.
8. Sood, M., Khandelwal, D., & Awasthi, A. Telemedicine in India: A Tool for Transforming Health Care in Rural and Remote Areas. Postgraduate Medical Journal, (2015). 91(1079), 615–620.
9. Pang, Z., Zheng, L., Tian, J., Kao-Walter, S., Dubrova, E., & Chen, Q. (2015). Design of a Terminal Solution for Integration of In-home Health Care Devices and Services towards the Internet-of-Things. Enterprise Information Systems, 9(1), 86–116.
10. Aga, D. C., & Sumari, P. (2020). A Review of Internet of Things for Healthcare Management. Journal of Physics: Conference Series, 1517, 012027.
11. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, Privacy and Trust in Internet of Things: The Road Ahead.

Computer Networks, 76, 146–164.

12. Wright, S. P., Hall Brown, T. S., Collier, S. R., & Sandberg, K. (2017). How consumer physical activity monitors could transform human physiology research. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 312(3), R358-R367.

13. Heinemann, L., Fleming, G. A., Petrie, J. R., Holl, R. W., Bergenstal, R. M., & Peters, A. L. (2018). Blood glucose monitoring: current and future applications. *Journal of Diabetes Science and Technology*, 12(1), 115-123.

14. Fitbit. (2020). Introducing Fitbit Sense: A New Era of Health. Retrieved from <https://www.fitbit.com/global/us/products/smartwatches/sense>

15. iHealth. (2020). iHealth Thermometer PT3: No-Touch Forehead Thermometer. Retrieved from <https://ihealthlabs.com/thermometer-pt3>

16. Schulz, T., & Fuglerud, K. S. (2015). Trends in health care and patient monitoring systems. In *Wireless Mobile Communication and Healthcare* (pp. 319-326). Springer, Cham.

17. Kumar, S., Nilsen, W. J., Abernethy, A., Atienza, A., Patrick, K., Pavel, M., ... & Hesse, B. (2016). Mobile health technology evaluation: The mHealth evidence workshop. *American Journal of Preventive Medicine*, 45(2), 228-236.

18. Zheng, Y., Mehrotra, S., & Tsudik, G. (2011). A privacy framework for mobile health and home-care systems. In *Proceedings of the ACM SIGSAC conference on Computer & communications security* (pp. 184-195).

19. Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*, 56, 684-700.

ДОДАТКИ

Додаток А

Лістинг програми

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
SoftwareSerial ser(9, 10);

float temp = 0;
volatile float pulse = 0; // Змінна тепер volatile
String apiKey = "00707TGA1BLUNN12";

// Змінні
int pulsePin = A0;
int blinkPin = 7;
int fadePin = 13;
int fadeRate = 0;

volatile int BPM;
volatile int Signal;
volatile int IBI = 600;
volatile boolean Pulse = false;
volatile boolean QS = false;

static boolean serialVisual = true;
volatile int rate[10];
volatile unsigned long sampleCounter = 0;
volatile unsigned long lastBeatTime = 0;
volatile int P = 512;
volatile int T = 512;
volatile int thresh = 525;
volatile int amp = 100;
volatile boolean firstBeat = true;
volatile boolean secondBeat = false;
```

					Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino.	Лім.			Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата						
Розроб.		Скрипник Ю.В.								
Перевір.		Ташук О.Ю.								
Т. Контр.						Арк.			Аркуші 1	
Реценз.						Циклова комісія Комп'ютерної інженерії				
Н. Контр.										
Затверд.		Ташук О.Ю.								

```

void setup() {
  lcd.begin(16, 2);
  pinMode(blinkPin, OUTPUT);
  pinMode(fadePin, OUTPUT);
  Serial.begin(115200);
  interruptSetup();

  // IF YOU ARE POWERING The Pulse Sensor AT VOLTAGE LESS THAN
  THE BOARD VOLTAGE,
  // UN-COMMENT THE NEXT LINE AND APPLY THAT VOLTAGE TO THE A-
  REF PIN
  // analogReference(EXTERNAL);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(" Patient Health");
  lcd.setCursor(0, 1);
  lcd.print(" Monitoring ");
  delay(4000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Initializing....");
  delay(5000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Getting Data....");
  ser.begin(9600);
  ser.println("AT");
  delay(1000);
  ser.println("AT+GMR");
  delay(1000);
  ser.println("AT+CWMODE=3");
  delay(1000);
  ser.println("AT+RST");
  delay(5000);
  ser.println("AT+CIPMUX=1");
  delay(1000);

```

					Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino.					Лім.		Маса		Масштаб		
Змн.	Арк.	№ докум.	Підпис	Дата												
Розроб.		Скрипник Ю.В.														
Перевір.		Ташук О.Ю.														
Т. Контр.																
Реценз.										Арк.		Аркушів 1				
Н. Контр.										Циклова комісія Комп'ютерної інженерії						
Затверд.		Ташук О.Ю.														

```

String cmd = "AT+CWJAP=\"Alexahome\", \"98765432\"";
ser.println(cmd);
delay(1000);
ser.println("AT+CIFSR");
delay(1000);
}

void loop() {
  serialOutput();
  if (QS == true) {
    fadeRate = 255;
    serialOutputWhenBeatHappens();
    QS = false;
  }
  ledFadeToBeat();
  delay(20);
  read_temp();
  esp_8266();
}

void ledFadeToBeat() {
  fadeRate -= 15;
  fadeRate = constrain(fadeRate, 0, 255);
  analogWrite(fadePin, fadeRate);
}

void interruptSetup() {
  TCCR2A = 0x02;
  TCCR2B = 0x06;
  OCR2A = 0x7C;
  TIMSK2 = 0x02;
  sei();
}

void serialOutput() {
  if (serialVisual == true) {
    arduinoSerialMonitorVisual('-', Signal);
  } else {
    sendDataToSerial('S', Signal);
  }
}

```


```

void serialOutputWhenBeatHappens() {
  if (serialVisual == true) {
    Serial.print("*** Heart-Beat Happened *** ");
    Serial.print("BPM: ");
    Serial.println(BPM);
  } else {
    sendDataToSerial('B', BPM);
    sendDataToSerial('Q', IBI);
  }
}

void arduinoSerialMonitorVisual(char symbol, int data) {
  const int sensorMin = 0;
  const int sensorMax = 1024;
  int sensorReading = data;
  int range = map(sensorReading, sensorMin, sensorMax, 0, 11);

  switch (range) {
    case 0: Serial.println(""); break;
    case 1: Serial.println("---"); break;
    case 2: Serial.println("-----"); break;
    case 3: Serial.println("-----"); break;
    case 4: Serial.println("-----"); break;
    case 5: Serial.println("-----|"); break;
    case 6: Serial.println("-----|---"); break;
    case 7: Serial.println("-----|-----"); break;
    case 8: Serial.println("-----|-----"); break;
    case 9: Serial.println("-----|-----"); break;
    break;
    case 10: Serial.println("-----|-----");
  }; break;
  case 11: Serial.println("-----|-----");
  break;
}

void sendDataToSerial(char symbol, int data) {
  Serial.print(symbol);
  Serial.println(data);
}

```

Змн.	Арк.	№ докум.	Підпис	Дата	Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino.	<i>Лім.</i>			<i>Маса</i>	<i>Масштаб</i>
Розроб.		Скрипник Ю.В.								
Перевір.		Ташук О.Ю.								
Т. Контр.										
Реценз.						Арк.			Аркушів 1	
Н. Контр.						Циклова комісія Комп'ютерної інженерії				
Затверд.		Ташук О.Ю.								

```
ISR(TIMER2_COMPA_vect) {
    cli();
    Signal = analogRead(pulsePin);
    sampleCounter += 2;
    int N = sampleCounter - lastBeatTime;

    if (Signal < thresh && N > (IBI / 5) * 3) {
        if (Signal < T) {
            T = Signal;
        }
    }

    if (Signal > thresh && Signal > P) {
        P = Signal;
    }

    if (N > 250) {
        if ((Signal > thresh) && (Pulse == false) && (N > (IBI / 5)
* 3)) {
            Pulse = true;
            digitalWrite(blinkPin, HIGH);
            IBI = sampleCounter - lastBeatTime;
            lastBeatTime = sampleCounter;

            if (secondBeat) {
                secondBeat = false;
                for (int i = 0; i <= 9; i++) {
                    rate[i] = IBI;
                }
            }

            if (firstBeat) {
                firstBeat = false;
                secondBeat = true;
                sei();
                return;
            }
        }
    }
}
```

					Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino.				Lім.	Маса	Масштаб
Zmn.	Apk.	Nº докум.	Підпис	Дата							
Розроб.		Скрипник Ю.В.									
Перевір.		Ташук О.Ю.									
T. Контр.											
Реценз.						Aрк.			Аркуші 1		
H. Контр.						Циклова комісія Комп’ютерної інженерії					
Затверд.		Ташук О.Ю.									

```

word runningTotal = 0;
for (int i = 0; i <= 8; i++) {
    rate[i] = rate[i + 1];
    runningTotal += rate[i];
}

```

</														

```

        rate[9] = IBI;
        runningTotal += rate[9];
        runningTotal /= 10;
        BPM = 60000 / runningTotal;
        QS = true;
        pulse = BPM;
    }
}

if (Signal < thresh && Pulse == true) {
    digitalWrite(blinkPin, LOW);
    Pulse = false;
    amp = P - T;
    thresh = amp / 2 + T;
    P = thresh;
    T = thresh;
}

if (N > 2500) {
    thresh = 512;
    P = 512;
    T = 512;
    lastBeatTime = sampleCounter;
    firstBeat = true;
    secondBeat = false;
}

sei();
}

void esp_8266() {
    String cmd = "AT+CIPSTART=4,\"TCP\", \"\"";
    cmd += "184.106.153.149";
    cmd += "\",80";
    ser.println(cmd);
    Serial.println(cmd);

    if (ser.find("Error")) {
        Serial.println("AT+CIPSTART error");
        return;
    }
}

```

					Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino.					Лім.		Маса		Масштаб		
Змн.	Арк.	№ докум.	Підпис	Дата												
Розроб.		Скрипник Ю.В.														
Перевір.		Ташук О.Ю.														
Т. Контр.																
Реценз.										Арк.		Аркуші 1				
Н. Контр.										Циклова комісія Комп'ютерної інженерії						
Затверд.		Ташук О.Ю.														

```

String getStr = "GET /update?api_key=";
getStr += apiKey;
getStr += "&field1=";
getStr += String(temp);
getStr += "&field2=";
getStr += String(pulse);
getStr += "\r\n\r\n";

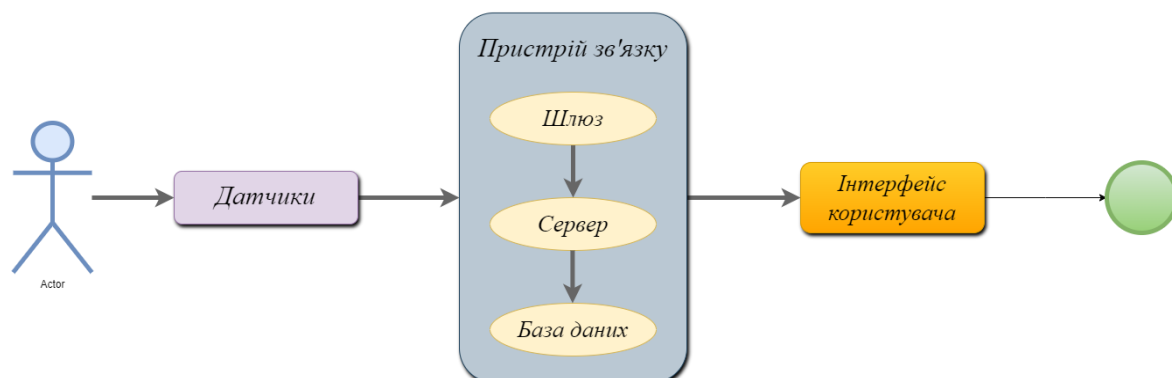
cmd = "AT+CIPSEND=4,";
cmd += String(getStr.length());
ser.println(cmd);
Serial.println(cmd);
delay(1000);
ser.print(getStr);
Serial.println(getStr);
delay(3000);
}

void read_temp() {
  int temp_val = analogRead(A1);
  float mv = (temp_val / 1024.0) * 5000;
  float cel = mv / 10;
  temp = cel; // Температура у градусах Цельсія
  Serial.print("Temperature:");
  Serial.println(temp);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("BPM :");
  lcd.setCursor(7, 0);
  lcd.print(BPM);
  lcd.setCursor(0, 1);
  lcd.print("Temp.");
  lcd.setCursor(7, 1);
  lcd.print(temp);
  lcd.setCursor(13, 1);
  lcd.print("C");
}

```

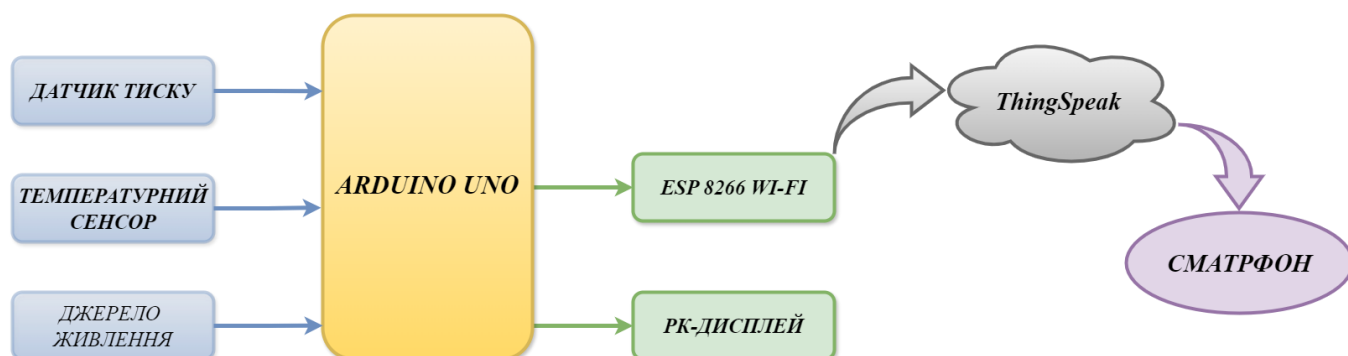
					Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino.	Літ.			Маса		Масштаб	
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Скрипник Ю.В.										
Перевір.		Ташук О.Ю.										
Т. Контр.												
Реценз.						Арк.			Аркушів			1
Н. Контр.						Циклова комісія Комп'ютерної інженерії						
Затверд.		Ташук О.Ю.										

Схема системи моніторингу здоров'я на основі IoT



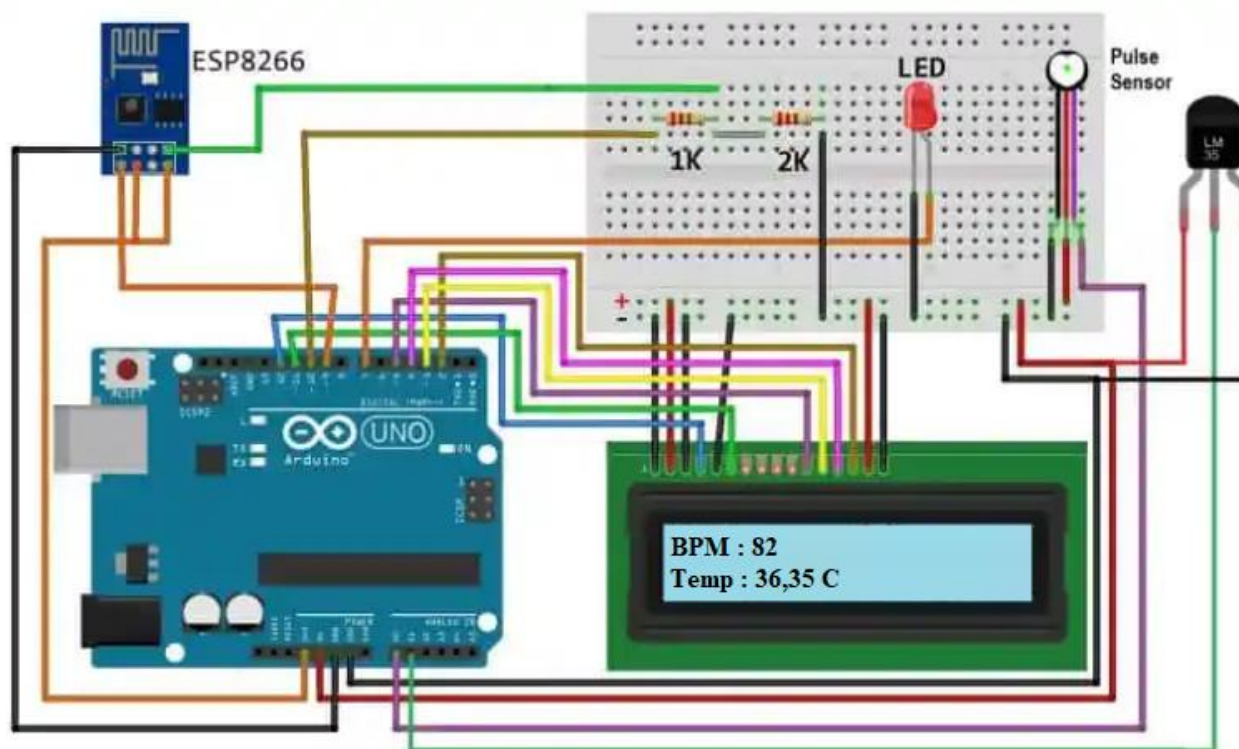
					Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino.	Літ.			Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата						
Розроб.		Скрипник Ю.В.								
Перевір.		Тащук О.Ю.								
Т. Контр.						Арк.			Аркуші 1	
Реценз.						Циклова комісія Комп'ютерної інженерії				
Н. Контр.										
Затверд.		Тащук О.Ю.								

Структурна схема підключення датчиків до плати Arduino Uno



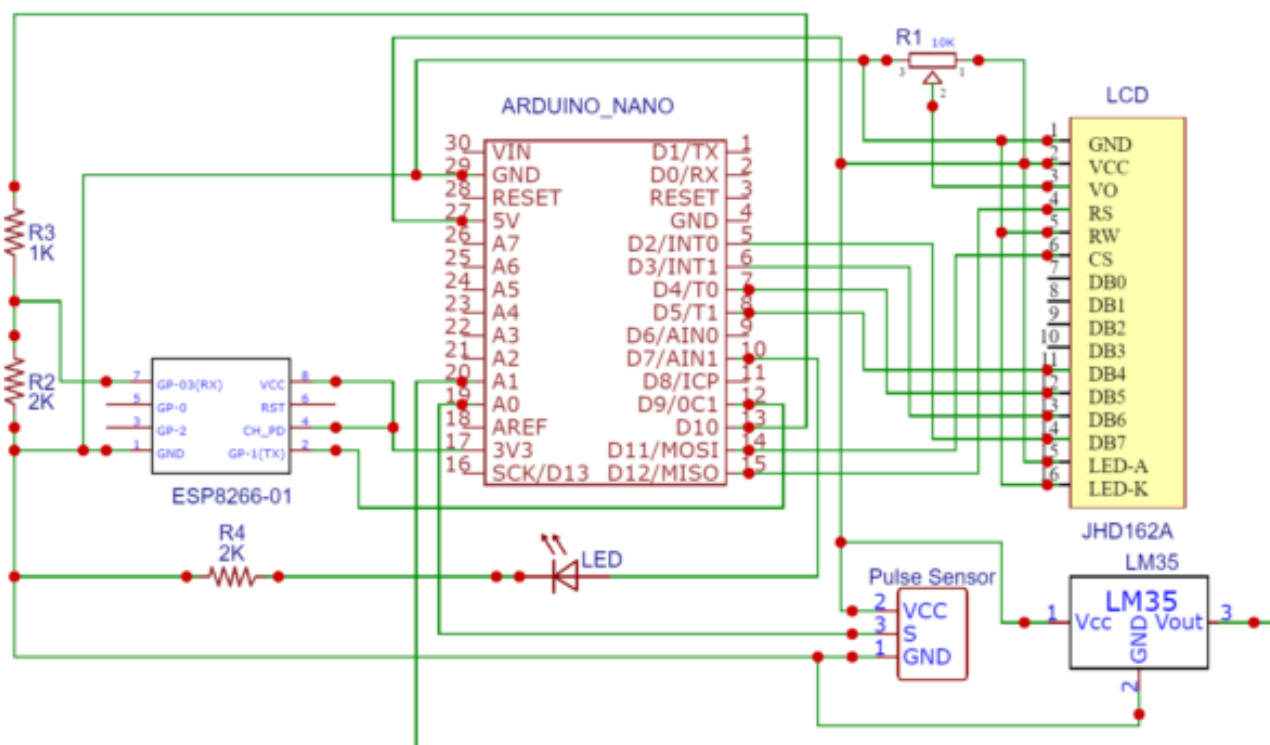
					Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino.	Лім.			Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата						
Розроб.		Скрипник Ю.В.								
Перевір.		Ташук О.Ю.								
Т. Контр.						Арк.			Аркуші 1	
Реценз.						Циклова комісія Комп'ютерної інженерії				
Н. Контр.										
Затверд.		Ташук О.Ю.								

Змн.	Арк.	№ докум.	Підпис	Дата
<i>Розроб.</i>		<i>Скрипник Ю.В.</i>		
<i>Перевір.</i>		<i>Ташук О.Ю.</i>		
<i>Т. Контр.</i>				
<i>Реценз.</i>				
<i>Н. Контр.</i>				
<i>Затверд.</i>		<i>Ташук О.Ю.</i>		



Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino.	Лім.			Маса		Масштаб			
	Арк.			Аркушів		1			
					Циклова комісія Комп'ютерної інженерії				

Змн.	Арк.	№ докум.	Підпис	Дата
<i>Розроб.</i>		<i>Скрипник Ю.В.</i>		
<i>Перевір.</i>		<i>Ташук О.Ю.</i>		
<i>Т. Контр.</i>				
<i>Реценз.</i>				
<i>Н. Контр.</i>				
<i>Затверд.</i>		<i>Ташук О.Ю.</i>		



Система моніторингу здоров'я пацієнтів на основі IoT з використанням ESP8266 і Arduino.	Лім.			Маса		Масштаб			
	Арк.			Аркушів		1			
					Циклова комісія Комп'ютерної інженерії				

