

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра автоматизації та комп'ютерно-інтегрованих технологій

ДОПУЩЕНО ДО ЗАХИСТУ
В. о. завідувача кафедри АКІТ,
кандидат технічних наук, доцент

_____ М. І. Сіделєв
“ ____ ” _____ 2023 р.

МАГІСТЕРСЬКА НАУКОВА РОБОТА

**на тему: «Автоматизована система параметричного контролю стану
пацієнта за групою показників»**

Пояснювальна записка

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»
151 – МНР – 671.21930201

Студент _____ Соровецький А.М

Керівник _____ Трунов О. М.

Консультант _____ Григор'єва Л.І.

Миколаїв – 2023

Чорноморський національний університет ім. Петра Могили
(повне найменування вищого навчального закладу)

Інститут, факультет, відділення
Кафедра, циклова комісія

Комп'ютерних наук
Кафедра автоматизації та
комп'ютерно-інтегрованих
технологій

Освітньо-кваліфікаційний рівень

магістр

Галузь знань 15 «Автоматизація, та приладобудування»

(шифр і назва)

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри, голова

циклової

комісії _____

“ ____ ”

_____ 20__ року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТА

Соровецький Антон Мірославович

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи)

«Автоматизована система параметричного контролю стану пацієнта за групою показників»

Керівник проекту (роботи) доктор технічних наук, професор

Трунов Олександр Мірославович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ ____ ” _____ 20__ року

№ _____

2. Строк подання студентом проекту (роботи) 1 травня 2023 року

3. Вихідні дані до проекту (роботи): Система віддаленого розташування пацієнта, що потребує відновлення та знаходиться під наглядом сімейного лікаря та спеціальних лікарів. Умови передбачають контроль та протоколювання параметрів за стандартними методиками МОЗ для відновлювального періоду хвороби або ушкодження. Завдання обмежено відновленням постінсультних пацієнтів, що мають залишкові спазми м'язів верхніх. Проведіть аналіз сучасних приладів і систем, що придатні забезпечити контроль, протоколювання та бездротову, безвтратну передачу для зберігання даних. Розвинена інфраструктура, устаткування, програмні комплекси, автоматизовані системи керування, що забезпечують ефективність до та пост процедурного контролю втому числі візуального та пацієнт, медсестра, сімейний лікар, спеціальні лікарі, страхова компанія, адміністрація лікарні. Мережа, сервер, доступ до персональних даних та даних спостереження за розділенням, при обмеженні та записом для різних учасників процесу.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) провести пошук існуючих систем, виявити найбільш прогресивні рішення. Розглянути теоретичні основи процесу автоматизації і конструювання мережевих приладів моніторингу параметрів стану пацієнта і засобів протоколювання. Перелік параметрів: ЕКГ, тиск, температура, пульс. Запропонувати структуру автоматизованої системи з усіма взаємодіючими компонентами у томи числі як таку, що оснащено засобами дворівневого контролю доступу, безвтратного стиснення, шифрування, передачі відкритими каналами зв'язку та організації доступу до бази даних способами, що дозволяють доступ і віддалений контроль та корекцію ходу процедур. Розробити функціональну блок-схему системи керування комплексом. Розробити метод безвтратного стиснення, шифрування та передачі даних. Запропонувати схемні рішення схему автоматичного запису ЕКГ та розробити алгоритм їх роботи.

Базуючись на алгоритмі роботи розробити користувацький інтерфейс системи управління параметричним контролем стану пацієнта за групою показників. Побудувати математичну модель та дослідити режими сучасними програмними засобами моделювання.

Розглянути та запропонувати засоби з охорони праці та безпеки у надзвичайних ситуаціях на робочих місцях в офісному приміщенні ТОВ «Солар Сервіс». Запропонувати рішення, що забезпечують вимоги освітлення, опалення та вентиляції при роботі в офісному приміщенні. Визначити проблеми охорони праці.

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

1. Функціональна блок-схема системи керування комплексом.
2. Математична модель системи.
3. Алгоритм роботи системи.
4. Електрична принципова схема системи керування комплексом.
5. Користувацький інтерфейс системи управління комплексом.

6. Консультанти розділів проекту (роботи)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|------------------------------------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| 1 | Трунов О.М. д-р техн. наук, проф. | | |
| 2 | Трунов О.М. д-р техн. наук, проф. | | |
| 3 | Трунов О.М. д-р техн. наук, проф. | | |
| Спеціальна частина з охорони праці | Григорєва Л.І. д-р біологічних наук, професор | | |

7. _____ Дата _____ видачі _____
завдання _____

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проекту (роботи) | Строк виконання етапів проекту (роботи) | Примітка |
|-------|--|---|----------|
| 1 | Затвердження пропозицій теми від керівника | 20.09.2022 | |
| 2 | Обговорення зі студентом затвердженої теми | 03.10.2022 | |
| 3 | Формування завдання | 17.10.2022 | |
| 4 | Визначення актуальності, об'єкту, предмету | 01.11.2022 | |
| 5 | Пошук літератури, патентний пошук, уточнення задач дослідження | 15.11.2022 | |
| 6 | Виконання першої частини | 01.12.2022 | |
| 7 | Аналіз керівником записки першої частини, формування зауважень та пропозицій | 29.12.2022 | |
| 8 | Опрацювання другої частини | 01.03.2023 | |
| 9 | Робота над третьою частиною | 03.04.2023 | |
| 10 | Робота над розділом з охорони праці | 19.04.2023 | |
| 11 | Передзахисти | 19.05.2023 | |
| 12 | Передача автореферату кваліфікаційної роботи в електронному вигляді | 12.06.2023 | |
| 13 | Передача кваліфікаційної роботи | 14.06.2023 | |

Здобувач освіти

(підпис)

А. Соровецький

(ініціали, прізвище)

Керівник магістерської
роботи

(підпис)

О. М. Трунов

(ініціали, прізвище)

д-р техн. наук, професор

Відгук керівника

на кваліфікаційну роботу магістра студента 671 групи ЧНУ імені Петра Могили Соровецького Антона Мірославовича за темою «Автоматизована система параметричного контролю стану пацієнта за групою показників» на здобуття ступеню магістра, за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології»

Кваліфікаційна робота присвячена комплексному аналізу, розробці та схемо-технічній і програмній реалізації АСК параметричного контролю стану пацієнта за групою показників. Проблема підвищення якості процесів віддаленого домашнього відновлення шляхом впровадження вебзастосунків, що реалізує бездротові мережеві технології керованої роботи медичних приладів на базі Андроїдного смартфона є актуальною, особливо в умовах пандемії або війни. У зв'язку з цим розгляд і комплексне застосування інноваційних рішень з розвитку комп'ютеризованих систем та їх елементів є багатообіцяючим підходом.

В ході роботи Соровецький Антон Мірославович виявив здатність до самостійної роботи, аналізу літературних джерел, та технічних рішень, продемонстрував вміння відстежувати тенденції розвитку комп'ютеризованих систем, працювати з науково-технічною літературою, технічною документацією, описами програмних продуктів та працювати у сучасних середовищах. На основі проведеного аналізу, обрано як базову блок схему, для якої шляхом порівняння характеристик компонентів обирається та визначається уточнена структура для дослідження і удосконалення програмного забезпечення.

Робота засвідчує здатність студента Соровецького Антона Мірославовича до проведення всіх етапів проектування і дослідження та експериментального доведення проекту комп'ютеризованої системи. Експериментальне випробування вебзастосунку, засвідчує повноту застосування знань, вмінь та навичок як показник якісного рівня освіти.

Таким чином, автором роботи Соровецьким Антоном Мірославовичем продемонстровані досягнуті програмні результати навчання: аналізувати фізичні особливості процесів, розробляти, обґрунтовувати структуру АСК, формувати моделі, обирати параметри на підставі моделювання, розробляти і пропонувати алгоритми, розробляти інтерфейси і характер представлення графічного виводу, практично обирати компоненти з урахуванням досягнень і стану ринку, формувати алгоритм писати і відлагоджувати програму роботи вебзастосунку у цілому.

Систематична робота, самостійність і якість виконання у ході розв'язку поставлених у роботі задач, позитивно характеризує Соровецького Антона Мірославовича як виконавця і дослідника.

Керівник, професор кафедри АКІТ,
д. т. н., професор АКІТ,
ЧНУ ім. Петра Могили



О. М. Трунов

АНОТАЦІЯ

до кваліфікаційної магістерської роботи

«Автоматизована система параметричного контролю стану пацієнта за групою показників»

Студент гр. 671 Соровецький Антон Мирославович

Керівник: д-р техн. наук, професор Трунов Олександр Миколайович

Актуальність теми кваліфікаційної роботи полягає в тому, що створення системи віддаленого контролю стану пацієнта, що перебуває під наглядом сімейного та спеціальних лікарів, дозволяє не лише зменшити навантаження на існуючі заклади лікування/реабілітації, а й зменшити пагубний вплив людського фактору на пацієнта й запобігти втраті даних, виникненню непередбачуваних ситуацій, що несуть за собою негативний вплив на систему.

Об'єкт дослідження – процеси автоматизації бездротового збору, отримання, передачі, обробки та представлення даних.

Предмет дослідження – методи, моделі та інструментальні і програмні засоби контролю стану пацієнта, що реалізовано на базі бездротових технологій та інформаційного вебзастосування.

У даній роботі поставлено за **мету** підвищення ефективності автоматизованої системи моніторингу і контролю стану пацієнта шляхом удосконалення архітектури, алгоритмів, програм та засобів їх первинної обробки для бездротової передачі і виведення даних за допомогою інформаційного вебзастосування.

Для досягнення мети необхідно виконати такі **завдання**:

- проаналізувати предметну область;
- ознайомитись з існуючими системами, їх перевагами, недоліками та особливостями;
- визначити компонентний склад системи;
- визначити техніку створення ІС;
- розробити схемне рішення системи;
- розробити алгоритми безвратного стиснення та шифрування даних;
- розробити користувацький інтерфейс;
- створити повноцінну автоматизовану систему, що дозволяє реалізувати віддалений контроль за станом пацієнта.

Кваліфікаційна робота містить перелік скорочень, вступ, чотири розділи, висновок, перелік джерел посилання та два додатки.

Вступ містить основні обґрунтування актуальності розробки обраної теми, об'єкт, предмет дослідження, мету та завдання, які необхідно виконати для досягнення поставленої мети.

У розділі 1 проводиться аналіз стану та тенденцій розвитку автоматизованих систем контролю стану пацієнта. Починаючи з огляду автоматизованих систем, висвітлюються їх особливості та можливості моніторингу стану пацієнта. Далі розглядаються програми та алгоритми, що використовуються в цих системах. Особлива увага приділяється тенденціям у дослідженні стану пацієнта.

У розділі 2 розглядається обґрунтування вибору методів, моделей, апаратного та програмного забезпечення для автоматизованих систем контролю стану пацієнта. Визначаються особливості структури таких систем та їх створення на базі мікроконтролерів. Розглядаються підходи до проектування та архітектури системи. Проводиться вибір системи керування базами даних та програмного забезпечення для розробки веб-застосунків. Також розглядаються особливості використання бездротових технологій для передачі даних.

У третьому розділі представлена реалізація автоматизованої системи контролю стану пацієнта та процес її тестування. Пояснюється схема реалізації апаратних засобів, розробляється математична модель системи та описується алгоритм її роботи. Проводиться програмування інструкцій мікроконтролерів та датчиків, а також впровадження серверної частини та бездротової передачі даних на смартфон. Окремо розглядається реалізація веб-застосунку. Представлені результати тестування автоматизованої системи.

У висновку описано результати виконання кваліфікаційної роботи.

Додатки містять код програмного забезпечення та інформацію про апробацію кваліфікаційної магістерської роботи.

В спеціальній частині з охорони праці та безпеки в надзвичайних ситуаціях розглянуто забезпечення вимог охорони праці на робочому місці та описано алгоритм дій у випадку настання надзвичайної ситуації.

Кваліфікаційна магістерська робота містить 98 сторінок (без додатків), 34 рисунки, 34 джерела посилання, 2 додатки.

ABSTRACT

of the qualifying Master's Thesis

"Automated system of parametric monitoring of the patient's condition by group of indicators"

Student of 671 Anton Myroslavovych Sorovetskyi

Supervisor: Dr. Tech. Sciences, Professor Oleksandr Mykolayovych Trunov

The **relevance of the topic** of the qualification work is that the creation of a system of remote monitoring of the patient's condition under the supervision of family and special doctors allows not only to reduce the burden on existing treatment/rehabilitation facilities, but also to reduce the harmful impact of the human factor on the patient and prevent data loss, the occurrence of unforeseen situations that have a negative impact on the system.

The **object of the study** is the processes of automation of wireless collection, reception, transmission, processing and presentation of data.

The **subject of the study** is methods, models, and instrumental and software monitoring of the patient's condition, implemented on the basis of wireless technologies and an informational web application.

In this work, **the goal** is to increase the efficiency of the automated system of monitoring and controlling the patient's condition by improving the architecture, algorithms, programs and means of their primary processing for wireless transmission and output of data using an informational web application.

To achieve the goal, it is necessary to complete **the following tasks**:

- analyze the subject area;
- familiarize yourself with the existing systems, their advantages, disadvantages and features;
- determine the component composition of the system;
- to determine the technique of creating IS;
- develop a schematic system solution;
- develop algorithms for lossless data compression and encryption;
- develop a user interface;
- to create a full-fledged automated system that allows remote monitoring of the patient's condition.

The thesis contains a list of abbreviations, an introduction, four chapters, a conclusion, a list of reference sources and two appendices.

The introduction contains the main justifications for the relevance of the development of the chosen topic, the object, the subject of research, the goal and the tasks that must be performed to achieve the goal.

Chapter 1 analyzes the state and development trends of automated patient monitoring systems. Starting with an overview of automated systems, their features and possibilities for monitoring the patient's condition are highlighted. Next, the programs

and algorithms used in these systems are considered. Particular attention is paid to trends in the study of the patient's condition.

Section 2 reviews the rationale for selecting methods, models, hardware, and software for automated patient monitoring systems. The peculiarities of the structure of such systems and their creation on the basis of microcontrollers are determined. Approaches to system design and architecture are considered. Database management system and web application development software are selected. Peculiarities of using wireless technologies for data transmission are also considered.

The third chapter presents the implementation of the automated patient monitoring system and its testing process. The hardware implementation scheme is explained, the mathematical model of the system is developed, and the algorithm of its operation is described. Microcontroller and sensor instructions are being programmed, as well as the implementation of the server part and wireless data transfer to a smartphone. The implementation of the web application is considered separately. The results of testing the automated system are presented.

The conclusion describes the results of the qualification work.

Appendices contain the software code and information about the approval of the qualifying master's thesis.

In the special part on occupational health and safety in emergency situations, the provision of occupational health and safety requirements at the workplace is considered and the algorithm of actions in the event of an emergency situation is described.

The qualifying master's thesis contains 98 pages (without appendices), 34 figures, 34 reference sources, 2 appendices.

ЗМІСТ

| | |
|---|------|
| ПЕРЕЛІК СКОРОЧЕНЬ | 13 |
| ВСТУП..... | 14 |
| РОЗДІЛ 1. АНАЛІЗ СТАНУ ТА ТЕНДЕНЦІЙ РОЗВИТКУ АВТОМАТИЗОВАНИХ СИСТЕМ КОНТРОЛЮ СТАНУ ПАЦІЄНТА | 16 |
| 1.1 Автоматизовані системи, особливості, моніторинг | 16 |
| 1.2 Програми, алгоритми..... | 20 |
| 1.3 Тенденції дослідження стану пацієнта..... | 23 |
| 1.4 Структура автоматизованих систем | 29 |
| 1.5 Постановка завдання..... | 33 |
| 1.6 Висновки до розділу | 34 |
| РОЗДІЛ 2. ОБГРУНТУВАННЯ ВИБОРУ МЕТОДІВ, МОДЕЛЕЙ, АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ | 36 |
| 2.1 Особливості структури автоматизованої системи | 36 |
| 2.2 Особливості створення автоматизованих систем на базі МК | 39 |
| 2.3 Підхід до проєктування, архітектура..... | 45 |
| 2.4 Вибір СКБД | 53 |
| 2.5 Вибір програмних засобів для розробки вебзастосунку..... | 56 |
| 2.6 Особливості застосування бездротових технологій для передачі даних | 63 |
| 2.7 Висновки до розділу | 66 |
| РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ, ТЕСТУВАННЯ | 67 |
| 3.1 Схема реалізації апаратних засобів | 67 |
| 3.2 Математична модель..... | 75 |
| 3.3 Алгоритм роботи..... | 79 |
| 3.4 Програмування інструкцій МК та датчиків | 82 |
| 3.5 Впровадження серверної частини та бездротової передачі даних на смартфон | 83 |
| 3.6 Реалізація вебзастосунку | 87 |
| 3.7 Тестування автоматизованої системи | 90 |
| 3.8 Висновки до розділу | 94 |
| РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ | 96 |
| ВИСНОВКИ | 110 |
| ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ..... | 112 |
| ДОДАТОК А Коди програм..... | 115 |
| A1 Код програми тонометра | 115 |
| A2 Код програми термометра | 117 |
| A3 Лістинг частини коду вебзастосунку..... | 117 |
| ДОДАТОК Б Публікація за темою роботи | 121 |
| ДОДАТОК В Публікація за темою роботи | 1213 |

ПЕРЕЛІК СКОРОЧЕНЬ

| | |
|------|---|
| ДСТУ | – державний стандарт України |
| МОНУ | – міністерство освіти і науки України |
| МОЗ | – міністерство охорони здоров'я України |
| ІС | – інформаційна система |
| МНР | – магістерська наукова робота |
| ПЗ | – програмне забезпечення |
| МК | – мікроконтролер |
| БД | – база даних |
| СКБД | – система керування базами даних |

ВСТУП

Сучасний стан та тенденції розвитку медицини свідчать про застосування комп'ютерно-інтегрованих технологій для інноваційних розробок. Однією з них є автоматизовані системи контролю стану пацієнта, що дозволяють збирати, обробляти та аналізувати різноманітну інформацію про стан здоров'я людини.

Ці системи дозволяють підтримувати постійний контроль за станом пацієнта, визначити наявність ризиків розвитку захворювань та вчасно реагувати на зміни в його стані. Завдяки автоматизованим системам контролю стану пацієнта можна забезпечити більш ефективно та персонально орієнтоване лікування, що дозволяє збільшувати шанси на повне відновлення здоров'я.

У зв'язку з цим зростає інтеграція медичних і інтегрованих комп'ютерних технологій, а автоматизовані системи контролю стану пацієнта набувають все більшої популярності та стають необхідним інструментом для багатьох медичних закладів.

У роботі розглянуто автоматизовану систему контролю стану пацієнта, її функціональні можливості, переваги та недоліки широкого використання. Також проаналізовано сучасні тенденції та перспективи розвитку цих систем у медичній сфері.

Для подальшої практичної реалізації результатів в роботі приділено увагу застосуванню бездротових технологій та дослідженню впровадження сучасних засобів реалізації мікроконтролерної архітектури та мультиплатформних вебзастосунків.

Мета роботи є підвищення ефективності автоматизованої системи моніторингу і контролю стану пацієнта шляхом удосконалення архітектури, алгоритмів, програм та засобів їх первинної обробки для бездротової передачі і виведення даних за допомогою інформаційного вебзастосунку.

Об'єкт дослідження – процеси автоматизації бездротового збору, отримання, передачі, обробки та представлення даних.

Предмет дослідження – методи, моделі та інструментальні і програмні засоби контролю стану пацієнта, що реалізовано на базі бездротових технологій та інформаційного вебзастосунку.

Новизна дослідження полягає в створенні вебзастосунку для спроектованого конструктивного рішення на базі МК та бездротових датчиків. Пошук існуючих систем та аналіз їх застосування дозволить сформулювати вимоги до апаратного та програмного забезпечення.

Результат роботи – автоматизована система контролю стану пацієнта, що дозволяє віддалено й оперативно отримувати дані про об'єкт спостереження та демонструвати їх уповноваженій особі в режимі реального часу на будь-який пристрій без прив'язки до конкретної операційної системи (Windows, Linux, MacOS, Android, IOS, тощо) та місцезнаходження.

РОЗДІЛ 1. АНАЛІЗ СТАНУ ТА ТЕНДЕНЦІЙ РОЗВИТКУ АВТОМАТИЗОВАНИХ СИСТЕМ КОНТРОЛЮ СТАНУ ПАЦІЄНТА

1.1 Автоматизовані системи, особливості, моніторинг

Сьогодні абсолютно спокійно і впевнено можна сказати, що комп'ютер «ввійшов в щоденну практику» життя та роботу людей, в тому числі в практиці працюючих лікарів. Для опису застосування технологій комп'ютерних пристроїв в сфері медицини існує термін m-Health.

M-Health розуміється як послуги, сервіси, ініціативи, програми, події та інші дії в галузі медицини та охорони здоров'я, реалізація яких використовує мобільні пристрої (телефони, смартфони, планшети) та різні технології бездротового з'єднання. У широкому розумінні M-Health – використання мобільні технології для медичних послуг та інформації в галузі охорони здоров'я [1].

Термін Mobile Health (m-Health) ввів в сучасний ужиток англійський професор в 2003 році, академік Роберт Істепанян. Професор, який широко визнаний провідним фахівцем і першо прохідцем в області мобільної охорони здоров'я, є науковим співробітником кількох дослідних центрів вузів і компаній (наприклад, Orange, Philips, Motorola) і провідним експертом в області M2M (Machine-to-Machine, міжмашинної взаємодії). Відомий тим, що розробив модернізовану концепцію нового рівня еволюції mHealth і ввів ще одне сучасне поняття - 4G Health [2].

В даний час у світі мобільна медицина ефективно використовується при лікуванні і профілактиці інфекційних та неінфекційних захворювань, піклуванні про людей похилого віку, інформування вагітних жінок, боротьбі з дитячою та материнською смертністю.

Відповідно до звіту компанії JSon & Partners Consulting, швидко зростаючим сегментом ринку M-Health є програми для смартфонів у сфері здоров'я. Основні категорії мобільних додатків можна поділити на 4 основні групи:

- 1) загальне здоров'я (фітнес, харчування, дієти тощо);
- 2) медична інформація (довідкові книги, інформування, діагностика, освіта);
- 3) віддалені консультації та моніторинг;
- 4) управління охороною здоров'я (електронні картки пацієнтів, логічна та платіжна підтримка).

В рамках M-Health реалізуються різні типи послуг за допомогою різних технологій складності, включаючи складні програми/платформи для мобільних телефонів, з інтегрованими можливостями/або відповідними пристроями для контролю життєво важливих показів - рівень цукру, тиск тощо.

Аналітики компанії Json & Partners Conning висвітлюють наступні типи послуг, які реалізуються як частина M-Health з різними по складності технологіями:

- проекти, що використовують SMS-розсилки для інформування, ініціаторами яких можуть бути держава, публічні організації, фармацевтичні компанії тощо;
- мобільна телемедицина;
- Завантажені програми з мережі Інтернет, які використовуються користувачем для різних цілей (фітнес, інформування тощо);
- Складні програми/платформи для мобільних телефонів, з інтегрованими можливостями/або відповідними пристроями для віддаленого контролю життєво важливих показів - рівень цукру, тиск тощо.

З щоденним всесвітнім процесом поширенням смартфонів M-Health все більший ухил робить саме на них, про це може свідчити стаття з Національної Бібліотеки Медицини, в якій досліджувалось саме питання поширення таких застосунків [3].

На рисунку 1.1 зображено гістограму, що відображає щорічну кількість програм для мобільних пристроїв завантажених до мобільних маркетів зі сфери M-Health.

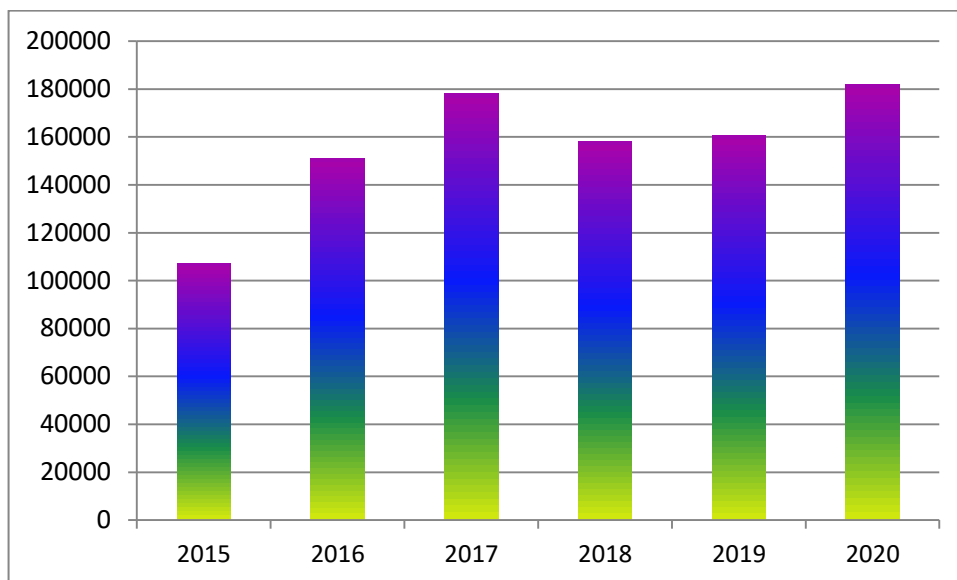


Рисунок 1.1 Щорічна кількість завантажених мобільних програм M-Health

Найбільша частина проектів M-Health у світі - 46% - реалізовано в галузі системи охорони здоров'я, на другому місці – сфера профілактики захворювань, у цій категорії 27% проектів (на рис. 1.2 приведена діаграма впроваджених проектів у сфері M-Health згідно з GSMA, MHealth Tracker) [4].

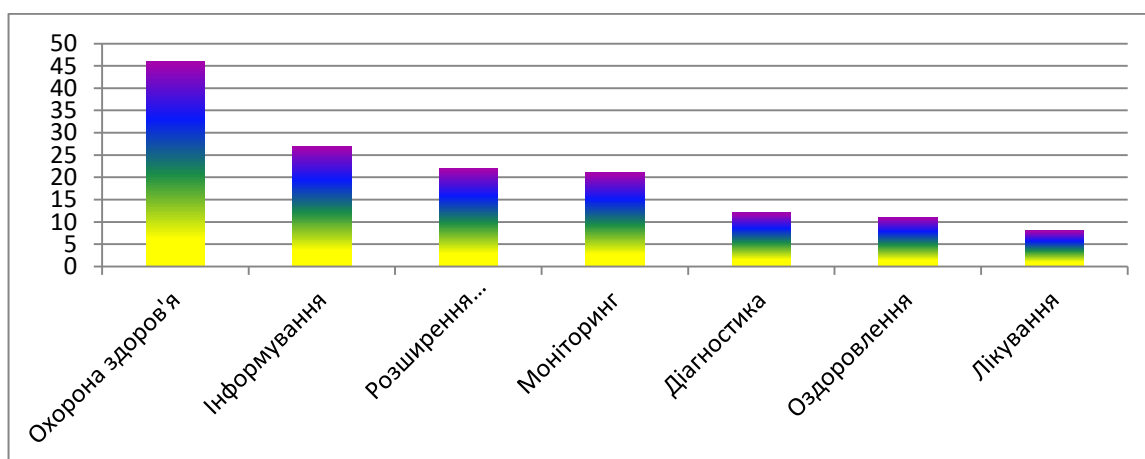


Рисунок 1.2 Реалізовані проекти в області M-Health по категоріям

Слід зазначити, що один Проект може бути реалізований одночасно в декількох категоріях. Аналіз найпопулярніших додатків MHealth для смартфонів у 2014 році в Сполучених Штатах компанія "Research2Guidance" виявила, що майже половину сегменту зайняли лише 2 типи програм: для фізичного контролю(фітнес) та довідкові матеріали[5]. На основі досліджених даних сформуємо діаграму популярності мобільних застосунків з області mHealth станом на 2014 рік в США рисунок 1.3:

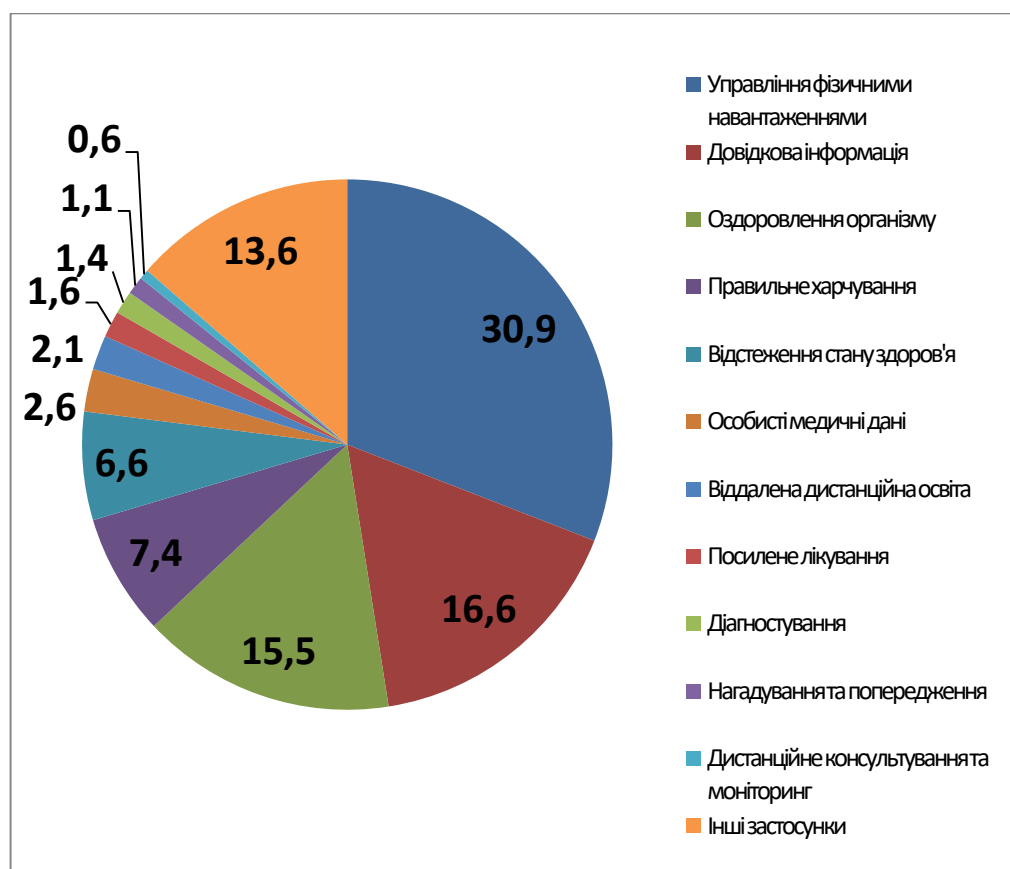


Рисунок 1.3 Найбільш популярні програми m-Health для смартфонів у США за 2014 рік

Аналізуючи дану діаграму приходимо до висновку про початок розвитку та поширення систем дистанційного консультування та моніторингу стану пацієнта. Звернувшись до звітної документації тієї ж компанії "Research2Guidance" за 2020 рік [6]. Приведемо дані у вигляді діаграми, рисунок 1.4:

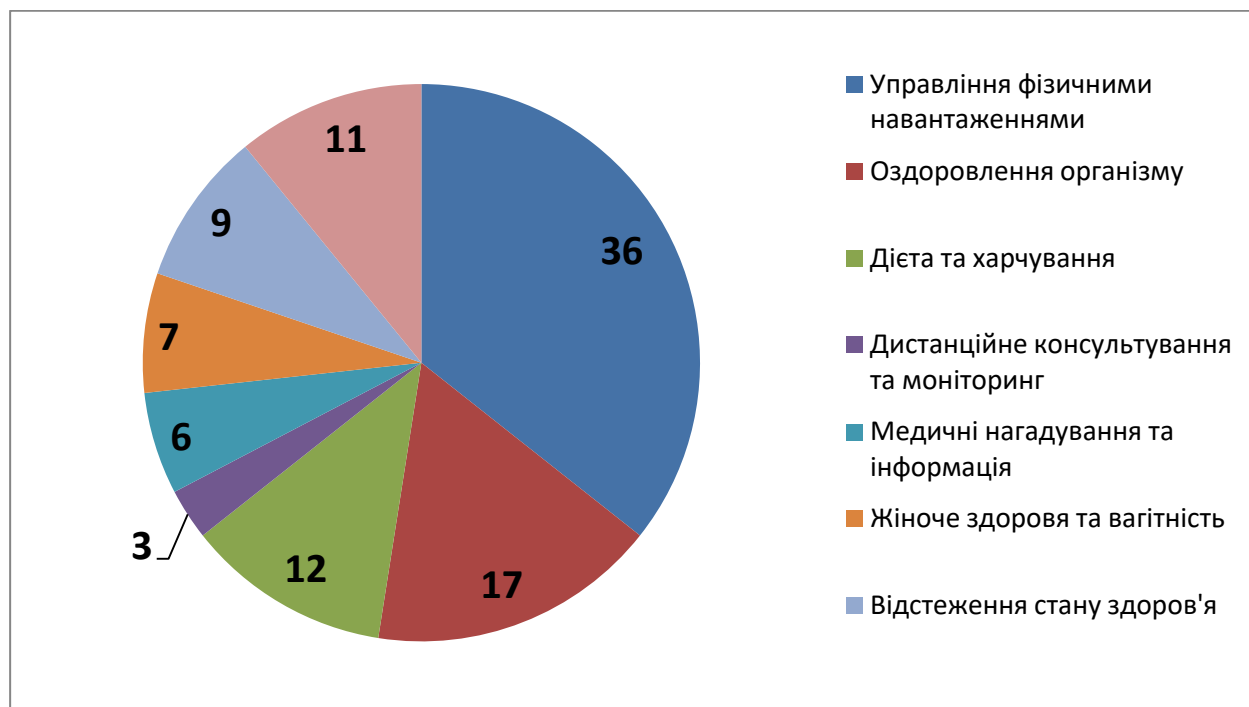


Рисунок 1.4 Найбільш популярні програми m-Health для смартфонів у США за 2020 рік

Слід помітити стрімке зростання попиту користувачів до систем відстеження здоров'я та дистанційного консультування та моніторингу стану. Зважаючи на ці дані та матеріали онлайн-видання «Аптека», яке з 1995 року займається дослідженням стану фармації та медицини в Україні, «mHealth – це не тільки сучасні технології, це абсолютно новий шлях взаємодії пацієнта з лікарем та догляд за хворим» [7].

1.2 Програми, алгоритми

Програми і алгоритми в області контролю стану пацієнта зазвичай використовуються у медичних закладах, щоб допомогти лікарям і медичному персоналу контролювати стан пацієнтів і забезпечити їм найбільш ефективне

лікування. Основні типи програм, що використовуються в цій області зображено на рисунку 1.5:



Рисунок 1.5 Основні типи програм в області контролю стану пацієнта

1. Програми моніторингу базових функцій: такі програми дозволяють медичному персоналу контролювати стан пацієнта, спостерігаючи за його базовими функціями, такими як пульс, тиск, температура тіла, рівень кисню в крові тощо. Зазвичай ці програми використовуються в інтенсивній терапії та реанімації.

2. Алгоритми контролю за лікуванням: ці алгоритми використовуються для контролю за лікуванням пацієнта та забезпечення правильної дози ліків. Вони можуть бути засновані на показниках крові, відгуку на лікування тощо.

3. Програми моніторингу хвороби: ці програми дозволяють пацієнтам контролювати свій стан за допомогою сенсорів та мобільних пристроїв. Вони можуть допомогти пацієнтам визначити, коли необхідно звернутися до лікаря, а також контролювати терапію.

4. Алгоритми контролю за станом післяопераційного періоду: ці алгоритми використовуються для контролювання стану пацієнта після операції. Вони можуть допомогти виявити можливі ускладнення та вчасно прийняти необхідні заходи.

Ці програми та алгоритми допомагають лікарям та медичному персоналу контролювати стан пацієнта та приймати рішення про необхідність додаткових заходів лікування або переведення в іншу палату, іншу одиницю лікування або спеціалізовану клініку.

Більшість з цих програм та алгоритмів базуються на використанні даних, які отримуються з медичних приладів, таких як монітори вітальних функцій, електрокардіографи, глюкометри, термометри тощо. Ці дані можуть передаватися в режимі реального часу до системи, що збирає та аналізує інформацію, і на основі цього розробляються алгоритми, які допомагають медичному персоналу приймати рішення про лікування пацієнта.

Крім того, у цій області активно використовуються технології штучного інтелекту, такі як машинне навчання та нейронні мережі. Ці технології дозволяють програмам та алгоритмам самостійно навчатися на основі отриманих даних, що може покращити точність прогнозування стану пацієнта та допомогти лікарям приймати більш обґрунтовані рішення.

Усі ці програми та алгоритми допомагають покращити якість надання медичної допомоги та знижують ризик ускладнень під час лікування. Однак, вони не замінюють кваліфікованого медичного персоналу та їх використання повинно бути додатковим інструментом у руках лікаря.

При створенні програмного забезпечення для контролю стану пацієнта важливими аспектами є:

- контроль аутентифікації користувачів;
- бездротове з'єднання та безвтратна та надійна передача даних;
- документування.

Дотримання даних аспектів напряду залежить на кінцевому продукті, рівень якого важливий для впровадження в роботу.

1.3 Тенденції дослідження стану пацієнта

Основна мета моніторингу стану пацієнта - відстежувати різні фізіологічні показники, такі як артеріальний тиск, пульс, температура тіла, рівень кисню в крові та інші.

Моніторинг може проводитися як у лікарні, так і вдома. Для цього використовуються спеціальні медичні пристрої, які дозволяють реєструвати ці показники в режимі реального часу. Інформація з пристроїв збирається і аналізується, щоб дати лікарям змогу зрозуміти, як пацієнт реагує на лікування та чи потрібні додаткові корективи.

Однією з найбільш поширених форм моніторингу є ЕКГ-моніторинг, який дозволяє стежити за ритмом серця та виявляти можливі порушення в його роботі. Іншими видами моніторингу є спеціалізовані пристрої для вимірювання кров'яного тиску, діагностичні монітори, які забезпечують інформацію про рівень кисню в крові та інші показники, а також моніторинг температури тіла.

Моніторинг в медицині допомагає лікарям вчасно виявляти можливі проблеми зі здоров'ям пацієнта та приймати відповідні заходи для їх вирішення. Цей метод є дуже важливим у лікуванні пацієнтів з серцево-судинними захворюваннями, діабетом та іншими захворюваннями, які можуть вимагати постійного контролю з боку медичного персоналу. Моніторинг також є важливим інструментом для забезпечення безпеки пацієнтів у лікарнях та інших медичних установах.

Окрім того, моніторинг може бути корисним у випадках, коли пацієнт потребує догляду вдома. У таких випадках медичні пристрої можуть бути підключені до спеціальних систем, які передають дані про стан пацієнта медичному персоналу в режимі реального часу. Це може допомогти вчасно виявляти можливі проблеми та забезпечити швидку реакцію.

Стрімкий розвиток потреби в даному напрямку привів до великої кількості пропозицій, теорій та розробок в даному напрямку, розглянемо деякі з них.

Таким чином, міжнародні аудиторська компанія Deloitte у своєму звіті зазначила, про щорічні темпи світових витрат на сферу охорони здоров'я в розмірі 5.4% [8].

При цьому, серед ключових тенденцій зазначають:

- Забезпечення сталого фінансування в умовах економічної невизначеності галузей охорони здоров'я.
- Використання нових моделей надання медичної допомоги для покращення якості, швидкості та своєчасності таких послуг.
- Інвестування в цифрові технології.
- Забезпечення нормативної відповідальності та кібербезпеки.

Важливу роль та надії покладають саме на впровадження та поширення mHealth.

З кожним роком збільшується кількість розробок в даній сфері, що дозволяє прийти до висновку про бурхливість розвитку даної сфери, в тому числі й в Україні. Розглянемо деякі з досліджень:

1. Методи та засоби формування загальних показників для автоматизації апаратів реабілітаційної медицини для постінсультних хворих [9].

У статті розглянуто сучасний процес відновлення післяінсультних та постінсультних пацієнтів в умовах становлення інституту сімейної медицини та страхових лікарів. Запропоновано впровадження автоматизованого модуля реабілітаційного обладнання (МАРД) для забезпечення довготривалості відновлювальних процедур, якості життя та зменшення трудовитрат. Обґрунтовано формат представлення Узагальнених індикаторів представлення пацієнтів (GIPS) - моделі інтегрованих показників, що дозволяє оцінити якість медичних послуг та підвищити ефективність використання даних; запропоновано послідовне застосування двох евклідових норм, що об'єднує показники різної

фізичної природи в обмеженому метричному просторі; розроблено модель відповідно до вимог МОЗ. На основі розкладів в ряд Тейлора, геометричних нерівностей та обмежених просторів встановлено взаємозв'язок між нижньою та верхньою межами, похибками, шириною ковзного вікна та похідними значеннями GIPS. Створено модель для оцінювання GIPS як нижньої межі та метод генерування інформації про її властивості.

Застосовано трирівневий компаратор і введено індикаторний вектор (VI) як інформаційне доповнення до часового ряду. Подальші можливості інтелектуального аналізу: представлено модель GIPS з VI; на прикладах значень VI продемонстровано застосовність VI до інтелектуального аналізу процесу відновлення; відкритість, доступність і прозорість GIPS і VI як інструментів КІТ продемонстровано на прикладах князівств державного управління (ДУ), кількісні та якісні показники в переліку яких здійснюються шляхом порівняння з наявністю та зводяться до кількісних контролів; діагностичні висновки та процедурні модифікації VI, "ковзних вікон" та програмного забезпечення як інструменту ПА та КІТ досліджуються чисельно.

2. Композитна оцінка та аналіз варіабельності серцевого ритму [10].

Патент передбачає систему та метод збору та аналізу даних про варіабельність серцевого ритму (ВСР) представлені для надання більш конкретних вказівок, які базуються на даних ВСР користувача, таких як налаштовані оцінки, налаштовані модифікації конкретних планів або програм, наприклад, програми керованого дихання, плани спортивних тренувань, валідація лікування або події, які, як очікується, вплинуть на показники ВСР. Зібрані дані про ВСР можна використовувати в додатках типу біологічного зворотного зв'язку в реальному часі, наприклад, включати показники ВСР або комбіновані показники з використанням даних ВСР як графіку відображення в потокових службах, використовувати показники ВСР або складені показники для зміни поведінки

систем, таких як ігрові системи, транспортні засоби, системи рекомендацій вмісту, системи вибору реклами тощо.

3. Медичний монітор [11].

У статті розглянуто базові терміни, прилади, алгоритми та готові системи які впроваджені в сфері контролю стану пацієнтів. Також описується роль розробки нових методів моніторингу в різних галузях медицини, таких як інтелектуальна медицина, біомедична допомога, нетрадиційна медицина, самостійна профілактична медицина та прогностична медицина. Моніторинг базується на зборі вичерпних медичних даних про пацієнтів, людей, що перебувають у групі ризику, і здорових людей, з використанням сучасних, розумних, мінімально інвазивних біомедичних пристроїв, біосенсорів, лабораторій-на-чіпі, а в майбутньому можуть використовуватись нанороботи. Дослідники працюють над тим, щоб задовольнити необхідність всебічного подальшого вивчення та особистого постійного клінічного моніторингу стану здоров'я, зберігаючи застаріле медичне втручання як крайній засіб. Ліки можуть пропонувати тимчасове полегшення симптомів, тоді як корінь медичної проблеми залишається невідомим без достатньої кількості даних усіх наших біологічних систем. Моніторинг у медицині заповнює прогалину для запобігання помилок діагностики та може допомогти у подальших медичних дослідженнях, аналізуючи всі дані багатьох пацієнтів.

4. 15 медичних проектів на Arduino [12].

В статті описуються можливості апаратно-програмних засобів Arduino, які дозволяють проводити самостійний моніторинг стану здоров'я людини. Ця електронна платформа з відкритим вихідним кодом надає гнучке та економічно ефективно рішення для створення медичних пристроїв і систем, які можна налаштувати відповідно до конкретних потреб.

У даній статті розглянуто 15 медичних проектів на основі Arduino, які демонструють універсальність і потенціал Arduino в галузі охорони здоров'я. Ці проекти варіюються від простих датчиків до складних систем і демонструють інноваційні способи використання Arduino для покращення результатів здоров'я та якості життя людей із різними захворюваннями.

5. Національний урядовий сервіс медицини (NGSmedicare)[13]

Сайт компанії, що займається медичним страхуванням та адмініструванням, яка зосереджується на забезпеченні ефективного та якісного обслуговування пацієнтів та провайдерів. При цьому компанія NGS Medicare пропонує широкий спектр приладів для моніторингу здоров'я, таких як електрокардіографи, пульсоксиметри, термометри, апарати для вимірювання артеріального тиску та інші.

Одним з їх ключових продуктів є мультіфункціональний прилад NGS Medical Check-up Pro, який може вимірювати пульс, кисневий насичення крові, артеріальний тиск та температуру тіла. Даний прилад має можливість зберігати дані в пам'яті та передавати їх на мобільний додаток за допомогою Bluetooth.

Крім того, NGS Medicare пропонує також лінійку біосенсорів, які дозволяють вимірювати рівень цукру в крові, холестерин, тригліцериди та інші показники. Дані прилади можуть бути корисними для людей з діабетом, серцево-судинними захворюваннями та іншими хронічними захворюваннями, які потребують постійного моніторингу свого стану здоров'я.

NGS Medicare також розробляє медичні програми для моніторингу здоров'я, які можуть бути інтегровані з приладами, дозволяючи лікарям та пацієнтам отримувати доступ до даних та аналізувати їх для виявлення тенденцій та змін у стані здоров'я.

Так, як це урядова компанія, доступ до продуктів можливий лише її клієнтам, тому поширення даних систем за межами США на даний момент неможливе.

6. Їстівна електроніка [14].

Стаття описує винахід вчених з Університету Сассекса. За допомогою водорозчинних магнітних і електричних компонентів та мікроенергетичних джерел на основі солодководних водоростей та інших біорозкладних матеріалів, група дослідників розробила "їстівну електроніку" - пристрої, які можуть бути проковтнуті людиною та розчинені в тілі.

Ці пристрої можуть бути використані для моніторингу різних параметрів здоров'я, таких як рівень глюкози, температура тіла і пульс, а також для доставки ліків.

Головним компонентом цих пристроїв є мікрочіп, який містить у собі тонкі шари електроніки на основі магнітів і електродів. Ці шари розміщені на "свіжій" або "живій" шкірі з солодководних водоростей, які можуть розчинятися в тілі.

За допомогою магнітів, дослідники можуть керувати рухом пристроїв в тілі, що дозволяє їм переміщатися до потрібного місця, наприклад, до кишечника, де вони можуть розчинятися.

Ці пристрої можуть бути використані для доставки ліків у місця, які не можуть бути досягнуті звичайним шляхом, а також для моніторингу стану здоров'я, що дозволяє вчасно виявляти певні хвороби або реагувати на зміни стану здоров'я.

За допомогою цієї технології дослідники надіються зробити медичну терапію більш ефективною та безпечною, а також знизити витрати на лікування.

Представлені розробки, дослідження та винаходи лише частково показують процес розвитку тенденцій в сфері дослідження стану здоров'я пацієнта, але при цьому досить чітко показують курс подальшого розвитку, який зосереджується на автоматизації самого процесу.

1.4 Структура автоматизованих систем

Автоматизована система - це складний механізм, що складається з різноманітних компонентів, що взаємодіють між собою, тому важливим аспектом будь-якої автоматизованої системи є її структура.

Основні компоненти структури автоматизованої системи включають:

1. Апаратне забезпечення - це комп'ютери, сервери, принтери, мережеві пристрої та інші пристрої, які використовуються для зберігання, обробки та передачі інформації. Важливо, щоб апаратне забезпечення відповідало потребам системи та мала достатній рівень надійності та продуктивності.
2. Програмне забезпечення - це програми, що використовуються для обробки даних та керування системою. Це може бути операційна система, база даних, програмне забезпечення для автоматизації бізнес-процесів, програми для взаємодії з користувачами та ін.
3. Мережеве забезпечення - це технології, що використовуються для передачі даних між різними компонентами системи. Це може бути провідна мережа, бездротова мережа, Інтернет-підключення та інші засоби зв'язку.
4. База даних - це складова системи, що використовується для зберігання та організації даних, що використовуються системою. Важливо, щоб база даних була структурованою та дозволяла ефективний доступ до інформації.
5. Користувачі - люди, що взаємодіють з системою та використовують її функціонал для виконання різних завдань. Важливо, щоб система була зручною для користування та мала достатній рівень наочності для користувача.

Структура автоматизованої системи може подаватись різним чином, але основним і найбільш часто використовуваним є структурна схема автоматизованої системи.

Структурна схема автоматизованої системи – це графічне зображення, що описує компоненти та взаємозв'язки між ними в системі. Вона використовується для опису логічної та фізичної структури системи. Структурні схеми можуть бути створені для різних рівнів абстракції, від загальної концептуальної схеми до детальної фізичної схеми. Також вони можуть бути використані для опису різних типів систем, від програмного забезпечення до апаратних засобів [15].

Основними компонентами, які можуть бути включені в структурну схему автоматизованої системи, є:

- Модулі або компоненти: це окремі частини системи, які виконують певні функції та можуть бути використані в інших частинах системи.
- Інтерфейси: це засоби взаємодії між компонентами системи. Вони можуть включати в себе протоколи, формати даних та інші способи взаємодії.
- Процеси: це послідовності дій, які виконуються для виконання певної функції в системі.
- Дані: це інформація, яка передається між компонентами та оброблюється системою.

Структурна схема може бути представлена в різних форматах, таких як блок-схеми, діаграми компонентів, схеми декомпозиції та інші. Крім того, структурні схеми можуть бути використані як засіб візуалізації та комунікації між розробниками та іншими учасниками проекту.

Типову структурну схему автоматизованої системи можна представити у вигляді, представленому на рисунку 1.6:

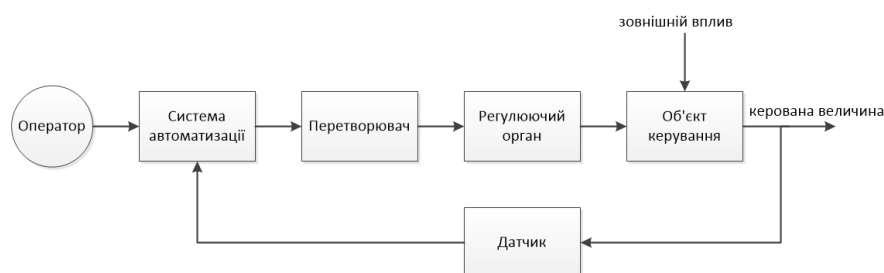


Рис. 1.6. Типова структурна схема автоматизованої системи

Наочність та детальність схеми визначається індивідуально на етапі постановки завдання. Чим детальніше реалізована схема, тим простіше виконавцям буде в процесі реалізації. Важливою частиною для наочності схеми є відображення каналів живлення, зв'язку, обміну інформації, протоколів. Розглянемо схему з більш деталізованим відображенням (рисунку 1.7).

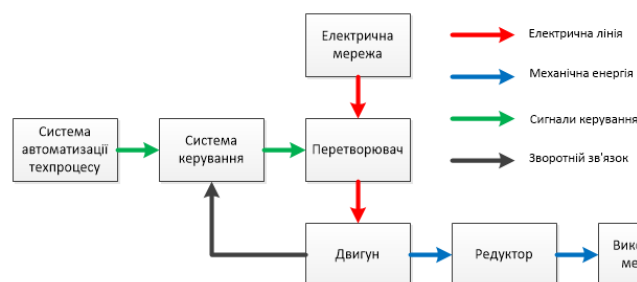


Рис. 1.7. Структурна схема автоматизованої системи

Аналізуючи типовий процес контролю стану пацієнта, що знаходиться в межах закладу охорони здоров'я, можна дійти висновку про даний процес. До пацієнта підключені медичні прилади (датчики), що вимірюють показники його стану, датчики під'єднані до системи автоматизації, яка зчитує покази з датчиків та трансформує їх у зрозумілий лікарю вид і виводить на монітор. В свою чергу лікар переглядаючи покази на моніторі робить висновок стосовно дій, що необхідно виконати і може передати команду на виконавчий пристрій маючи в цьому потребу. Зобразимо структурну схему на рисунку 1.8:

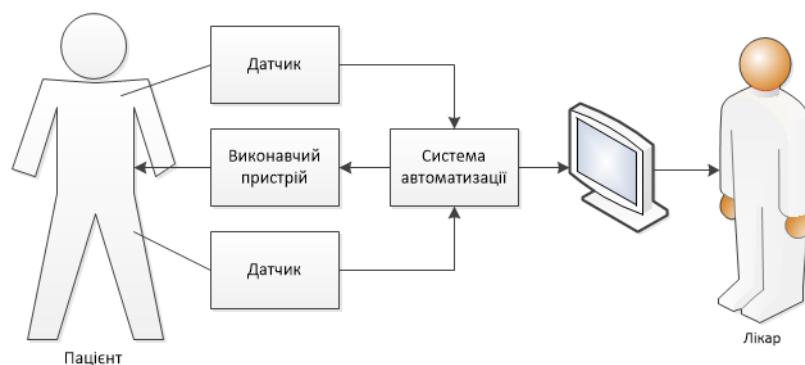


Рис. 1.8. Структурна схема системи контролю стану пацієнта

Слід зауважити, що дана схема є спрощеною і не враховує великої кількості аспектів, але надає уявлення на типову структуру в закладах охорони здоров'я. В процесі модернізації та автоматизації, слід додати деталізацію та розподілити можливість доступу до даних стану пацієнта лікарем за межами палати. Для вирішення даного питання альтернативним варіантом є використання розподіленої мережі та сервера, що в свою чергу надає такі переваги:

- доступ до даних з кабінета чергового лікаря;
- швидкість відклику лікаря на зміни стану;
- можливість одночасного спостереження даних декількох пацієнтів;
- зменшення необхідності планованого обходу.

Автоматизовану систему контролю стану пацієнта з використанням локальної мережі та серверу можна представити у вигляді наступної структурної схеми, рисунок 1.9:

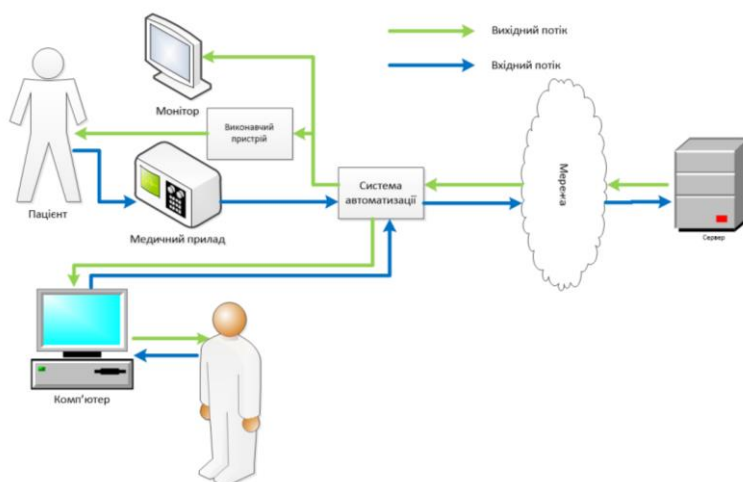


Рис. 1.9. Структурна схема автоматизованої системи контролю стану пацієнта

Дана реалізація процесу контролю стану пацієнта хоч і має переваги, але досить широко застосування на даному етапі розвитку сфери охорони здоров'я не набула. При цьому, це не кінцевий варіант. Наступним етапом модернізації можна вважати перехід з локальної мережі на глобальну, застосування мультиплатформних вебзастосунків та бездротових медичних пристроїв, що в свою чергу надає можливості до спостереження не критичних пацієнтів на відстані, наприклад на домашньому лікуванні чи відновленні.

1.5 Постановка завдання

Актуальність даної теми роботи полягає в тому, що створення системи віддаленого контролю стану пацієнта, що перебуває під наглядом сімейного та спеціальних лікарів, дозволяє не лише зменшити навантаження на існуючі заклади лікування/реабілітації, а й зменшити пагубний вплив людського фактору на пацієнта й запобігти втраті даних, виникненню непередбачуваних ситуацій, що несуть за собою негативний вплив на систему. Дані, які система буде контролювати відповідають стандартним методикам МОЗ обмежено відновленням постінсультних пацієнтів, що мають залишкові спазми м'язів.

Протоколювання, передача та збереження даних повинно відбуватись за допомогою бездротових технологій. Важливою частиною також є обробка, шифрування і архівація даних та контроль ініціалізації/доступу до них.

Мета дипломної роботи полягає у організації обліку, обробки та візуалізації даних автоматизованої системи параметричного контролю стану пацієнта за групою показників шляхом розробки й в тому числі інформаційної системи.

Завдання: розробити автоматизовану систему параметричного контролю стану пацієнта за групою показників, зі застосуванням інноваційних підходів, методів та технологій створення застосунків, реалізацією засобів дворівневого контролю доступу, безвратного стиснення, шифрування, передачі відкритими каналами зв'язку та організації доступу до бази даних способами, що дозволяють доступ і віддалений контроль та корекцію ходу процедур.

Для досягнення поставленої задачі необхідно виконати наступні завдання:

- 1) проаналізувати предметну область;
- 2) ознайомитись з існуючими системами, їх перевагами, недоліками та особливостями;
- 3) визначити компонентний склад системи;
- 4) визначити техніку створення ІС;
- 5) розробити схемне рішення системи;
- 6) розробити алгоритми безвратного стиснення та шифрування даних;
- 7) розробити користувацький інтерфейс;
- 8) створити повноцінну автоматизовану систему, що дозволяє реалізувати віддалений контроль за станом пацієнта.

1.6 Висновки до розділу

1. Автоматизовані системи контролю стану пацієнта є важливими інструментами в сучасній медицині. Вони дозволяють збирати та аналізувати

дані про стан пацієнта, що відкриває нові можливості для діагностування та лікування різноманітних захворювань.

2. Детальний аналіз стану та тенденцій розвитку автоматизованих систем контролю стану пацієнта засвідчує, що на сьогоднішній день системи контролю стану пацієнта є поширеними лише в країнах з розвинутою системою охорони здоров'я. Останнє дозволяє оцінити якість використання таких систем та поширити їх передовий досвід. Перспективними є системи, які зібрані з декількох модулів, що розширює параметрів інформації, такі як вітальні функції пацієнта, склад крові та інші показники здоров'я.

3. Тенденції розвитку автоматизованих систем контролю стану пацієнта полягають у подальшому вдосконаленні алгоритмів збору та обробки даних, а також застосуванні бездротових технологій їх передачі, що зменшує помилки та збільшує точність діагностики. Крім того, збір даних у реальному часі забезпечує оперативне реагування на зміни у стані пацієнта та своєчасне лікування.

4. Загальний інноваційний потенціал автоматизованих систем контролю стану пацієнта полягає у поліпшенні діагностики та лікування різноманітних захворювань при високому рівні захисту даних пацієнтів та безпеки використання цих систем. Галузі використання систем контролю стану пацієнта поширюватимуться на лікарні, шпиталі та віддалене відновлення пацієнтів.

Отже обрана тема магістерської кваліфікаційної роботи є актуальною та набуває практичного значення, особливо в теперішній час.

РОЗДІЛ 2. ОБГРУНТУВАННЯ ВИБОРУ МЕТОДІВ, МОДЕЛЕЙ, АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Особливості структури автоматизованої системи

Розглянувши основні структури автоматизованих систем, що використовуються в системах контролю стану пацієнта, слід розглянути переваги і недоліки кожної з них, табл. 2.1:

Таблиця 2.1

Порівняння структур автоматизованих систем

| Назва Критерій | Стандартна структура | Мережева структура | Бездротова структура |
|----------------------------------|--|--|--|
| Сучасність | застаріла | актуальна | передова |
| Надійність | надійна | надійна | надійна |
| Зона дії | палата | будівля | Не обмежена |
| Можливість релокації пацієнта | Відсутня | присутня в межах будівлі | присутня |
| Доступ до інформації лікарем | При безпосередньому знаходженні біля пацієнта | При перегляді з комп'ютера всередині лікарняного закладу | З мобільного пристрою без обмежень в місцезнаходженні |
| Масштабування | ускладнене | просте при наявності розвинутої локальної мережі | просте |
| Монтаж | простий | ускладнений | простий |
| Налаштування | складне | ускладнене | просте |

Відповідно до аналізу таблиці, можна обрати основні два типи структур: мережеву(дротову) та бездротову мережеву архітектури.

Дротова та бездротова передача даних мають свої переваги та недоліки. Деякі з них перераховані нижче.

Переваги дротової передачі даних:

- Більш висока швидкість передачі даних, порівняно з бездротовою передачею.
- Менше вразливості до інтерференції, що може впливати на якість сигналу та швидкість передачі.
- Більша надійність зв'язку, оскільки дротові з'єднання зазвичай мають меншу схильність до переривань зв'язку або інших помилок передачі.

Недоліки дротової передачі даних:

- Обмежена мобільність, оскільки передача даних може бути можливою лише тоді, коли пристрої знаходяться поблизу дротового з'єднання.
- Обмежена гнучкість у встановленні та зміні мережі, оскільки дротові з'єднання вимагають фізичного підключення до мережі.
- Неможливість передачі даних на великі відстані без додаткових засобів передачі, таких як оптичні кабелі або підсилувачі.

Переваги бездротової передачі даних:

- Більша мобільність, оскільки передача даних може здійснюватися незалежно від фізичного підключення до мережі.
- Більша гнучкість у встановленні та зміні мережі, оскільки бездротові з'єднання можуть бути встановлені та змінені без необхідності у фізичному підключенні.
- Можливість передачі даних на великі відстані без додаткових засобів передачі.

Недоліки бездротової передачі даних:

- Нижча швидкість передачі даних, порівняно з дротовою передачею.
- Більша вразливість до інтерференції, яка може вплинути на якість сигналу та швидкість передачі.
- Нижча надійність зв'язку, оскільки бездротові з'єднання можуть бути більш схильні до переривань зв'язку або інших помилок передачі, особливо на великих відстанях або в умовах зі зниженою якістю сигналу.
- Більша витрата енергії на передачу даних, що може впливати на час роботи батареї пристрою, особливо на мобільних пристроях.

Отже, обираючи між дротовою та бездротовою передачею даних, потрібно враховувати їх переваги та недоліки та відповідність до конкретних потреб користувача та умов використання. Наприклад, дротова передача може бути кращою вибором для стаціонарних пристроїв, таких як комп'ютери або телевізори, які не потребують частоті мобільності, тоді як бездротова передача може бути більш підходящою для мобільних пристроїв, таких як смартфони або планшети, які потребують більшої мобільності та гнучкості у встановленні з'єднання.

Проаналізувавши дані порівняння різних типів структур, перевага однозначно надається бездротовій структурі.

Відповідно до обраного типу створимо структурну схему автоматизованої системи контролю стану пацієнта, рисунок 2.1:

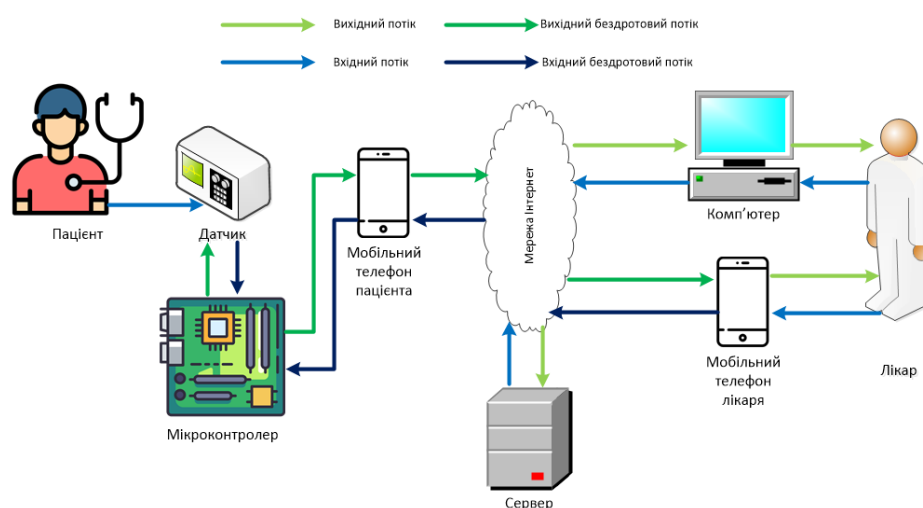


Рис. 2.1. Структурна схема автоматизованої системи контролю стану пацієнта

Відповідно до рисунку 2.1 опишемо систему. До пацієнта, що знаходиться під наглядом лікаря підключено відповідні датчики для контролю стану, датчики підключені до мікроконтролера, який в свою чергу «прив'язаний» до мобільного телефону пацієнта за допомогою бездротової технології. Телефон за допомогою мережі інтернет передає покази на сервер, з якого інформація стає доступною лікарю на вебзастосунку, який він може відкрити на мобільному чи стаціонарному комп'ютерному пристрої. Також існує зворотній зв'язок, який надає можливість лікарю в режимі реального часу прослідкувати за показами датчиків розташованих на пацієнта, шляхом передачі відповідної команди через вебзастосунок.

2.2 Особливості створення автоматизованих систем на базі МК

Мікроконтролер є інтегральною схемою, яка містить в собі необхідні електронні компоненти, такі як центральний процесор (CPU), пам'ять, вхідно-вихідні порти (GPIO), таймери та інші функціональні блоки. Він є основою для багатьох автоматизованих систем, таких як системи керування освітленням, системи контролю клімату, системи безпеки та багато інших.

Зараз на ринку представлена величезне різноманіття мікроконтролерів від різних виробників для різних задач, на рисунках 2.2-2.4 представлені одні з найпопулярніших варіантів.

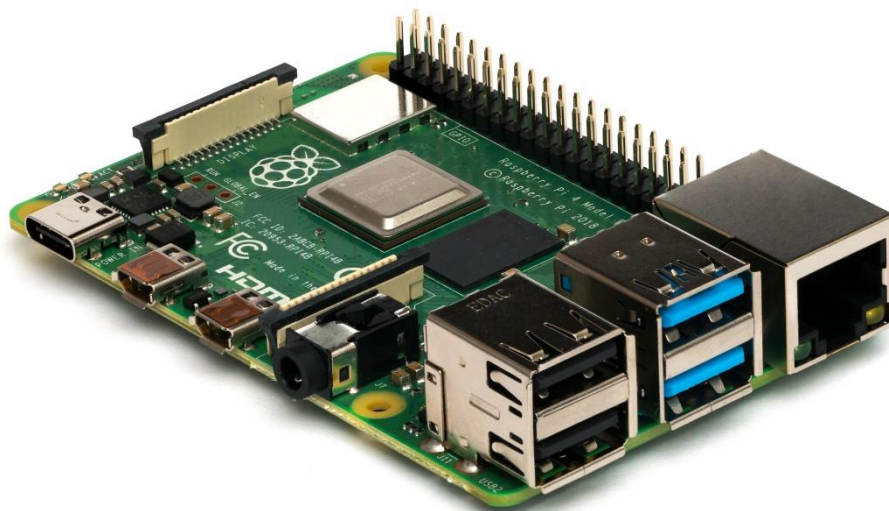


Рис. 2.2. Мікроконтролер Raspberry Pi

Розробка організації Raspberry Pi Foundation першопочатково для розвитку навичок програмування в закладах шкільної освіти, яка може задовольнити потреби не лише школярів, а й досвідчених інженерів та розробників. Основна дошка Raspberry Pi містить процесор, оперативну пам'ять, порти USB та Ethernet, а також виходи HDMI та аудіо. До цієї дошки можна підключати різноманітні модулі та датчики, що дозволяє створювати різноманітні автоматизовані системи та пристрої. Крім того, Raspberry Pi працює на базі операційної системи Linux та підтримує багато програмувальних мов, таких як Python, C++, Java та інші [16].

Оскільки Raspberry Pi має відкритий код та широкий спектр документації, він став популярним у галузі DIY-розробок, особливо у галузі IoT (Internet of Things). Raspberry Pi можна використовувати для різних проектів, таких як створення домашнього медіацентру, системи моніторингу, автоматизованих систем управління та багатьох інших.



Рис. 2.3. Мікроконтролер Nucleo STM32

Розробка компанії STMicroelectronics. Ці мікроконтролери використовуються в багатьох галузях, таких як автомобільна та промислова автоматика, медична техніка, пристрої IoT (Internet of Things) та багато інших.

Основні переваги STM32 полягають у високій продуктивності та енергоефективності, а також у широкому спектрі доступних опцій та розширювальних можливостях. STM32 має високу швидкість обробки даних, багато периферійних пристроїв, таких як таймери, пристрої для зберігання даних та зв'язку, а також може працювати на різних частотах тактування [17].

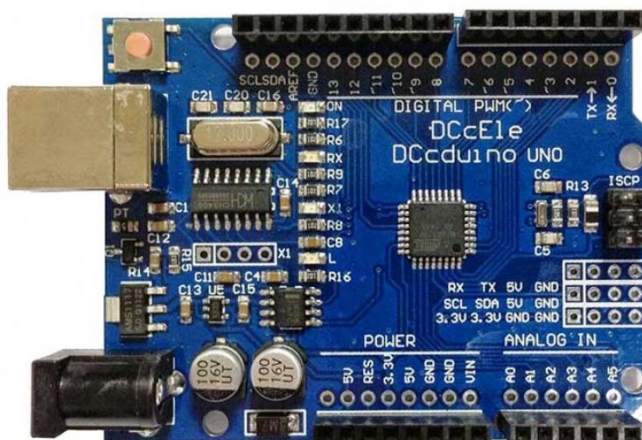


Рис. 2.4. Мікроконтролер Arduino UNO

Розробка компанії Arduino. Базується на мікроконтролері, який програмується за допомогою мови програмування C++ та відповідного середовища розробки.

Arduino є досить популярним вибором для початківців та для тих, хто має мало досвіду в розробці електронних пристроїв. Arduino відкрите середовище, що дозволяє долучатися до великої спільноти користувачів, що надає доступ до безлічі корисної документації, бібліотек та проектів.

Основна перевага Arduino полягає в його легкій використовуваності та доступності. Arduino має простий інтерфейс та невеликі вимоги до обладнання, що дозволяє початківцям легко створювати різноманітні електронні пристрої та системи.

Arduino має багато периферійних пристроїв, таких як таймери, засоби зв'язку та зберігання даних, що дозволяє розробникам зосередитися на програмуванні функціональності пристрою, а не на деталях взаємодії з обладнанням [18].

Незалежно від обраного мікроконтролера та середовища розробки, є базові особливості створення автоматизованих систем на базі МК, які полягають у наступному:

- Мікроконтролери програмуються за допомогою спеціальних мов програмування, таких як C або C++. Розробники можуть використовувати додаткові бібліотеки та фреймворки для полегшення процесу програмування та взаємодії з іншими компонентами системи.
- Для створення автоматизованої системи на базі мікроконтролера необхідно розробити схему підключення компонентів до GPIO портів мікроконтролера та забезпечити правильну взаємодію з іншими елементами системи, такими як датчики, клавіатури, LCD-екрани та інші.
- Мікроконтролери мають малі розміри та низьке енергоспоживання, що дозволяє їх використовувати в малих пристроях, наприклад, вбудованих системах в електроприладах. Важливо враховувати ці параметри під час вибору мікроконтролера для конкретної системи.
- Мікроконтролери мають високу надійність і стійкість до електромагнітних перешкод та інших факторів впливу зовнішнього середовища. Інтерфейси: Мікроконтролери підтримують різні інтерфейси для забезпечення взаємодії з іншими компонентами системи. До них відносяться інтерфейси такі як I2C, SPI, UART та інші. Вибір конкретного інтерфейсу залежить від вимог системи та наявності підтримки у мікроконтролера.
- Використання мікроконтролера може знизити вартість розробки автоматизованої системи, оскільки ціни на мікроконтролери зазвичай нижчі, ніж на відповідні комп'ютерні системи.

- Мікроконтролери прості у використанні та програмуванні, тому вони є популярними для створення прототипів та експериментальних систем.

Недоліки створення автоматизованих систем на базі мікроконтролера полягають у наступному:

- Мікроконтролери мають обмежені можливості порівняно з комп'ютерними системами, що може обмежити функціональність створеної системи.
- Мікроконтролери мають обмежені ресурси, такі як обсяг пам'яті та швидкість процесора, що може вплинути на продуктивність системи.
- Розробка системи на базі мікроконтролера може бути складнішою порівняно з розробкою на комп'ютерних системах, оскільки необхідно враховувати обмежені можливості та ресурси мікроконтролера.
- Взаємодія з іншими компонентами системи може бути складною та вимагати додаткових зусиль з боку розробника.

Отже, створення автоматизованих систем на базі мікроконтролера має свої переваги та недоліки. Використання мікроконтролера дозволяє створити компактну та енергоефективну систему з високою точністю керування та забезпечити взаємодію з іншими компонентами системи. Однак, обмежені можливості та ресурси мікроконтролера можуть обмежити функціональність та продуктивність системи. Також, розробка на базі мікроконтролера може бути складнішою порівняно з розробкою на комп'ютерних системах, оскільки необхідно враховувати обмежені можливості та ресурси мікроконтролера.

При створенні автоматизованих систем на базі мікроконтролера необхідно добре вивчити можливості мікроконтролера та врахувати його обмеження. Крім того, необхідно враховувати вимоги до системи та підбирати відповідні компоненти, що дозволять досягти бажаної функціональності та продуктивності системи.

Узагалі, мікроконтролери є важливим елементом у створенні автоматизованих систем, оскільки дозволяють забезпечити точність та ефективність керування, а також забезпечити взаємодію з іншими компонентами системи.

В процесі аналізу приведеної вище інформації, прийнято рішення до використання в якості мікроконтролера плати Arduino Uno.

2.3 Підхід до проєктування, архітектура

На початку своєї історії всі вебзастосунки будувалися без будь-яких архітектурних принципів, програма складалася з безлічі наступних один за одним рядків, без класових залежностей, наслідувань і інших сучасних можливостей мов програмування. Однак, прагнення до спрощення розробки, а також до структуризації як коду, так і всього програми могло бути передумовою до появи такого поняття як «Архітектура ПЗ» [19].

В сучасному світі існує безліч типів підходів до архітектурного проєктування, але основними є:

1. Монолітний підхід.
2. Модульний підхід.
3. SOA1 або сервіс-орієнтований підхід.

Монолітний підхід є найстарішою моделлю проєктування ПЗ оскільки саме з нього і почалася розробка всього програмного забезпечення. У рамках даного підходу складної структури веб-додатку може не бути: сервер зберігає всю бізнес-логіку, а база даних - дані необхідні серверу для роботи. Як правило подібні програми не відрізняються складністю в розробці і її великою вартістю на ранніх етапах, коли список необхідного функціоналу не відрізняється великою кількістю рядків, а помилкові дії досить просто виправляються на початкових стадіях. Нова функціональність додається легко і швидко. Однак з плином часу або в поспіху за датою випуску цілком вірогідне зростання ризику припуститися помилки, яка

перетворюється в «технічний борг», який необхідно буде виправити після випуску продукту. Однак з плином часу при монолітному підході, таких «Помилки» може накопичитися дуже багато і таким чином, підтримувати монолітну систему в довгостроковій перспективі дуже складно і дорого, тому що команди розробників можуть змінюватися і виходить так, що простіше зробити «латочку», ніж розібратися як працює та чи інша частина програми.

Проблеми з підтримкою монолітного додатку можна пояснити і тим, що рано чи пізно невеликої список функціоналу в системі поповнюється новими вимогами та ідеями, які вимагають внесення змін в уже існуючий код, що вже відбивається в останньому пропозиції попереднього абзацу. Крім того, в рамках веб-додатків однією з проблем «моноліту» є масштабованість. Всі складові системи розташовані в одній точці, а тому потужність самої точки повинна бути відповідною для підтримки сервера і бази даних в робочому стані.

Приклад монолітної архітектури можна знайти в будь-якому додатку, навіть якщо він включає в себе різні класи. Варто відзначити, що в цей момент може здатися, що за стосунок в цей момент ставати модульним, проте все зовсім не так. Монолітне застосунок по праву може включати модулі, однак, вони повністю або частково залежать один від одного. На рисунку 2.5 наведено приклад структури простого веб-додатку, що складається з сервера, бази даних, а також брандмауера, зазвичай за захист відповідає хостинг, але якщо створювати щось своє, то і захищатися доведеться самому.

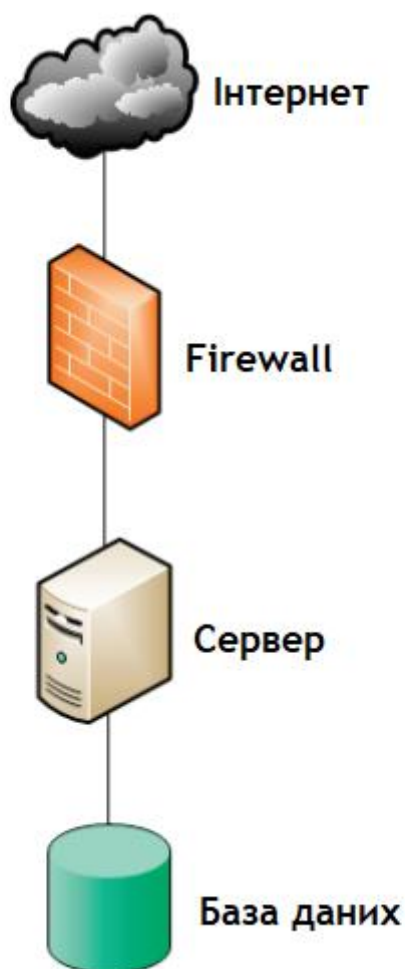


Рис. 2.5 Монолітна архітектура

Модульна архітектура на відміну від монолітної має на увазі розбиття всього функціоналу програми на окремі модулі, кожен з яких відповідає за певну частину функціоналу програми таким чином, що модульну архітектуру можна описати як сукупність безлічі монолітних модулів всередині однієї програми.

Кожен модуль програми є функціонально незалежним від іншого, а тому його застосування в різних ділянках коду не викликати збоїв в працездатності, по суті це надає простоту рефакторинга і перенесення меж модуля. Варто відзначити, що при зміні одного модуля, інші модулі порушені не будуть, а тому спрощується задача дебагінгу. Все, що потрібно, це налаштувати інші модулі, що

використовують функціонал зміненого модуля під нові можливості і вимоги станнього, якщо зміни в ньому можуть призвести до збою в роботі програми.

Крім того, всі модулі взаємодіють один з одним в синхронному порядку, оскільки знаходяться на одній апаратній частині, а при необхідності для модуля можна виділити окрему базу даних або ж частина даних, з якими цей модуль буде працювати. Прикладом можуть виступати сучасні РНР фреймворки застосовуються для швидкої розробки веб-додатків. В даних фреймворками кожен створений клас-модуль зазвичай містить певні налагодження додатка в разі помилок або запуск програми в режимі який дозволяє відловити різного роду помилки спрямованості і ніяк не взаємодіє безпосередньо з тією частиною програми, для якої не був призначений.

Прикладом модуля може виступати клас Router, який би використовувався для визначення оброблюваних в додатку маршрутів. Будь-які не зазначені маршрути приводили б користувача на сторінку з кодом помилки 404.

Варто відзначити, що розробка додатків на основі модульної архітектури на перших етапах вже кілька вище за вартістю, ніж розробка аналогічного монолітного додатки, оскільки при внесенні нового функціоналу завжди повинен вставати питання про те, який конкретний модуль буде реалізовувати цей функціонал. Однак зростання вартості утримання подібних додатків не такий різкий, що дозволяє підтримувати дані додатку досить довго.

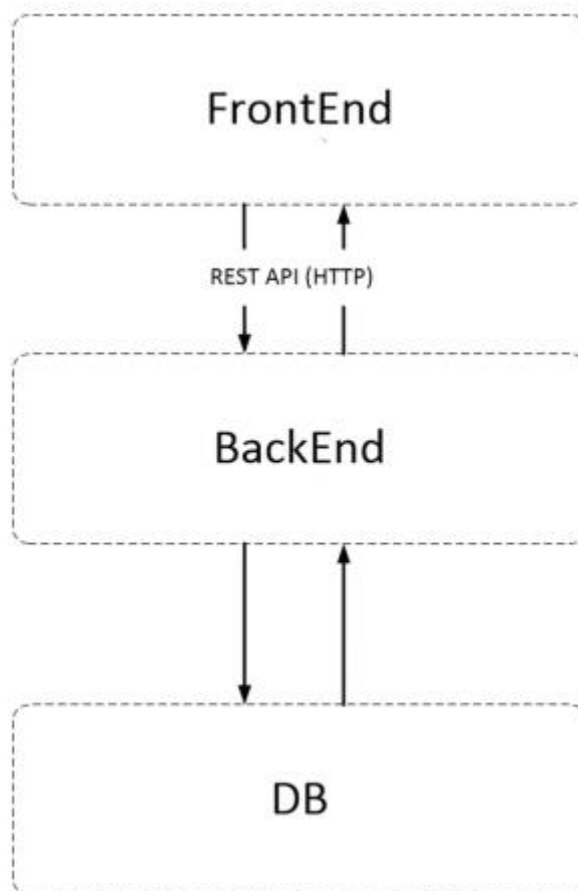


Рис. 2.6 Модульний за стосунок, що складається з клієнтської та серверної частин

На рисунку 2.6 наведений приклад структури модульного додатка, який, припустимо також розташовується на сервері з рисунка 2.5, і яке складається з двох модулів: клієнтської частини і серверної частини. Кожна частина реалізує свій функціонал: клієнтська частина відповідає за висновок призначеного для користувача інтерфейсу, а серверна частина за все бізнес-логіку додатка, виконуючи всі вимогливі обчислення на стороні сервера, в тому числі роботу з базою даних.

Клієнтська частина слугує для виведення одержуваних від серверної частини даних. Остання в свою чергу віддає дані за запитом від клієнтської

частини, а також включає в себе всю основну бізнес-логіку додатка, оскільки клієнт повинен залишатися якомога легшим в плані розміру підсумкового файлу.

Модульний підхід дозволив розробляти більш складні додатки, що складаються з безлічі різних модулів. Розвиток даного підходу призвів до того, що деякі модулі почали виносити на окремі апаратні частини, що таким чином призвело до виникнення цілих окремих сервісів, а відповідно сервіс-орієнтованого підходу.

Сервіси є окремими повністю самодостатніми модулями, що володіють власною апаратною базою, а саме розташовуються на окремому сервері. Крім того, вони можуть володіти власною базою даних, а оскільки вони розташовуються на окремих апаратних пристроях, навіть якщо віртуально, то і взаємодія між сервісами здійснюється асинхронно, що може надати як певні переваги, так і деякі недоліки. Однак, одним з ключових переваг SOA це те, що вона надає можливість незалежного масштабування компонентів, маючи на увазі збільшення потужностей тільки того сервісу, який цього вимагає, не зачіпаючи апаратну частину інших.

Можливість розробки цілісного додатку сервіс-орієнтованим підходом, надає можливість розробки сервісів на різних мовах програмування. Простий приклад: сервіс доставки повідомлень може бути написаний на NodeJS, де мовою програмування використовується JavaScript, в той час як основний додаток, що використовує цей сервіс, може складатися з зв'язки React на клієнтській частині і PHP на серверній частині. Для взаємодії цих двох частин необхідно тільки правильно використовувати програмний інтерфейс сервісу в основній частині програми.

Розробка сервісів відбувається окремо один від одного, а тому зміни в коді одного сервісу будуть впливати тільки на нього, якщо тільки не змінилася видача даних в програмному інтерфейсі сервісу, що відбувається вкрай рідко, оскільки з форматом виведення визначаються ще не перших етапах розробки сервісу.

На рисунку 2.7 зображена топологія мережі сервіс-орієнтованого веб-додатку типу «кільце».

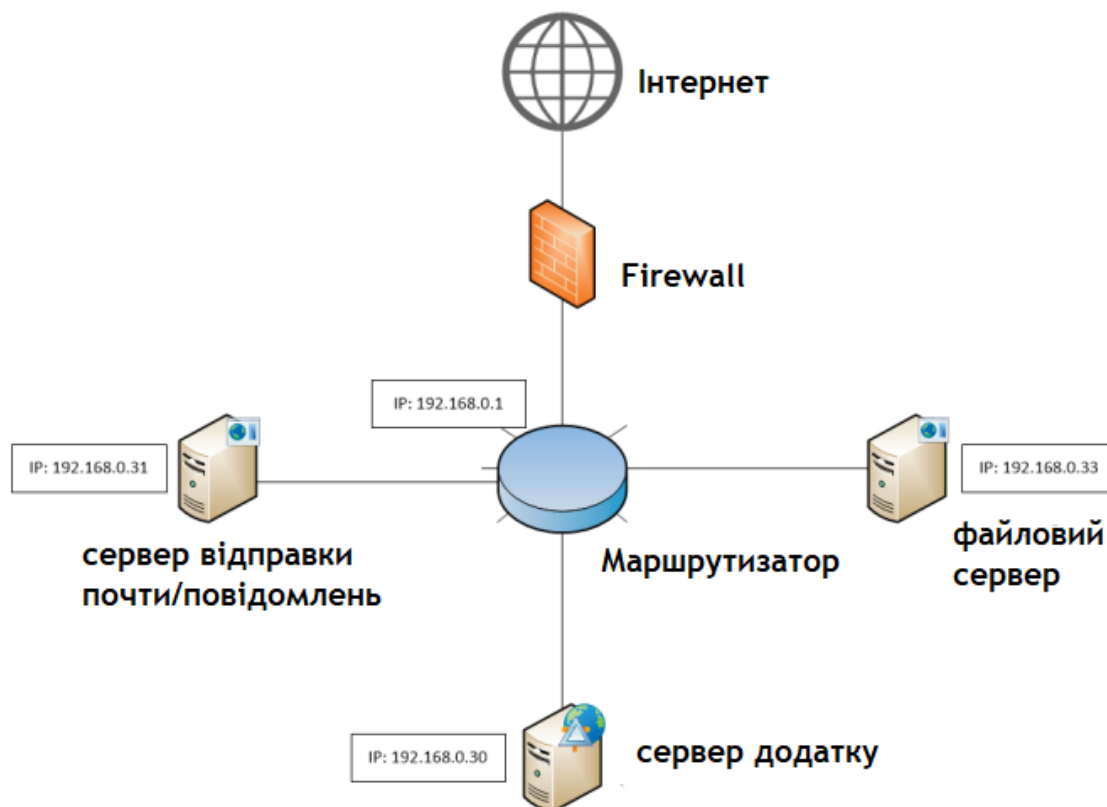


Рис. 2.7 Топологія мережі сервіс-орієнтованого додатку

В рамках цієї програми користувач завжди потрапляє на сервер додатку з IP-адресою 192.168.0.30 і отримує загальний інтерфейс для роботи з користувачем. Припустимо, що вебзастосунок відправляє пошту і повідомлення, а відповідно або цей функціонал потрібно реалізувати на сервері додатків, що є модульним підходом, або виділити йому окремий сервер в якості окремого сервісу, що і зображено на картинці вище.

Сервер відправки пошти і повідомлень забезпечує роботу однойменного сервісу, розташованого по IP-адресою 192.168.0.31 і основне завдання якого – відправка пошти та різного типу повідомлень. Сервіс також надає інтерфейс для взаємодії з ним іншими сервісами і, таким чином, сервер додатку, при

необхідності відправлення повідомлення одному або кільком користувачам, буде відправляти запит на відповідний сервіс і, таким чином, полегшуючи роботу собі.

При необхідності оновлення, всі проведені маніпуляції з програмної або апаратної частиною сервісу ніяким чином не торкнуться роботи сервера додатку.

Однак розробка програми сервіс-орієнтованим підходом не настільки проста. Справа в тому, що необхідно строго і чітко окреслити межі функціоналу того чи іншого сервісу. Крім того, оскільки кожен сервіс розробляється на окремій апаратній частині, то перевірити компілятором того чи іншого сервісу передані дані з іншої частини програми буде неможливо, оскільки схеми взаємодії між сервісами додатку в явному вигляді не існує і сервіс знає тільки про самого себе. Знати про інших сервісах він може тільки той, що ми самі вказали, а тому обробка помилок - це також додаткова проблема, якій варто приділити увагу при розробці додатків сервіс-орієнтованим підходом.

Одним з основних орієнтирів у виборі архітектурного підходу є вартість підтримки програми при тому чи іншому підході. На підставі всього вищесказаного, можна скласти наступний графік, що зображено на рисунку 2.8, який відображає співвідношення зростання вартості обслуговування до зростання функціоналу в додатку, побудованому на підставі кожного архітектурного підходу.

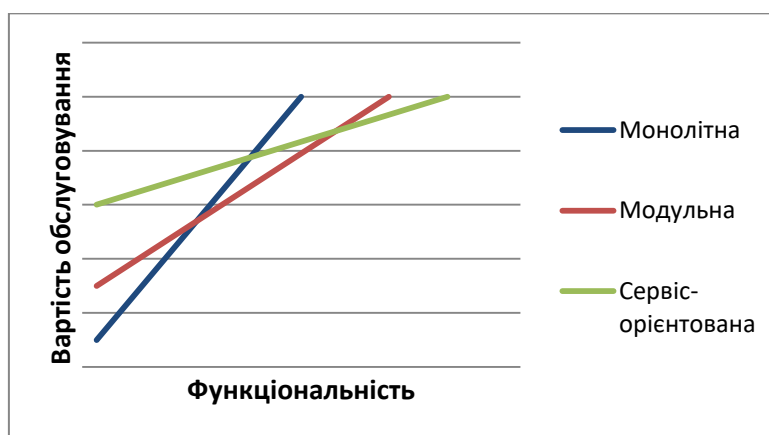


Рис. 2. 8 Діаграма відношення зростання вартості обслуговування до зростання функціональності

Як власне було сказано на початку і як показано на рисунку, монолітний підхід надає досить швидкий старт і вимагає найменше витрат, однак вартість обслуговування додатків, розроблених на основі даного підходу зростає досить швидко. Модульний підхід вимагає певних витрат на старті, однак зростання вартості обслуговування не такий різкий, що дозволяє підтримувати дані системи набагато довше. Нарешті, SOA-підхід вимагає серйозних витрат на старті, що включає в себе досить комплексне планування того як буде розвиватися система [20].

2.4 Вибір СКБД

Система керування базами даних (СКБД) - це комплекс програмно-мовних засобів, що дозволяють створити бази даних і управляти даними. Іншими словами, СКБД - це набір програм, що дозволяє організовувати, контролювати і адмініструвати бази даних [21]. Більшість сайтів не можуть функціонувати без бази даних, тому СКБД використовується практично всюди.

Основні функції СКБД:

- керування даними в зовнішній пам'яті (на дисках);
- керування даними в оперативній пам'яті з використанням дискового кешу;
- журналізація змін (збереження історії), резервне копіювання і відновлення бази даних після збоїв;
- підтримка мов БД (мова визначення даних, мова маніпулювання даними).

Кожна СКБД ґрунтується на будь-якій моделі даних, це є одним з ознак класифікації. За моделлю даних СКБД бувають:

- 1) Ієрархічні. У цій моделі даних використовується уявлення БД у вигляді дерева, що складається з даних різних рівнів.

- 2) Мережеві. Дана модель є розширенням ієрархічного підходу. Ієрархічна модель має на увазі, що запис-нащадок може мати строго одного предка, в той час як в мережевій структурі нащадок може мати будь-яку кількість предків.
- 3) Реляційні. СКБД, орієнтовані на організацію даних як набір пов'язаних записів і атрибутів в двовимірній таблиці.
- 4) Об'єктно-орієнтовані. Для управління БД, заснованими на об'єктній моделі даних. Як правило ґрунтуються на об'єктно-орієнтованих мовах програмування.
- 5) Об'єктно-реляційні. Об'єднує в собі концепції реляційної моделі з додатковими об'єктно-орієнтованими можливостями.

Сьогодні, як і раніше найбільш популярними при створенні веб-додатків і сервісів залишаються реляційні бази даних. Для керування базами даних використовується мова SQL (Structured Query Language - структурована мова запитів). Спочатку SQL був інструментом роботи користувача з базою даних, однак з часом мова ускладнилась і стала швидше інструментом розробника, ніж кінцевого користувача [22].

Різні рейтинги найпопулярніших СКБД очолюють MySQL, Microsoft SQL Server, PostgreSQL, MongoDB, Oracle, більш точний рейтинг наведено на рисунку 2.9, дані неведені станом на 2019 рік.

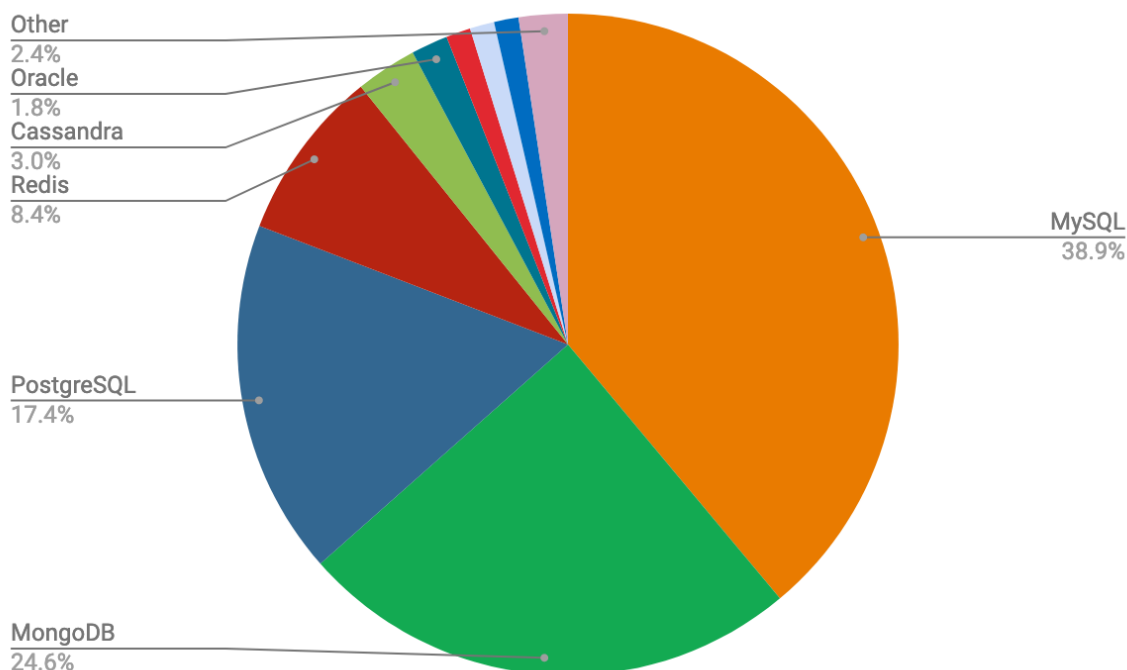


Рис. 2.9 Рейтинг популярності СКБД станом на 2019

Розглянемо популярні СКБД більш детально.

MySQL вважається однією з найпоширеніших СУБД. MySQL - реляційна СУБД з відкритим вихідним кодом, головними плюсами якої є її швидкість і гнучкість, яка забезпечена підтримкою великої кількості різних типів таблиць. Крім того, це надійна безкоштовна система з простим інтерфейсом і можливістю синхронізації з іншими базами даних. У сукупності ці фактори дозволяють використовувати MySQL як великим корпораціям, так і невеликим компаніям.

Microsoft SQL Server як випливає з назви, фірмова СУБД, розроблена Microsoft. Оптимальна для використання в операційних системах сімейства Windows, проте може працювати і з Linux. Система дозволяє синхронізуватися з іншими програмними продуктами компанії Microsoft, а також забезпечує надійний захист даних і простий інтерфейс, проте відрізняється високою вартістю ліцензії і підвищеним споживанням ресурсів. В цілому, однак, зберігає свою

популярність, в чималому ступені через те, що продукти корпорації Microsoft використовуються багатьма компаніями.

СУБД PostgreSQL - ще одна популярна і безкоштовна система. Найбільше застосування знайшла для управління БД веб-сайтів і різних сервісів. Вона універсальна, тобто підійде для роботи з більшістю популярних платформ. При цьому PostgreSQL - об'єктно-реляційна СУБД, що дає їй деякі переваги над іншими безкоштовними СУБД, в більшості є реляційними.

Oracle. Перша версія цієї об'єктно-реляційної СУБД з'явилася в кінці 70-х, і з тих пір зарекомендувала себе як надійна, функціональна і практична. СУБД Oracle постійно розвивається і допрацьовується, спрощуючи установку і первинне налаштування та розширюючи функціонал. Однак істотним мінусом даної СУБД є висока вартість ліцензії, тому вона використовується в основному великими компаніями і корпораціями, які працюють з величезними обсягами даних.

2.5 Вибір програмних засобів для розробки вебзастосунку

Бекенд(backend) – програмно-апаратна складова частина сервісу. Бекенд відповідає за функціонування внутрішньої частини вебзастосунку та недоступний для користувача. Він є так званим «прошарком» між користувацьким інтерфейсом та БД. Для наочного розуміння на рисунку 2.10 представлена схема роботи вебзастосунку. На схемі детально видно місце розташування бекенду в системі вебзастосунку.

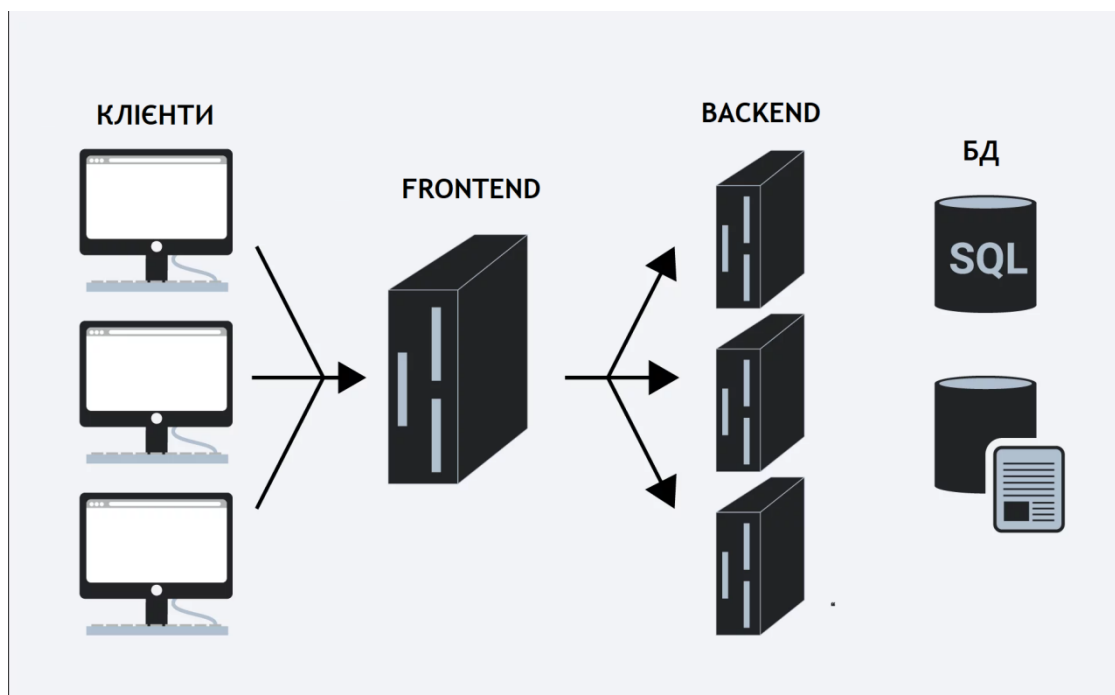


Рис 2.10 Схема роботи вебзастосунку

Backend - своєрідний мозок сайту, який отримує інформацію, дані та HTTP-запити від клієнтської частини, обробляє їх і виробляє відповідні дії. Бекенд - невидима для користувача частина ресурсу, яка розташовується на віддаленому сервері і забезпечує безперебійну роботу сайту.

Розробка backend частини вважається більш складним завданням, в порівнянні зі створенням frontend частини, в тому числі з-за необхідності глибоких знань мов програмування і архітектури. Для прискорення і спрощення роботи бекенд-розробники використовують фреймворки, що представляють собою якийсь кістяк або вже готову структуру серверної частини сайту з великим набором інструментів, функцій і даних. Розглянемо найпопулярніші фреймворки, за допомогою яких створюються серверні частини сайтів [23].

Laravel – PHP-framework Laravel за короткий термін став популярним у багатьох американських і європейських розробників, завдяки об'єктно-реляційному відображенню Eloquent й інших видів зв'язку безлічі варіантів додаткових компонент. У цьому фреймворку реалізований шаблонизатор Blade, а

також передбачена можливість автоматичного завантаження класів, що дозволяє розробникам відмовитися від ручного підключення файлів і завантаження невикористовуваних компонент. До переваг Laravel можна віднести інтерфейс командного рядка Artisan, що спрощує механізм роботи з інтегрованими командами і відкриває можливість створення своїх, простий синтаксис і зручні вбудовані механізми. Фреймворк оснащений значним списком функцій, які прискорюють і покращують роботу програмістів. А завдяки системі міграцій істотно спрощується процес розгортання та оновлення створених продуктів. До недоліків Laravel можна віднести відсутність користувальницької документації російською мовою, проблеми сумісності між різними версіями фреймворка, незручне розташування каталогів і файлів [24].

Ruby on Rails – фреймворк, відомий також як RoR, написаний на динамічній мові програмування Ruby. Головним достоїнством даного фреймворка є висока швидкість розробки і зрозумілий синтаксис, завдяки чому розробники можуть легко читати код, написаний навіть сторонніми фахівцями, і створювати інтернет-ресурси. У розпорядженні програмістів - величезна база готових шаблонів, плагінів і технічної документації, що допомагає в програмуванні бекенд-частини сайту. При роботі з цим framework забезпечується високий ступінь безпеки проекту. Разом з великим набором вбудованих інструментів RoR може похвалитися і хорошою системою автоматизованого тестування, що дозволяє швидко налагоджувати програму, виявляти і виправляти помилки. До недоліків фреймворка слід віднести складність в освоєнні, непростий процес розгортання і старту на продакшені, і набагато менша кількість фахівців, в порівнянні з розробниками на PHP і Java.

Django – фреймворк спрощує роботу програмістам на Python, володіє величезною стандартної бібліотекою і додатковими функціями, які прискорюють процес написання коду. В Django передбачена маршрутизація і відмінний шаблонизатор, є можливість масштабованості, панель адміністратора, міграція

бази даних, спрощена система аутентифікації користувачів. Даний framework є досить гнучким і багатofункціональним. Він задає структуру всього проекту, що допомагає фахівцям краще і швидше в ньому орієнтуватися. Ще одна перевага цього фреймворка - високий рівень безпеки. Django - монолітний framework з величезним співтовариством, але розвивається порівняно повільно. Ще до недоліків можна віднести спільне розгортання компонент, складність адаптації Django ORM.

Node.js – платформа з відкритим вихідним кодом запускається в V8 і використовує популярну мову JavaScript. Це дозволяє писати і серверну, і клієнтську частини ресурсу на одній мові. Фреймворк повноцінно працює з різними базами даних, має великий інструментарій, справляється з серйозним навантаженням і демонструє хорошу швидкість роботи. Крім вбудованої бібліотеки framework підтримує підключення зовнішніх модулів і пакетів, які можна створювати і самому. Node.js є однією з найпопулярніших платформ, його активно використовують в Google, Microsoft, Ebay, Uber і в інших корпораціях зі світовим ім'ям. Завдяки такій популярності і величезній армії користувачів, фреймворк продовжує активно розвиватися і поліпшуватися. Але при цьому він не позбавлений і недоліків. Серед них - відсутність основних інструментів і бібліотек, безліч альтернативних варіантів яких розробникам доводиться випробувати на собі. Мова JavaScript порівняно складна у вивченні, відповідно і з освоєнням Node.js можуть виникнути труднощі.

Ідеального фреймворку під всі завдання немає, і в кожному конкретному випадку необхідно вибирати найбільш підходящий framework і мову програмування.

Frontend - це візуальна частина web-додатків (вебсайтів), з якою користувач взаємодіє і контактує особисто. У Frontendі міститься відображення функціональних завдань, призначеного інтерфейсу для користувача, що

виконуються на стороні клієнта, а також обробка запитів користувачів. По суті, фронтенд - це те, з чим стикається користувач при відкритті web-сторінки.

У свою чергу, вебзастосунок - клієнт-серверний додаток, в якому клієнтом виступає браузер, а сервером – віддалений web-сервер. Логіка web-додатків розподілена між сервером і клієнтом. Зберігання даних відбувається в основному на сервері, обмін інформацією та даними відбувається по мережі. Простими словами, це все те, що може бачити користувач і ті дії які виконуються щоразу, коли відбувається підключення до мережі та в браузері відкривається відповідний вебзастосунок [25].

Ставлючи за мету створення потрібного і доступного продукту (веб-додатку) фронтенд-розробнику потрібно взаємодіяти з іншими програмістами, маркетологами, дизайнерами, аналітиками та іншими фахівцями.

Компоненти фронтенд розробки:

- HTML (HyperText Markup Language) простими словами - це мова гіпертекстової розмітки елементів, компонентів і документів на сторінці, і їх взаємодію в загальній структурі веб-сторінки.
- CSS (Cascading Style Sheets) - це мова характеристики і стилізації зовнішнього вигляду документа. За допомогою CSS-коду веб-переглядач розуміє, як необхідно відобразити елементи. CSS створює кольори, шрифти, визначає розташування блоків сайту, тощо. Також адаптує один документ в різних стильових обрамленнях, виводить передачу на екран або для читання голосом.
- JavaScript - мова, створена оживити веб-сторінки. Завдяки JavaScript - відгукуванню на дії користувача, обробка натискань клавіш, переміщення курсора, кліки мишею розпізнаються. JavaScript також надає можливість вводити повідомлення, відправляти запити на сервер, а також завантажувати дані без перезавантаження сторінки, тощо.

Написання фронтенд частини на чистих HTML, CSS, JS в сучасному світі зустрічається крайне рідко. Оскільки для пришвидшення та полегшення написання існують різні фреймворки. Фреймворків насправді просто безліч, але поговоримо про основні (найбільш поширені).

React.js. Гідний JavaScript фреймворк, випущений компаніями Instagram і Facebook в 2013 році. Використовуючи його, програміст зможе легко і просто створювати складні та масштабні динамічні додатки. Найчастіше використовується для побудови інтерфейсів призначених для користувача. Переваги:

- Кредо React: «Вивчи один раз - пиши всюди».
- Безкоштовний та Open Source(відкритий).
- Може використовувати код написаний раніше.
- Підтримує віртуальну функціональність DOM.

Не зважаючи на плюси, фреймворк містить хоч і не значні, але недоліки, такі як:

- алгоритм Virtual DOM досить неточний і повільний;
- для спілкування з сервером потрібне асинхронне програмування, що є досить складним.

Наступний фреймворк **Angular**. AngularJS - це програмний комплекс для розробки веб-додатків від компанії Google. Відмінно підходить для динамічних вебзастосунків, з використанням HTML для статичних веб-сторінок. Необхідний фреймворк не тільки для розробників програмного забезпечення, але й для дизайнерів. AngularJS, Angular 2 і Angular 4 широко поширені фреймворки, які не втрачають своєї популярності [26]. Переваги:

- Відкритий вихідний код.
- Доступна функція збереження частин коду для використання в наступному.

- Менша кількість помилок, оскільки прив'язка даних будується на базі власних елементів.
- Підтримка різних елементів MVC.
- Відмінно працює в середовищі Agile.
- Велика кількість інструментів для тестування.

Недоліки:

- Складність в освоєнні новачку.
- Складність архітектури, в порівнянні з Vue.
- Величезна API Angular, для вдалого початкового знайомства необхідно на старті познайомитись з великою кількістю концепцій.

Vue.js. Фреймворк використовується для створення користувацьких інтерфейсів. Розроблений з нуля для ширшої застосованості. Складається з доступної базової бібліотеки, яка працює виключно з рівнем уявлення, і екосистеми підтримки великої множини бібліотек, що допомагає справлятися з масштабними вебзастосунками. Переваги:

- Не потребує будь-яких компіляторів за замовчуванням.
- Трансформується в фреймворк з бібліотеки в процесі використання.
- Надає можливості керування просунутими односторінковими додатками.
- Відмінний баланс між читаемістю, ремонтпридатністю коду і написанню самого коду.

Недоліки:

- Незначні помилки часу виконання в шаблонах;
- Складний компонентний підхід в порівнянні з React.

Ember.js. Один з найпопулярніших MVC JavaScript фреймворків. Поява анонсована в 2011 році під відкритим вихідним кодом. За допомогою його можна досить легко створювати масштабовані односторінкові веб-додатки і забезпечувати двосторонню прив'язку даних. Відмінно виконує DOM-рендеринг

на стороні сервера. Підтримується багатьма популярними сайтами, такими як: LinkedIn, Vine, Discourse, Groupon. Переваги:

- Простота налаштування.
- Розгортає великі інтерфейси.
- Двостороння прив'язка даних.

Недоліки:

- жорстка структура проектів;
- немає базового набору UI-елементів.

jQuery – це бібліотека, хоч і найстаріша і найпопулярніша, але не фреймворк. jQuery і JavaScript давня і міцна зв'язка яка широко поширена. При використанні ліцензії MIT ця бібліотека дає розробникам можливість писати більш короткий код, скорочуючи тим самим робоче навантаження [27]. jQuery підтримує маніпуляції з DOM і разом з CSS може вирішити будь-яку задачу.

Плюси:

- широко розповсюджений завдяки швидкій обробці.
- Одноманітна, очікувана поведінка в різних браузерах.
- Дуже добре підходить як початківцям для простих додатків, так і досвідченим фахівцям.

Недоліки:

- безліч функцій, що полегшують роботу з DOM, вже реалізовані нативно;
- неможливість нормальної реалізації об'єктно-орієнтованої підтримки.

2.6 Особливості застосування бездротових технологій для передачі даних

Бездротові технології передачі даних є важливою складовою сучасного світу зв'язку і з'єднання. Вони забезпечують безпроводну передачу і отримання

даних між пристроями, що відкриває широкі можливості для спілкування, доступу до інформації та автоматизації процесів.

Особливості застосування бездротових технологій для передачі даних включають:

1. Мобільність: бездротові технології дозволяють з'єднуватися з мережею та передавати дані, необхідні для роботи, з будь-якого місця, де є доступ до мережі. Це дозволяє забезпечити постійну зв'язність під час пересування, що особливо важливо для мобільних пристроїв, таких як смартфони, ноутбуки, планшети та інші пристрої.

2. Висока швидкість передачі даних: бездротові технології постійно розвиваються і надають все більшу швидкість передачі даних. Наприклад, стандарти Wi-Fi, такі як 802.11ac та 802.11ax (відомий як Wi-Fi 6), забезпечують значно вищу швидкість передачі даних порівняно з попередніми версіями. Це дозволяє швидше завантаження великих файлів, стрімінг відео високої якості та інші вимогливі застосунки.

3. Легкість встановлення: бездротові технології не потребують проводів або фізичного з'єднання між пристроями. Це робить їх легкими у встановленні та використанні. Для багатьох бездротових технологій достатньо мати сумісний пристрій і доступ до мережі. Наприклад, для підключення до Wi-Fi мережі потрібно лише вибрати доступну мережу та ввести пароль.

4. Масштабованість: бездротові технології можуть бути легко масштабовані для використання в різних масштабах і на різних відстанях. Наприклад, Wi-Fi мережа може бути розширена шляхом додавання додаткових точок доступу для забезпечення покриття великих приміщень або будівель. Бездротові технології також дозволяють підключати велику кількість пристроїв одночасно, що робить їх ефективними для використання в інтернеті речей (IoT) та інших сценаріях, де багато пристроїв потребують з'єднання з мережею.

5. Безпека: забезпечення безпеки є важливим аспектом бездротових технологій передачі даних. Стандарти безпеки, такі як WPA2 (Wi-Fi Protected Access 2) і WPA3, використовуються для захисту бездротових мереж від несанкціонованого доступу та шифрування переданих даних. Додаткові заходи безпеки, такі як використання віртуальних приватних мереж (VPN), можуть бути використані для забезпечення конфіденційності та захисту від зловмисників [28].

Ці особливості роблять бездротові технології незамінними для передачі даних у різних сферах, від домашнього використання до бізнесу та індустрії.

Для бездротової передачі даних з МК на смартфон пацієнта можна використовувати Bluetooth або Wi-Fi. Переваги використання Bluetooth для передачі даних з Arduino на смартфон має свої переваги в порівнянні з Wi-Fi. Ось деякі з них:

1. Енергоефективність: Bluetooth має значно нижчий рівень споживання енергії порівняно з Wi-Fi. Це особливо важливо для проектів, які працюють на батареї, так як енергозбереження дозволяє подовжити тривалість роботи пристрою.

2. Простота налаштування: підключення по Bluetooth може бути простіше встановити порівняно з Wi-Fi, особливо у випадках, коли необхідно швидко встановити бездротове з'єднання між Arduino і смартфоном. На відміну від Wi-Fi, яке вимагає налаштування точки доступу, пароля та інших параметрів мережі, Bluetooth може просто паруватися з пристроями.

3. Прямий зв'язок: Bluetooth дозволяє встановлювати безпосереднє з'єднання між Arduino і смартфоном. Це означає, що ви можете взаємодіяти з Arduino безпосередньо з мобільного додатку, навіть без доступу до Інтернету або точки доступу Wi-Fi. Це корисно в ситуаціях, коли ви знаходитесь в місці без доступу до Wi-Fi.

4. Мінімальна інтерференція: оскільки Bluetooth має менший радіус дії порівняно з Wi-Fi, ймовірність інтерференції від інших пристроїв є меншою. Це

може бути корисно в обстановках, коли є багато бездротових пристроїв поряд, як, наприклад, на переповненому Wi-Fi каналі.

5. Масштабованість: хоча Wi-Fi зазвичай має вищу швидкість передачі даних порівняно з Bluetooth, Bluetooth може бути більш масштабованим для проектів, які вимагають з'єднання з великою кількістю пристроїв одночасно. Bluetooth може підтримувати підключення до декількох пристроїв одночасно, що робить його ідеальним для застосунків Інтернету речей (IoT) [29].

Враховуючи ці переваги, використання Bluetooth для передачі даних з Arduino на смартфон більш пріоритне в даній роботі.

2.7 Висновки до розділу

Мережева структура з використанням бездротової передачі даних для реалізації автоматизованих систем стає все більш популярною кожного дня. Зважаючи на всі переваги та беручи до уваги підвищену порівняно з іншими структурами складність розробки, вибір конкретно даної реалізації, зі створенням вебзастосунку для виконання МНР було визначено найдоцільнішим.

У процесі вибору апаратних засобів та стеку технологій для створення вебзастосунку, найоптимальнішими було визначено наступні:

- Arduino Uno в якості мікроконтролера;
- Bluetooth в якості бездротової технології передачі даних з МК на смартфон;
- MySQL в якості СКБД;
- Laravel для розробки бекенд частини вебзастосунку;
- Angular та Ionic в якості фреймворків для розробки фронтенд частини.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ, ТЕСТУВАННЯ

3.1 Схема реалізації апаратних засобів

В якості МК використовується Arduino Uno, який зображено на рисунку 3.1:

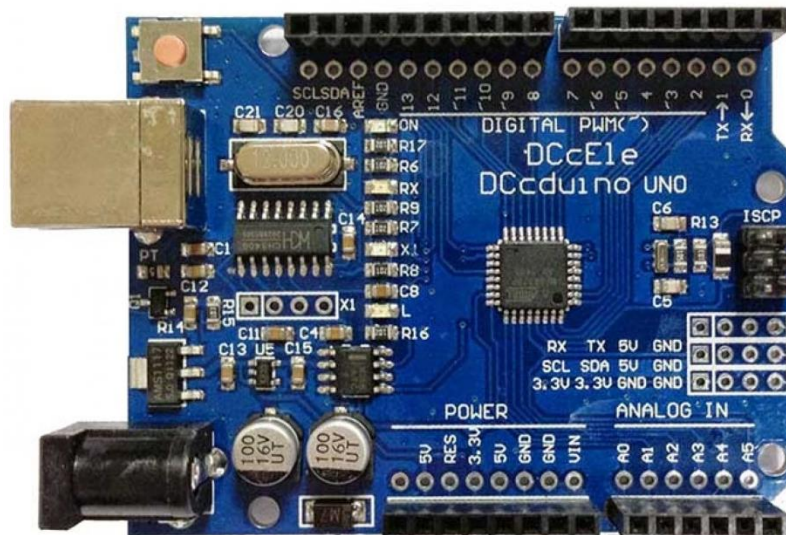


Рисунок 3.1 – Зовнішній вигляд Arduino Uno

Для забезпечення бездротового з'єднання в попередньому розділі було обрано Bluetooth, для реалізації даного протоколу обрано модуль HC-06, який зображено на рисунку 3.2:



Рисунок 3.2 – Зовнішній вигляд Bluetooth модуля HC-06

Відповідно до структури автоматизованої системи, для електрокардіографії обрано модуль ЕКГ AD8232, що зображено на рисунку 3.3:

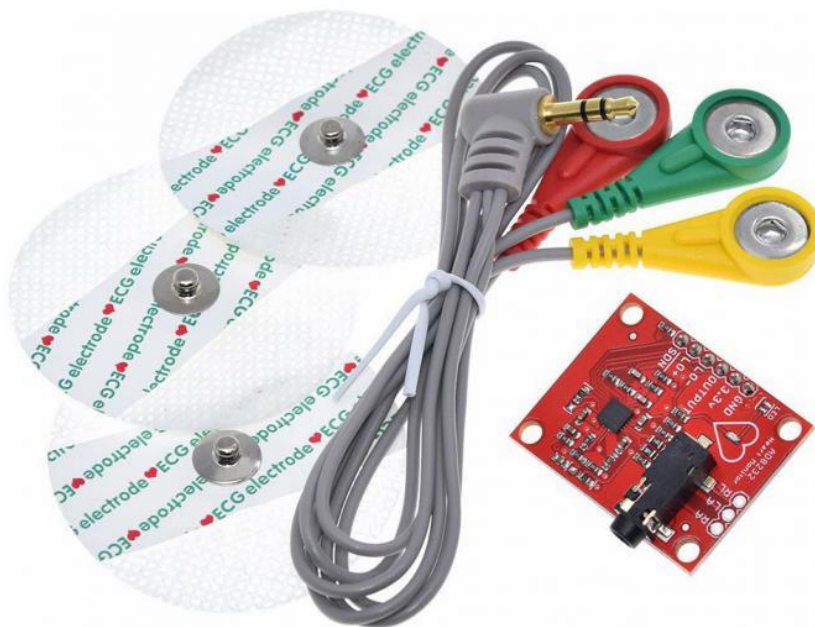


Рисунок 3.3 – Зовнішній вигляд ЕКГ модуля AD8232

Модуль включає в себе:

- основну плату;
- комплект з трьох самоклеючих електродів;
- кабель для підключення електродів до плати.

Схема підключення модулів до МК зображена на рисунку 3.4:

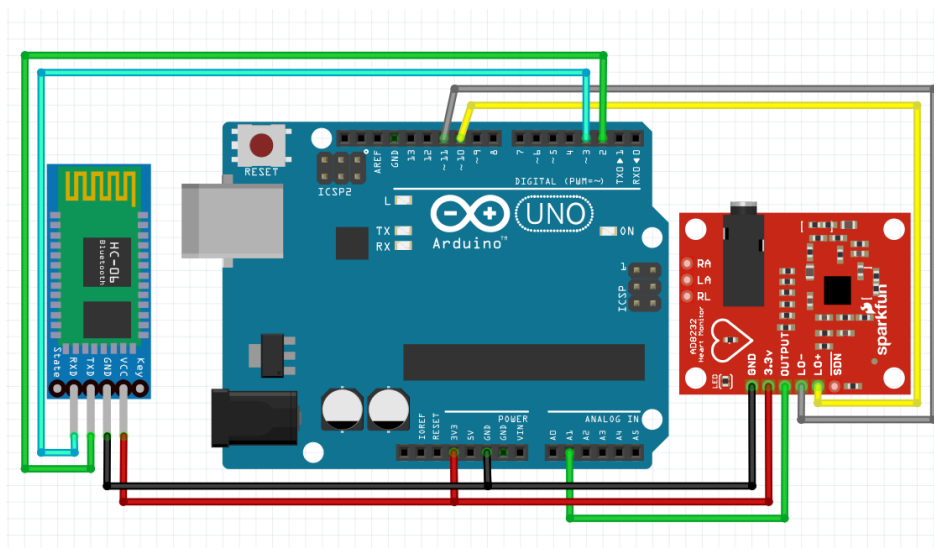


Рисунок 3.4 – Підключення модулів HC-06 та AD8232 до МК

Наступним кроком є реалізація прилада для фонетричних вимірювань. В даному приладові було прийняте рішення використовувати датчик тиску MPX2050GP (Рис 3.5). Даний датчик цілком підходить для визначених цілей, оскільки діапазон вимірювання входить в діапазон систолічного і діастолічного тиску.



Рисунок 3.5 – Зовнішній вигляд датчика тиску MPX2050GP

Також на даному датчику відносно невелика похибка. Детальні характеристики датчика вказані в таблиці 3.1. Також за інформацією з офіційного сайту розробника, датчик тиску цілком можна використовувати в медицині [31].

Таблиця 3.1 – Технічні параметри датчика тиску MPX2050GP

| | |
|-------------------|---|
| Робочий тиск | 50 кПа (375.03 мм. рт. ст.) |
| Вихідна напруга | 0 мВ ~ 40 мВ (при напрузі живлення 10 В) |
| Напруга живлення | 10В ~ 16В |
| Особливості | Компенсується температура (макс. +125 °С) |
| Чутливість | 0.8мВ/кПа (при напр. живлення 10 В) |
| Максимальний тиск | 29.01PSI (200 кПа) |

Але оскільки діапазон вихідної напруги лише від 0 до 40 мВ, а найвище значення тиску в манжеті, яке ми будемо використовувати, 180 мм.рт.ст., що відповідає 19,2 мВ на виході з датчика, при такому малому вихідному сигналі ми не зможемо побачити осциляції тиску, що будуть створюватись в манжеті. Для того, щоб на АЦП подавався сигнал, на якому будуть добре вирізнятись осциляції тиску, необхідно підсилити цей сигнал. Саме тому нам необхідний підсилювач сигналу, що буде збільшувати величину вихідного сигналу.

Сигнал необхідно підсилити приблизно в 200 разів, таким чином з діапазону 0...20 мВ буде отримано 0...4 В. Для цих цілей було вибрано операційний підсилювач AD620, який зображено на рисунку 3.6:

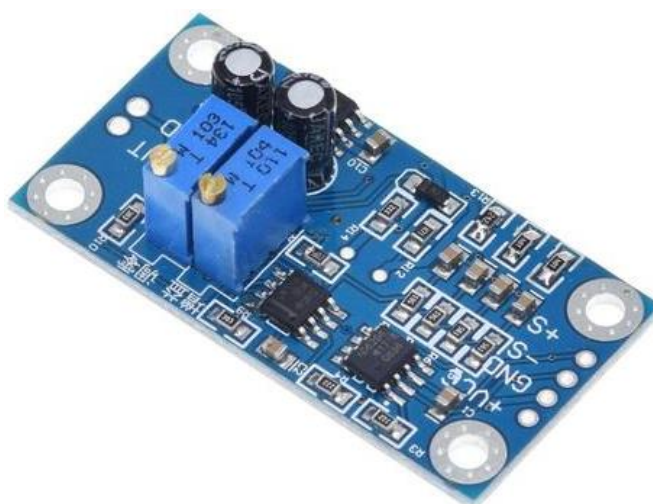


Рисунок 3.6 – Зовнішній вигляд операційного підсилювача AD620

Даний операційний підсилювач вирізняється простотою в використанні, непоганими характеристиками вихідного сигналу, та широким діапазоном напруги живлення. В технічних характеристиках підсилювача вказано, що коефіцієнт підсилення встановлюється резистором, що підключається між відповідними виходами.

Постійна і змінна складові сигналу в подальшому будуть подаватись на АЦП. Для даного приладу було вибрано АЦП ADS1115, що зображений на рисунку 3.7.

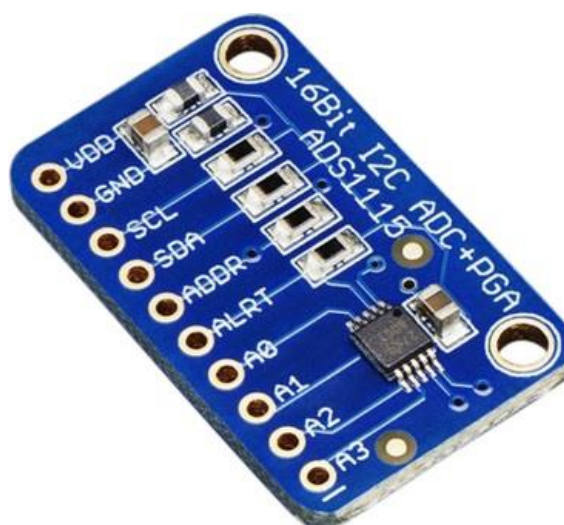


Рисунок 3.7 – Аналого-цифровий перетворювач ADS1115

Даний перетворювач має 16 розрядів. Це означає, що максимальне число, виміряне цим приладом, буде 32767 (для порівняння, вбудований АЦП в Arduino може видавати тільки 1024 значення), що буде відповідати 3,3 В при подачі відповідної напруги з мікроконтролера. Після створення сигналу, з яким зручно працювати, він буде передаватись на мікроконтролер для подальших операцій з ним.

Для керування тиску в системі були підібрані компресор та клапан. Компресор КРМ14А, вирізняється компактністю, невеликим рівнем шуму, і низькою напругою живлення. Основні технічні параметри вказані в таблиці 3.2.

Таблиця 3.2 – Технічні параметри компресору КРМ14А

| | |
|---------------------------|----------------------------------|
| Максимальний тиск | > 450 мм рт.ст. |
| Витрата повітря | > 0,8 л./хв. |
| Номінальна напруга | 3 В |
| Номінальний струм | <350 мА |
| Максимальний струм витoku | <3 мм рт.ст. / хв від 300 мм рт. |
| Шум | <52 дБ (30 см) |
| Термін служби | 50000 раз |

Даний компресор зображений на рисунку 3.8.



Рисунок 3.8 – Компресор КРМ14А

Також в приладі будемо використовувати клапан JQF1. Цей компонент малий по габаритам, також має низьку напругу живлення, доволі точно керується сигналом, і добре ізолює тиск в системі. Технічні характеристики клапану вказані в таблиці 3.3 і зображення клапану показане на рисунку 3.9.

Таблиця 3.3 – Технічні параметри компресору KPM14A

| | |
|-----------------------|--------------------|
| Робочий тиск | < 300 мм рт. ст. |
| Герметичність | < 3 мм рт. ст./мин |
| Номінальна напруга | 3 В |
| Номінальний струм | 75 мА |
| Номінальна потужність | 0,225 Вт |

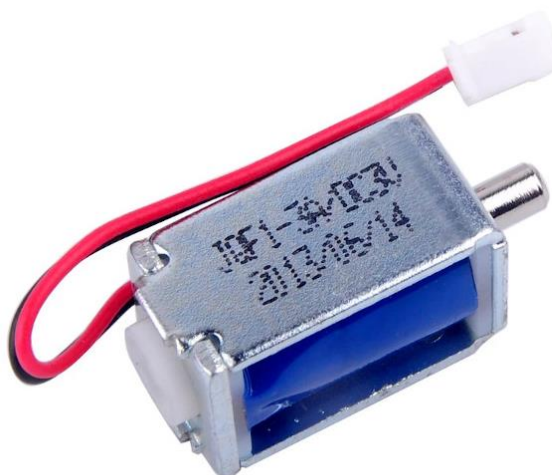


Рисунок 3.9 – Компресор KPM14A

Керування даними елементами буде реалізовано за допомогою польового NPN транзистора. На базу емітера подається керуючий сигнал. Коли значення сигналу велике (стан 1), струм буде протікати на землю, коло буде замкнене, і відповідно компресор і клапан будуть працювати (компресор буде нагнітати повітря, клапан – закриватись). Також на схемі бачимо діод, що паралельно підключений до компресору. Цей діод не дасть згоріти транзистору і контролеру в

Відповідно до визначеного та описаного вище апаратного забезпечення фізично реалізується автоматизована система. Для практичної реалізації слід виконати програмну частину розробки, для цього існують математичні моделі.

3.2 Математична модель

У ході побудови математичної моделі автоматизованої системи параметричного контролю стану пацієнта за групою показників виникла необхідність будови моделі тонометра.

Вимірювання, що здійснює тонометр проводяться у фіксовані моменти часу t . За ініціалізацією u проводиться включення компресора, що нагнітає повітря до манжети. Датчик тиску починає вимірювати тиск. Одночасно включається датчик підрахунку ударів пульсу. Повітря нагнітається до тих пір, поки датчик ударів пульсу не перестане їх фіксувати. Після фіксації цього моменту компресор перестає нагнітати повітря до манжети. Через проміжок часу відкривається випускний електромагнітний клапан і починається викид повітря. При першому ударі пульсу проводиться вимірювання фактичного значення величини систолічного тиску, а у момент останнього удару діастолічного. Одночасно підраховується кількість ударів пульсу. Зафіксовані виміряні значення виводяться як три компоненти вихідного вектора \bar{Y} . Представимо модель в графічному редакторі, рисунок 3.12:

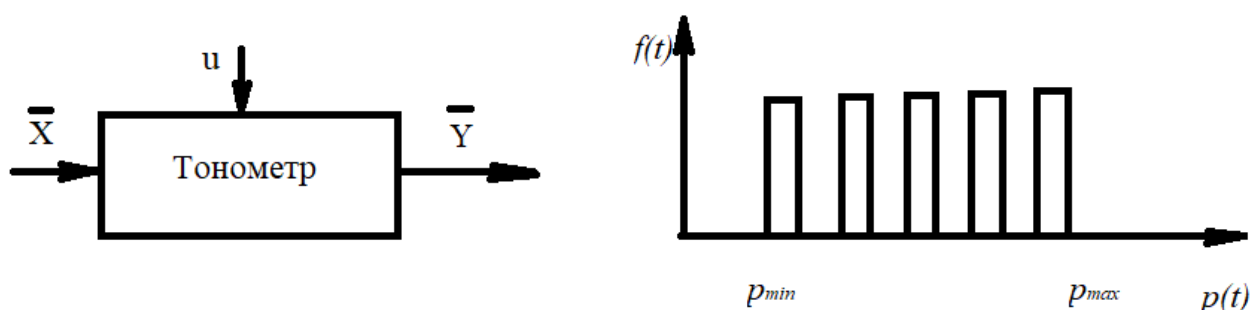


Рисунок 3.12 – Представлення графічним способом моделі

Введемо вхідну матрицю, що ініціює ініціалізацію:

$$u = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (3.1)$$

Наступним кроком слід представити модель роботи у період нагнітання повітря до манжети. Для цього, введемо позначення характеристик:

Компресор:

$$p(t) = p_0 + at + bt^2 \quad (3.2)$$

Датчик тиску:

$$p_o(t) = kp(t) \pm \Delta = k(p_0 + at + bt^2) \pm \Delta \quad (3.3)$$

Датчика ударів серця:

$$f_o(t) = af(t); \quad (3.4)$$

Введемо системний час i , за еталон часу візьмемо Δt , що визначено частотою роботи кварца ν .

$$\Delta t = \frac{1}{\nu}; t = i\Delta t = i\frac{1}{\nu}; \quad i = \overline{1, n}. \quad (3.5)$$

Визначимо найбільший тиск нагнітання компресором.

$$p_{max} = \begin{cases} \text{If } f_o(t) = af(t) \leq \Delta_{пч} \text{ THEN} & t = \frac{i}{\nu} \text{ and} \\ p(t) = p_0 + at + bt^2 = p_0 + a\frac{i}{\nu} + b\left(\frac{i}{\nu}\right)^2; \end{cases} \quad (3.6)$$

Тепер представимо представимо компоненти вектора Y :

Систолічний тиск

$$p_s = \left\{ p_o\left(\frac{i}{\nu}\right), \text{ if } p_o\left(\frac{i}{\nu}\right) < p_{max} \text{ and } f_o > 0 \text{ and } i_{max} = i \right. \quad (3.7)$$

Діастолічний тиск

$$p_{\partial} = \begin{cases} p_{\partial} \left(\frac{i}{v} \right), & \text{if } p_{\partial} \left(\frac{i}{v} \right) < p_{\max} \text{ and } f_{\partial} \leq 0 \text{ and } i_{\min}. \end{cases} \quad (3.8)$$

Пульс

$$P = \frac{i_{\max} - i_{\min}}{60v} \quad (3.9)$$

Тепер вихідне вимірювання представимо

$$Y = \begin{bmatrix} p_s, 0, 0 \\ 0, p_{\partial}, 0 \\ 0, 0, P \end{bmatrix} u^T = \begin{bmatrix} p_s \\ p_{\partial} \\ P \end{bmatrix} = [p_s, p_{\partial}, P]^T. \quad (3.10)$$

Таким чином отримана математична модель надає початкове розуміння майбутнього алгоритму роботи тонометра в автоматизованій системі та впроваджує ґрунтовність алгоритмізації.

Для моделювання роботи автоматизованої системи параметричного контролю стану пацієнта за групою показників виникла необхідність будови моделі Електрокардіографа. Вимірювання, що здійснює ЕКГ прилад проводяться у фіксовані моменти часу t . За ініціалізацією u проводиться включення.

Представимо спочатку як чорну скриньку електрокардіограф. Обмежимо розглядом моделі з трьома відведеннями. Введемо чотирьох мірну вектор функцію u активації функції входів X та Y виходів, які теж розглядаємо як чотирьох вимірні (3.11). Представимо модель в графічному редакторі, рисунок 3.13:

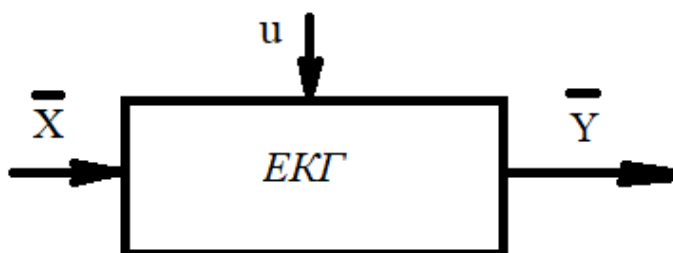


Рисунок 3.13 – Представлення графічним способом моделі

Таким чином отримуємо вектор функцію активації та функцію входів:

$$u = \begin{bmatrix} 1 \\ 1 \\ 1 \\ i \end{bmatrix}, \quad X = \begin{bmatrix} X_1 - X_4 \\ X_2 - X_4 \\ X_3 - X_4 \\ 1 \end{bmatrix} \quad (3.11)$$

Проведемо обчислення математичної моделі.

$$Y^T = \begin{bmatrix} \frac{\mathcal{E}_{1out}}{\mathcal{E}_{col}}(X_1 - X_4) \\ \frac{\mathcal{E}_{2out}}{\mathcal{E}_{col}}(X_2 - X_4) \\ \frac{\mathcal{E}_{3out}}{\mathcal{E}_{col}}(X_3 - X_4) \\ 1 \end{bmatrix}^T \begin{bmatrix} 1,0,0,0 \\ 0,1,0,0 \\ 0,0,1,0 \\ 0,0,0,i \end{bmatrix} = \begin{bmatrix} \frac{\mathcal{E}_{1out}}{\mathcal{E}_{col}}(X_1 - X_4), \frac{\mathcal{E}_{2out}}{\mathcal{E}_{col}}(X_2 - X_4), \frac{\mathcal{E}_{3out}}{\mathcal{E}_{col}}(X_3 - X_4), i \end{bmatrix} \quad (3.13)$$

Отримана математична модель у вигляді вектора надає початкове розуміння майбутнього алгоритму роботи електрокардіографа в автоматизованій системі та впроваджує ґрунтовність алгоритмізації.

3.3 Алгоритм роботи

Алгоритм роботи – це детальний набір інструкцій, які визначають послідовність дій, необхідних для досягнення певної мети або вирішення конкретної задачі. Він включає в себе опис кожного кроку, який потрібно виконати, враховуючи умови, обмеження та інші фактори, які можуть вплинути на процес виконання [33].

Для представлення алгоритму є безліч способів та методів, але найефективнішим та найпоширенішим прийнято вважати блок-схему.

За допомогою блок-схеми відобразимо алгоритм роботи системи параметричного контролю стану пацієнта, зображену на рисунку 3.14:



Рисунок 3.14 – Загальна блок-схема алгоритму роботи автоматизованої системи

Загальна схема алгоритму відображає процес роботи системи в цілому, що передбачає собою постійне очікування ініціації вимірювання. В момент передачі відповідного значення, система з стану очікування переходить в стан роботи. Спрацьовують відповідні підсистеми, такі як: термометр, тонометр, ЕКГ. Відобразимо відповідно блок схему роботи термометра, зображену на рисунку 3.15:



Рисунок 3.15 – Блок-схема алгоритму роботи термометру

Так, аналізуючи алгоритм, слід зазначити його простоту та не високу деталізацію, відповідно до поставленого завдання.

Наступною підсистемою, що буде розглянуто є система вимірювання тиску та пульсу, що позначена на загальній блок-схемі на рисунку 3.14 «Тонometr», так як дана підсистема є більш архітектурно складною, розробка алгоритму відповідно більш кропітка, результат у вигляді блок-схеми зображено на рисунку 3.16:



Рисунок 3.16 – Блок-схема алгоритму роботи тонометру

З наведеного вище алгоритму видно, що процес роботи умовно поділяється на 3 етапи:

1. Заповнення манжети повітрям за допомогою компресору.
2. Безпосередньо вимірювання значень.
3. Обчислення та інтерпретація даних.



Рисунок 3.17 – Блок-схема алгоритму роботи ЕКГ

3.4 Програмування інструкцій МК та датчиків

Програмування інструкцій мікроконтролера та датчиків – це процес створення програмного забезпечення для керування мікроконтролерами та отримання даних з датчиків.

Для перевірки справності роботи модулю серцевого ритму AD8232 та подальшого його налаштування слід використовувати наступний програмний код, вказаний розробниками даного модулю:

```
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  if((digitalRead(10) == 1) || (digitalRead(11) == 1))
  {
    Serial.println('!');
  }
}
```

```
}  
else{  
    Serial.println(analogRead(A0));  
}  
delay(1);  
}
```

Програмний код включає в себе налаштування пінів підключення модулю до основної плати Arduino Uno, перевірку підключення кожного з трьох електродів до пацієнта та виведення даних в монітор порту. В результаті, за допомогою розділу «Графік за послідовним підключенням» середовище Arduino IDE, можна в графічному вигляді подати отримані значення, тобто, представити графік ЕКГ.

Для реалізації тонометричних вимірювань відповідно до розробленої математичної моделі та алгоритму використовується код, наведений в додатку А1. Відповідно до алгоритму роботи, ключові значення, які оброблюються в застосунку це:

- P_MAX – тиск максимального нагнітання компресором.
- P_MIN – тиск мінімального нагнітання компресором.
- P_D – тиск діастолічний.
- P_S – тиск систолітичний.
- P – пульс.

Опрацювання даних відбувається в процесі виконання та відображаються до монітору порту за допомогою команди:

```
Serial.println('');
```

В результаті, в моніторі порту можна отримати дані процесу вимірювання, а також отримані значення.

Реалізація програмного коду для роботи термометру не викладає труднощів, оскільки використання рекомендованого виробником коду, приводить до справного та стабільного результату роботи. Результати роботи аналогічно попереднім виводяться в монітор порту.

3.5 Впровадження серверної частини та бездротової передачі даних на смартфон

Впровадження серверної частини застосунку слід розпочинати з остаточним визначенням структури бази даних MySQL, яку було обрано в

попередньому розділі. При реалізації структури БД було використано наступні сутності: «Лікар», «Пацієнт» та «Журнал», схема наведена на рисунку 3.18:

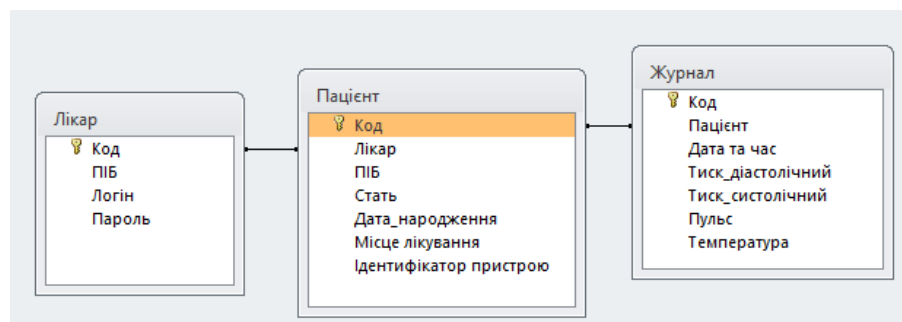


Рисунок 3.18 – Схема бази даних

Таблиця «Лікар» (doctor) містить автоінкрементний первинний ключ «id», прізвище, ім'я та по-батькові лікаря «name», а також логін для авторизації та пароль «login» та «password» відповідно. Для створення даної таблиці засобами MySQL використовується відповідний код:

```
CREATE TABLE doctor (`id` bigint(20) NOT NULL AUTO_INCREMENT,
`name` VARCHAR(30), `login` VARCHAR(12), `password` VARCHAR(15),
PRIMARY KEY(id));
```

Структуру даної таблиці відображено на рисунку 3.19:

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|-------------|-------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|--------------------|
| id | BIGINT(20) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| name | VARCHAR(30) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| login | VARCHAR(12) | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| password | VARCHAR(15) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

Рисунок 3.19 – Структура таблиці «Лікар»

Таблиця «Пацієнт» (patient) містить автоінкрементний первинний ключ «id», прізвище, ім'я та по-батькові «name», зовнішній ключ коду лікаря «doctor», стать «gender», дату народження «db», адресу місця в якому проводиться лікування «address» та ідентифікатор пристрою «mac». Для створення даної таблиці засобами MySQL використовується відповідний код:

```
CREATE TABLE patient ( `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
`name` VARCHAR(30), `doctor` INT, `gender` CHAR(1), `db` DATETIME,
2023 р.
```

```
`address` VARCHAR(55), `mac` VARCHAR(20), PRIMARY KEY (`id`), UNIQUE
INDEX `id_UNIQUE` (`id` ASC) VISIBLE, CONSTRAINT `id` FOREIGN KEY
() REFERENCES `mydb`.`doctor` ();
```

Структуру даної таблиці відображено на рисунку 3.20:

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G |
|-------------|-------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|
| id | BIGINT(20) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| name | VARCHAR(30) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| doctor | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| gender | CHAR(1) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| db | DATETIME | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| address | VARCHAR(55) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| mac | VARCHAR(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Рисунок 3.20 – Структура таблиці «Пацієнт»

Таблиця «Журнал» (journal) містить автоінкрементний первинний ключ запису «id», прізвище, ім'я та по-батькові «name», зовнішній ключ коду лікаря «doctor», стать «gender», дату народження «db», адресу місця в якому проводиться лікування «address» та ідентифікатор пристрою «mac». Для створення даної таблиці засобами MySQL використовується відповідний код:

```
CREATE TABLE journal ( `id` INT NOT NULL AUTO_INCREMENT,
`patient` BIGINT(20), `dt` DATETIME, `p_d` INT, `p_s` INT, `p` INT,
`t` FLOAT NULL, PRIMARY KEY (`id`), UNIQUE INDEX `id_UNIQUE` (`id`
ASC) VISIBLE, INDEX `patient_idx` (`patient` ASC) VISIBLE, CONSTRAINT
`patient` FOREIGN KEY (`patient`) REFERENCES `mydb`.`patient`
(`id`))
```

Структуру даної таблиці відображено на рисунку 3.21:

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G |
|-------------|------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|
| id | INT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| patient | BIGINT(20) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| dt | DATETIME | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| p_d | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| p_s | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| p | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| t | FLOAT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Рисунок 3.21 – Структура таблиці «Журнал»

Впровадження бездротової передачі даних зі смартфона на Arduino Uno слід розпочинати з підключення Bluetooth-модуля до МК та створення скетча для перевірки адекватності роботи. Для виконання даного завдання можна використовувати приведений нижче програмний код, який обробляє введене значення в Bluetooth-терміналі смартфона та відповідно до отриманого значення вмикатиме вбудований в плату світлодіод.

```
int value;
int LED = 13;

void setup()
{
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
}

void loop()
{
  if (Serial.available())
  {
    value = Serial.read();
    if (value == '1')
    {
      digitalWrite(LED, HIGH);
      Serial.println("On");
    }
    if ( value == '0')
    {
      digitalWrite(LED, LOW);
      Serial.println("Off");
    }
  }
}
```

Після завантаження скетчу, підключаємось до МК та запускаємо термінал на смартфоні, на рисунку 3.22 відображено як виглядає підключення:

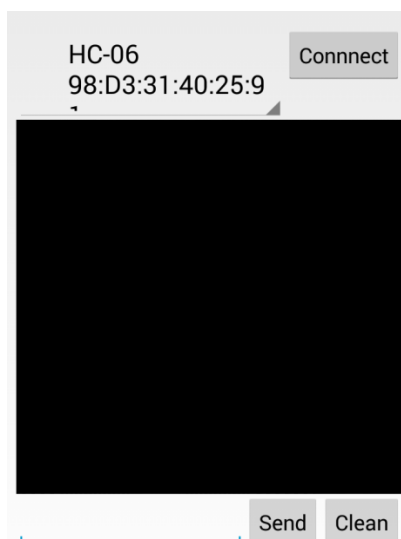


Рисунок 3.22 – Bluetooth-термінал при підключенні МК

Перевірка справності та функціональності даного рішення надає можливість подальшого написання програмного коду опрацювання ініціалізації параметричних вимірювань стану пацієнта відповідно до вказівки зі смартфона, детальний код наведено в додатку А4.

3.6 Реалізація вебзастосунку

Застосунок розроблено на базі класичної трьохланкової архітектури. Сигнал ініціалізації передається від лікаря (із використанням протоколу http) на вебсервер та записується у базу даних, далі передається на смартфон пацієнта з якого відповідний сигнал поступає на мікроконтролер. Розроблений користувацький інтерфейс підключається до вебсерверу та отримує із БД дані щодо стану пацієнта в реальному часі [34].

Розробка вебзастосунку починається з визначення з дизайном, який було обрано в мінімалістичному стилі в темних тонах.

Першопочатковою сторінкою слугує сторінка авторизації, на якій користувач вводить логін та пароль відповідно до власного облікового запису, форма авторизації виглядає наступним чином, рисунок 3.23:

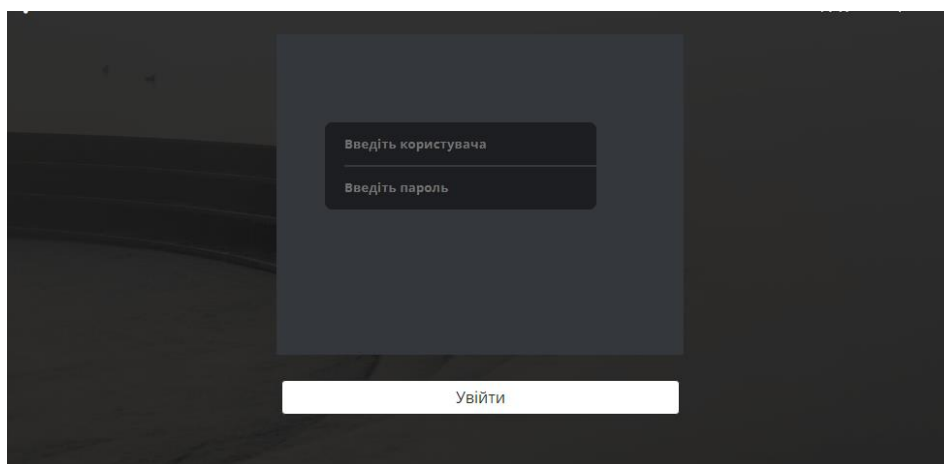


Рисунок 3.23 – Форма авторизації

Після успішної авторизації, відображається стартова сторінка, що містить інформацію про лікаря та відображає підключених пацієнтів, сторінка виглядає наступним чином, рисунок 3.24:

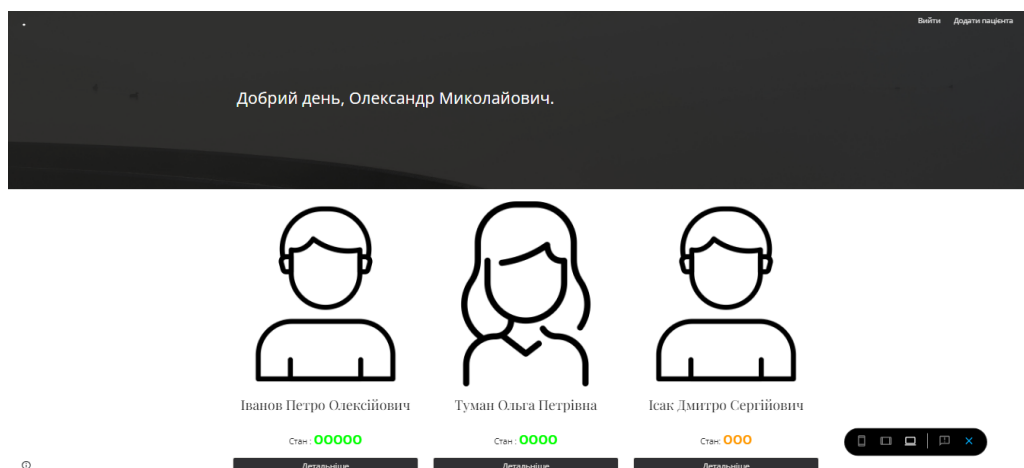


Рисунок 3.24 – Стартова сторінка

Додати пацієнта можливо за допомогою кнопки «Додати пацієнта» в куті екрану, або ж за допомогою відповідної піктограми, зображеної на рисунку 3.25:



Рисунок 3.25 – Піктограма додавання пацієнта

При натисненні відповідної кнопки, або піктограми, з'являється форма нового пацієнта, зображена на рисунку 3.26, вона містить основні необхідні дані, які зберігатимуться в базі даних на сервері.

A screenshot of a web form for adding a patient. At the top right are links "Вийти" and "Додати пацієнта". The name "Піб" is displayed in large font. To the left is a person icon with a "Завантажити зображення" button below it. To the right are input fields for "Стать:", "Дата народження:", "Місце лікування:", "Останні покази:", "Стан:", and "Адреса пристрою:". A "Підключитись" button is at the bottom right.

Рисунок 3.26 – Форма додавання пацієнта

Після заповнення відповідних полів відбудеться додавання пацієнта та його відображення на стартовій сторінці, для перегляду повної інформації необхідно натиснути кнопку «Детальніше». Відкриється нова сторінка, зображена на рисунку 3.27:



Рисунок 3.27 – Форма перегляду повної інформації пацієнта

Відповідно, кнопка «Підключитись» ініціює процес вимірювання показників, відповідний сигнал передається на сервер, а з серверу на мікроконтролер, процес вимірювання відбувається в режимі реального часу.

3.7 Тестування автоматизованої системи

Тестування автоматизованих систем відіграє важливу роль у забезпеченні якості програмного забезпечення та ефективного функціонування системи. Тестування надає можливість виявити помилки, дефекти та проблеми в програмному забезпеченні, які можуть бути критичними або неприпустимими на етапі розробки, коли продукт не надано в доступ користувачам.

Тестування слід проводити з найпростіших модулів поступово дійшовши до повної функціональної роботоздатності системи. Запустимо окремо модуль тестування роботи термометра, результат зображено на рисунку 3.28

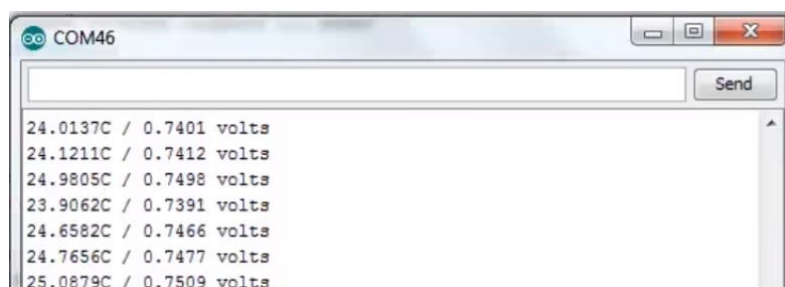


Рисунок 3.28 – Перевірка роботи термометру

Вимірювалась температура повітря в кімнаті, порівнюючи з показниками температури на термометрі, можна дійти до висновку, що показники відповідають дійсності.

Проведемо тестування роботи ЕКГ, в результаті отримуємо графік зображений на рисунку 3.29:

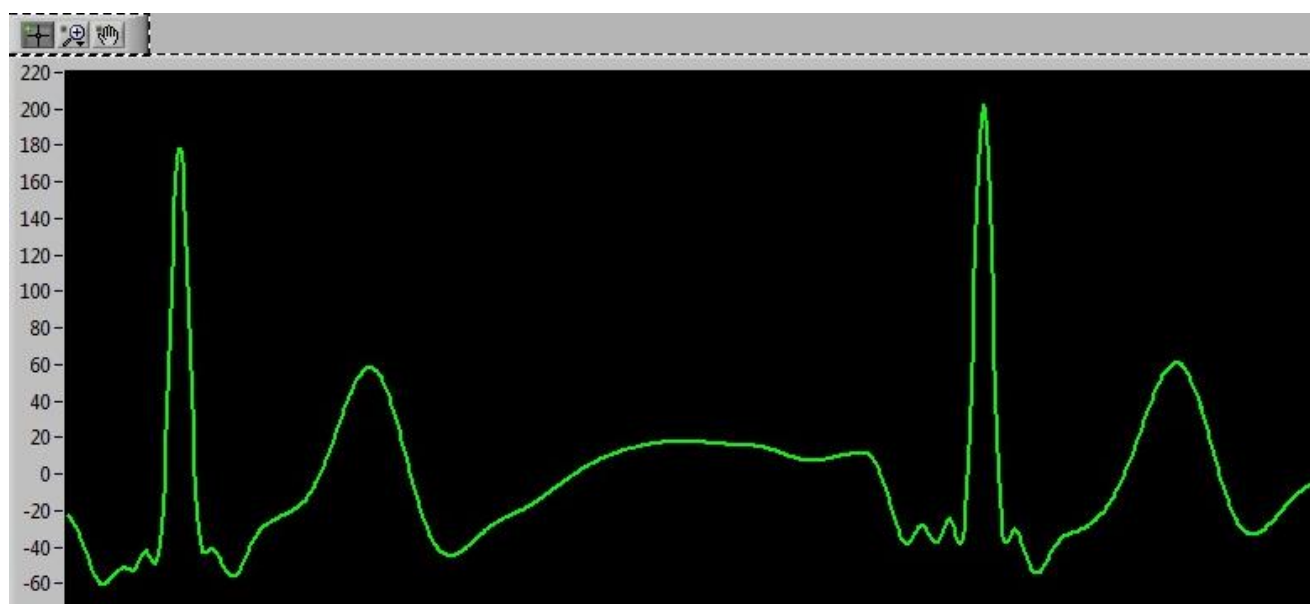


Рисунок 3.29 – Перевірка роботи ЕКГ

Аналізуючи отриманий графік, на ньому видно ізоелектричну лінію, сегменти, інтервали та зубці, які цілком характеризують певну достовірність дослідження, отже можна прийти до висновку про функціональну справність даного модулю.

Наступний кроком йде, тестування роботи тонометру, основні показники, які повинні бути відображені це: показники тиску в манжеті, які мають зростати до пікового значення та поступово спадати, фіксуючи значення діастолічного та систолічного тиску, а також пульсу, результат тестування відображено на рисунку 3.30:

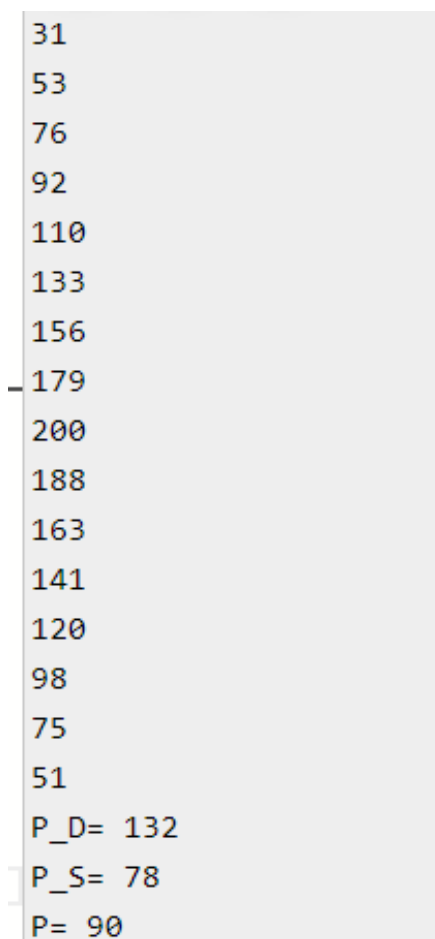


Рисунок 3.30 – Монітор порту при запуску тонометру

Відображені результати дозволяють дійти до висновку про справність роботи даного модуля.

Проведемо тестування роботи вебзастосунку на прикладі додавання пацієнта, введемо тестові дані, які зображено на рисунку 3.31:

Рисунок 3.31 – Тестові дані для додавання пацієнта

Відповідні тестові дані передаються на веб-сервер та заносяться у вигляді запису до таблиці БД «Пацієнт», що відображено на рисунку 3.32:

| id | name | doctor | gender | db | address | mac |
|-------|----------------|--------|--------|------------|---------------------------------------|-------------------|
| 32119 | Тест Тест Тест | 1188 | m | 18.06.1992 | м. Тест, вулиця тестова, будинок тест | 24-B6-FD-32-41-98 |

Рисунок 3.32 – Запис текстових даних до БД

Так, як тестові дані записані до БД, на стартовій сторінці має з'явитись відповідний запис про пацієнта, натиснувши який, можна побачити наступне, рисунок 3.33:

Рисунок 3.33 – Відображення даних нового пацієнта в вебзастосунку

Остаточним кроком в процесі тестування автоматизованої системи є тестування повноцінної функціональності, з відображенням вимірюваних значень в БД і як наслідок, на сторінці вебзастосунку(рисунок 3.34).

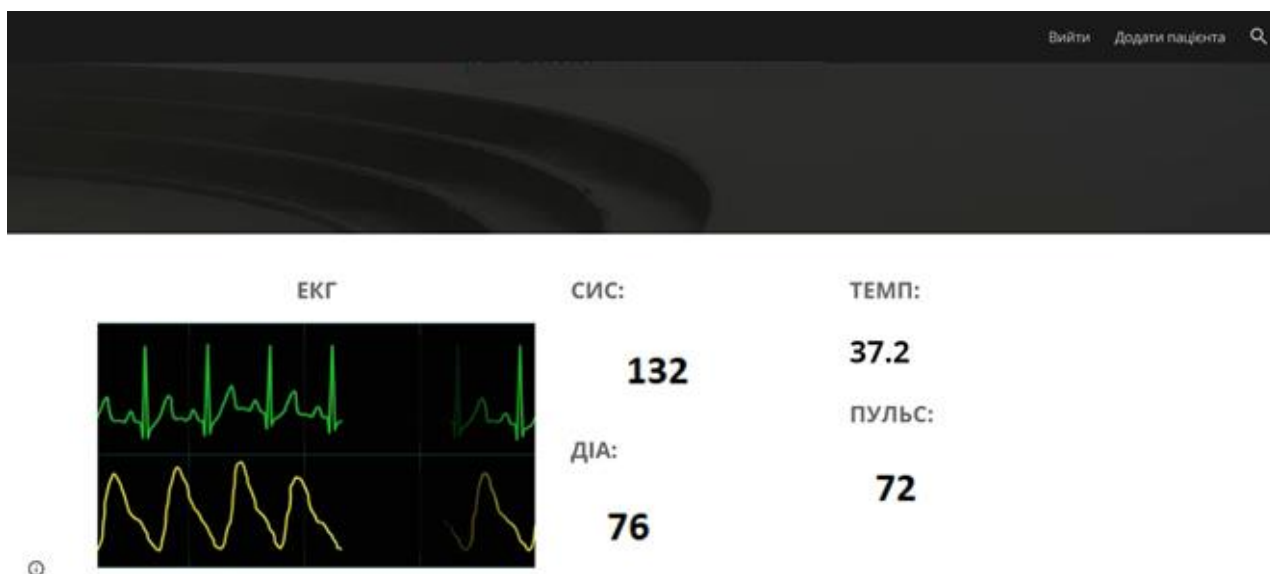


Рисунок 3.34 – Відображення вимірюваних значень в вебзастосунку

3.8 Висновки до розділу

1. У даному розділі МНР було апаратно і програмно реалізовано та описано процес створення автоматизованої системи параметричного контролю стану пацієнта за групою показників.

2. Було ретельно описано фізичну інфраструктуру системи, включаючи апаратне забезпечення, його конфігурацію та взаємозв'язки між компонентами, наведено схеми підключення та відповідний програмний код для їх впровадження.

3. У розділі було описано абстрактну модель системи у вигляді математичної моделі, яка базується на математичних принципах і формулах, що в подальшому полегшило процес написання програмного коду, а також оптимізації роботи системи. Було представлено алгоритм роботи у вигляді блок-схеми, в якому визначено логіку та порядок виконання операцій. Ретельно описано процес програмування інструкцій для керування МК і отримання даних з датчиків, що включає в себе програмування функцій, обробку даних та комунікацію з іншими компонентами системи. Наведено опис процесу

налаштування серверної інфраструктури та реалізації бездротової передачі даних на смартфон, а також процес реалізації вебзастосунку, який використовується для взаємодії з системою через веб-браузер. Реалізація вебзастосунку забезпечує зручний та інтуїтивно зрозумілий інтерфейс для користувачів системи. Проведено тестування системи, в ході якого було виявлено та усунено помилки і дефекти в системі, а також переконано в правильності її роботи та підтверджено відповідним вимогам та очікуванням до розробки.

4 Спеціальний розділ

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

до наукової роботи

на тему: «**Автоматизована система параметричного контролю стану
пацієнта за групою показників**»

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»

151 – МНР – 671. 21930201

Студент _____ Соровецький А.М.

Консультант _____ Григор'єва Л.І.

Миколаїв – 2023

Зміст

| | |
|--|-----|
| ВСТУП..... | 98 |
| 1 ОСНОВНІ ВИМОГИ ДО РОБОЧОГО МІСЦЯ ОПЕРАТОРА ПК. РОЗРАХУНОК ЗАХОДІВ НОРМАЛІЗАЦІЇ МІКРОКЛІМАТУ Й ТЕПЛОЗАХИСТУ ОРГАНІЗМУ ЛЮДИНИ..... | 99 |
| 2 АЛГОРИТМ ДІЙ В ЗАКЛАДІ ОХОРОНИ ЗДОРОВ'Я ПРИ ВИНИКНЕННІ НАДЗВИЧАЙНОЇ СИТУАЦІЇ..... | 105 |
| ВИСНОВКИ..... | 108 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 109 |

ВСТУП

Сучасний розвиток технічного та технологічного стану виробництва передбачає постійну автоматизацію та оптимізацію виробничих процесів. Реальне виробництво характеризується наявністю технічних або інших шкідливих, небезпечних виробничих факторів. Поліпшення стану охорони праці на виробництві завжди вимагає значних матеріальних витрат та запровадження на практиці результатів науково-дослідних робіт у галузі охорони праці.

Для вирішення питань гігієни, виробничої санітарії, техніки безпеки та пожежної безпеки існує охорона праці, яка регламентується з боку держави Законом України про охорону праці [1].

Важливу роль в процесі автоматизації робочих місць та виробництва в цілому відіграють оператори – автоматизатори, програмісти, серверні та сервісні фахівці. Від їх загального самопочуття залежить безперервність процесу виробництва, швидкість усунення недоліків та помилок в роботі. Значною мірою на стан робітників відіграє мікроклімат на робочому місці.

Робочим місцем техніка – програміста є кабінет, в якому зосереджено декілька робочих станцій, що складаються з персонального комп'ютера, монітора, принтера та/або серверного обладнання. Всі ці пристрої в деякій мірі створюють негативний вплив на працівника, тому важливо розрахувати всі фактори впливу, аби звести їх пагубність до мінімуму.

В кваліфікаційній роботі проводиться оптимальний розрахунок мікрокліматичних умов співробітника техніка – програміста, їх оцінка, тощо.

1 ОСНОВНІ ВИМОГИ ДО РОБОЧОГО МІСЦЯ ОПЕРАТОРА ПК. РОЗРАХУНОК ЗАХОДІВ НОРМАЛІЗАЦІЇ МІКРОКЛІМАТУ Й ТЕПЛОЗАХИСТУ ОРГАНІЗМУ ЛЮДИНИ

Умови праці – це сукупність елементів виробничого середовища, які впливають на функціональний стан людини.

Сприятливими вважаються такі умови праці, коли сукупність елементів, що їх утворюють, впливає на людину і сприяє духовому і фізичному розвитку особистості, формування творчого ставлення до праці, почуття задоволеності ним, високої працездатності.[2]

Несприятливі умови праці спричиняють передчасну втому, яка знижує можливості людини і може призвести до хворобливого стану.

Найважливішими елементами умов праці є освітленість, стан повітряного середовища (температура, вологість, рухливість і чистота повітря), шум, вібрація.[3]

Поліпшення освітленості робочого місця ґрунтується на таких рекомендаціях:

- В усіх випадках перевагу слід надати природному освітленню;
- При загальному штучному освітленні бажано використовувати люмінесцентні лампи нейтрального білого світла;
- Місцеве освітлення повинно поліпшувати умови робочої зони. Цього можна досягти зручним розміщенням джерела світла і сприятливим розподілом яскравості в полі зору. Місцеве освітлення не повинно створювати різку межу із загальним освітленням (не слід працювати при настільній лампі в дуже темній кімнаті).
- Фарбувати стіни кабінету у відносно світлі тони.
- Світло не повинно осліплювати, рефлектувати.

Стан повітряного середовища має забезпечувати профілактику переохолодження і перегрівання, необхідну вологість і швидкість руху повітря.

Основні вимоги до робочого місця оператора ПК.

- Робочі місця операторів за дисплеями слід розміщувати в спеціально відведеному приміщенні, яке відповідає гігієнічним вимогам щодо площі, умов природного освітлення та вентиляції.

- Для роботи за дисплеєм більше підходить приміщення з північною, північно-східною або північно-західною орієнтацією вікон.

- Площа приміщення повинна відповідати вимогам санітарних норм: 4,5 кв.м на одну людину. Об'єм виробничого приміщення на одного оператора повинен бути не меншим 15 куб.м.

- Робоче місце складається з стола з розміщеним на ньому екраном, клавіатурою і підставкою під документи, крісла, підставки для ніг.

- Розміри стола залежать від розмірів екрану, орієнтовні розміри: довжина 160 см, ширина 90 см, загальна площа 1,44 кв.м. Висота стола повинна регулюватись у відповідності з антропометричними даними людини в межах 68-84 см.

- Доцільне розміщення клавіатури окремо від екрана. Це забезпечує вибір оптимального положення, висоти та нахилу всіх складових обладнання робочого місця.

- Екран повинен знаходитись нижче рівня очей прямо, або з нахилом на оператора. Кут зору, при якому забезпечується оптимальне розміщення символів на екрані в межах 0,5. Екран повинен розміщуватись на відстані 40-90 см від очей оператора. Оптимальна відстань при висоті символів 2,5 мм - 50 см; при висоті символів 3-4 мм її можна збільшити до 80 см.

- Документи повинні читатись оператором з підставки, висота і нахил якої регулюються. Нахил документа 600 дозволяє максимально зменшити відблиск від нього.

- Робоче крісло повинно бути рухоме. Короткі підлокітники крісла повинні забезпечувати положення рук трохи вище стола.
- Підставка для ніг рекомендується розмірами 40x30x15 см з кутом нахилу 30, без переміщення по підлозі.
- Освітленість робочих місць повинна бути в межах від 300 до 500 як в зоні розміщення документів і клавіатури.
- В робочій зоні відношення яскравості поверхонь не повинно перевищувати 3:1, а між робочою поверхнею столу та навколишніми поверхнями (стіл, обладнання і т.п.) 10:1.
- Для уникнення засвіток екранів і зниження перепадів яскравості в полі зору при природному освітленні, робочі столи операторів необхідно розміщувати далі від вікон і таким чином, щоб віконні шиби знаходились збоку від операторів, а природне світло падало з лівої сторони. Вікна повинні бути оснащені шторами, які розсіюють світло, або регульованими жалюзі. При південній орієнтації вікон найбільш ефективним засобом для створення сприятливого світлового середовища в приміщенні і зниженні теплової дії ізоляції є використання на вікнах плівки з металізованим покриттям.
- Рекомендується робота на клавіатурі дисплея сліпим десяти пальцевим методом. Це дозволяє знизити втомлюваність зору за рахунок його постійного переключання з документа на екран і клавіатуру та шкідливу дію блищання клавіатури.

Розрахунок мікрокліматичних умов робочого місця.

Розрахунок мікрокліматичних умов робочого місця техніка – програміста слід почати з температурних параметрів, оскільки влітку не повинно бути спекотно, а взимку холодно. Для цього слід розглянути питання вибору кліматичного обладнання.

Вибір «спліт» - кондиціонера здійснюють за потужністю (охолодження) з урахуванням усіх теплоприпливів – зовнішнього, від обладнання та робітників. Зробити орієнтовний розрахунок потрібної потужності (Q_x) «спліт»-кондиціонера по формулі:

$$Q_x = Q_z + Q_o + Q_p \quad (1.1)$$

де Q_z – зовнішній приплив тепла, орієнтовно

$$Q_z = q_0 * V \quad (1.1.1)$$

для вікон південної орієнтації – $q_0 = 40$ Вт/м³, для північної – $q_0 = 30$ Вт/м³, середнє значення $q_0 = 35$ Вт/м³ (обирається в залежності від азимуту світлових прорізів, рахуючи, що північ – це проміжок від 0° до 45°, та від 315° до 360°, південь – від 135° до 225°, а все інше – середнє значення);

V – об'єм приміщення, м³:

$$V = a \times b \times h$$

Q_o – виділення тепла від обладнання, Вт

$$Q_o = 0,3P + n_k Q_{ok} \quad (1.1.2)$$

$Q_o = 0,3 * P$ - для електричних приладів, де P – паспортна потужність, Вт;

n_k – кількість одиниць оргтехніки;

$Q_{ok} = 300$ Вт орієнтовно для персонального комп'ютера та копіювального пристрою;

Q_p – виділення тепла від робітників в залежності від витрат енергії (1 ккал/год = 1,167 Вт).

$$Q_p = n_p Q_{op} \quad (1.1.3)$$

n_p - кількість працівників;

Q_{op} – енерговитрати організму, Вт.

Для виконання задачі з розрахунку системи обігріву приміщення необхідно визначити: внутрішню температуру повітря в приміщенні в холодний період року залежно від категорії роботи (в даному випадку 22°C); розрахункову зовнішню

температуру повітря для даного кліматичного району; орієнтовні втрати тепла будинком; тепловиділення від: людей, електрообладнання, нагрітих поверхонь та ін.; кількість тепла на опалення приміщень; поверхню нагрівальних приладів; загальну кількість секцій; годинні витрати води (повітря) на опалення; необхідну поверхню нагріву, тип та ККД котла.[4]

Кількість тепла, що втрачається будівельною конструкцією Q_k , залежить від різниці температур, величини їх значень, площі та виду матеріалу і може бути підрахована для плоских поверхонь за формулою:

$$Q_k = k \cdot F_k (t_{\text{вн}} - t_{\text{зовн}}), (\text{ккал/год.}) \quad (1.2.1)$$

де k — коефіцієнт теплопередачі конструкції огорожі (стін), що залежить від матеріалу з якого побудовані стіни;

F_k — поверхня огорожувальної конструкції, через яку втрачається тепло м²;

$$F_k = a \cdot h \quad (1.2.1.1)$$

$t_{\text{вн}}$ — нормована температура повітря в приміщенні, °С;

$t_{\text{зовн}}$ — розрахункова температура зовнішнього повітря, яка приймається за кліматичними даними для даного міста, °С (для Миколаєва $t_{\text{зовн}} = -16^\circ\text{C}$)

Поверхню нагріву нагрівальних приладів, що віддає тепло, визначають в еквівалентних квадратних метрах (е. к. м.), а потім перераховують на метраж прийнятих для установки типів приладів. Визначаємо відносну витрату води на е. к. м, яка буде складати:

$$q = \frac{7.98(\Delta t - 10)}{\Delta T_{\text{прил}} \cdot L}, \text{ ккал/год.} \quad (1.2.2)$$

де Δt — різниця температур між середньою температурою теплоносія в нагрівальному приладі та температурою в приміщенні, °С;

$$\Delta t = \frac{t_{\text{поч}} + t_{\text{кін}}}{2} - t_{\text{вн}} \quad (1.2.2.1)$$

перепад температур теплоносія в нагрівальному приладі, °С.

$$\Delta T_{\text{прил}} = t_{\text{поч}} + t_{\text{кін}} \quad (1.2.2.2)$$

вода з початковою температурою $t_{\text{поч}} = +100 \text{ }^\circ\text{C}$ і кінцевою $t_{\text{кін}} = +60 \text{ }^\circ\text{C}$

L — кількість води, що подається зверху донизу, $\text{кг}/\text{м}^2 \cdot \text{год}$. $L = 17,4 \text{ кг}/\text{м}^2 \cdot \text{год}$.

Значення $e. k. m.$ можна порахувати за формулою:

$$q_{e.k.m} = 7,98(\Delta t - 10) \cdot \alpha, (\text{ккал}/\text{год. e. k. m.}), \quad (1.2.3)$$

де α — поправочний коефіцієнт, що залежить від відносної витрати води ($\alpha = 0,92$).

Необхідну поверхню приладів $e. k. m.$ $F_{\text{пр}}$ можна визначити за формулою:

$$F_{\text{пр}} = \frac{Q_k}{q_{e.k.m}}, \text{ м}^2 \quad (1.2.4)$$

Необхідна кількість секцій радіаторів М-140 ($f_{e.k.m} = 0.31 \text{ м}^2$) дорівнює:

$$n_{\text{пр}} = \frac{F_{\text{пр}}}{f_{e.k.m}}, \text{ штук.} \quad (1.2.5)$$

Зведемо всі підрахунки до таблиці 1:

Таблиця 1

| Показник | Формула № | Значення |
|--------------------------|-----------|------------------------|
| Q_z | 1.1.1 | 3780 Вт |
| Q_o | 1.1.2 | 695Вт |
| Q_p | 1.1.3 | 159 Вт |
| Q_x | 1.1.4 | 4634Вт |
| Q_k | 1.2.1 | 630 ккал/год |
| F_k | 1.2.1.1 | 18 м ² |
| q | 1.2.2 | 0.55 ккал/год |
| Δt | 1.2.2.1 | 58 ^o C |
| $\Delta T_{\text{прил}}$ | 1.2.2.2 | 40 ^o C |
| $q_{e.k.m}$ | 1.2.3 | 352 ккал/год. e. k. m. |
| $F_{\text{пр}}$ | 1.2.4 | 1.8 м ² |
| $n_{\text{пр}}$ | 1.2.5 | 6 штук |

2 АЛГОРИТМ ДІЙ В ЗАКЛАДІ ОХОРОНИ ЗДОРОВ'Я ПРИ ВИНИКНЕННІ НАДЗВИЧАЙНОЇ СИТУАЦІЇ

Залежно від існуючої або прогнозованої обстановки з питань цивільного захисту в установі може бути встановлено один з трьох режимів функціонування міської субланки територіальної підсистеми єдиної державної системи цивільного захисту:

- режим повсякденного функціонування;
- режим підвищеної готовності;
- режим надзвичайної ситуації.

Режими встановлюються за рішенням виконавчого комітету або міського голови.

В районі розміщення установи чи організації можуть виникнути наступні надзвичайні ситуації:

- надзвичайні ситуації природного характеру: руйнування споруд в наслідок землетрусу, зсувів, обвалів, сильний дощ, снігопад, дуже сильний мороз, спека; катастрофічне затоплення території;
- надзвичайні ситуації техногенного характеру: вибух в будівлі, пожежа, розлив хімічної небезпечної речовини;
- надзвичайні ситуації воєнного характеру: загроза авіаційного або ракетного удару, хімічного або радіаційного зараження в наслідок бойових дій, застосування супротивником звичайних видів зброї.

При виникненні надзвичайної ситуації оповіщення адміністрації, робітників та службовців щодо надзвичайних ситуацій проводиться за заздалегідь розробленою схемою. Крім того при отриманні інформації про надзвичайну ситуацію черговий диспетчер міськвиконкому вмикає електросирени міської системи оповіщення. Протяжне звучання сирен в продовж 20-30 секунд означає подачу сигналу «Повітряна тривога», переривчасте звучання сирен більше 4-х

циклів означає подачу сигналу «Хімічна (радіаційна) загроза». Переривчасте звучання сирен менше 4-х циклів означає подачу сигналу «Відбій тривоги».[5]

Порядок дій персоналу при отриманні сигналу «Повітряна тривога»:

1. До початку роботи закладу, необхідно:

1.1 розробити та довести до відома працівників локальну інструкцію про порядок дій працівників на випадок «повітряної тривоги» та надзвичайної ситуації через воєнні дії.

1.2 розробити маршрут евакуації до найближчого укриття як для персоналу, так і для відвідувачів. Довести маршрут до відома працівників, та розмістити інформацію для відвідувачів про найближче укриття.

1.3 призначити відповідальних осіб за оповіщення, евакуацію, збереження майна та інші передбачені інструкцією дії, а також порядок і графік чергування на випадок відмови або неможливості сторонніх осіб та відвідувачів полишити заклад.

2. Дії під час «Повітряної тривоги»:

2.1 Забезпечити негайне оповіщення персоналу та відвідувачів про повітряну тривогу, інші надзвичайні ситуації. Не забувайте про осіб, що мають особливі потреби: нечують, поганочують, мають проблеми із зором, не бачать, пересуваються на візках.

2.2 За наявності небезпеки (включення електросирен, вибухи, пожежа внаслідок обстрілу):

- організувати негайну евакуацію персоналу та відвідувачів затвердженими шляхами евакуації, повідомити відвідувачів про найближче укриття, яке знаходиться (адреса):
- повідомити про надзвичайну ситуацію відповідні служби (ДСНС, поліція, швидка допомога), керівника;
- організувати зустріч нарядів рятувальних служб;
- надати домедичну допомогу постраждалим.

2.3 У разі припинення роботи та переміщення працівників в укриття або у випадку евакуації, відповідальним працівникам забезпечити виконання розпоряджень військової (місцевої) адміністрації, органів ДСНС, локальної інструкції, зокрема, відключення систем живлення збереження матеріальних цінностей.

2.4 У випадку відмови відвідувачів полишати приміщення та направлятись в укриття або неможливості евакуації, забезпечити чергування персоналу.[6]

Усі працівники установ повинні вжити необхідних заходів щодо зберігання матеріальних цінностей при загрозі або виникненні надзвичайних ситуацій.

Заходи із захисту від надзвичайних ситуацій або з ліквідації їхніх наслідків повинні враховувати необхідність попередження або зменшення можливих збитків установам.

Відповідальність за організацію охорони майна установ під час захисту від надзвичайних ситуацій або ліквідації їхніх наслідків покладається на відповідальні особи.[7]

Особливості дій працівників при загрозі хімічного зараження:

1. Вентиляційні установки та кондиціонери терміново виключаються, закриваються вікна, двері, кватирки, приміщення герметизуються. Вихід із будівлі й вхід до неї припиняється до особливого розпорядження адміністрації.

2. Працівникам видаються засоби індивідуального захисту, одночасно вживаються заходи із забезпечення відвідувачів ватно-марлевими пов'язками.

3. При виявленні у приміщенні, де укриваються працівники, небезпечної хімічної речовини працівники повинні вийти або з дозволу адміністрації залишити зону зараження. Виходити із зони необхідно тільки у засобах індивідуального захисту та рухатися в напрямку, перпендикулярному напрямку вітру.

ВИСНОВКИ

Охорона праці в сучасному світі є дуже важливим параметром вибору робочого місця. Людство усвідомлює та розуміє недоліки своїх професій, можливі в майбутньому професійні хвороби, тощо. Аби не допустити пагубних впливів уряди багатьох країн світу впровадили закони, кодекси, статті про охорону праці. Не винятком є й Україна.

Зважаючи на нормативні акти про охорону праці України було змодельовано та прораховано необхідні заходи нормалізації мікроклімату й теплозахисту організму людини.

Мікрокліматичні умови середовища перебування працівника дуже важливі і повинні бути виконаними. Згідно наведених вище розрахунків та результатів необхідне кліматичне обладнання для одного працівника в офісі розмірами 6х6х3 метрів при наявному персональному комп'ютеру та принтеру складає:

- для підтримання оптимальної температури в літній сезон необхідний «спліт» - кондиціонер потужністю 3780 Вт;
- для підтримання оптимальної температури під час зимового сезону необхідно 6 секцій радіаторів М-140.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Основи охорони праці: Підручник. 2-ге видання / К.Н.Ткачук, М.О.Халімовський, В.В.Зацарний та ін. – К.: Основа, 2006 – 448 с. 2.
2. Практикум із охорони праці: Посібник / В.Ц.Жидецький, В.С.Джигирей, В.М. Сторожук та ін. – Львів: Афіша, 2000 – 352 с.
3. Про охорону праці : Закон України від 14 жовтня 1992 р. № 49. URL: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення: 07.05. 2021).
4. Про затвердження Правил пожежної безпеки: Наказ від 30 грудня 2014 р. № 1417. URL: <https://zakon.rada.gov.ua/laws/show/z0328-19> (дата звернення: 08.05.2021).
5. Правила охорони праці під час експлуатації електронно – обчислювальних машин : веб-сайт. URL: https://dnaop.com/html/245_3.html (дата звернення: 08.05.2021).
6. Охорона праці в офісі. Вимоги до робочого місця працівника : веб-сайт. URL: <https://gc.ua/uk/oxorona-praci-v-ofisi-vimogi-do-robochogo-miscya-ofisnogo-pracivnika> (дата звернення: 08.05.2021).
7. Охорона праці. Загальні вимоги : веб-сайт. URL: https://studwood.ru/1676935/informatika/ohorona_pratsi (дата звернення: 08.05.2021).

ВИСНОВКИ

1. У магістерській науковій роботі розроблено автоматизовану систему параметричного контролю стану пацієнта за групою показників перелік, яких визначається за типом хвороби та регламентується стандартами МОЗ України.
2. Планування проектних завдань і розраховані терміни виконання кожного завдання, забезпечило вчасне виконання проекту. Проведена установка та налаштування обраного програмного забезпечення, а також тестування працездатності функцій, підтвердило правильність реалізації автоматизованої системи віддаленого спостереження і контролю стану як бездротову та кросплатформну.
3. Одночасне застосування у проєкті принципу бездротовості та кросплатформності будови автоматизованої системи дозволить співробітникам закладів охорони здоров'я отримувати актуальну інформацію про стан пацієнта в режимі реального часу, а також переглядати динаміку показників. В свою чергу пацієнт отримує можливість проводити курс лікування або реабілітації поза закладом охорони здоров'я перебуваючи під доглядом за допомогою пристроїв вимірювання показників, що отримано в лікарні на термін лікування або придбано, апаратної частини, яку визначено у проєкті, вебзастосунку, який розроблено та Андроїдного смартфона.
4. Автоматизована система, яка розроблена, забезпечує віддалений контроль стану пацієнта за умов застосування математичних моделей для підконтрольної організації роботи приладів виміру параметрів показників за принципами ефективної будови моделей, оптимальної алгоритмізації систем автоматизованого керування.
5. Вирішені в процесі виконання роботи поставлені завдання та проведені тестування роботи автоматизованої системи дозволяють рекомендувати архітектуру автоматизованої системи та запропоновані апаратні засоби з

відповідними схемами підключення та комунікації і вебзастосунок, які забезпечуватимуть повноцінне функціонування.

6. Тестування програмного рішення по модульне та автоматизованої системи вцілому, що застосувало врахувало результати досліджень роботи з використанням побудованої математичної моделі та алгоритму роботи довело її функціональні можливості та підтвердило необхідність реалізації системи за принципом мультиплатформного вебзастосунок.

7. Розроблена автоматизована система параметричного контролю стану пацієнта за групою показників, що регламентовано МОЗ України і дозволяє модернізувати існуючий підхід та впроваджувати віддалений контроль стану, а також вирішує проблеми існуючих інформаційних систем.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Robert S., Istepanian H., Woodward B. MHealth: Fundamentals and Applications. Los Angeles: ISBN, 2016. 424 p.
2. Robert S., Istepanian H. 4G Health- The Long Term Evolution of m-health. London: MINT, 2012. 133 p.
3. Рейтинг мобільних застосунків в розділі mHealth. URL: <https://pubmed.ncbi.nlm.nih.gov/34914718> (дата звернення: 28.04.2023)
4. Звіт компанії GSMA. URL: <https://gsma.com/report/wp-content/uploads/mhealth-project> (дата звернення: 28.04.2023)
5. mHealth Solutions Market Trends & Growth Drivers | MarketsandMarkets. URL: <https://www.marketsandmarkets.com/Market-Reports/mhealth-apps-and-solutions-market-1232.html> (дата звернення: 28.04.2023)
6. MHealth Solutions Market Trends. URL: <https://www.marketsandmarkets.com/Market-Reports/mhealth-apps.html> (дата звернення: 29.04.2023)
7. Інноваційні технології в сфері охорони здоров'я. URL: <http://www.arteka.ua/article/326752> (дата звернення: 29.04.2023)
8. Звіт аналітичної компанії Deloitte : вебсайт. URL: <https://www.deloitte.com/content/dam/assets-shared/legacy/docs/perspectives/2022/gx-lshc-hc-outlook-2019.pdf> (дата звернення: 30.04.2023)
9. Трунов А., Беглиця В., Грищенко Г., Зюзін В., Кошовий В. Методи та засоби формування загальних показників для автоматизації апаратів реабілітаційної медицини для постінсультних хворих. *Східно-Європейський журнал підприємницьких технологій*, 2021. 46 с.
10. Композитна оцінка та аналіз варіабельності серцевого ритму: пат. 2021225745 США: US16/867,629; заявл. 09.04.2021; опубл. 11.11.2021, Бюл. № 16. 12 с.

11. Моніторинг. URL: [https://uk.wikipedia.org/wiki/ Моніторинг_\(медицина\)](https://uk.wikipedia.org/wiki/Моніторинг_(медицина))
(дата звернення: 30.04.2023)
12. Медичні проекти Arduino. URL: <https://medevel.com/15-arduino-medical-projects/> (дата звернення: 30.04.2023)
13. Національна державна служба охорони здоров'я. URL: https://www.ngsmedicare.com/NGS_LandingPage/ (дата звернення: 30.04.2023)
14. Їстівна електроніка. URL: <https://techxplore.com/news/2023-03-edible-electronics-seaweed-skin-health.html> (дата звернення: 30.04.2023)
15. Барало О. В. Автоматизація технологічних процесів і систем автоматичного керування: навч. посіб.: у 3 кн. Київ : Аграрна освіта, 2010. Кн. 2. 227 с.
16. Raspberry Pi. URL: <https://www.raspberrypi.org/about/> (дата звернення: 30.04.2023)
17. STM32. URL: https://www.st.com/content/st_com/en/about/st_company_information/who-we-are.html (дата звернення: 30.04.2023)
19. Arduino. URL: <https://www.arduino.cc/en/about> (дата звернення: 30.04.2023)
20. Клімушин П. С. Інформаційні системи та технології в економіці: навч. посіб. Харків: ХаРІ НАДУ «Магістр», 2011. 448 с
21. Якобсон А., Буч Г., Рамбо Д. Уніфікований процес розробки програмного забезпечення: пер. з англ. – Фоліо: Харків, 2002. 496 с.
22. Брукс Ф. Міфічний людино-місяць, або як створюються програмні системи. Символ–Плюс: Харків, 2006. 304 с.
23. Ахметов І.Г., Юний вчений, міжнародний науковий журнал №9: Пер. з англ. Символ–Плюс: Харків, 2018. 124 с.
24. Берко А. Ю. Системи баз даних та знань: навч. посіб. Львів : Магнолія, 2008. 456 с.

25. Литвин В.В. Методи та засоби інженерії даних та знань. Львів: Магнолія-2006, 2012. 241 с.
26. Моделювання бізнес процесів та архітектури ПЗ: вебсайт. URL: http://dspace.tneu.edu.ua/retrieve/17733/FCIT_kKN_mIPZPZS_dMBP_LEC.pdf (дата звернення 01.06.2021).
27. Фреймворк Laravel: вебсайт. URL: <https://laravel.su/docs/> (дата звернення 02.05.2021).
28. WI-FI : вебсайт. URL: <https://www.wi-fi.org/> (дата звернення 02.05.2021).
29. Bluetooth: вебсайт. URL: <https://www.bluetooth.com/> (дата звернення 02.05.2021).
30. Ігнатенко А.А., Основи програмування мовою JavaScript: тексти лекцій. Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002. 83 с.
31. NPX Semiconductors : вебсайт. URL: <https://www.nxp.com/docs/en/datasheet/MPX2050.pdf> (дата звернення 19.05.2023).
32. TMP36 Datasheet: вебсайт. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf (дата звернення 19.05.2023).
33. Алгоритм: вебсайт. URL: <https://uk.wikipedia.org/wiki/Алгоритм> (дата звернення 19.05.2023).
34. Хендерсон Кен. Delphi 3 и системи клієнт/сервер: керівництво розробника. Пер. с англ. - К.: Діалектика, 1997. 736 с.

ДОДАТОК А

Коди програм

A1 Код програми тонометра

```
#include <SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
#include <Relay.h>
#include <Wire.h>

const float NIVEL_INFERIOR = 8061.0;
const float NIVEL_SUPERIOR = 49603.0;

const byte RELE_NIVEL_INFERIOR = 8;
const byte RELE_NIVEL_SUPERIOR = 9;
const unsigned long TIEMPO_MPXPIN = 2500UL;
const unsigned long TIEMPO_PRESENTACION = 5000UL;
float nivel;
float litros;
float centimetros;
int ESTADOS;
const byte mpxpin = A0;
unsigned long tiempo_mpxpin = 0;
unsigned long tiempo_presentacion = 0;
enum estados_1 {
    ESTADO_MPXPIN,
    ESTADO_LITROS,
    ESTADO_CENTIMETROS,
    ESTADO_NIVEL_INFERIOR,
    ESTADO_NIVEL_SUPERIOR
} estado = mpxpin;

void setup() {
    Wire.begin();
    lcd.begin(20, 4);
    lcd.init();
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(0, 0);
    Serial.begin(9600);
    lcd.print("D. Marmo Servicios");
    lcd.setCursor(0, 2);
    lcd.print(" cisterna 50 000 ");

    tiempo_presentacion = millis();

    pinMode(A0, INPUT);
    pinMode(RELE_NIVEL_INFERIOR, OUTPUT);
```

```

pinMode(RELE_NIVEL_SUPERIOR, OUTPUT);
digitalWrite (RELE_NIVEL_INFERIOR, HIGH);
digitalWrite (RELE_NIVEL_SUPERIOR, HIGH);
tiempompxpin = millis();
delay (10);
}

void loop() {

    if (millis() - tiempompxpin > 2500UL) {
        nivel = analogRead (mpxpin);
        tiempompxpin = millis();
        float mv = (nivel / 1023.0) * 5000.0;
        float centimetros = mv / 10.960;
        float litros (centimetros * 116.1);

        Serial.print("litros ");
        Serial.println(litros);
        Serial.print("   ctm: ");
        Serial.println(centimetros);
        Serial.print("   mv: ");
        Serial.println(mv);
        lcd.setCursor(0, 0);
        lcd.print("litros   ");
        lcd.print(litros);
        lcd.setCursor(0, 2);
        lcd.print("centimetros ");
        lcd.print(centimetros);

        //if (NIVEL_INFERIOR) {
        if ( litros < NIVEL_INFERIOR ) {
            digitalWrite (RELE_NIVEL_INFERIOR, LOW);
        } else {

            digitalWrite (RELE_NIVEL_INFERIOR, HIGH);
        }
        //if (NIVEL_SUPERIOR) {
        if (litros > NIVEL_SUPERIOR ) {

            digitalWrite (RELE_NIVEL_SUPERIOR, LOW);
        } else {

            digitalWrite (RELE_NIVEL_SUPERIOR, HIGH);
        }
    }
}

```

A2 Код програми термометра

```
#define sensorPin A0
void setup()
{
  Serial.begin(9600);
  analogReference(INTERNAL);
}

void loop()
{
  int reading = analogRead(sensorPin);
  float voltage = (reading * 1.1) / 1024.0;
  float temperatureC = (voltage - 0.5) * 100 ;
  Serial.print(voltage);
  Serial.println(" volts");
  Serial.print(temperatureC);
  Serial.println(" degrees C");
  delay(1000);
}
```

A3 Лістинг частини коду вебзастосунку

Лістинг коду файлу explore-container.component.ts

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';
import { IonicModule } from '@ionic/angular';
import { DoctorPage } from './doctor.page';

describe('DoctorPage', () => {
  let component: DoctorPage;
  let fixture: ComponentFixture<DoctorPage>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ DoctorPage ],
      imports: [IonicModule.forRoot()]
    }).compileComponents();
```

```

    fixture = TestBed.createComponent(DoctorPage);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });

```

Лістинг коду файлу modal-patient-edit.page.ts

```

import {Component, Input, OnInit} from '@angular/core';
import {ModalController} from '@ionic/angular';
import {ApiService} from '../services/api.service';

@Component({
  selector: 'app-modal-patient-edit',
  templateUrl: './modal-patient-edit.page.html',
  styleUrls: ['./modal-patient-edit.page.scss'],
})
export class ModalPatientEditPage implements OnInit {
  @Input() PatientName: string;
  @Input() patientId: number;

  constructor(private api: ApiService, private modalCtrl: ModalController) {}

  ngOnInit() {
  }

  back(type) {
    this.modalCtrl.dismiss({
      type
    });
  }

```

```
}
```

```
save() {
  this.api.editPatient(this.patientId, this.patientName).subscribe(
    data => this.back('save')
  );
}
```

Лістинг коду файлу modal-patients-edit.page.ts

```
import {Component, Input, OnInit} from '@angular/core';
import {ApiService} from '../services/api.service';
import {ModalController} from '@ionic/angular';

@Component({
  selector: 'app-modal-patients-edit',
  templateUrl: './modal-patients-edit.page.html',
  styleUrls: ['./modal-patients-edit.page.scss'],
})
export class ModalPatientsEditPage implements OnInit {

  @Input() patientsName: string;
  @Input() patientsId: number;
  @Input() patientId: number;
  @Input() journalId: string;

  patterns: any[];
  curPattern: any;
  baseUrl: string;

  constructor(private api: ApiService, private modalCtrl: ModalController) {
    this.baseUrl = api.getBaseUrl();
  }
}
```

```
this.api.getPatterns().subscribe(  
  data => {  
    this.patterns = data as any[];  
    this.refreshCurPattern();  
  }  
); }  
  
refreshCurPattern() {  
  this.curPattern = this.patterns.find(elem => elem.id === this.journalId);  
}  
  
ngOnInit() {  
  console.log(this.journalId);  
}  
  
back(type) {  
  this.modalCtrl.dismiss({  
    type  
  });  
}  
  
save() {  
  this.api.editPatients(this.patientsId, this.patientId, this.patientsName, this.journalId).subscribe(  
    data => this.back('save')  
  );  
}  
}
```


ДОДАТОК Б

Публікація за темою роботи

Тези доповіді на XVII Міжнародна наукова конференція «Ольвійський форум – 2023: стратегії країн Причорноморського регіону в геополітичному просторі»

ДОДАТОК В

Результат перевірки на плагіат