**ЗАСТОСУНОК ДЛЯ ВИМІРЮВАННЯ ВІДНОСНОГО РІВНЯ СО У ВИДИХУ ЛЮДИНИ З ОТРИМАННЯМ ДАНИХ ЧЕРЕЗ WEB-ІНТЕРФЕЙС**

ЗМІСТ

|  |  |
| --- | --- |
| ВСТУП | 3 |
| РОЗДІЛ 1 ОГЛЯД ЛІТЕРАТУРИ | 5 |
| * 1. Актуальність діагностики зовнішнього дихання. | 5 |
| РОЗДІЛ 2 СТВОРЕННЯ СКЛАДОВИХ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖ | 7 |
| 2.1. Програмно-апаратна частина комплексу. | 7 |
| 2.2. Локальна мережа. | 9 |
| 2.2.1. FTP протокол. | 13 |
| 2.2.2. FTP проти HTTP. | 14 |
| 2.2.3 Створення локальної мережі. | 14 |
| 2.2.4 FTP-клієнт. | 15 |
| РОЗДІЛ 3 ПРОГРАМНИЙ КОД ДЛЯ ВЕБ-ЗАСТОСУНКУ | 18 |
| 3.1. Frontend частина для відображення даних. | 18 |
| 3.1.1. Середовище розробки та HTML код для додатку. | 19 |
| 3.1.2. CSS, JS для стилізації. | 19 |
|  |  |
|  |  |

**Вступ**

В сучасному світі дуже важливо слідкувати за станом здоров`я людини для високої працездатності та збільшення тривалості життя. І на допомогу нам в побутовому житті приходить багато різноманітних приладів для моніторингу показників життєдіяльності людини. Наприклад пульсометри, тонометри, глюкометри та різні аналізатори газів в організмі людини [3].

Аналіз повітря дихання пацієнта відноситься до неінвазійних методів діагностики, що й викликає підвищений інтерес. З розвитком інформаційних технологій стало легше створювати нові апаратні засоби та методики визначення процентного вмісту різних газів. Один з способів діагностики є аналіз складу видихуваного повітря, яке представляє собою суміш газів різного ендогенного походження з дихальних шляхів [1]. Компоненти видихуваного повітря можуть свідчити про стан здоров’я людини. Для того щоб якось можна було фіксувати та аналізувати маркери певних газів, зокрема, монооксиду вуглецю, створені різноманітні пристрої для діагностики зовнішнього дихання. Загалом, розробка апаратури і методик для віддаленого контролю стану здоров’я людини на основі мікропроцесорної техніки сприяє підвищенню ефективності діагностики і є актуальним напрямком у медицині [1].

Наприклад, у випадках хронічної обструкції легень чи ревматоїдного артриту, куріння хворого є фактором, що впливає на результат проведених аналізів [1]. Тому перед взяттям проби хворих просять утриматися від куріння на термін 8 годин. Але, як показала статистика, частина пацієнтів ігнорує попередження, що впливає на результати. Отже важливо мати прилади, що можуть встановити, чи палили пацієнти перед проведенням процедури.

Ще один приклад використання датчиків для виявлення чадного газу у диханні був представлений в роботі [2]. У ній автор пропонує використовувати сенсори для вимірювання рівня CO у мобільних смартфонах та персональних комп’ютерах. А саме вводити їх в конструкцію гаджетів. Важливою особливістю цих «втручань» є об'єктивна оцінка стану курця, через частий контроль вмісту чадного газу в диханні. Необхідна об’єктивна оцінка куріння, оскільки курці часто помилково класифікують себе як некурців під час спроб кинути палити. Крім того, деякі дані свідчать про те, що лише моніторинг CO в диханні може сприяти зменшенню куріння [2].

Необхідно розробити портативний апаратно-програмний комплекс, який буде мати досить хорошу чутливість для аналізу видиху людини і міг би показати коректні данні в реальному часі з підключенням до інтернету для аналізу рівня чадного газу у видиху людини.

Враховуючи все вищезазначене при виконанні даного курсового проекту переді мною було поставлено завдання розробити веб-застосунок для відображення та аналізу даних, отриманих від апаратної частини комплексу для вимірювання відносного рівня монооксиду вуглецю у видиху людини.

**РОЗДІЛ 1**

**РОЗДІЛ 1 ОГЛЯД ЛІТЕРАТУРИ**

1.1 Актуальність діагностики зовнішнього дихання.

Оперативний контроль стану здоров’я та працездатності організму людини за складом видиху повітря полягає, в основному, у вимірюванні відносної концентрації СО2, О2 і СО. Більш інформативним є одночасні виміри й інших характеристик організму людини, таких як частота дихання, частота серцевих скорочень, вологість та температура повітря, що видихається. В роботі [1] наведено методику аналізу повітря видиху на наявність маркерних газів і наведено шляхи апаратного вирішення проблеми. Саме моніторинг концентрації СО2, О2 і СО, визначення об’ємів цих газів, виділених при видиху, їх співвідношення, має виконуватись в портативному інтелектуальному приладі на основі мікропроцесорної техніки з використанням електрохімічних та інших сенсорів [1].

Основною функцією мого приладу буде вимірювання відносного рівня монооксиду вуглецю у видиху людини. СО, або чадний газ, є одним з найбільш токсичних компонентів продуктів горіння. Чадний газ входить до складу диму і виділяється при тлінні та горінні майже всіх горючих речовин. У невеликих кількостях він навіть утворюється в організмі людини. Підступність чадного газу проявляється у тому, що він легко проходить через бар’єр легень, потрапляючи у кров, легко вступає в контакт з білком гемоглобіном. Найгірше те, що моноокис вуглецю набагато швидше та сильніше може зв’язуватися з гемоглобіном у порівнянні з киснем, витісняючи його і утворюючи досить стійку сполуку — карбоксигемоглобін. Кров при цьому втрачає здатність переносити і правильно використовувати кисень, що пошкоджує мозок та інші органи. В результаті спричиняється кисневе голодування організму в цілому, що при тривалому вдиханні великої концентрації чадного газу призводить до летального наслідку [1].

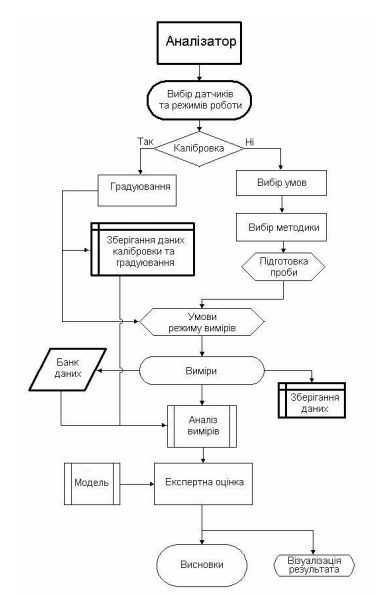


Рис. 1.1. Структурна схема загального алгоритму роботи газоаналізатора [1].

Загальний алгоритм роботи газоаналізатора можна розбити на декілька основних етапів, як це показано на рис. 1.1:

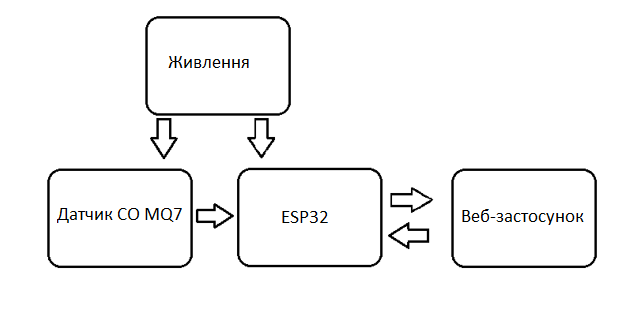
* вибір датчиків та їх калібрування, враховуючи умови режиму вимірів;
* виміри та зберігання даних;
* аналіз та візуалізація результатів через мережу.

**РОЗДІЛ 2**

**СТВОРЕННЯ СКЛАДОВИХ ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖ**

**2.1** Програмно-апаратна частина комплексу.

На рис. 2.1. наведено структурну схему пристрою, що розробляється.

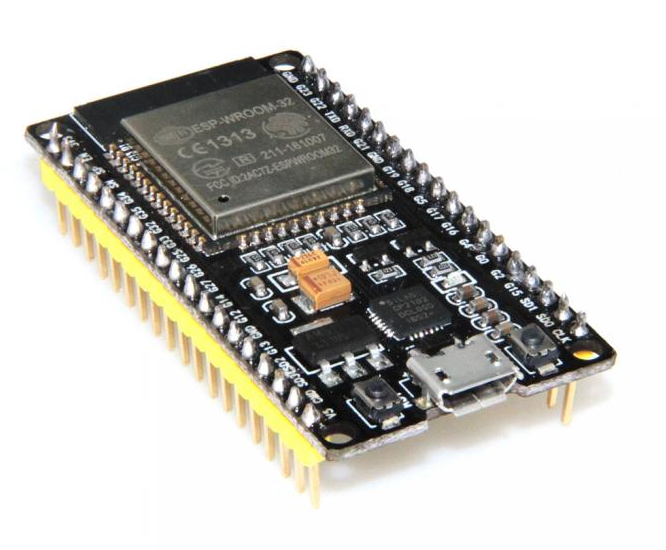
****

**Рис 2.1. Структурна схема програмно-апаратного комплексу.**

Для створення газоаналізатора буде використано мікроконтролер ESP-32 побудований на мікромодулі **ESP-WROOM-32** новому мініатюрному високопродуктивному чіпі, що дозволить знімати показання з датчика MQ7 [3] для детектування рівня чадного газу та передавати дані у веб-застосунок через стек протоколів стандартів WI-FI 802.11n [4]. Основні характеристики ESP-32 наведені в табл. 1.

Таблиця 1.

|  |  |
| --- | --- |
| Напруга живлення | 5В |
| Wi-Fi Стандарти | FCC/CE/IC/TELEC/KCC/SRRC/NCC |
| Мережеві протоколи: | TCP/UDP/HTTP/FTP/MQTT |
| Частотний діапазон | ГГц 2.4 ~ 2.5 |
| Робочий струм, середній | 80 мА |
| Робочий струм піковий | 500 мА |
| USB-UART конвертер | CP2102 |
| Діапазон робочих температур | -40°C ~ +85°C |



**Рис. 2.2. Зовнішній вигляд м**ікроконтролера ESP-32.

MQ7 – нормований датчик для детектування рівня чадного газу, також має досить непогану чутливість до природного газу, побутового газу і різних зріджених попутних нафтових газів (пропан, пропілен, бутан та інші). Основним робочим елементом датчика є нагрівальний елемент, за рахунок якого відбувається хімічна реакція, в результаті якої на аналоговий вихід надходить сигнал, пропорційний до концентрації газу. Основні характеристики датчика MQ7 наведені у табл. 2.

Таблиця 2.

|  |  |
| --- | --- |
| Напруга живлення | 5 В |
| Струм споживання | 160 мА |
| Час розжарення нагрівального елементу | 60-90 с |
| Потужність нагрівача | 350 мВт |
| Температурний діапазон | -10 - 50 °C |
| Діапазон чутливості | 10 – 10000 ppm |

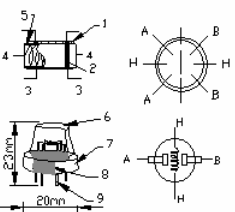


Рис. 2.3. Схематична будова датчика СО MQ7 [3].

2.2 Локальна мережа.

Локальні мережі – це об’єднання різних електронних пристроїв в радіусі до одного кілометра, які часто використовуються в межах кімнати, будинків та невеликих територій [6].

Еталона модель OSI (The Open Systems Interconnection mode), яка лягла в основу ARPANET (Advanced Research Projects Agency Network) – це 7-ми рівнева концепція стеку мережевих протоколів, які підлягають глобальним стандартам та мають чіткі правила побудови кожного рівня [6].

Фізичний рівень – це найнижчий рівень моделі, призначений безпосередньо для передачі потоку інформації. Здійснює передавання сигналів через різні види кабелів і відповідно їхнє приймання і перетворення на біти даних відповідно до [методів кодування цифрових сигналів](https://uk.wikipedia.org/wiki/%D0%92%D0%B8%D1%8F%D0%B2%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F_%D1%96_%D0%B2%D0%B8%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F_%D0%BF%D0%BE%D0%BC%D0%B8%D0%BB%D0%BE%D0%BA). Іншими словами здійснює інтерфейс зв’язку між пристроєм та носієм. На цьому етапі ми маємо справи з різними ретрансляторами, концентраторами. Цей рівень приймає кадр даних від канального рівня, кодує його в послідовність сигналів, які потім передаються у лінію зв'язку. В наш час використовуються 3 основних середовища передавання: мідний кабель (copper), оптичне волокно (fiber) та бездротове середовище передавання (wireless). Для мідного кабелю сигнали, що представляють біти даних є електричними імпульсами, для оптичного волокна — імпульсами світла. У випадку використання бездротових з'єднань сигнали є радіохвилями (електромагнітними хвилями). Основними функціями фізичного рівня є: фізичні компоненти, кодування даних та їх передавання.

Канальний рівень – головною метою якого є забезпечення взаємодії мереж на фізичному рівні й контролю за помилками, які можуть виникнути. Отримані з фізичного рівня дані він упаковує в кадри, перевіряє на цілісність, якщо потрібно — виправляє помилки й відправляє на мережний рівень. На цьому рівні працюють [комутатори](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%80%D0%B5%D0%B6%D0%B5%D0%B2%D0%B8%D0%B9_%D0%BA%D0%BE%D0%BC%D1%83%D1%82%D0%B0%D1%82%D0%BE%D1%80), [мости](https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D1%81%D1%82_(%D1%82%D0%B5%D0%BB%D0%B5%D0%BA%D0%BE%D0%BC%D1%83%D0%BD%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%97)) й [мережеві адаптери](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%80%D0%B5%D0%B6%D0%B5%D0%B2%D0%B0_%D0%BF%D0%BB%D0%B0%D1%82%D0%B0).

Мережевий рівень - 3-й рівень мережної моделі OSI, призначений для визначення шляху передавання даних. Відповідає за трансляцію логічних адрес. Визначає найкоротший маршрут, комутацію й маршрутизацію пакетів. Відстежує затори у мережі. На цьому рівні працює такий мережний пристрій, як [маршрутизатор](https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%88%D1%80%D1%83%D1%82%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80).

Транспортний рівень — 4-й рівень моделі OSI, призначений для доставлення даних без помилок, втрат і дублювання в тій послідовності, у якій вони були передані. При цьому немає значення, які дані передаються, звідки й куди, тобто він визначає сам механізм передачі. Блоки даних він розділяє на фрагменти, розмір яких залежить від протоколу, короткі об'єднує в один, довгі розбиває. Протоколи цього рівня призначені для взаємодії типу точка-точка.

Сеансовий рівень - відповідає за підтримку сеансу зв'язку, дозволяючи програмам взаємодіяти між собою тривалий час. Рівень керує створенням та завершенням сеансу, обміном інформацією, синхронізацією даних.

Рівень представлення - цей рівень відповідає за перетворення протоколів і кодування та декодування даних. Запити програм, отримані з прикладного рівня, він перетворює у формат для передачі по мережі, а отримані з мережі дані перетворює у формат, зрозумілий для застосунків. На цьому рівні здійснюється стиснення та розпакування даних, а також адресація запитів мережевих ресурсів.

Прикладний рівень - верхній 7-й рівень моделі, забезпечує взаємодію мережі та користувача. Рівень надає доступ користувачеві через  [прикладі програми](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA) до мережних служб, таких як обробник запитів до [баз даних](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85), доступ до [файлів](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB), пересилання [електронної пошти](https://uk.wikipedia.org/wiki/%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0_%D0%BF%D0%BE%D1%88%D1%82%D0%B0). Також відповідає за передачу службової інформації, надає програмам інформацію про помилки й формує запити до рівня представлення. Структурна схема еталонної моделі OSI наведена на рис. 2.4.

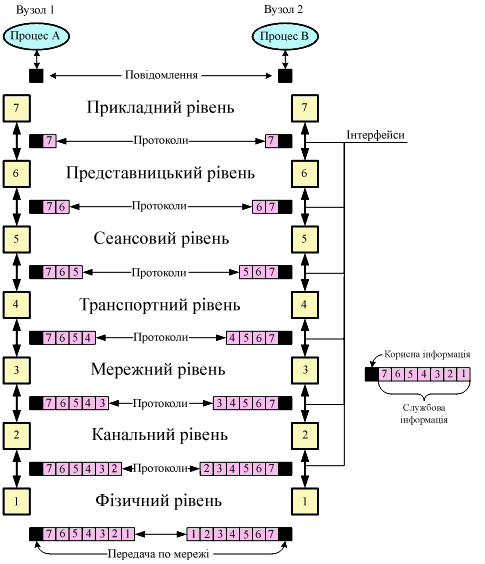


Рис. 2.4. Структурна схема еталонної моделі OSI [6].

Еталонна модель OSI характерезує фундаментальну базу побудови комп’ютерних мереж, але у неї є деякі недоліки у порівняні з моделлю TCP/IP, яка використовується сьогодні.

TCP/IP описує спосіб передачі даних від джерела до отримувача. Головним плюсом цієї моделі є більш покращені протоколи передачі інформації та взаємодія на всіх рівнях, яких є всього чотири. Канальний, мережевий, транспортний та прикладний рівень. Вони представлюять собою повноціні та самостійні рівні мережевої моделі, забезпечують коректну роботу служб та інтерфейсів, які у моделі OSI були на порядок гіршим через несвоєчасне створення,невдалі технології та саму реалізацію. Тому для створення локальної мережі буде використаний FTP протокол, який працює на прикладному рівні TCP/IP.

2.2.1. FTP протокол.

FTP – це протокол передачі файлів по мережі [7]. Він потрібен для того щоб простіше організувати правильну роботу веб-застосунку. Структурна схема роботи FTP-протоколу зображена на рис. 2.5. Протокол побудований на архітектурі «клієнт - сервер» і використовує мережеве з’єднання для передачі команд та даних між клієнтом і сервером. Досить яскрава особливість протоколу FTP полягає в тому, що він використовує множинне підключення. При цьому один канал є керуючим, через який надходять команди до серверу і повертають його відповіді, зазвичай через TCP-порт 21, а через решту передаються дані. Протокол FTP, як і HTTP, має двійковий режим передачі, що скорочує накладні витрати трафіку та зменшує час обміну даними під час передачі великих файлів [5].

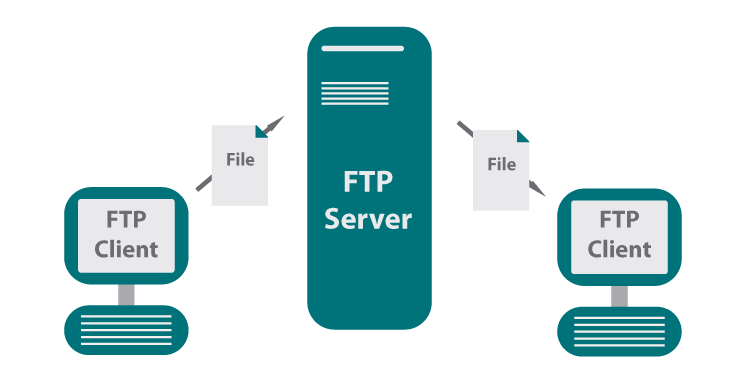


Рис. 2.5. Структурна схема роботи FTP-протоколу [8].

2.2.2. FTP проти HTTP.

HTTP – протокол передачі даних, що використовується в комп’ютерних мережах для передачі гіпертекстових документів. Він, так само як і FTP, належить до прикладного рівня. Основним призначенням протоколу є передача веб-сторінок, хоча за допомогою нього успішно передаються інші файли.

Проте, використовуючи FTP протокол можна досягнути кращого з’єднання за допомогою вбудованої автентифікації, режиму передачі як двійкової так і текстової інформації. При використання протоколу FTP є можливість підтримувати операції над файловою системою та головною перевагою є те, що FTP вміє перетворювати данні на льоту, що дуже важливо при роботі з різними системними архітектурами. Також дуже добре, що FTP надає широкий діапазон клієнтів, який спрошує створення локальної мережі.

2.2.3 Створення локальної мережі.

Кожний веб-застосунок під час розробки поділяється на дві основні частини: Frontend та Backend. Backend – програмно-апаратна частина сервісу, яка відповідає за функціонування внутрішньої логіки. Для представлення своїх функцій backend реалізує API (Application Programming Interface) інтерфейс, який використовує frontend. Таким способом фронтенд-розробнику не потрібно знати особливості реалізації серверу.

Для запуску серверу в локальній мережі, потрібно використати WI-FI модуль апаратної частини. Схема локальної мережі, що буде створена за допомогою мікроконтролера ESP32 наведена на рис. 2.6.



Рис. 2.6. Схема локальної мережі, що буде створена за допомогою ESP32.

Плата мікроконтролера ESP-32 працюватиме як точка доступу, до якої буде підключатись будь-який пристрій з доступом до інтернету та інформація яка знімається з датчиків та поступає в пам'ять контролера, буде відразу передаватись на веб-сторінку. Для створення точки доступу в базовому коді мікроконтролера потрібно прописати її параметри та правила взаємодії з FTP-клієнтом.

#include "WiFi.h"

//Підключення бібліотеки

**const** **char**\* ssid = "yourNetworkName";

**const** **char**\* password = "yourNetworkPassword";

//Змінні для збереження характеристик точки доступу

ESP8266WebServer **HTTP**(**80**);

FtpServer ftpSrv;

WiFi.softAP(ssid);

SPIFFS.begin();

HTTP.begin();

ftpSrv.begin(ssid,password );

// Старт точки доступу

String **getContentType**(String filename){

**if** (filename.endsWith(".html")) **return** "text/html";

**else** **if** (filename.endsWith(".css")) **return** "text/css";

**else** **if** (filename.endsWith(".js")) **return** "application/javascript";

**else** **if** (filename.endsWith(".png")) **return** "image/png";

**return** "text/plain";

}

//Обробка файлів для передачі через клієнт

2.2.4 FTP-клієнт.

FTP-клієнт – це комп’ютерна програма для спрощеня доступу до FTP сереверу. Для роботи з клієнтом потрібно включити контролер та прописати усі параметри в програмі: ім’я точки доступу, пароль та порт. Після підключення в програмі FTP-клієнту перемістити всі файли в локальну папку контролера, як це показано на рис. 2.7.

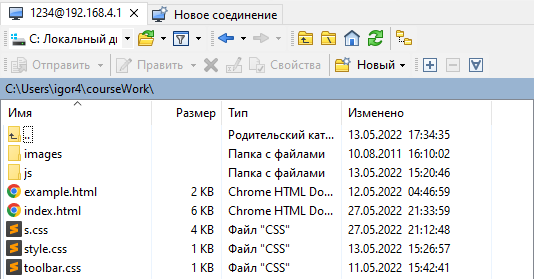


Рис. 2.7. Скріншот екрану програми FTP-клієнту.

Налаштувавши серверну частину та передачу даних необхідно реалізувати інтерфейс веб-застосунку. Схема передачі даних через FTP клієнт від датчика MQ7 до застосунку для візуалізації та аналізу зображена на рис. 2.8.

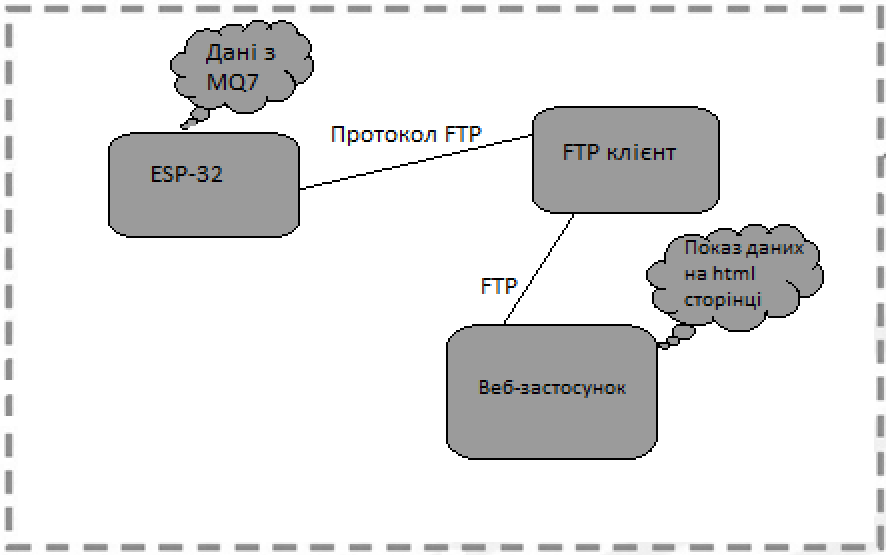


Рис. 2.8. Схема передачі даних від датчика СО до застосунку через FTP клієнт.

**РОЗДІЛ 3**

**ПРОГРАМНИЙ КОД ДЛЯ ВЕБ-ЗАСТОСУНКУ.**

3.1. Frontend частина веб-застосунку для відображення даних.

Frontend – частина програми, яка надає інтерфейс для взаємодії між користувачем і серверною частиною. В  будь-якій програмній архітектурі може бути багато рівнів між апаратним забезпеченням та кінцевим користувачем. Кожен з цих рівнів може мати як frontend так і backend. Front частина представляє собою абстракцію, спрощення базового компоненту через надання користувачу зручного інтерфейсу.

3.1.1. Середовище розробки та HTML код для додатку.

Для створення проекту було використано простий редактор коду VS Code, де потрібно створити усі файли формату html, css та js для передачі через ftp-клієнт.

HTML – мова програмування для розмітки веб сторінок. Кожна сторінка починається з <!DOCTYPE html> для розпізнавання браузером (див. рис. 3.1), та складається з основних тегів <html>,<head>,<body>,<div>. Тег <html> - основа самої сторінки, який означає початок та її кінець. Тег <head> - це місце, де підключається основні стилі, js-файли та бібліотеки. Тег <body> - початок та кінець html коду. Тег <div> з якого складається всі блоки, надписи та форми на сторінці та за допомогою якого, можна стилізувати окремі частини веб-застосунку через різноманітні атрибути.

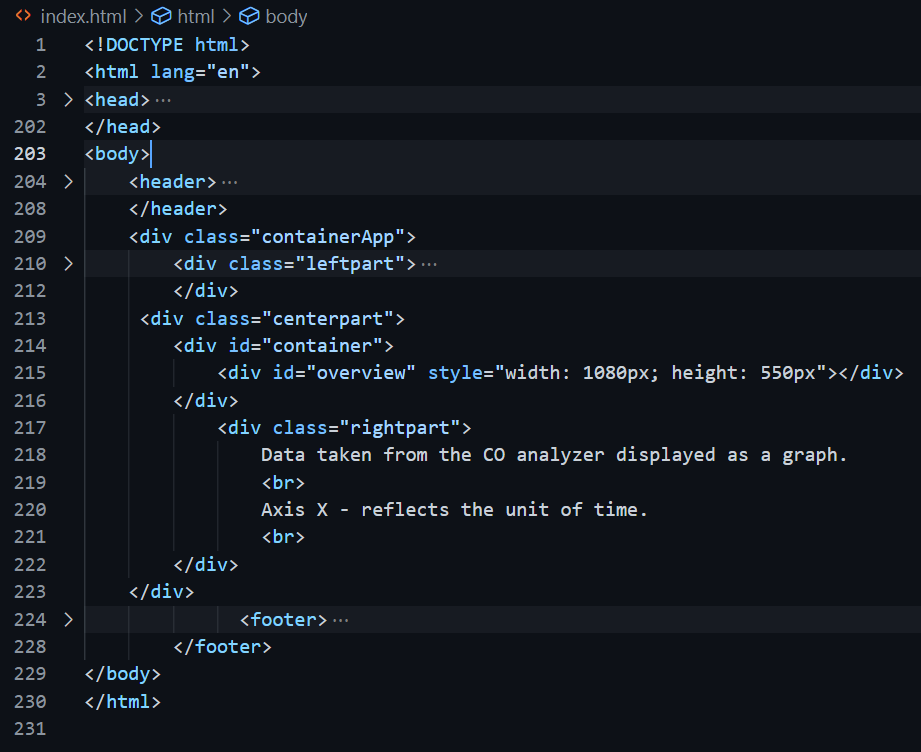


Рис. 3.1. Скріншот екрану з кодом на HTML.

3.1.2. CSS, JS для стилізації веб-застосунку.

CSS – таблиця каскадних стилів, яка покликана вирішувати складні завдання для створення гарної картинки на веб сторінках. JavaScript – це повноцінна мова програмування з усіма можливими парадигмами, яка дозволяє вирішувати багато завдань, пов’язаних з веб розробкою. JavaScript також вмонтований в усі сучасні браузери, що дозволяє чітко налаштовувати зв'язок між усіма інструментами, які потрібні для frontend розробки.

Після підключення до головного html всіх файлів CSS описується правила побудови у вигляді - селектор { властивість : значення}, як це видно із рис. 3.2.



Рис. 3.2. Скріншот екрану з кодом на CSS.

Та за допомогою javaScript реалізується основна логіка побудови графіку для візуалізації наших даних, які приходять з серверної частини у нашу локальну змінну datappm (див. рис. 3.3).

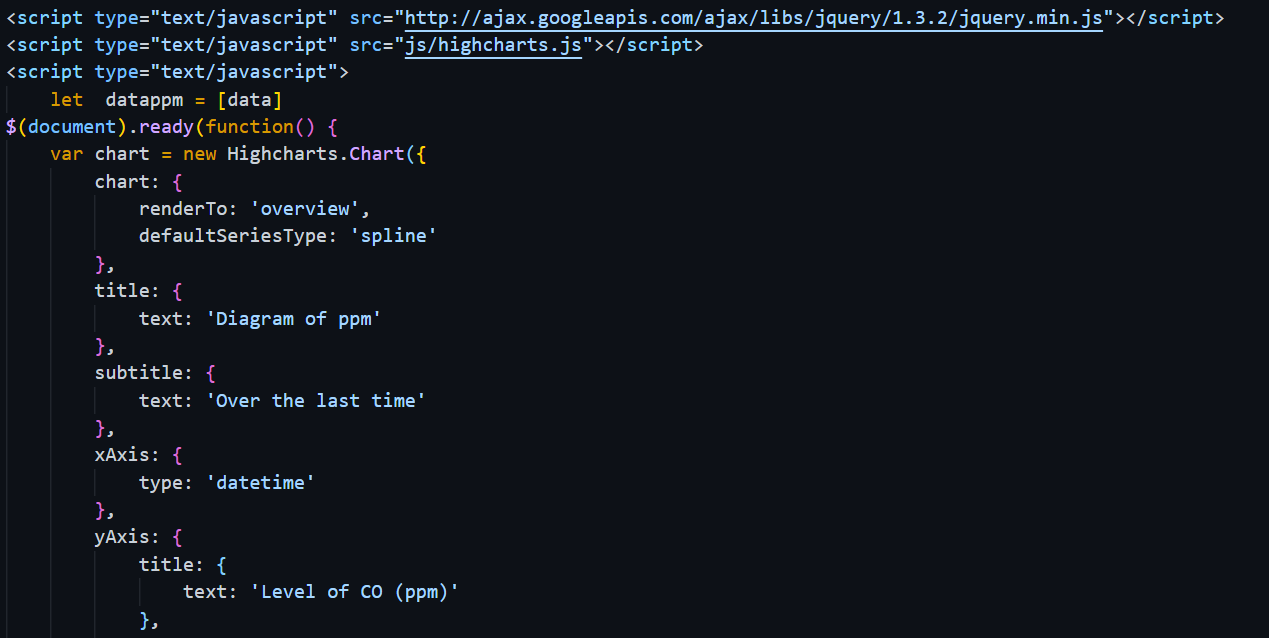


Рис. 3.3. Скріншот екрану з кодом на JS [9].

Після реалізації усього коду будується графік, де вісь ординат буде відповідати рівню СО, а вісь абсцис відповідатиме часу спостереження. Приклад такого графіку наведено на рис. 3.4.

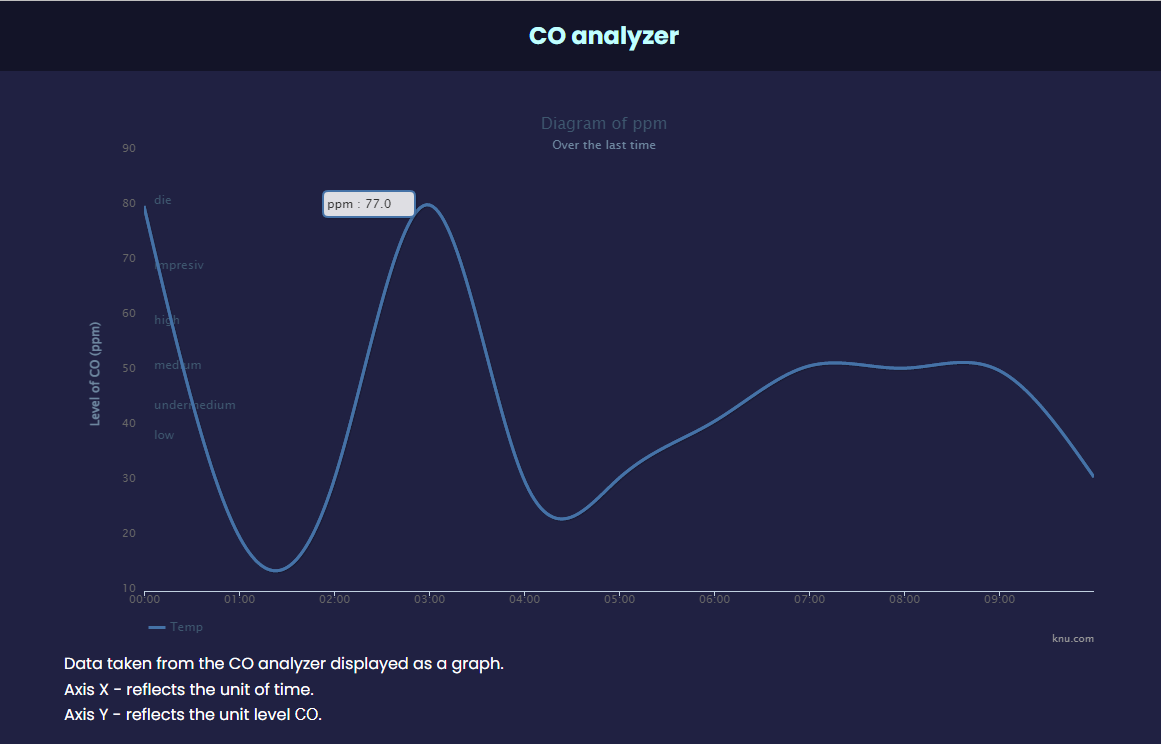


Рис. 3.4. Скріншот веб-застосунку із графіком залежності рівня СО від часу спостереження.