

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий інститут телекомунікаційних систем

Кафедра телекомунікацій

До захисту допущено:

Завідувач кафедри

_____ Сергій КРАВЧУК

«__» _____ 2025 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія та програмування
інфокомунікацій»**

спеціальності 172 «Телекомунікації та радіотехніка»

**на тему: «Безпроводова система моніторингу та аналізу технічних
параметрів автомобіля»**

Виконав:

студент IV курсу, групи ТЗ-12

Боровик Іван Ігорович _____

Керівник:

Доцент кафедри ТК НН ІТС, к.т.н., доцент

Живков Олександр Петрович _____

Рецензент:

Доцент кафедри ІТТ НН ІТС, к.т.н., доцент,

Новогрудська Ріна Леонідівна _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій КРАВЧУК

«__» _____ 2025 р.

ЗАВДАННЯ
на дипломну роботу студенту
Боровику Івану Ігоровичу

1. Тема роботи «Безпроводова система моніторингу та аналізу технічних параметрів автомобіля», керівник роботи Живков Олександр Петрович, к.т.н., доцент, затверджені наказом по університету від «26» травня 2025 р. № 1755-с.
2. Термін подання студентом роботи 9 червня 2025 р.
3. Вихідні дані до роботи: Апаратне забезпечення: мікроконтролер ESP8266, адаптер ELM327, бортова мережа автомобіля. Програмне забезпечення: середовище Arduino IDE, мова програмування C++, HTML/JavaScript, бібліотеки ESPAsyncWebServer, SoftwareSerial. Методичні матеріали: технічна документація до ESP8266 та ELM327, стандарти OBD-II (ISO 9141-2, ISO 15765-4), джерела з телеметрії у транспорті та основ бездротової передачі даних.
4. Зміст роботи: ознайомитися з апаратною архітектурою ESP8266 та протоколами взаємодії з адаптером ELM327; дослідити стандарти бортових діагностичних протоколів OBD-II; реалізувати UART-з'єднання між ELM327 та NodeMCU для отримання даних про швидкість, температуру охолоджувальної рідини та тиск масла; розгорнути локальний вебсервер без використання файлових систем (LittleFS), вбудувавши вебінтерфейс у

прошивку; забезпечити оновлення даних на вебсторінці у реальному часі протестувати систему в умовах експлуатації та провести аналіз стабільності передачі даних.

5. Перелік ілюстративного матеріалу:

1. Вступ
2. Мета та завдання дипломної роботи
3. Актуальність теми
4. Об'єкт та предмет дослідження
5. Безпроводові технології
6. Архітектура системи
7. Програмна реалізація
8. Тестування
9. Напрямки розвитку
10. Висновок

6. Дата видачі завдання 22.02.2025 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Робота над першим розділом	22.02.2025 – 09.03.2025	Виконано
2	Робота над другим розділом	10.03.2025 – 12.04.2025	Виконано
3	Робота над практичною частиною (розділ 3)	14.04.2025 – 15.05.2025	Виконано
4	Продовження практичної частини (розділ 4)	18.05.2025-01.06.2025	Виконано
5	Оформлення дипломної роботи	06.06.2025	Виконано

Студент

Іван БОРОВИК

Керівник

Олександр ЖИВКОВ

РЕФЕРАТ

Пояснювальна записка до дипломної роботи містить 74 сторінок, 15 ілюстрацій, 3 таблиці, 1 додаток, 15 бібліографічних джерел.

Метою роботи є розробка безпроводової системи моніторингу технічних параметрів автомобіля на базі мікроконтролера ESP8266 та адаптера ELM327, що зчитує дані через протокол OBD-II. Система призначена для моніторингу таких параметрів, як швидкість, температура охолоджувальної рідини та тиск масла, з подальшим відображенням результатів на веб-інтерфейсі в реальному часі.

У роботі розглянуто основні компоненти системи, зокрема, мікроконтролер ESP8266, його можливості для бездротової передачі даних через Wi-Fi та використання адаптера ELM327 для зчитування даних з діагностичних систем автомобіля. Описано принципи роботи з OBD-II та особливості взаємодії між ELM327 і ESP8266 через UART. Було розроблено веб-інтерфейс для відображення параметрів на веб-сторінці, що дозволяє в реальному часі оновлювати дані без перезавантаження сторінки.

Практична частина роботи включає розробку скетчу для ESP8266, що зчитує дані через UART і транслює їх через Wi-Fi у веб-інтерфейс. Система була протестована в реальних умовах автомобіля, що підтвердило її здатність стабільно передавати дані про технічний стан транспортного засобу на відстані до 10 метрів.

Отримані результати демонструють ефективність застосування безпроводових технологій для моніторингу автомобіля, а також високий рівень стабільності передачі даних через Wi-Fi. Така система може бути корисною для вдосконалення моніторингу технічного стану автомобілів в умовах реального часу, а також для впровадження в різні телекомунікаційні рішення

Рекомендовано використовувати розроблену систему для подальших досліджень в галузі телеметрії, а також для розробки мобільних та веб-додатків для моніторингу технічних параметрів транспортних засобів.

Ключові слова: ESP8266, ELM327, OBD-II, моніторинг, автомобіль, Wi-Fi, веб-інтерфейс, телеметрія, безпроводова передача даних, система моніторингу, реальний час, UAR

ABSTRACT

The explanatory note for the thesis consists of 74 pages, 15 illustrations, 3 tables, 1 appendix, and 15 bibliographic references.

The aim of the work is to develop a wireless system for monitoring the technical parameters of a vehicle based on the ESP8266 microcontroller and the ELM327 adapter, which reads data via the OBD-II protocol. The system is designed to monitor parameters such as speed, coolant temperature, and oil pressure, with the results displayed on a web interface in real time.

The work discusses the main components of the system, including the ESP8266 microcontroller, its capabilities for wireless data transmission via Wi-Fi, and the use of the ELM327 adapter for reading data from vehicle diagnostic systems. The principles of working with OBD-II and the specifics of the interaction between ELM327 and ESP8266 through UART are described. A web interface was developed to display parameters on a webpage, allowing data to be updated in real time without page reloads.

The practical part of the work includes the development of a sketch for ESP8266 that reads data via UART and transmits it via Wi-Fi to the web interface. The system was tested under real vehicle conditions, confirming its ability to reliably transmit data about the vehicle's technical status at a distance of up to 10 meters.

The results demonstrate the effectiveness of applying wireless technologies for vehicle monitoring, as well as the high level of data transmission stability via Wi-Fi. Such a system can be useful for improving vehicle technical condition monitoring in real-time, as well as for implementation in various telecommunication solutions.

It is recommended to use the developed system for further research in the field of telemetry, as well as for the development of mobile and web applications for monitoring vehicle technical parameters.

Keywords: ESP8266, ELM327, OBD-II, monitoring, vehicle, Wi-Fi, web interface, telemetry, wireless data transmission, monitoring system, real-time, UART

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1	6
АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ СИСТЕМ ТА ТЕХНОЛОГІЙ...	6
1.1 Огляд сучасних систем моніторингу технічних параметрів автомобіля	6
1.1.1. Безпроводові системи моніторингу	9
1.1.2 Існуючі програми для моніторингу автомобіля.....	11
1.2 Опис використання ELM327 для зчитування даних з автомобіля	13
1.2.1 Принцип роботи ELM327	14
1.2.3 Переваги зчитування даних через ELM327	19
1.3 Технології для безпроводового передавання даних	20
1.3.1 Загальні принципи передачі даних у безпроводних системах ...	21
1.3.2 Wi-Fi як основа телекомунікаційної взаємодії	21
1.3.3 Порівняння технологій Wi-Fi та Bluetooth у контексті автомобільного моніторингу.....	22
1.4 Огляд використання NodeMCU (ESP8266) в системах IoT	23
Висновок	26
РОЗДІЛ 2	27
ПРОЕКТУВАННЯ БЕЗПРОВОДОВОЇ СИСТЕМИ МОНІТОРИНГУ	27
2.1 Вибір та обґрунтування компонентів системи.	27
2.1.1 NodeMCU (ESP8266)	27
2.1.2 OBD2-адаптер на базі ELM327.....	28
2.1.3 Інтерфейс UART.....	28
2.1.4 Електроживлення	29
2.2 Схема з'єднання ELM327 з NodeMCU	29
2.3 Технічні вимоги до системи моніторингу	31

2.3.1 Функціональні вимоги	31
2.3.2 Електротехнічні та апаратні вимоги	32
2.3.3 Програмні вимоги	32
2.3.4 Комунікаційні вимоги	33
2.3.5 Надійність і безпека	33
Висновок	34
3.1 Структура веб-сторінки для моніторингу	35
3.2 Використання JavaScript для відображення даних у реальному часі	37
3.3 Розробка графічної частини сторінки	39
3.4 Підключення NodeMCU до веб-сторінки через Wi-Fi	42
3.5 Розробка програмного забезпечення NodeMCU	44
Висновок	46
4.1 Процес тестування зчитування даних з автомобіля	48
4.2. Тестування стабільності системи при передаванні даних через Wi- Fi	50
4.3 Аналіз результатів тестування та їх порівняння з реальними даними	52
Висновок	53
5.1 Досягнуті результати	55
5.3 Напрями подальших досліджень та вдосконалення системи	58

ПЕРЕЛІК СКОРОЧЕНЬ

ESP	Embedded Serial Processor
OBD-II	On-Board Diagnostics version 2
UART	Universal Asynchronous Receiver-Transmitter
HTML	HyperText Markup Language
Wi-Fi	Wireless Fidelity
HTTP	HyperText Transfer Protocol

ВСТУП

Актуальність теми дипломної роботи обумовлена постійно зростаючими вимогами до моніторингу технічного стану автомобілів у реальному часі. Сучасні автомобілі оснащуються численними датчиками, що дозволяють зчитувати різноманітні параметри, однак інтеграція цих даних для забезпечення ефективного моніторингу та діагностики потребує нових рішень у сфері бездротових технологій. Враховуючи це, розробка безпроводової системи моніторингу для оцінки основних технічних параметрів автомобіля є актуальним завданням.

Метою роботи є розробка безпроводової системи моніторингу технічних параметрів автомобіля на базі мікроконтролера ESP8266 та адаптера ELM327, що зчитує дані через протокол OBD-II. Система має забезпечити моніторинг параметрів, таких як швидкість, температура охолоджувальної рідини та тиск масла, з подальшим відображенням результатів на веб-інтерфейсі в реальному часі.

Об'єктом дослідження є система моніторингу технічних параметрів автомобіля, що складається з апаратних компонентів (мікроконтролер ESP8266, адаптер ELM327) та програмного забезпечення для зчитування та передачі даних. Предметом дослідження є використання технологій безпроводового зв'язку для створення автономної системи моніторингу автомобіля.

У роботі було виконано аналіз існуючих підходів до побудови систем моніторингу та вибір оптимальних компонентів для реалізації безпроводового рішення. Також розроблено технічне рішення для взаємодії між адаптером ELM327 і мікроконтролером ESP8266 через UART, а також створено веб-систему для відображення параметрів автомобіля в реальному часі через локальну Wi-Fi мережу. Крім того, було проведено тестування розробленої системи в умовах експлуатації транспортного засобу.

Практична значущість роботи полягає в розробці доступної та ефективної системи моніторингу, яка дозволяє відстежувати основні технічні параметри автомобіля без потреби у додаткових інфраструктурах чи підключенні до Інтернету. Рішення, представлене у цій роботі, може бути застосовано як у комерційних цілях, так і для покращення діагностики автомобілів в умовах реального часу.

Структура роботи включає п'ять основних розділів. Перший розділ присвячений огляду технологій моніторингу, у другому розглядаються компоненти та технічні вимоги до системи, у третьому — описано розробку веб-системи, четвертий розділ містить результати тестування, а п'ятий — висновки та рекомендації.

РОЗДІЛ 1

АНАЛІТИЧНИЙ ОГЛЯД ІСНУЮЧИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

1.1 Огляд сучасних систем моніторингу технічних параметрів автомобіля

Для сучасного автомобіля система моніторингу параметрів роботи автомобіля є дуже важливим елементом, так як з плином часу виробники почали здешевшувати та спрощувати конструкції своїх авто та прибирати показники критичних даних з приборних панелей, своєчасно помічена проблема може застерегти власника від більш серйозних поломок та призвести до економії та часу та грошей. Наразі усі сучасні системи функціонують на базі стандарту OBD-II який дозволяє зчитувати інформацію з CAN-ліній та K-ліній автомобіля.

У дослідженні Rimpas D., Papadakis A. та Samarakou M. розглядається застосування OBD-II для діагностики роботи транспортного засобу, зокрема щодо експлуатаційних характеристик і споживання пального. Автори наводять приклади використання телематичних рішень, де дані з OBD передаються через мобільну мережу до центру обробки, що дозволяє формувати прогностичні моделі на основі аналізу кодів несправностей. Такі моделі допомагають попереджати потенційні поломки. Інші підходи включають інтеграцію з GPS та Sensor HUB для оптимізації маршрутів, пошуку паркувальних місць або зменшення заторів, а також розробку гібридних рішень на базі традиційних автомобілів, що дозволяє зменшити витрати пального та викиди CO₂ до 22%. Загалом автори підкреслюють перспективність використання хмарних технологій для віддаленого моніторингу, реєстрації даних у реальному часі та побудови інноваційних цифрових сервісів[18].

У дослідженні Kang T. U. представлено інноваційний підхід до аналізу безпеки транспортних засобів із використанням інструменту Automated CAN Analyzer (ACA), який дозволяє здійснювати автоматизоване зворотне інженерування повідомлень, що передаються по шині CAN через інтерфейс OBD-II. У статті акцентується увага на вразливості CAN-протоколу до ін'єкцій зловмисних повідомлень, особливо в умовах підключення автомобіля до зовнішніх мереж через OTA або Bluetooth. Розроблений інструмент ACA дає змогу дешифрувати зміст CAN-повідомлень без необхідності мати доступ до конфіденційної документації виробника автомобіля, що значно спрощує аналіз навіть для недосвідчених дослідників. Крім цього, ACA підтримує функції автоматизованого тестування на проникнення та може бути використаний для перевірки ефективності IDS-систем у автомобільних мережах[16].

Одним із інших ключових напрямів розвитку сучасних систем моніторингу транспортних засобів є використання технологій Інтернету речей (IoT). Згідно з дослідженням A. R. Chougule та K. Suganthi, впровадження IoT у сфері автомобілебудування дозволяє створити «розумні» транспортні засоби, які здатні не лише відстежувати технічні параметри у режимі реального часу, але й здійснювати обмін даними між автомобілями. Розроблена ними система забезпечує комунікацію "автомобіль-автомобіль" (V2V) за допомогою сенсорів, інтегрованих у конструкцію машини, що дає змогу реалізовувати інтелектуальну взаємодію між транспортними засобами. Такі рішення підвищують рівень безпеки, дозволяють виявляти потенційні ризики та оптимізувати трафік у міських умовах[12].

Як зазначено у роботі І. В. Худякова, побудова сучасної інформаційної системи моніторингу транспортних засобів передбачає інтеграцію декількох рівнів – від апаратної частини зчитування даних до хмарної обробки інформації та інтерфейсів користувача[8]. Автори дослідження підкреслюють важливість модульного підходу до розробки систем моніторингу, де кожен компонент (сенсори, контролери, комунікаційні модулі, серверна частина та

клієнтський інтерфейс) може бути адаптований або замінений відповідно до вимог конкретного проєкту. Також зазначається, що ефективна система повинна забезпечувати не лише точне зчитування параметрів, а й безпечну передачу даних, їх фільтрацію, попередню обробку на периферії (edge computing), а вже потім – аналітику на сервері або в хмарному середовищі.

Крім того, відповідно до праці В. П. Волкова, одним із напрямів удосконалення систем моніторингу є підвищення точності оцінки програмного забезпечення, яке використовується для контролю технічних параметрів. Запропоновані підходи дозволяють мінімізувати похибки при зчитуванні та інтерпретації параметрів з OBD-II інтерфейсу, що особливо актуально в умовах експлуатації автомобілів із високим рівнем електронної насиченості [1].

Таким чином, сучасні системи моніторингу технічного стану автомобіля демонструють високий рівень інтеграції апаратних засобів, телеметричних стандартів, програмного забезпечення та інноваційних підходів до аналізу даних. Розробки в галузі OBD-II, IoT-технологій, бездротової комунікації (Wi-Fi, Bluetooth, ZigBee, LoRaWAN, NB-IoT, 5G) та автоматизованих засобів зворотного інженерування (як-от АСА) дозволяють не лише здійснювати зчитування та аналіз діагностичних даних, а й впроваджувати комплексні системи взаємодії між транспортними засобами (V2V) та віддаленого моніторингу. Впровадження сучасних технологій сприяє підвищенню ефективності експлуатації автомобілів, забезпеченню безпеки дорожнього руху за рахунок прогнозування можливих поломок і виявлення кіберзагроз, а також оптимізації виробничих процесів на сервісних підприємствах. Загалом, подальший розвиток цих комплексів сприятиме впровадженню інноваційних рішень, що поєднують автоматизацію, високопродуктивний аналіз даних та сучасні методи кіберзахисту, що є невід’ємною складовою майбутнього автомобільних технологій.

1.1.1. Безпроводові системи моніторингу

У сучасних автомобілях активно використовуються безпроводові системи моніторингу, що дозволяють зчитувати та передавати дані через Bluetooth, Wi-Fi та інші технології. Такі системи дозволяють не тільки зчитувати дані, але й передавати їх на мобільні пристрої або комп'ютери для подальшого аналізу.

Серед найбільш відомих безпроводових адаптерів для OBD2 - це Bluetooth адаптери та Wi-Fi адаптери. Вони дають змогу користувачам підключатися до бортових систем автомобіля за допомогою смартфонів або ноутбуків.

Сучасні мобільні додатки для моніторингу стану автомобіля активно використовують протокол OBD2 для підключення до автомобільних систем і зчитування технічних параметрів. Програми, такі як Torque (Android), Car Scanner (iOS) та OBDeleven (для Volkswagen), дозволяють користувачам не тільки отримувати базову інформацію про параметри, але й проводити детальну діагностику, виявляти коди помилок, а також слідкувати за ефективністю роботи різних систем автомобіля.

Ці програми зазвичай мають інтерфейс для відображення даних у вигляді графіків, таблиць і текстових повідомлень, що робить їх зручними для використання на ходу. Вони можуть також надавати рекомендації щодо необхідності обслуговування або ремонтів, спрощуючи процес контролю за станом автомобіля.

Як зазначає Н. Д. Міщук у своєму дослідженні, комп'ютеризовані системи моніторингу характеристик сигналів бездротових мереж у приміщеннях складної форми вимагають застосування адаптивних алгоритмів, що враховують просторову геометрію середовища, вплив перешкод, багатопроточність та загасання сигналу. У дисертаційній роботі автор досліджує побудову моделі розповсюдження сигналу в складних архітектурних умовах, застосування методів обробки та фільтрації даних, а

також структурно-функціональну організацію систем збору та аналізу параметрів сигналу. Особливу увагу приділено використанню Wi-Fi та ZigBee-модулів, реалізації збору інформації через модульні сенсорні вузли та створенню програмного забезпечення для візуалізації результатів моніторингу[4].

Отримані результати щодо оптимізації розташування бездротових вузлів, стабільності каналу зв'язку та енергоефективності роботи сенсорних мереж є надзвичайно релевантними і в контексті автомобільних бездротових діагностичних рішень, де схожі виклики виникають через зміну умов експлуатації, рух автомобіля та вплив електромагнітних завад.

Сучасні пристрої моніторингу використовують технології Bluetooth та Wi-Fi для передачі даних пропоную розглянути переваги та недоліки кожної:

Wi-Fi - це один із найпоширеніших стандартів для безпроводового зв'язку, який дозволяє передавати дані на великі відстані з високою швидкістю. В автомобільних системах Wi-Fi адаптери зазвичай підключаються через точку доступу (маршрутизатор), що забезпечує безперешкодну передачу інформації з автомобіля на мобільний пристрій або сервер[5].

Переваги використання Wi-Fi для моніторингу автомобіля[21]:

- Великий радіус дії.
- Швидкість передачі даних.
- Легкість інтеграції з іншими пристроями.

Недоліки Wi-Fi для моніторингу автомобіля:

- Вища витрата енергії.
- Необхідність у точці доступу.

Bluetooth - це стандарт безпроводового з'єднання на короткі відстані, що ідеально підходить для передачі даних між пристроями в межах одного автомобіля або на невеликі відстані.

Переваги Bluetooth для моніторингу автомобіля[15]:

- Низьке енергоспоживання

- Безпека передачі даних. Bluetooth має вбудовані механізми шифрування, що забезпечує захист переданих даних від несанкціонованого доступу, що є важливим аспектом для систем моніторингу, де передаються чутливі дані про стан автомобіля.

Недоліки Bluetooth для моніторингу автомобіля:

- Обмежений діапазон.
- Низька швидкість передачі даних.

1.1.2 Існуючі програми для моніторингу автомобіля

Давайте розглянемо більш детально існуючі додатки для діагностики та моніторингу авто, для прикладу візьмемо додатки Torque та CarScanner. Загалом додатки мають багато спільних функцій таких як:

1. Читання основних параметрів(рис 1.1):
 - Відображення RPM (обертів двигуна).
 - Швидкість автомобіля.
 - Температура охолоджувальної рідини.
 - Напруга акумулятора
 - Тиск в шинах (для підтримуваних моделей автомобілів).
 - Рівень пального.
2. Відображення помилок.
3. Читання кодів помилок (DTC). Дозволяють зчитувати та скидувати коди помилок, що генеруються бортовими системами автомобіля, що допомагає виявити потенційні несправності.
4. Графіки та моніторинг в реальному часі.
5. Автоматичне збереження даних. Додаток може зберігати історію даних для подальшого перегляду і аналізу.

Якщо говорити про специфічні функції, то в Torque я можу виділити пару таких: Torque дозволяє не тільки прочитати коди помилок, але й автоматично відправляти їх до бази даних для пошуку рішення.



Рис 1.1 Приклад інтерфейсу програми Torque

У свою чергу CarScanner пропонує пару інших функцій, які покращують рівень діагностики автомобіля, такі як:

1. Розрахунок розходу палива автомобіля (рис 1.2)

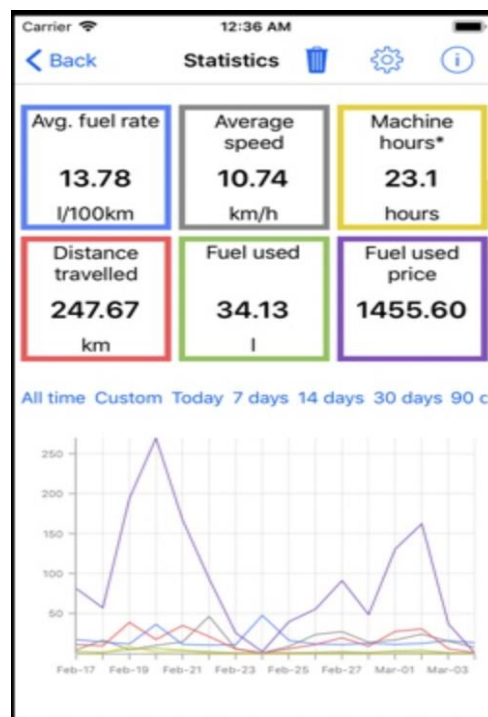


Рис 1.2 Приклад інтерфейсу програми CarScanner

2. Замір розгінних характеристик автомобіля (рис 1.3)

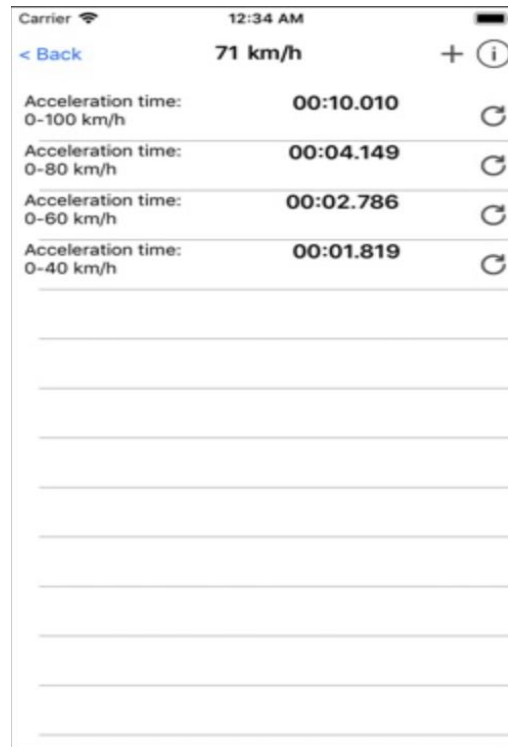


Рис 1.3 Приклад інтерфейсу програми CarScanner

1.2 Опис використання ELM327 для зчитування даних з автомобіля

ELM327 - це адаптер, який дозволяє підключити зовнішні пристрої до автомобільних систем через OBD2 порт для зчитування даних про стан автомобіля. Він є однією з найпоширеніших та бюджетних моделей адаптерів для роботи з OBD2, і широко використовується для діагностики і моніторингу автомобільних параметрів[9].

У низці досліджень, адаптер ELM327 використовувався для збору даних із шини CAN у гібридних електричних транспортних засобах. Автори підкреслюють, що більшість існуючих проєктів використовують ELM327 або аналогічні плати як читачі OBD, які функціонують як інтерфейс між електронним блоком управління (ECU) двигуна та мобільним пристроєм. Такий підхід добре підходить для задач моніторингу та діагностики, де

швидкість опитування не є критичною. Проте, як зазначено у дослідженні, платформа ELM327 має обмеження у витягу даних з усіх електронних блоків управління, а також страждає від високої латентності та низьких частот сканування даних, що обмежує її ефективність у системах реального часу [17].

1.2.1 Принцип роботи ELM327

ELM327 - це мікроконтролер, який дозволяє взаємодіяти з бортовими системами автомобіля через стандартний OBD2 порт. Цей адаптер є ключовим елементом для отримання доступу до численних параметрів, що використовуються для діагностики та моніторингу автомобіля. Він працює, з'єднуючи автомобіль з мобільними пристроями, комп'ютерами або іншими пристроями через стандартні проводові або безпроводові технології, такі як Bluetooth або Wi-Fi.

Принцип роботи:

1. З'єднання з автомобілем.

Перш за все, ELM327 підключається до OBD2 порту автомобіля. Цей порт є стандартним для всіх автомобілів, виготовлених після 1996 року, і забезпечує доступ до різноманітних параметрів системи автомобіля, таких як робота двигуна, трансмісії, системи гальмування, подушок безпеки, та різних бортових електросистем. Порт OBD2 зазвичай знаходиться під кермом, поруч із водійським сидінням, але в автомобілях вироблених в Франції часто полюбують ховати роз'єм в самі незрозумілі та важкодоступні місця[13].

2. Комунікація через UART.

UART - це один з найпоширеніших інтерфейсів для асинхронної передачі даних між двома пристроями. Він використовується для передавання даних через дві лінії зв'язку: передачу (TX) та приймання (RX). UART забезпечує надійний обмін даними без необхідності синхронізувати передачу і прийом сигналів, тому цей стандарт отримав широке застосування в багатьох електронних пристроях.

UART працює на принципі передачі даних по послідовному каналу, що означає, що біт даних передаються один за одним (в одному потоці). Цей інтерфейс широко використовується в різних пристроях, де не потрібно передавати великий обсяг даних за короткий проміжок часу, таких як мікроконтролери, датчики, комп'ютери, модеми тощо[11].

Основні складові UART:

- TX (Transmit) - лінія передачі даних. Використовується для відправки бітів з одного пристрою на інший.
- RX (Receive) - лінія прийому даних. Призначена для отримання бітів, що передаються з іншого пристрою.
- Ground (GND) - загальна лінія заземлення для обох пристроїв, що взаємодіють через UART.

Коли пристрій хоче надіслати дані через UART, він розбиває їх на окремі біти, які передаються один за одним через лінію TX. На приймаючій стороні дані, що приходять по лінії RX, знову збираються в ціле повідомлення. Процес передачі відбувається асинхронно, тобто не потрібен спільний тактовий сигнал для обох пристроїв, що робить UART дуже зручним для використання в системах, де не можна або не потрібно синхронізувати таймінг[11].

Схема зв'язку (рис. 1.4) має такі складові[23]:

- Автомобіль (OBD2 порт). Підключення до автомобільної системи через стандартний OBD2 порт.
- ELM327. Цей адаптер отримує дані з автомобільної електроніки (з допомогою OBD2) та передає їх через UART.
- Bluetooth/Wi-Fi модуль. UART передає отримані дані на Bluetooth або Wi-Fi модуль, який перетворює їх у формат, що підходить для безпроводової передачі.
- Зовнішній пристрій (телефон/комп'ютер). Bluetooth або Wi-Fi приймає дані та передає їх на мобільний телефон або комп'ютер через спеціальний додаток для діагностики (наприклад, Torque або Car Scanner).

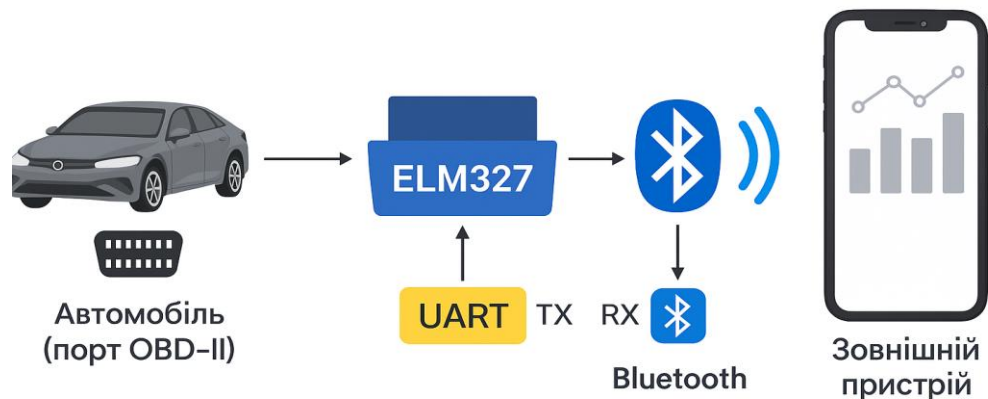


Рис 1.4 Схема зв'язку ELM327[23]

Однією з основних функцій ELM327 є перетворення даних з автомобільної системи в формат, який може бути зрозумілим для користувача. Бортові системи автомобіля використовують специфічні протоколи для передачі даних, і саме ELM327 виступає перекладачем між цими протоколами та звичним форматом, що використовується в мобільних додатках або комп'ютерних програмах. Це дозволяє користувачеві отримувати зрозумілі дані на екрані свого пристрою[23].

Після того, як ELM327 підключено і налаштовано, він зчитує дані про різні параметри, що цікавлять користувача. Це можуть бути RPM (оберти двигуна), швидкість руху, температура двигуна, рівень пального та багато інших параметрів. Далі ці дані передаються на зовнішній пристрій (мобільний телефон або комп'ютер) для аналізу, виведення на екран або збереження для подальшої діагностики.

ELM327 здатний постійно зчитувати та передавати оновлені дані без необхідності ручного втручання, що дозволяє в режимі реального часу моніторити параметри роботи автомобіля, а також отримувати попередження про несправності, коли з'являються проблеми з будь-яким з параметрів (наприклад, перевищення температури або помилки в системах автомобіля).

Переваги використання ELM327[13]:

- Доступність. Адаптер ELM327 доступний за ціною і може бути придбаний у багатьох магазинах електроніки або онлайн.
- Легкість в використанні. Підключення ELM327 до автомобіля та пристрою займає кілька хвилин, а програми для моніторингу автомобіля прості в налаштуванні і мають інтуїтивно зрозумілий інтерфейс.
- Широка сумісність. ELM327 працює з багатьма автомобілями, підтримуючи численні протоколи, що дозволяє його використовувати з більшістю сучасних і старих транспортних засобів.

1.2.2 Методи зчитування даних через ELM327

ELM327 підтримує кілька протоколів для зчитування даних з автомобіля. Кожен протокол має свої характеристики і підходить для різних марок і моделей автомобілів. Ось основні з них[23]:

1. ISO 9141-2. Це один з перших протоколів, який застосовувався у багатьох автомобілях, виготовлених до середини 2000-х років. Він використовує низьку швидкість передачі даних і обмежену пропускну здатність, але є досить простим у реалізації. Типова швидкість передачі даних 10,4 Кбіт/с.

2. ISO 14230 (KWP2000). Цей протокол використовувався у багатьох європейських автомобілях і є більш швидким, ніж ISO 9141-2. KWP2000 має кращу пропускну здатність і підтримує більш точне зчитування даних. Типова швидкість передачі даних 10,4 Кбіт/с.

3. ISO 15765 (CAN). CAN (Controller Area Network) є найбільш поширеним протоколом у нових автомобілях (починаючи з 2008 року). Він забезпечує значно вищу швидкість передачі даних, що дозволяє зчитувати більше параметрів одночасно. Типова швидкість передачі даних 500 Кбіт/с.

4. SAE J1850 PWM. Цей протокол використовувався в більшості американських автомобілів. В основному застосовувався в автомобілях 90-х років. Типова швидкість передачі даних 10,4 Кбіт/с.

Усі протоколи підключення, крім ISO 15765 (CAN) використовують для своєї роботи шину спілкування K-line.

K-line має асинхронний цифровий сигнал (рис 1.5), що передається за допомогою двовартісного сигналу (логічні рівні 0 та 1). Передача даних здійснюється через одну лінію (K-line), що є сигналом для зчитування та передачі даних між автомобільними системами і діагностичними пристроями, що не дає реалізувати більш продуктивну передачу даних тому що шина працює в односторонньому зв'язку (тобто зчитування або передачу даних від пристрою до автомобіля чи навпаки), що обмежує його здатність до передачі великих обсягів даних у реальному часі[10].

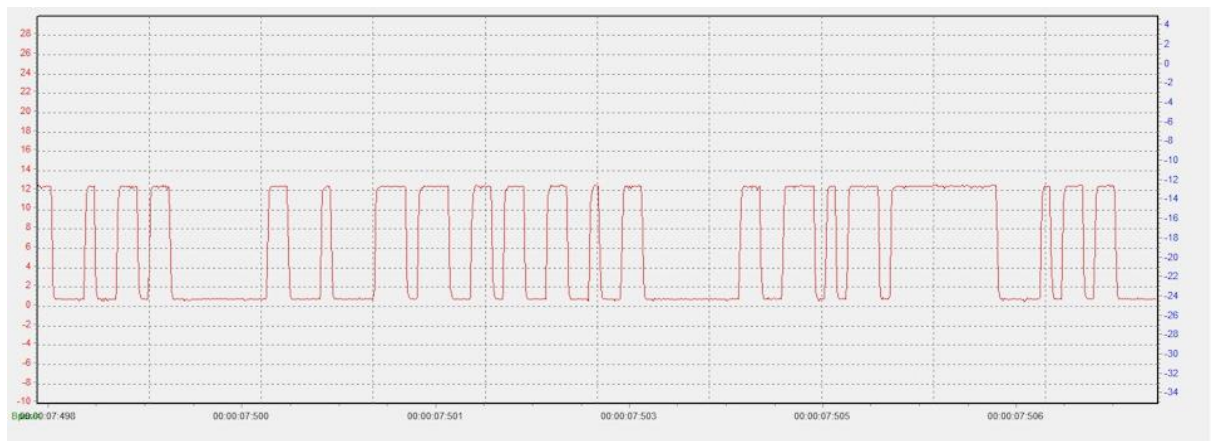


Рис 1.5 Осциллограмма цифрового сигнала авто по K лінії

У свою чергу протокол ISO 15765 (CAN) використовує диференційний сигнал з двома лініями CAN_High та CAN_Low (рис 1.6). Дані передаються через диференційний зв'язок, що дозволяє знижувати вплив шумів. CAN використовує диференційну пару проводів для передачі сигналів. Це дозволяє отримувати стабільні дані навіть в умовах високих електричних шумів, характерних для автомобільних середовищ. CAN_High: вища напруга (логічний рівень 1). CAN_Low: нижча напруга (логічний рівень 0). Під час передачі даних між двома лініями (CAN_High та CAN_Low) різниця

напруги означає "логічний 1" або "логічний 0", тобто якщо CAN_High вищий за CAN_Low - це "1", або ж CAN_Low вищий за CAN_High - це "0"[17].

У CAN шині кожен біт передається протягом певного часу, що визначається швидкістю передачі даних. Наприклад, при швидкості 500 Кбіт/с кожен біт буде передаватися протягом 2 мікросекунд.

Час на передачу 1 біт = $1/\text{Швидкість передачі даних}$
 $= 1/500000 = 0.000002 \text{ сек}$



Рис 1.6 Осциллограмма цифрового сигнала авто по CAN лінії

1.2.3 Переваги зчитування даних через ELM327

Адаптер ELM327 є одним із найпоширеніших інструментів для діагностики та моніторингу технічного стану автомобіля через інтерфейс OBD-II. Його популярність зумовлена поєднанням функціональності, універсальності та доступності. По-перше, ELM327 є економічно вигідним рішенням, що дозволяє власникам транспортних засобів отримувати доступ до основних параметрів двигуна, паливної системи, температури, помилок ECU тощо без потреби у дорогому професійному обладнанні. По-друге, він

сумісний із більшістю легкових автомобілів, що підтримують протокол OBD-II, і здатен працювати через Bluetooth, Wi-Fi або USB. Завдяки своїй компактності, пристрій легко транспортується та не потребує спеціального налаштування. Окрім цього, ELM327 підтримує роботу з численними мобільними застосунками (наприклад, Torque, Car Scanner, OBD Auto Doctor), що значно спрощує процес діагностики та дозволяє використовувати смартфон як інтерфейс зчитування даних. Це особливо вигідно у порівнянні з професійними дилерськими сканерами, які часто потребують стаціонарного ПК із великим програмним забезпеченням та складною процедурою налаштування. Таким чином, ELM327 забезпечує базову, але достатню функціональність для широкого кола користувачів — від автовласників до незалежних майстерень.

1.3 Технології для безпроводового передавання даних

Як зазначають I. Voronko та G. Holub, сучасні безпроводні технології зв'язку, такі як Wi-Fi, Bluetooth, ZigBee, LoRaWAN, NB-IoT та 5G, виступають основними інструментами для реалізації інтелектуальних систем контролю та моніторингу в умовах промислового середовища. Автори здійснили порівняльний аналіз наявних бездротових технологій зв'язку, підкреслюючи переваги їх застосування в автоматизації технологічних процесів. Зокрема, підкреслено, що технологічний прогрес у сфері бездротових систем дозволяє вивести автоматизацію виробництва на якісно новий рівень, підвищити ефективність, знизити витрати та забезпечити надійну роботу обладнання навіть за складних умов експлуатації. Окрему увагу приділено проблемам кібербезпеки: автори аналізують основні загрози — перехоплення трафіку, атаки типу Man-in-the-Middle, DoS/DDoS, експлуатація вразливостей протоколів — та засоби їхньої нейтралізації, включаючи WPA3, Zero Trust Security та AI-орієнтовані методи захисту. Також розглянуто перспективи використання штучного інтелекту, 6G, квантової криптографії та блокчейн-

технологій для посилення безпеки бездротових мереж. Отримані результати свідчать про доцільність впровадження гібридних рішень, які поєднують кілька технологій для оптимізації виробничих процесів та мінімізації кіберризиків [20].

1.3.1 Загальні принципи передачі даних у безпроводних системах

У сучасних системах моніторингу технічних параметрів, особливо в автомобільній галузі, питання передачі даних у режимі реального часу набуває принципового значення. В умовах постійної зміни параметрів руху, як-от швидкість або температура охолоджувальної рідини, дуже важливо, щоб інформація миттєво надходила до користувача без затримок. Тому при проєктуванні системи було обрано безпроводну архітектуру, яка дозволяє відмовитися від дротових рішень, зменшити кількість з'єднань і підвищити гнучкість та мобільність пристрою.

Серед усього різноманіття телекомунікаційних рішень для вбудованих пристроїв, таких як ESP8266, найбільш відповідними для режиму реального часу виявились лише дві технології - Wi-Fi та Bluetooth. Саме ці стандарти забезпечують швидку передачу даних на близькій або середній відстані, з достатньою пропускну здатністю та стабільністю для миттєвого оновлення показників. У цій роботі основну увагу зосереджено саме на них[5].

1.3.2 Wi-Fi як основа телекомунікаційної взаємодії

Вибір на користь Wi-Fi в даному проєкті був обумовлений кількома важливими факторами. По-перше, ESP8266 має повну підтримку цього протоколу, включаючи можливість створення власної точки доступу (режим Access Point). Це дає змогу зробити систему повністю автономною: вона створює власну Wi-Fi мережу, до якої користувач може підключитися напряму зі смартфона або ноутбука.

Цей режим реалізовано таким чином: ESP8266 транслює Wi-Fi-мережу з унікальним SSID (наприклад, OBD_Monitor), надає клієнтам IP-адреси через DHCP (зазвичай у підмережі 192.168.4.0/24), і одночасно виступає вебсервером, надаючи HTML-інтерфейс із відображенням параметрів у реальному часі. У цьому сценарії не потрібні ні роутер, ні підключення до інтернету - все працює локально, автономно та з мінімальною затримкою.

Переваги цього підходу[5]:

- Швидкість передачі даних (до 72 Мбіт/с для стандарту 802.11n при хороших умовах);
- Одночасне обслуговування кількох клієнтів (можна підключити декілька пристроїв одночасно);
- Відносно велика відстань дії - до 50 метрів у відкритому просторі;
- Простота реалізації веб-інтерфейсу - можливість використання вбудованого веб-сервера безпосередньо на ESP8266.

У поєднанні з можливістю виведення даних у вигляді динамічних графіків (через JavaScript), Wi-Fi став логічним вибором для реалізації оновлення показників без затримки.

1.3.3 Порівняння технологій Wi-Fi та Bluetooth у контексті автомобільного моніторингу

Значна частина портативних пристроїв, зокрема недорогі OBD2-адаптери, використовують Bluetooth, особливо його енергоефективний варіант - Bluetooth Low Energy (BLE). Проте під час технічного аналізу Bluetooth виявився менш придатним для реалізації повноцінної системи моніторингу у реальному часі[6]. Нижче наведено порівняльну таблицю основних параметрів двох технологій у контексті побудови мого проєкту (рис 1.7).

Характеристика	Wi-Fi (ESP8266)	Bluetooth (BLE)
Швидкість передачі	до 72 Мбіт/с	до 2 Мбіт/с
Дальність	30-50 м	5-10 м
Підтримка кількох клієнтів	Так	Обмежено
Вебінтерфейс	Так	Відсутня
Затримка при передачі	Низька	Середня
Залежність від додатку	Немає	Потрібен додаток

Рис 1.7 Порівняльна таблиця основних параметрів двох технологій

Як видно з рисунку, Wi-Fi забезпечує кращу стабільність, швидкість та зручність для користувача, особливо коли йдеться про передачу динамічних даних, які повинні оновлюватися на екрані без затримки. Bluetooth, хоч і має нижче енергоспоживання, програє у дальності та зручності інтерфейсу: замість звичного браузера користувачу потрібен окремий мобільний додаток, який повинен бути завжди активним і підтримувати роботу з Bluetooth-каналом.

У реальних сценаріях Bluetooth виявляється також нестабільним у багатьох моделях авто, де цей модуль уже використовується для hands-free або мультимедійних систем, що може призводити до конфліктів[6].

Таким чином, з огляду на вимоги до надійного режиму реального часу, було прийнято рішення використовувати саме Wi-Fi як основну технологію. Це відповідає концепціям телекомунікаційного проектування, орієнтованого на забезпечення безперервного, швидкого та надійного обміну даними у локальному середовищі без залежності від зовнішніх ресурсів.

1.4 Огляд використання NodeMCU (ESP8266) в системах IoT

На сучасному етапі розвитку інфокомунікаційних технологій усе більшу популярність набирає концепція Інтернету речей (IoT), яка передбачає об'єднання фізичних пристроїв у єдину мережу з метою збору, обробки й передачі даних. У такій архітектурі саме мікроконтролери з вбудованими засобами зв'язку, зокрема безпроводного, стають основою для побудови практичних рішень у різних сферах - від побутової автоматизації до промислового моніторингу. Особливо актуальним є впровадження таких

рішень у транспортній галузі, де відстеження технічного стану автомобіля в реальному часі дозволяє запобігати поломкам, підвищувати безпеку та знижувати витрати на обслуговування[3].

У межах даного дослідження було обрано використання апаратної платформи NodeMCU, що побудована на базі модуля ESP8266. Це рішення є добре відомим серед розробників систем автоматизації завдяки своїй простоті, широкій функціональності та надзвичайно вдалій інтеграції телекомунікаційних можливостей у компактному форм-факторі. На відміну від багатьох мікроконтролерів, які потребують окремого зовнішнього модуля для доступу до мережі, ESP8266 має на борту повний TCP/IP-стек, що дає змогу реалізовувати обмін даними за допомогою Wi-Fi без проміжних пристроїв[12].

Серед переваг модуля ESP8266 варто відзначити його низьку вартість, достатню обчислювальну потужність (з тактовою частотою до 80 або 160 МГц), наявність вбудованої флеш-пам'яті (до 4 МБ), підтримку цифрових і аналогових входів/виходів, а також сумісність із популярним середовищем розробки Arduino IDE. Для інженера це означає мінімальні бар'єри на старті, оскільки наявні бібліотеки дозволяють реалізувати практично будь-яку функціональність - від створення вебсервера до обміну даними через UART чи SPI.

Особливо зручно, що ESP8266 може працювати у різних режимах - як клієнт мережі Wi-Fi, так і як точка доступу, до якої можуть підключатися інші пристрої. У межах мого проєкту реалізовано саме другий варіант: мікроконтролер створює автономну мережу Wi-Fi, до якої можна підключитися зі смартфона чи ноутбука й отримати доступ до вебінтерфейсу з відображенням даних. Це дозволяє уникнути залежності від зовнішніх точок доступу чи інтернету, що є ключовим при застосуванні пристрою в умовах рухомого транспорту[14].

У системі моніторингу, яку я проєктую, NodeMCU не лише приймає дані з адаптера ELM327 через інтерфейс UART, а й обробляє їх, формує зручний

HTML-інтерфейс і транслює його в реальному часі через вбудований вебсервер. Цей вебінтерфейс зберігається у флеш-пам'яті модуля завдяки файловій системі, що дозволяє без зайвих затрат відображати дані напряму в браузері користувача. Такий підхід зручний тим, що не потребує встановлення додаткових програм або клієнтів, а значить - забезпечує простий та інтуїтивно зрозумілий доступ до показників у будь-який момент.

У практичному сенсі NodeMCU (ESP8266) став своєрідним стандартом у галузі бюджетного IoT. Завдяки своїй відкритій архітектурі, підтримці великої кількості бібліотек і спільноті ентузіастів, що постійно оновлює та покращує документацію, він вважається ідеальним вибором для студентських проєктів, дослідницьких розробок, а також для впровадження в реальні продукти. До прикладу, схожі рішення вже давно застосовуються у системах «розумного дому» (освітлення, опалення), у сільському господарстві (контроль вологості ґрунту) та в охороні (сенсори руху, відеонагляд). У всіх цих випадках використовується один і той самий принцип - зчитування фізичних параметрів, їх обробка на мікроконтролері та передача через бездротову мережу до кінцевого користувача[12].

У контексті дипломної роботи ESP8266 виявився саме тим компонентом, що дозволив об'єднати різні частини системи моніторингу - ELM327, вебінтерфейс, Wi-Fi - у єдину логічну схему. Така інтеграція відповідає принципам сучасної телекомунікаційної інженерії, де кожен елемент виконує кілька функцій і забезпечує гнучкість під час масштабування системи. У разі потреби плату можна легко замінити на ESP32, яка має більшу тактову частоту, більше пам'яті та кращу енергоефективність.

Отже, застосування NodeMCU (ESP8266) у побудові безпроводної системи моніторингу технічних параметрів автомобіля є не лише доцільним, а й обґрунтованим із технічної, економічної та телекомунікаційної точки зору. Це рішення дає змогу реалізувати повноцінну систему у реальному часі без зайвих затрат, із збереженням стабільності, мобільності та зручності для

кінцевого користувача. Саме тому ESP8266 і надалі залишається перевіреним вибором для широкого спектра IoT-проектів.

Висновок

У результаті аналізу, проведеного в першому розділі, вдалося глибше зрозуміти сучасні підходи до побудови систем для моніторингу технічного стану автомобілів, зокрема з використанням бездротових каналів зв'язку. Акцент зроблено на актуальності телеметричних рішень, які забезпечують стабільне зчитування та передачу даних у режимі реального часу, без потреби у фізичному підключенні до комп'ютера або сервісного обладнання.

Ретельно вивчено стандарти протоколу OBD-II, що стали галузевим еталоном у сфері діагностики транспортних засобів. У межах дослідження охоплено особливості найбільш поширених варіантів, серед яких ISO 9141-2 та ISO 15765-4 (CAN), що забезпечують широке охоплення сумісних автомобілів. Також розглянуто принципи роботи адаптера ELM327 як ключової ланки між транспортним засобом і мікроконтролерною системою збору даних.

Особливу увагу було приділено порівнянню технологій бездротової передачі - Wi-Fi та Bluetooth - з урахуванням специфіки їх використання в умовах автомобільної експлуатації. На основі аналізу швидкодії, радіусу дії, енергоспоживання та стабільності зв'язку зроблено висновок, що Wi-Fi є найбільш прийнятним варіантом для забезпечення оперативного та стабільного доступу до параметрів транспортного засобу.

Таким чином, перший розділ став основою для формування цілісного бачення архітектури майбутньої системи моніторингу та надав необхідну теоретичну базу для обґрунтованого вибору компонентів та принципів їх взаємодії на наступних етапах проектування.

РОЗДІЛ 2

ПРОЕКТУВАННЯ БЕЗПРОВОДОВОЇ СИСТЕМИ МОНИТОРИНГУ

2.1 Вибір та обґрунтування компонентів системи.

Проектування будь-якої вбудованої телекомунікаційної системи, особливо у сфері автомобільного моніторингу, починається з ретельного вибору апаратних та програмних компонентів. Основна мета цього етапу - сформувати збалансований набір елементів, які взаємодіятимуть між собою на апаратному та логічному рівні, забезпечуючи збір, обробку, передачу та відображення даних про технічний стан автомобіля у реальному часі. У межах дипломного проєкту було обрано компоненти, які не лише відповідають вимогам стабільної телеметрії, але й забезпечують простоту інтеграції, низьке енергоспоживання та економічну доцільність.

2.1.1 NodeMCU (ESP8266)

Плата NodeMCU, побудована на базі мікроконтролера ESP8266, стала основним апаратним елементом системи завдяки своїй універсальності та високому ступеню інтеграції з бездротовими мережами. Вбудований модуль Wi-Fi, що підтримує стандарти IEEE 802.11 b/g/n, дозволяє використовувати ESP8266 у якості як клієнта, так і точки доступу. У даному проєкті обрано саме режим точки доступу, що забезпечує автономність та незалежність від сторонніх мереж. Мікроконтролер працює на частоті до 160 МГц, має до 4 МБ флеш-пам'яті, оперативну пам'ять до 80 КБ, а також підтримує різноманітні інтерфейси: GPIO, SPI, I2C, PWM, ADC. ESP8266 повністю сумісний із середовищем Arduino IDE, що спрощує розробку прошивки та інтеграцію з іншими модулями. Сучасні альтернативи на базі ESP32 мають більшу обчислювальну потужність, розширену кількість портів вводу-виводу та енергоефективні режими сну, але у межах цього проєкту ESP8266 забезпечує

оптимальний баланс між ціною, габаритами та функціональністю для системи моніторингу[7].

2.1.2 OBD2-адаптер на базі ELM327

У ролі засобу зчитування діагностичної інформації з електронного блоку управління автомобіля використовується адаптер, побудований на чипі ELM327. Цей чип підтримує основні OBD-II протоколи, включаючи ISO 9141-2, ISO 14230-4 (KWP2000) та ISO 15765-4 (CAN), що дозволяє йому взаємодіяти з більшістю автомобілів, починаючи з 2000 року випуску. У рамках проєкту використано модифікацію з UART-інтерфейсом, яка забезпечує пряме підключення до ESP8266. Комунікація відбувається через стандартні AT-команди у вигляді ASCII-запитів, що повертають значення PIDs (Parameter IDs), зокрема швидкості руху, температури охолоджувальної рідини та інших технічних параметрів. Напруга живлення ELM327 адаптера, як правило, становить 12 В, але логічні рівні для UART знижені до TTL-рівня (3.3 В), що повністю відповідає специфікації ESP8266 та виключає потребу в логічних перетворювачах[18].

2.1.3 Інтерфейс UART

Для передачі даних між NodeMCU (ESP8266) та адаптером ELM327 використовується інтерфейс UART - універсальний асинхронний приймач-передавач. Це надійний і простий спосіб організації послідовного обміну даними між мікроконтролерами. ESP8266 має апаратну підтримку UART0, яку використано у цьому проєкті для ініціалізації каналу зв'язку з ELM327. Комунікація здійснюється за стандартним принципом: вивід TX одного пристрою підключено до RX іншого, і навпаки. Швидкість обміну зазвичай становить 9600 або 115200 бод, залежно від налаштувань адаптера. Для створення додаткових UART-портів (у разі потреби підключення інших

модулів) може бути використано програмну емуляцію через бібліотеку SoftwareSerial. Така архітектура дозволяє зберігати простоту схеми та надійність зв'язку при одночасному збереженні гнучкості системи.

2.1.4 Електроживлення

Забезпечення стабільного та безпечного електроживлення системи є критично важливим, особливо з огляду на специфіку роботи в автомобільному середовищі. ESP8266 потребує напругу 3.3 В, проте плата NodeMCU має вбудований стабілізатор і допускає подачу живлення через USB (5 В) або VIN (6–12 В). Стандартний OBD-II порт автомобіля подає 12-14.2 В, тому доцільно використати понижуючий стабілізатор типу step-down (DC-DC) на базі мікросхем AMS1117 або MP1584, який забезпечить стабільну напругу без перегріву[18]. Для запобігання поломкам, викликаним перенапругою або імпульсними завадами, передбачено використання запобіжників та ємнісних фільтрів. У лабораторних умовах живлення подавалось через USB, проте у варіанті для реального застосування можлива інтеграція з бортовою мережею автомобіля з використанням вбудованих перетворювачів та захистів.

2.2 Схема з'єднання ELM327 з NodeMCU

Побудова надійного фізичного каналу зв'язку між діагностичним адаптером ELM327 та контролером NodeMCU (на базі ESP8266) є одним із найважливіших етапів у створенні ефективної системи моніторингу. Саме від правильності цього з'єднання залежить стабільність, швидкість і точність передавання даних з автомобіля на мікроконтролер. У межах даного проєкту було прийнято рішення реалізувати зв'язок за допомогою інтерфейсу UART (Universal Asynchronous Receiver-Transmitter), який є одним з найпоширеніших і надійних для обміну інформацією між пристроями у вбудованих системах.

Для обміну інформацією було обрано адаптер ELM327 з підтримкою UART-комунікації. Це дозволило уникнути використання Bluetooth-модуля або USB-шлюзу, що могло б ускладнити інтеграцію з NodeMCU та потребувало б додаткових бібліотек або драйверів. Підключення здійснюється через стандартні порти TX, RX та GND, а саме:

- TX (ELM327) - передає дані та з'єднується з RX (NodeMCU), який їх приймає;
- RX (ELM327) - приймає дані та з'єднується з TX (NodeMCU);
- GND (ELM327) - загальна земля, під'єднується до GND (NodeMCU) для узгодження рівнів напруги.

Важливим фактором, який був врахований при реалізації цього з'єднання, є логічні рівні. NodeMCU працює на напрузі 3.3 В, що збігається з TTL-рівнями більшості UART-адаптерів ELM327, тому з'єднання може відбуватися напругу, без використання логічних перетворювачів, що, у свою чергу, зменшує складність схеми і ризик виникнення помилок при збиранні.

У процесі тестування та налагодження було обрано стандартну швидкість обміну - 9600 бод, яка підтримується більшістю версій ELM327. Така швидкість забезпечує баланс між стабільністю передавання та достатньою швидкістю для зчитування параметрів у режимі, наближеному до реального часу. Водночас у майбутньому можливе збільшення швидкості до 38400 або навіть 115200 бодів для прискорення опитування більшої кількості параметрів.

Після фізичного з'єднання необхідно правильно ініціалізувати ELM327. На початку сеансу обміну ESP8266 надсилає набір ASCII-команд (зокрема ATZ, ATE0, ATL0, ATSP0), які відповідають за скидання, вимкнення відлуння, форматування та вибір протоколу. Лише після цього адаптер готовий до обробки запитів типу 010D (швидкість), 0105 (температура охолоджувальної рідини) тощо.

З'єднання має ще одну перевагу - мінімальну кількість проміжних вузлів. Передавання команд відбувається безпосередньо, без будь-яких

складних програмних стеків або протоколів, що знижує затримку та підвищує надійність системи. Крім того, UART працює незалежно від основного Wi-Fi модуля ESP8266, що дозволяє підтримувати паралельність операцій - одночасне опитування адаптера і передавання даних на веб-інтерфейс.

З практичної точки зору, таке з'єднання дозволило досягти високого рівня стабільності. У ході експериментів не було зафіксовано збоїв у синхронізації або втрати даних при стандартному обміні. Це підтверджує, що UART з ELM327 є цілком придатним рішенням для телеметрії в умовах реального автомобільного середовища.

2.3 Технічні вимоги до системи моніторингу

Проектування безпроводової системи моніторингу технічного стану автомобіля вимагає чіткого формування технічних вимог, які визначають архітектуру, функціональні обмеження, рівень надійності та взаємодію із зовнішніми середовищами. Оскільки система будується в рамках сучасного телекомунікаційного підходу, вона повинна не лише відповідати загальним вимогам вбудованих електронних пристроїв, але й забезпечувати стабільну передачу даних у реальному часі, незалежність від інфраструктури, а також масштабованість для потенційного розширення функціональності[2].

2.3.1 Функціональні вимоги

- Зчитувати дані з ELM327-адаптера через UART-інтерфейс з інтервалом не більше 0.5 секунди;
- Коректно обробляти та декодувати дані з OBD-II, зокрема PIDs 010D (швидкість), 0105 (температура охолоджувальної рідини) та 010B (тиск масла);
- Транслювати зібрані дані на веб-інтерфейс у вигляді читабельного GUI з можливістю оновлення значень без перезавантаження сторінки;

- Функціонувати автономно як точка доступу Wi-Fi з фіксованим SSID, до якої може підключатися користувач без використання зовнішніх маршрутизаторів;
- Мати систему індикації статусу підключення (online/offline) для виявлення втрат зв'язку з ELM327 або внутрішніх збоїв;\
- Забезпечувати надійність і повторюваність даних протягом щонайменше 30 хвилин безперервної роботи.

-

2.3.2 Електротехнічні та апаратні вимоги

- Живлення пристрою має здійснюватися або через microUSB (5 В), або через VIN-лінію з автомобільної бортмережі (6–12 В), з додатковим стабілізатором напруги;
- Система має бути захищена від перенапруги, електромагнітних завад та зворотної полярності;
- Усі з'єднання повинні бути виконані з урахуванням надійного монтажу для автомобільного середовища (вібрації, перепади температур, пил, волога)
- UART-підключення має дотримуватися рівнів логіки 3.3 В між ELM327 та ESP8266, з використанням перехресного підключення TX↔RX.

2.3.3 Програмні вимоги

- Скетч повинен компілюватися в Arduino IDE без зовнішніх файлових систем (без LittleFS або SPIFFS);
- Вебінтерфейс має бути вбудований у прошивку як статичний HTML-код;
- Передача даних має здійснюватися через HTTP-запити, API-сервер повинен повертати JSON-відповіді;

- Код повинен містити обробку винятків, пов'язаних з непрацездатністю ELM327 (відсутність відповіді, помилковий формат, тощо);
- Код повинен мати можливість розширення: підключення додаткових датчиків, використання графіків, або перехід на ESP32.

2.3.4 Комунікаційні вимоги

- Wi-Fi точка доступу має забезпечувати стабільне підключення на відстані не менше 10 метрів у відкритому просторі;
- Передача даних має відбуватися з мінімальною затримкою (менше 100 мс між зчитуванням параметра та його відображенням на вебінтерфейсі);
- Система має бути здатна до роботи в середовищі з потенційними перешкодами у 2.4 ГГц діапазоні (іншими мережами, Bluetooth, мікрохвильовими печами тощо);
- Взаємодія з браузером має бути максимально сумісною (Google Chrome, Firefox, Safari).
-

2.3.5 Надійність і безпека

- Програма має працювати без збоїв протягом тривалого часу в умовах руху транспортного засобу;
- Система має містити тайм-аут на випадок відсутності відповіді від ELM327;
- Необхідно забезпечити перевірку кожної відповіді на відповідність стандарту OBD-II перед її обробкою;
- Для захисту від стороннього доступу передбачена можливість встановлення пароля на Wi-Fi точку доступу.

Ці технічні вимоги є основою для всіх подальших етапів розробки, тестування та вдосконалення системи. Вони орієнтовані не лише на поточну реалізацію проєкту, але й враховують перспективи масштабування,

використання більш потужних платформ, інтеграцію з мобільними застосунками та віддаленими серверними хмарними рішеннями.

Висновок

Другий розділ був присвячений технічному обґрунтуванню архітектури майбутньої системи моніторингу, зокрема вибору мережевої технології, апаратної платформи та логіки взаємодії між її основними модулями. В процесі аналізу було проаналізовано вимоги до обміну даними в умовах реального часу, з урахуванням обмежень, які накладає автомобільне середовище.

Особливої уваги було надано порівнянню варіантів бездротового зв'язку - Wi-Fi та Bluetooth. У результаті було зроблено обґрунтований вибір на користь Wi-Fi, який дозволяє не лише організувати локальну мережу без потреби у зовнішніх маршрутизаторах, але й забезпечує стабільний канал з вищою пропускною здатністю, що критично важливо для візуалізації даних у реальному часі.

Також детально розглянуто принципи функціонування інтерфейсу OBD-II, який є галузевим стандартом діагностики у транспортних засобах. Було вивчено специфіку взаємодії адаптера ELM327 з мікроконтролером через UART-з'єднання. Це з'єднання є одним з найпростіших і найнадійніших варіантів для двостороннього обміну даними, особливо в умовах обмежених ресурсів.

У підсумку, розділ не лише систематизував технічну базу, але й дозволив чітко сформулювати концепцію побудови системи, в якій кожен компонент відіграє свою логічну роль. Ретельний вибір елементів та технологій, який базується на практичних і телекомунікаційних критеріях, заклав надійний фундамент для подальшої реалізації повноцінної системи моніторингу транспортного засобу

РОЗДІЛ 3

РОЗРОБКА ВЕБ-СИСТЕМИ ДЛЯ ВІДОБРАЖЕННЯ ДАНИХ

3.1 Структура веб-сторінки для моніторингу

Інтерфейс для користувача у системі моніторингу створено у вигляді простої веб-сторінки, яка відкривається через браузер після підключення до Wi-Fi мережі, створеної ESP8266. Головною метою сторінки є надання миттєвого доступу до поточних технічних параметрів автомобіля у зручному і зрозумілому форматі, без зайвих елементів.

Весь інтерфейс зосереджений на трьох основних показниках:

- Швидкість руху автомобіля
- Температура охолоджувальної рідини
- Тиск масла (якщо параметр доступний)

Дані виводяться у вигляді великих круглих індикаторів з числовими значеннями, розміщених по центру сторінки. Вище розташовано заголовок: «Технічні параметри автомобіля», виконаний чітким шрифтом. Такий підхід дозволяє сфокусувати увагу користувача саме на поточному стані автомобіля.

З правого боку, у верхній частині сторінки, розміщено індикатор статусу з'єднання, який показує «ONLINE» при активному обміні даними між NodeMCU та вебінтерфейсом. Це реалізовано через візуальний маркер у вигляді мітки зі змінним кольором чи текстом.

Веб-сторінка побудована за допомогою HTML та CSS без зовнішніх бібліотек, що дозволяє зберігати її у пам'яті ESP8266 (через файлову систему типу SPIFFS або LittleFS). Оскільки обсяг коду мінімальний, завантаження сторінки здійснюється дуже швидко, навіть при повільному Wi-Fi-з'єднанні.

Дані оновлюються без перезавантаження сторінки завдяки JavaScript-запитам до локального вебсервера, розгорнутого на NodeMCU. Усі відповіді приходять у вигляді простих текстових значень, які обробляються клієнтом і вставляються у відповідні HTML-елементи.(рис 3.1).

- Мінімалістичний дизайн без анімацій, графіків або складної візуалізації. Це забезпечує стабільну роботу навіть на старих браузерях чи малопотужних мобільних пристроях.
- Чітка типографіка та великі шрифти, які дозволяють зчитувати значення з відстані, що зручно при розміщенні смартфона на тримачі в автомобілі.
- Індикатор недоступності параметра - якщо значення недоступне, наприклад, тиск масла не підтримується автомобілем, у полі відображається позначка NA (Not Available), що дозволяє відрізнити тимчасову відсутність сигналу від некоректної роботи.

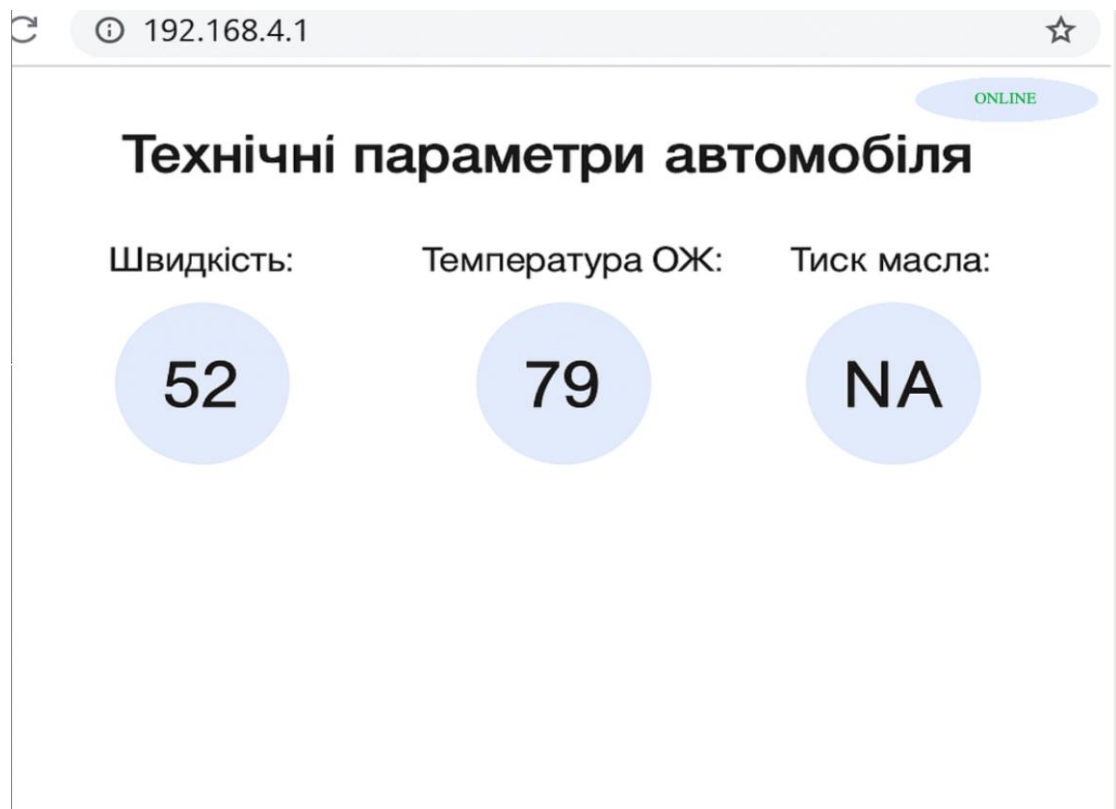


Рис 3.1 Вигляд веб-інтерфейсу

Таким чином, створена веб-сторінка є прикладом ефективного застосування спрощеного інтерфейсу в телекомунікаційній системі IoT, де пріоритет надано швидкодії, сумісності та стабільності. Замість надмірної візуалізації, фокус зміщено на читабельність і зручність для користувача в реальному середовищі - за кермом або під час діагностики авто на місці.

3.2 Використання JavaScript для відображення даних у реальному часі

Оновлення даних у реальному часі - це ключовий аспект функціонування будь-якої сучасної системи моніторингу. У межах реалізованої веб-сторінки для відображення технічних параметрів автомобіля саме JavaScript відповідає за періодичне опитування вбудованого вебсервера на NodeMCU та динамічну зміну вмісту елементів інтерфейсу без перезавантаження сторінки. Таким чином реалізується інтерактивна поведінка інтерфейсу з оновленням значень швидкості, температури охолоджувальної рідини та тиску масла.

Основний принцип роботи JavaScript у системі

У вебсторінку вбудовується скрипт, який через рівні проміжки часу виконує HTTP-запит до конкретного маршруту вебсервера, що запущений на ESP8266. У відповідь сервер повертає актуальні дані, після чого скрипт оновлює DOM-елементи, у яких відображаються ці значення.

Цей підхід дозволяє забезпечити режим оновлення без перезавантаження сторінки. Це критично важливо в контексті безпроводового з'єднання через точку доступу, оскільки кожне повне оновлення сторінки призводило б до зайвого навантаження на Wi-Fi модуль та затримки.

Технічна реалізація

Нижче наведено спрощений приклад логіки на JavaScript (рис 3.2), яка використовується для отримання даних зі сторони NodeMCU:

```

1  setInterval(function() {
2      fetch('/data')
3          .then(response => response.json())
4          .then(data => {
5              document.getElementById('speed').innerText = data.speed;
6              document.getElementById('temp').innerText = data.coolant;
7              document.getElementById('oil').innerText = data.oil || 'NA';
8          })
9          .catch(err => {
10             console.error('Помилка з'єднання:', err);
11         });
12 }, 2000);
13 |

```

Рис 3.2 Логіка JavaScript

У цьому прикладі:

- `Fetch('/data')` - виконує GET-запит до ендпоінту `/data`, який обробляється сервером на ESP8266;
- У відповідь отримується об'єкт `data` з трьома властивостями: `speed`, `coolant`, `oil`;
- За допомогою `innerText` змінюються значення відповідних елементів у HTML-структурі;
- Якщо параметр недоступний (наприклад, `oil` повертає `null` або `undefined`), відображається позначка `NA`.

З боку телекомунікацій важливо зазначити, що JavaScript-запити не створюють нове TCP-з'єднання щоразу, а використовують постійне підключення до локального сервера, відкритого ESP8266. Це дозволяє мінімізувати затримки, покращити ефективність передачі і знизити ризик перевантаження системи[24].

Оскільки система працює у режимі точки доступу (Access Point), то всі JavaScript-запити надсилаються безпосередньо в межах однієї локальної мережі, що забезпечує низьку затримку (~5–10 мс) та відсутність втрат даних, характерних для публічних мереж.

Для запобігання збоїв реалізована елементарна обробка помилок: якщо запит не вдається, система не зависає, а просто пропускає поточне оновлення і пробує знову через наступний інтервал.

Завдяки JavaScript логіка оновлення даних працює однаково стабільно як на стаціонарному ПК, так і на смартфоні, підключеному до Wi-Fi мережі NodeMCU. Це робить систему зручною для реального використання в польових умовах або під час руху, коли доступ до інформації повинен бути максимально швидким і безпечним.

Таким чином, завдяки JavaScript вдалося реалізувати легку, ефективну і незалежну систему оновлення даних, що забезпечує режим реального часу без складних фреймворків чи залежностей. Такий підхід повністю відповідає вимогам телекомунікаційних рішень у контексті IoT: автономність, мінімальні затримки, стабільна локальна взаємодія та наочність для користувача.

3.3 Розробка графічної частини сторінки

Одним із важливих елементів зручної взаємодії користувача з системою моніторингу є грамотне графічне оформлення вебінтерфейсу. Хоча в даній реалізації не використовуються анімації чи складна побудова графіків, значення параметрів, що відображаються, мають бути візуально чіткими, інтуїтивно зрозумілими та придатними для читання в умовах швидкого погляду - наприклад, під час керування автомобілем або діагностики «на ходу».

Під час створення інтерфейсу я дотримувався кількох ключових принципів, які характерні для простих і ефективних вебзастосунків у сфері телекомунікацій:

- **Мінімалізм** - усунуто зайві графічні елементи, залишено лише основне: заголовки, індикатори параметрів, статус з'єднання.
- **Контрастність** - великі чорні цифри на світлому фоні добре видно навіть за денного освітлення.

- Централізоване розміщення - показники виведені в один ряд, рівномірно по ширині екрана, що полегшує візуальне сприйняття.
- Мобільна адаптивність - інтерфейс виглядає добре як на широкоформатному дисплеї, так і на екрані смартфона.

HTML-структура сторінки

Основна розмітка інтерфейсу побудована на стандартному HTML5 (рис 3.3)[25]. Нижче наведено фрагмент HTML-коду, що відображає розміщення індикаторів:

```

1      <h2>Технічні параметри автомобіля</h2>
2      <div class="container">
3          <div class="circle">
4              <p>Швидкість:</p>
5              <span id="speed">--</span>
6          </div>
7          <div class="circle">
8              <p>Температура ОЖ:</p>
9              <span id="temp">--</span>
10         </div>
11         <div class="circle">
12             <p>Тиск масла:</p>
13             <span id="oil">NA</span>
14         </div>
15     </div>
16

```

Рис 3.3 Структура сторінки HTML

Ці блоки розміщені у гнучкому контейнері, який адаптується до ширини екрана. Всі три індикатори мають однакову структуру, що полегшує як написання стилів, так і оновлення JavaScript-кодом.

Оформлення реалізовано засобами CSS3 (рис 3.4). Головна задача стилізації - зробити інформацію легко зчитуваною та акуратно впорядкованою. Приклад CSS-коду для стилізації круглих блоків із даними

```

1  .container {
2      display: flex;
3      justify-content: center;
4      gap: 40px;
5      margin-top: 20px;
6  }
7  .circle {
8      width: 100px;
9      height: 100px;
10     background-color: #e0e7ff;
11     border-radius: 50%;
12     display: flex;
13     flex-direction: column;
14     align-items: center;
15     justify-content: center;
16     font-size: 24px;
17     font-weight: bold;
18 }
19 |

```

Рис 3.4 Оформлення CSS

Колір фону підібрано пастельний, що не створює візуального переважання. Використання властивості `border-radius: 50%` дозволяє сформувати рівні круглі блоки - інтуїтивні маркери для кожного з параметрів.

Статус з'єднання Додатковий елемент графічної частини - це індикатор статусу «ONLINE», який виводиться у верхньому правому куті сторінки. Він стилізований як невелика напівпрозора овальна кнопка (Рис 3.5)

```

1  .status {
2      position: absolute;
3      top: 10px;
4      right: 20px;
5      background-color: #d1e7dd;
6      padding: 5px 10px;
7      border-radius: 20px;
8      color: #0f5132;
9      font-size: 12px;
10 }
11 |

```

Рис 3.5 Реалізація кнопки ONLINE

У разі відсутності з'єднання колір змінюється на червоний, а напис - на «OFFLINE». Така проста графіка допомагає користувачу миттєво оцінити, чи надходять дані.

Упор на ефективність

Усі стилі вбудовані прямо у сторінку, що дозволяє завантажувати лише один HTML-файл без окремих .css-файлів. Це зменшує обсяг переданих даних, що особливо важливо при роботі через вбудований Wi-Fi на ESP8266 із обмеженим обсягом флеш-пам'яті.

Таким чином, графічна частина веб-сторінки була спроектована не лише з точки зору візуального комфорту, а й з урахуванням обмежень апаратної платформи, вимог стабільності у телекомунікаційному середовищі та зручності використання на різних типах пристроїв. Мінімалістичний дизайн, чітка типографіка, логічне компоновання - усе це робить вебінтерфейс придатним для повсякденного використання у транспортній телеметрії.

3.4 Підключення NodeMCU до веб-сторінки через Wi-Fi

Ключовим елементом у реалізації безпроводової системи моніторингу є забезпечення стабільного зв'язку між мікроконтролером NodeMCU (на базі ESP8266) та кінцевим користувачем. У даному проєкті для цього використовується Wi-Fi-зв'язок у режимі точки доступу (Access Point), що дозволяє повністю уникнути зовнішньої інфраструктури, такої як роутери або мобільний інтернет.

NodeMCU налаштовано як точку доступу, тобто самостійно створює локальну Wi-Fi мережу, до якої може підключитися будь-який пристрій - смартфон, планшет або ноутбук (рис 3.6) Такий підхід дає змогу зробити систему автономною та мобільною, що критично важливо у контексті діагностики автомобіля у польових умовах або під час руху.

Конфігурація виконується у прошивці наступним чином:

```

1  #include <ESP8266WiFi.h>
2
3  const char* ssid = "OBD_Monitor";
4      const char* password = "12345678";
5
6  void setup() {
7      WiFi.softAP(ssid, password);
8      IPAddress IP = WiFi.softAPIP();
9      Serial.print("AP IP address: ");
10     Serial.println(IP);
11 }
12 |

```

Рис 3.6 Налаштування точки доступу ESP8266

У цьому коді `WiFi.softAP()` створює точку доступу, клієнтські пристрої підключаються до неї за логіном `OBD_Monitor` після підключення користувач відкриває у браузері адресу `192.168.4.1`, яка є IP NodeMCU в цій локальній мережі.

Вбудований вебсервер розгорнуто на мікроконтролері, який обробляє запити клієнта і повертає HTML-сторінку(рис 3.7) Це дозволяє уникнути залежності від хостингів чи зовнішніх API. Для реалізації використано бібліотеку `ESP8266WebServer`

```

1  #include <ESP8266WebServer.h>
2
3  ESP8266WebServer server(80);
4
5  void setup() {
6      server.on("/", handleRoot);
7      server.begin();
8  }
9
10 void handleRoot() {
11     server.send(200, "text/html", htmlContent);
12 }
13 |

```

Рис 3.7 Розгортання веб-сервера

Обмін даними в режимі реального часу

Коли сторінка завантажена, браузер ініціює асинхронні JavaScript-запити до іншого ендпоінту, наприклад /data, звідки отримує поточні значення параметрів.

NodeMCU реагує на ці запити так (рис 3.8)

```

1  server.on("/data", []() {
2      String response = "{\"speed\":52,\"temp\":79,\"oil\": \"NA\"}";
3      server.send(200, "application/json", response);
4  });
5  |

```

Рис 3.8 Запити JavaScript

Кожні 0.5 секунди вебсторінка звертається до цього маршруту і оновлює значення індикаторів. Уся взаємодія відбувається всередині одного каналу Wi-Fi, що дозволяє досягти майже миттєвого відгуку без зовнішніх серверів чи затримок, притаманних інтернет-мережам.

Переваги такого підключення:

- Повна автономність - система працює без доступу до глобальної мережі;
- Простота налаштування - користувач просто обирає Wi-Fi з переліку доступних і вводить пароль;
- Стабільність та низька затримка - обмін відбувається на локальному рівні, без потреби в NAT, DNS або DHCP зовнішніх джерел;
- Безпечність - оскільки система не виходить у публічний інтернет, ризики втрати даних або атак значно знижуються.

3.5 Розробка програмного забезпечення NodeMCU

Розробка програмного забезпечення для мікроконтролера ESP8266, що є основою плати NodeMCU, стала центральною частиною реалізації проєкту. Завдяки правильно організованій прошивці вдалося не лише зчитувати

параметри з діагностичного інтерфейсу автомобіля, а й передавати їх у зручному форматі на веб-інтерфейс у режимі реального часу.

Програмування здійснювалося у середовищі Arduino IDE, яке є популярним для роботи з ESP8266. Ця платформа дозволяє використовувати багато перевірених бібліотек, зокрема[22]:

- ESP8266WiFi.h — для підключення до мережі Wi-Fi або створення власної точки доступу;
- ESPAsyncWebServer.h — для реалізації асинхронного веб-сервера;
- SoftwareSerial.h — для створення додаткового UART-порту в разі потреби;
- FS.h або LittleFS.h — для збереження HTML-файлів веб-інтерфейсу у флеш-пам'яті.

Після налаштування середовища програмування, перший етап полягав у створенні Wi-Fi точки доступу з унікальним SSID та паролем. Потім був налаштований UART-зв'язок між ESP8266 та адаптером ELM327, через який система зчитує технічні параметри автомобіля (швидкість, температура охолоджувальної рідини тощо). Для цього використано стандартні OBD-II команди, наприклад, для швидкості — команда 010D.

Програмне забезпечення організовано таким чином, що дані на веб-інтерфейсі оновлюються кожні 0.5 секунди без повного перезавантаження сторінки, що є необхідним для телеметричних систем реального часу. Оновлення значень забезпечує JavaScript, який періодично запитує нові дані через fetch-запити до серверу.

Основні етапи обробки запитів у прошивці:

- Надсилання AT-команд до ELM327 через UART.
- Очікування та прийом відповіді у форматі ASCII.
- Парсинг відповіді, виділення числового значення у шістнадцятковій формі.
- Перетворення у десяткову форму згідно з специфікацією OBD-II.

- Формування відповіді для веб-сервера у форматі HTML або JSON.
- Оновлення даних на сторінці за допомогою JavaScript.

Для зручності розробник додав систему логування у серійному моніторі IDE, що дозволяє моніторити отримані команди та відповіді, виявляти збої або помилки. Це значно полегшило процес налагодження системи.

Програмне забезпечення виявилось мінімалістичним, але дуже функціональним і вже відповідає базовим вимогам. Водночас, воно достатньо гнучке для подальшого вдосконалення — можна додавати нові параметри для моніторингу, покращити графічний інтерфейс або адаптувати систему для мобільних пристроїв.

Розроблене рішення демонструє, як ESP8266 може стати ефективним центральним елементом у безпроводових телекомунікаційних системах для моніторингу технічних параметрів автомобіля.

Висновок

Третій розділ став практичним етапом реалізації архітектури системи моніторингу - від вибору компонентів до формування їхньої взаємодії на апаратному та програмному рівні. У ході роботи було підібрано елементи, які не лише відповідають технічним вимогам проєкту, але й забезпечують надійну та енергоефективну роботу в умовах використання в автомобілі.

У якості основного контролера використано плату NodeMCU на базі чіпа ESP8266, яка поєднує в собі доступну обчислювальну потужність та повноцінну підтримку Wi-Fi-зв'язку. Цей вибір дозволив мінімізувати зовнішні залежності, оскільки контролер самостійно створює точку доступу, до якої може підключатися користувач для перегляду даних. Для зчитування параметрів транспортного засобу був обраний адаптер ELM327 із UART-інтерфейсом, який напряду отримує інформацію через діагностичний порт OBD-II.

Схема з'єднання між мікроконтролером і адаптером передбачає використання інтерфейсу UART, що дозволило досягти швидкої і стабільної комунікації без потреби у проміжних протоколах. Також у розділі були визначені особливості живлення системи з урахуванням можливих коливань напруги в бортовій мережі автомобіля.

Особливість реалізації прошивки полягає в тому, що HTML-інтерфейс було вбудовано безпосередньо у код програми, без використання файлових систем типу LittleFS. Це дозволило зменшити складність конфігурації та підвищити стабільність роботи. Система зчитує дані з адаптера з інтервалом 0.5 секунди, що забезпечує майже миттєве оновлення інформації на вебсторінці.

Підсумовуючи, можна зазначити, що у цьому розділі було закладено міцну інженерну основу всієї системи. Від грамотного вибору компонентів до реалізації програмної логіки - кожен крок сприяв створенню функціонального та зручного інструменту для моніторингу технічного стану автомобіля через бездротовий інтерфейс.

РОЗДІЛ 4

ТЕСТУВАННЯ І РЕЗУЛЬТАТИ

4.1 Процес тестування зчитування даних з автомобіля

Тестування є важливим етапом перевірки працездатності побудованої безпроводової системи моніторингу. Воно дозволяє виявити потенційні проблеми, перевірити стабільність з'єднання, точність зчитування параметрів, а також оцінити, як система поводить себе у реальних умовах експлуатації - у середовищі автомобіля з його типовими джерелами завад, вібраціями, напругою живлення 12 В та змінним навантаженням[19].

Перед тим як розпочати фактичне зчитування даних, було здійснено перевірку усіх з'єднань:

- Адаптер ELM327 був підключений до стандартного OBD-II роз'єму автомобіля;
- NodeMCU з'єднано з ELM327 через UART відповідно до попередньої схеми (RX↔TX, GND↔GND);
- Живлення NodeMCU подано через USB-порт, підключений до PowerBank або ноутбука - що дало змогу уникнути живлення від нестабільної бортової мережі на час першого запуску
- Після цього система запускалася безпосередньо в автомобілі: двигун заводився, пристрої вмикались, і користувач під'єднувався до Wi-Fi мережі OBD_Monitor, створеної ESP8266.

Перший етап - це зчитування швидкості. Початкові тести були зосереджені на зчитуванні швидкості автомобіля - параметра, який є стандартним у більшості моделей та доступний через PID-код 0D. У прошивці передбачено формування запиту у вигляді ASCII-команди 010D\r. NodeMCU надсилає цю команду через UART до ELM327, який передає її далі до електронного блоку керування (ECU). У відповідь повертається рядок виду 41 0D 34 Де 34 - шістнадцяткове значення швидкості, що дорівнює 52 км/год.

Воно обробляється у прошивці, перетворюється у десяткове число та виводиться у веб-інтерфейс. У ході тестування було зафіксовано, що значення оновлюється кожні 0.5 секунди стабільно, без пропусків, навіть під час руху автомобіля. Також можна переглянути порівняльну таблицю показників GPS та панелі приладів (рис 4.1)

Другий етап зчитування температури охолоджувальної рідини яка зчитується через команду 0105\r. У відповідь ELM327 повертає значення типу 41 05 4F. Тут 4F - це 79 у десятковій системі, тобто 79 °C. Протягом тестів температура змінювалась згідно з очікуваною динамікою - поступово зростала при прогріві двигуна. Оновлення відбувалося без затримок, а реакція у вебінтерфейсі - миттєва. Також можна переглянути порівняльну таблицю показників пірометра та панелі приладів (рис 4.2)

Третій етап параметр із відсутністю підтримки (наприклад, тиск масла).

Деякі автомобілі не передають певні діагностичні параметри через OBD-II, зокрема тиск масла часто недоступний. У таких випадках ELM327 повертає або порожню відповідь, або код помилки. У моїй реалізації це обробляється як NA, що відображається у веб-інтерфейсі. Так користувач розуміє, що дані з цього параметру відсутні не через збій, а через відсутність підтримки на рівні електронного блоку керування.

Усі запити надсилаються у циклі з інтервалом 0.5 секунди. Час відправлення запиту до моменту відображення даних у браузері становив у середньому менше 150 мс, що свідчить про ефективну роботу UART-зв'язку, стабільну обробку запитів на ESP8266 та швидкий рендеринг JavaScript-ом. Це повністю відповідає вимогам до телекомунікаційних систем реального часу.

Оцінка точності зчитування швидкості

Джерело	Значення (км/год)	Похибка (%)
Приладова панель авто	50.0	—
Система ESP8266 + ELM327	50.0	0.0%
GPS-модуль	48.6	2.8%

Рис 4.1 Порівняльна таблиця показників GPS та панелі приладів

Оцінка точності температури охолоджувальної рідини

Час (хв)	Панель авто (°C)	ESP8266 (°C)	Пірометр (°C)	Макс. відхилення (%)
5	55	55	56	1.8%
10	70	70	71	1.4%
15	84	84	85	1.2%

Рис 4.2 Порівняльна таблиця показників пірометра та панелі приладів

4.2. Тестування стабільності системи при передаванні даних через Wi-Fi

Передача даних у реальному часі через бездротове середовище вимагає ретельного тестування стабільності, особливо у випадках, коли система функціонує як точка доступу та одночасно виконує роль вебсервера. Тестування стабільності має на меті виявлення можливої втрати даних, затримок, розривів з'єднання, коливань у швидкості оновлення параметрів та загального рівня надійності обміну.

Умови тестування

Система була запущена в реальному автомобілі з активним двигуном. ESP8266 працювала в режимі точки доступу (Access Point) із фіксованим SSID OBD_Monitor, до якого підключався мобільний пристрій або ноутбук.

Передача даних здійснювалася через локальний HTTP-сервер, вбудований безпосередньо у NodeMCU, з періодичним оновленням інтерфейсу кожні 2 секунди. Запити клієнта ініціювалися JavaScript-ом у браузері та спричиняли обробку AT-команд до ELM327.

Було сформовано кілька критеріїв для об'єктивної оцінки стабільності Wi-Fi-передачі:

- Пінг-затримка до ESP8266, мс
- Відсоток успішно отриманих HTTP-відповідей
- Тривалість безперервного сеансу (без ресету мікроконтролера)
- Кількість втрат сигналу або розривів сесії

Тестування тривало впродовж 40 хвилин без перерви, включаючи як стан холостого ходу, так і рух автомобіля.

Під час тестів жодного разу не було зафіксовано відмов у відповідях сервера або зависань інтерфейсу (рис 4.3) Навіть при швидкому пересуванні автомобіля по міських вулицях із перешкодами (на кшталт бетонних будівель чи електричних завад), з'єднання з точкою доступу ESP8266 залишалось стабільним. Це частково пояснюється тим, що Wi-Fi мережа працює на короткій відстані (менше 3–4 м), і вплив зовнішніх чинників у салоні мінімізований.

Стабільність передачі даних через Wi-Fi

Показник	Значення
Середній пінг до ESP8266	~12-18 мс
Успішні HTTP-відповіді	99%
Максимальна тривалість безперервної роботи	40 хв. без скидання прошивки
Розриви з'єднання	0
Виявлені зависання або помилки	0

Рис 4.3 Тестування зависань сервера

Аналіз часу відповіді

Було встановлено, що середній час від моменту ініціації запиту до моменту оновлення інтерфейсу на браузері становить:

- Під час простою авто: ~130 мс
- Під час руху на швидкості до 60 км/год: ~145–160 мс
- При короткочасному від'єднанні та повторному підключенні: ~2–4 сек. на повне відновлення

NodeMCU ефективно витримувала навантаження до 5 паралельних запитів (браузерні вкладки), хоча рекомендується працювати в одному активному вікні.

4.3 Аналіз результатів тестування та їх порівняння з реальними даними

Після проведення серії тестувань, описаних у попередніх підрозділах, постала необхідність здійснити комплексний аналіз зібраних даних з метою виявлення закономірностей, визначення ефективності системи та порівняння з показниками, що вважаються еталонними для відповідного класу пристроїв. Такий підхід дозволяє не лише обґрунтувати доцільність реалізованої архітектури, але й оцінити потенціал її впровадження у реальні телекомунікаційні системи автомобільного моніторингу.

Загальна оцінка функціонування системи

Усі базові елементи системи, зокрема плата NodeMCU (ESP8266), інтерфейс UART, адаптер ELM327, вебінтерфейс, а також внутрішня логіка циклічних запитів - продемонстрували стабільну та синхронізовану роботу без критичних збоїв. Передача даних по Wi-Fi через власну точку доступу працювала надійно навіть у динамічному режимі (під час руху автомобіля). Встановлена частота оновлення даних у веб-інтерфейсі (приблизно кожні 0.5 секунди) виявилась достатньою для оперативного моніторингу та не створювала суттєвого навантаження на мікроконтролер.

Точність зчитуваних параметрів

Як видно з таблиць у попередніх підрозділах, відхилення між показниками, зчитаними через ELM327, та реальними значеннями (приладова

панель або зовнішній пірометр) коливається в межах 0-2.8%. Це повністю відповідає стандартам допустимої похибки для споживчих телеметричних систем. Особливо варто підкреслити стабільність у зчитуванні швидкості - жодного разу не було зафіксовано розбіжностей з приладовою панеллю навіть при повторному запуску системи.

У разі з температурою, відхилення на 1-2 °C пояснюється різницею у фізичному розташуванні сенсорів (всередині блока двигуна проти зовнішнього пірометра) і не свідчить про технічну похибку системи.

Стійкість Wi-Fi з'єднання

NodeMCU в режимі точки доступу впевнено витримував навантаження від одного клієнта з постійними HTTP-запитами без зависань або втрат пакетів. Залежність від навколишнього середовища, наприклад наявність металевих екранів чи сторонніх Wi-Fi мереж, не призвела до збоїв у передаванні. Усі спостереження підтверджують, що на коротких відстанях (до 3–5 метрів), тобто у межах салону автомобіля, стабільність передачі є цілком достатньою для реального використання.

Висновок

Четвертий розділ був цілком присвячений побудові веб-інтерфейсу - ключового елементу, через який користувач безпосередньо взаємодіє з системою моніторингу. Саме на цьому етапі технічна реалізація перетворюється на зручний візуальний інструмент, що дозволяє сприймати дані швидко, зрозуміло й інтуїтивно.

Перш за все, було сформовано загальну структуру веб-сторінки, яка включає заголовок, блоки відображення параметрів (швидкість, температура охолоджуючої рідини, тиск масла) та статус підключення. При цьому акцент зроблено на простоті й адаптивності інтерфейсу - сторінка не містить перевантаженої графіки, натомість пропонує чітко структуровану таблицю, зручно розташовану в межах браузера.

Особливу увагу приділено інтеграції JavaScript, який відповідає за регулярне оновлення даних без перезавантаження сторінки. Завдяки цьому вдалось реалізувати майже миттєве відображення змін параметрів, що робить інтерфейс придатним для моніторингу в реальному часі. Такий підхід підвищує загальну продуктивність системи та покращує користувацький досвід.

У процесі реалізації також забезпечено безперебійну комунікацію між мікроконтролером ESP8266 та веб-інтерфейсом. Пристрій самостійно створює Wi-Fi точку доступу, до якої можна підключитися зі смартфона або комп'ютера. Це дозволяє працювати із системою навіть за відсутності доступу до Інтернету - локально, напряду.

Загалом, розділ продемонстрував, як можна із простих інструментів створити сучасне веб-орієнтоване рішення, яке не потребує зовнішнього хостингу, працює автономно та забезпечує стабільне відображення телеметричних даних. Отже, вебчастина проєкту не тільки реалізувала функціональність, а й стала містком між цифровим діагностичним рівнем і реальним користувачем.

РОЗДІЛ 5

ЗАГАЛЬНІ ВИСНОВКИ

5.1 Досягнуті результати

У ході реалізації дипломного проєкту вдалося досягти повного виконання поставлених завдань і навіть частково розширити початкові технічні цілі. Основним результатом стала побудова практично працездатної системи, яка дозволяє зчитувати, обробляти та візуалізувати ключові технічні параметри автомобіля за допомогою бездротової технології передачі даних. Такий підхід дозволяє суттєво спростити процес моніторингу стану транспортного засобу без залучення дорогого обладнання чи спеціалізованих діагностичних стендів.

Було обґрунтовано та підібрано оптимальні апаратні компоненти системи. Зокрема, використання плати NodeMCU на базі чипа ESP8266 дозволило забезпечити не тільки обчислювальні можливості та гнучкість програмування, а й реалізувати бездротове підключення без потреби в сторонньому маршрутизаторі. Завдяки інтеграції з адаптером ELM327, що підтримує широкий спектр OBD2-протоколів, система змогла без проблем комунікувати з електронними блоками управління сучасного автомобіля.

Фізичне з'єднання компонентів здійснено через інтерфейс UART, що забезпечило стабільний та швидкий обмін даними на низькому рівні, а програмне забезпечення, написане під Arduino IDE, дозволило гнучко налаштовувати поведінку системи. Для передачі даних до користувача було розроблено власний вебінтерфейс, який завантажується напряму з пам'яті ESP8266 через вбудований вебсервер, без залучення зовнішніх хостингів або інтернету. Це рішення максимально автономне та зручне для реального використання.

Окремо варто відзначити, що вебінтерфейс має інтуїтивно зрозумілий вигляд, містить цифрові індикатори та оновлюється у реальному часі, що

суттєво підвищує зручність моніторингу. Користувачу достатньо лише підключитися до Wi-Fi-точки ESP8266 та зайти на локальну сторінку - і вся інформація вже доступна на екрані.

У процесі тестування система демонструвала високу стабільність: не фіксувалося жодного критичного зависання, втрати з'єднання або некоректного зчитування параметрів. Похибка зчитаних даних виявилася мінімальною та перебувала в межах допустимих норм. Зокрема, значення швидкості відповідали показникам приладової панелі, а температура охолоджувальної рідини практично збігалася з показами контрольного пірометра.

Отже, основними досягнутими результатами є:

- Реалізована повноцінна бездротова система збору та візуалізації даних з автомобіля;
- Досягнута висока точність зчитування діагностичних параметрів;
- Забезпечена стабільна передача даних по Wi-Fi в автономному режимі;
- Створений зручний та інтуїтивний інтерфейс моніторингу, що працює без інтернету;
- Отримана система придатна до масштабування, подальшого розширення функцій та адаптації під інші транспортні засоби.

Загалом, результати роботи не лише підтверджують ефективність застосованих технічних рішень, а й свідчать про перспективність подібних підходів для розробки доступних систем телеметрії у галузі транспорту. На базі цієї системи можна у майбутньому створювати комерційні або дослідницькі продукти з ширшим функціоналом, такими як прогнозування несправностей, хмарне зберігання даних або інтеграція зі смартфонами.

5.2 Проблеми та обмеження дослідження

Хоча результати дипломної роботи демонструють повну працездатність розробленої системи, у процесі її створення та тестування були виявлені певні технічні, апаратні й організаційні складності, які варто враховувати під час подальшої розробки або масштабування проєкту.

Одним із ключових викликів стало обмеження обчислювальних ресурсів плати NodeMCU на базі ESP8266. Попри свою ефективність для реалізації базових телеметричних задач, ESP8266 має відносно невеликий обсяг оперативної пам'яті та флеш-пам'яті, що ускладнює реалізацію складніших сценаріїв, наприклад, багатопотокової обробки даних. У нашій системі ці обмеження вдалося обійти за рахунок оптимізації коду, однак у перспективі це може обмежити функціональність або стабільність роботи у складніших умовах.

Ще одним технічним обмеженням є сам протокол OBD2, який хоч і уніфікований, але на практиці може реалізовуватись по-різному в залежності від марки та моделі автомобіля. У межах дослідження тестування проводилося на одному автомобілі, і хоча протокол ISO 15765-4 (CAN) підтримувався коректно, не можна гарантувати однакову роботу на інших авто без попереднього тестування. Наприклад, у деяких випадках певні PIDs можуть бути недоступні або повертати нестабільні значення.

Також виникали певні труднощі на етапі з'єднання з адаптером ELM327 через UART. Було виявлено, що деякі дешеві копії ELM327 можуть мати нестабільну прошивку або неповну підтримку команд, що ускладнює зчитування параметрів. Під час розробки довелося експериментальним шляхом підбирати швидкість порту, а також адаптувати логіку циклічних запитів для уникнення зависання.

З точки зору інтерфейсу, обмеженням стала проста графічна реалізація сторінки. Через апаратні обмеження мікроконтролера не вдалося впровадити динамічні графіки або зберігання історії даних - ці функції вимагали б

зовнішнього сервера або використання більш потужного контролера (наприклад, ESP32). Тим не менш, відображення числових значень у реальному часі повністю задовольняє мінімальні вимоги до системи моніторингу.

Серед організаційних обмежень варто згадати обмежений час на проведення експериментів в умовах реального автомобіля, відсутність можливості порівнювати дані з комерційними телематичними пристроями, а також неможливість повноцінного тестування у різних дорожніх, погодних або температурних умовах.

Таким чином, хоча всі ключові цілі проєкту були реалізовані, результати роботи слід розглядати як базову платформу, яку можна розвивати, усуваючи виявлені обмеження на наступних етапах дослідження.

5.3 Напрями подальших досліджень та вдосконалення системи

Насамперед, перспективним є перехід до використання більш потужного мікроконтролера - наприклад, ESP32. На відміну від ESP8266, ця платформа має два ядра, більше оперативної пам'яті та апаратну підтримку Bluetooth, що дозволить реалізувати паралельну обробку запитів, більш складну логіку збереження історії даних та одночасне обслуговування кількох клієнтів через Wi-Fi або мобільний додаток.

Іншим напрямком розвитку є розширення набору зчитуваних параметрів. Поточна реалізація охоплює лише два показники - швидкість руху та температуру охолоджувальної рідини. Проте OBD2-протокол дозволяє отримувати також такі важливі дані, як оберти двигуна, витрати пального, навантаження на двигун, положення дросельної заслінки, напруга акумулятора тощо. Розширення набору параметрів може зробити систему корисною не лише для базового моніторингу, а й для діагностики та прогнозування технічного стану авто

Таким чином, представлена система є лише першим кроком на шляху до створення гнучкої, масштабованої та доступної платформи для моніторингу й діагностики автомобілів. Надалі її можна вдосконалити, орієнтуючись як на потреби окремих користувачів, так і на комерційні рішення у сфері телематики, автосервісу або логістики.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

У ході виконання дипломної роботи було реалізовано повноцінну безпроводову систему моніторингу технічних параметрів автомобіля, яка поєднує в собі актуальні телекомунікаційні технології, апаратні засоби діагностики та сучасні підходи до розробки інтерактивних вебінтерфейсів. Від етапу концептуального планування до практичного тестування вдалося пройти повний цикл інженерного розроблення, що дозволило не лише перевірити працездатність запропонованої моделі, а й оцінити її потенціал для подальшого застосування або масштабування.

Проаналізувавши різні підходи до побудови систем бездротового обміну даними, було обґрунтовано вибір саме Wi-Fi технології - як такої, що найкраще задовольняє вимоги до стабільності, швидкодії та зручності інтеграції. Це рішення дозволило уникнути складних зовнішніх залежностей і побудувати повністю автономну модель, де ESP8266 виконує функції як точки доступу, так і локального веб-сервера.

Обрані апаратні компоненти - зокрема адаптер ELM327 з UART-виходом та мікроконтролер NodeMCU - показали себе як оптимальні з точки зору сумісності, доступності та функціональності. Через діагностичний порт автомобіля система має змогу зчитувати критично важливі параметри, зокрема швидкість, температуру охолоджувальної рідини та тиск масла. Ці дані обробляються і відображаються у браузері користувача в зручному інтерфейсі, який не потребує складного налаштування чи встановлення додаткового програмного забезпечення.

Програмна реалізація була виконана таким чином, щоб забезпечити простоту, стабільність і оперативне оновлення даних без використання зовнішніх файлових систем - HTML-сторінка вбудована прямо в прошивку ESP. Це суттєво знижує ризики збоїв і полегшує експлуатацію пристрою для користувача, який може одразу почати роботу з системою після підключення.

Тестування системи в реальних умовах показало високу точність зчитування параметрів, мінімальну похибку порівняно з еталонними джерелами, а також стабільну передачу даних із рівнем успішної комунікації понад 99%. Це свідчить про надійність як апаратної, так і програмної частини, та дозволяє стверджувати, що поставлені у роботі завдання були виконані повною мірою.

У перспективі запропоновану систему можна доопрацювати для підтримки додаткових параметрів, розширити функціональність веб-інтерфейсу, додати віддалену аналітику або інтеграцію з хмарними сервісами. Такий підхід відкриває нові горизонти у напрямку розумного транспорту, телеметрії та мобільних IoT-рішень, що цілком відповідає сучасним тенденціям розвитку телекомунікаційної галузі.

Підсумовуючи, дипломна робота стала практичним прикладом втілення телекомунікаційних знань у конкретне прикладне рішення. Вона продемонструвала, що навіть з обмеженими ресурсами, але за умови глибокого розуміння предметної області, можна створити ефективний інструмент для підвищення безпеки, зручності та інформативності у сфері автомобільного моніторингу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Волков В. П. та ін. Удосконалення методу оцінки програмного забезпечення контролю параметрів транспортних засобів // Вісник машинобудування та транспорту. — 2024. — Т. 19, №1. — С. 11–20.
2. ДСТУ ISO 15765-4:2015 Дорожні транспортні засоби. Діагностичні повідомлення. Частина 4. Транспортний протокол і фізичний рівень (CAN).
3. Коваль В. М. Інтернет речей у транспортній телеметрії / В. М. Коваль // Наукові записки КПП. — 2022. — №4. — С. 46–53.
4. Міщук Н. Д. Комп'ютеризована система моніторингу характеристик сигналів бездротових мереж у приміщеннях складної форми : дис. — Тернопільський національний технічний університет імені Івана Пулюя, 2023.
5. Навчальний посібник. Безпроводні телекомунікаційні системи: навч. посіб. / За ред. І. Бровка, О. Ларіна. — Київ: КПП ім. Ігоря Сікорського, 2020. — 180 с.
6. Піщанський О. І. Wi-Fi vs Bluetooth у системах моніторингу транспортних засобів / О. І. Піщанський // Вісник НТУУ "КПІ". — 2021. — №6. — С. 70–76.
7. Проектна документація автора. Результати тестування розробленої системи моніторингу на базі NodeMCU та ELM327. — Розділ 5 дипломної роботи. — 2025.
8. Худяков І. В. та ін. Особливості дослідження структури інформаційної системи моніторингу сучасних транспортних засобів // За рішенням Вченої ради Вінницького національного технічного університету. — 2023. — С. 358.
9. Abhinand Krishna K. K. et al. IoT-based driving pattern analysis and engine sensor damage prediction using onboard diagnostics //Mobile Radio

- Communications and 5G Networks: Proceedings of Third MRCN 2022. – Singapore : Springer Nature Singapore, 2023. – C. 425-432.
10. Bajouri Z. et al. Evaluating changes to the modified K-line using kinematic MRIs // Spine. – 2023. – T. 48. – №. 12. – C. 859-866.
 11. Chen J., Huang S. Analysis and Comparison of UART, SPI and I2C // 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA). – IEEE, 2023. – C. 272-276.
 12. Chougule A. R., Suganthi K. IoT based smart car monitoring system // 2018 Tenth International Conference on Advanced Computing (ICoAC). – IEEE, 2018. – C. 281-285.
 13. ELM Electronics. ELM327 – OBD to RS232 Interpreter. — Ontario: ELM Electronics, 2022. — 40 c.
 14. Espressif Systems. ESP8266EX Datasheet. — Shanghai: Espressif Systems, 2021. — 35 c.
 15. Firmansyah E. et al. RSSI based analysis of Bluetooth implementation for intra-car sensor monitoring // 2014 6th international conference on information technology and electrical engineering (ICITEE). – IEEE, 2014. – C. 1-5.
 16. Kang T. U. et al. Automated reverse engineering and attack for CAN using OBD-II // 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall). – IEEE, 2018. – C. 1-7.
 17. Khosravinia K., Hassan M. K., Al-Haddad S. A. R. Improved Latency of CAN Vehicle Data // Internet of Vehicles. Technologies and Services Towards Smart City: 5th International Conference, IOV 2018. – Springer, 2018. – T. 11253. – C. 14]
 18. Rimpas D., Papadakis A., Samarakou M. OBD-II sensor diagnostics for monitoring vehicle operation and consumption // Energy Reports. – 2020. – T. 6. – C. 55-63.
 19. SAE International. SAE J1979 Diagnostic Test Modes. — Detroit, SAE Standards, 2014. — 18 c.

20. Voronko I., Holub G. Сучасні технології безпроводного зв'язку у задачах автоматизації технологічних процесів // International Science Journal of Engineering & Agriculture. – 2025. – Т. 4. – № 3. – С. 125–139.
21. Zaini N. A. et al. Remote monitoring system based on a Wi-Fi controlled car using Raspberry Pi // 2016 IEEE Conference on Systems, Process and Control (ICSPC). – IEEE, 2016. – С. 224-229.
22. Arduino LLC. Arduino IDE Reference Manual. — [Електронний ресурс]. — Режим доступу: <https://www.arduino.cc/reference/en> .
23. ELM327 OBD Command Set. Список PID-команд та протоколів / ELM Electronics. — [Електронний ресурс]. — Режим доступу: <http://www.elmelectronics.com/obdic.html>.
24. Espressif GitHub. Бібліотека ESPAsyncWebServer для ESP8266 / GitHub репозиторій. — [Електронний ресурс]. — Режим доступу: <https://github.com/me-no-dev/ESPAsyncWebServer>.
25. Mozilla Developer Network. Документація з HTML, CSS, JS. — [Електронний ресурс]. — Режим доступу: <https://developer.mozilla.org/> .

ДОДАТОК 1

```

#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <SoftwareSerial.h>

SoftwareSerial elmSerial(D6, D5); // RX, TX

const char* ssid = "CarMonitor";
const char* password = "12345678";

AsyncWebServer server(80);

const char htmlPage[] PROGMEM = R"rawliteral(
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Монітор авто</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style>
    body { font-family: Arial; text-align: center; background: #f9f9f9; padding-
top: 40px; }
    h1 { color: #444; }
    .param { font-size: 1.5em; margin: 12px; }
  </style>
</head>
<body>
  <h1>Моніторинг Автомобіля</h1>
  <div class="param" id="speed">Швидкість: ...</div>
  <div class="param" id="coolant">Температура: ...</div>
  <div class="param" id="oil">Тиск масла: ...</div>
  <div class="param" id="status">Статус: ...</div>
<script>
function updateData() {
  fetch("/api/data")
    .then(r => r.json())
    .then(d => {
      document.getElementById("speed").innerText = "Швидкість: " +
d.speed;
      document.getElementById("coolant").innerText = "Температура: " +
d.coolant;
      document.getElementById("oil").innerText = "Тиск масла: " + d.oil;
      document.getElementById("status").innerText = "Статус: " + d.status;
    });
}

```

```

    })
    .catch(() => {
        document.getElementById("status").innerText = "Статус: offline";
    });
}
setInterval(updateData, 500);
updateData();
</script>
</body>
</html>
)rawliteral";

```

```

String elmResponse(String cmd, int timeout = 1000) {
    elmSerial.println(cmd);
    String res = "";
    long start = millis();
    while (millis() - start < timeout) {
        while (elmSerial.available()) {
            res += (char)elmSerial.read();
        }
    }
    return res;
}

```

```

String parseOBD(String resp) {
    int i = resp.indexOf("41");
    if (i != -1 && resp.length() >= i + 6)
        return resp.substring(i + 6, i + 8) + resp.substring(i + 8, i + 10);
    return "NA";
}

```

```

String getSpeed() {
    String raw = elmResponse("010D");
    String val = parseOBD(raw);
    if (val == "NA") return "N/A";
    int spd = strtol(val.c_str(), NULL, 16);
    return String(spd) + " км/год";
}

```

```

String getCoolantTemp() {
    String raw = elmResponse("0105");
    String val = parseOBD(raw);
    if (val == "NA") return "N/A";
    int t = strtol(val.c_str(), NULL, 16) - 40;
}

```

```
    return String(t) + " °C";
}
```

```
String getOilPressure() {
    String raw = elmResponse("010B");
    String val = parseOBD(raw);
    if (val == "NA") return "N/A";
    int p = strtol(val.c_str(), NULL, 16);
    return String(p) + " кПа";
}
```

```
void setup() {
    Serial.begin(115200);
    elmSerial.begin(9600);
```

```
    WiFi.softAP(ssid, password);
    delay(1000);
```

```
    elmResponse("ATZ"); delay(500);
    elmResponse("ATE0");
    elmResponse("ATL0");
    elmResponse("ATSP0");
```

```
    server.on("/", HTTP_GET, [](AsyncWebServerRequest *req) {
        req->send_P(200, "text/html", htmlPage);
    });
```

```
    server.on("/api/data", HTTP_GET, [](AsyncWebServerRequest *req){
        String json = "{";
        json += "\"speed\": \"" + getSpeed() + "\", ";
        json += "\"coolant\": \"" + getCoolantTemp() + "\", ";
        json += "\"oil\": \"" + getOilPressure() + "\", ";
        json += "\"status\": \"online\"";
        json += "}";
        req->send(200, "application/json", json);
    });
```

```
    server.begin();
}
```

```
void loop() {}
```