Міністерство освіти і науки України

Ніжинський державний університет імені Миколи Гоголя

Навчально-науковий інститут природничо-математичних, медико-біологічних наук та інформаційних технологій

Кафедра інформаційних технологій, фізико-математичних та економічних наук

Освітня програма: Комп’ютерні науки

Спеціальність: 122 Комп’ютерні науки

**КВАЛІФІКАЦІЙНІ РОБОТА**

на здобуття освітнього ступеня *бакалавр*

**Моделювання дистанційної діагностики i обслуговування для підтримки різних систем, включаючи транспорт**

студента **Балуби Юрія Ігоровича**

**Науковий керівник:**

Казачков Іван Васильович,

доктор технічних наук, професор

**Рецензент:**

Фетісов Валерій Сергійович,

кандидат економічних наук, доцент

**Допущено до захисту:** 2023 р.

Завідувач кафедри

проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Казачков І.В.

Ніжин – 2023

**АНОТАЦІЯ**

**Актуальність теми.** З появою нових технологій транспорту збільшується складність технічних систем, що вимагає нових інструментів для контролю та діагностики стану транспорту.

**Метою дослідження** є розробка та валідація прототипної системи з використанням теоретичних знань та практичного досвіду в цій області.

**Практична цінність** даної дипломної роботи полягає в моделюванні системи дистанційної діагностики та обслуговування для підтримки різних систем в тому числі і транспортних.

Дипломна робота складається з вступу, трьох розділів та висновків.

У першому розділі розглядається актуальність використання даних систем а також основні проблеми, з якими можуть зіткнутися різні компанії в процесі забезпечення ефективності та безпеки.

У другому розділі описано апаратну та програмну складову технології дистанційної діагностики.

У третьому розділі роботи буде описано розробку прототипної системи дистанційної діагностики та обслуговування

У висновках проаналізовані отримані результати роботи.

**Ключові слова:** дистанційна діагностика, обслуговування, прототип, транспортна система, сенсорні мережі, аналіз даних, віддалений доступ, ефективність, надійність, безпека

**ABSTRACT**

**Topic relevance**. With the emergence of new transportation technologies, the complexity of technical systems increases, requiring new tools for monitoring and diagnosing the state of transportation.

**The aim of the research** is to develop and validate a prototype system using theoretical knowledge and practical experience in this area.

**The practical value** of this thesis lies in the modeling of a remote diagnostic and service system to support various systems, including transportation.

The thesis consists of an introduction, three chapters, and conclusions.

In the first chapter, the relevance of using data systems and the main problems that various companies may face in ensuring efficiency and safety are discussed.

The second chapter describes the hardware and software components of remote diagnostic technology.

The third chapter of the work will describe the development of a prototype remote diagnostic and maintenance system.

In the conclusions, the obtained results of the work are analyzed.

**Keywords:** remote diagnostics, service, prototype, transportation system, sensor networks, data analysis, remote access, efficiency, reliability, safety.

ЗМІСТ

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ 4](#_Toc135869137)

[ВСТУП 5](#_Toc135869138)

[АКТУАЛЬНІСТЬ ВИКОРИСТАННЯ ТА ОСНОВНІ ПРОБЛЕМИ 6](#_Toc135869139)

[1.1 Перспективи використання: можливості та виклики 6](#_Toc135869140)

[1.2 Основні проблеми з якими можуть зіткнутися компанії 9](#_Toc135869141)

[1.3 Порівняння та аналіз використання технологій 10](#_Toc135869142)

[1.4 Аналіз використання дистанційної діагностики в різних країнах 15](#_Toc135869143)

[1.5 Висновки до розділу 1 18](#_Toc135869144)

[АПАРАТУРНА ТА ПРОГРАМНА СКЛАДОВА 19](#_Toc135869145)

[2.1 Сенсори, засоби зв'язку та збір даних 19](#_Toc135869147)

[2.2 Програмна складова: обробка даних та інструменти аналізу 21](#_Toc135869148)

[2.3 Принципи функціонування апаратної та програмної складової 23](#_Toc135869149)

[2.4 Приклади використання апаратної та програмної складової 25](#_Toc135869150)

[2.5 Робота апаратної та програмної складових: переваги та недоліки 27](#_Toc135869151)

[2.6 Висновок до розділу 2 29](#_Toc135869152)

[РОЗРОБКА ПРОТОТИПНОЇ СИСТЕМИ 30](#_Toc135869153)

[3.1 Концепт визначення вимог до систем під час їх створення 30](#_Toc135869155)

[3.2 Етапи розробки апаратної складової 31](#_Toc135869156)

[3.3 Розробка програмної складової 33](#_Toc135869157)

[3.4 Інтеграція апаратної та програмної складових 34](#_Toc135869158)

[3.5 Створення прототипної системи її тестування та удосконалення 36](#_Toc135869159)

[3.5.1 Розробка прототипної системи 37](#_Toc135869160)

[3.5.2 Тестування та вдосконалення прототипу 49](#_Toc135869161)

[3.6 Висновок до розділу 3 59](#_Toc135869162)

[ЗАГАЛЬНИЙ ВИСНОВОК 61](#_Toc135869163)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 63](#_Toc135869164)

[ДОДАТОК A 66](#_Toc135869165)

[ДОДАТОК Б 67](#_Toc135869166)

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

ТЗ – Технічне забезпечення.

.ui – це розширення файлу яке використовується для файлів інтерфейсу користувача у Framework’y PyQt5.

.py - це розширення файлу яке вказує на файл, що містить вихідний код на мові програмування Python.

PyQt5 - є набором інструментів для розробки графічних інтерфейсів користувача на основі Qt Framework у мові програмування Python.

QtDesigner - є графічним інструментом, який входить до складу Qt Framework і використовується для проектування інтерфейсів користувача.

Інтероперабельність - Інтероперабельність відноситься до здатності різних систем або компонентів взаємодіяти та співпрацювати один з одним.

OBD - OBD (On-Board Diagnostics) є системою діагностики, яка використовується у багатьох сучасних автомобілях.

# ВСТУП

За останні роки, потреба у подібних технологіях тільки збільшилася, оскільки урбанізація вносить свою лепту в популярність різних систем. Через збільшення різних технологій з’являється потреба у більш швидкому та ефективному ремонті де в свою чергу дистанційна діагностика виступає лідером в цьому питані. Адже вона має значні переваги порівняно з традиційними методами. Перш за все, вона дозволяє зменшити кількість відвідуваних місць розташування обладнання для діагностики та обслуговування, що може бути дуже важливо для систем, розташованих на великих відстанях або у важкодоступних місцях. Ну і крім того, за допомогою дистанційних технологій можна виявляти проблеми та здійснювати їх віддалене вирішення, що значно зменшує час, необхідний для їх виправлення.

Оскільки дистанційна діагностика має великий потенціал для поліпшення ефективності і зниження витрат на обслуговування різних систем є велика ймовірність, що функції дистанційної діагностики будуть поширюватись практично на всі області технологій. Наприклад вже зараз деякі виробники побутової техніки використовують дистанційну діагностику для забезпечення швидкого та ефективного обслуговування своїх пристроїв. Наприклад, за допомогою спеціального додатку можна дізнатися про стан пральної або посудомийної машини та відправити запит в технічну підтримку.

У рамках дослідження в дипломній роботі будуть розглянуті різні аспекти дистанційної діагностики та обслуговування різних систем. Аналіз методик та технологій, що використовуються для віддаленого контролю та діагностики, а також будуть розглянуті можливості їх оптимізації та вдосконалення. Дипломна робота буде містити практичну частину, в рамках якої буде розроблена прототипна система дистанційної діагностики. Результати дослідження можуть бути корисними для інженерів-проектувальників, розробників програмного забезпечення та інших технічних спеціалістів.

# АКТУАЛЬНІСТЬ ВИКОРИСТАННЯ ТА ОСНОВНІ ПРОБЛЕМИ

## Перспективи використання: можливості та виклики

Використання дистанційної діагностики та обслуговування в різних системах має великий потенціал для покращення ефективності та зниження витрат на обслуговування. Нові технології, такі як машинне навчання та штучний інтелект, можуть допомогти забезпечити більш точну діагностику та передбачуваність проблем з автомобілями.

Одним з перспективних напрямів є використання дистанційного обслуговування в міському транспорті, яке може допомогти покращити ефективність і знизити витрати на обслуговування автобусів, трамваїв та метро. Це можливо завдяки системам моніторингу та діагностики, які дозволяють виявляти проблеми на ранніх стадіях та вчасно їх вирішувати.

Дистанційне обслуговування може бути корисним і в галузі логістики та доставки, забезпечуючи більш точний моніторинг стану транспортних засобів, оптимізацію маршрутів та планування обслуговування транспорту.

Однак, ці технології також стикається з викликами. Одним з них є забезпечення безпеки даних, особливо з урахуванням того, що технології стають все більш залежними від збору та обробки персональної інформації. Також потрібно вирішувати питання щодо інтероперабельності та стандартизації між різними виробниками, а також питання щодо надійності та доступності технологій у віддалених регіонах.

Усі ці виклики потребують уваги та вирішення, але перспективи використання дистанційної діагностики та обслуговування залишаються дуже обнадійливими. Такі технології можуть значно поліпшити ефективність обслуговування різних систем, зменшити витрати на підтримку транспортних мереж та покращити безпеку на дорогах.

Наприклад, дистанційне діагностування може допомогти зменшити кількість аварій та неполадок на дорогах, оскільки відповідні системи можуть виявляти проблеми з автомобілем задовго до того, як вони призведуть до серйозних наслідків. Крім того, такі технології можуть забезпечувати оперативну допомогу водіям у разі неполадок або аварій, що також може допомогти зменшити кількість інцидентів на дорогах.

Також важливо враховувати, що дистанційне обслуговування може допомогти зменшити витрати на підтримку транспортних мереж, оскільки такі системи можуть забезпечувати більш ефективне використання ресурсів та планування ремонтів та обслуговування на основі даних з сенсорів та моніторингових систем.

Іншим важливим аспектом використання дистанційної діагностики та обслуговування є забезпечення більш точного та швидкого реагування на зміни в транспортних потоках та інші події на дорозі. Такі системи можуть забезпечувати розумне управління транспортними мережами, що може допомогти зменшити затори та покращити ефективність транспортних систем.

Дистанційна діагностика та обслуговування можуть також допомогти зменшити витрати на паливо, оскільки можуть забезпечувати більш точне керування двигуном та іншими системами автомобіля. Крім того, ці технології можуть зробити автомобілі більш екологічно чистими, зменшивши викиди в атмосферу.

Також, дистанційна діагностика та обслуговування може допомогти забезпечити більш швидкий та якісний сервіс для автомобілів, що важливо для підтримки максимальної продуктивності транспортної системи. Завдяки цьому можна забезпечити більш якісний та швидкий ремонт автомобілів, що дозволить швидше відновлювати роботу транспортної системи після аварій та поломок.

Тому, дистанційна діагностика та обслуговування має великий потенціал для вдосконалення різних систем в тому числі і транспортних та забезпечення більшої безпеки на дорозі. Але, щоб реалізувати цей потенціал, потрібно вирішити виклики та проблеми, пов'язані з безпекою даних, стандартизацією та регулюванням, а також забезпечити належний рівень освіти та технічних знань сервісних працівників. Крім того, потрібно забезпечити відповідну інфраструктуру та доступ до мережі Інтернет, особливо в тих регіонах, де інтернет-підключення є обмеженим. Для ефективного використання цих технологій також необхідно вирішити проблеми з недостатньою автоматизацією та відсутністю єдиної платформи для збору та аналізу даних. З цими викликами можна успішно боротися за допомогою узгоджених зусиль виробників, постачальників послуг та органів державного регулювання. Якщо вдало вирішити ці проблеми, то дистанційна діагностика та обслуговування можуть значно покращити ефективність систем та зменшити витрати на обслуговування.

## Основні проблеми з якими можуть зіткнутися компанії

В процесі використання дистанційної діагностики та обслуговування систем можуть виникнути різноманітні проблеми, які можуть вплинути на ефективність та безпеку функціонування. Нижче перераховано деякі з основних проблем, з якими можуть зіткнутися компанії:

* Недостатня надійність технології: Оскільки дистанційна діагностика та обслуговування передбачає використання різних електронних датчиків, засобів зв'язку та програмного забезпечення, можуть виникнути проблеми з їх надійністю та стабільністю роботи, що може призвести до помилкових діагнозів та збоїв в системі.
* Високі витрати на обладнання та програмне забезпечення: Для використання дистанційної діагностики потрібно встановити спеціальне обладнання та програмне забезпечення, що може вимагати значних витрат.
* Недостатня кваліфікація персоналу: Для ефективного використання необхідний персонал, який має високу кваліфікацію та досвід роботи з цією технологією. Недостатня кваліфікація персоналу може призвести до помилок в роботі системи та значно знизити її ефективність.
* Відсутність стандартів та нормативних документів: Існування стандартів та нормативних документів є дуже важливим у забезпеченні якості та безпеки використання дистанційної діагностики та обслуговування. Однак, на сьогоднішній день таких стандартів не існує, що може призвести до використання неякісних технологій та зниження рівня безпеки.
* Проблеми зі зберіганням та захистом даних: Використання дистанційної діагностики та обслуговування різних систем пов'язане зі збором та обробкою великої кількості даних. Це створює проблеми зі зберіганням та захистом цих даних, оскільки вони можуть бути підвернені ризику крадіжки, витоку чи несанкціонованого доступу.
* Недостатня інфраструктура: Використання дистанційної діагностики та обслуговування вимагає наявності відповідної інфраструктури, такої як мережа зв'язку та програмне забезпечення. Відсутність необхідної інфраструктури може суттєво обмежити можливості використання цієї технології.

## Порівняння та аналіз використання технологій

У цьому розділі розглянемо різні технології дистанційної діагностики та обслуговування для різних видів систем. Проведемо порівняння та аналіз кожної з технологій.

Одна з найбільш поширених технологій дистанційної діагностики та обслуговування в автомобільній промисловості - це технологія On-Board Diagnostics (OBD). Ця технологія дозволяє зчитувати та аналізувати дані, що надходять з різних датчиків та систем автомобіля, та виявляти можливі проблеми. Однак, OBD має свої обмеження, зокрема, не може виявляти проблеми, які пов'язані з електричними схемами та електронними компонентами.

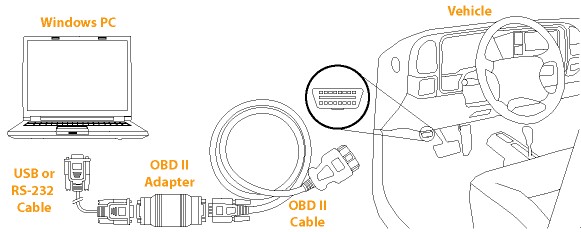


Рисунок 1.1 – Принцип роботи OBD

Інша технологія - це дистанційна діагностика та обслуговування для вантажних автомобілів та автобусів. Вона включає в себе використання моніторингу даних з сенсорів та інших пристроїв на транспортному засобі, а також використання програмного забезпечення для аналізу цих даних. Ця технологія дозволяє відстежувати роботу двигуна, споживання палива, знос деталей та інші параметри. Вона також дозволяє виявляти можливі проблеми та забезпечувати їх швидке усунення.

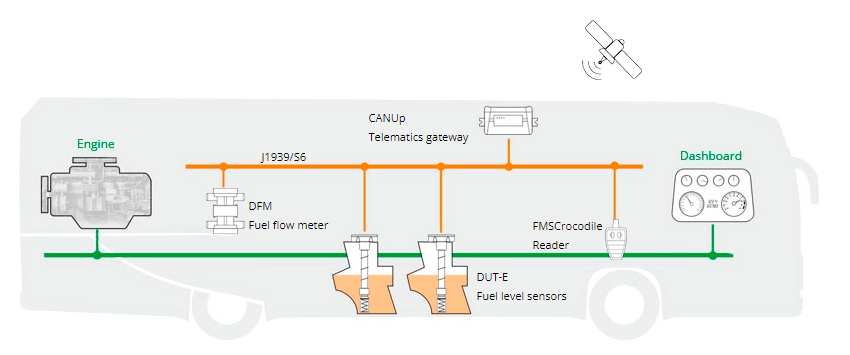


Рисунок 1.2 – Принцип роботи дистанційної діагностики на прикладі автобуса

Ще одна технологія - це дистанційна діагностика та обслуговування для залізничних систем та метрополітенів. Вона включає в себе використання різних пристроїв, таких як датчики вібрації, температурні датчики та інші, для моніторингу стану коліс, моторів та іншого обладнання, що використовується в залізничному транспорті.

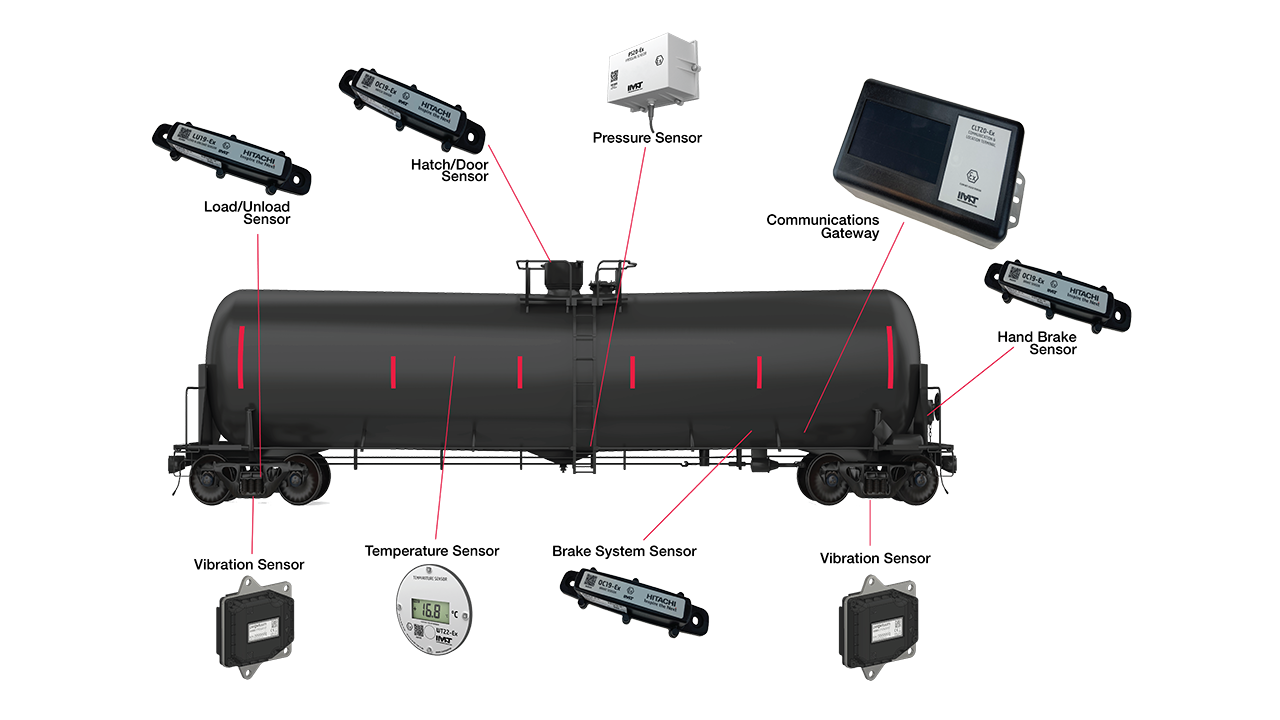


Рисунок 1.3 – Елементи дистанційної діагностики в залізничному транспорті

Однак, слід зазначити, що існують також певні проблеми які виклики при використанні дистанційної діагностики та обслуговування для залізничних та метрополітенів. Наприклад, вимірювання вібрації та температури може бути складним та не точним в разі, коли обладнання має високу швидкість руху.

Іншим прикладом технології дистанційної діагностики та обслуговування є її використання для морського транспорту. В цьому випадку використовуються датчики для моніторингу різних параметрів, таких як тиск, температура, вібрація, що дозволяє визначати стан двигуна, пропелера, системи охолодження та інших систем. Таким чином, можна забезпечити вчасне виявлення проблем та їх усунення ще до того, як вони приведуть до відмови або аварії.

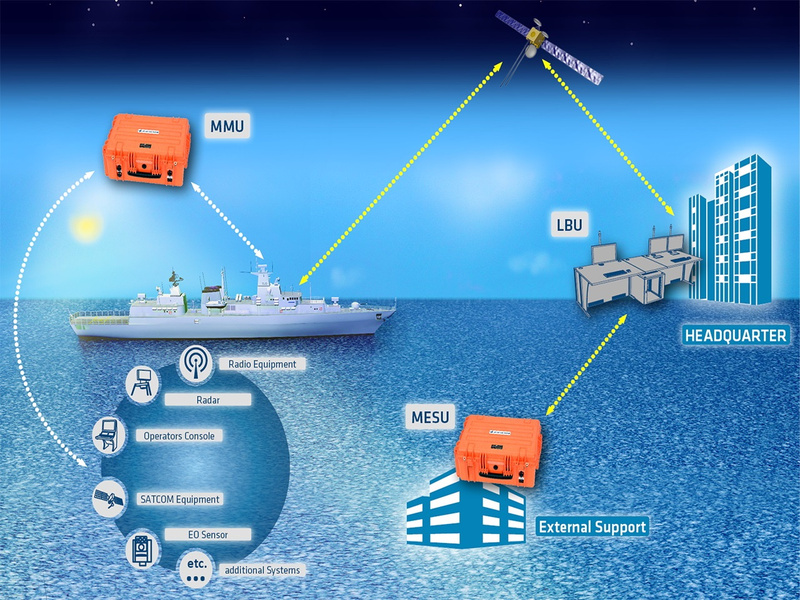


Рисунок 1.4 – Робота дистанційної діагностики на морському транспорті

Також є приклад використання дистанційної діагностики в повітряному транспорті. Тут як і попередніх прикладах використовуються датчики для моніторингу різних параметрів, таких як тиск, температура, швидкість, що дозволяє визначати стан двигуна, крил, системи керування та інших систем. Присутня функція передачі даних в реальному часі, що дозволяє отримувати дані з борту літака та відстежувати їхній стан з будь-якої точки світу.

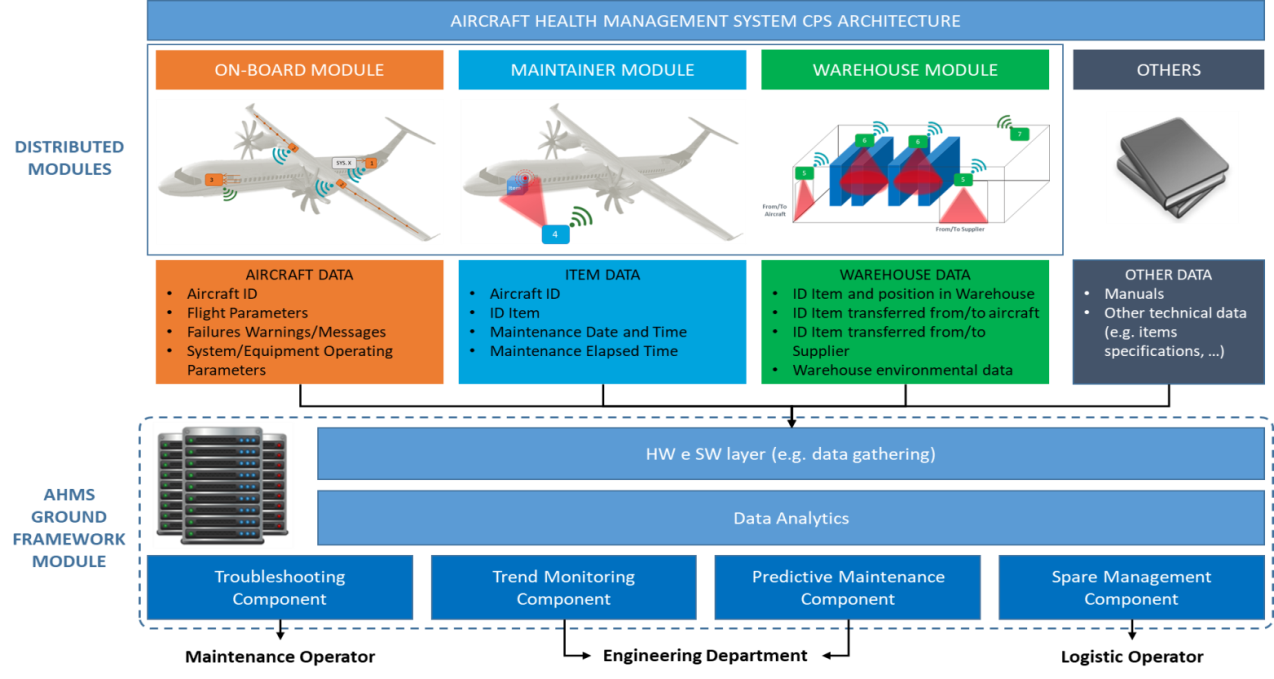


Рисунок 1.5 – Принцип роботи дистанційної діагностики на повітряному судні

Слід згадати і про будівельну галузь, тут ми можемо використовувати дистанційну діагностику та обслуговування зчитуючи дані з датчиків різних систем котрі залучені до будівельної галузі. За допомогою подібної діагностики ми можемо виявляти потенційні проблеми та уникати аварій. Крім того, ми робимо наші системи більш ефективнішими в роботі, що призводить до зниження витрат та підвищення комфорту користувачів.

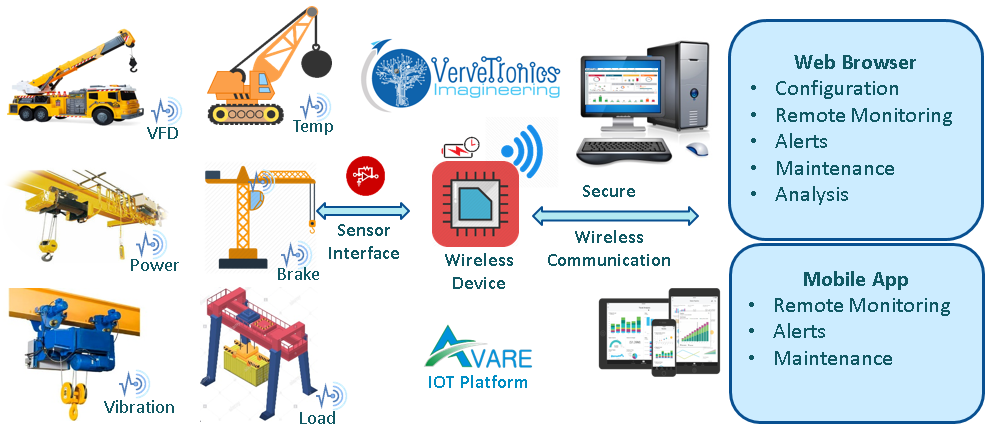


Рисунок 1.6 – Дистанційний моніторинг будівельних систем

## Аналіз використання дистанційної діагностики в різних країнах

Розділ про аналіз використання дистанційної діагностики та обслуговування в різних країнах світу може бути корисним для з'ясування ефективності та результативності цієї технології в різних контекстах.

Наприклад, можна проаналізувати досвід використання дистанційної діагностики та обслуговування в Японії, де ця технологія є широко поширеною і дозволяє підтримувати ефективність транспортної системи. Також можна розглянути досвід використання дистанційної діагностики та обслуговування в США, де ця технологія не є менш популярною ніж в Японії

Японія має розвинену автомобільну промисловість та є однією з провідних країн у використанні дистанційної діагностики та обслуговування автомобілів. Вона вважається піонером у використанні цих технологій, що свідчить про її професіоналізм та великий розвиток в цьому напрямі.

Один з прикладів використання дистанційної діагностики в Японії є програма "NissanConnect", яка надає власникам автомобілів марки Nissan можливість дистанційно контролювати стан своєї машини, здійснювати замовлення сервісного обслуговування та звертатися до підтримки в режимі реального часу.

Також в Японії існує Honda Link - це система, що використовується компанією Honda для забезпечення дистанційної діагностики та зв'язку з власниками їхніх автомобілів. За допомогою Honda Link, власники можуть переглядати інформацію про стан автомобіля, включаючи рівень палива, температуру двигуна, кількість пройдених кілометрів, інформацію про заміну масла та інші дані. Крім того, система може повідомляти власників про несправності або необхідність технічного обслуговування. Окрім діагностики та моніторингу стану автомобіля, Honda Link також надає можливість користувачам віддалено керувати деякими функціями автомобіля, такими як замикання та розмикання дверей, запуск двигуна з віддаленої точки або встановлення часу запуску автомобіля.

Ще один приклад - це система Mitsubishi Remote Control, яка дозволяє власникам автомобілів Mitsubishi з мобільних пристроїв віддалено контролювати підігрів двигуна та інші функції автомобіля.

Загалом, досвід використання дистанційної діагностики та обслуговування в Японії є досить успішним. Японські автовиробники впроваджують ці технології для поліпшення якості обслуговування, зменшення часу ремонту та зниження витрат на технічне обслуговування.

У США дана ніша стає все більш популярною, особливо серед власників електромобілів та гібридних автомобілів. Одним з провідних постачальників цих послуг є компанія Tesla, яка використовує технології, які дозволяють діагностувати та вирішувати проблеми з автомобілями віддалено. Зокрема, Tesla здатна віддалено проводити оновлення програмного забезпечення автомобілів, а також відстежувати деякі параметри їх роботи, що дозволяє вчасно виявляти та усувати можливі проблеми.

Крім Tesla, інші автовиробники також пропонують дистанційні сервіси для своїх автомобілів. Наприклад, General Motors має систему OnStar, яка дозволяє власникам автомобілів звернутися за допомогою в разі неполадок або аварій. Більшість нових автомобілів також оснащені датчиками та системами, які дозволяють віддалено контролювати рівень палива, тиск у шинах, рівень масла та інші параметри роботи автомобіля. Крім того, вона надає можливість дистанційно блокувати автомобіль в разі його крадіжки.

Іншим прикладом є система FordPass, яка використовується компанією Ford. Ця система надає можливість дистанційного контролю над станом автомобіля, запуску та зупинки двигуна, замовлення запчастин та ремонту, а також дистанційного блокування та розблокування дверей.

Однак, деякі власники автомобілів можуть скептично відноситися до дистанційної взаємодії зі своїм авто, через страх щодо конфіденційності персональних даних, таких як місцезнаходження та інші дані про водіння. Тому важливо, щоб автомобільні компанії забезпечували найвищий рівень конфіденційності даних користувачів та застосовували найкращі практики в цьому напрямку. Наприклад, вони можуть застосовувати шифрування даних, контролювати доступ до інформації, додати вимикати всіх дистанційних функцій свого авто, якщо користувач не бажає передавати свої дані.

Потрібно врахувати, що основою розвитку цих технологій як правило є економічна та політична стабільність. Оскільки ми можемо проаналізувати країни, котрі знаходяться в стані постійної війни, мають слабку позицію на політичній арені, мають високий рівень корупції, низький рівень освіти тощо. То можемо побачити, що в таких країнах розвиток цих технологій або взагалі відсутній, або розвивається дуже та дуже повільно. Наприклад, нестабільність в Сирії чи Афганістані може ускладнювати впровадження дистанційних технологій в автомобільну галузь в цих країнах.

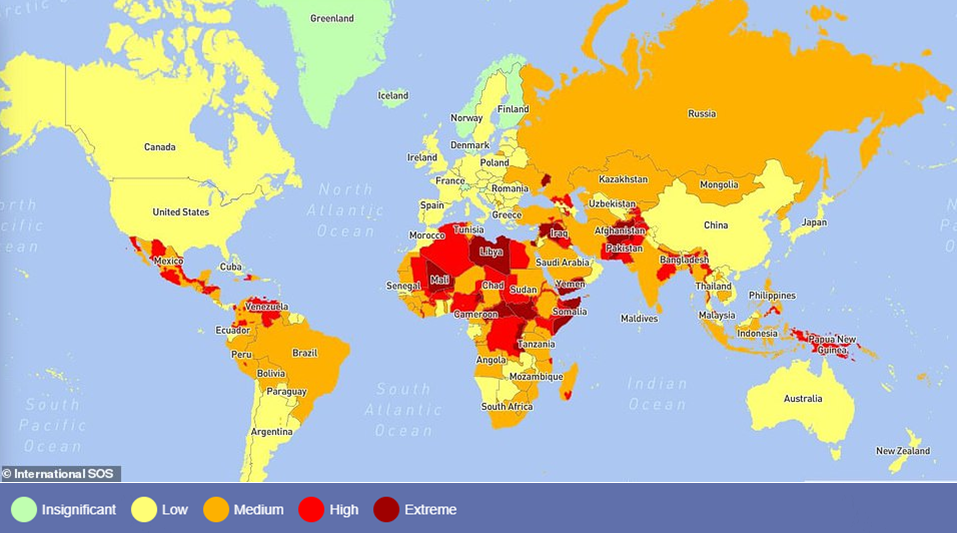


Рисунок 1.7 – статистика “The most dangerous countries for 2021”

В країнах з високим рівнем економічного розвитку та стабільним політичним становищем, таких як Японія, США, Італія, Німеччина, розвиток дистанційних технологій може прискорюватись та мати більш успішну реалізацію.

Крім того, на розвиток цих технологій може впливати також культурний та соціальний контекст країни. Наприклад, у країнах з високим рівнем індивідуалізму може бути більше попиту на технології, які забезпечують більшу автономію та контроль над процесами, тоді як у країнах з вищим рівнем колективізму може бути більша популярність технологій, що базуються на колективному підході. Крім того, важливим фактором є наявність інфраструктури та технічних засобів, що дозволяють використовувати ці технології, таких як швидкі інтернет-з'єднання, мобільні пристрої тощо. Таким чином, розвиток дистанційних технологій може залежати від багатьох факторів, які варіюються в залежності від країни та її контексту.

## Висновки до розділу 1

В даному розділі дипломної роботи розглянуті різні застосування технологій дистанційного обслуговування та діагностики. Були виділені основні переваги цих технологій, проведено їх аналіз використання, розглянуті перспективи використання, розглянуто, що впливає на їх розвиток. Були встановлені основні проблеми використання та впровадження цих технологій.

Із переваг можна відмітити:

* Заощадження часу
* Більш точний аналіз системи
* Моніторинг стану обладнання в реальному часі
* Ефективне виявлення проблеми

Із недоліків можна відмітити:

* Велику затрату коштів для виготовлення технологій
* Витіснення людей з їх робочих місць
* Недосконала система захисту даних
* Система не застрахована від збоїв
* Нездатність впровадити ці технології по всьому світу

Проаналізувавши використання в різних країнах світу, можна сказати, що технології дистанційного обслуговування, розвиваються в країнах зі стабільною економікою та стабільним політичним становищем. Також на сам розвиток технологій впливає і культурна основа країни.

# АПАРАТУРНА ТА ПРОГРАМНА СКЛАДОВА



## Сенсори, засоби зв'язку та збір даних

Апаратна складова технології дистанційної діагностики складається з різних компонентів, які забезпечують збір і передачу даних про стан транспортного засобу на віддалений сервер. Основні компоненти апаратної складової дистанційної діагностики включають наступні:

Сенсори - вимірюють фізичні параметри транспортного засобу, такі як температура, тиск, швидкість, вібрація, рівень рідин та інші. Сенсори можуть бути встановлені на різних частинах транспортного засобу, наприклад, на двигуні, колесах, паливних баках тощо.

Засоби зв'язку - передають дані з сенсорів на сервер, де проводиться аналіз і обробка даних. Засоби зв'язку можуть бути бездротовими, такими як мережі Wi-Fi, Bluetooth або GSM, або дротовими, такими як кабельна зв'язок.

Засоби збору даних - збирають дані з сенсорів та передають їх на сервер. Засоби збору даних можуть бути програмними засобами, що використовуються на транспортному засобі, або ж апаратними засобами, такими як додатковий блок збору даних.

Крім основних компонентів, в апаратну складову дистанційної діагностики також можуть входити елементи живлення, такі як батареї або акумулятори, або ж системи контролю та управління, які дозволяють віддалено управляти транспортним засобом.

Узагальнюючи, апаратна складова технологій дистанційної діагностики забезпечує збір даних про стан транспортного засобу, їх передачу на сервер та подальшу обробку. Вона може містити різні компоненти, такі як сенсори, засоби з'язку, контролери, комп'ютери та інше обладнання, залежно від конкретної системи та її функціональних вимог.

Основною функцією апаратної складової є збір даних про стан транспортного засобу. Для цього використовуються різноманітні сенсори, які можуть вимірювати такі параметри, як тиск, температура, вібрація, швидкість, знос деталей тощо. Ці сенсори можуть бути встановлені на різних частинах транспортного засобу, наприклад, на двигуні, колесах, гальмах, системах паливоподачі, електронних блоках тощо.

Для передачі даних зі сенсорів до сервера використовуються різні засоби зв'язку, такі як мережі зв'язку GSM, 3G, 4G, Wi-Fi, Bluetooth тощо. Залежно від конкретних умов експлуатації, можуть бути використані різні комбінації засобів зв'язку.

Крім збору даних, апаратна складова технології дистанційної діагностики може також включати контролери та комп'ютери для обробки даних, а також інші компоненти, такі як системи живлення, системи охолодження, системи захисту від впливу навколишнього середовища тощо. В цілому, апаратна складова технології дистанційної діагностики є важливим елементом системи та забезпечує збір та передачу даних, їх обробку та виконання необхідних дій для забезпечення безпеки та ефективності транспорту.

## Програмна складова: алгоритми обробки даних та інструменти аналізу

Програмна складова технології дистанційної діагностики є не менш важливою, ніж апаратна. Вона включає в себе алгоритми обробки даних та інструменти аналізу, які допомагають зрозуміти стан транспортного засобу та виявляти можливі проблеми.

У дистанційній діагностиці та обслуговуванні використовуються різні алгоритми обробки даних, залежно від конкретної задачі та типу системи.

Основні алгоритми включають:

1. Аналіз аномалій: цей алгоритм використовується для виявлення незвичайних або аномальних значень в даних, що може свідчити про несправність або проблему в системі. Наприклад, відслідковування аномалій може допомогти виявити несправності в роботі автомобільного двигуна.
2. Кластеризація: цей алгоритм дозволяє групувати дані за спільними ознаками, що може допомогти відслідковувати стан обладнання або прогнозувати можливі поломки.
3. Класифікація: цей алгоритм використовується для визначення класу або категорії, до якої належить даний елемент. Наприклад, класифікація може допомогти виявити, чи є певний сигнал належним до певного типу обладнання.
4. Прогнозування: цей алгоритм використовується для прогнозування майбутніх значень на основі історичних даних. Наприклад, прогнозування може допомогти визначити, коли потрібно замінити певні деталі або провести технічне обслуговування.
5. Машинне навчання: цей алгоритм використовується для навчання системи на основі історичних даних та автоматичного виявлення паттернів та залежностей. Наприклад, машинне навчання може допомогти виявити проблеми, які раніше не були відомі або навіть неможливі для виявлення людиною.
6. Глибинний аналіз даних - цей алгоритм дозволяє проводити аналіз даних на глибинному рівні, що дозволяє зробити висновки про стан системи та її можливі проблеми, які не були виявлені при поверхневому аналізі.
7. Обробка зображень - цей алгоритм використовується для обробки відео та фото з системи та дозволяє зробити висновки про стан обладнання, яке може бути недоступне для інших методів діагностики.
8. Нейромережевий аналіз - цей алгоритм використовується для аналізу даних та визначення закономірностей в їх залежності, що дозволяє зробити висновки про причини виникнення проблем та розробити стратегію їх вирішення.

Для реалізації алгоритмів обробки даних та аналізу використовуються різні програмні інструменти. Наприклад, для аналізу стану двигуна автомобіля можуть використовуватися програмні продукти, такі як LabVIEW, MATLAB, або Python, які дозволяють візуалізувати та обробляти дані з сенсорів, проводити аналіз спектрів та інші дослідження.

Крім того, програмна складова може містити модулі для збору та обробки даних зі сенсорів, моделі падінь рівня ефективності, що дозволяють прогнозувати майбутні поломки, а також інші функції, які допомагають підтримувати транспортні засоби в робочому стані.

Програмна складова технології дистанційної діагностики є важливою складовою, яка допомагає зрозуміти стан транспортного засобу та виявляти можливі проблеми.

## Принципи функціонування апаратної та програмної складової

Основні принципи функціонування апаратної та програмної складових технології дистанційної діагностики можуть бути описані наступним чином:

Апаратна складова:

Збір даних про стан транспортного засобу: сенсори, що встановлені на різних компонентах транспортного засобу, збирають дані про стан і параметри роботи цих компонентів. Це можуть бути, наприклад, датчики тиску масла, температури двигуна, швидкості транспортного засобу, а також датчики вібрації, які вказують на стан коліс та підвіски. Зібрані дані потім передаються на сервер для подальшої обробки.

Обробка даних: на сервері виконується аналіз зібраних даних з метою виявлення проблем або незвичайних станів. Для цього можуть використовуватися різні алгоритми обробки даних, такі як глибинний аналіз, машинне навчання або статистичні методи.

Повідомлення про стан: після обробки даних система дистанційної діагностики може згенерувати повідомлення про стан транспортного засобу, яке може бути відправлено оператору, водію або службі технічного обслуговування. Повідомлення може містити інформацію про те, який саме компонент потребує уваги, а також пропозиції щодо рекомендованих дій.

Передача даних: зібрані дані передаються на сервер або хмарну платформу за допомогою засобів зв'язку, таких як Wi-Fi, Bluetooth або мобільна мережа. Також можуть використовуватися спеціальні пристрої, які вбудовані в транспортний засіб, наприклад, вбудовані модулі зв'язку, які можуть відправляти дані через супутникову мережу. Ці дані можуть бути надіслані в реальному часі або збиратися на певному проміжку часу для подальшої обробки.

Зберігання даних: отримані дані зберігаються на сервері або хмарній платформі. Дані зберігаються на сервері або хмарній платформі, де вони можуть бути доступні для подальшої обробки та аналізу. Особливо важливо забезпечити безпеку та конфіденційність зберігання даних, оскільки вони можуть містити конфіденційну інформацію про технічний стан транспортного засобу та його експлуатацію.

Програмна складова:

Обробка даних: отримані дані піддаються аналізу та обробці за допомогою різних алгоритмів та методів, що дозволяють зрозуміти стан транспортного засобу та виявити можливі проблеми. Під час інтерпретації оброблених даних використовуються різні інструменти та методи, зокрема, машинне навчання та інші. На основі оброблених результатів приймаються рішення щодо подальшої обслуговування та ремонту об'єкта дистанційної діагностики.

Відображення результатів: після обробки даних результати відображаються на спеціальному інтерфейсі користувача в зручному для сприйняття вигляді. Дані про стан транспортного засобу можуть бути візуалізовані за допомогою спеціального програмного забезпечення, що дозволяє відслідковувати різні параметри, такі як швидкість, температура, тиск та інші. Крім того, можуть використовуватися алгоритми машинного навчання для виявлення відхилень від нормального функціонування, що дозволяє прогнозувати можливі поломки та уникнути їх.

Оповіщення про проблеми: якщо під час аналізу було виявлено можливі проблеми, програмна складова може оповістити користувача або технічний персонал про необхідність ремонту чи заміни деталей.

Таким чином, основні принципи функціонування апаратної та програмної складової технології дистанційної діагностики полягають у зборі даних про стан транспортного засобу, їх обробці та виконанні аналізу з метою виявлення проблем і повідомлення про них відповідним особам. Загалом апаратна та програмна складова доповнюють одна одну.

## Приклади використання апаратної та програмної складової

Існує багато прикладів використання апаратної та програмної складової дистанційної діагностики та обслуговування, ось декілька з них:

Дистанційна діагностика автомобілів: за допомогою датчиків, які знаходяться в різних частинах автомобіля, збираються дані про його роботу. Ці дані передаються на центральний сервер, де вони обробляються та аналізуються. На основі результатів аналізу, можна виявити можливі проблеми з автомобілем та запропонувати відповідні рішення щодо їх вирішення.

Моніторинг промислового обладнання: датчики вимірюють параметри промислового обладнання, такі як температура, тиск, рівень вібрації тощо. Ці дані передаються на центральний сервер, де вони обробляються та аналізуються. На основі результатів аналізу, можна виявити можливі проблеми з обладнанням та запропонувати відповідні рішення щодо його обслуговування та ремонту.

Системи моніторингу електроенергії: сенсори вимірюють різні параметри електричної мережі, такі як напруга, струм, потужність, фактор потужності тощо. Ці дані передаються на центральний сервер, де вони обробляються та аналізуються. На основі результатів аналізу, можна виявити можливі проблеми з електричною мережею та запропонувати відповідні рішення щодо їх вирішення.

Моніторинг стану людей: сучасні пристрої можуть вимірювати різні параметри здоров'я людини, такі як пульс, тиск, рівень кисню у крові тощо. Ці дані потім можуть бути передані за допомогою засобів зв'язку до спеціального програмного забезпечення для обробки та аналізу даних. Такі пристрої можуть бути корисними для моніторингу стану людей з певними захворюваннями або для спортсменів, які хочуть відслідковувати свій прогрес.

Моніторинг енергоспоживання: різні пристрої можуть бути встановлені для вимірювання енергоспоживання в будівлях або на підприємствах. Ці дані можуть бути передані до програмного забезпечення для аналізу та оптимізації використання енергії, що може допомогти зменшити витрати на електроенергію та зменшити негативний вплив на довкілля.

Моніторинг рівня забруднення: сенсори можуть бути встановлені для вимірювання рівня шкідливих речовин у повітрі, воді або ґрунті. Ці дані можуть бути передані до програмного забезпечення для аналізу та розробки планів зменшення забруднення та збереження довкілля.

Управління транспортом: сенсори та засоби зв'язку можуть бути встановлені на різних транспортних засобах, щоб збирати дані про рух, швидкість, витрату палива тощо. Ці дані можуть бути передані до централізованої системи, де вони обробляються та аналізуються для планування оптимальних маршрутів, контролю швидкості, підвищення безпеки на дорозі та ефективного використання палива. Наприклад, водії можуть використовувати системи GPS та навігації, щоб знайти оптимальний маршрут та уникнути заторів. Також, ці системи можуть забезпечувати автоматичний контроль за відстанню між транспортними засобами, що знижує ризик дорожньо-транспортних пригод.

## Аналіз роботи апаратної та програмної складових: переваги та недоліки

Робота зі збіркою дистанційної діагностики складається з кількох етапів, і кожен з них має свої особливості та вимоги.

Перший етап - планування та аналіз вимог. На цьому етапі визначається, які параметри будуть вимірюватися, які сенсори будуть використовуватися, які алгоритми обробки даних будуть застосовуватися, які метрики успішності будуть використовуватися для оцінки роботи системи.

Другий етап - розробка апаратної складової. На цьому етапі розробляється апаратна складова системи, яка буде відповідати вимогам, визначеним на першому етапі. Ця складова може містити різні компоненти, такі як сенсори, мікроконтролери, передавачі даних тощо.

Третій етап - розробка програмної складової. На цьому етапі розробляється програмна складова системи, яка буде відповідати вимогам, визначеним на першому етапі. Ця складова може містити різні компоненти, такі як алгоритми обробки даних, модулі передачі даних на сервер, інтерфейс користувача тощо.

Четвертий етап - тестування та налагодження. На цьому етапі проводиться тестування системи та налагодження її складових. Він дозволяє виявити та виправити помилки, перевірити відповідність системи вимогам та метрикам успішності.

Останній етап - впровадження та підтримка. На цьому етапі система впроваджується у роботу та підтримується відповідно до вимог та плану підтримки. В цей час можуть бути виявлені нові вимоги або проблеми, які потребують додаткової роботи з вдосконалення системи або розширення її функціональності.

Під час впровадження системи дистанційної діагностики необхідно враховувати вимоги технічної безпеки та конфіденційності даних. Також важливо забезпечити належне навчання персоналу, який буде користуватись системою, та підтримку відповідного інфраструктури, що забезпечить безперебійну роботу системи.

Після впровадження системи дистанційної діагностики необхідно забезпечити постійну підтримку та обслуговування системи, а також розробку та впровадження нових функцій та можливостей для підвищення її ефективності та функціональності. Також можуть виникати необхідність у заміні окремих компонентів системи або їх модернізації для забезпечення більш ефективної роботи.

Отже, впровадження та підтримка системи дистанційної діагностики є важливим етапом, що дозволяє забезпечити безперебійну та ефективну роботу системи та забезпечити максимальну користь від її використання. Однак слід зазначити, що сучасні технології апаратної та програмної складової мають ряд переваг та недоліків.

Переваги:

* Дистанційна діагностика дозволяє оперативно виявляти можливі проблеми та попереджати їх виникнення.
* Зменшення часу та витрат на проведення діагностики, оскільки можна віддалено контролювати стан транспортного засобу та вчасно виявляти проблеми.
* Збільшення ефективності та продуктивності, оскільки можна забезпечити більш точну діагностику та управління процесами.
* Зменшення ризиків для працівників, які займаються діагностикою, оскільки вони не повинні бути на місці для проведення діагностики.
* Збільшення надійності та безпеки транспортного засобу, оскільки можна відслідковувати та контролювати стан його компонентів в реальному часі.

Недоліки:

* Висока вартість встановлення систем дистанційної діагностики.
* Потрібна налагоджена технічна інфраструктура та високі технології для ефективного збору та аналізу даних.
* Необхідність спеціальної кваліфікації для розуміння та інтерпретації даних, які надходять з систем дистанційної діагностики.
* Небезпека зловживання системами дистанційного моніторингу з боку зловмисників, які можуть використовувати зібрані дані для злочинних цілей.

## Висновок до розділу 2

Провівши аналіз можемо зазначити, що не зважаючи на те, що системи дистанційної діагностики мають свої переваги та недоліки, вони є важливою складовою для забезпечення надійності та безпеки транспортних засобів. За допомогою цих систем можна вчасно виявляти можливі проблеми та здійснювати попереджувальний ремонт, що зменшує ризик виникнення аварій та збільшує ефективність використання транспортних засобів. Крім того, системи дистанційної діагностики дозволяють вести моніторинг транспортних засобів в режимі реального часу, що дозволяє оперативно реагувати на зміни в їх стані та уникати негативних наслідків.

Наразі деякі системи дистанційної діагностики можуть бути досить затратними, особливо для менших компаній та підприємств. Крім того, іноді можуть виникати проблеми з точністю збору та обробки даних, що може призводити до неправильних діагнозів та рішень. Також, системи дистанційної діагностики можуть вимагати певної кваліфікації та підготовки для їх встановлення та ефективного використання.

# РОЗРОБКА ПРОТОТИПНОЇ СИСТЕМИ



## Концепт визначення вимог до систем під час їх створення

Визначення вимог до систем - це перший і один з найважливіших етапів проектування будь-якої системи, включаючи програмне забезпечення. Цей етап передбачає аналіз потреб користувачів та інших зацікавлених сторін, формулювання вимог до системи та її функціональних характеристик.

Процес визначення вимог до системи починається зі збору відомостей про потреби користувачів та інші особливості проекту. Це можна зробити шляхом проведення досліджень та аналізу конкурентів на ринку, опитування потенційних користувачів, вивчення ринку та технологічних тенденцій.

Далі, на основі зібраних даних, формулюються вимоги до системи, її функцій та можливостей. Ці вимоги можуть бути представлені у вигляді списку функцій, які має виконувати система, а також вимог до її ефективності, надійності, масштабованості, безпеки тощо.

Особливу увагу при визначенні вимог до системи слід приділяти цільовій аудиторії та її потребам. Якщо система розробляється для бізнесу, то важливо врахувати бізнес-потреби, які можуть включати збільшення ефективності, зниження витрат тощо. Якщо система розробляється для споживачів, то важливо враховувати їхні потреби та очікування, наприклад, простоту використання, зручність тощо.

Після визначення вимог до системи, їх необхідно ретельно проробити та уточнити. Цей процес може зайняти певний час, але це дозволить уникнути помилок та недорозумінь на пізніших етапах проектування. Після того, як вимоги були визначені та уточнені, їх необхідно проаналізувати та розробити план дій для реалізації проекту.

Визначення вимог до системи є критичним етапом у процесі розробки будь-якої інформаційної системи. Недостатня чіткість вимог може призвести до помилок у розробці та некоректної роботи системи. Крім того, невірно визначені вимоги можуть призвести до додаткових витрат на переробку та редизайн системи, що може значно затримати процес її впровадження.

У процесі визначення вимог до системи, першим кроком є зібрання вимог від потенційних користувачів системи та інших зацікавлених сторін. Для цього можуть використовуватися різні методи, такі як інтерв'ю, опитування, спостереження та інші.

Після збору вимог, необхідно проаналізувати їх та визначити, які функції та можливості повинна мати система для задоволення цих вимог. Наступним кроком є розробка детального опису функцій та можливостей системи.

Опис функцій та можливостей системи повинен бути чітким та детальним, включати всі необхідні деталі та технічні вимоги до системи. Для цього можна використовувати різні методи, такі як створення таблиць, блок-схем, діаграм та інші.

Також, важливим етапом є визначення технічних вимог до системи. Необхідно визначити, які технології та програмні засоби будуть використовуватися для реалізації системи, які вимоги до обладнання, програмного забезпечення та мережевої інфраструктури.

Після визначення вимог до системи, їх необхідно формалізувати та документувати у вигляді Технічного завдання (ТЗ). ТЗ має містити детальний опис функцій та можливостей системи, вимоги до її надійності та стійкості, опис процесу взаємодії з користувачем, а також вимоги до забезпечення безпеки та захисту даних.

## Етапи розробки апаратної складової

Розробка апаратної складової є важливим етапом у створенні будь-якої системи дистанційної діагностики. Цей етап включає в себе проектування та розробку апаратних компонентів системи, таких як датчики, пристрої збору даних та інші елементи.

Перший крок у розробці апаратної складової - це визначення технічних вимог до датчиків та інших елементів системи. Необхідно розробити детальний план дій, який визначає типи датчиків, їхні параметри та характеристики, які необхідно забезпечити для збору необхідної інформації про стан об'єкта діагностики. Вимоги до апаратної складової повинні бути зрозумілі та конкретні, з можливістю перевірки на різних етапах розробки.

Наступним етапом є проектування схеми апаратної складової та розробка прототипу. При проектуванні датчиків та інших елементів системи дистанційної діагностики необхідно враховувати різні фактори, такі як тип датчика, діапазон вимірювання, точність вимірювання та стійкість до шуму та перешкод.

Після проектування необхідно здійснити розробку прототипу системи. Це дозволяє виконати тестування прототипу та перевірити його роботу в реальних умовах. Результати тестування можуть бути використані для вдосконалення апаратної складової та виправлення помилок.

Крім того, важливим етапом є вибір технологій та матеріалів для виготовлення датчиків та інших елементів системи. Необхідно враховувати і технічні та економічні параметри при виборі матеріалів та технологій. Наприклад, деякі матеріали можуть бути дорогими, але вони можуть забезпечувати більшу точність вимірювання, або мати більшу стійкість до зносу. Також важливо враховувати можливості масового виробництва, щоб забезпечити ефективність та конкурентоздатність системи.

Після вибору матеріалів та технологій, можна розпочати процес виготовлення апаратних компонентів системи. Важливо дотримуватись всіх необхідних стандартів та вимог щодо якості та безпеки виготовлення компонентів, а також розробляти проект з урахуванням майбутнього обслуговування та ремонту. Після виготовлення компонентів, необхідно провести тестування для перевірки їх працездатності та точності вимірювань.

Завершення розробки апаратної складової передбачає монтаж та налаштування всіх компонентів системи. Також може бути необхідно розробити документацію з експлуатації, допоміжні програмні засоби для роботи з апаратною складовою, а також надати користувачам інструкції зі використання та обслуговування системи.

## Розробка програмної складової

Процес розробки програмної складової може бути розділений на декілька етапів, включаючи:

* Аналіз вимог до програмної складової. На цьому етапі необхідно уточнити вимоги до функціональності програми, а також визначити технічні обмеження та вимоги до безпеки даних.
* Проектування архітектури програмного забезпечення. На цьому етапі розробляється загальна структура програми, включаючи модулі та компоненти, які будуть використовуватися в процесі роботи.
* Розробка коду програмного забезпечення. На цьому етапі розробляється сам код програми, який відповідає вимогам до функціональності та технічним вимогам. Розробка коду може виконуватися з використанням різних мов програмування, залежно від специфіки проекту.
* Тестування програмного забезпечення. Після розробки коду необхідно виконати тестування програми, щоб переконатися, що вона відповідає вимогам до функціональності та безпеки даних. Тестування може включати різні типи тестів, такі як модульні тести, інтеграційні тести та приймальні тести.
* Реліз та підтримка програмного забезпечення. Після тестування програмне забезпечення готове до релізу. Після релізу можуть виникати проблеми, які не вдалося виявити під час тестування, тому необхідна підтримка програмного забезпечення та вирішення виявлених проблем.

При розробці програмної складової необхідно враховувати багато вимог, серед яких найбільш важливими є безпека даних, швидкість роботи програми та простота використання.

Безпека даних є найважливішою вимогою для більшості систем. У програмній складовій необхідно враховувати різні аспекти безпеки даних, такі як захист від несанкціонованого доступу, захист від вірусів та шкідливих програм, забезпечення приватності користувачів та інших вимог.

Швидкість роботи програми також є важливою вимогою, оскільки користувачі очікують, що програма буде працювати швидко та ефективно. Це може вимагати оптимізації алгоритмів, використання швидкодіючих баз даних та інших технік.

Простота використання також є важливою вимогою для більшості користувачів. Програмна складова повинна бути інтуїтивно зрозумілою та простою у використанні, щоб користувачі могли легко знаходити та використовувати функції системи без додаткової навчання або підтримки.

Крім цього, вимоги користувачів можуть включати такі аспекти, як мобільність, доступність, функціональність та інші. Всі ці вимоги повинні бути враховані при розробці програмної складової, щоб забезпечити найбільш оптимальне та зручне використання системи користувачами.

## Інтеграція апаратної та програмної складових

Інтеграція апаратної та програмної складових є одним з ключових етапів розробки системи збору та аналізу даних. На цьому етапі відбувається злиття апаратної та програмної складових в єдину функціональну систему, що забезпечує збір, обробку та аналіз даних.

Процес інтеграції складається з кількох етапів:

Підготовчий етап. На цьому етапі відбувається підготовка апаратної та програмної складових до інтеграції. Для цього необхідно перевірити наявність необхідних програмних драйверів для обладнання, переконатися в наявності всіх необхідних компонентів, а також встановити необхідне програмне забезпечення.

Інтеграція апаратної складової. На цьому етапі проводиться підключення апаратної складової до комп'ютера або сервера та перевірка її роботи. Необхідно перевірити правильність підключення та налаштування обладнання, а також провести тестування роботи датчиків та інших елементів системи.

Інтеграція програмної складової. На цьому етапі відбувається підключення програмного забезпечення до апаратної складової та перевірка його роботи. Важливо перевірити правильність підключення програми до апаратної складової, а також налаштування параметрів програми.

Тестування взаємодії. Після інтеграції апаратної та програмної складових необхідно провести тестування їхньої взаємодії. Під час тестування перевіряється коректність збору та передачі даних, а також правильність обробки та аналізу даних. В цьому процесі використовуються спеціальні тестові сценарії, які дозволяють перевірити роботу системи в різних умовах.

Тестування взаємодії можна проводити як автоматизовано, так і вручну. При автоматизованому тестуванні використовуються спеціальні програми-тестери, які можуть створювати різні сценарії взаємодії та перевіряти результати роботи системи. При ручному тестуванні тестувальник виконує різні дії в системі та перевіряє, чи відбувається правильна реакція системи.

Після проведення тестування взаємодії необхідно аналізувати результати та виявлені помилки. Якщо були виявлені помилки, їх необхідно виправити та повторити тестування. Після успішного тестування можна переходити до наступних етапів розробки системи.

## Створення прототипної системи її тестування та удосконалення

Першим етапом тестування є функціональне тестування, під час якого перевіряється, чи виконує система свої основні функції. Для цього можуть бути використані спеціально розроблені тестові сценарії, які передбачають різні варіанти роботи системи та перевіряють її реакцію на різні ситуації.

Другим етапом є тестування продуктивності системи, під час якого перевіряється швидкість роботи системи та її можливості при великих навантаженнях. Це може бути важливо, якщо система передбачає обробку великої кількості даних або має велику кількість користувачів.

Третім етапом є тестування безпеки системи, під час якого перевіряється її стійкість до атак та можливість витоку даних. Для цього можуть бути використані різні методи та інструменти, що дозволяють виявити потенційні уразливості системи та запобігти їхньому використанню зловмисниками.

Потім, як були розроблені апаратна та програмна складові системи та їх було інтегровано, необхідно провести тестування прототипу системи. Метою тестування є виявлення можливих проблем та недоліків, що потребують виправлення, а також перевірка відповідності розробленої системи вимогам

Після тестування прототипу системи виявляються можливі проблеми та недоліки, які потребують виправлення. Це може включати зміни у програмній або апаратній складовій, вдосконалення процесів взаємодії між компонентами системи та інші заходи.

### Розробка прототипної системи

Розробка прототипної системи є важливим етапом в створенні нових продуктів та технологій. Прототип - це перший функціонуючий зразок майбутньої системи або продукту, що дозволяє перевірити її основні функції та можливості.

Процес розробки прототипу зазвичай починається зі створення концепції, визначення основних вимог та функціональності системи. Після цього виконуються етапи проектування та розробки, включаючи вибір необхідних технологій та матеріалів.

Розробка прототипу зазвичай здійснюється ітеративно - на кожному етапі розробки проводиться тестування та аналіз результатів для покращення системи. Крім того, розробка прототипу може бути розділена на етапи залежно від обсягу робіт та вимог до системи.

Основна мета розробки прототипу - це створення функціонуючого зразка системи для перевірки її можливостей та визначення потенційних проблем. Після успішної розробки прототипу можна продовжувати роботу над системою, удосконалюючи її функціональність та покращуючи якість.

Отже створимо приклад прототипної системи для дистанційного аналізу та обслуговування. Для створення системи будемо використовувати мову програмування Python.

Концепція:

Система повинна бути спроможна проаналізувати поточний стан в нашому випадку автомобіля, зчитувати дані з датчиків та у разі виявлення поломки сповістити власника.

Визначення вимог та функціональність:

Функціональність: система повинна виконувати певні функції, які були визначені на етапі розробки концепції. Наприклад, якщо прототипна система призначена для дистанційної діагностики, вона повинна мати можливість зчитувати дані з сенсорів та проводити аналіз даних для визначення стану системи.

Ефективність: система повинна працювати швидко та ефективно. Використання Python може мати вплив на швидкість роботи системи, тому необхідно забезпечити оптимальну роботу програми та використовувати ефективні алгоритми.

Гнучкість: система повинна бути гнучкою та легко змінюватись у випадку необхідності. Використання Python та інших відкритих технологій може допомогти забезпечити гнучкість та легкість змін у системі.

Користувацький інтерфейс: система повинна мати зручний та інтуїтивно зрозумілий користувацький інтерфейс, щоб користувачі могли легко взаємодіяти з системою та отримувати необхідну інформацію.

Сумісність: система повинна бути сумісною з різними операційними системами

Враховуючи основні вимоги до програми, створимо просту систему діагностики котра буде імітувати аналіз автомобіля та відображати поточну інформацію з датчиків

Для створення даного програмного забезпечення будемо використовувати мову програмування Python з бібліотекою PyQt5 та середовище Qt Designer.

Спочатку потрібно створити основне вікно нашої програми, для цього і в подальшому для створення додаткових вікон використовуємо Qt Designer.

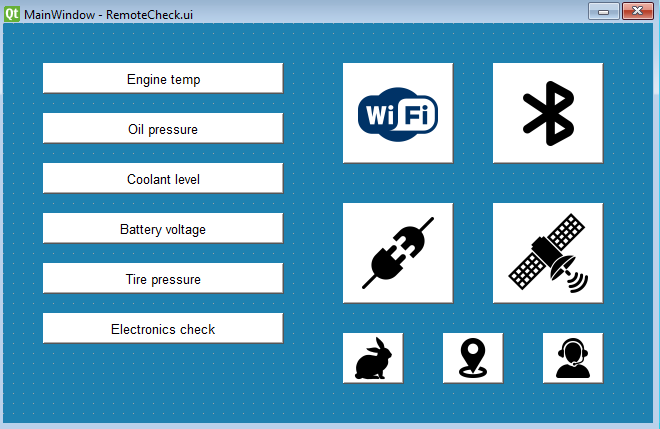


Рисунок 3.1 – Інтерфейс головного вікна програми

Після створення основного вікна та дизайну кнопок, створюємо дочірні вікна які будуть викликатися натисканням безпосередньо на ці вікна.

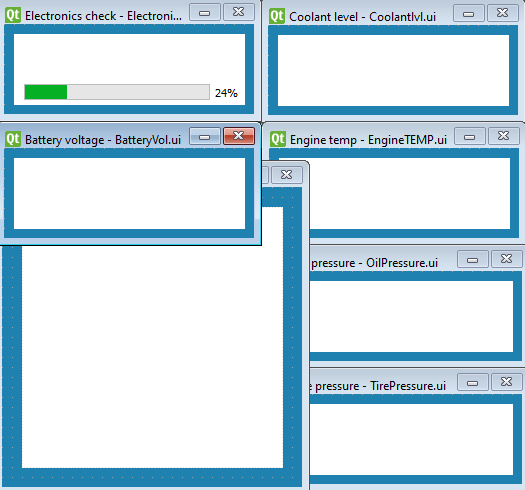


Рисунок 3.2 – Дочірні вікна програми

Після створення основних вікон для програми, потрібно реалізувати їх функціонал. Для цього спочатку потрібно конвертувати наші файли **.ui** в **.py** для подальшої роботи з ними.

Відкриваємо термінал та вводимо команду для конвертування. Подібні дії проведемо з кожним “.ui” файлом.



Рисунок 3.3 – Команда конвертування

Тепер коли отримано вікна в форматі “.py”потрібно зайнятись їх функціоналом. Насамперед потрібно встановити потрібні бібліотеки для подальшої роботи, а саме такі як:

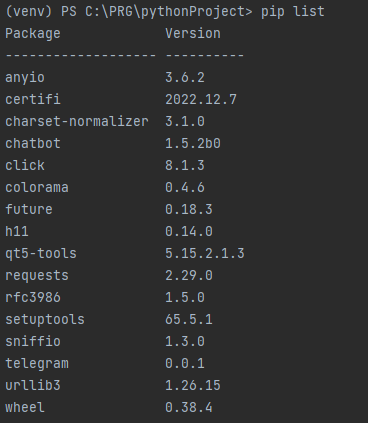


Рисунок 3.4 – Основні бібліотеки

Отримавши всі потрібні бібліотеки можна приступити до написання коду, так як в основному вікні реалізовувати якісь складні операції, не потрібно, тому переходимо до інших вікон, наприклад до клавіш які будуть імітувати діагностику окремих елементів автомобіля.

Так як готовий інтерфейс вже є, тепер залишається на віджеті “QLabel” розмістити текст який буде там відображатися.

Для цього використаємо вже готовий код, котрий ми отримали при конвертації та додамо до нього основні елемент який відоповідає за відображення тексту, тобто “setText”, оскільк нам потрібно, щоб при запуску вікна, ми отримували не просто чисте поле, а напис “Loading…”, що в свою чергу буде вказувати на загрузку наших даних з датчиків.

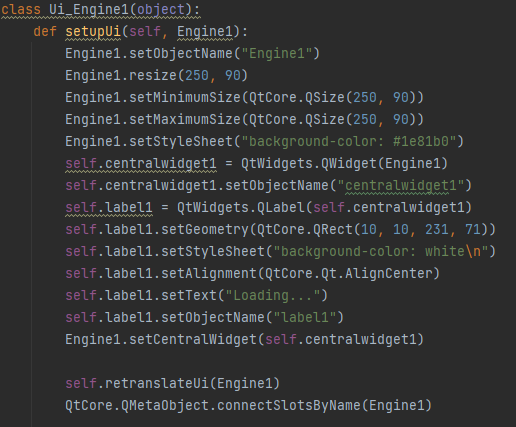


Рисунок 3.5 – Загальний вигляд дочірнього вікна у вигляді коду

Функція “update\_temp” генерує випадкове число в діапазоні від 70 до 100, і потім оновлює текст та колір елемента мітки “label1” залежно від згенерованого значення. Якщо згенероване число менше 87 або більше 105, то текст мітки містить значення температури та “Temp not in normal” і має червоний колір, в іншому випадку текст мітки містить значення температури та “Temp in normal” і має зелений колір. Також для більшої гнучкості роботи програми додамо таймер який буде оновлювати дані на віджеті кожні дві секунди.

Для решти вікон реалізовуємо подібний код та підхід так як вони не мають сильно відрізнятися по дизайну та по самій структурі.



Рисунок 3.6 – Реалізація функціоналу описаного вище

Тепер коли ми реалізували основні вікна, потрібно реалізувати вікно “швидкої діагностики” котра буде видавати інформацію відразу про всі елементи автомобіля. Вікно GPS навігатора за допомогою якого ми зможете відслідковувати місцезнаходження свого авто та реалізувати кнопку технічної підтримки, щоб при виявленні будь-якої проблеми в програмі чи в цілому потрібна консультація, то можна буде за допомогою телеграм бота відправити скаргу яку розглянуть спеціалісти.

Щоб відображати мапу на нашому навігаторі, використовуємо раніше створене вікно в Qt Designer та просто використовуємо імпорт для інтеграції веб-браузера у програми на мові Python з використанням PyQt5.



Рисунок 3.7 – Інтеграція веб-браузера в програму

Після чого реалізовуємо за допомогою коду відображення мапи на віджеті

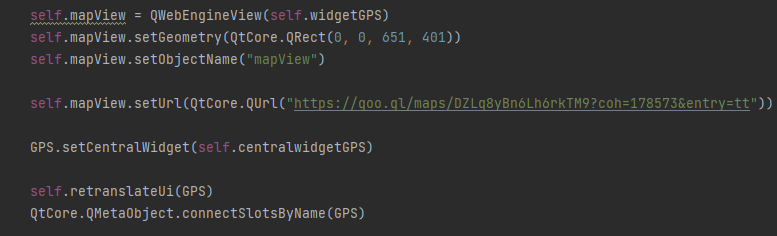


Рисунок 3.8 – Код для реалізації відображення мапи

Наступне, що ми повинні реалізувати це телеграм бота для підтримки користувачів. Для створення даної функції використовуємо телеграм бота для створення власного, а саме “BotFather”. Запускаємо його та використовуємо команду “/start” після обираємо “/newbot” та прописуємо назву для свого бота, в нашому випадку використовуємо “Remote Diagnostic Support”. Після нам пропонують створити username за допомогою якого користувачі будуть знаходити нашого бота, так як ми повинні зберігати в username назву нашого бота, то просто напишемо “RemoteDiagnosticSupport\_BOT”. “BotFather” нам надасть API нашого бота, який знадобиться нам у подальшій роботі вже в Python. Також створимо спеціальну групу для спеціалістів, куди будуть надсилатися заявки з проблемами від користувачів. Тепер коли в є закрита група спеціалістів та API нашого бота, напишемо код який реалізовуватиме його функціонал.



Рисунок 3.9 – Установка бібліотеки для роботи з ботом

Імпортуємо модуль “telebot”, який дозволяє створювати та керувати чат-ботами у Telegram.

Далі вказуємо токен бота, який використовується для з'єднання з API Telegram.

Додаємо змінну “group\_chat\_id” яка містить ідентифікатор групового чату, в який будуть надсилатись повідомлення від користувачів.

Далі визначимо функцію “start\_handler”, яка буде відповідати на команду “/start” від користувача. Функція надсилає привітальне повідомлення з запитом на опис проблеми.

Друга функція “message\_handler” буде відповідати на всі повідомлення, що надійшли до бота. Вона надсилатиме користувачеві повідомлення про те, що запит отримано та що спеціалісти розглянуть його протягом 1-2 робочих днів. Також вона надсилатиме повідомлення до групового чату з текстом повідомлення та ідентифікатором користувача.

Самий останній це метод “polling” який буде виконувати безкінечний цикл, щоб бот постійно слухав вхідні повідомлення від користувачів і відповідав на них. І додаємо “none\_stop=True” це буде означати, що бот продовжуватиме роботу, навіть якщо виникне помилка.

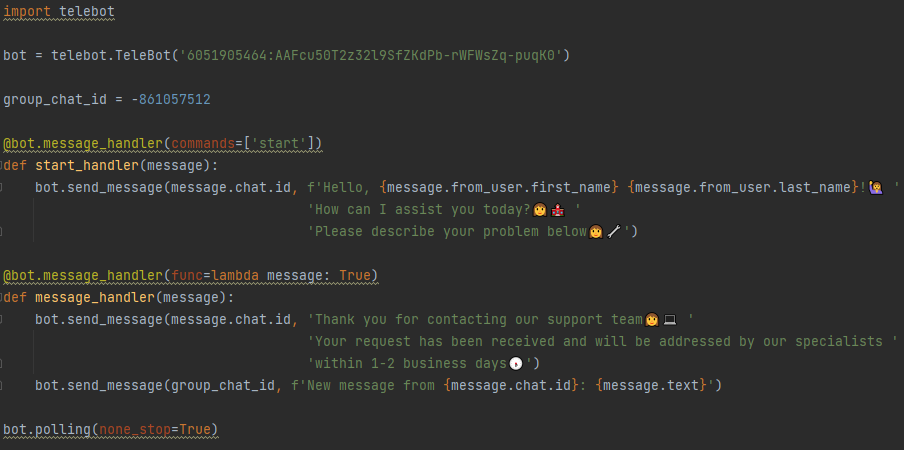


Рисунок 3.10 – Реалізація телеграм бота в коді

Переходимо тепер до реалізації вікна швидкої діагностики Цей код є графічним інтерфейсом користувача на мові програмування Python з використанням бібліотеки PyQt5.

Основним елементом у нас є вікно “fastFT” з розмірами 240х230 пікселів, яке містить центральний віджет “centralwidgetFT”. Цей віджет містить ще один віджет “widgetFT” з розмірами 200х200 пікселів, що розміщується в “centralwidgetFT”. Функція random.randint(0, 1) випадковим чином генерує значення стану деяких систем автомобіля. Ці значення використовуються для оновлення тексту лейблів на виджеті. Лейбли містять назву кожної системи автомобіля та опис її стану (нормальний або не нормальний). Крім того, кожен лейбл має свій власний кольоровий індикатор, що залежить від згенерованого значення.

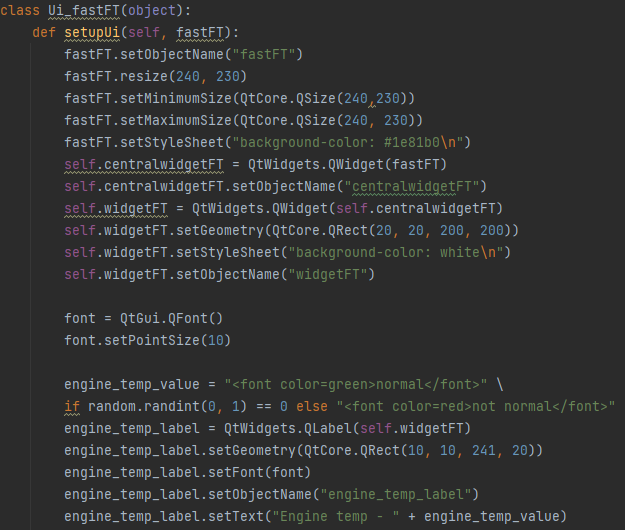


Рисунок 3.11 – Частина реалізованого запиту вікна швидкої діагностики

Після того, як ми написали функціонал для кожного вікна потрібно прив’язати кожне дочірнє вікно до кнопкок на основному. Реалізовувати це все ми будемо за допомогою імпортів, в новий проект. Коли імпорт проведено, слід перейти до реалізації функціоналу кнопок на основному вікні. Потрібно встановити залежність при якій на натискання певної клавіші, буде з’являтися відповідне вікно. Для того щоб реалізувати дану функцію, використаємо посилання на поточну кнопку підключивши через сигнал потрібний нам слот. Тобто “self.ui.btn1.clicked.connect(self.openEngine1)”. За таким принципом підключаємо решту кнопок. Тепер коли підключення налаштоване, потрібно правильно прописати відображення цих вікон в коді основного вікна.

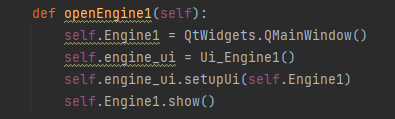


Рисунок 3.12 – Реалізація відображення дочірнього вікна

Наша демо-версія програми майже готова, залишається лиш реалізувати блокування кнопок діагностики доки користувач не обере тип підключення до авто. Нижче буде наведено приклад, як реалізувати це для однієї кнопки. Щоб залучити всі кнопки в цей процес, просто потрібно продовжити список в колонці (False) та (True).

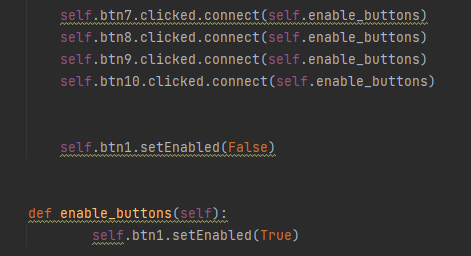


Рисунок 3.13 – Реалізація активації кнопок інтерфейсу по підключенню

Тепер при запуску повинні бути активні тільки чотири клавіші підключення, після вибраного типу підключення, решта кнопок стають також активними та готовими до використання.

### Тестування та вдосконалення прототипу

Сформувавши повністю програму - протестуємо її та виявимо недоліки

Запустивши вікно програми можна помітити, що всі клавіші неактивні, окрім чотирьох, які відповідають за підключення до системи.

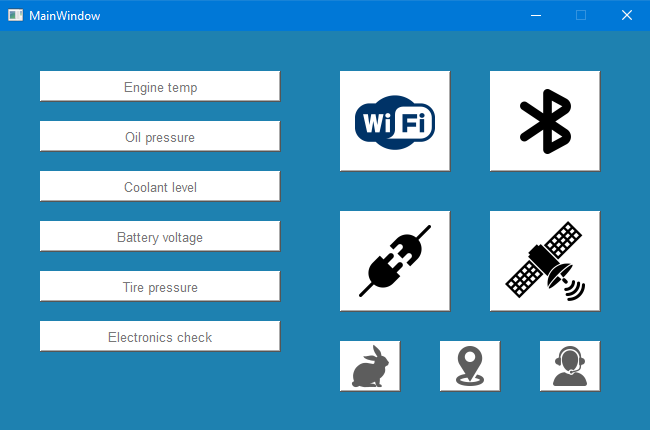


Рисунок 3.14 – Головне вікно нашої програми

Обираємо довільну клавішу, та переходимо до тестування решти функціоналу. Безпосередньо перейдемо до перевірки шести основних елементів автомобіля. Запустивши кожне з них можемо переконатися в коректній роботі програми.

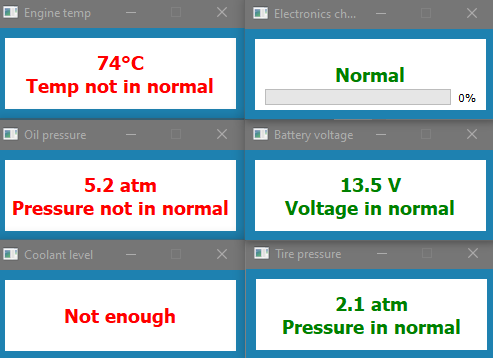
****

Рисунок 3.15 – Дочірні вікна діагностики основних елементів авто

Всі вікна працюють коректно. Отже слід переходити до GPS – навігатору.

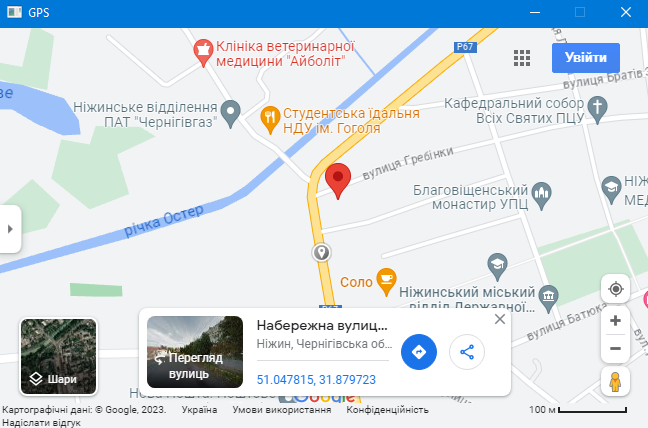


Рисунок 3 .16 – Демонстрація вікна GPS

Клавіша та вікно працюють коректно, тому ідемо далі та відкриваємо вікно швидкої діагностики.

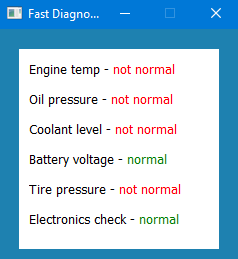


Рисунок 3.17 – Вікно швидкої діагностики

Як ми бачимо в режимі швидкої діагностики, висвітлюється стан відразу всіх основних елементів автомобіля але інформаційно цей варіант діагностики є урізаним оскільки тут відсутні одиниці вимірювання температури, тиску, заряду тощо. Також в цьому режимі дані не оновляються кожні 2 секунди, що зменшує точність самої діагностики. Вікно та клавіша працюють коректно, тому наступне, що ми протестуємо, буде служба підтримки.

Натиснувши клавішу підтримки, нас перекине в браузер котрий працює за замовчуванням та відкриє посилання на нашого бота, де ми зможете обрати відкрити його в desktop версії чи в web. Після запуску бота, він привітається та попросить описати свою проблему, після опису проблеми він дякує за звернення та інформує, що відповідь на запитання, прийде протягом 1-2 робочих днів. Саме ж повідомлення користувача дублюється в закритий чат спеціалістів.

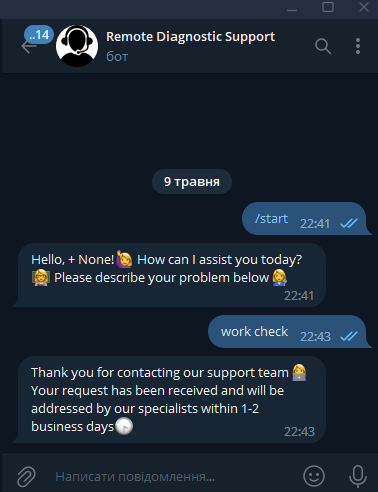


Рисунок 3.18 – Діалог з ботом підтримки

В закритому чаті ми бачимо повідомлення які надходять нам з бота, вони містять id користувача та повідомлення які він надіслав у службу підтримки. В даній функції вікно саме по собі відсутнє, натискання на клавішу відразу відкриває нам телеграм бота.

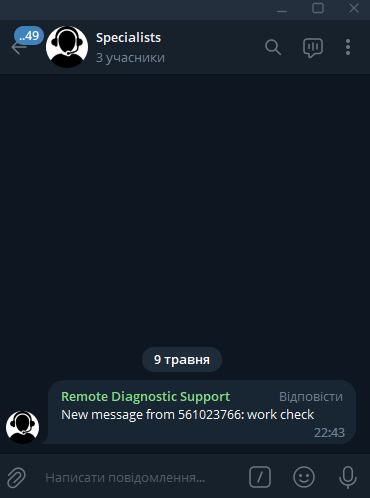


Рисунок 3.19 – Транслювання повідомлень з бота у групу

Під час тестування створеного програмного забезпечення були виявленні такі проблеми:

1. Кнопки вибору підключення залишаються активними постійно
2. Відсутність клавіші відключення системи від авто
3. Відсутність повної синхронної діагностики авто
4. Відсутність будь-якого захисту

Тепер коли ми під час тестування знайшли незручності в програмі, то почнемо роботу з її удосконалення спираючись на зауваження в пунктах вище.

Так як проблеми пунктів 1 та 2 є дуже пов’язаними, то розглянемо їх разом. Справа в тому, що після вибору підключення, кнопки які за це відповідають залишаються активними, що не є логічними. Також із-за відсутності клавіші котра відповідала б за відключення, програму доводиться закривати та відкривати знову, що обрати інший тип підключення. Тому спираючись на цю інформацію реалізовуємо наступні елементи коду:

Додаємо клавішу котра буде імітувати відключення системи від авто.

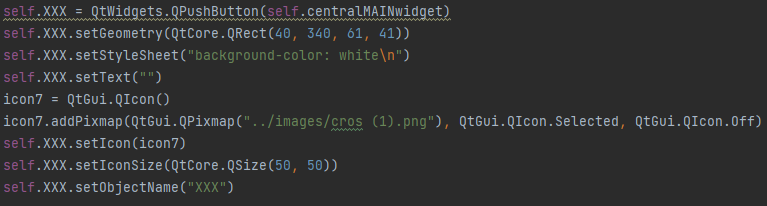


Рисунок 3.20 – Створення нової кнопки з функцією відключення

Після створення кнопки реалізовуємо активацію та деактивацію певних кнопок. Сама суть схожа на роботу ту яку ми проводили вище, але тут ми додаємо нову залежність спираючись на кнопку “відключення”.

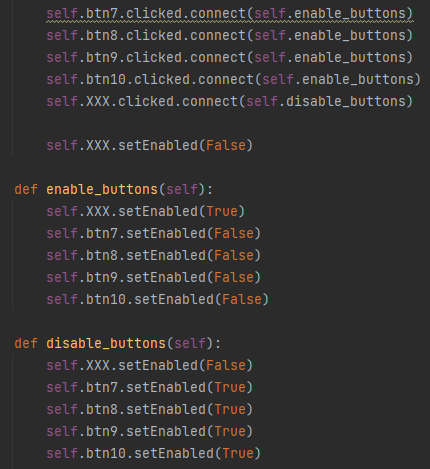


Рисунок 3.21 – Удосконалення функції активації та деактивації кнопок

Після успішної реалізації коду вище, переходимо до 3 пункту. Справа в тому, що користувач може провести діагностику або швидким методом та отримати менш детальну інформацію, або провести діагностику кожного елементу авто, що є зручним, але є енергозатратним коли нам потрібно провести детальну діагностику відразу всіх елементів. Для цього було вирішено, що при натисканні певної кнопки ми отримаємо відкриття відразу всіх вікон котрі стосуються окремних елементів автомобіля.

Перейдемо безпосередньо до створення самої кнопки, використовуємо стилі котрі були використані у попередніх прикладах. Розміщуємо кнопку поряд з кнопкою котра відповідає за “швидку діагностику”.

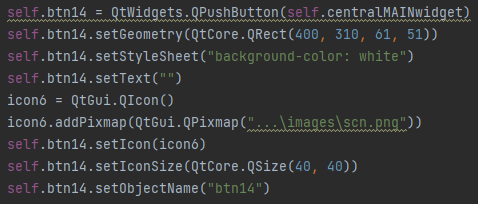


Рисунок 3.22 – Створення кнопки повної діагностики

Після створення слід підключити дочірні вікна основних елементів авто.

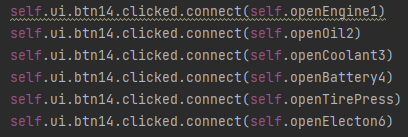


Рисунок 3.23 – Підключення вікон до кнопки

Тепер виникає проблема в розміщенні вікон, при натисканні, викликаються всі вікна, які накладають одне на одне. Потрібно, розмістити їх навколо основного вікна, щоб вони не накладались та не закривали інші кнопки. У функціях відображення додаємо до кожного вікна ці рядки відповідно до назви.

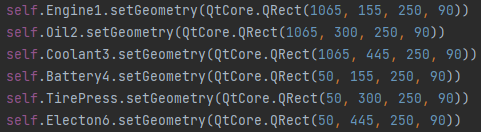


Рисунок 3.24 – Налаштування розміщення вікон

До пункту 4 ми перейдемо пізніше, адже для початку нам потрібно перевірити коректність нашої програми після внесених змін в коді.

Запустивши код ми отримуємо дане вікно, та заблоковані всі кнопки окрім кнопок підключення.

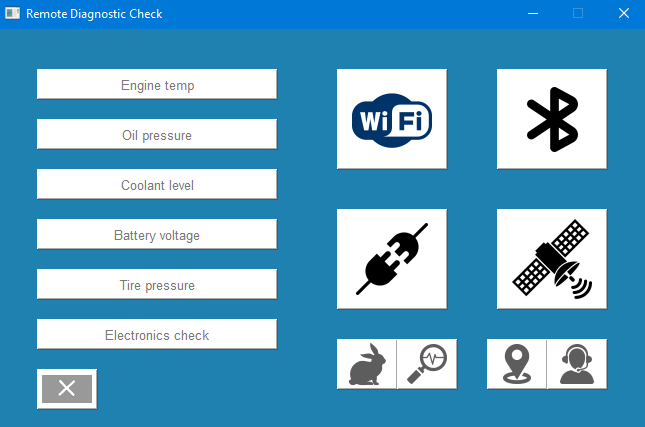


Рисунок 3.25 – Головне вікно

Обираємо тип підключення та перевіряємо кнопку повної діагностики.

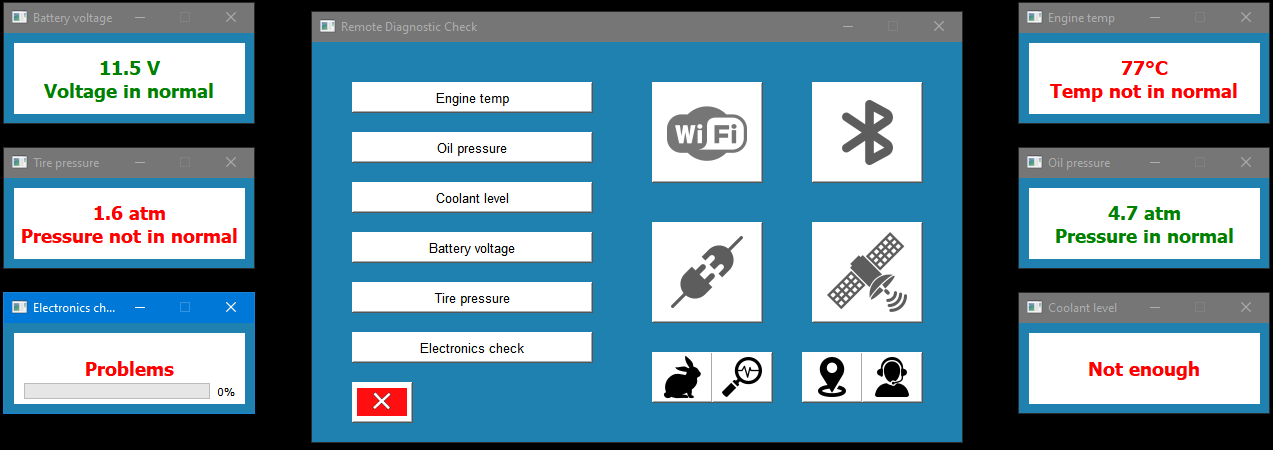


Рисунок 3.26 – Робота повної діагностики

Як бачимо все працює коректно, вікна не накладаються одне на одне, розміщуються навколо основного вікна, кнопки підключення стають неактивними, а кнопка відключення навпаки, активною. Натискаючи на неї ми відключаємось від системи так закриваємо всі вікна.

Коли ми повністю завершили роботу над нашою програмою скомпілюємо наш Python файл в “.exe”. Для цього встановлюємо спеціальну бібліотеку, прописавши “pip install pyinstaller”.

Після завантаження бібліотеки виконуємо в терміналі команду “pyinstaller -w -F RemoteDIA.py”.

-w вказує, що програма має бути запущена у режимі без консолі (без вікна командного рядка). Це означає, що при запуску створений виконуваний файл не відображатиме консольне вікно командного рядка.

-F вказує, що ми хочете створити єдиний виконуваний файл. Замість створення кількох окремих файлів, таких як виконуваний файл, залежності та ресурси, ми отримаємо один єдиний виконуваний файл, який містить усе необхідне для запуску програми.

Тепер після збору нашої програми перейдемо до 4 пункту та реалізуємо простий захист нашої програми. Напишемо код в якому при введенні правильного паролю, буде відкриватися наша програма.

Для початку перейдемо до створення простого але інтуїтивно зрозумілого інтерфейсу вікна із запитом на введення паролю.

Якщо пароль буде введено неправильно, то програма видасть помилку про те що пароль був неправильним.

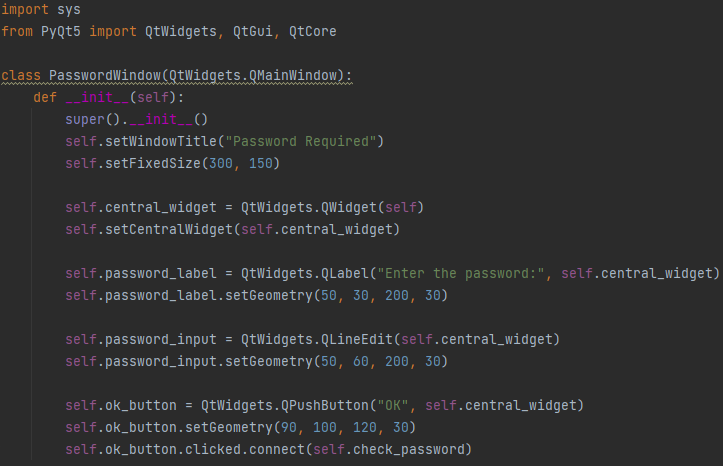


Рисунок 3.27 – Створення інтерфейсу для вікна з паролем

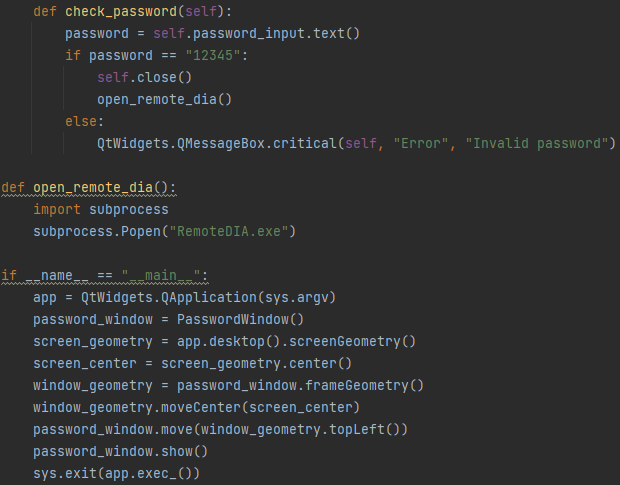


Рисунок 3.28 – Реалізація роботи вікна з паролем

Після створення вікна котре буде запитувати пароль у користувача, перейдемо до конвертації його в формат “.exe”. Для цього в терміналі виконуємо команду “pyinstaller -w -F pws.py”.

Запускаємо наш exe файл та вводимо невірний пароль, щоб перевірити чи буде працювати сповіщення про неправильний пароль.

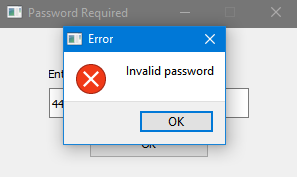


Рисунок 3.29 – Отримання помилки про введення неправильного паролю

Для перевірки вводимо невірний пароль, як бачимо наше вікно сповіщає про те що пароль був неправильний, вводимо тепер вірний пароль, який був заданий раніше в коді “12345” та отримуємо доступ до нашої програми.

Потрібно відмітити, що дана розробка програми не є до кінця робочою, адже вона лише імітує підключення та аналіз автомобіля. Ця програма лише є прототипом, що може послугувати далі в розробці повного програмного забезпечення з підключенням до автомобіля використовуючи створений інтерфейс та основні елементи як шаблон.

## Висновок до розділу 3

В третьому розділі розглядалися основні вимоги до системи її програмна та апаратна складова. Були проаналізовані їх етапи.

Також даний розділ включає в себе розробку прототипної системи діагностики автомобіля, використовуючи бібліотеку PyQt5. Розглянуті основні елементи інтерфейсу, реалізації залежностей та створення телеграм бота для підтримки.

На прикладі створення власної прототипної системи за допомогою бібліотеки PyQt5 та програми QtDesigner було наочно показано простоту використання даних платформ особливо, що стосується створення інтерфейсу.

Також були дослідженні основні елементи автомобіля котрі потребують аналізу як швидкого так і повного. Були створені основні зв’язки з різними елементами прототипної системи, а саме, підтримка, гул карта, вибір підключення, відключення та основі покажчики системи .

Була проведена вдала інтеграція нових кнопок, які були додані задля комфортної роботи або цього вимагав сам функціонал системи. Нові кнопки були стилізовані під загальний інтерфейс нашого прототипу.

Були проведені тести системи та були визначені основні недоліки які в ході удосконалення зазнали виправлень. Загальний функціонал програми був розширений. Також був доданий захист даної системи.

Основні вдосконалення та функції системи виглядають так:

* Кнопка відключення від ПЗ
* Активація та деактивація кнопок
* Реалізація повної діагностики
* Створення телеграм-бота
* Створення захисту

# ЗАГАЛЬНИЙ ВИСНОВОК

В даній дипломній роботі був проведений аналіз такого явища, як дистанційна діагностика. Були визначені основні переваги від традиційної діагностики та недоліки. Проаналізовано також перспективу розвитку, використання та призначення систем дистанційної діагностики та обслуговування в різних галузях промисловості.

Основними перевагами дистанційного обслуговування та діагностики є:

1.Простота у використанні

2.Більш точний аналіз

3.Зручність

4.Економія коштів

5.Змога відстежувати стан техніки онлайн

В дипломній роботі також була зібрана статистика та проведений аналіз стосовно використання даної технології в світі, визначені були основні чинники які впливають на розвиток подібних технологій. Основними із яких є:

1. Рівень корупції

2. Низьке культурне становище населення

3. Політичне становище країни

4. Рівень загальної економіки

Були наведені приклади одних із самих популярних застосунків для дистанційної діагностики та обслуговування від різних компаній. Також були визначені галузі в яких використовується або може бути використана технологія дистанційної діагностики.

Проведено аналіз алгоритмів обробки даних кожен з яких використовується для конкретної задача та типу системи. Одними із основних є:

1. Аналіз аномалій

2. Кластеризація

3. Класифікація

4. Прогнозування

5. Машинне навчання

6. Глибинний аналіз даних

7. Обробка зображень

8. Нейромережевий аналіз

У практичній частині дипломної роботи, були визначені вимоги до прототипної системи:

1. Система має імітувати аналіз поточного стану автомобіля

2. Зчитувати дані з датчиків нашого умовного автомобіля

3. У разі віднайдених помилок відображати їх користувачу

4. Мати легкий та інтуїтивний інтерфейс

5. Система повинна бути розроблена з упором на її модифікацію в майбутньому

Для розробки прототипної системи було прийнято рішення використовувати мову програмування Python з бібліотекою PyQt5 та програму для створення елементів інтерфейсу QtDesigner.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

On-Board Diagnostics (OBD) – introduction to the Modes of Operation (Diagnostic Services) [Електронний ресурс] // x-engineer – Режим доступу до ресурсу: <https://x-engineer.org/on-board-diagnostics-obd-modes-operation-diagnostic-services/>.

Worrell C. Health Monitoring in Real Time [Електронний ресурс] / Carolina Worrell // RAILWAY AGE. – 2022. – Режим доступу до ресурсу: <https://www.railwayage.com/news/health-monitoring-in-real-time/>.

Proven, Deployed and IT-Secure Remote Maintenance System [Електронний ресурс] // ATLAS ELEKTRONIK Group. – 2021. – Режим доступу до ресурсу: <https://www.atlas-elektronik.com/newsroom/article/stay-at-home-vs-mission-critical-maintenance-abroad.html>.

Europe Vehicle Remote Diagnostic Market [Електронний ресурс] // Futuremarketinsights. – 2022. – Режим доступу до ресурсу: <https://www.futuremarketinsights.com/reports/europe-vehicle-remote-diagnostic-market>.

Newton J. The most dangerous countries in the world for 2021 revealed: Interactive map shows Libya, Syria and Afghanistan are riskiest, with Switzerland and Norway among the safest [Електронний ресурс] / Jennifer Newton // MAILONLINE. – 2020. – Режим доступу до ресурсу: <https://www.dailymail.co.uk/travel/travel_news/article-9006233/The-dangerous-countries-world-2021-revealed.html>.

Павленко В. М. ІНТЕЛЕКТУАЛЬНІ АГЕНТИ В РОЗРОБЦІ МУЛЬТИАГЕНТНОГО ПІДХОДУ ПРИ ОБСЛУГОВУВАННІ АВТОМОБІЛІВ [Електронний ресурс] / В. М. Павленко, М. Є. Хорін // Технічний сервіс агропромислового, лісового та транспортного комплексів. – 2020. – Режим доступу до ресурсу: <http://ts.khntusg.com.ua/index.php/ts/article/view/497/502>.

Opel OnStar vorgestellt – So funktioniert der Opel Assistent [Електронний ресурс] // Autohub. – 2022. – Режим доступу до ресурсу: <https://autohub.de/opel-onstar-vorgestellt-so-funktioniert-der-opel-assistent/>.

autocentre. Ford презентував технологію FordPass Connect [Електронний ресурс] / autocentre // Автоцентр. – 2017. – Режим доступу до ресурсу: <https://www.autocentre.ua/ua/opyt/tehnologii/ford-prezentovav-tehnologiyu-fordpass-connect-344636.html>.

Клапан В. Система HondaLink: Социальные сети в автомобиль [Електронний ресурс] / Владимир Клапан // Автоцентр. – 2012. – Режим доступу до ресурсу: <https://www.autocentre.ua/ua/opyt/tehnologii/sistema-hondalink-sotsialnye-seti-v-avtomobil-84489.html>.

МОБІЛЬНИЙ ДОДАТОК NISSANCONNECT SERVICES [Електронний ресурс] // Nissan – Режим доступу до ресурсу: <https://www.nissan.ua/ownership/nissan-connect-apps/nissan-connect-services-details.html>.

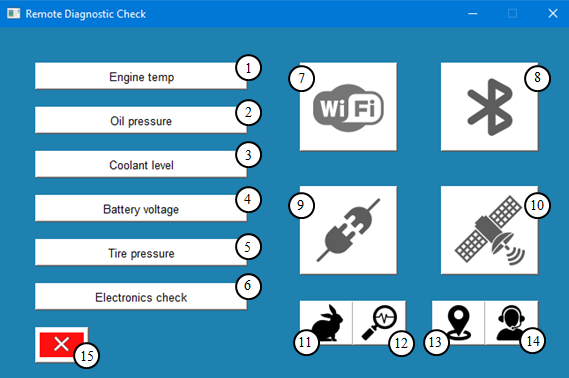
Lambert F. Tesla has a new self-diagnostic feature in its app [Електронний ресурс] / Fred Lambert // Electrek. – 2022. – Режим доступу до ресурсу: <https://electrek.co/2022/07/18/tesla-new-self-diagnostic-feature-app/>.

Аулін В. В. УДОСКОНАЛЕННЯ ПРОЦЕСУ ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ АВТОМОБІЛІВ З ВИКОРИСТАННЯМ МЕТОДІВ ДИСТАНЦІЙНОЇ ЕЛЕКТРОННОЇ ДІАГНОСТИКИ [Електронний ресурс] / В. В. Аулін, Д. Є. Панарін // Вісник Вінницького політехнічного інституту. – 2014. – Режим доступу до ресурсу: <https://visnyk.vntu.edu.ua/index.php/visnyk/article/view/913/912>.

Особливості моделювання та побудови інформаційної системи дистанційного моніторингу технічного стану транспортних засобів. // Вісник машинобудування та транспорту. – 2021. – №14. – С. 140–148.

Обов'язковий технічний контроль транспортних засобів [Електронний ресурс] // wikipedia.org – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B2%27%D1%8F%D0%B7%D0%BA%D0%BE%D0%B2%D0%B8%D0%B9_%D1%82%D0%B5%D1%85%D0%BD%D1%96%D1%87%D0%BD%D0%B8%D0%B9_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D1%8C_%D1%82%D1%80%D0%B0%D0%BD%D1%81%D0%BF%D0%BE%D1%80%D1%82%D0%BD%D0%B8%D1%85_%D0%B7%D0%B0%D1%81%D0%BE%D0%B1%D1%96%D0%B2>.

# ДОДАТОК A

****

Опис кнопок програми

1 – кнопка яка показує температуру двигуна

2 – кнопка яка показує тиск масла

3 – кнопка яка показує рівень охолоджуючої рідини

4 – кнопка яка показує заряд батареї

5 – кнопка яка показує тиск в шинах

6 – кнопка яка показує стан електроніки автомобіля

7 – кнопка підключення через WiFi

8 – кнопка підключення через Bluetooth

9 – кнопка підключення через кабель

10 – кнопка підключення через супутник

11 – кнопка швидкої діагностики

12 – кнопка повної діагностики

13 – кнопка виклику GPS мапи

14 – кнопка виклику підтримки

15 – кнопка відключення системи від авто

# ДОДАТОК Б

Нижче будуть наведені посилання з повним доступом до прототипної системи, як і вже скомпільований файл так і у вигляді коду.

Готовий прототип - <https://drive.google.com/file/d/1-e-J-t4vwdDkCDoizEPN49h9KXqfS6Sm/view?usp=sharing>

Прототип у вигляді коду - <https://docs.google.com/document/d/1hEj5gxSTckUydmXy7HI7gMZkkUXI0FzaxuR6rAEVH0g/edit?usp=sharing>