

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно інформаційних систем та програмної інженерії
(назва факультету)
Програмної інженерії
(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи)

Магістр

(освітній ступінь (освітньо-кваліфікаційний рівень))

На тему: Розробка програмної системи для діагностики
автомобілів на основі протоколу OBD-II

Виконав: студент (ка) VI курсу, групи СПм-61
спеціальності (напряму підготовки) Інженерія
програмного забезпечення

(шифр і назва спеціальності (напряму підготовки))

Топоровський В.В.

(підпис)

(прізвище та ініціали)

Керівник

Кінах Я.І.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Бойко І.В.

(підпис)

(прізвище та ініціали)

Рецензент

Литвиненко Я.В.

(підпис)

(прізвище та ініціали)

м. Тернопіль – 2019

АНОТАЦІЯ

Магістерська робота на тему «Розробка програмної системи для діагностики автомобілів на основі протоколу OBD-II» Топоровського Віталія Васильовича. – Тернопільський національний технічний університет імені Івана Пулюя, Факультет комп'ютерно-інформаційних систем і програмної інженерії, Кафедра програмної інженерії, група СПМ-61 // Тернопіль, 2019. С. – 90, рис. – 33, табл. – 6, слайдів. – 12, додат. – 4, бібліогр. – 22.

Метою дипломної роботи є дослідження та розробка технології для системи дистанційної діагностики автомобілів. Використовуючи для з'єднання із автомобілем діагностичний протокол OBD-II та зовнішній пристрій-адаптер ELM327. В даній магістерській роботі розглянуто головні засоби для розробки діагностичного програмного забезпечення, проведено їх аналіз та визначено найкращий засіб для реалізації поставленої задачі.

Методи та програмні засоби, використані при виконанні розробки системи: мова програмування C# та її бібліотеки, середовище розробки Visual Studio, база даних MySQL, інтерфейс програмування додатків Windows Forms та супутні програми.

Результатом роботи є готовий програмний засіб, який можуть використовувати як звичайні автолюбители так і працівники автомобільних сервісних центрів.

Ключові слова: АВТОМОБІЛЬНА ДІАГНОСТИКА, ЕЛЕКТРОННИЙ БЛОК КЕРУВАННЯ, АНАЛІЗ НЕСПРАВНОСТЕЙ, АВТОМАТИЗАЦІЯ, ПРОГРАМНА СИСТЕМА, ПРОТОКОЛ ПІДКЛЮЧЕННЯ.

ABSTRACT

Master's thesis on "Development of software system for vehicle diagnostics based on OBD-II protocol" by Vitalii Toporovskyi. - Ivan Pulyuy National Technical University of Ternopil, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, SPM-61 group // Ternopil, 2019.

C. - 90, fig. - 33, tab. - 6 slides. - 12, add. - 4, bibliography. - 22.

The aim of the thesis is to research and develop technology for remote vehicle diagnostics. Using OBD-II Diagnostic Protocol and ELM327 External Adapter to connect with cars. In this master's thesis the main tools for the development of diagnostic software are considered, their analysis is performed and the best means for the realization of the task is determined.

Methods and software used in the development of the system: C # programming language and its libraries, Visual Studio development environment, MySQL database, Windows Forms application programming interface and related programs.

The result of the work is a ready-made software tool that can be used by both ordinary motorists and working car service centers.

Keywords: AUTOMOBILE DIAGNOSTICS, ELECTRONIC CONTROL UNIT, FAILURE ANALYSIS, AUTOMATION, SOFTWARE, CONNECTION PROTOCOL.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення.

ПС – програмна система, комплекс програмного забезпечення.

ПК – персональний комп'ютер, робоча машина для розробки та виконання програм.

MFC - Microsoft Foundation Class.

CS - байт контрольної суми.

CAN – (Controller Area Network) інтерфейс передачі інформації.

K-Line – діагностична лінія зв'язку, встановлена між електронними блоками управління компонентами автомобіля і діагностичним роз'ємом.

CHECK ENGINE – це індикатор, який використовує комп'ютеризована система управління двигуном для вказівки на несправність.

ECU – Engine Control Unit.

PID – ідентифікатор процесу.

ELM327 – автомобільний адаптер, який використовується для діагностики.

ЕСКД – електронно-цифрова система керування двигуном.

ОBD-II – стандарт бортової діагностики автомобіля.

ЕБК – електронний блок керування.

ЗМІСТ

ВСТУП	8
1 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ.....	11
1.1 Аналіз вимог до предметної області	11
1.1.1 Аналіз предметної області	11
1.1.2 Постановка завдання.....	15
1.1.3 Пошук актантів та варіантів використання	17
1.2 Розгляд аналогів ПЗ	19
1.2.1 Програма OpenDiag.....	19
1.2.2 Програма ScanMaster	21
1.2.3 Програма FORScan.....	23
1.3 Вибір процесу розробки системи та архітектури	25
1.4 Вибір технологій та інструментів для розробки	26
1.4.1 Мова програмування С#.....	26
1.4.2 Windows Forms	28
1.4.3 Середовище розробки Visual Studio.....	30
1.4.4 База даних MySQL	31
1.4.5 Супутні програми.....	34
1.5 Специфікація обміну даними з ЕБК.....	36
1.5.1 Keyword Protocol 2000	36
1.5.2 Коди відповідей протоколу Keyword Protocol 2000	38
1.5.3 Функції протоколу Keyword Protocol 2000.....	39
1.5.4 Функція readEcuIdentification.....	41
1.5.5 Функція readDataByLocalIdentifier	42
1.5.6 Функція inputOutputControlByLocalIdentifier	45
2 ПРОЦЕС ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ ПРОГРАМИ	49
2.1 Структура програми та її використання	49

2.2	Загальна логіка програмної системи	54
2.3	Тестування програмної системи	55
2.3.1	Тест елементів керування.....	55
2.3.2	Тест відображення параметрів автомобіля у реальному часі.....	55
2.3.3	Тест функцій керування двигуном.....	56
2.3.4	Тест функції знаходження несправностей	57
2.3.5	Тест функції редагування допусків.....	57
2.3.6	Тест функції збереження документів.....	59
2.3.7	Тест часу нагріву двигуна	59
3	ОРГАНІЗАЦІЙНО-ЕКОНОМІЧНА ЧАСТИНА.....	60
3.1	Розрахунок норм часу на виконання науково-дослідної роботи	60
3.2	Визначення витрат на оплату праці та відрахувань	62
3.3	Розрахунок матеріальних витрат.....	63
3.4	Розрахунок витрат на електроенергію	64
3.5	Розрахунок суми амортизаційних відрахувань.....	65
3.6	Обчислення накладних витрат.....	66
3.7	Складання кошторису витрат та визначення собівартості	67
3.8	Визначення економічної ефективності і терміну окупності капітальних вкладень.....	67
3.9	Визначення витрат на супровід і модернізацію програмного продукту, уточнений аналіз ефективності вкладених інвестицій.....	69
4	ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	75
4.1	Охорона праці.....	75
4.2	Безпека в надзвичайних ситуаціях	79
	ВИСНОВКИ.....	85
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	87
	ДОДАТКИ.....	90

ВСТУП

Життя сучасної сім'ї неможливо уявити без особистого транспорту. Для того щоб транспортний засіб прослужив як можна довше, він потребує регулярного догляду і акуратної експлуатації. Але рано чи пізно настає момент, коли будь-які елементи автомобіля виходять з ладу і вимагають ремонту. Сучасний автомобіль - це складний організм, що містить в собі сотні, а іноді тисячі різних датчиків, призначених для контролю декількох систем. Щоб визначити ту чи іншу несправність, потрібно максимально точно і якісно провести процес діагностики за допомогою професіоналів. Подібні послуги часто займають багато часу і вимагають значних матеріальних вкладень в залежності від складності системи.

У зв'язку з цим стає актуальною самостійна діагностика за допомогою використання недорогого діагностичного обладнання і персонального пристрою з наявністю програмного забезпечення відповідного характеру. Якщо казати про автолюбителів, як про звичайних користувачів транспортного засобу, то можна припустити, що в більшості випадків їм не потрібно знати, як все налаштоване та працює, а що найголовніше, водій хоче експлуатувати справний автомобіль, який не буде його підводити. Також мають місце бути непередбачувані обставини, які так або інакше, впливають на появу цих несправностей, наприклад, дорожні транспортні пригоди. У таких випадках звичайний користувач потребує швидкого і не витратного вирішенні цієї проблеми. Тому і з'являються програми для швидкого аналізу різних систем автомобіля.

Використовуючи подібне програмне забезпечення користувач може проаналізувати дані, що приходять з датчиків автомобіля або отримати конкретну інформацію про помилки в системі. Таким чином автолюбитель в короткі терміни та без зайвих витрат має уявлення про ту чи іншу проблему, що сталася з його транспортним засобом, і може передати ці дані

представникам інстанцій, що займаються безпосередньо ремонтом, або ж усунути несправність самостійно. Крім відображення помилок, також може зчитуватись інформація, яку виробляють сигнали з датчиків швидкості і витрати палива, що виведені в салон автомобіля на спеціальний роз'єм для підключення маршрутного комп'ютера. Маршрутний комп'ютер дозволяє відобразити різні параметри: час у дорозі, час у русі, пройдений шлях, загальна витрата палива, витрата палива на холостому ходу, поточну швидкість, а також широкий спектр величин, розрахованих на їх основі (середня швидкість шляху, середня швидкість руху і т.д.).

Також в меню програмного засобу є можливість побачити на екрані ряд параметрів – що буде дуже корисним. Однак важливо не тільки побачити, а й вчасно зреагувати на позаштатну ситуацію. Тут вкрай корисною виявиться можливість програми постійно контролювати ряд параметрів і при необхідності оперативно сигналізувати про їх зміну.

При порушенні роботи системи охолодження або в дуже жаркий день можна не помітити, що температура двигуна перевищила норму, в цьому випадку є великий ризик "закипіти", що в кінцевому підсумку може обернутися серйозними проблемами для автомобіля. Додаток ні на секунду не втратить контроль над ситуацією і негайно попередить вас про перегрів двигуна звуковим сигналом. Точно так же він проконтролює і попередить про перевищення рекомендованої швидкості руху, про значення бортового напруги, про необхідність чергового техобслуговування.

Враховуючи те, що за останні десятиліття технологічний розвиток автомобілебудування набув швидкого розвитку і на сьогоднішній час майже все у транспортному засобі контролюється електронними системами. Із поширенням бездротового зв'язку та комунікаційних технологій можна надавати нові види послуг, що стосуються технічного обслуговування автомобілів, а саме дистанційної діагностики автомобіля. Це така послуга, що забезпечує можливість дистанційного ведення роботи по діагностиці

транспортних засобів. Існують оптимістичні очікування щодо потенціалу надання таких послуг в автомобільній галузі. Ці очікування ґрунтуються на припущенні, що дистанційна діагностика автомобіля може запобігти поломкам шляхом виявлення проблеми з транспортними засобами на ранній стадії. Однак справжній потенціал ще не розкритий, а виробники виконують боротьбу за пошук вигідних бізнес-моделей[1]. Метою даної магістерської роботи є розробка програмної системи для діагностики транспортних засобів за допомогою протоколу OBD-II.

Для досягнення поставленої мети потрібно вирішити такі завдання:

1. Проаналізувати область комп'ютерної діагностики автомобілів;
2. Провести порівняльний аналіз актуального ПО і діагностичних адаптерів;
3. На основі проведеного аналізу розробити макет - прототип майбутньої програми;
4. Розробити додаток для діагностики транспортних засобів згідно макету;
5. Протестувати додаток на працездатність і наявність помилок.

Очікуваним результатом роботи є розроблений додаток зі зручним і зрозумілим графічним інтерфейсом, що надає можливість стежити за динамікою параметрів ЕБК в реальному часі і дозволяти керувати деякими з них. Також додаток має володіти функціями редагування діапазону допустимих значень кожного параметра ЕБК, пошуку помилок і збереження звітів в форматі PDF.

1 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

1.1 Аналіз вимог до предметної області

1.1.1 Аналіз предметної області

В рамках даного проекту виконується розробка програмної системи для діагностики автомобілів на основі протоколу OBD-II.

Об'єктом дослідження виступає автоматизований процес, який дозволяє простим автолюбителям самостійно переглядати помилки, які виникли в процесі експлуатації транспортного засобу, а також отримувати інформацію із різноманітних датчиків автомобіля у зрозумілому вигляді.

Сучасні автомобілі мають в собі декілька комп'ютерів, вони наділені різноманітним програмним забезпеченням, що містить мільйони рядків програмного коду і виробляє гігабайти даних. Усі транспортні засоби, які випускаються в даний час, здатні оптимізувати свої показники, щоб зменшити витрату палива, виявляти помилки в двигуні, показувати найкращий та найкоротший шлях, враховуючи при цьому поточний рух транспорту та погодні умови.

Більшість легкових автомобілів і вантажівок оснащені бортовою діагностикою або портом OBD II, який забезпечує доступ до даних з блоку управління двигуном (ECU).

Для обробки всієї інформації, зібраної з пристрою OBD-II, нам потрібен спеціальний додаток для діагностики автомобіля. За допомогою такої програми ми можемо виконувати моніторинг інформації про транспортний засіб, відстежувати дані, які можна налаштувати на основі наших потреб.

Ще недавно на автомобілях існувала можливість вважати діагностичні коди несправностей за допомогою миготіння індикатора "CHECK ENGINE" на комбінації приладів. Для цього було необхідно виконати деякі маніпуляції

з висновками діагностичного роз'єму. Виробниками розроблялась власна система з індивідуальним набором сигналів. Це відбувалось лише до 1988 року Товариством автомобільних інженерів для стандартизації тестових сигналів. Подальший перегляд Агенцією з охорони навколишнього середовища призвели до сучасних стандартів, таких як OBD (бортова діагностика).

У сучасних автомобілів ця можливість залишилася тільки для деяких периферійних систем. Але у автовласників і сьогодні є можливість самостійно діагностувати основні системи автомобіля, тепер уже із застосуванням щодо недорогих і простих діагностичних приладів, що працюють за стандартом OBD-II. Щоб отримати інформацію з даного блоку, необхідно підключитись до спеціального зовнішнього пристрою. Специфікації OBD-II забезпечують стандартизований апаратний інтерфейс - 16-контактний (2x8) J1962 роз'єм, тому для більшості моделей автомобілів можна використовувати один і той ж самий зовнішній пристрій (Рис. 1.1).

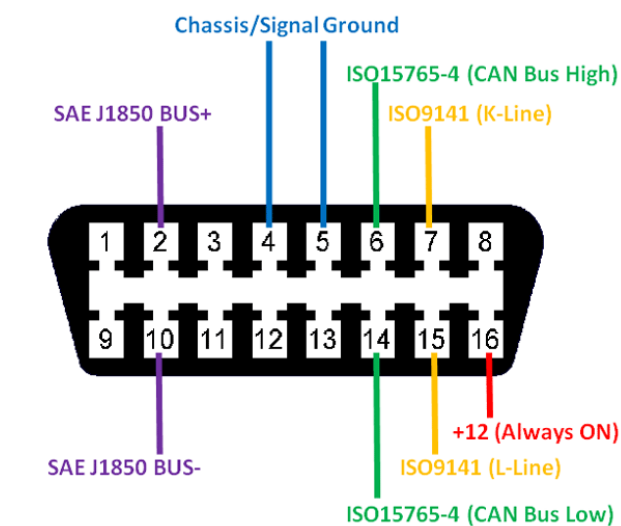


Рис. 1.1 – Графічний вигляд роз'єму OBD-II

Багато з таких зовнішніх пристроїв дозволяють автомобілістам не тільки самостійно провести діагностику при появі індикації несправності

двигуна "CHECK ENGINE", але і контролювати вибрані параметри роботи різних систем в русі. При включенні запалювання загоряються індикатори на комбінації приладів, а потім в залежності від логіки роботи системи частина з них гасне через кілька секунд, а частина з них гаснуть після того як двигун запуснитися.

OBD-II - це вбудована система діагностики, інтегрована в більшість сучасних транспортних засобів. Бортова система автомобіля обладнана вбудованими можливостями для діагностики, які відслідковують практично кожен компонент, який може вплинути на продуктивність транспортного засобу. Дана система може вирішувати екологічні проблеми, пов'язані із забрудненням та рівнями викидів шляхом моніторингу та коригування необхідних компонентів системи. Бортова система відповідає за виконання діагностичних процедур кожного компонента, щоб переконатися, що усе функціонує належним чином. В разі виникнення проблеми, система діагностики висвітлює попереджувальний сигнал на приладовій панелі, попереджаючи водія.

Діагностику також можна проводити "поза бортом", тобто отримуючи інформацію для зовнішнього аналізу. Такі системи зазвичай використовуються для проведення за допомогою складних діагностичних функцій, коли є час, місце та можливість. Інструментом для такої діагностики зазвичай виступає ноутбук чи мобільний пристрій. Даній інструменти зазвичай використовуються технічними працівниками в майстернях, щоб виокремити та визначити характер проблеми (Рис. 1.2).



Рис. 1.2 - Приклади двох типів діагностичних систем

Однак найчастіше саме поєднання бортової та бортової діагностики дає найбажаніші результати. Наприклад, 16 операцій самокорегування та збору даних повинні виконуватися на місці, тобто на борту, у той час як трудомісткі операції, такі як обробка даних та операції керування повинні здійснюватися з допомогою бортової системи діагностики.

Найпопулярнішими на сьогоднішній день діагностичні сканери на базі чіпа ELM327, але вони не мають власного ПО. Вони лише є сполучною ланкою, тобто адаптером, між автомобілем і комп'ютером, планшетом або смартфоном[2]. Зв'язок комп'ютера з адаптерами різних типів може здійснюватися або дротовим чином через порт COM або USB, або по бездротовому зв'язку через Bluetooth або Wi-Fi. Найбільш поширеними на сьогоднішній день є прилади, реалізовані на мікросхемі ELM327 або її емуляторах. Не завжди якщо на сканері є напис ELM327 це означає, що сканер виконаний саме на цьому чіпі. Є багато клонів, випущених на інших чіпах, але вони працюють нітрохи не гірше. Причина їхньої популярності в дешевизні виробництва і, відповідно, в доступній ціні. Швидкість зчитування даних трохи вище у сканерів на базі USB, але у Bluetooth є переваги такі як відсутність проводів і можливість підключення через смартфони та інші пристрої.

Функціонал адаптерів повністю залежить від ПО, встановленого на комп'ютері. Сьогодні таке ПО представлено досить широко як для персональних комп'ютерів, так і для мобільних пристроїв на базі операційних систем Android (планшети і смартфони), iOS (iPhone, iPad), Windows CE (штатні мультимедійні пристрої автомобілів, навігатори), Symbian (смартфони).

Підключення діагностичного приладу (адаптера) здійснюється в наступній послідовності: необхідно вимкнути запалювання, підключити прилад до діагностичного роз'єму OBD-II автомобіля, включити запалювання[3]. Далі попередньо встановлене на ноутбучі чи смартфоні

програмне забезпечення дозволить здійснити зв'язок з адаптером, і Ви зможете зчитувати параметри автомобіля з екрану Вашого гаджета.

Адаптер ELM327 підтримує такі протоколи стандарту OBDII: J1850 VPW, J1850 PWM, ISO9141-2, ISO14230-4 (KWP2000), ISO15765-4 (CAN).

Підтримувані моделі:

- моделі для ринку США з 1996 року;
- моделі для ринку Японії з 2002 року;
- моделі для ринку Європи з 2001 року і моделі з дизельними двигунами з 2004 року[3].

Однак деякі моделі з описаних вище можуть не підтримувати стандарт OBD II.

1.1.2 Постановка завдання

Завданням магістерської роботи є розробка програмного додатку для діагностики автомобілів на основі протоколу OBD-II. Дана програмна система повинна мати зручний та приємний інтерфейс який буде зрозумілим для любого користувача та нестиме в собі максимальну інформативність.

Програмний засіб повинен забезпечувати експрес-аналіз електрообладнання автомобіля та справно працювати в нормальних умовах. Проаналізувавши предметну область, можемо виділити перелік задач, які необхідно вирішити:

- Проаналізувати область комп'ютерної діагностики автомобілів;
- Провести порівняльний аналіз наявного ПЗ для діагностики;
- Розробка архітектури системи;
- Проектування бази даних;
- Реалізація системи;
- Тестування системи.

До основних функцій системи можна віднести :

- Відображення параметрів ЕБК в реальному часі;

- Побудова графіків;
- Пошук помилок;
- Редактор допустимих значень параметрів ЕБК;
- Справка по параметрах ЕБК;
- Збереження звітів у форматі PDF.

Дана предметна область є дуже актуальною та корисною в сучасному автомобільному світі, оскільки її цільовою аудиторією зможуть буди не лише працівники автосервісу чи фахівці, які займаються безпосередньо діагностикою електронних систем автомобіля, а й звичайні автовласники. Якщо ж порівнювати комп'ютерну діагностику із звичайною, то першу ми можемо розглядати як вершину діагностичних технологій. Вона дозволяє нам знайти майже всі несправності в автомобілі та не потребує великих затрат по часу.

Сутність постановки задачі полягає в розробці програмного забезпечення, яке буде складатися з трьох основних компонентів, а саме, OBD-II, адаптера ELM327 і портативного пристрою. На рисунку 1.3 ми можемо спостерігати архітектуру системи.

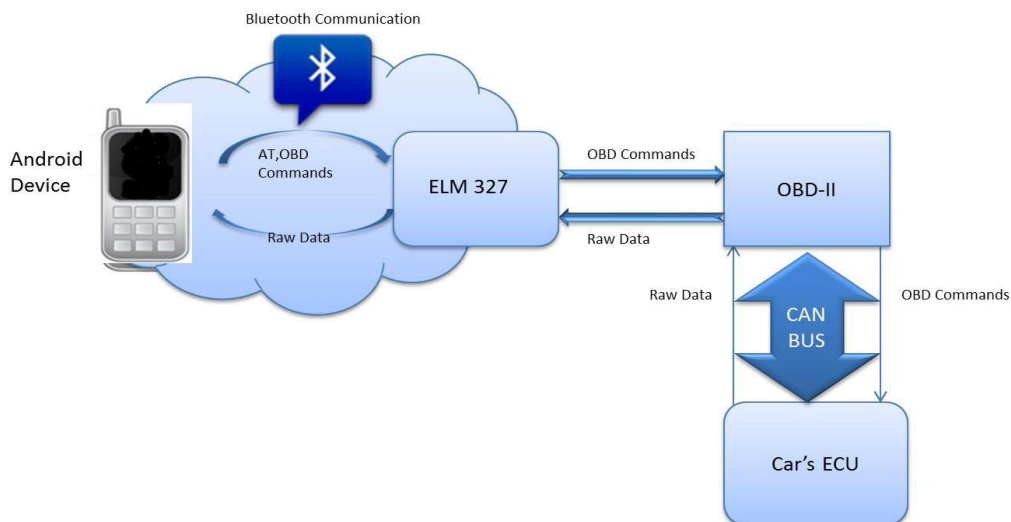


Рис. 1.3 – Архітектура системи для передачі даних між автомобілем та портативним пристроєм

Адаптер ELM327 розроблений з метою стати засобом зв'язку OBD портів і інтерфейсу стандарту RS232, простіше кажучи це послідовний порт, з'єднує комп'ютер і автомобіль[5]. Крім можливості автоматично визначати і працювати із одним з 9 OBD протоколів, ELM327 також підтримує високошвидкісну передачу даних, режим сну з низьким енергоспоживанням і протокол «J1939» для вантажівок і автобусів[6]. Також цей пристрій має велику кількість налаштувань, що дозволяє максимально адаптувати його під свої потреби.

Пристрій ELM327 підключається до ЕБК автомобіля через OBD-II і CAN шину. Портативний пристрій зв'язується з адаптером ELM327. Він посилає спеціальні команди до пристрою ELM327. Потім він відправляє свої команди OBD-II. У свою чергу, через даний протокол команди відправляються до ЕБК через CAN шину і отримує вихідні дані в шістнадцятковому форматі[7]. Ці дані відправляється на портативний пристрій за допомогою ELM327 адаптера.

Таким чином, ми зробили висновок про те, що тематика нашої програмної системи відноситься до суб'єктивного автоматизованого типу комп'ютерної діагностики, що використовує ноутбук в командному режимі діагностування. Розробку даного програмного додатку було прийнято вести з використанням адаптера ELM327, як найоптимальнішого варіанта для самостійної діагностики.

1.1.3 Пошук актантів та варіантів використання

Користуючись переліком завдань до виконання в рамках даної роботи, виконано пошук основних акторів системи та варіанти використання для них.

В роботі з системою буде присутній лише один актант – водій, тобто потенційний користувач програми. Власне він і може виконувати

дистанційну діагностику свого автомобіля. Розглянемо в таблиці нижче можливі варіанти використання для даного актанту.

Табл. 1.1 - Опис варіантів використання

1	Водій	Виклик справки	Так як деякі поняття, що стосуються діагностики автомобіля є дуже вузькоспеціалізованими, водію авто може знадобитись допомога в розумінні тих чи інших термінів.
2	Водій	Перегляд відомостей про програму	При необхідності користувач може переглянути інформацію про використовувану програму.
3	Водій	Редагування допусків	Даний параметр дозволяє змінити діапазон допустимих значень, щодо інформації із датчиків.
4	Водій	Зміна параметрів	Дозволяє здійснювати керування певними можливостями автомобіля
5	Водій	Пошук помилок	Здійснюється пошук помилок, що стосуються роботи автомобіля.
6	Водій	Обнулення помилок	При наявності помилок, дозволяє їх “скинути” та перевірити їх повторну появу.
7	Водій	Збереження у текстовий файл	Дозволяє зберегти інформацію про наявні помилки у текстовому файлі формату PDF.
8	Водій	Перегляд параметрів із датчиків	Виводить інформацію із присутніх в автомобілі датчиків у зрозумілому для користувача вигляді.
9	Водій	Вибір параметрів для перегляду	Дозволяє вивести інформацію лише з тих датчиків, які потрібні для аналізу.

10	Водій	Перегляд графіків	Виконується графічна візуалізація даних із необхідного датчика.
----	-------	-------------------	---

Варіанти використання нашої програми представлені на рисунку 1.4:

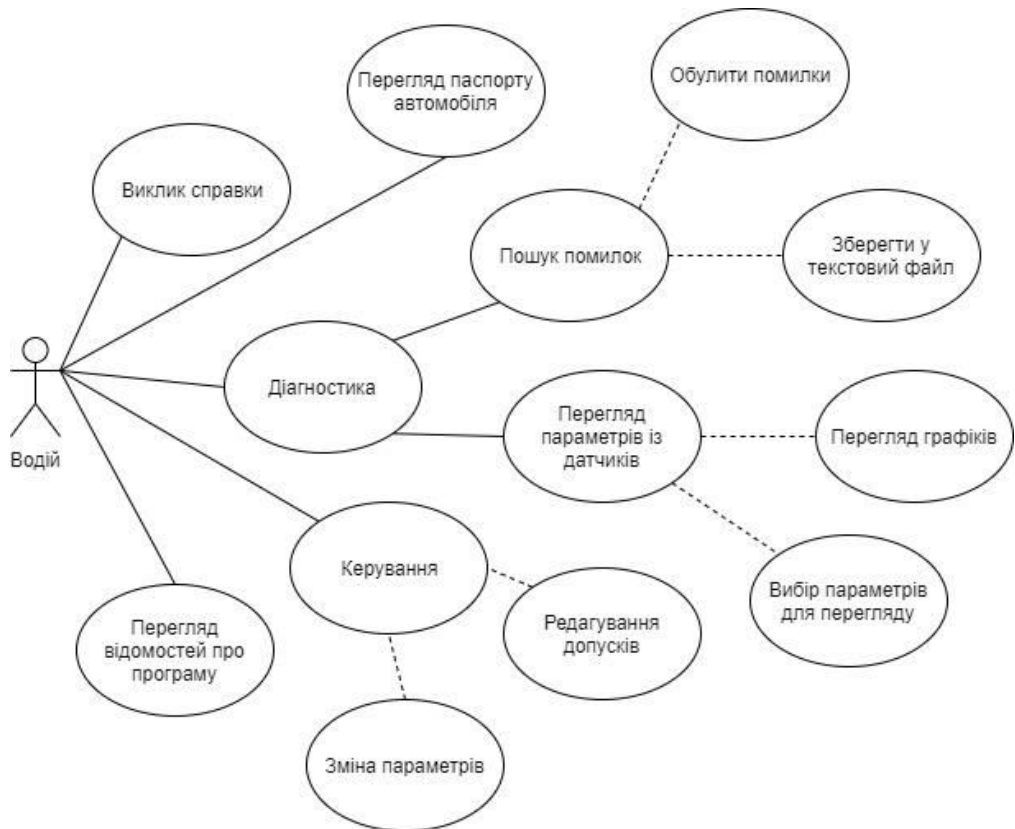


Рис. 1.4 – Ключові варіанти використання

1.2 Розгляд аналогів ПЗ

1.2.1 Програма OpenDiag

OpenDiag - програма для комп'ютерної діагностики автомобілів сімейства ВАЗ, ГАЗ, УАЗ і ЗАЗ. Власне, принцип роботи утиліти безпосередньо пов'язаний з читанням даних з електронного блоку управління

машини. Програма самостійно виводить на екран всі важливі параметри, необхідні для аналізу стану автомобіля, а також дозволяє зчитувати і очищати коди помилок.

Програма OpenDiag призначена для комп'ютерної діагностики автомобільних систем інжекторних автомобілів російського виробництва. Програма використовується для проведення технічного обслуговування і ремонту автомобілів в автосервісах, СТО або ж звичайним власником автомобіля при наявності відповідного обладнання (адаптера і персонального комп'ютера під керуванням Windows). Головне вікно програми зображено на Рисунку 1.5.

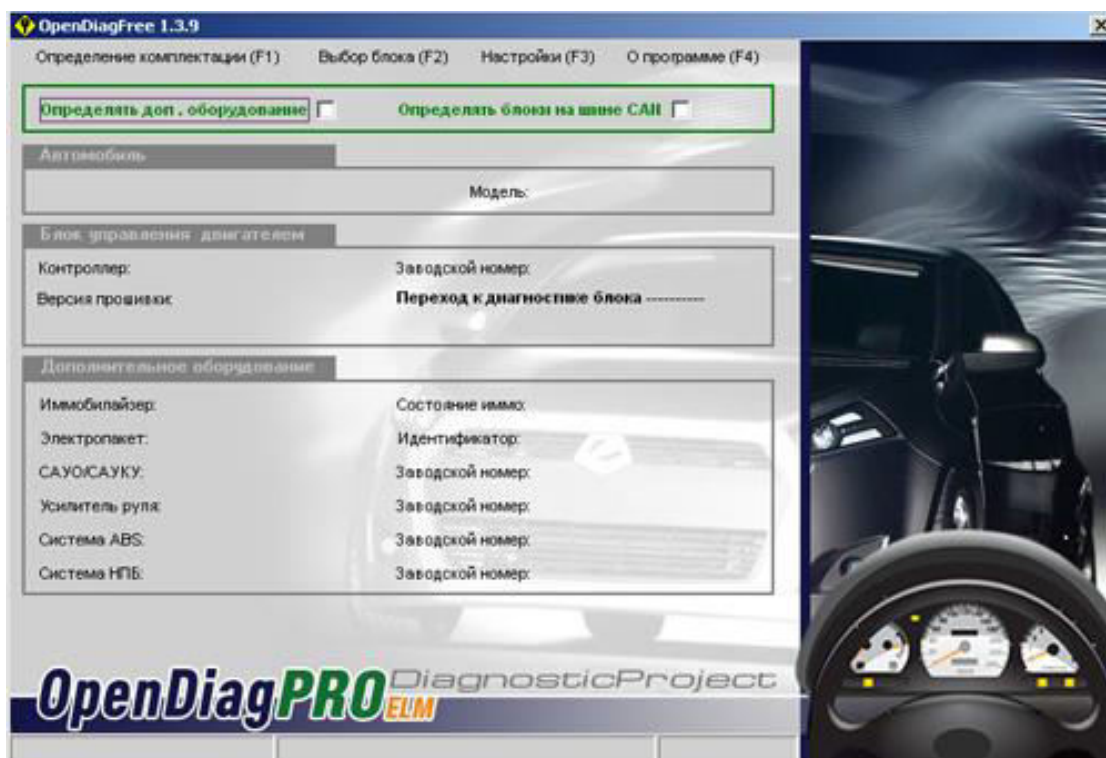


Рис. 1.5 – Головне вікно програми OpenDiag

Для запуску OpenDiag вам буде потрібно підключити адаптер (ELM372 або K-line) до ноутбука та автомобіля. Після цього з'явиться меню вибору типу пристрою. Далі відкриється головний екран управління діагностикою. У вкладці «Автомобіль» відображається модель машини, в розділі «Блок

управління двигуном» - виводиться версія прошивки, тип контролер і заводський номер, а в секції «Додаткове обладнання» - загальна інформація про другорядні пристроях. Як тільки програма визначить підключений адаптер, опція «Перехід до діагностики блоку» стане активною. Саме ця функція відповідає за відображення всіх головних параметрів автомобіля.

Окремої згадки заслуговує модель поширення програми. Спочатку проект замислювався як безкоштовний засоби діагностики для вітчизняних машин. Власне, фінальним релізом безкоштовного видання стала версія OpenDiag 1.4. Вона доступна всім користувачем і донині без будь-яких обмежень. У 2012 році розробники вирішили продовжити підтримку свого дітища, в результаті чого з'явилася платна версія під назвою OpenDiagPro. Формально вона пропонує додаткові опції і можливість підключення модулів іномарок, але весь основний функціонал і інтерфейс вона успадкувала від безкоштовної версії.

OpenDiag відома багатьом вітчизняним автолюбителям. В основному її використовують для виявлення несправностей і ремонту російських і українських машин в різних автосервісах і СТО.

1.2.2 Програма ScanMaster

Даний програмний продукт призначений для діагностики автомобілів за стандартами OBD-II/EOBD, які були розроблені спеціально для ELM327 чіпу, від компанії ELM Electronics і підтримує всі OBD-2 діагностичні протоколи зв'язку, такі як SAE J1850 (PWM & VPW), ISO15765, CAN, ISO9141 і ISO14230.

Це одна з найпопулярніших програм для діагностики автомобілів за допомогою сканерів на чіпі ELM327. Вирізняється з поміж інших універсальністю, великим списком можливостей, зручним і функціональним інтерфейсом, простотою використання. Програма здатна працювати з

адаптерами ELM327 через різні інтерфейси: ELM327 Bluetooth, ELM327 USB.

У поєднанні з адаптером ELM327, утиліта стає ідеальним рішенням для діагностики всіх OBD-II протоколів і підходить як для професійних ремонтних майстерень, так і для приватного використання. В доповнення до стандартних діагностичних можливостей, дана програма пропонує велику кількість додаткових опцій, заснованих на даних діагностичного інтерфейсу[8]. Головне вікно програми ScanMaster зображене на Рисунку 1.6.

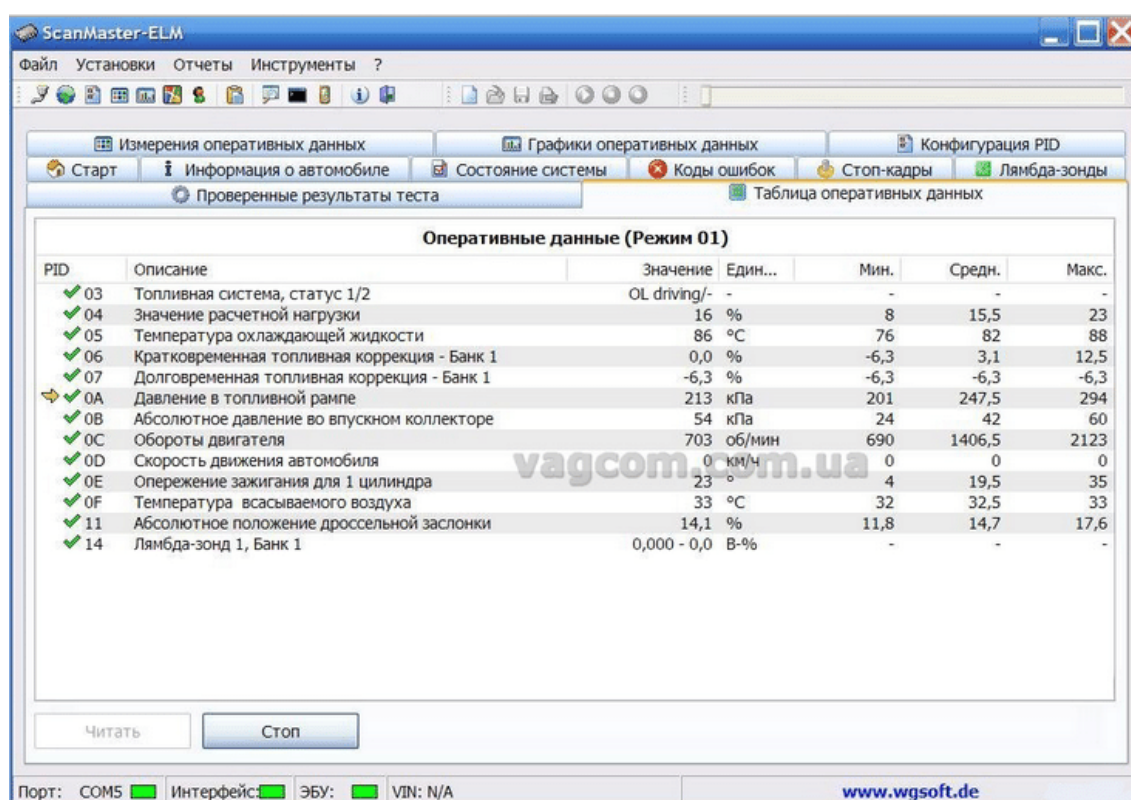


Рисунок 1.6 – Интерфейс програми Scanmaster

До основних можливостей даної програмної системи можна віднести:

- Моніторинг параметрів мотору в реальному часі (всі або за вибором);
- Побудова графіків залежностей (наприклад оберти двигуна-тиск в рампі);

- Перегляд стоп-кадрів системи;
- Перегляд і стирання помилок;
- Відображення статусу паливної системи: Open Loop або Closed Loop.

1.2.3 Програма FORScan

FORScan - програма для комп'ютерного діагностування автомобілів Ford, Mazda, Lincoln та Mercury. Вона виступає в якості засобу керування OBD-II адаптером, який підключається безпосередньо до автомобіля і передає дані на комп'ютер. З її основних можливостей варто відзначити: обробку даних з датчиків, визначення конфігурації бортової мережі, відображення важливих параметрів автомобіля, читання і скидання кодів помилок, проведення різних тестів і сервісних процедур. Вікно програми FORScan зображено на Рисунку 1.7.

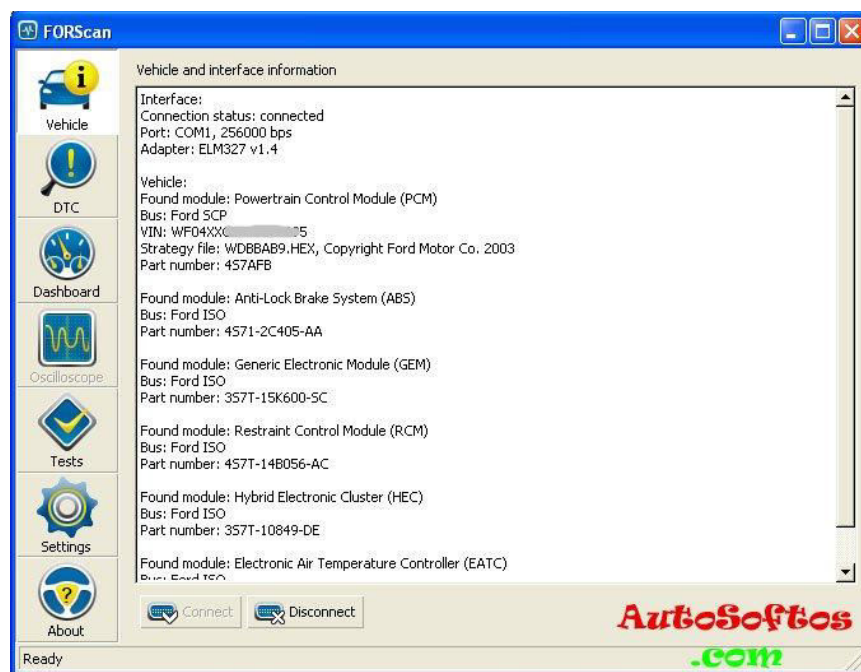


Рисунок 1.7 – Вікно програми FORScan

Після запуску програми і підключення до автомобіля з'являється основний екран з 8 вкладками в лівій колонці. Найперша опція (“Інформація про автомобіль і інтерфейсі”) надає інформацію по з'єднанню. Друга (“Читання кодів і помилок”) - виводить на екран всі поточні помилки і попередження. Третя (“Параметри”) - фіксує всі важливі дані в режимі реального часу. Причому за допомогою кнопки “Налаштування приладів” користувач може самостійно вибрати ті параметри, які будуть відображатися в загальній таблиці або на графіку. Четверта вкладка (“Тести на автомобілі”) використовується для тестування різних елементів. П'ята опція відкриває доступ до сервісних функцій, шоста - до меню конфігурації і програмування датчиків, сьома - до налаштувань програми, а восьма - до завантаження ключа розширеної ліцензії. Використовуючи всі перераховані вище засоби діагностики, вона може допомогти самостійно виявити проблеми і несправності без залучення фахівців з автосервісу.

FORScan є нестандартним OBD-II сканером, який підтримує лише певні автомобільні марки. За словами розробників програми, подібна спеціалізація сприяє підвищенню точності діагностики і дає деяку перевагу перед звичайними сканерами[9].

FORScan може виявляти модулі, які не бачать звичайні OBD-II сканери, може показати вам коди несправностей, які ніколи не покажуть OBD-II сканери, може показати вам параметри (PID), які не підтримуються OBD-II сканерами, виконувати специфічні тести і сервісні процедури. Іншими словами, FORScan може робити те, що не можуть робити звичайні OBD-II сканери.

До основних можливостей програми належать:

- Визначення конфігурації бортової мережі автомобіля;
- Читання і скидання кодів помилок всіх модулів;
- Зчитування даних з датчиків модулів;
- Виконання діагностичних тестів;

- Виконання сервісних процедур.

Підтримуються автомобілі будь-яких марок Ford, Mazda, Lincoln, Mercury, які можуть бути прочитані підтримуваними адаптерами. Це 99% моделей після 2000 року і близько 92% моделей після 1996 року (коли стандарт OBD-II був прийнятий в США).

1.3 Вибір процесу розробки системи та архітектури

Здійснивши глибокий аналіз вимог до системи ми вирішили використовувати шаблон проектування Model-View-Controller (MVC). MVC паттерн забезпечує плавний і гнучкий поділ компонентів в системі шляхом ділення програми (Рис. 1.8).

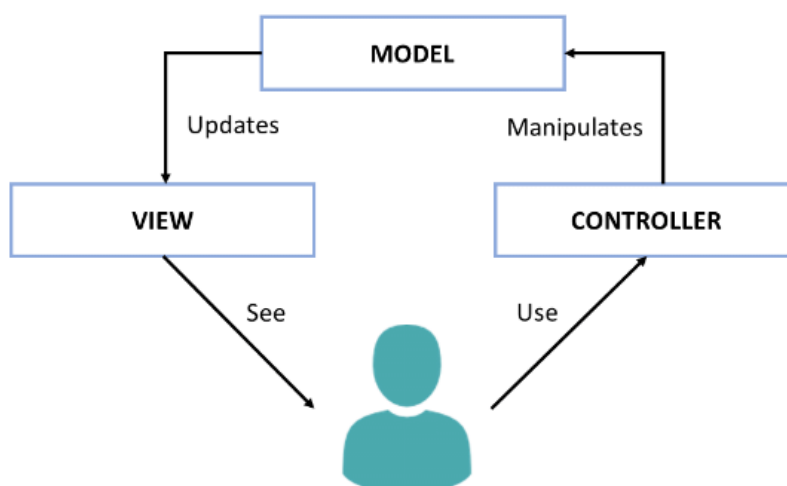


Рис. 1.8 - Взаємодія основних частин шаблону MVC

Даний шаблон розділяє роботу додатку на три окремі функціональні ролі: модель даних (model), призначений для користувача інтерфейс (view) і керуючу логіку (controller). Таким чином, зміни, що вносяться в один з компонентів, надають мінімально можливий вплив на інші компоненти[10].

Контролер перехоплює подію ззовні і відповідно до закладеної в нього логіки, реагує на цю подію змінюючи Модель, за допомогою виклику відповідного методу. Після зміни Модель використовує подію про те що вона

змінилася, і всі підписані на це події Представлення, отримавши його, звертаються до Моделі за оновленими даними, після чого їх і відображають. В даному паттерні модель не залежить від представлення або керуючої логіки, що робить можливим проектування моделі як незалежного компонента і, наприклад, створювати кілька представлень для однієї моделі.

Вперше цей шаблон був застосований в фреймворку, що розроблявся для мови Smalltalk в кінці 1970-х років. З цього моменту він відіграє основну роль в більшості фреймворків з призначенням для користувача інтерфейсом. Він докорінно змінив погляд на проектування додатків.

1.4 Вибір технологій та інструментів для розробки

Для досягнення поставленої задачі було прийнято рішення вести розробку в середовищі Visual Studio 2017 на мові C# і використовувати Windows Forms в як тип програми. Visual Studio - це інтегроване середовище розробки, що дозволяє працювати практично з будь-яким кодом, а також виконувати налагодження та складання додатків на базі всіх популярних операційних систем[11]. Варто відзначити, що в Visual Studio забезпечена висока підтримка мови програмування C#. Це об'єктно-орієнтована мова, що має безліч функцій, які допомагають підтримувати безпеку і надійність продукту, що розробляється. Також до цих функцій відноситься прибирання сміття, обробка виключень, а також сувора типізація мови, яка не дозволяє використовувати неініціалізовані змінні і виходити за межі масивів. Visual Studio прив'язана до загальнодоступного сховища, в якому зберігаються різні бібліотеки усіляких тематик. Деякі з них, а саме IO.Ports, призначена для управління портом і iTextSharp (для створення і обробки звітів в форматі PDF), вони і були використані в ході розробки даного програмного продукту.

1.4.1 Мова програмування C#

C# - це об'єктно-орієнтована мова програмування. Проста і в той же час потужна мова програмування, яка дозволяє швидше, аніж будь-яка інша мова програмування створювати багатфункціональні програми. Вона поєднує в собі найкращі ідеї сучасних мов програмування Java, C ++, Visual Basic і т.д. C# вирізняється надійністю та елегантністю і надає можливості для програмування практично будь-якого типу програмного забезпечення або компонента не тільки для платформи Windows, але і для iOS та Android.

Мова програмування C# була створена відносно не давно - в кінці 1998 року в групі фахівців компанії Майкрософт. Метою створення мови програмування C# була можливість створення різних програм для платформи Microsoft.NET. Дана прихильність платформи Microsoft.NET і мови C # була усунена в подальшому, але сама платформа .NET у всіх версіях в своєму складі містить компілятор кодів даної мови програмування, що дозволяє запускати додатки без установки додаткового програмного забезпечення.

Найменування мови програмування C# пов'язана з тим, що синтаксис дуже нагадує такий C++. Взагалі, нова мова програмування C# включила в собі багато позитивних рис своїх попередників - Delphi, C++, Java і т.д. При цьому з мови програмування C# вилучені проблемні алгоритми, які показали свою нестабільність в перерахованих мовах програмування. Також для повноцінного ознайомлення з даною мовою програмування цілком підходять курси допомоги по C# (відео, статті, тренінги). Мова C# не важка для вивчення, якщо розробник раніше активно займався програмуванням в середовищі C і C++.

C# являється безпечною мовою програмування типу. Вона не дозволяє перетворення типів, що могли б призвести до втрати певних даних чи створити інші проблеми, саме це дозволяє розробникам писати безпечний код і власне зосереджується на написанні ефективного коду.

Мова C# актуальна в першу чергу тому, що дозволяє більш раціонально створювати популярні на сьогодні Інтернет-додатки. Мова C# тісно інтегрована з мовою XML, різними веб-технологіями. Мова програмування C# включила в собі переваги мови Java і C ++, що й обумовлює популярність даної мови серед розробників.

1.4.2 Windows Forms

Windows Forms - це набір керованих бібліотек в .NET Framework, призначений для розробки багатих клієнтських додатків. Це графічний API для відображення даних та керування взаємодією користувачів з простішим розгортанням та кращою безпекою в клієнтських додатках.

Windows Forms пропонує широку клієнтську бібліотеку, що надає можливість для доступу до власних елементів графічного інтерфейсу Windows та графіків з керованого коду[12]. Він побудований з керованою подіями архітектурою, подібною клієнтам Windows. Windows Forms подібний до бібліотеки Microsoft Foundation Class (MFC) при розробці клієнтських додатків. Він пропонує обгортку, що складається з набору класів C++ для розробки програм Windows. Однак він не забезпечує рамки програми за замовчуванням, як MFC.

Кожен елемент керування в додатку Windows Forms - це конкретний екземпляр класу. Макетом управління в графічному інтерфейсі та його поведінкою керуються за допомогою методів. Windows Forms надає різноманітні елементи керування, такі як текстові поля, кнопки та веб-сторінки, а також параметри створення спеціальних елементів керування. Він також містить класи для створення пензлів, шрифтів, піктограм та інших графічних об'єктів (наприклад, лінії та кола).

Windows Forms Designer - це інструмент у Visual Studio.NET, який використовується для вставлення елементів керування у форму та упорядкування їх відповідно до потрібного макета з можливістю додавання

коду для обробки подій, які реалізують взаємодію користувачів. Табличні дані, які пов'язані з XML, базою даних тощо, можуть відображатися за допомогою керування DataGrid View у вигляді рядків і комірок.

Налаштування програми - це ще одна особливість Windows Forms для створення, зберігання та підтримки інформації про стан виконання у формі XML, яка може бути використана для отримання бажаних налаштувань користувача, таких як позиції панелі інструментів та останні останні використовувані списки. Ці параметри можна повторно використовувати в майбутньому додатку. На рисунку 1.9 зображено конструктор форм Windows Forms.

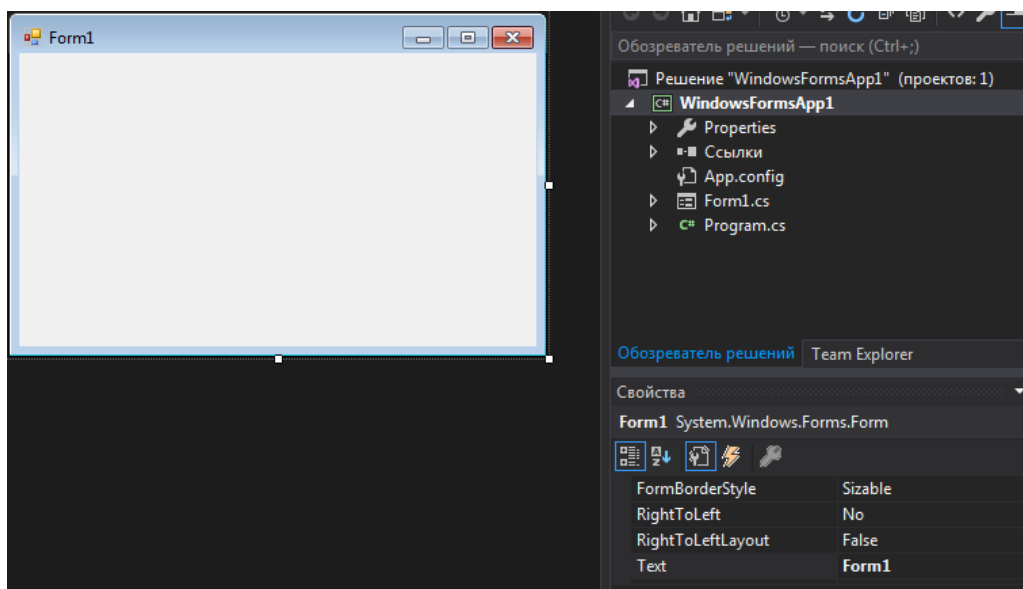


Рис. 1.9 – Конструктор форм

Система презентації Windows (WPF) - це найновіша технологія візуалізації користувацьких інтерфейсів у програмах Windows GUI з такими функціями, як підтримка 2D/3D, інтерактивна візуалізація даних та читаність вмісту. Він покладається на DirectX, а не на GDI (графічний інтерфейс пристрою) для надання моделі програмування, де інтерфейс користувача відокремлений від бізнес-логіки. Однак, маючи можливість

взаємодії з WPF (де це потрібно), Windows Forms є хорошим вибором для програм, які не потребують багатогалузевого графічного інтерфейсу та інших функцій WPF, таких як шаблони даних / керування, друкарські та текстові функції візуалізації

1.4.3 Середовище розробки Visual Studio

Visual Studio .NET – це інтегроване середовище розробки (IDE), яке можна використовувати для розробки консолей, графічних інтерфейсів користувача (GUI), Windows Forms, веб-служб та веб-додатків. Visual Studio використовується для запису нативного коду та керованого коду, що підтримується Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight. Редактор коду Visual Studio .NET підтримує IntelliSense та рефакторинг коду, тоді як інтегрований налагоджувач Visual Studio .NET підтримує налагодження як вихідного, так і машинного рівня. Visual Studio .NET включає інші вбудовані інструменти, наприклад дизайнер форм, який корисний при створенні програм GUI; веб-дизайнер, який створює динамічні веб-сторінки; дизайнер класів, який використовується для створення користувацьких бібліотек, і конструктор схем для підтримки бази даних.

На рисунку 1.10 зображено робочу область середовища Visual Studio.

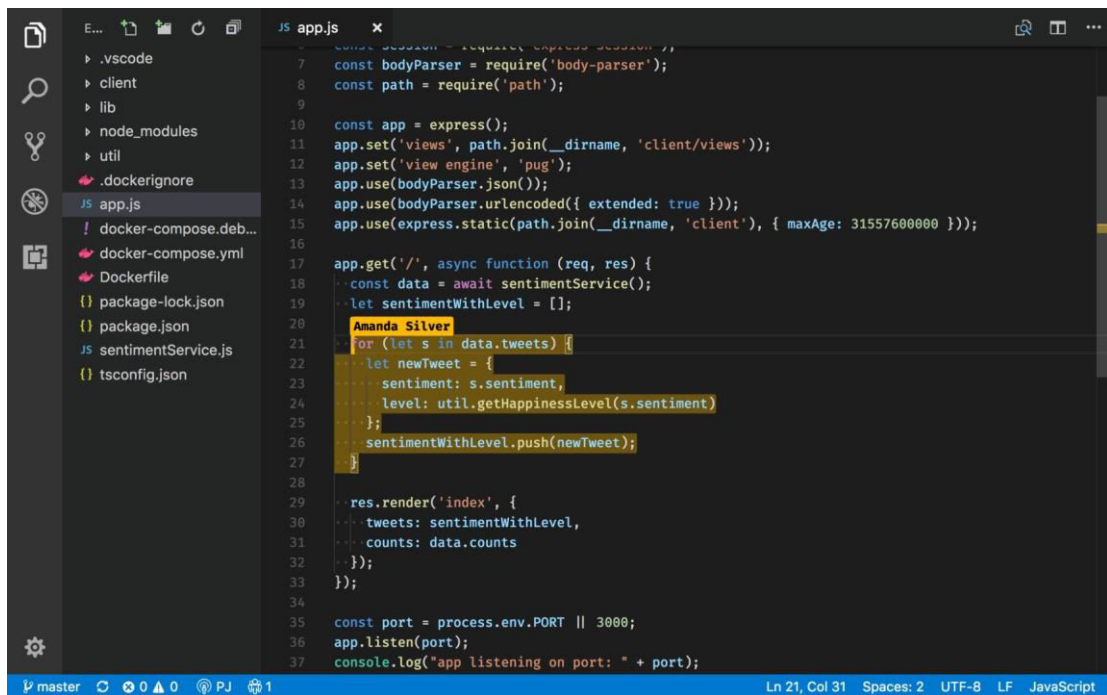


Рис. 1.10 – Робоча область середовища Visual Studio

Вбудовані мовні служби пропонують підтримку різних відомих мов (наприклад, C#, C++, VB та F#), тоді як мовні служби можуть встановлюватися окремо для підтримки мов, включаючи M, Python та Ruby, серед інших підтримуваних мов.

Існують версії Visual Studio, які підтримують окремі мови, які є економічно вигідним рішенням для початківців, яким потрібен лише Visual Basic, або для просунутих користувачів, які працюють над програмами швидкої обробки, але вони не зацікавлені в низько-швидкісному Visual Basic.

1.4.4 База даних MySQL

MySQL - це система управління базами даних, яка використовується для зберігання та отримання всієї інформації вашої програми. MySQL - це система управління реляційними базами даних з відкритим кодом. Вона працює як сервер і дозволяє багатьом користувачам керувати та створювати численні бази даних. Це центральний компонент у LAMP стеку програмного забезпечення для веб-додатків з відкритим кодом, який використовується для

створення веб-сайтів. LAMP розшифровується як Linux, Apache, MySQL та PHP.

Щоб отримати дані з бази даних, додаток запускає SQL запити для динамічного генерування вмісту. SQL розшифровується як "Структурована мова запитів" і є мовою програмування, яка зазвичай використовується для запитів баз даних.

На рисунку 1.11 зображено принцип роботи клієнт-сервер.

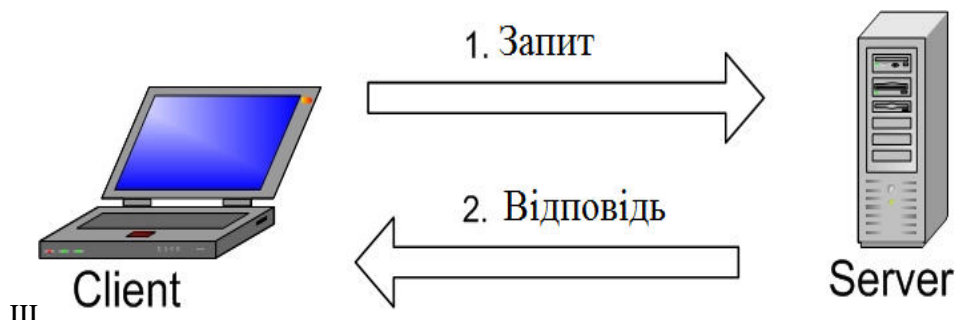


Рис. 1.11 - Принцип роботи бази даних

Наша програмна система містить три різні бази даних, вони представлені чотирма різними класами в пакеті моделі. Один з класів, а саме клас бази даних є суперкласом для трьох інших підкласів: два локальних класи бази даних і розподіленого класу бази даних.

Перша локальна база даних складається з двох таблиць. Одна з них містить поточну інформацію про автомобіль (Рис 1.12). Друга таблиця містить інформацію про загальний стан автомобіля та дані із 17 датчиків. Кожна таблиця має первинний ключ, який має автоматично згенерований номер. Завдання бази даних полягає в тому, щоб зберігати всі вихідні дані в базі даних MySQL. Ці дані витягуються з черги FIFO, переданої в якості аргументів в класі, коли конкретизується зовнішня додаткова інформація, така як номер порту, driverID, carID і т.д., також передається в якості аргументів до цього класу.

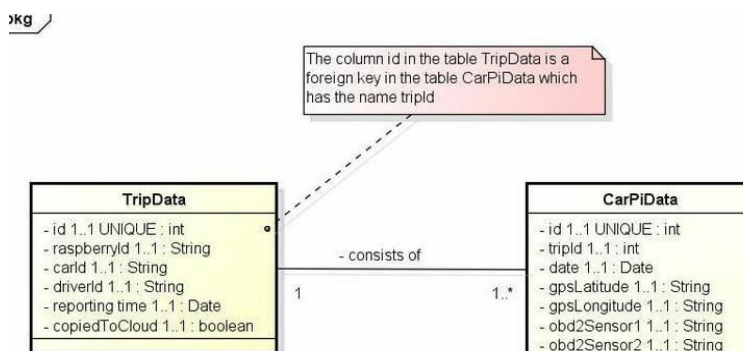


Рис. 1.12 – Концептуальна модель локальної бази даних

Друга локальна база даних (LocalDatabase) має завдання зберігання інформації про повну поїздку, наприклад, пробіг, витрата палива (Рис. 1.13). Додаткова інформація, така як номер порту, драйвери, ідентифікатор автомобіля і т.д. також надсилаються в якості аргументів цього класу.

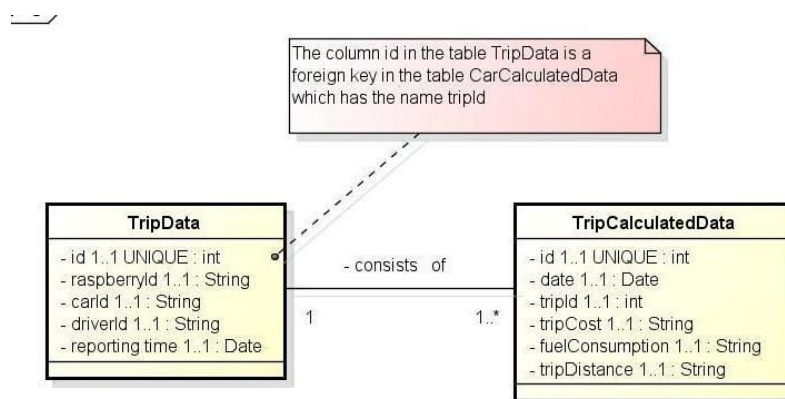


Рис. 1.13 – Концептуальна модель локальної бази даних

Сервер бази даних містить систему керування базами даних, яка обробляє дані моделі. Вона забезпечує стійкість, узгодженість і цілісність даних для моделі даних. Будь-який запит який зроблено з клієнтського додатку або люба відповідь, що доставляється з віддаленого пристрою буде

пересилатися через віддалений сервер, щоб зберегтися в базі даних сервера. Для підвищення продуктивності і керувати кількома операціями з базами даних, був обраний модульний тип розвитку, тобто користуючись наданими властивостями, такими як збережені процедури і тригери. Деякими з переваг, одержуваних від використання цих методів є зниження мережевого трафіку, а також підвищення загальних типів цілісності і збереження послідовності. Такі властивості мають вирішальне значення для системи з урахуванням того, що база даних діагностики, розташованої на сервері баз даних, є одним з найважливіших компонентів пропонованої архітектури. Вона вимагає великої інфраструктури, щоб керувати великою кількістю даних, зібраних від автомобіля протягом всього терміну служби. Крім того, цілісність бази даних повинна бути збережена в зв'язку з важливою місією збору даних діагностики, тобто утворює основу для майбутнього розвитку діагностики і прогностичних застосувань.

1.4.5 Супутні програми

В ході розробки були використані додаткові програми з метою вивчення процесу відправлення запитів на ЕБК через діагностичний кабель і моніторингу відповідей. Це програми Terminal 1.9b і Advanced Serial Port Monitor. Terminal 1.9b дозволяє встановити параметри з'єднання з пристроєм, після чого на нього можна відправляти запити відповідно до протоколу обміну. Вона є корисною і невеликою за розмірами програмою для емуляції послідовного порту (RS232), також, дана програма дозволяє передавати макрокоманди[13]. Головне вікно програми зображено на риснку 1.14.

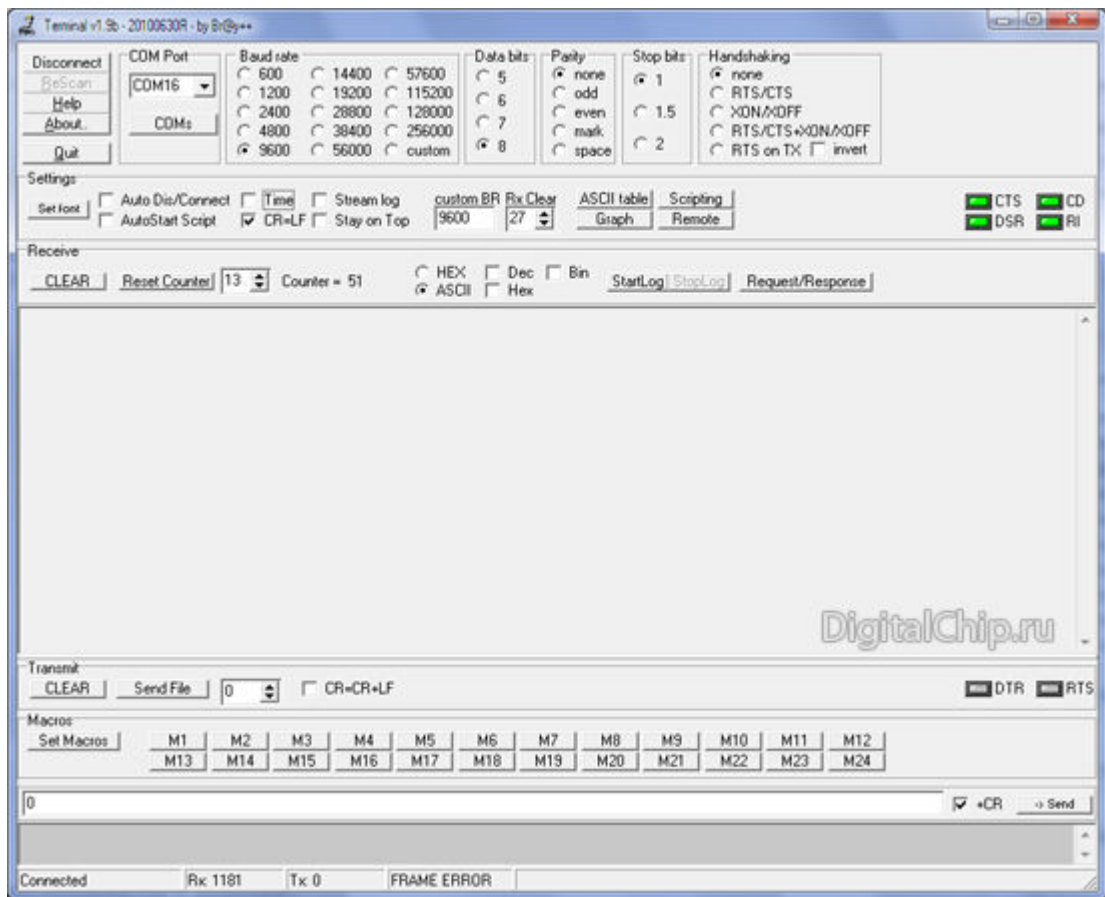


Рис. 1.14 – Головне вікно програми Terminal 1.9b

В ході використання цієї програми з'ясувалося, що кожна команда, відправлена на пристрій, дає відповідь, який закінчується символом ">". Програма Advanced Serial Port Monitor призначена для відстеження повідомлень, що приходять на пристрій, і відповідей на них. Ця програма дозволяє перевірити потік даних через COM-порти комп'ютера[14]. Як видно з назви програмного забезпечення, ця програма може працювати як монітор послідовного порту. Вона підтримує повний дуплексний режим, гнучке регулювання параметрів, виведення отриманих даних у файл, автоматичний та ручний режими, безкоштовне джерело даних, моделювання послідовних пристроїв, плагіни та безліч іншого.

Advanced Serial Port Monitor дозволяє працювати з будь-яким обладнанням, що працює з любым послідовним інтерфейсом RS232/RS485/RS422. Ви можете встановити монітор послідовних портів для

контролю обміну даними між послідовними пристроями, підключеними до послідовного порту та будь-яким додатком Windows. Це може бути як вимірювальний прилад, промисловий контролер чи радіоаматорська радіостанція та інший персональний комп'ютер, підключений через кабель z-модем. Також додаток Advanced Serial Port Monitor не вимагає встановлювати додаткове обладнання або програмне забезпечення для аналізу протоколу.

Після кількох запитів, надісланих на пристрій за допомогою програми Terminal, стало відомо, що формат відправленого повідомлення відрізняється від того, що було написано в поле введення. Наприклад, запит "AT Z" на самому справі відправляється як "AT Z\r\n", тобто в кінці додається символ розриву рядків і переводу каретки. Результати використання цих додаткових програм стали основоположними у створенні алгоритму відправки, прийому і обробки даних.

Робоче вікно програми зображено на рисунку 1.15.

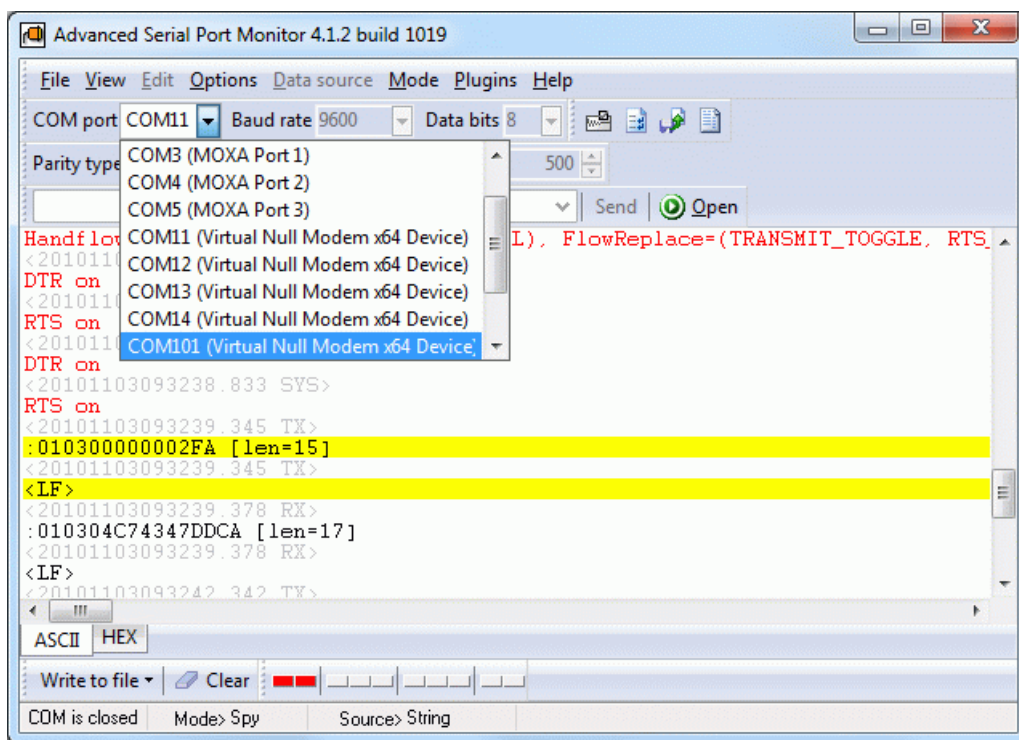


Рис. 1.15 – Робоче вікно програми Advanced Serial Port Monitor

1.5 Специфікація обміну даними з ЕБК

1.5.1 Keyword Protocol 2000

Взаємодія програмного забезпечення даного додатку та ЕБК автомобіля здійснюється за правилами протоколу «Keyword Protocol 2000».

Цей документ базується на наступних міжнародних стандартах:

- ISO 14229;
- ISO 14230-1;
- ISO 14230-2;
- ISO 14230-3;
- ISO 14230-3G;
- SAE J1930;
- SAE J2012.

Даний документ створений для конкретної реалізації програмного забезпечення контролера електронної системи керування двигуном[15].

На рисунку 1.16 представлена передача даних по послідовному каналу на автомобілях марки ВАЗ.

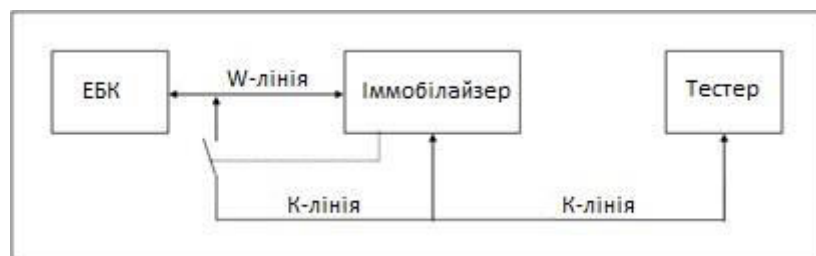


Рис. 1.16 - Передача даних по послідовному каналу

К-лінія призначена для ініціалізації і обміну пакетами даних між ЕБК і діагностичним адаптером, L-лінія не задіяна. На початковому етапі, при наявності імобілайзера (автомобільна протиугінна система), підключення

відбувається через W-лінію, і тільки безпосередньо після завершення процесу іммобілізації пристрій підключається безпосередньо до K-лінії, при чому іммобілайзер ніяк не впливає на обмін даними. Якщо автомобіль не укомплектований іммобілайзером, то зв'язок пристрою і ЕБК здійснюється відразу через K-лінію.

Формат повідомлень, які відправляються до ЕБК являє собою команду з 3 частин:

- заголовок (Header);
- біти даних (Data bytes);
- контрольна сума (Checksum);

На рисунку 1.17 зображено протокол, який включає в себе два типи заголовків.

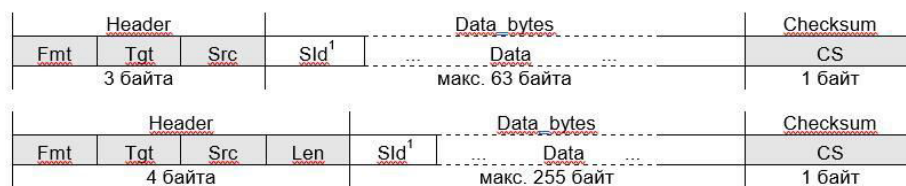


Рис. 1.17 - Заголовки протоколу

Пояснення до малюнка:

Fmt - байт формату повідомлення;

Tgt - байт адреси приймача повідомлення;

Src - байт адреси джерела повідомлення;

Len - байт довжини повідомлення при заголовку, що складається з 4 байт;

1 - байт типу переданих даних, відноситься до байтам даних;

CS - байт контрольної суми.

На рисунку 1.18 представлені позитивні та негативні відповіді на запити від пристрою.

Діагностичний тестер	Контролер
<Ідентифікатор> Запит[...]	<Ідентифікатор> Позитивна Відповідь[...]
<Ідентифікатор> Запит[...]	<Ідентифікатор> Негативна Відповідь[КВ]
<Ідентифікатор> Запит[...]	<Ідентифікатор> Позитивна Відповідь[...]

Рис. 1.18 - Відповіді на запити пристрою

Ідентифікатор - байт, що визначає формат і тип даних, а КВ - код відповіді, який визначає подальші дії, в разі отримання негативної відповіді.

1.5.2 Коди відповідей протоколу Keyword Protocol 2000

Формат відповіді протоколу Keyword Protocol 2000 на запит складається з ідентифікатора негативної відповіді "7F", ідентифікатора викликаємої функції та коду помилки. Нижче представлені негативні коди відповідей на різні запити при обміні даними з ЕБУ:

- **generalReject (10)** - даний код відповіді означає, що запит відхилений, але специфікація причини відхилення не визначена;
- **serviceNotSupported (11)** - даний код відповіді означає, що запит не може бути виконаний через відсутність підтримки запитів даного виду;
- **subFunctionNotSupported-invalidFormat (12)** - даний код відповіді означає, що запит не може бути виконаний у зв'язку з тим, що приймач не підтримує дані аргументи або їх формат байт містить помилки;
- **busy-RepeatRequest (21)** - даний код відповіді означає, що приймач сильно завантажений і не може виконати запит;

- requestOutOfRange (31) - даний код відповіді означає, що запит не може бути виконаний тому, що значення байт даних вийшло за рамки допустимого

діапазону;

- transferAborted (71) - даний код відповіді означає, що передача даних була перервана за невідомих причин, і клопотання не буде виконаний пізніше;

- blockTransferDataChecksumError (72) - даний код відповіді означає, що контрольна сума даних прийнятого повідомлення не збігається з очікуваною.

1.5.3 Функції протоколу Keyword Protocol 2000

Обмін даними між ЕБУ і діагностичним кабелем відбувається за допомогою звернення до однієї або кількох функцій протоколу Keyword Protocol 2000:

- startCommunication (запит: 81, позитивна відповідь: C1) - дана функція призначена для ініціалізації обміну даними між ЕБУ і діагностичним кабелем, відповідь на даний запит не може бути негативним;

- stopCommunication (запит: 82, позитивна відповідь: C2) - дана функція призначена для завершення обміну даними між ЕБУ і діагностичним кабелем в разі успішного завершення функції startCommunication;

- startDiagnosticSession (запит: 10, позитивна відповідь: 50) - дана функція призначена для початку сеансу діагностичного обміну даними між ЕБУ і діагностичним кабелем, може бути викликана тільки після успішної ініціалізації;

- stopDiagnosticSession (запит: 20, позитивна відповідь: 60) - дана функція призначена для завершення поточного діагностичного сеансу обміну даними, може бути викликана тільки після дозволу сесії обміну даними і діагностики;

- `ecuReset` (запит: 11, позитивна відповідь: 51) - дана функція призначена для скидання ЕБК;
- `clearDiagnosticInformation` (запит: 14, позитивна відповідь: 54) - дана функція призначена для стирання кодів несправностей з пам'яті ЕБК, в випадку відсутності помилок в системі функція повинна повернути позитивну відповідь;
- `readDiagnosticTroubleCodesByStatus` (запит: 18, позитивна відповідь: 58) - дана функція призначена для зчитування кодів несправностей і їх статусу, збережених в пам'яті ЕБК;
- `readEcuIdentification` (запит: 1A, позитивна відповідь: 5A) - дана функція призначена для запиту інформаційних даних про автомобіль і ЕБК;
- `readDataByLocalIdentifier` (запит: 21, позитивна відповідь: 61) - дана функція призначена для запиту у ЕБК набору даних відповідно до конкретним параметром;
- `readMemoryByAddress` (запит: 23, позитивна відповідь: 63) - дана функція призначена для запиту набору даних певної довжини з пам'яті ЕБК, починаючи з конкретної адреси;
- `inputOutputControlByLocalIdentifier` (запит: 30, позитивна відповідь: 70) - дана функція призначена для керування деякими змінними внутрішніх параметрів ЕБК, наприклад, для установки значення обертів холостого ходу;
- `writeDataByLocalIdentifier` (запит: 3B, позитивна відповідь: 7B) - дана функція призначена для запису даних в пам'ять ЕБК;
- `testerPresent` (запит: 3E, позитивна відповідь: 7E) - дана функція призначена для повідомлення ЕБК про присутність діагностичного кабелю на лінії зв'язку і запобігання повернення ЕБК до нормального режиму роботи при відсутності запитів від пристрою не менше, ніж один раз в секунду.

В ході розробки цього додатка функції `startCommunication`, `stopCommunication`, `startDiagnosticSession`, `stopDiagnosticSession` були задіяні в процесі ініціалізації з'єднання з ЕБК і початку діагностичного сеансу. Для

отримання інформації про автомобілі і про його параметри, а також контролю над ними були використані функції `readEcuIdentification`, `readDataByLocalIdentifier` та `inputOutputControlByLocalIdentifier`.

1.5.4 Функція `readEcuIdentification`

Отримання загальної інформації про автомобіль і про його ЕБК відбувається за допомогою параметра `identificationOption` функції `readEcuIdentification`, яка визначає тип запитуваних даних. Можливі значення даного параметра описані нижче:

- `ECUIdentificationDataTable` (80) – дане значення параметра запитує інформацію про абсолютно всі ідентифікаційні дані;
- `VIN (Vehicle Identification Number)` (90) - дане значення параметра запитує інформацію, що містить назву моделі автомобіля;
- `vehicleManufacturerECUHardwareNumber` (91) - дане значення параметра запитує інформацію, що містить номер ЕБК згідно заводської документації;
- `systemSupplierECUHardwareNumber` (92) - дане значення параметра запитує інформацію, що містить код ЕБК по позначенню постачальника;
- `systemSupplierECUSoftwareNumber` (94) - дане значення параметра запитує інформацію, що містить код програмного забезпечення ЕБК по позначенню постачальника;
- `systemNameOrEngineType` (97) – дане значення параметра запитує інформацію, що містить код для запасних частин;
- `repairShopCode` (98) - дане значення параметра запитує інформацію, що містить код для запасних частин;
- `ProgrammingDate` (99) - дане значення параметра запитує інформацію, що містить дату підготовки прошивки ПЗУ;

- `vehicleManufacturerECUIdentifier` (9A) – дане значення параметра запитує інформацію, містить ідентифікаційні дані згідно позначенню виробника.

Уявімо також приклад запиту інформації про модель автомобіля і можливі відповіді на нього:

Запит: "1A 90", де 1A - ідентифікатор запиту `readECUIdentification`, 90 - значення параметра `readECUIdentification`.

Позитивна відповідь: "5A 90 56 41 5A 32 31 30 38 33 2D 30 30 30 30 30 31 30 2D 32 30 ", де 5A - позитивна відповідь `readECUIdentification`, 90 – значення параметра `readECUIdentification`, 56 41 5A 32 31 30 38 33 2D 30 30 30 30 31 30 2D 32 30 - запис VAZ21083-0000010-20 в шістнадцятковому форматі. Негативна відповідь: "7F 1A XX", де 7F - негативна відповідь, 1A - ідентифікатор запиту `readECUIdentification`, XX - код помилки.

1.5.5 Функція `readDataByLocalIdentifier`

Отримання інформації про значення параметрів з датчиків електронної системи керування двигуном відбувається за допомогою параметра `recordLocalIdentifier` функції `readDataByLocalIdentifier`, який визначає набір запитуваних даних. Можливі значення даного параметра описані нижче:

- `afterSalesServiceRecordLocalIdentifier` (01) - дане значення параметра запитує дані, необхідні для аналізу системи управління двигуном на СТО;
- `endOfLineRecordLocalIdentifier` (02) – дане значення параметра запитує дані, необхідні для аналізу системи управління двигуном в кінці складальної лінії автомобіля;
- `factoryTestRecordLocalIdentifier` (03) – дане значення параметра означає, що тестер запитує дані необхідні для аналізу блоку управління двигуном при вхідному контролі;
- `immobilizerRecord` (A0) - дане значення параметра означає, що тестер запитує статус алгоритму іммобілізації блоку управління;

#9	Слово прапорців даних несправностей 3	XX	Немає
#10	Слово прапорців даних несправностей 4	XX	Немає
#11	Температура охолоджуючої рідини	XX	$N=E-40$ [°C]
#12	Співвідношення повітря/паливо	XX	$N=14.7*(E+128)/256$
#13	Положення дросельної заслонки	XX	$N=E$ [%]
#14	Швидкість обертання двигуна	XX	$N=E*40$ [об/хв]
#15	Швидкість обертання двигуна на холостому ході	XX	$N=E*10$ [об/хв]
#16	Бажане положення регулятора холостого ходу	XX	$N=E$ [кроків]
#17	Дане положення регулятора холостого ходу	XX	$N=E$ [кроків]
#18	Коефіцієнт корекції часу впрыску	XX	$N=(E+128)/256$
#19	Кут випердження запалювання	XX	$N=E/2$ [гр.КВ]
#20	Швидкість автомобіля	XX	$N=E$ [км/г]
#21	Напруга бортової мережі	XX	$N=5.2 +E/20*0.05$ [В]
#22	Бажані оберти холостого ходу	XX	$N=E*10$ [об/хв]
#23	Напруга на датчику кисню	XX	$N=1.25*(E/256)$ [В]
#24	Прапорці стану датчика кисню	XX	Немає
#25	Тривалість імпульсу впрыску (мол. байт)	XX	Немає
#26	Тривалість імпульсу впрыску (ст. байт)	XX	$N=E/125$ [мсек]
#27	Масовий розхід повітря (мол. байт)	XX	Немає
#28	Масовий розхід повітря (ст. байт)	XX	$N=E/10$ [кг/год]
#29	Цикловий розхід повітря (мол. байт)	XX	Немає
#30	Цикловий розхід повітря (ст. байт)	XX	$N=E/6$ [мг/такт]
#31	Годинний розхід палива (мол. байт)	XX	Немає
#32	Годинний розхід палива (ст. байт)	XX	$N=E/50$ [л/год]
#33	Миттєвий розхід палива (мол. байт)	XX	Немає
#34	Миттєвий розхід палива (ст. байт)	XX	$N=E/128$ [л/100км]
#35	Контрольна сума ПЗУ (мол. байт)	XX	Немає
#36	Контрольна сума ПЗУ (ст. байт)	XX	Немає

1.5.6 Функція `inputOutputControlByLocalIdentifier`

Керування елементами ЕСКД здійснюється з допомогою параметрів функції `inputOutputControlByLocalIdentifier` `inputOutputLocalIdentifier`, яка однозначно визначає ідентифікаційний номер входу, виходу або внутрішньої змінної і `controlOption`, яка дозволяє керувати станами входів, виходів і внутрішніх змінних. Можливі значення даного параметра описані нижче:

Параметр `inputOutputLocalIdentifier`:

1. injector1OutputControl (01) - дане значення параметра запитує пряме управління форсункою 1.
2. injector2OutputControl (02) - дане значення параметра запитує пряме управління форсункою 2.
3. injector3OutputControl (03) - дане значення параметра запитує пряме управління форсункою 3.
4. injector4OutputControl (04) - дане значення параметра запитує пряме управління форсункою 4.
5. ignition1OutputControl (05) - дане значення параметра запитує пряме управління катушкою запалювання 1 і 4 циліндрів.
6. ignition2OutputControl (06) - дане значення параметра запитує пряме управління катушкою запалювання 2 і 3 циліндрів.
7. fuelPumpRelayOutputControl (09) – дане значення параметра запитує пряме управління реле бензонасоса.
8. coolingSystemFanRelayOutputControl (0A) - дане значення параметра запитує пряме управління реле вентилятора системи охолодження двигуна.
9. airConditionRelayOutputControl (0B) – дане значення параметра запитує пряме управління реле кондиціонера.
10. malfunctionIndicationLampOutputControl (0C) - дане значення параметра запитує пряме управління лампою індикації несправностей.
11. canisterPurgeValveOutputControl (0D) – дане значення параметра запитує пряме управління клапаном продувки адсорбера.
12. idleStepMotorPositionAdjustment (41) – дане значення параметра запитує пряме управління становищем регулятора холостого ходу.
13. idleEngineSpeedAdjustment (42) – дане значення параметра запитує пряме управління оборотами холостого ходу.

Параметр controlOption:

1. returnControlToECU (00) - дане значення параметра інформує ЕБК про те, що адаптер більше не управляє входом / виходом або внутрішньої змінної.

2. reportCurrentState (01) - дане значення параметра інформує ЕБК про те, що адаптер отримує контроль над входом / виходом або внутрішньої змінної.

3. reportIOConditions (02) - дане значення параметра інформує ЕБК про те, що адаптер запитує умови управління входом / виходом або внутрішньої змінної.

4. reportIOScaling (03) - дане значення параметра інформує ЕБК про те, що адаптер запрошувати спосіб масштабування входу / виходу або внутрішньої змінної.

5. resetToDefault (04) - дане значення параметра інформує ЕБК про те, що адаптер запрошувати скидання входу / виходу або внутрішньої змінної.

6. freezeCurrentState (05) - дане значення параметра інформує ЕБК про те, що адаптер запрошувати "заморозку" поточного стану входу / виходу або внутрішньої змінної.

7. executeControlOption (06) - дане значення параметра інформує ЕБК про те, що адаптер запрошувати виконати дію визначається послідовністю даних.

8. shortTermAdjustment (07) - дане значення параметра інформує ЕБК про те, що адаптер запрошувати підстроювання стану входу / виходу або внутрішньої змінної визначається в ОЗУ до значення переданого в послідовності.

9. longTermAdjustment (08) - дане значення параметра інформує ЕБК про те, що адаптер запрошувати підстроювання стану входу / виходу або внутрішньої змінної визначається в EEPROM до значення переданого в послідовності.

10. reportIOCalibrationParameters (09) – дане значення параметра інформує ЕБК про те, що адаптер калібрувальні дані для входу / виходу або внутрішньої змінної.

У даній реалізації протоколу параметр InputOutputLocalIdentifier підтримує не всі можливі значення controlOption, перераховані вище. Підтримувані значення параметра InputOutputControl представлені на Рисунок 1.19.

InputOutputLocalIdentifier	Значення	InputOutputControlParameter									
Найменування параметра	Hex	00	01	02	03	04	05	06	07	08	09
Injector1 Output Control	01	✓	✓					✓			
Injector2 Output Control	02	✓	✓					✓			
Injector3 Output Control	03	✓	✓					✓			
Injector4 Output Control	04	✓	✓					✓			
Ignition1 Output Control	05	✓	✓					✓			
Ignition2 Output Control	06	✓	✓					✓			
Fuel Pump Relay Output Control	09	✓	✓					✓			
Cooling System Fan Relay Output Control	0A	✓	✓					✓			
Air Condition Relay Output Control	0B	✓	✓					✓			
Malfunction Indication Lamp Output Control	0C	✓	✓					✓			
Canister Purge Valve Output Control	0D	✓	✓					✓			
IdleStepMotorPositionAdjustment	41	✓	✓							✓	
IdleEngineSpeedAdjustment	42	✓	✓							✓	

Рис. 1.19 - Значення параметра InputOutputControl

Приклад запиту прямого управління лампою індикації несправностей:

Запит: "30 0C 06 01", де 30 – ідентифікатор запиту inputOutputControlByLocalIdentifier, 0C – значення 50 параметра inputOutputLocalIdentifier, 06 і 01 – значення параметра controlOption. Позитивна відповідь: "70 0C 06 01", де 70 - позитивна відповідь inputOutputControlByLocalIdentifier, 0C - значення параметра inputOutputLocalIdentifier, 06 і 01 - значення параметра controlOption.

Негативна відповідь: "7F 30 XX", де 7F - негативна відповідь, 30 - ідентифікатор запиту `inputOutputControlByLocalIdentifier`, XX - якийсь код помилки.

Приклад запиту виставлення значення оборотів холостого ходу:

Запит: "30 42 07 C8", де 30 – ідентифікатор запиту `inputOutputControlByLocalIdentifier`, 42 – значення параметра `inputOutputLocalIdentifier`, 07 – значення параметра `controlOption` і C8 – бажане значення оборотів холостого ходу в шістнадцятковому вигляді (2500 обертів за хвилину). Позитивна відповідь: "70 42 07 32 1E 52 C8", де 70 - позитивна відповідь `inputOutputControlByLocalIdentifier`, 42 - значення параметра `inputOutputLocalIdentifier`, 07 - значення параметра `controlOption`, 32 - бажане положення регулятора холостого ходу, 1E - поточний стан 51 регулятора холостого ходу, 52 - поточне значення обертів холостого ходу, C8 – бажане значення оборотів холостого ходу.

Негативна відповідь: "7F 30 XX", де 7F - негативна відповідь, 30 - ідентифікатор запиту `inputOutputControlByLocalIdentifier`, XX - якийсь код помилки.

2 ПРОЦЕС ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ ПРОГРАМИ

2.1 Структура програми та її використання

Розроблена нами програма система складається з 4 форм (Face - рисунок 2.2, Info - рисунок 2.3, About - рисунок 2.4, Diag) і двох класів (Logic, Document). Структура програми в загальному вигляді представлена на блок-схемі (рис. 2.1).

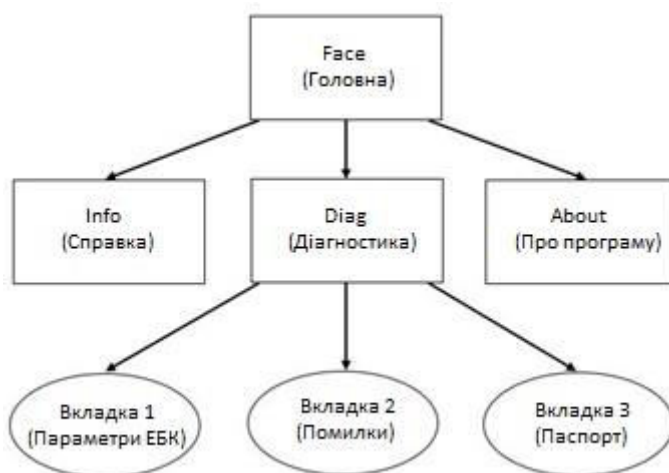


Рис. 2.1 - Блок-схема

Далі будуть розглянуті всі елементи даної структури окремо.

Рис. 2.2 - Головна форма

На даній формі присутні кнопки «Справка» і «Про програму», для переходу на форми Info і About, а також кнопка «Перейти до діагностики», яка відкриває форму, яка містить в собі основне поле діяльності користувача. При запуску програми після налаштування пристрою, яка супроводжується заповненням смуги прогресу, можна побачити назву моделі автомобіля, що діагностується, версію його ЕБК і інформацію про електронну системукерування двигуном, а також стан порту та пристрою.

На формі Info (рис. 2.3) користувач може отримати інформацію про всі параметри, які відображаються під час діагностики.

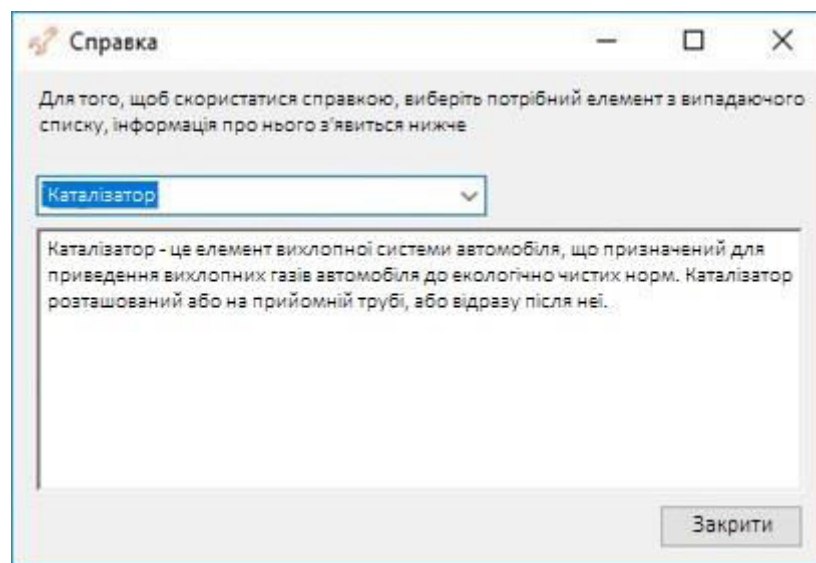


Рис. 2.3 – Info

Форма About (рис. 2.4) представляє короткі відомості про призначення програми і про розробника.

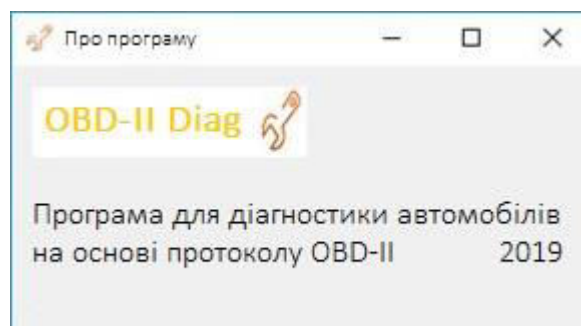


Рис. 2.4 - About

Форма Diag несе в собі основне смислове навантаження програмної системи і містить в собі елементи управління, з допомогою яких можна відстежувати значення параметрів роботи автомобіля (в тому числі у форматі графіків), а також змінювати деякі з них. На формі присутні 3 вкладки: «Параметри ЕБК», «Помилки» і «Паспорт». На вкладці «Параметри ЕБК» (рис. 2.5) можна побачити, як змінюються параметри роботи автомобіля в реальному часі як у вигляді списку, так і на графіку. Список параметрів що відображаються можна змінювати, для цього існують кілька наборів параметрів, в тому числі набір всіх параметрів. Також можна відобразити тільки ті параметри, які цікавлять користувача. Ще одна функція на цій вкладці - керування деякими складовими автомобіля, такими як лампа індикації несправностей, реле вентилятора та ін. Також передбачена можливість змінювати число обертів двигуна в хвилину.

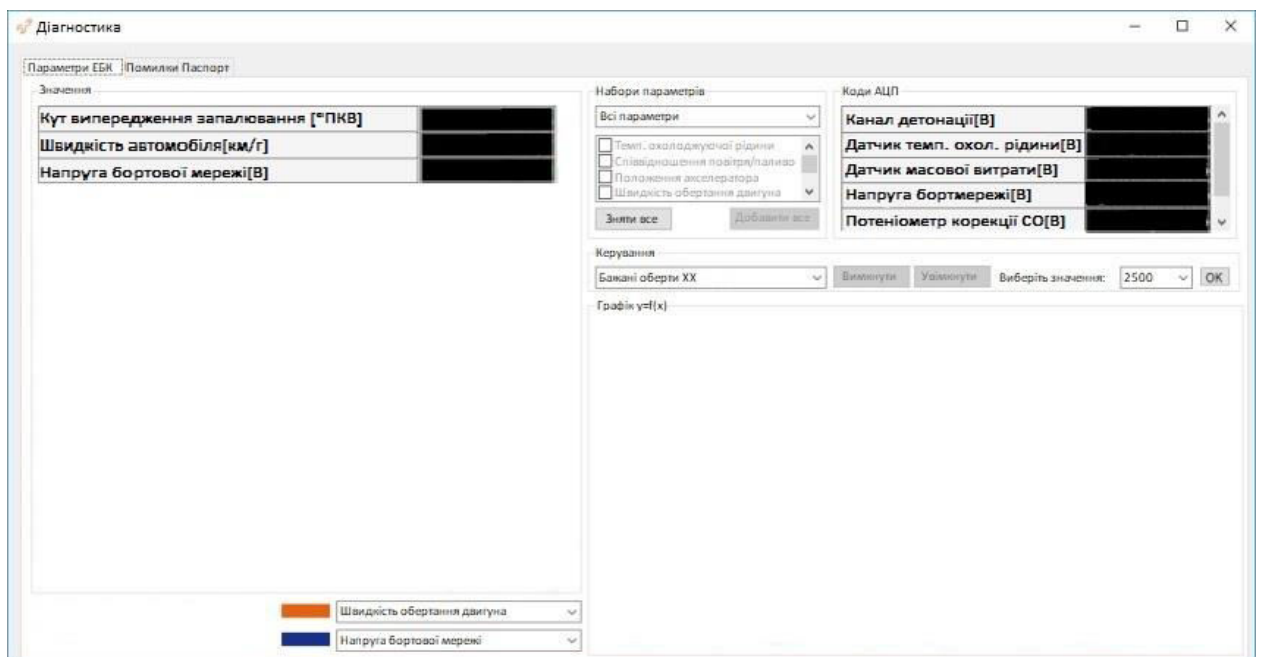


Рис. 2.5 - Вкладка «Параметри ЕБК»

Вкладка «Помилки» (рис. 2.6) призначена для пошуку несправностей в системі. Дані про помилки можна зберегти в текстовому документі. Також на цій формі присутній редактор допусків. На формі Параметри ЕБК зеленим

кольором виділені значення в допустимому діапазоні, а ті, які з нього виходять - червоним. Функція редактора полягає в тому, щоб змінювати цей діапазон для кожного з відображених параметрів.

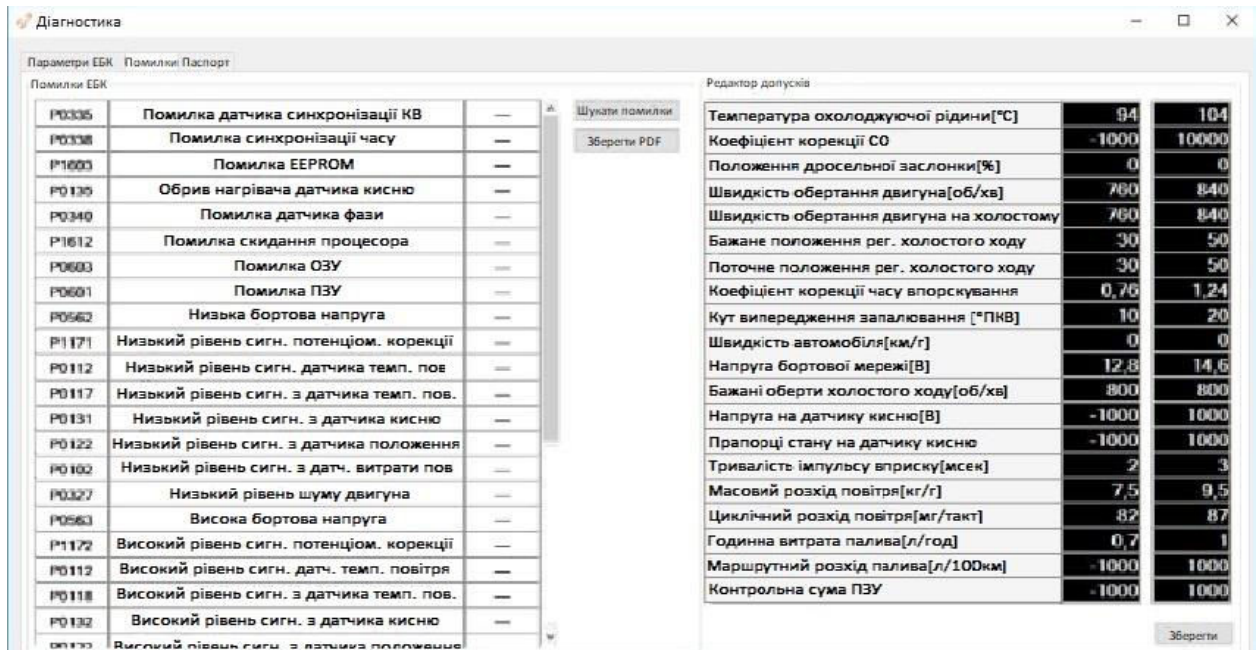


Рис. 2.6 - Вкладка «Помилки»

На вкладці Паспорт (рис. 2.7) зображені блоки більш конкретної інформації про автомобіль, його ЕБК і комплектації.

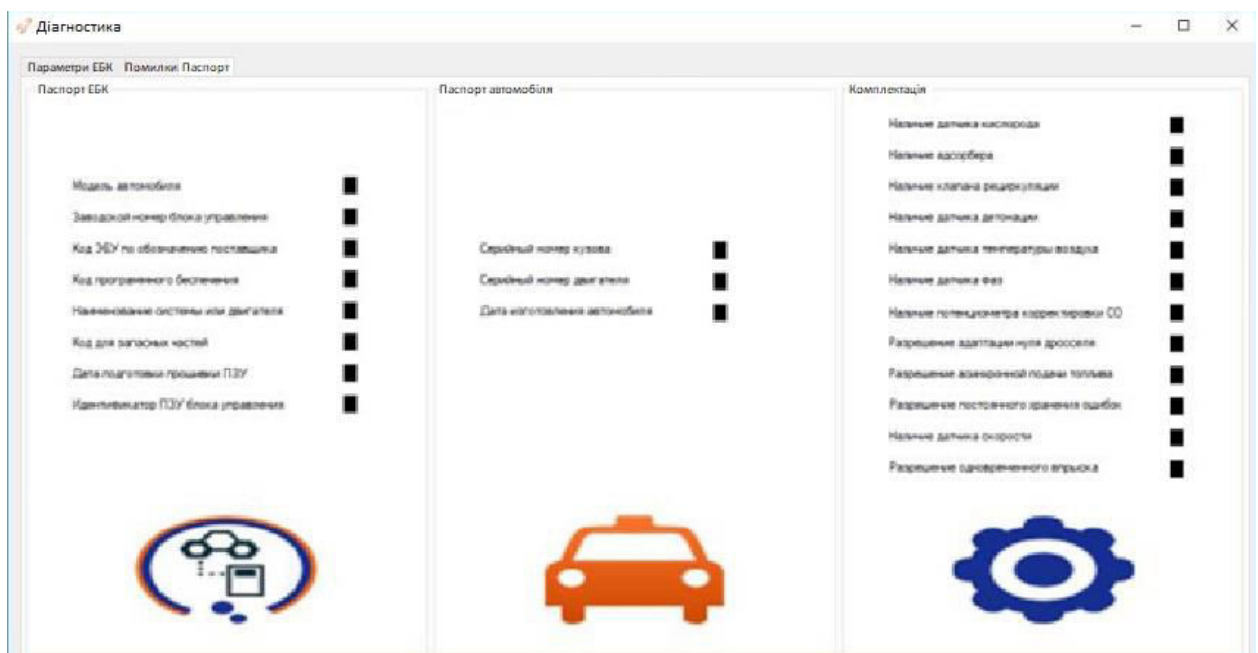


Рис. 2.7 - Вкладка «Паспорт»

Клас Logic містить в собі методи, визначають алгоритми, за якими програма взаємодіє з автомобілем через адаптер, а саме:

1) Init_Config - даний метод служить для ініціалізації пристрою, що супроводжується смугою прогресу.

2) Auto_Info - даний метод призначений для виведення різної інформації про автомобіль.

3) RLI_ASS - даний метод служить для обчислення і виведення значень параметрів afterSalesServiceRecordLocalIdentifier.

4) RLI_FT - даний метод служить для обчислення і виведення значень параметрів factoryTestRecordLocalIdentifier.

5) Parameters_manager_ON та Parameters_manager_OFF - ці методи призначені для включення / вимикання деяких функцій автомобіля.

6) Value_Color - даний метод призначений для виділення тексту кольором, який відображаються в списку параметрів автомобіля. Це залежить від діапазону допустимих значень.

7) Engine_Speed_Control - даний метод призначений для керування кількістю обертів двигуна за хвилину.

8) Errors_Search - даний метод призначений для пошуку і відображення поточних несправностей.

9) Define_Excludes - даний метод призначений для визначення комплектації автомобіля.

Клас Document містить в собі методи, які реалізують створення, збереження і друк текстових документів, що містять інформацію про автомобіль. Нижче наведені методи, які використовуються цим класом:

1) Save_Errors - даний метод призначений для збереження звіту про помилки в форматі PDF.

2) Save_Full - даний метод призначений для збереження повного звіту про автомобіль в форматі PDF.

2.2 Загальна логіка програмної системи

Для забезпечення безперебійної роботи пристрою необхідно звертатися до нього не рідше, ніж раз в 5 секунд. Тому при запуску програми, після пошуку і налаштування пристрою, починає працювати компонент `Backgroundworker` - потік, в якому виконуються які-небудь операції незалежно від основного потоку додатку. Таким чином, з перших секунд роботи програми на пристрій приходять запити з необхідною періодичністю. Користувач може залишатися на головній формі або переходити на форми `About` і `Info`, при цьому зв'язок з ЕБК буде стабільним і безперервним. Але при переході на форму діагностики даний компонент вимикається і починає працювати інший об'єкт класу `Backgroundworker`, в якому виконуються методи класу `Logic`. Але на відміну від першого, другий працює в іншому режимі, а саме сам себе зупиняє, а потім знову запускає. Це зумовлено тим, що звернення до елементів керування на формі, створеним не в поточному потоці неможливо, тому в процесі роботи програми передбачена проміжна стадія між подіями обміну даними з ЕБК, в якій обробляються і виводяться на екран значення параметрів[16]. Користувачеві доступні інтерактивні можливості вибору відображуваних значень параметрів, а також конкретних даних для відображення на графіках. Керування деякими параметрами автомобіля також відбувається в безпечному режимі, тому користувач може в будь-який момент скористатися цією функцією. На вкладці «Помилки» відображається список кодів несправностей і їх опис, а найголовніше - їх статус. Натиснувши на кнопку «Шукати помилки» функція пошуку помилок буде викликана 1 раз і оновить статус поточних несправностей. Також користувач може зберегти документ, що містить інформацію про поточні несправності. Зміна значень діапазонів в редакторі допусків зберігається протягом усього часу роботи програми. На вкладці «Паспорт» функція отримання інформації про автомобіль викликається один раз без участі користувача.

2.3 Тестування програмної системи

Об'єктом випробувань для даного проекту став автомобіль ВАЗ 2110 (седан) 2007 року випуску, двигун 1,6 л, 80 л.с. Моделі сімейства ЛАДА прийнято вважати першими автомобілями з інжекторними двигунами, коректна робота якого здійснюється за допомогою ЕБК[17]. Даний автомобіль має електронний блок керування “Январь 7.2”

У цьому розділі буде розглянуто тестування додатки на предмет роботи всіх заявлених функцій. В ході розробки було проведено 7 тестів.

2.3.1 Тест елементів керування

Дана програма складається з безлічі форм і елементів керування: кнопки, списки, що випадають, графіки і так далі. Мета даного тесту полягає в тому, щоб перевірити всі елементи керування на предмет коректної роботи. Всі елементи керування повинні виконувати свої функції миттєво і працювати незалежно один від одного.

По ходу тесту були задіяні всі елементи управління окремо і одночасно. Спостерігалася стабільна робота кожного елемента управління, які не впливає на інші.

2.3.2 Тест відображення параметрів автомобіля у реальному часі.

Даний тест перевіряє роботу програми на предмет відображення параметрів автомобіля на формі в реальному часі. Назва кожного параметра має гарно відображатися, текст назви повинен містити одиницю вимірювання певного параметра. Текст, який містить значення параметрів, повинен бути виділений зеленим, в разі якщо відповідає діапазону допустимих значень, або червоним, якщо виходить за його рамки. Отримані дані повинні оновлюватися в реальному часі з періодичністю меншою, ніж один раз в секунду (рис. 2.8).

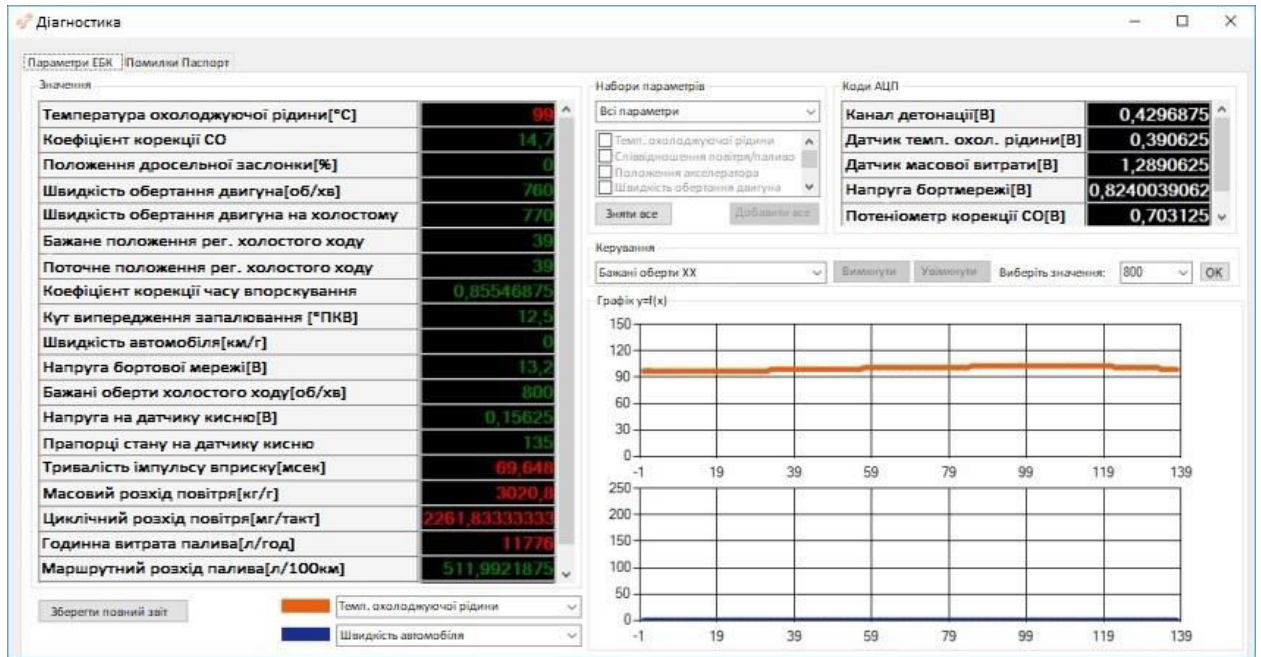


Рис. 2.8 - Параметри ЕБУ

2.3.3 Тест функцій керування двигуном

Даний тест перевіряє функцію керування деякими параметрами автомобіля, такими як увімкнення/вимкнення лампи індикації несправностей, реле вентилятора і т.д. Також цей тест перевіряє функції самостійного виставлення значень обертів холостого ходу (рис. 2.9).

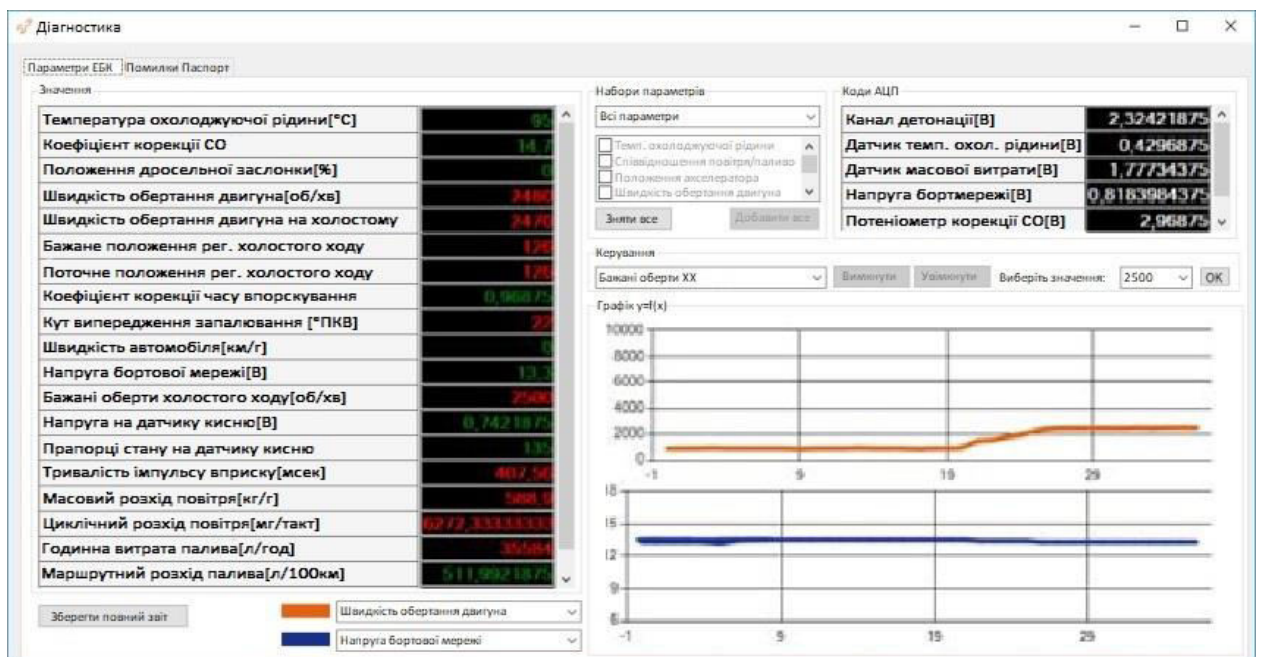


Рис. 2.9 - Виставлення значення оборотів холостого ходу (2500 об/хв)

На малюнку видно, що значення оборотів холостого ходу зростає, як і очікувалося, також це відображається на верхньому графіку.

2.3.4 Тест функції знаходження несправностей

Даний тест перевіряє функцію пошуку несправностей автомобіля. Функція викликається користувачем при натисканні на кнопку Шукати помилки, після чого в таблиці на формі повинен відобразитися статус поточних несправностей. Для перевірки даної функції були від'єднані датчик синхронізації колінчатого валу та датчик фази. В результаті це відобразилося на формі (рис. 2.10).

P0335	Помилка датчика синхронізації КВ	так
P0338	Помилка синхронізації часу	ні
P1603	Помилка EEPROM	ні
P0135	Обрив нагрівача датчика кисню	ні
P0340	Помилка датчика фази	так

Рис. 2.10 - Результат роботи функції пошуку помилок

2.3.5 Тест функції редагування допусків

Даний тест полягає в перевірці роботи функції редагування допусків. Поля введення мінімальних і максимальних значень повинні бути доступні для редагування. Після редагування одного або декількох значень результат роботи даної функції повинен відобразитися у вкладці «Параметри ЕБК ». В даному випадку діапазон допустимих значень температури охолоджуючої рідини був змінений з (94, 104) до (1500, 2000). При відсутності будь-яких непередбачуваних зовнішніх факторів значення температури охолоджуючої рідини ніколи не досягне цих величин. Такі значення діапазону були введені для демонстрації роботи функції редактора допусків (рис. 2.11).

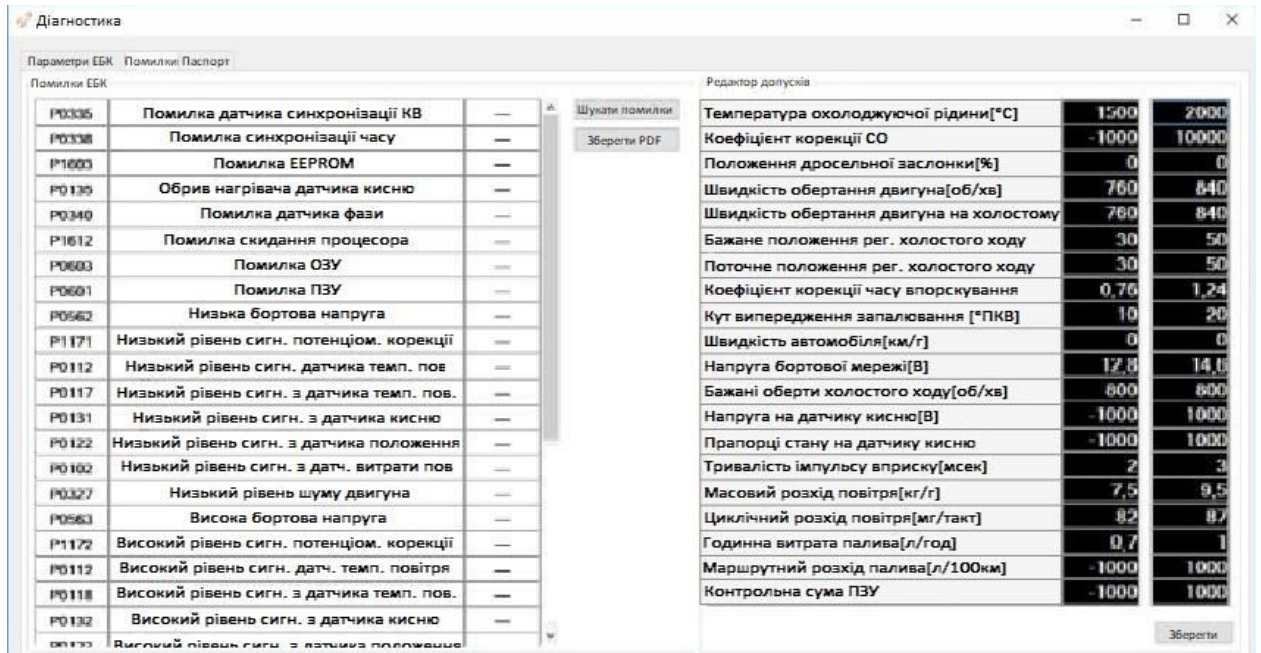


Рис. 2.11 - Зміна діапазону допустимих значень температури охолоджуючої рідини

Очікуваним результатом є зміна кольору, яким виділяється текст значення, що відображається на формі, після зміни діапазону (рис. 2.12).

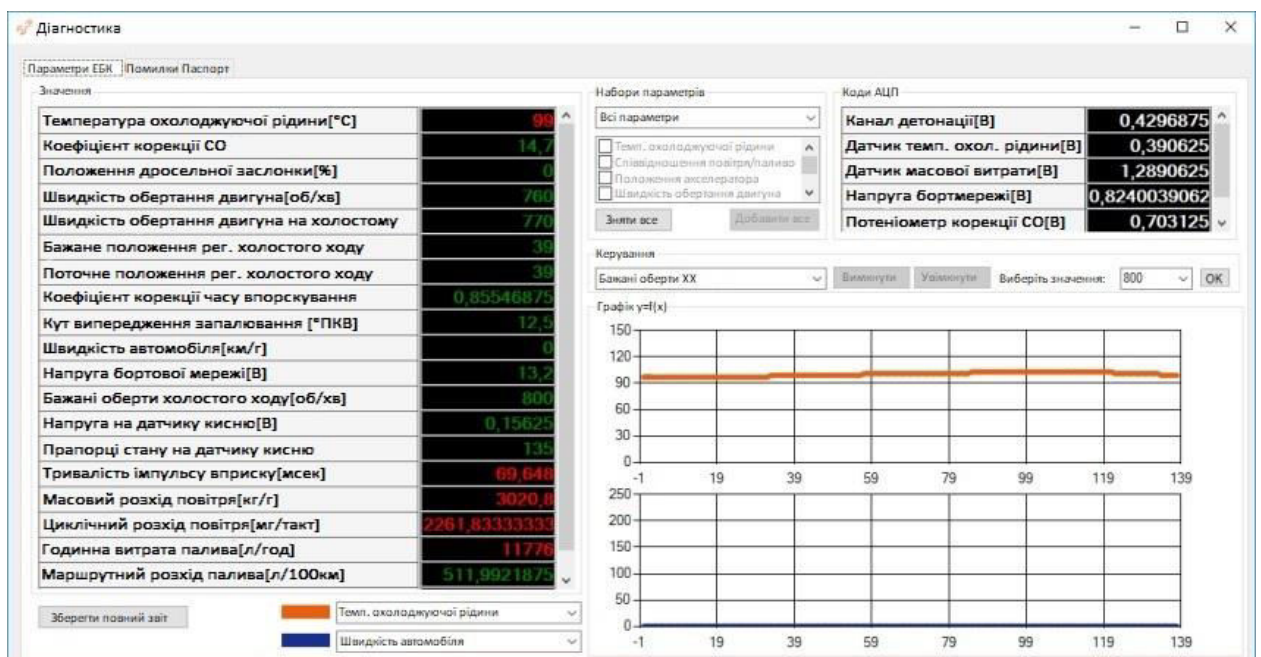


Рис. 2.12 - Результат тесту

2.3.6 Тест функції збереження документів

Даний тест перевіряє роботу функції збереження звітів в форматі PDF. Ця функція викликається користувачем при натисканні на кнопки «Зберегти в PDF», «Зберегти повний звіт», після чого відкривається діалог збереження Windows (рис. 2.13).

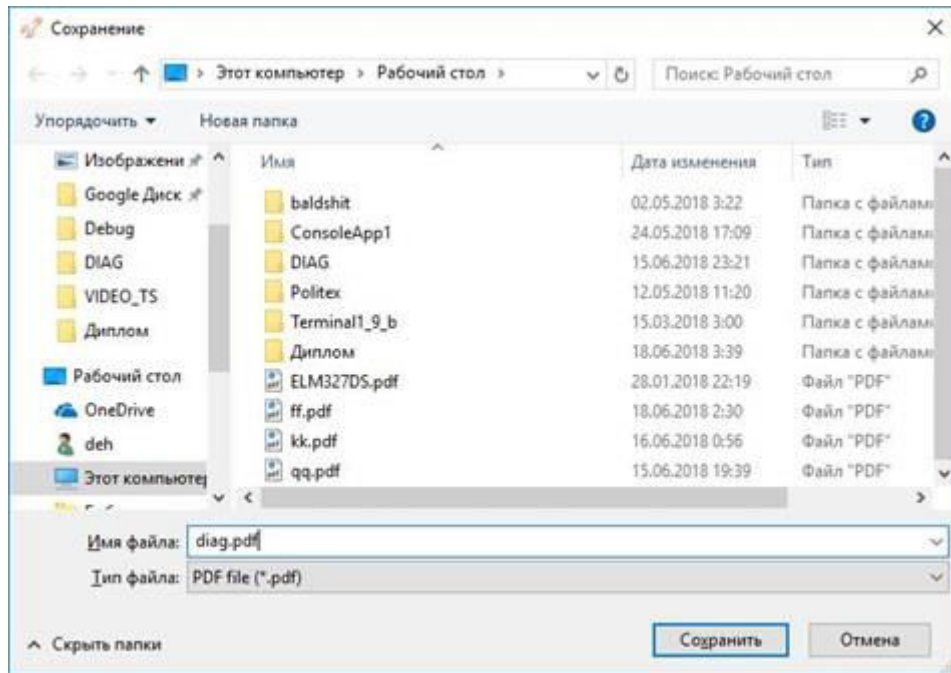


Рис. 2.13 - Діалогове вікно збереження документів

2.3.7 Тест часу нагріву двигуна

Даний тест визначає час, за яке значення температури охолоджуючої рідини доходить від 0 до 94 (оптимальна температура) градусів за цельсієм на холостому ході ти при середньостатистичній їзді (швидкість нерівномірна, подача палива нерівномірна, є зупинки по місту). За підсумком випробувань стало ясно, що на холостому ході двигун нагрівається до оптимальної температури за 9 хвилин і 25 секунд. При їзді по місту двигун нагрівся до оптимальної температури за 6 хвилин і 47 секунд. Варто

відзначити, що при русі автомобіля двигун отримує додаткове охолодження за рахунок зустрічного повітря, яке потрапляє під капот автомобіля.

3 ОРГАНІЗАЦІЙНО-ЕКОНОМІЧНА ЧАСТИНА

3.1 Розрахунок норм часу на виконання науково-дослідної роботи

Метою розділу є встановлення економічної доцільності розробки програмної системи для діагностики автомобілів з точки зору економічної ефективності.

На сучасному ринку інформаційних технологій існує багато програмних продуктів. Однак, вслід за швидкими змінами потреб ринку з'являється необхідність у нових функціях та методах. Ці потреби можна задовольнити двома шляхами методом вдосконалення існуючого програмного забезпечення та розробкою нового.

Від економічних показників залежить популярність програмної системи, кількість її продажів та поширеність серед різними сферами та користувачами.

Головною метою розділу є встановлення економічної доцільності проведення даного дослідження.

Ефективне використання часу має велике значення тому, що коефіцієнт корисної дії залежить від оптимального використання часу.

Розробку поділяють на декілька етапів, що дозволить полегшити і структурувати виконання розробки.

Основні етапи при виконанні розробки інформаційної системи наступні:

1. Підготовка опису задачі.
2. Збір необхідної інформації для аналізу програмної системи.
3. Вибір програмного забезпечення.
4. Розробка структури програми.
5. Розробка основних та додаткових можливостей програми.
6. Тестування програми.

Витрати часу по окремих операціях технологічного процесу відображені в таблиці 3.1.

Таблиця 3.1 – Операції технологічного процесу та час їх виконання

	Місячний оклад, грн	Денна зарплата, грн	Трудомісткість, людино-дні		Основна заробітня плата, грн	
			Процедурний підхід	Об'єктно- орієнтований підхід	Процедур- ний підхід	Об'єктно- орієнтований підхід
Керівник проекту	6000	272	3	3	817	817
Інженер- програміст	6800	308	10	15	3080	4620
Інженер- тестувальник	5400	245	2	2	490	490
Всього			15	20	4387 грн	5927 грн

Для оцінки тривалості виконання окремих робіт використовують нормативи часу або попередній досвід.

3.2 Визначення витрат на оплату праці та відрахувань

Відповідно до Закону України “Про оплату праці” заробітна плата – це “винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу”[19].

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

$$ЗП_{осн1} = 4387 \text{ грн}; ЗП_{осн2} = 5927 \text{ грн.}$$

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов’язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

$$ЗП_{дод} = 0,2 \cdot ЗП_{осн}; \quad (3.1)$$

$$ЗП_{дод1} = 0,2 \cdot ЗП_{осн1} = 877 \text{ грн};$$

$$ЗП_{дод2} = 0,2 \cdot ЗП_{осн2} = 1185 \text{ грн.}$$

Таким чином загальний фонд заробітної плати, що обчислюється за формулою:

$$\Phi ЗП = ЗП_{осн} + ЗП_{дод}; \quad (3.2)$$

$$\Phi ЗП_1 = 4387 + 877 = 5264 \text{ грн};$$

$$\Phi ЗП_2 = 5927 + 1195 = 7122 \text{ грн.}$$

Крім того, слід визначити відрахування на соціальні заходи:

- єдиний соціальний внесок – 3,6 %;
- військовий збір – 1,5 %;
- ПДФО (прибутковий податок) – 15 %.

Отже, сума відрахувань на соціальні заходи буде становити:

$$\text{Відр}_{\text{ЕСВ1}} = 0,036 \cdot \text{ФЗП} = 189,50 \text{ грн};$$

$$\text{Відр}_{\text{ЕСВ2}} = 0,036 \cdot \text{ФЗП} = 256,40 \text{ грн};$$

$$\text{Відр}_{\text{ВЗ1}} = 0,015 \cdot \text{ФЗП} = 78,96 \text{ грн};$$

$$\text{Відр}_{\text{ВЗ2}} = 0,015 \cdot \text{ФЗП} = 106,83 \text{ грн};$$

$$\text{Відр}_{\text{ПДФО1}} = 0,15 \cdot \text{ФЗП} = 789,6 \text{ грн};$$

$$\text{Відр}_{\text{ПДФО2}} = 0,15 \cdot \text{ФЗП} = 1068,3 \text{ грн}.$$

Нарахування на фонд оплати праці, які включають відрахування до Пенсійного фонду, фонду з тимчасової втрати працездатності, фонду з безробіття і фонду страхування від нещасних випадків на виробництві; для бюджетної організації тариф на фонд оплати праці встановлено на рівні 36,3%.

Зокрема, видання програмного забезпечення – 36,77%.

Нарахування на Фонд оплати праці (ФОП): $\text{ФОП}_{\text{ЕСВ}} = 0,3677 \cdot \text{ФЗП}$

$$\text{ФОП}_{\text{ЕСВ1}} = 0,3677 \cdot \text{ФЗП} = 1935,57 \text{ грн};$$

$$\text{ФОП}_{\text{ЕСВ2}} = 0,3677 \cdot \text{ФЗП} = 2618,75 \text{ грн}.$$

Всього витрат:

$$\text{В}_{\text{ЗП1}} = \text{ФЗП}_1 + \text{ФОП}_{\text{ЕСВ1}} = 5264 + 1935,57 = 7199,57 \text{ грн};$$

$$\text{В}_{\text{ЗП2}} = \text{ФЗП}_2 + \text{ФОП}_{\text{ЕСВ2}} = 7122 + 2618,75 = 9740,75 \text{ грн}.$$

3.3 Розрахунок матеріальних витрат

Матеріальні витрати визначаються як добуток кількості витрачених матеріалів та їх ціни:

$$M_{vi} = q_i \cdot p_i, \quad (3.3)$$

де: q_i – кількість витраченого матеріалу i -го виду;

p_i – ціна матеріалу i -го виду.

Звідси, загальні матеріальні витрати можна визначити:

$$Z_{m.v.} = \sum M_{vi}. \quad (3.4)$$

Таблиця 3.2 – Зведені розрахунки матеріальних витрат

Найменування матеріальних ресурсів	Один. Виміру	Фактично витрачено матеріалів	Ціна 1-ці., грн.	Загальна сума витрат, грн
Папір для друку	листів	100	0,24	25
Чорнила для принтера	шт	1	85	85
Флеш-накопичувач	шт	1	240	240
Всього				350

Отже, загальна сума матеріальних витрат становить 350 гривень.

В багатьох випадках існує ряд додаткових витрат які пов'язані із реалізацією проекту, але в більшості випадків їхня сума не перевищує десяти відсотків від загальної собівартості реалізації проекту.

3.4 Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_e = W \cdot T \cdot S, \quad (3.5)$$

де W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії.

Вартість кіловат-години електроенергії слід приймати згідно існуючих на даний час тарифів. Отже, 1 кВт з ПДВ коштує 2,5 грн.

Потужність комп'ютера для створення проекту – 400 Вт, кількість годин роботи необхідних для проекту – 120 години при процедурному підході та 160 годин при об'єктно орієнтованому підході.

$$Z_{e1} = 0,4 \cdot 120 \cdot 2,5 = 120$$

$$Z_{e2} = 0,4 \cdot 160 \cdot 2,5 = 160$$

3.5 Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення. Амортизаційні відрахування використовуються для повного відтворення зношених основних фондів (на реновацію), а також для їх часткового відшкодування (на капітальний ремонт і модернізацію).

Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{C_B \cdot N_A \cdot T_{\text{ФАК}}}{T_{\text{год}}} \quad (3.6)$$

де C_B – балансова вартість обладнання, грн;

N_A – норма амортизаційних відрахувань в рік, %;

$T_{\text{год}}$ – річний робочий фонд часу, год;

$T_{\text{ФАК}}$ – фактичний час роботи обладнання по написанню програми, год.

Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 % (квартальна – 15 %).

Отже, використовуючи в роботі 1 комп'ютер балансовою вартістю 12000 грн. Отже, амортизаційні відрахування будуть рівні:

$$A_1 = (12000 \cdot 0,6 \cdot 120) / 2080 = 415,40 \text{ грн};$$

$$A_2 = (12000 \cdot 0,6 \cdot 160) / 2080 = 553,84 \text{ грн}.$$

3.6 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління спілкою та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20-60 % від суми основної та додаткової заробітної плати працівників.

$$H_B = 0,5 \cdot 3\Pi_{\text{осн}} \quad (3.7)$$

де H_B – накладні витрати.

Отже, накладні витрати:

$$H_{e1} = 4387 \cdot 0,5 = 2193,5 \text{ грн}; H_{e2} = 5927 \cdot 0,5 = 2963 \text{ грн.}$$

3.7 Складання кошторису витрат та визначення собівартості

Проведемо розрахунок вартості створюваного програмного продукту. Вартість продукції включає у собі собівартість і планований прибуток.

Собівартість продукції – це сума грошових витрат підприємства (фірми) на виробництво і збут одиниці продукції, виконання робіт та надання послуг.

Повна собівартість програмного продукту дорівнює сумі усіх витрат на його виробництво:

Прийmemo прибуток на рівні 30%. Для нових інноваційних продуктів, що користуються високим попитом на ринку, ринкову вартість V_r можна встановити вищу.

Отже, вартість розробленого програмного забезпечення:

$$V_{r1,2} = C_{v1,2} + 0,3 \cdot C_{v1,2} = 8400,72 + 0,3 \cdot 8400,72 = 10920,9 \text{ грн.}$$

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

3.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Економічна ефективність (E_p) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_p = \frac{\Pi}{C_B} \quad (3.8)$$

де Π – прибуток;

C_B – собівартість.

Плановий прибуток ($\Pi_{пл}$) знаходимо за формулою:

$$\Pi_{пл} = V_p - C_B \quad (3.9)$$

Розраховуємо плановий прибуток:

$$\Pi_{пл} = 10920,9 - 8400,72 = 2520,21 \text{ грн.}$$

Отже, формула для визначення економічної ефективності набуде вигляду:

$$E_p = \frac{\Pi_{пл}}{C_B} \quad (3.10)$$

Тоді,

$$E_p = 2520,21 / 8400,72 = 0,3.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

$$T_{ок} = \frac{1}{E} \quad (3.11)$$

Термін окупності дорівнює:

$$T_{ок} = 1 / 0,3 = 3,3 \text{ роки}$$

У нашому випадку $\text{Ток1} = \text{Ток2} = 1/0,30 = 3,33$ років, що є нормальним, оскільки допустимим вважається термін окупності до 5 років.

Даний розрахунок виконаний у розрахунку на 1 екземпляр програмного продукту без врахування його тиражування.

Загальна вартість пропонованих робіт по розробці програмного продукту становить 10920,7 грн. Оскільки ефективність для обидвох проектів відповідно до встановленого рівня прибутку становить 0,3, що є високим показником, то проводити дані роботи варто і вкладені кошти окупляться за 3,33 року. Також слід врахувати можливість не одиничного замовлення програми, відповідно її ціна в такому випадку значно понизиться, а при продажі понад план прибуток зросте.

3.9 Визначення витрат на супровід і модернізацію програмного продукту, уточнений аналіз ефективності вкладених інвестицій

Ключові питання супроводу ПЗ – це управлінські, вимірювальні і вартісні. Відомий фахівець в області ПЗ Дж. Леман (1970 р.) запропонував розглядати супровід як еволюційну розробку програмних систем, оскільки здана в експлуатацію система не завжди цілком завершена, її треба змінювати протягом терміну експлуатації.

Внаслідок змін система стає більш складною і погано керованою. За різними оцінками фахівців витрати на підтримку і модернізацію програмного забезпечення, написаного процедурним методом, становлять більше 50% витрат на його створення, а інколи і перевищують доходи від його реалізації. Об'єктно-орієнтоване представлення програми дозволяє навіть середньому програмісту швидко і ефективно супроводжувати і модернізувати програми довжиною до декількох десятків тисяч рядків, що значно скорочує подальші витрати на супровід і модернізацію[20].

Виходячи із експертних оцінок і складності програми, приймемо величину витрат на супровід і модернізацію програмного забезпечення, створеного за процедурним методом 60% від початкових витрат, а за об'єктно-орієнтованим – 20%.

Собівартість модернізації:

$$C_{в}M_1 = 0,6 \cdot C_{в1} = 0,6 \cdot 8400,72 = 5040,43 \text{ грн};$$

$$C_{в}M_2 = 0,2 \cdot C_{в2} = 0,2 \cdot 8400,72 = 1680,14 \text{ грн.}$$

Для споживача вартість модернізації:

$$M_1 = 0,6 \cdot V_1 = 0,6 \cdot 10920,9 = 6552,54 \text{ грн};$$

$$M_2 = 0,2 \cdot V_1 = 0,2 \cdot 10920,9 = 2184,18 \text{ грн.}$$

Таким чином, уже після першої модернізації, загальні витрати на створення і супровід ПЗ для виробника за об'єктно-орієнтованим методом менші, ніж за процедурним, навіть якщо його собівартість є дещо дорожчою.

$$ЗВ_{1(вир)} = 8400,72 + 5040,43 = 13441,15 \text{ грн};$$

$$ЗВ_{2(вир)} = 8400,72 + 1680,14 = 10080,86 \text{ грн.}$$

Як і для споживача:

$$ЗВ_1 = 10920,9 + 6552,54 = 17473,44 \text{ грн};$$

$$ЗВ_2 = 10920,9 + 2184,18 = 13105,08 \text{ грн.}$$

Річна економія витрат за всіма можливими напрямками і додатковими витратами, пов'язаними з супроводом і тільки одноразовою модернізацією (у розрахунку на одиницю продукції) при об'єктно-орієнтованому методі порівняно із процедурним:

$$\Delta C_{(вир)} = ЗВ_{1(вир)} - ЗВ_{2(вир)} = 13441,15 - 10080,86 = 3360,29 \text{ грн};$$

$$\Delta C = ЗВ_1 - ЗВ_2 = 17473,44 - 13105,08 = 4368,36 \text{ грн.}$$

При постійному супроводі і модернізації програмного забезпечення різниця між загальними витратами за двома варіантами значно збільшується.

Чистий приведений дохід (ЧПД) визначається як різниця між сукупними доходами (сукупний грошовий потік) і сукупними витратами (сукупними інвестиціями) взятими за весь період життя інвестицій і дисконтована ними в кожному році на фактор часу. Дисконтування являє собою визначення вартості майбутніх грошових потоків у теперішній момент часу. Ефективним вважається той проект, який забезпечує максимум ЧПД, оскільки при цьому досягається найвища дохідність власників інвестицій.

Визначення ЧПД відбувається за формулою:

$$\text{ЧПД} = \sum_{i=1}^t \text{ГП}_i \alpha_{\text{ТВ}i} - \sum_{i=1}^t \text{ІК}_i \alpha_{\text{ТВ}i};$$

де ГП_i – грошовий потік i -го розрахункового року;

ІК_i – сума інвестицій i -го розрахункового року;

$\alpha_{\text{ТВ}i}$ – коефіцієнт дисконтування (коефіцієнт приведення інвестицій і грошового потоку до теперішньої вартості).

Грошовий потік – це фінансовий показник, що характеризує ефект інвестицій у вигляді грошових коштів, які повертаються інвестору.

Коефіцієнт дисконтування показує, яку величину грошових коштів ми отримаємо з урахуванням фактору часу та ризиків. Він дозволяє перетворити майбутню вартість у вартість на даний момент[21].

Для розрахунку коефіцієнта дисконтування (коефіцієнта приведення) грошових потоків за роками періоду економічного життя інвестицій використовується формула:

$$\alpha = \frac{1}{(1+i)^n};$$

де i – ставка дисконтування або норма дисконту, $i = 0,2$;

n – час або кількість періодів (років), протягом якого планується отримання доходу.

$$\alpha_0 = 1, \alpha_1 = \frac{1}{1+0,2} = 0,83.$$

Вважатимемо, що обидва програмних продукта однаково забезпечують потреби і вимоги споживача, і тому придбання першої чи другої програми однаково вплинуть на розмір його додаткових доходів на вкладений капітал. Тому приймемо цю величину за постійну, а порівняння дохідності двох проектів проведемо тільки за витратами.

$$\text{ЧПД}'_1 = \text{ГП} + 0,83 \cdot \text{ГП} = 10920,9 - 0,83 \cdot 6552,54 = 1,83\text{ГП} - 5482,30 \text{ грн};$$

$$\text{ЧПД}'_2 = \text{ГП} + 0,83 \cdot \text{ГП} = 10920,9 - 0,83 \cdot 2184,18 = 1,83\text{ГП} - 9108,03 \text{ грн}.$$

Чим менші витрати, тим більша дохідність проекту.

$$ЗВ_1 = 10920,9 + 6552,54 = 17473,44 \text{ грн};$$

$$ЗВ_2 = 10920,9 + 2184,18 = 13105,08 \text{ грн}.$$

Таблиця 3.3 – Техніко–економічні показники програмного продукту

Показник	Процедурний підхід	Об'єктно-орієнтований підхід
Зарплата основна, грн	4387	5927
Зарплата додаткова, грн	877	1185
Фонд заробітньої плати, грн	5264	4211
Відрахування на ФОП, грн	1935,57	2618,75
Разом на виплату плаці, грн	7199,57	9740,75
Матеріальні витрати, грн	350	350
Електроенергія, грн	139,2	185,6
Амортизація, грн	969,24	969,24
Накладні витрати, грн	2193	2963
Разом на ін.витрати, грн	3651,44	4467,84
Собівартість	8400,72	10920,9
Прибуток	2520,21	2811,79

Вартість розробленого ПЗ	8400,72	10920,9
Економічна ефективність	0,30	0,30
Термін окупності, років	3,33	3,33
Собівартість модернізації	5040,43	1680,14
Супровід і модернізація	6552,54	2184,18
Загальні витрати на розробку	13411,15	10080,86
Порівняльна економія витрат (для виробника)	-	3276,21
Загальні витрати (для споживача, на придбання програмного прод.)	17473,44	13105,08

Продовження Табл. 3.3

Порівняльна економія витрат для споживача)	-	3 652,2
Дохідність проекту для споживача за витратною частиною	-5482,30	-9108,03
Економія	-	3625,7

Економія витрат у випадку придбання, супроводу і одноразової модернізації програмного продукту, створеного за об'єктно-орієнтованим підходом, становить 3625,7 грн.

Отже, сучасну комп'ютеру програму доцільно виконувати по об'єктно орієнтованій парадигмі.

В даному розділі дипломної роботи магістра було розраховано основні техніко-економічні показники аналізу функціонування програмного продукту (таблиця 3.3).

Розраховане значення економічної ефективності, яке становить 0,82.

Загальна вартість пропонованих робіт по розробці програмного продукту становить 13411,15 грн. для першого варіанту та 10080,86 грн. для другого.

Оскільки ефективність для обидвох проектів відповідно до встановленого рівня прибутку становить 0,3 що є високим показником, то проводити дані роботи варто і вкладені кошти окупляться за 3,3 року, бо нормальним терміном окупності є термін, який коливається від 1 до 3 років, тоді розробка вважається доцільною і економічно вигідною[22].

При використанні об'єктно-орієнтовного підходу зменшується кількість працівників, які залучаються у проект, та зменшуються витрати на реалізацію проекту, але для підтримки проекту і його подальшої модернізації все ж в майбутньому потрібні набагато більші витрати.

Для функціонального підходу потрібна більша кількість працівників, часу і коштів, але на модернізацію і підтримку в загальному потрібно менше ресурсів, і у висновку програма виконана по функціональній парадигмі стає дешевшою для споживача, і приносить економію ресурсів для розробників.

Отже, програмний продукт може бути впроваджений та мати подальший розвиток, оскільки він є економічно вигідним за всіма основними техніко-економічними показниками.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Дослідження роботи з розробки програмної системи для діагностики автомобілів на основі протоколу OBD-II напряму пов'язані з роботою на комп'ютері.

Розробка та дослідження програмної системи проводилося в приміщенні, план якого приведено на рис. 4.1.

Приміщення має одностороннє природне освітлення (вікно: висота = 1,5 м, ширина = 2.5 м) і загальне штучне освітлення. Стіни і стеля обклеєні світлими шпалерами, підлога вкрита світлим ламінатом. У приміщенні відсутні сильні вібрації та шкідливі речовини. Склад повітря в відповідає вимогам.

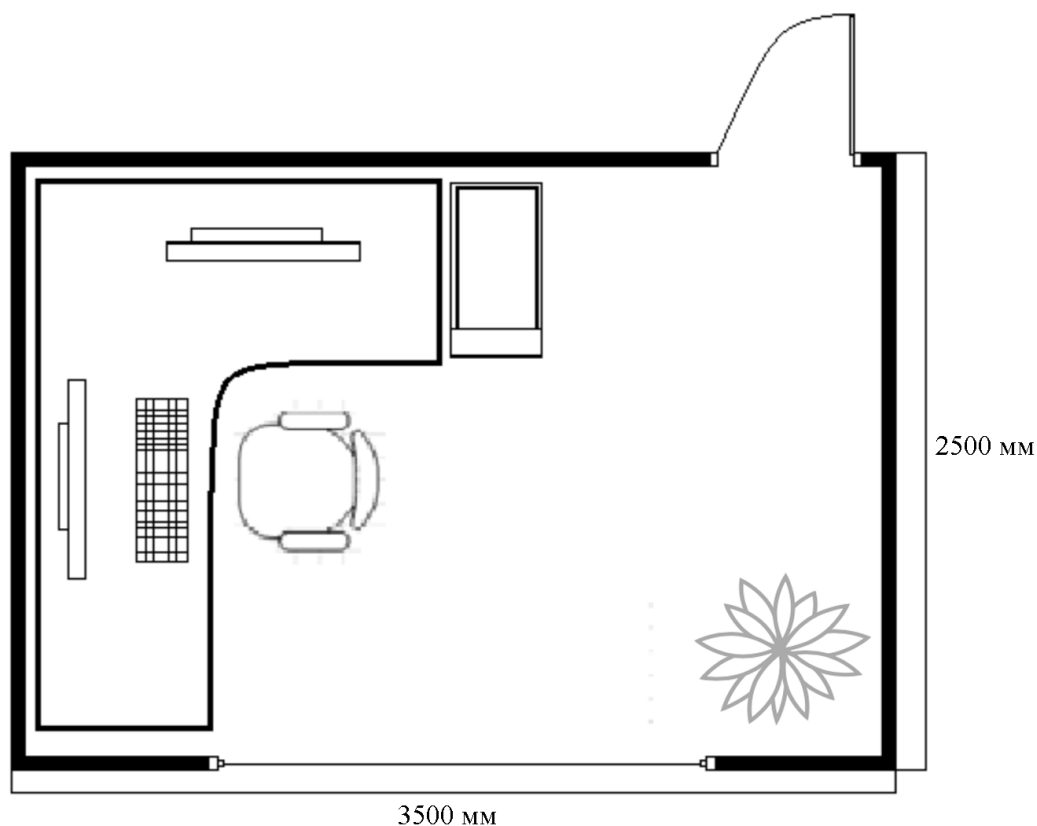


Рисунок 4.1 – План приміщення

Приміщення має довжину 3.5 м, ширину 2.5 м, висоту стелі 3.2 м. Кількість робочих місць - одне. Приміщення знаходиться на другому поверсі двоповерхової цегляної будівлі. Площа – 8.75 м², об'єм – 28 м³. Виходячи з цього, отримаємо дані, наведені в таблиці 4.1. Нормативні значення відповідають площі та об'єму для одного робочого місця оператора згідно з вимогами ДСанПіН 3.3.2-007-98.

Таблиця 4.1 – Фактичні та нормативні значення параметрів приміщення

Параметр	Норма	Реальні параметри
Площа, S	Не менше 6 м ²	8.75 м ²
Об'єм, V	Не менше 20 м ³	28 м ³

Можна зробити висновок, що отримані показники відповідають існуючим нормам та вимогам.

Характеристика робочого місця відповідає всім нормам та вимогам відповідно до НПАОП 0.00-7.15-18, та «Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» (ДСанПіН 3.3.2-007-98).

З обладнання в приміщенні знаходиться 1 комп'ютер та 2 монітори ACER VH27IPS 27". На все обладнання є паспорт та інструкція з експлуатації українською мовою. Згідно з супроводжувальної документації обладнання відповідає стандартам України і його можна використовувати без загрози здоров'ю та життю працюючого.

Згідно постанови про "Санітарні норми мікроклімату виробничих приміщень (ДСН 3.3.6.042-99)" цю роботу можна віднести до категорії легка 1а. Джерелами тепла в цьому приміщенні є люди, електроустаткування, освітлювальні прилади в темний час доби і система опалювання зимою. Оператором виділяється до 120 Ккал теплової енергії за годину.

Всі показники щодо мікроклімату приміщення відповідають вимогам зазначеним в санітарних нормах мікроклімату виробничих приміщень (ДСН 3.3.6.042-99) для робіт категорії легка 1а і є задовільними для здоров'я людини.

Природне освітлення здійснюється за допомогою вікна, площа якого складає $S' = 2,5 * 1,5 = 3,75 \text{ м}^2$ та являється боковим освітленням.

Штучне освітлення здійснюється за допомогою світлодіодних ламп білого світіння.

Робота за дисплеєм ПЕОМ за розрядом зорових робіт відноситься до III розряду. При загальному висвітленні освітленість робочого місця повинна становити від 200 до 400 лк.

У приміщенні застосовуються світильники, у яких встановлені 2 лампи. Висоту підвісу світильника визначимо з формули :

$$h = H - h_c - h_p - h_n \quad (4.1)$$

H - висота приміщення, м; h_c - висота світильника, м; h_n - відстань від стелі до підвісу, м; h_p - висота робочої поверхні, м.

Для розглянутого приміщення :

$$H = 3,2 \text{ м}; h_c = 0,2 \text{ м}; h_n = 0,18 \text{ м}; h_p = 0,7 \text{ м}. \quad (4.2)$$

звідси:

$$h = 3,2 - 0,2 - 0,18 - 0,7 = 2,12 \text{ м}. \quad (4.3)$$

Світильники розташовані в 1 ряд. Відстань між світильниками 2 метра, відстань від ряду до стіни 0,8 метра. Приміщення має наступні габарити:

довжина $A = 3,5$ метрів,

ширина $B = 2,5$ метрів.

Фактична освітленість на робочому місці становить 363.05 лк, що відповідає вимогам до III розряду зорової роботи. Можна зробити висновок, що для роботи з дисплеєм цілком достатньо існуючих джерел світла.

Джерелом шуму в приміщенні є комп'ютер. Кулери комп'ютера є сучасними і мають низький рівень шуму за рахунок нового підшипника із тиском рідини який самостабілізується (SSO2). Згідно технічній документації шум обумовлений кулером в блоці живлення складає 7-12 дБ, кулером процесора – 7-12 дБ, загальний, - 20 дБ. Враховуючи незначний рівень шуму від персонального комп'ютера та незначний рівень фонового шуму від іншого устаткування - сумарний рівень шумового забруднення приміщення не перевищує максимально допустимий рівень коригованої звукової потужності і складає не більше 50 дБ.

При роботі з персональним комп'ютером в робочому приміщенні значення характеристик вібрації на робочих місцях не перевищує допустимих значень відповідно до ДБН В.2.5-28-2018.

У приміщенні відсутні інфрачервоні, ультрафіолетові та електромагнітні випромінювання, бо усі монітори ПК вироблені на основі LED матриці, підсвітка якої здійснюється за рахунок органічних світлодіодів, що не має сильного електромагнітного випромінювання і є сертифіковані в Україні.

Технічні заходи із запобігання електротравм від контакту з нормально струмопровідними елементами електроустаткування:

- електромережа в приміщенні розведена в спеціальних каналах стін і підлоги;
- величина напруги мережі 220В;
- всі нормально струмопровідні елементи (в першу чергу електричні дроти) вкриті ізоляційними матеріалами;
- в джерелі безперебійного живлення персонального комп'ютера використовується механічне захисне блокування, що забезпечує вимикання напруги при його відкриванні.

Можна зробити висновок, що дане приміщення задовольняє нормам електробезпеки для встановлення ПК.

У приміщенні, що розглядається, можуть горіти: вироби з дерева, пластмас, тканини і паперу, ламінат. Горючі рідини, пил та волокна у приміщенні не використовуються і не виділяються, тому воно відноситься, відповідно до нормативної документації, до зони П-Па відповідно до державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98 і до категорії пожежної небезпеки В.

Аналіз умов праці для реалізації програмного забезпечення призначеного для діагностики автомобілів в розглянутому робочому приміщенні показав, що умови праці з ПЕОМ відповідають вимогам ДСН 3.3.6.037-99, оскільки площа та об'єм не менше нормативних значень, рівні шуму, вібрації і загазованості не перевищують нормативних обмежень.

Для підтримання параметрів мікроклімату в приміщенні встановлено радіатор центральної водяної системи опалення, що складається з 7 секцій.

Ергономіка робочого місця і режим зорової роботи задовольняють вимогам і сприяють зниженню втоми.

4.2 Безпека в надзвичайних ситуаціях

В даному розділі проаналізовані умови та можливості для підвищення стійкості роботи підприємства автомобільної галузі у воєнний час.

Підготовка транспортної системи до сталого функціонування у воєнний час проводиться з урахуванням особливостей всіх видів транспорту і включає аналіз стану і підвищення стійкості:

- об'єктів всіх видів транспорту;
- транспортних вузлів і пунктів взаємодії різних видів транспорту;
- основних транспортних напрямків кожного виду транспорту;
- транспортних систем регіонів країни в цілому.

Підготовка кожного виду транспорту до роботи в умовах війни включає розробку і здійснення комплексу завчасних заходів, що забезпечують швидке відновлення руху в заданих розмірах на кожному транспортному напрямку, намічених до першочергового відновлення.

Заходи охоплюють підготовку шляхів сполучення, об'єктів і транспортних вузлів, рухомого складу і систем управління.

На основі аналізу і узагальнення результатів "Вимог щодо підвищення стійкості" різних видів транспорту, наукових досліджень, практичної роботи з цієї проблеми можуть бути рекомендовані заходи, які ефективні для всіх видів автомобільних підприємств. Транспортні вузли є найбільш уразливими елементами транспорту країни у воєнний час. Транспортні комунікації і вузли можуть бути пошкоджені від впливу стихійних лих, блоковані при масових заворушеннях та заворушеннях. Для ефективної взаємодії всіх видів транспорту, комплексного використання збережених в надзвичайних ситуаціях ресурсів транспортних вузлів повинні готуватися пункти стику або перевантажувальні райони вантажів. Основними показниками, що характеризують підготовленість пунктів стику є: кількість взаємодіючих видів транспорту; обсяг вантажопереробки; підготовленість єдиного органу управління роботою всіх видів транспорту; комплексність розвитку технічних засобів; будівництво сполучних доріг, обходів, стиків міських і замських магістралей, розвиток дорожньої мережі в замській зоні; підготовка та створення дублюючих мостових переправ і переходів.

Надійне забезпечення експлуатаційними, технічними матеріалами та енергоресурсами: електроенергією, ПММ, водою, запчастинами. Важливе місце займає підготовка до використання місцевих ресурсів; підготовка постійних транспортних пристроїв, рухомого складу (судів) і виробничо-технічної бази транспорту до роботи в умовах війни, в т.ч. до виконання перевезень в умовах зараження, забезпечення захисту вантажів від зараження небезпечними речовинами, а також в умовах світломаскування; підготовка

транспортних засобів до перевезень військових сил, засобів і матеріально-технічних ресурсів, необхідних для проведення військових, а також до здійснення евакуаційних заходів; розвиток ремонтної бази, створення мобільного резерву ремонтних засобів; вдосконалення технології перевезень, методів вантажно-розвантажувальних робіт з використанням високопродуктивних засобів механізації, використання контейнерів; підготовка резерву кадрів; підготовка до розгортання тимчасових перевантажувальних пунктів поблизу найбільш ймовірних ділянок порушення комунікацій; розробка (уточнення) нормативних документів, спрямованих на забезпечення безаварійності транспортного процесу і досягнення мінімального ризику виникнення аварій і катастроф на транспорті; підвищення ефективності контролю за станом транспортних засобів і режимів процесу.

Розглянуті вимоги повинні виконуватися на всіх видах транспортних підприємств з метою підготовки його до сталого функціонування в умовах війни.

Автомобільний транспорт відіграє надзвичайно важливу роль в житті країни, на його частку припадає 39% вантажних перевезень і 39% пасажирообороту країни. Однією з найважливіших завдань держави є подальший розвиток автомобільного підприємства, перш за все транспорту загального користування. Це обумовлюється тим, що автомобільний транспорт володіє найбільшою мобільністю і живучістю, великим радіусом дії, здатністю діяти незалежно від пори року і доби, високою прохідністю, що забезпечує можливість пересування майже будь-якими дорогами і часто в умовах повного бездоріжжя, здатністю доставляти пасажирів і вантажі від місця посадки (навантаження) до місця призначення.

В умовах сучасної війни, а також в надзвичайних ситуаціях мирного часу автомобільне підприємство відіграє важливу роль в життєзабезпеченні населення і роботи економіки.

Автомобільний транспорт складається з наступних основних елементів: засобів сполучення, шляхів сполучення і підприємств, що забезпечують безперебійну роботу засобів і шляхів сполучення.

До засобів комунікації відноситься рухомий склад автомобільного транспорту: автомобілі, тягачі, причіпні системи і автопоїзда.

До шляхів сполучення автотранспорту належать автомобільні дороги з усіма спорудами на них: мостами, тунелями, шляхопроводами та т.д. Дуже часто вузли автомобільних доріг загальної мережі країни знаходяться в великих містах.

До підприємств, що забезпечує безперебійну роботу засобів і шляхів сполучення, відносяться:

- автоексплуатаційні вантажні, легкові, автобусні та таксомоторні парки, змішані автомобільні господарства, паливно-заправні станції і станції технічного обслуговування, гаражі, автовокзали, автостанції і т.д. ;

- авторемонтні сервіси і майстерні;

- дорожньо-будівельні, дорожньо-експлуатаційні та дорожньо-ремонтні підприємства.

Проведений аналіз стану і уразливості елементів автомобільного транспорту показує, що найбільш стійкими є: шосейні дороги, металеві та залізобетонні мости; найуразливішими: автозаправні станції, рухомий склад, а також виробничі будівлі автотранспортних підприємств.

Слід враховувати, що у воєнний час і в умовах надзвичайної ситуації мирного часу автомобільний транспорт буде виконувати різні види перевезень на короткі відстані, а при порушенні функціонування залізничного, водного транспорту він може виявитися в окремих регіонах основним видом транспорту.

Народногосподарські перевезення за надзвичайної ситуації будуть характеризуватися зміною клієнтури, обсягів і номенклатури вантажів, зміною маршрутів перевезень і строгим пропускним режимом на них.

Військові перевезення автомобільним транспортом будуть виконуватися в окремих випадках: за особливим розпорядженням, за планом і під керівництвом військового командування.

Підвищення стійкості і безпеки функціонування автомобільного підприємства є важливим завданням. При цьому слід враховувати, що автомобільний транспорт є основним джерелом забруднення навколишнього середовища в містах. В аваріях і катастрофах на автомобільному транспорті гине найбільша кількість людей.

З метою підвищення стійкості функціонування автомобільного підприємства у воєнний час необхідно дотримуватись таких рекомендацій:

1. Підготовка постійних транспортних засобів та виробничо-технічної бази транспорту до роботи в умовах надзвичайних ситуацій та у воєнний час.
2. Забезпечення експлуатаційними, технічними матеріалами та енергоресурсами для роботи в умовах надзвичайних ситуацій та у воєнний час.
3. Підготовка транспорту до перевезення сил і засобів, ресурсів, необхідних для ліквідації наслідків, а також для виконання евакуаційних дій.
4. Забезпечення захисту вантажів від зараження (забруднення).
5. Розробка (уточнення) нормативних документів, спрямованих на забезпечення безаварійності транспортного процесу.
6. Підвищення ефективності контролю за станом транспортних засобів і режимів перевізного процесу.
7. Підготовка об'їздів можливих зон бойових дій.
8. Підготовка до прокладання шляхів руху автотранспорту в осередках ураження.
9. Встановлення оптимальних маршрутів перевезення небезпечних речовин для забезпечення мінімального ризику аварій і зниження тяжкості їх наслідків для населення і території.

10. Підготовка автотранспорту і дорожньої мережі до перевезень евакуйованого населення.
11. Раціональне розміщення виробничої бази технічного обслуговування, ремонту і заправки автомобілів.
12. Розробка і підготовка до використання спрощених методів технічного обслуговування, ремонту та відновлення техніки.
13. Створення і зберігання в резерві мобільних засобів ТО і спец. обробки зараженої автотехніки.
14. Накопичення та зберігання запасних частин, обладнання, інструменту.
15. Удосконалення системи технічного прикриття автодоріг і об'єктів, що входять в основні транспортні напрямки.
16. Будівництво автодоріг поблизу залізниць, підготовка підходів до місць дублювання мостових переходів.
17. Підготовка виробничих потужностей підприємств (асфальтових заводів, кар'єрів і т. П.) для будівництва та ремонту доріг.
18. Удосконалення дорожньо-будівельної та містобудівельної техніки з урахуванням експлуатації її в умовах ведення бойових дій.
19. Розробка типових проектів рішень, варіантів і нормативів проведення робіт по відновленню постійних автотранспортних пристроїв, рухомого складу та виробничо-технічної бази.

ВИСНОВКИ

В процесі виконання магістерської роботи нами було проведено огляд та аналіз в області комп'ютерної діагностики для уточнення технічного завдання, виходячи з якого для розробки нашої програми була запропонована структура експрес-аналізу. Надалі продукт був реалізований на основі даної структури, після чого був протестований.

Система реалізована таким чином, що є досить зручною і надійною при використанні. Варто відзначити, що даний проект реалізований так, що його супровід, що включає в себе, в тому числі, розширення функціональності, як базової, так і проектної та не вимагає великих витрат. Таким чином, можна зробити висновок, що реалізована програма є працездатною, не має будь яких помилок, і придатна для впровадження в сервісних центрах або для особистого користування.

Насамперед нами був проведений огляд і аналіз в області комп'ютерної діагностики. Було дано визначення комп'ютерної діагностики і розглянута класифікація методів її проведення. Комп'ютерна діагностика транспортного засобу - це отримання відомостей про різні системи транспортного засобу, з використанням блоку управління автомобілем. Також стало відомо, що таке ЕБК і ЕСКД, і які вони бувають. Було проведене порівняння аналогів ПЗ і декількох популярних діагностичних кабелів. Таким чином, зроблений висновок про те, що тематика розроблюваного нами додатку відноситься до суб'єктивного автоматизованого типу комп'ютерної діагностики, що використовує ноутбук в командному режимі діагностування. Розробку даного додатку було прийнято вести з використанням адаптера ELM327, як найоптимальнішого варіанта.

В подробицях були розглянуті основні функції обміну даними з ЕБК по протоколу “Keyword Protocol 2000”. Були наведені приклади запитів, формат яких в подальшому був використаний при розробці алгоритмів, призначених для отримання, розшифровки і виведення на екран цих даних.

Ми описали середовище розробки, в якому розроблено програму, зокрема, одна була розглянута одна з її бібліотек, а також супутні програми, які допомогли визначити точний формат, в якому надсилаються команди на ЕБК. Була описана архітектура додатки, а саме призначення форм і класів, що визначають логіку роботи програми. Також було розказано про елементи управління даних форм і про методи вище згаданих класів.

В майбутньому розвитку запропонованої архітектури програмного забезпечення ми пропонуємо деякі центральні аспекти, які були б цінними ознаки або ідеями, які слід враховувати при подальшій роботі. Вони стосуються питань, стиснення даних, здатність виконувати оновлення програмного забезпечення віддалено і оцінки накопичених даних діагностики.

Через особливості вимог нашої архітектури програмного забезпечення, ми прийняли кілька методів, щоб звести до мінімуму переданих даних (наприклад, використання фільтрації). Для подальшого зменшення кількості даних, що передаються по бездротовій мережі, дані можуть бути стиснуті перед передачею. Реалізація цієї функції буде сприяти більш оптимального використання смуги пропускання, тим самим ще більше підвищити економічну ефективність архітектури програмного забезпечення.

Також, можна вважати, що багато можливостей забезпечуються необхідністю архітектури програмного забезпечення для зберігання великих обсягів даних постійної діагностики в базі даних системи. Дані діагностики можуть бути основою для майбутнього розвитку нових методів діагностики а, отже, могли б розширити ряд різних послуг.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Протоколи стандарту OBD2 [Електронний ресурс] – Режим доступу до ресурсу: <http://embeddedsystem.ru/index.php?page=protokoly-standarta-obd2>.
2. Інструкція ELM327 [Електронний ресурс] – Режим доступу до ресурсу: <https://елм327.рф/instrukciya-elm-327.html>.
3. Роз'єм діагностики OBD-II, як інтерфейс для IoT [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/company/unet/blog/408941/>.
4. Хендерсон Б. OBD-II и Электронные системы управления двигателем. Руководство по обслуживанию, диагностике и ремонту систем управления двигателем / Б. Хендерсон, Д. Х. Хейнес., 2011.
5. OBD-II standard car engine diagnostic software development [Електронний ресурс] – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/7062704/authors#authors>.
6. OBD-2 та електронні системи керування двигуном. Основні поняття, методи, процедури діагностики. Ілюстрації та фото. Керівництво для автолюбителів., 2019. – 248 с.
7. Розробка софту для ELM327 [Електронний ресурс] – Режим доступу до ресурсу: <https://openecu.net/blog/1708.html>.
8. Програма Scanmaster [Електронний ресурс] – Режим доступу до ресурсу: <https://vagcom.com.ua/programmy/programmy-dlya-elm327/scanmaster-elm-v21-ru/>.
9. Офіційний сайт "FORScan" [Електронний ресурс] – Режим доступу до ресурсу: <https://forscan.org/>.
10. Design Patterns - MVC pattern [Електронний ресурс] – Режим доступу до ресурсу: https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm.

11. Visual Studio .NET [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techopedia.com/definition/15740/visual-studio-net>.
12. Windows Forms [Електронний ресурс] – Режим доступу до ресурсу: <https://www.techopedia.com/definition/24300/windows-forms-net>.
13. Terminal 1.9b — працюємо з СОМ-портом [Електронний ресурс] – Режим доступу до ресурсу: <https://digitalchip.ru/terminal-1-9b-rabotaem-s-com-portom>.
14. Advanced Serial Port Monitor to read and record COM port data [Електронний ресурс] – Режим доступу до ресурсу: <https://www.eltima.com/advanced-serial-port-monitor/>.
15. Keyword Protocol 2000 [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Keyword_Protocol_2000.
16. Basics of OBD2 Scan Tools – Ultimate Guide for Getting and Using OBD2 Scanners [Електронний ресурс] – Режим доступу до ресурсу: <https://www.totalcardiagnostics.com/learn/basics-obd2-ultimate-guide-getting-using-obd-2-scan-tools/>.
17. Яковлев В. ДИАГНОСТИКА ЭЛЕКТРОННЫХ СИСТЕМ АВТОМОБИЛЯ / Володимир Яковлев., 2010.
18. Яловий Г.К. Економіка та організація виробництва / Яловий Г.К., Пашін В.П., Сичов В.С. – К.: "Політехніка", 2004. – 80 с.
19. Закон України "Про оплату праці" [Електронний ресурс]. – 1995. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/108/95-вр>.
20. Богданюк В.Є. Методичні вказівки до виконання організаційно-економічного розділу дипломних проектів / Богданюк В.Є., Березовський К.В., Пашін В.П. – К.: НТУУ "КПІ", 1999. – 66 с.
21. Методичні вказівки до виконання магістерської роботи освітнього рівня “магістр” студентами усіх форм навчання для напряму підготовки 121 – “Інженерія програмного забезпечення” / Укладачі : Петрик М.Р., Михалик

Д.М., Кінах Я.І., Гладь С.В., Цуприк Г.Б. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2016 – 26 с.

22. Методичні рекомендації по виконанню розділу техніко-економічного обґрунтування дипломних робіт студентами технічних спеціальностей напряму підготовки 8.05010302 «Інженерія програмного забезпечення» освітньо-кваліфікаційного рівня «Магістр» / Укладачі: Петрик М.Р., Михалик Д.М., Кінах Я.І., Гладь С.В., Цуприк Г.Б. – Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2016 – 28 с.

ДОДАТКИ

ДОДАТОК А -ТЕХНІЧНЕ ЗАВДАННЯ

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії
Кафедра програмної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедру
програмної інженерії
“ ___ “ _____ 2019 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської роботи
на тему: «Розробка програмної системи для діагностики автомобілів на
основі протоколу OBD-II»

Керівник роботи:
к.т.н., доцент Кінах Я. І.
“ ___ ” _____ 2019 р.

Виконавець:
студент групи СПм-61
Топоровський Віталій Васильович
“ ___ ” _____ 2019 р.

ЗМІСТ

Вступ

1. Підстави до розробки
2. Призначення до розробки
3. Вимоги до програмного продукту
 - 3.1 Функціональні характеристики
 - 3.2 Склад та параметри технічних засобів
 - 3.3 Інформаційна та програмна сполучність
4. Стадії розробки
5. Програмна документація
6. Порядок контролю та приймання

1 ПІДСТАВИ ДО РОЗРОБКИ

Розробка проводиться у відповідності до графіку навчального плану на 2019 рік, та згідно наказу на виконання магістерської роботи студента-магістра.

Тема проекту: «Розробка програмної системи для діагностики автомобілів на основі протоколу OBD-II».

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

Система для дистанційної діагностики автомобілів дозволить виконати пошук несправностей, спростити сприйняття даних, які транспортний засіб передає через спеціальний пристрій. Ця інформація буде доступною та зрозумілою навіть для недосвідченого користувача автомобіля.

Метою дипломної роботи є дослідження та розробка технології для системи дистанційної діагностики автомобілів. Використовуючи для з'єднання із автомобілем діагностичний протокол OBD-II та зовнішній пристрій-адаптер ELM327. В даній магістерській роботі розглянуто головні засоби для розробки діагностичного програмного забезпечення, проведено їх аналіз та визначено найкращий засіб для реалізації поставленої задачі.

Використовуючи подібне програмне забезпечення користувач може проаналізувати дані, що приходять з датчиків автомобіля або отримати конкретну інформацію про помилки в системі. Таким чином автолюбитель в короткі терміни та без зайвих витрат має уявлення про ту чи іншу проблему, що сталася з його транспортним засобом, і може передати ці дані представникам інстанцій, що займаються безпосередньо ремонтом, або ж усунути несправність самостійно. Результатом роботи є готовий програмний засіб, який можуть використовувати як звичайні автолюбители так і працівники автомобільних сервісних центрів.

3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

3.1 Функціональні характеристики

Програмне забезпечення має виконувати наступні дії:

- перегляд параметрів ЕБК в реальному часі;
- пошук помилок та можливість їх “скидання”;
- збереження звітів в форматі PDF;
- змінювати діапазон допустимих значень для деяких із параметрів;
- керування певними можливостями автомобіля;
- виводити графічну візуалізацію даних із необхідного датчика.

3.2 Склад та параметри технічних засобів

1) ПК або планшетний комп'ютер з 1024 Мб оперативної пам'яті, встановленою системою Windows 7, 8, 8.1, 10. Не менше 350 Мб вільного місця на жорсткому диску. Двоядерний процесор з тактовою частотою від 1 GHz і більше.

2) наявність встановленої .Net.

3.3 Інформаційна та програмна сполучність

Програмний продукт повинен коректно функціонувати в операційних системах Windows, різних поколінь, на яких доступне для встановлення бібліотека фреймворку .Net. Розроблювана бібліотека класів повинна бути пристосована до використання у інформаційних системах та програмних засобах. Розробку виконувати з використанням бібліотек та технологій мови C# в середовищі програмування Visual Studio 2017 з використанням .Net.

4. СТАДІЇ РОЗРОБКИ

В ходів реалізації роботи проект повинен пройти крізь наступні стадії розробки:

- аналіз предметної області;
- розробка макету - прототипу майбутньої програми;
- розробка архітектури додатку для діагностики згідно макету;
- реалізація графічного інтерфейсу програми;
- тестування результатів розробки на наявність помилок;
- оформлення супровідної документації;
- здача роботи.

5. ПРОГРАМНА ДОКУМЕНТАЦІЯ

Для програмного продукту повинні бути розроблені наступні документи:

- Пояснювальна записка;
- Технічне завдання;
- Презентаційний матеріал;
- Додатки.

6. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Розроблений програмний продукт має виконувати всі вимоги, що складаються з перерахованих у п. 3.1 характеристик.

Приймання проводиться спеціально створеною екзаменаційною комісією в термін до:

“__” грудень 2019р.

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Магістерська робота на тему:
«Розробка програмної системи для діагностики
автомоблів на основі протоколу OBD-II»

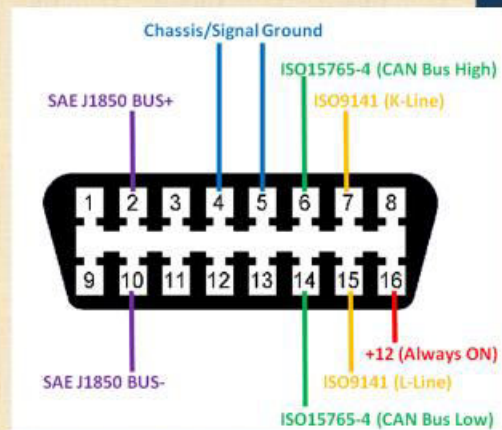


Виконав:
студент групи СПм-61,
Топоровський В.В.
Керівник: к.т.н., доцент
Кінах Я.І.

OBD-II

OBD-II (On-board diagnostics) - бортова діагностика, стандарт розроблений в середині 90-х, що надає повний контроль над двигуном.

Протокол OBD-II дозволяє здійснювати зчитування і стирання кодів помилок, перегляд поточних даних та параметрів роботи двигуна. Специфікація OBD-II являє собою стандартизований апаратний інтерфейс.



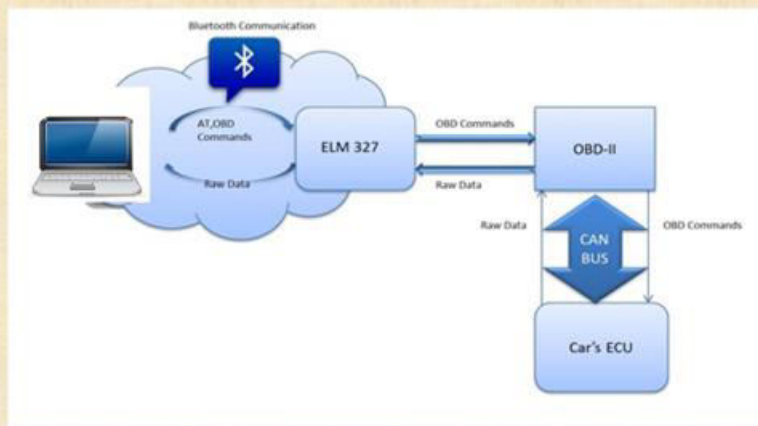
ELM327

ELM327 – універсальний автомобільний адаптер, який підключається до бортової мережі автомобіля через роз'єм OBD-II.

Bluetooth-версія даного адаптера дозволяє виконувати дистанційну діагностику.



Архітектура системи для передачі даних



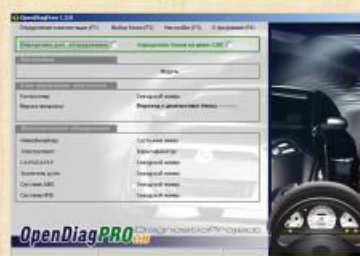
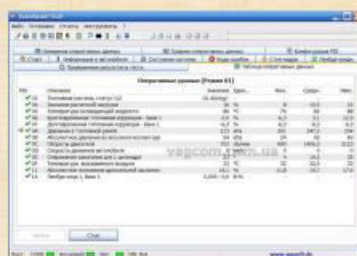
Завдання для досягнення мети:

- Проаналізувати область комп'ютерної діагностики автомобілів;
- Провести порівняльний аналіз актуального ПО і діагностичних адаптерів;
- На основі проведеного аналізу розробити макет - прототип майбутньої програми;
- Розробити додаток для діагностики транспортних засобів згідно макету;
- Протестувати додаток на працездатність і наявність помилок.



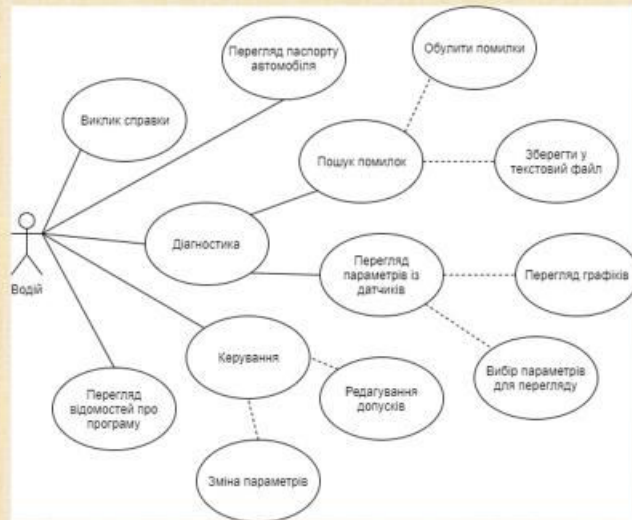
Аналоги програмної системи:

- Програма OpenDiag;
- Програма ScanMaster;
- Програма FORScan.



UML-діаграма варіантів використання

В ролі користувача програмної системи виступає водій автомобіля.



Технології та засоби реалізації ПЗ

Для написання проекту було обрано наступні технології:

- Мова програмування C# та технологія .NET ;
- Середовище розробки Visual Studio 2017;
- База даних MySQL та програма Workbench 8.0;



Головне вікно діагностики

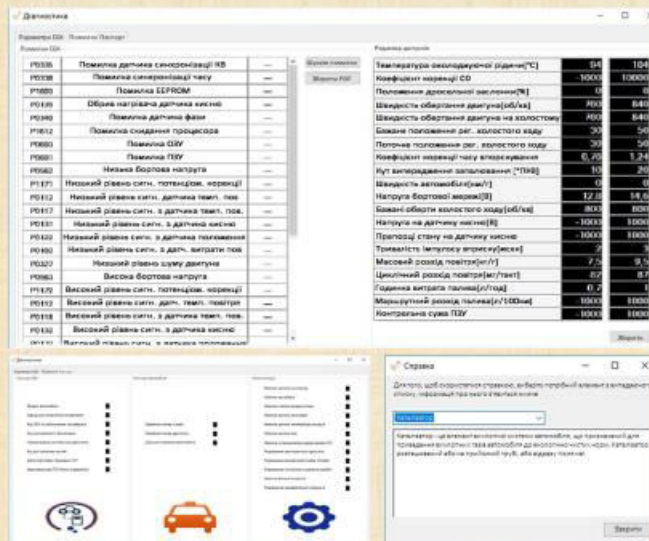


Вікно із помилками

R0335	Помилка датчика синхронізації KB	ТАК
R0338	Помилка синхронізації часу	НІ
R1603	Помилка EEPROM	НІ
R0135	Обрив нагрівача датчика кисню	НІ
R0340	Помилка датчика фази	ТАК

Для перевірки даної функції були від'єднані датчик синхронізації колінчатого валу та датчик фази. В результаті це відобразилося на формі

Додаткові вікна



Висновки

У рамках магістерської роботи виконано:

- Аналіз предметної області;
- Спроектовано та розроблено програмну систему для діагностики автомобілів;
- Розроблено широкий функціонал програми;
- Протестовано систему.



