

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“___” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи мережевих прикінцевих
пристроїв автомобіля”

Виконав здобувач вищої освіти
IV курсу, групи КІ-22-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Самарін С.О.
«___» _____ 2026 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Петренюк В.І.
«___» _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань_12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 27 » лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Самаріну Сергію Олексійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи мережесих
прикінцевих пристроїв автомобіля*

2. Керівник роботи *Петренюк Володимир Ілліч, канд. фіз.-мат. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 136-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту *23.05.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка
програмного забезпечення системи мережесих пристроїв автомобіля*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

Структурна схема системи *1 аркуш*

Функціональна схема системи *1 аркуш*

Діаграма процесів *1 аркуш*

Блок-схема алгоритму роботи додатку *2 аркуша*

7. Дата видачі завдання « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Петренюк В.І.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Самарін С.О.
(прізвище та ініціали)

АНОТАЦІЯ

Самарін С.О. Програмне забезпечення системи мережевих прикінцевих пристроїв автомобіля. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи мережевих прикінцевих пристроїв автомобіля.

Метою розробки є програмне забезпечення системи мережевих прикінцевих пристроїв автомобіля.

Результат роботи – програмна реалізація системи мережевих прикінцевих пристроїв автомобіля.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, мережеві прикінцеві пристрої автомобіля.

ABSTRACT

Samarin S.O. Software for the system of network terminal devices of a car. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for the system of network terminal devices of a car.

The purpose of the development is the software for the system of network terminal devices of a car.

The result of the work is the software implementation of the system of network terminal devices of a car.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on a PC with OS Windows 10/11.

The program was developed in the Python environment.

Keywords: computer engineering, vehicle network endpoints.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	16
2.3 Розгорнута постановка завдання	21
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	23
3.1 Опис функціонування системи	23
3.2 Розробка структурної схеми.....	30
3.3 Розробка функціональної схеми	43
3.4 Розробка діаграми процесів.....	67
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	69
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	69
4.2 Захист розробленого програмного забезпечення.....	78
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	81
6 ОСНОВНІ ВИСНОВКИ.....	83
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	85

					ВКРБ-123.26.0017.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Програмне забезпечення системи мережевих прикінцевих пристроїв автомобіля</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	Самарін С.О.					Б	1	92
<i>Перев.</i>	Петренко В.І.							
<i>Н.контр.</i>	Коваленко А.С.					ЦНТУ КІ-22-2		
<i>Затв.</i>	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ВМ	—	виконавчими механізмами двигуна
ЕБУ	—	електронний блок управління
OBDII	—	протокол читання даних з автомобіля
USB	—	універсальна серійна шина

К6ПЗ_2026

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Самостійна діагностика у 2026 році – це не просто очищення індикатора перевірки двигуна. Сучасні автомобілі мають більше модулів, чіткішу логіку контролю викидів та більше робочих процесів, які вимагають скидання налаштувань або активного тестування після заміни деталі.

Якщо вам потрібен зручний для самостійного використання сканер та діагностичний інструмент для транспортних засобів, спочатку зосередьтеся на трьох речах: надійний доступ до всієї системи, достатнє покриття для скидання налаштувань, які ви фактично виконуєте, та чи потрібен вам двонаправлений сканер для активних тестів.

Імпульс ринку також пояснює швидкий розвиток діагностичних інструментів. У звіті Research and Markets за 2025 рік зазначається, що світовий ринок автомобільних діагностичних інструментів у 2025 році складе 36,1 млрд доларів США, причому зростання буде зумовлене штучним інтелектом, бездротовим підключенням та потребами в діагностиці електромобілів/гібридів. GlobeNewswire

Зрештою, пам'ятайте, що фізичний інтерфейс все ще стандартизований для більшості легкових автомобілів. SAE описує J1962 як такий, що призначений для задоволення вимог США до роз'ємів OBD (знайомий 16-контактний порт). SAE International

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи мережевих прикінцевих пристроїв автомобіля.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем мережевих прикінцевих пристроїв автомобіля.
- Дослідження системи мережевих прикінцевих пристроїв автомобіля.
- Програмна реалізація системи мережевих прикінцевих пристроїв автомобіля.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі мережесистемних пристроїв автомобіля.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи мережесистемних пристроїв автомобіля, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

К6ПЗ_2026

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Технологія автоматичного сканування значно розвинулася у 2026 році. Вона використовує передові алгоритми та датчики для аналізу транспортних засобів у режимі реального часу. Ця інновація підвищує безпеку та ефективність на дорозі. Автоматичне сканування може виявляти такі проблеми, як тиск у шинах та робота двигуна. Воно також завчасно виявляє потенційні загрози безпеці. Технологія пропонує водіям душевний спокій.

Використання цієї технології має помітні переваги. Водії отримують миттєвий зворотний зв'язок про стан свого автомобіля. Швидка діагностика може заощадити час і гроші на ремонт. Крім того, автоматичне сканування допомагає запобігти аваріям, попереджаючи водія про потенційні проблеми. Регулярне сканування сприяє проактивному підходу до технічного обслуговування автомобіля.

Поради щодо максимізації переваг автоматичного сканування включають планування регулярних сканувань. Це гарантує, що проблеми будуть виявлені до того, як вони загостряться. Також подумайте про всебічний перегляд результатів сканування. Поміркуйте над своїми звичками водіння та тим, як вони впливають на стан вашого автомобіля. Використовуйте висновки кожного сканування. Коригування ваших звичок може продовжити термін служби вашого автомобіля та підвищити безпеку.

1.2 Область застосування

Технологія автоматичного сканування використовує передові датчики та алгоритми для швидкого аналізу транспортних засобів. Під час процесу вона сканує різні системи в режимі реального часу. Технологія може виявляти такі проблеми, як несправності двигуна або нерівномірності тиску в шинах. Цей швидкий аналіз допомагає уникнути дорогих поломок.

Процес включає кілька компонентів, таких як камери та діагностичні інструменти. Ці компоненти працюють разом, щоб забезпечити точні показники. Однак технологія не є безпомилковою. Іноді вона може неправильно інтерпретувати дані, що призводить до непотрібного ремонту. Це підкреслює важливість людського нагляду в автомобільній діагностиці.

Крім того, інтерфейс відображає інформацію у зручній для користувача формі. Користувачі можуть легко зрозуміти технічні деталі без попередніх знань. Однак деякі користувачі можуть спочатку вважати інтерфейс заплутаним. Це підкреслює необхідність належного навчання та ознайомлення з системою. Загалом, хоча технологія автоматичного сканування забезпечує ефективність, вона вимагає ретельного впровадження та розуміння.

Передова технологія автоматичного сканування у 2026 році принесе численні переваги як споживачам, так і промисловості. Ця технологія підвищує безпеку транспортних засобів, дозволяючи сканувати навколишнє середовище в режимі реального часу. Вона виявляє перешкоди та допомагає в навігації, зменшуючи ризики зіткнень. Це має вирішальне значення в жвавих міських умовах, де пішоходи та велосипедисти є звичайним явищем.

Ще однією значною перевагою є ефективність. Завдяки покращеним можливостям сканування транспортні засоби можуть аналізувати схеми руху та оптимізувати маршрути. Це призводить до скорочення часу подорожі та зниження витрати палива. Однак це також може призвести до надмірної

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

залежності від технологій. Водії можуть стати менш уважними, занадто довіряючи системі. Це критичний момент для майбутнього розвитку.

Крім того, передова технологія автоматичного сканування сприяє обізнаності щодо технічного обслуговування. Вона може виявляти механічні проблеми та попереджати водіїв, перш ніж вони погіршаться. Такий проактивний підхід допомагає запобігти поломкам та дорогому ремонту. Проте існує потреба в навчанні щодо інтерпретації цих сповіщень. Користувачам може бути важко зрозуміти складність технології. Таким чином, хоча переваги очевидні, проблеми залишаються.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи мережевих прикінцевих пристроїв автомобіля, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Для написання програмного забезпечення поставленого в завданні на бакалаврське проектування необхідно провести огляд існуючих програм діагностики автомобіля.

Carman Scan Wi

Carman Scan Wi – програмно-апаратний мультимарковий сканер на базі ПК для діагностики азіатських (включаючи праворульні японські автомобілі) і європейських (німецькі й французькі моделі) до поточного року випуску включно. Ця модель є програмною версією сканера Carman Scan Lite.

Carman Scan Wi є професійним діагностичним апаратом, виконаним у вигляді незалежного модуля з USB – портом і картою LAN для бездротового з'єднання.

Основні мови програмного забезпечення: Український, Англійський, Китайський, Німецький, Французький, Іспанський, Італійський і ін. Відновлення виходять періодично й накачуються через інтернет у прилад через комп'ютерну програму.

Програмне забезпечення для ПК:

- Аналіз записаних даних за результатами вимірів.
- Завантаження відновлень через спеціальну програму.

Характеристики основного блоку сканера:

- Пам'ять: 32Мб Внутрішня Флеш Карта.
- Робоча напруга: 7-36 Вольт.
- Споживання електрики: 4,5 Ват.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

- Зв'язок із PC: USB кабель або бездротова LAN.
- Робоча температура: 0-60°C.
- Колір корпусу: чорний.
- Колір захисного гумового чохла: чорний.
- Either NET: IEEE 802.11b 11M bps.
- Ширина*Довжина*Товщина: 100*180*25 мм.
- Вага основного блоку: 300 р.

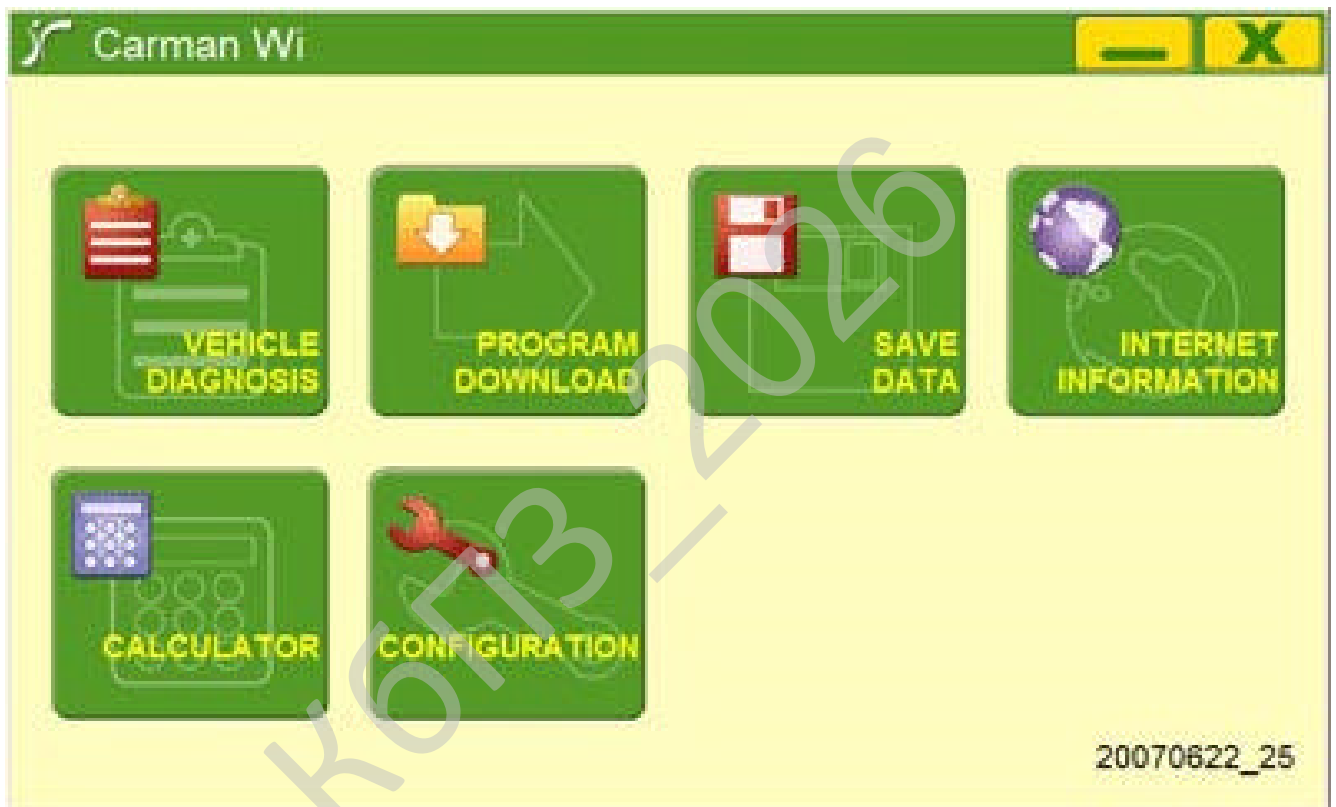


Рисунок 2.1 – Інтерфейс користувача Carman Scan Wi

opelscan

Комплекс призначений для діагностики електронних блоків управління автомобілів Opel/Holden/Vauxhall, зроблених з 1987 по 2005 рік. Комплекс має можливість діагностувати наступні модулі:

- Engine – Двигун.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

- ABS – Антиблокувальна система гальм.
- Airbag – Подушки безпеки.
- CDLS – Центральний замок.
- Automatic Transmission – автоматична трансмісія.
- Immobilizer – Імобілізатор.
- Instrument – Комбінація приладів.
- MID – Мультиінформаційний дисплей.
- TID – Трьох-інформаційний дисплей.
- CID – Кольоровий інформаційний дисплей.
- Radio – Радіо.
- Climat Control – Управління клімат-контролем.
- Traction Control – Протибуксовочна система.
- ESP – Протизаносна система.
- EPS – Електричне рульове управління.
- EHPS – Електрогідравлічне рульове управління.
- BC – бортовий комп'ютер.
- Add-On-Heater – Додатковий підігрівник.
- Cooling System – Система охолодження.

При запуску програми відкривається меню вибору модуля для діагностики. Спочатку вибирається Рік Випуску (Select Year) автомобіля, потім модель (Select Model), далі система (Select System) і конкретний ECU (Select Engine/Module). У програму закладена інформація блоків у які роки вони встановлювалися на автомобіль.

Кожний діагностуємий блок управління має свій паспорт із ідентифікаційними даними (ECU Identification), що містить наприклад таку інформацію, як номер блоку по маркуванню Опеля, номер блоку по маркуванню виробника блоку, дата програмування, дата випуску, альфа-код, і ін. Дане вікно показує паспорт блоку.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

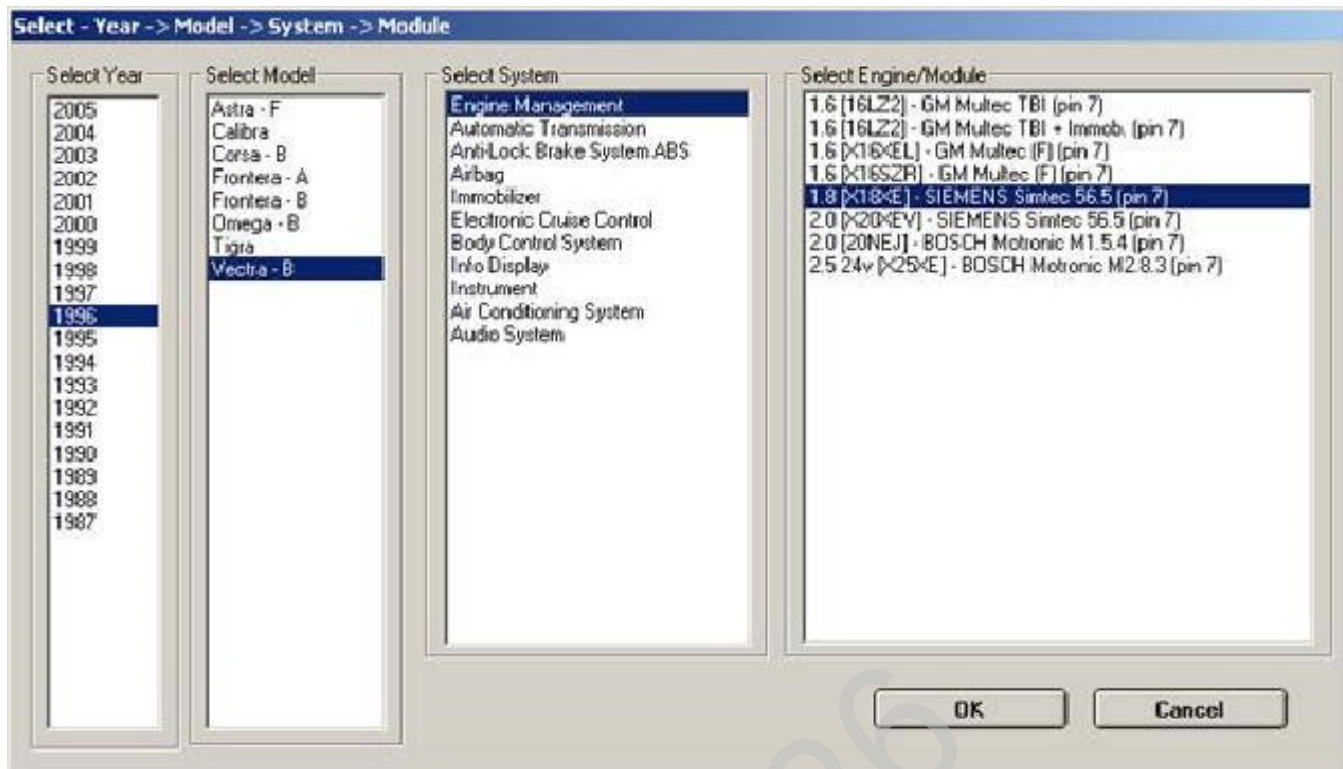


Рисунок 2.2 – Інтерфейс користувача opelscan

Кожний ECU має можливість видачі своїх параметрів у реальному часі. Дане вікно призначене для показу таких даних. У лівій частині вікна розташована таблиця з усіма доступними для даного блоку параметрами. Включити відображення й відновлення всіх параметрів можна кнопкою "Check/Uncheck All". Також є можливість включити/виключити кожної з параметрів, поставивши (або відповідно забравши) напроти назви параметра галочку. Необхідно також урахувувати, що для деяких протоколів включення великої кількості параметрів може привести до істотної з відновлення. Крім того, поточне значення будь-якого одного параметра можливо подивитися великим шрифтом (може бути зручно, якщо комп'ютер перебуває вдалечині), а також у вигляді графіка. Відображення у вигляді графіка може бути зручним при вивченні сигналів, що змінюються в часі, наприклад сигналу датчика кисню й ін. Також для багатьох блоків для більшості параметрів у спеціальному вікні виводиться інформація про номінальне значення обраного параметра, що дозволяє судити про коректність роботи системи.

У частини блоків є можливість завантажити дамپ ЕЕПРОМ. При зчитуванні спочатку задається назва файлу, куди буде записаний завантажений ЕЕПРОМ (за замовчуванням – C:\EEPROM_.bin). Потім необхідно натиснути на кнопку – "Download".

У частини блоків є можливість завантажити FLASH пам'ять. Спочатку задається назва файлу, куди буде записаний завантажений FLASH (за замовчуванням – C:\FLASH_.bin). Потім необхідно натиснути на кнопку – "Download".

Існує вікно, яке служить для одночасного відображення 4 параметрів у вигляді графіків. Потрібний параметр вибирається зі спадаючого списку (Parameter). Також фіксується для кожного параметра його поточне, мінімальне, максимальне й середнє значення.

Деякі блоки мають можливість програмування певних параметрів системи, наприклад, зміни заданих обертів холостого ходу. Дане вікно призначене для виконання цієї процедури. Спочатку необхідно вибрати з лівого списку, який параметр необхідно змінити. Праворуч у верхньому вікні з'явиться поточне значення даного параметра. Далі, пересуваючи бігунок, можна змінити його бажаним образом, причому нове значення буде знизу. Після вибору потрібного параметра необхідно натиснути на кнопку (Program New Value), тим самим змінений параметр збережеться в ECU і буде постійно зберігатися при роботі блоку управління.

TOPDON ArtiDiag600S

TOPDON ArtiDiag600S – один із найкращих універсальних діагностичних інструментів, доступних у JB Tools. Він забезпечує професійний рівень продуктивності за ціною, яка підходить як для домашніх майстрів, так і для технічних спеціалістів.

Цей сканер пропонує повну діагностику системи, що дозволяє отримати доступ до ключових систем автомобіля, таких як PCM, TCM, ABS та SRS.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Завдяки чутливому сенсорному екрану та простому інтерфейсу, ним легко користуватися одразу після розпакування.

Основні характеристики:

- Повна системна діагностика основних систем автомобіля.
- 8 сервісних функцій, включаючи скидання рівня оливи, EPB та SAS.
- Довічні безкоштовні оновлення програмного забезпечення.
- Швидка ідентифікація транспортного засобу AutoVIN.
- Компактний та портативний дизайн.

ArtiDiag600S вирізняється тим, що пропонує розширені можливості без високої вартості, яка зазвичай пов'язана з професійними інструментами.

Найкраще для: майстрів-аматорів та техніків, які шукають високу продуктивність та цінність

Autel MaxiCheck MX900

Для розширеної діагностики Autel MaxiCheck MX900 є одним із найпотужніших інструментів, доступних у JB Tools. Він розроблений для користувачів, яким потрібен глибший доступ та контроль над системами автомобіля.

Однією з його видатних особливостей є двонаправлене керування, яке дозволяє активно тестувати такі компоненти, як паливні насоси, реле та виконавчі механізми. Це полегшує швидке та точне виявлення проблем.

Основні характеристики:

- Повна діагностика всіх основних систем.
- Двонаправлене керування для тестування активних компонентів.
- Розширені сервісні функції та скидання налаштувань для обслуговування.
- Великий сенсорний дисплей.
- Широка сумісність з транспортними засобами.

MX900 ідеально підходить для професійних майстерень та досвідчених користувачів, яким потрібна надійна, глибока діагностика сучасних автомобілів.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Найкраще для: професійних механіків та досвідчених користувачів

Тестовий комплект Relay Buddy Pro від Electronic Specialties 193

Не кожену проблему можна діагностувати за допомогою сканера. Електричні проблеми часто потребують спеціалізованого обладнання, і комплект для тестування реле Electronic Specialties Relay Buddy Pro є обов'язковим для такого типу роботи.

Цей інструмент призначений для перевірки релейних ланцюгів у системах на 12 та 24 вольти. Він допомагає швидко виявити несправні реле та проблеми з проводкою, зменшуючи здогадки та заощаджуючи час під час ремонту.

Основні характеристики:

- Тестує релейні кола 12 В та 24 В.
- Швидко виявляє несправні реле.
- Включає адаптери для різних типів реле.
- Ідеально підходить для діагностики електричних проблем.
- Міцна конструкція для використання в магазині.

Електрична діагностика може бути складною, але такі інструменти, як Relay Buddy Pro, роблять процес набагато ефективнішим.

Найкраще підходить для: техніків, які працюють з електричними системами та діагностикою реле

Launch Millennium 60

Якщо вам потрібне просте та доступне рішення, зчитувач кодів Launch Millennium 60 – чудовий варіант, особливо завдяки зниженій пропозиції, доступній у JB Tools.

Цей інструмент зосереджений на основах, швидко зчитуючи та очищуючи коди несправностей двигуна. Він ідеально підходить для водіїв, які хочуть мати простий спосіб зрозуміти, як працює індикатор «Check Engine».

Основні характеристики:

- Зчитує та очищує коди помилок двигуна.
- Просте підключення та працюйте.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

- Компактний та зручний для зберігання.
- Вигідний варіант розпродажу.

Хоча він не містить розширених функцій, це надійний інструмент початкового рівня, який виконує свою роботу.

Найкраще для: початківців та користувачів з обмеженим бюджетом.

Купуючи в JB Tools, ви отримуєте доступ до високоякісних автомобільних діагностичних інструментів від перевірених брендів, таких як TOPDON, Autel, Electronic Specialties та Launch.

Переваги покупок з JB Tools:

- Конкурентні ціни на професійні інструменти.
- Широкий вибір для будь-якого рівня кваліфікації.
- Перевірені бренди, якими користуються професіонали.
- Швидка доставка та надійний сервіс.

Незалежно від того, чи ви оновлюєте свій гараж, чи облаштовуєте професійну майстерню, JB Tools має необхідне діагностичне обладнання.

Чому діагностичні інструменти є важливими у 2026 році

Сучасні транспортні засоби залежать від складних комп'ютерних систем для контролю продуктивності, безпеки та викидів. Без діагностичного інструменту виявлення проблем стає складним та трудомістким процесом.

Якісний сканер дозволяє вам:

- Швидке визначення кодів несправностей.
- Контролюйте продуктивність автомобіля в режимі реального часу.
- Виконайте скидання налаштувань для обслуговування.
- Тестові компоненти та системи.
- Уникайте непотрібного ремонту.

Наявність правильного діагностичного інструменту дає вам контроль та впевненість під час роботи із сучасними транспортними засобами.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Як вибрати правильний діагностичний інструмент

Вибір правильного інструменту залежить від вашого рівня досвіду та виду роботи, яку ви виконуєте.

- Початківець. Базовий зчитувач коду, такий як Launch Millennium 60, – чудова відправна точка.
- Для любителів самостійного ремонту TOPDON ArtiDiag600S пропонує розширені функції за низькою ціною
- Професійний технік Autel MaxiCheck MX900 забезпечує повний доступ до системи та двонаправлене керування
- Електрична діагностика. Relay Buddy Pro необхідний для усунення електричних проблем.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – динамічна інтерпретована об'єктно-орієнтована скриптова мова програмування із строгою динамічною типізацією. Офіційний сайт мови програмування Python <https://www.python.org/>. Python – багатоцільова мова програмування, яка дозволяє писати код, що добре читається. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротше свого аналога, написаного на іншій мові. Python – багатоплатформова мова програмування. Це означає, що програми на Python можна запускати в різних операційних системах без будь-яких змін.

Ще однією перевагою Python є його стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, для побудови графічного інтерфейсу програм тощо. Програми, написані на мові програмування Python, можуть бути як невеликими скриптами, так і складними системами. Python абсолютно безкоштовний.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Швидкість виконання коду Python

Один з можливих недоліків Python – швидкість виконання коду. Python не є компільованою мовою. Код на Python спочатку компілюється у внутрішній байт-код, який потім виконується інтерпретатором Python. У більшості випадків при використанні Python виходять програми повільніші в порівнянні з такими мовами, як C.

Втім, сучасні комп'ютери мають таку обчислювальну потужність, що для більшості застосунків швидкість розробки важливіша швидкості виконання, а програми на Python зазвичай пишуться набагато швидше.

Окрім того, Python легко розширюється модулями, написаними на C або C++. Такі модулі можуть використовуватися для виконання частин програми, що створюють інтенсивне навантаження на процесор.

Використання Python

Python використовується для різних цілей: для створення ігор і веб-застосунків, розробки внутрішніх інструментів для різноманітних проектів. Мова також широко застосовується в науковій області для досліджень і розв'язування прикладних завдань.

Застосування мови програмування Python:

1. BitTorrent – протокол для обміну даними.
2. Ubuntu Software Center – вільне програмне забезпечення для пошуку, установки і видалення пакунків в системі Ubuntu Linux.
3. Blender – програма для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, пост-обробки відео, а також створення відеоігор.
4. GIMP – растровий графічний редактор, із підтримкою векторної графіки.
5. World of Tanks.
6. Вільна енциклопедія Вікіпедія.
7. Пошукова система Google.

8. DropBox – файловий хостинг, що включає персональне хмарне сховище, синхронізацію файлів і програму-клієнт.

9. YouTube – популярне відеосховище.

Версії Python

Мови програмування з часом змінюються – розробники додають в них нові можливості, а також виправляють помилки. Так з’являються різні версії мови. Наприклад, код написаний на Python 2 у більшості випадків не буде працювати у версії Python 3 без внесення додаткових змін.

Процесор є найважливішим компонентом в комп’ютері. Одна з основних функцій процесора – це обробка даних згідно комп’ютерної програми, яка є списком інструкцій, шляхом виконання арифметичних і логічних операцій над фрагментами даних.

Кожна інструкція в програмі – це команда, яка «повідомляє» процесору, яку операцію він повинен виконати. Процесор комп’ютера може розуміти лише ті інструкції, які написані на машинній мові. Машинна мова – це штучна мова, створена для передачі команд комп’ютеру. За допомогою машинної мови створюються ефективні програми, оскільки розробник отримує доступ до всіх можливостей процесора. Машинна мова – мова низького рівня.

Інструкція машинної мови існує для кожної операції, яку процесор здатний виконати – є інструкція для додавання чисел, є інструкція для віднімання чисел і т.д. Увесь набір інструкцій, який центральний процесор може виконати, відомий як набір інструкцій процесора.

Наприклад, у вас є певна програма, яка зберігається на диску вашого комп’ютера. Для виконання програми, ви здійснюєте подвійний клік на значку програми. Це змушує програму копіюватися з диска в оперативну пам’ять, після чого процесор комп’ютера виконує копію програми, яка знаходиться в оперативній пам’яті.

Коли процесор виконує інструкції програми, він бере участь у процесі, який є відомим як цикл fetch – decode – execute (отримати – декодувати –

виконати). Цей цикл виконується для кожної інструкції у програмі і складається з трьох кроків:

Отримати

Програма – це послідовність інструкцій на машинній мові. Першим кроком циклу є завантаження (отримання) наступної інструкції з пам'яті в процесор.

Декодувати

Інструкція машинної мови – це двійкове число, яке представляє команду, що повідомляє процесору виконати певну операцію. На цьому кроці процесор декодує інструкцію, яку було «витягнуто» з пам'яті, для визначення того, яка операція повинна виконуватись.

Виконати

Останній крок циклу – виконати операцію.

Хоча процесор комп'ютера розуміє тільки машинну мову, людині непрактично писати програми на машинній мові. Така програма може мати тисячі або навіть мільйони бінарних інструкцій, і написання такої програми буде дуже обтяжливим процесом.

З цієї причини була створена мова асемблера як альтернатива машинній мові. Замість використання двійкових чисел для написання інструкцій, мова асемблера використовує короткі слова, відомі як мнемокоди.

Незважаючи на те, що мова асемблера не вимагає двійкових інструкцій, як у випадку машинної мови, проте вона вимагає високих знань про процесор. Використовуючи мову асемблера, навіть для найпростішої програми, необхідно написати велику кількість інструкцій.

Мова програмування високого рівня дозволяє створювати складні програми, не знаючи, як працює процесор, і не записуючи великої кількості інструкцій низького рівня. Крім того, більшість мов програмування високого рівня використовують слова, які легко зрозуміти.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Python – одна із популярних сучасних мов програмування високого рівня. Python – інтерпретована мова програмування. Python – це високорівнева інтерпретована мова програмування, на відміну від C++, яка є прикладом компільованої мови програмування. Назва Python відноситься як до мови програмування, так і до інтерпретатора – комп'ютерної програми, яка зчитує початковий код (написаний на Python) і виконує інструкції (команди).

Для перекладу мови високого рівня на машинну мову доступні два типи програм:

1. Компілятор.
2. Інтерпретатор.

Завантаження Python

Версії інтерпретатора Python для різних операційних систем доступні для безкоштовного завантаження за адресою <https://www.python.org/downloads>.

Середовище програмування для Python

Для написання програм використовують текстові редактори або інтегровані середовища розробки, які включають в себе різні інструменти для роботи з кодом: засіб для написання коду (текстовий редактор), інтерактивний інтерпретатор, відлагоджувач тощо.

Текстові редактори та інтегровані середовища програмування для Python:

- IDLE – стандартний редактор Python. Встановлюється разом з Python для користувачів Windows, окремим пакунком для користувачів Linux.
- Notepad++ – безкоштовний текстовий редактор початкового коду, який підтримує велику кількість мов, в тому числі і Python. Лише для користувачів Windows.
- Visual Studio Code – це легкий, але потужний редактор початкового коду, який розповсюджується безкоштовно і доступний у версіях для платформ Linux, Windows і macOS.
- PyScripter – інтегроване середовище розробки для мови програмування Python. Для користувачів Windows. Поширюється безкоштовно.

– Wing IDE 101 – вільне інтегроване середовище для Python, розроблене для навчання програмістів-початківців. Для користувачів Linux, Windows і macOS. Поширюється безкоштовно.

– Geany – вільний текстовий редактор з базовими елементами інтегрованого середовища розробки, доступний для операційних систем Linux, Windows і macOS.

– PyCharm – інтегроване середовище розробки для мови програмування Python. PyCharm є власницьким програмним забезпеченням. Наявна безкоштовна версія Community з усіченим набором можливостей. Для користувачів Linux, Windows і macOS.

– Thonny – IDE для вивчення програмування мовою Python. Для користувачів Linux, Windows і macOS.

– Mu – редактор коду Python для програмістів-початківців. Для користувачів Linux, Windows і macOS.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи мережевих прикінцевих пристроїв автомобіля.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Сучасне автомобільне програмне забезпечення призначене для вирішення різноманітних завдань, від керування невеликими датчиками для допомоги водієві до прийняття рішень у режимі реального часу для автономного водіння. Таке розмаїття завдань означає, що як виробник оригінального обладнання (ОЕМ) або постачальник автомобільного програмного забезпечення, вам потрібно переконатися, що ваші продукти є високобезпечними, добре продуктивними та адаптованими до постійно мінливих вимог ринку.

У цьому розділі ми обговорюємо відповідні тенденції розробки автомобільного програмного забезпечення та аналізуємо, яких труднощів слід очікувати та як їх вирішувати, прагнучи створити високоякісне індивідуальне програмне рішення для автомобілів. Ми також розглядаємо ключові аспекти розробки автомобільного програмного забезпечення, щоб допомогти вам зрозуміти нюанси цього процесу.

Цей розділ буде корисним для виробників оригінального обладнання (ОЕМ) та постачальників автомобільного програмного забезпечення, які хочуть зрозуміти вплив розробки індивідуального автомобільного програмного забезпечення на вирішення ключових галузевих проблем та хочуть створити або вдосконалити власні інтелектуальні, підключені та автономні транспортні системи.

Що таке розробка автомобільного програмного забезпечення?

Розробка автомобільного програмного забезпечення – це процес створення спеціалізованих додатків та систем для сучасних транспортних засобів. Залежно від формату, таке програмне забезпечення може керувати однією або кількома важливими функціями керування транспортним засобом, зокрема:

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

- Управління акумулятором або силовим агрегатом.
- Зв'язок між транспортним засобом та всім (V2X).
- Інформаційно-розважальна система.
- Допомога водієві.
- Автономне водіння.
- Розширені системи допомоги водієві (ADAS).

Щоб створити безпечне та ефективне автомобільне програмне забезпечення, вам знадобиться команда досвідчених інженерів-програмістів з глибоким розумінням механіки транспортних засобів, відповідних технологічних тенденцій та специфіки кібербезпеки. А оскільки сучасні розумні автомобілі можуть стати мішенню для кіберзлочинців, впровадження принципів безпечного життєвого циклу розробки програмного забезпечення (SDLC) має вирішальне значення для зміцнення вашого продукту від можливих кібератак.

Поширені типи програмних рішень для автомобілів

Можна окреслити кілька ключових категорій автомобільних програмних рішень, кожна з яких відповідає за управління певними функціями транспортного засобу:

- Вбудоване програмне забезпечення та драйвери відповідають за основні функції автомобіля, такі як керування двигуном, інтеграція датчиків та інформаційно-розважальна система. Вбудовані рішення забезпечують зв'язок у режимі реального часу між апаратними та програмними компонентами автомобіля, забезпечуючи їх безперебійну роботу.

- Рішення для силових агрегатів та програмне забезпечення для керування акумуляторами (BMS) оптимізують споживання палива в традиційних автомобілях та керують часом роботи та продуктивністю акумулятора в електромобілях (EV). Такі рішення контролюють стан силового агрегату або акумулятора, щоб покращити продуктивність та запобігти таким проблемам, як перегрів та перезарядження.

- Передові системи керування транспортними засобами інтегрують

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

передові системи допомоги водієві (ADAS) та технології автономного водіння. Ці системи обробляють дані датчиків у режимі реального часу, щоб допомагати водіям або замінювати їх, підтримуючи автономність від 2 до 5 рівнів. Такі функції, як адаптивний круїз-контроль та запобігання зіткненням, підвищують безпеку, а штучний інтелект та машинне навчання забезпечують автономні можливості.

– Інформаційно-розважальні системи забезпечують навігацію, розваги та зв'язок як для водіїв, так і для пасажирів. Ці системи можуть інтегруватися із зовнішніми пристроями, такими як смартфони та планшети, а також із такими сервісами, як потокове передавання медіа та голосові помічники.

– Програмне забезпечення для збору даних та діагностики збирає дані про продуктивність транспортного засобу з різних компонентів та аналізує їх для виявлення несправностей та неефективності. Разом вони забезпечують можливості прогнозного обслуговування та допомагають своєчасно виявляти та запобігати несправностям транспортного засобу.

– Телематичні системи забезпечують зв'язок у режимі реального часу між транспортними засобами та зовнішніми системами, підтримуючи дистанційну діагностику, управління автопарком та прогнозне обслуговування. Телематика є важливою для підключених транспортних засобів та зв'язку V2X.

– Рішення з кібербезпеки для підключених транспортних засобів захищають конфіденційні дані та критично важливі операції в підключених транспортних засобах, захищаючи їх від кібератак. Ці рішення зазвичай включають системи виявлення вторгнень, технології шифрування та рішення для моніторингу в режимі реального часу.

Тепер, коли ми дослідили ключові типи автомобільного програмного забезпечення, давайте обговоримо тенденції в розробці програмного забезпечення для автомобільних НМІ, які можуть вплинути на ваш підхід до створення такого програмного забезпечення.

Ключові тенденції у розробці автомобільного програмного забезпечення

Нові технологічні досягнення та зміни в уподобаннях клієнтів безпосередньо впливають на розвиток тенденцій розробки автомобільного програмного забезпечення. Чим ефективніше ви обираєте та враховуєте ці тенденції для задоволення потреб ваших клієнтів, тим вищі ваші шанси випередити конкурентів та домінувати на ринку.

Давайте розглянемо деякі з найважливіших технологічних рішень, на які повинна звернути увагу ваша команда розробників:

1. Електрифікація

EFESO прогнозує, що до 2030 року кожен другий автомобіль, проданий у Європі, буде повністю електричним. Зі прискоренням переходу на електромобілі також зростає потреба в передовому програмному забезпеченні для керування силовими агрегатами та акумуляторами. Ефективне програмне забезпечення для керування акумуляторами допомагає максимізувати запас ходу автомобіля та контролювати стан акумулятора, підтримуючи довгострокову продуктивність та надійність.

Хоча Tesla може бути одним із найвідоміших брендів електромобілів, зараз очевидно, що весь автомобільний ринок все більше надає пріоритет електромобілям. Великі виробники, такі як BMW, Hyundai та Volvo, активно розширюють свої портфелі електромобілів та зміщують довгострокові стратегії в бік моделей з акумуляторним живленням. Як результат, виробники оригінального обладнання (OEM) більше інвестують у платформи та дослідження, орієнтовані на електромобілі, включаючи оптимізацію систем транспортних засобів та технологій акумуляторів, щоб підвищити ефективність, масштабованість та конкурентоспроможність на ринку, орієнтованому на електромобілі.

2. Підключення

Електричні чи ні, сучасні транспортні засоби можуть взаємодіяти з навколишнім середовищем інакше, ніж автомобілі, випущені десять чи два роки

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

тому. Зі зростанням переходу до програмно-визначених транспортних засобів програмне забезпечення стає ключовим елементом, що визначає функціональність автомобіля. Наприклад, прошивка Інтернету речей дозволяє обмін даними в режимі реального часу, дистанційну діагностику та управління автопарком, створюючи повністю підключену автомобільну екосистему.

Компанії вже переходять до концепцій 5G Advanced і навіть ранніх 6G, щоб забезпечити наднизьку затримку, автономність на периферії та прийняття рішень у режимі реального часу. Цей перехід допомагає транспортним засобам обробляти більше даних локально та менше покладатися на хмару.

Розумна інфраструктура також стає важливою. Виробники оригінального обладнання (OEM) розгортають придорожні пристрої, датчики по всьому місту та дорожні системи на основі штучного інтелекту, щоб транспортні засоби могли краще розуміти навколишнє середовище та реагувати на нього. Це призводить до безпечнішого, екологічнішого та ефективнішого транспорту. Зв'язок та телематика V2X також забезпечують обмін даними в режимі реального часу, оновлення по бездротовій мережі (OTA) та дистанційну діагностику, необхідні для розумнішого управління дорожнім рухом та підвищення безпеки.

3. Автономія

Програмне забезпечення для автономного водіння швидко розвивається, особливо в сфері автономності рівня 3+. Сучасні транспортні засоби використовують радарні та лідарні системи для розуміння навколишнього середовища та навігації в складних дорожніх умовах зі все більшою точністю.

Серед найвідоміших компаній, що розробляють автомобілі з автономним керуванням, є Waymo, Kodiak Robotics та Gatik. Однак ця технологія потребує вдосконалення, зокрема з точки зору безпеки водіння. Більшість автономних систем все ще потребують нагляду людини та мають обмежене застосування.

Для підтримки цих досягнень виробникам оригінального обладнання (OEM) необхідно використовувати надійне проміжне програмне забезпечення: наприклад, ROS 2 та оновлені архітектури AUTOSAR, що підтримують хмарні

розгортання. За допомогою цих технологій компанії отримують кращу оркестрацію модулів сприйняття, планування та керування. Ці технології також допомагають командам розробників масштабувати автономні системи на різних апаратних платформах та конфігураціях парків обладнання.

Більше того, моделювання в реальному часі, високоточне моделювання датчиків та безперервна перевірка сценаріїв стають основними компонентами автономних конвеєрів розробки. Як наслідок, складність життєвих циклів програмного забезпечення зростає, оскільки саме програмне забезпечення вимагає більш складних інструментів та сумісності між платформами моделювання, вбудованими системами та серверними службами.

4. Розширена персоналізація

Можливості голосового керування та інтелектуальні персональні асистенти дозволяють взаємодіяти без використання рук, зменшуючи відволікання водія та підвищуючи комфорт.

Сучасні інформаційно-розважальні системи також пропонують розширені можливості персоналізації для пасажирів, включаючи розваги на борту та доступ до інформації в режимі реального часу. Окрім інформаційно-розважальних систем, персоналізація тепер включає моделювання поведінки водія, адаптивне макетування інтерфейсу користувача та контекстно-залежні рекомендації на основі стилю водіння, місця розташування та навіть моделей використання автомобіля. Ці функції допомагають автовиробникам забезпечувати більш захопливий користувацький досвід, водночас вирізняючи свої автомобілі на дедалі конкурентнішому ринку.

5. Кібербезпека

Зі збільшенням зв'язку виникає більша потреба в надійних заходах кібербезпеки, особливо з огляду на зростаюче впровадження розумних автомобілів та передових інформаційно-розважальних рішень.

Сучасні стратегії розвитку автомобільної галузі тепер включають:

– безпечні практики SDLC;

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

- апаратно-закріплена ідентифікація;
- постійний моніторинг безпеки.

Крім того, деякі компанії вже досліджують інтеграцію з новими стандартами постквантової криптографії (PQC) як частину свого довгострокового планування безпеки. Оскільки транспортні засоби стають все більш програмно-орієнтованими, виробники оригінального обладнання (OEM) повинні підвищувати стійкість кібербезпеки на кожному рівні, від бортових мереж до хмарної інфраструктури. Основна увага спільноти розробників зосереджена на забезпеченні належної безпеки цих систем, щоб запобігти інцидентам, які можуть призвести до крадіжки автомобілів, несанкціонованого вилучення даних або несанкціонованого контролю над критично важливими функціями транспортного засобу.

6. Штучний інтелект

Штучний інтелект стає одним із найпотужніших рушійних сил розробки автомобільного програмного забезпечення. Виробники оригінального обладнання (OEM) тепер використовують інструменти штучного інтелекту для пришвидшення розробки, підвищення безпеки та підвищення продуктивності системи в усьому транспортному засобі.

Моделювання на базі штучного інтелекту, цифрові двійники та сценарне тестування допомагають перевірити функції автономних систем та систем допомоги водієві (ADAS) у мільйонах віртуальних ситуацій, перш ніж вони потраплять на дорогу. Такі інструменти скорочують час розробки, зменшують витрати та зменшують потребу у фізичному тестуванні.

Синтетичні дані допомагають тестувати рідкісні та ризиковані сценарії водіння, які важко вловити в реальному житті, що важливо для навчання точних моделей сприйняття.

Нові рішення на основі штучного інтелекту, такі як агентний штучний інтелект та багатоагентні системи прийняття рішень, також набирають обертів. Вони підтримують більш скоординоване прийняття рішень між підсистемами

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

транспортних засобів. Водночас, інші пілоти на основі штучного інтелекту допомагають інженерним командам виконувати такі завдання, як:

- Генерація коду.
- Автоматизація тестування.
- Виявлення аномалій.
- Налаштування та оркестрація симуляції.

Технології MLOps та autoMLOps зараз мають вирішальне значення для управління повним життєвим циклом моделей безпілотних автомобілів та систем допомоги водієві (ADAS). Ці технології забезпечують постійне перенавчання, перевірку та безпечне розгортання моделей на різному обладнанні та в автопарках.

Щоб йти в ногу з цими тенденціями, виробники оригінального обладнання (OEM) та постачальники впроваджують спеціалізовані платформи, мови програмування та фреймворки розробки, які підтримують швидшу доставку, надійніші гарантії безпеки та безперешкодну інтеграцію в складні системи транспортних засобів. Давайте розглянемо ключові інструменти та технології, що використовуються для сучасної розробки автомобільного програмного забезпечення.

3.2 Розробка структурної схеми

Інструменти та технології для розробки автомобільного програмного забезпечення

Залежно від поставленого завдання, розробники використовують різні інструменти та технології розробки автомобільного програмного забезпечення, від загальних мов програмування до передових фреймворків зворотного проектування. Нижче ми коротко розглянемо деякі з найпоширеніших:

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Давайте детальніше розглянемо кожну з цих категорій.

Мови програмування

Хоча вибір мов програмування практично безмежний, найпоширенішими мовами для написання автомобільного програмного забезпечення є:

- C/C++ – необхідний для створення вбудованих систем, що керують функціями двигуна, гальмівними системами та інформаційно-розважальними платформами. Ці мови забезпечують високу продуктивність та виконання в режимі реального часу.

- Rust – дедалі частіше використовується для критично важливих для безпеки застосувань завдяки зосередженості на безпеці пам'яті та паралельності, що робить його придатним для керування автономними транспортними засобами.

- Python – широко використовується для розробки штучного інтелекту та аналізу даних, особливо в системах автономного водіння та ADAS, де обробка даних у режимі реального часу є критично важливою.

- Golang – відомий своєю простотою, сильною моделлю паралельного використання та швидким виконанням. Go часто використовується для створення бекенд-сервісів, телематичних платформ, систем оновлень OTA та хмарних компонентів, які потребують надійного масштабування.

- Carbon – розроблений для спрощення розробки, зберігаючи при цьому високу продуктивність. Інтерес до Carbon зростає для майбутнього автомобільного інструментарію, прототипування та критично важливих для продуктивності застосувань, де потрібен безпечніший та зручніший у підтримці код.

Платформи розробки

Стандартизовані платформи та фреймворки спрощують швидке створення якісних автомобільних продуктів. Інструменти, які можуть бути корисними для вашої команди розробників, включають:

- MATLAB та Simulink – популярні для проектування та моделювання на основі моделей. Ці інструменти дозволяють інженерам моделювати компоненти

транспортних засобів, такі як силові агрегати та гальмівні системи, перед розгортанням, забезпечуючи надійність та продуктивність.

– QNX – операційна система реального часу, що використовується в системах ADAS, інформаційно-розважальних системах та інших критично важливих системах.

– AUTOSAR та Adaptive AUTOSAR – фреймворки, що забезпечують масштабованість та сумісність програмного забезпечення для керування в режимі реального часу, гарантуючи безперебійну комунікацію між компонентами транспортного засобу.

– ROS 2 – широко поширена платформа робототехніки, що використовується в дослідженнях та прототипуванні автономного водіння. ROS 2 забезпечує можливості зв'язку, об'єднання датчиків та проміжного програмного забезпечення, що допомагають командам створювати та тестувати складні автономні системи на основі компонентів робототехніки.

Розробка штучного інтелекту

Моделі штучного інтелекту відіграють вирішальну роль у різних автомобільних завданнях, від побудови системи прогнозного технічного обслуговування до впровадження розширеної допомоги водієві. Інструменти, які допоможуть вашій команді створити безпечну та ефективну модель штучного інтелекту, включають:

– TensorFlow та PyTorch – широко використовуються для створення моделей на основі штучного інтелекту для автономного водіння та об'єднання датчиків.

– OpenCV – критично важлива бібліотека для розробки систем комп'ютерного зору, що дозволяє транспортним засобам обробляти візуальні дані та приймати рішення.

– Моделі OpenAI – їх можна налаштовувати та точно налаштовувати для конкретних автомобільних застосувань, що пришвидшує процес розробки та заощаджує ваш бюджет.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

– Платформи розробки на основі штучного інтелекту – DataRobot, H2O.ai та Amazon SageMaker – дозволяють командам оптимізувати навчання, розгортання та моніторинг моделей, забезпечуючи масштабованість та безпеку.

Зворотне проектування

Від забезпечення сумісності з погано документованими сторонніми рішеннями до підтримки застарілих продуктів без документації та виправлення критичних проблем безпеки, зворотне проектування незамінне, коли йдеться про створення безпечного та добре продуктивного програмного продукту. Інструменти, які ваша команда може використовувати для зворотного проектування, включають:

– IDA Pro та Ghidra – популярні інструменти зворотного проектування, які зазвичай використовуються для аналізу застарілих систем та забезпечення сумісності із сучасним програмним забезпеченням.

– Hex-Rays – декомпілятор IDA для роботи зі скомпільованим кодом, що дозволяє оновлювати або модифікувати старе програмне забезпечення транспортного засобу.

CI/CD та DevOps

Використання конвеєрів CI/CD допомагає керувати складністю тестування та розгортання програмного забезпечення в різних системах транспортних засобів. Ось деякі інструменти, які ваша команда може використовувати для цих завдань:

– Jenkins – Автоматизує процеси складання та тестування, забезпечуючи безперебійну інтеграцію оновлень у існуючі системи транспортних засобів.

– Docker та Kubernetes – інструменти контейнеризації для керування масштабними розгортаннями, що забезпечують узгоджену поведінку програмного забезпечення на різних платформах.

З такою кількістю інструментів для використання та такою кількістю завдань, які потрібно вирішити, процес створення автомобільного програмного забезпечення може стати складним. Давайте розглянемо основні проблеми, з

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

якими команда розробників, ймовірно, зіткнеться під час створення програмних рішень для автомобілів.

Проблеми розробки автомобільного програмного забезпечення

Постачальники автомобільного програмного забезпечення стикаються з різними викликами, від забезпечення кібербезпеки підключених систем до масштабування оновлень програмного забезпечення на мільйони автомобілів. Нижче ми аналізуємо основні пастки, до яких розробники автомобільного програмного забезпечення повинні бути готові.

1. Забезпечення надійної кібербезпеки

Кіберзлочинці зазвичай атакують підключені до мережі транспортні засоби та ті, що оснащені автономними або навіть розширеними функціями UX. Зловмисники можуть використовувати вразливості в програмному коді для надсилання команд з неавторизованого облікового запису або компрометації платежів в автомобілі, що є загрозами, які намагаються зменшити як Visa, так і Mastercard. Зловмисники можуть атакувати дані датчиків, алгоритми автономного водіння або платформи підключених транспортних засобів, що потенційно може призвести до витоків даних або, що ще гірше, поставити під загрозу безпеку пасажирів. Тому забезпечення надійної кібербезпеки має бути головним пріоритетом для розробників автомобільного програмного забезпечення.

Щоб покращити кібербезпеку вашого автомобільного програмного продукту, розгляньте можливість шифрування всіх конфіденційних даних і критично важливих транзакцій, впровадження моніторингу безпеки в режимі реального часу та механізмів виявлення вторгнень для захисту транспортних засобів з покращеним програмним забезпеченням від зовнішніх загроз. Безпечні оновлення ОТА в автомобільній галузі також необхідні для розгортання останніх патчів безпеки та достатньо швидкого виправлення вразливостей.

2. Управління масштабованістю та оновленнями

Оскільки програмне забезпечення стає центральним фактором функціональності транспортних засобів, важливо синхронізувати оновлення

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

програмного забезпечення для мільйонів автомобілів по всьому світу. Розробники автомобільного програмного забезпечення повинні забезпечити безперебійне оновлення широкого спектру електронних блоків керування (ЕБУ), датчиків та систем зв'язку в кожному транспортному засобі.

Оновлення ОТА можуть допомогти виробникам виправляти вразливості безпеки, виправляти помилки та дистанційно розгортати нові функції. Однак управління всіма цими оновленнями в масштабах та в режимі реального часу також створює значні труднощі, від фрагментації програмного забезпечення та проблем із продуктивністю до появи нових прогалин у безпеці.

Щоб вирішити ці проблеми, важливо спланувати та впровадити ефективну систему управління, яка допоможе виробникам організувати та спростити масштабування цих оновлень.

3. Інтеграція програмного та апаратного забезпечення

Автомобільне програмне забезпечення повинно безперешкодно інтегруватися з різноманітними датчиками, виконавчими механізмами та вбудованими системами, водночас враховуючи обмеження базового обладнання. На відміну від споживчого програмного забезпечення, яке часто може працювати на апаратному забезпеченні загального призначення, автомобільне програмне забезпечення безпосередньо взаємодіє з фізичними компонентами транспортного засобу.

Ця інтеграція є особливо складною, оскільки програмне забезпечення повинно враховувати як обмеження в реальному часі, пов'язані з критично важливими для безпеки функціями, так і значну варіабельність апаратних характеристик різних моделей транспортних засобів.

Таким чином, розробникам автомобільного програмного забезпечення необхідно забезпечити адаптивність, стійкість та оптимізацію їхнього програмного забезпечення для апаратного забезпечення, яким воно керує, зокрема для силових агрегатів, систем допомоги водієві (ADAS) та систем керування акумуляторами.

4. Розробка зручного інтерфейсу користувача

Ще одним суттєвим викликом є створення інтерфейсу людина-машина (НМІ), який є одночасно інтуїтивно зрозумілим та адаптивним. Зі зростанням складності технологій в автомобілях, таких як інформаційно-розважальні системи, голосове керування та проекційні дисплеї, вкрай важливо, щоб користувацький досвід був безперебійним та ненав'язливим.

Ось чому ваша команда розробників повинна розробити НМІ, який забезпечить водіям легкий доступ до основних функцій, не відволікаючи їх. Балансування функціональності, безпеки та зручності користувача є важливим для створення успішного автомобільного НМІ.

5. Робота зі старими системами

Багато автовиробників продовжують покладатися на застарілі системи для основних функцій транспортних засобів. Оновлення або інтеграція нового програмного забезпечення в ці старі системи є складним завданням, яке вимагає ретельного зворотного проектування та розробки для забезпечення сумісності, безпеки та безперебійної роботи. Розробники також часто стикаються з труднощами під час міграції застарілого програмного забезпечення на сучасні платформи, зберігаючи при цьому стабільність старих систем. Наприклад, вашій команді розробників, можливо, доведеться працювати із застарілою документацією, аналізувати застарілі архітектури або забезпечувати сумісність нового програмного забезпечення та існуючого обладнання.

Щоб вирішити ці проблеми, вам може знадобитися посилити вашу команду розробників додатковими експертами з реверс-інжинірингу.

6. Дотримання галузевих стандартів та норм

Стандарти та правила, що стосуються розробки програмного забезпечення для автомобільної промисловості, швидко розвиваються, оскільки регуляторні органи працюють над тим, щоб йти в ногу з новими технологіями. Команди розробників повинні бути в курсі цих змін, щоб уникнути прогалин у відповідності, затримок проектів або потенційних регуляторних санкцій.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Існує кілька правил і стандартів, які ваша команда повинна враховувати під час процесу розробки:

– ISO 26262 – визначає вимоги до функціональної безпеки дорожніх транспортних засобів, гарантуючи, що критичні системи (такі як гальмування, рульове керування та ADAS) відповідають суворим рівням цілісності безпеки.

– ISO/SAE 21434 – встановлює вимоги до кібербезпеки автомобільних систем, що охоплюють оцінку ризиків, безпечну розробку та безперервний моніторинг протягом усього життєвого циклу транспортного засобу.

– AUTOSAR – забезпечує стандартизовані програмні архітектури, що допомагають забезпечити сумісність, масштабованість та надійний зв'язок між компонентами транспортного засобу.

– Керівні принципи NHTSA – надають рекомендації щодо автономних систем та систем ADAS, включаючи оновлення щодо показників безпеки, прозорості даних та очікувань щодо звітності для технологій автоматизованого водіння.

– Закон ЄС про штучний інтелект – запроваджує регуляторні вимоги для систем штучного інтелекту з високим рівнем ризику, включаючи багато випадків використання в автомобільній галузі та системах допомоги водієві (ADAS), з особливим акцентом на прозорість, управління даними, управління ризиками та валідацію моделей протягом усього життєвого циклу штучного інтелекту. Ви можете докладніше ознайомитися з цими вимогами в нашій статті про Закон ЄС про штучний інтелект. Залежно від даних, які обробляє конкретне рішення, командам розробників також може знадобитися враховувати вимоги правил та стандартів захисту даних, включаючи GDPR або CCPA для персональних даних та PCI DSS для платіжних даних. Також важливо забезпечити, щоб ваше програмне забезпечення не лише відповідало цим вимогам, але й було достатньо гнучким, щоб адаптуватися до постійно мінливого регуляторного ландшафту. Не кажучи вже про те, що невідповідність може призвести до юридичних наслідків, відкликання продукції або порушення безпеки.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Переваги розробки індивідуального автомобільного програмного забезпечення

Один із ефективних способів вирішення проблем, про які ми говорили вище, – це створення індивідуального автомобільного рішення. Такий підхід пропонує значні переваги для постачальників та виробників автомобільного програмного забезпечення.

1. Адаптуйте продукт до ваших унікальних вимог – Завдяки індивідуальній розробці ви можете спроектувати функції, повністю оптимізовані для архітектури конкретного транспортного засобу, покращуючи все: від продуктивності ADAS до зручності використання HMI. Це призводить до кращого користувацького досвіду, підвищення експлуатаційної ефективності та підвищення надійності.

2. Покращення кібербезпеки – індивідуальне рішення дозволяє вам впроваджувати передові заходи кібербезпеки, орієнтовані на конкретний транспортний засіб, з самого початку. Вирішуючи потенційні вразливості на ранніх етапах процесу розробки, індивідуальні рішення можуть забезпечити надійніший захист від загроз, спрямованих на підключені транспортні засоби, автономні системи та зв'язок V2X.

3. Забезпечення масштабованості та готовності до майбутніх потреб – індивідуальне автомобільне програмне забезпечення створюється з урахуванням масштабованості. Незалежно від того, чи розгортаєте ви програмне забезпечення для всього автопарку, чи керуєте кількома блоками керування двигунами (ECU) в одному транспортному засобі, індивідуальні рішення забезпечують гнучкість, необхідну для ефективного масштабування. Розробка на замовлення також дозволяє інтегрувати можливості оновлення ОТА, гарантуючи, що ваше програмне забезпечення може розвиватися новими функціями та патчами безпеки ще довго після того, як автомобіль покине завод.

4. Покращення інтеграції та модернізації застарілих систем – Розробка на замовлення дозволяє створювати мостові рішення, які сприяють інтеграції

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

застарілих компонентів з новими технологіями, подовжуючи термін служби вашого існуючого обладнання та водночас впроваджуючи нові можливості.

5. Підтримка довгострокової економічної ефективності – Програмне забезпечення на замовлення може бути розроблене для оптимізації продуктивності транспортного засобу, зменшення потреби в частих оновленнях та оптимізації процесів технічного обслуговування, що, своєю чергою, зменшує довгострокові експлуатаційні витрати.

Крім того, індивідуальні рішення зменшують ймовірність проблем з інтеграцією, вирішення яких може бути дорогим, та забезпечують кращу довгострокову підтримку сучасних та майбутніх архітектур транспортних засобів.

Для компаній, які шукають індивідуальні рішення для вирішення цих завдань, Apriorit пропонує провідний галузевий досвід у розробці індивідуального автомобільного програмного забезпечення.

Розробка програмного забезпечення в автомобільній промисловості є важливою для виробників оригінального обладнання (ОЕМ) та постачальників, які створюють розумні, підключені та автономні транспортні засоби. Однак, під час створення автомобільного програмного забезпечення постачальники повинні забезпечувати кібербезпеку, інтегрувати програмне забезпечення зі складним обладнанням та керувати оновленнями в режимі реального часу для мільйонів автомобілів. Розробка програмного забезпечення на замовлення для автомобільної галузі допомагає подолати ці проблеми, пропонуючи індивідуальні рішення, що підвищують безпеку, оптимізують продуктивність та впроваджують масштабовані, перспективні архітектури. У поєднанні з безпечним SDLC цей підхід гарантує ефективність, адаптивність до технологій, що розвиваються, та високий рівень захисту автомобільних систем.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

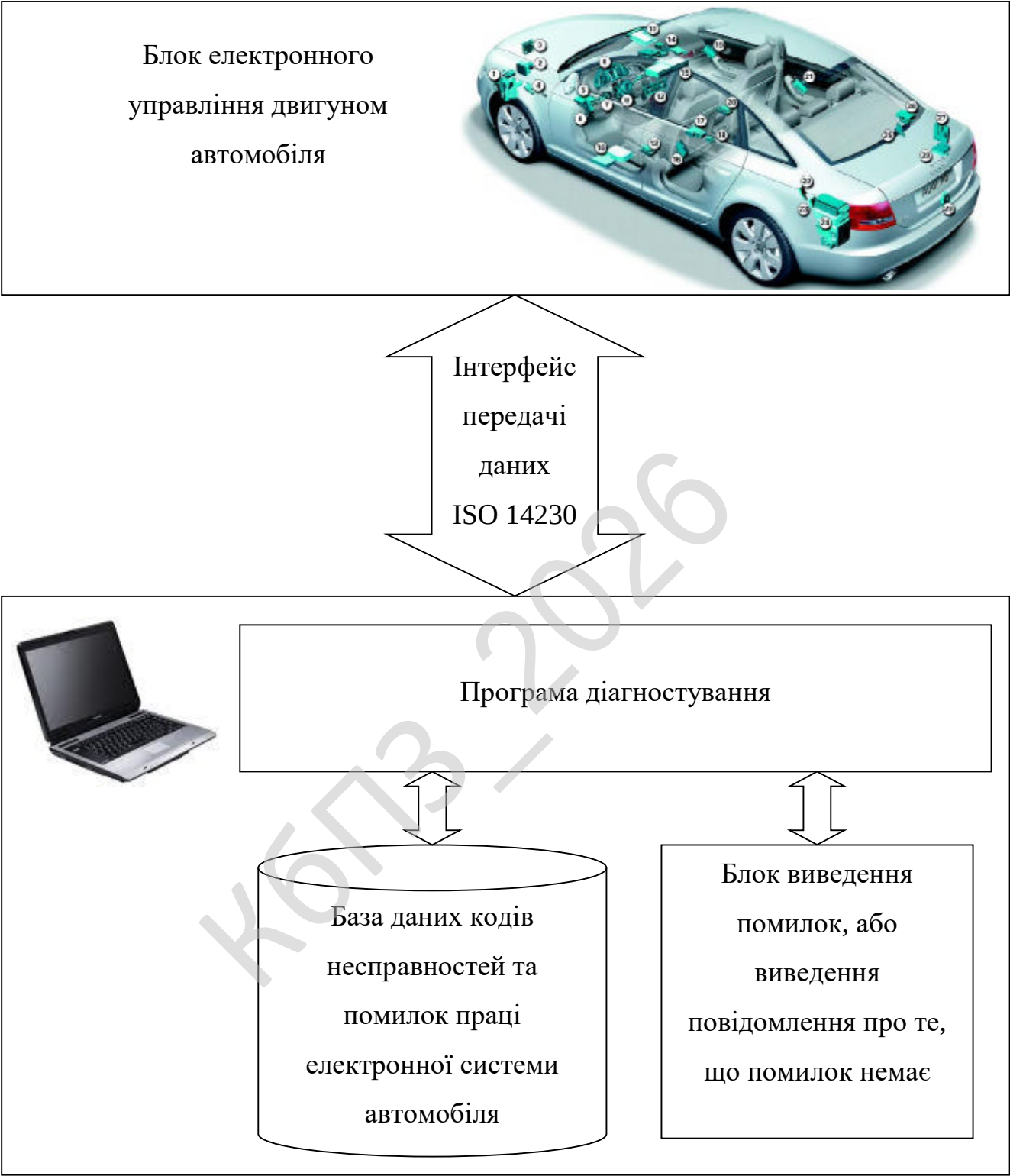


Рисунок 3.1 – Структурна схема системи

Структурна схема системи зображена на рисунку 3.1.

З нього ми бачимо, що основними структурними блоками, які взаємодіють між собою, у ній є наступні:

- Персональний комп'ютер.
- Автомобіль (блок електронного управління двигуном автомобіля).
- Інтерфейс передачі даних ISO 14230.
- Програма діагностування.
- База даних кодів несправностей та помилок праці електронної системи автомобіля.
- Блок виведення помилок, або виведення повідомлення про те, що помилок немає.

Розглянемо як працює розроблена система. Аббревіатура Е-К-В в автомобільній тематиці має на увазі електронний блок управління або ЕБУ. Тобто ECU це Electronic Control Unit. У сучасному автомобілі є безліч різноманітних ECU. Вони відносяться до гальм, трансмісії, підвісі, системі охорони, кліматичній установці, навігації й іншому. Найважливіший це блок управління двигуном. У різних джерелах йому можуть відповідати назви як ECU, так і DME (Digital Motor Electronics), ECM (Engine Control Module), PCM (Powertrain Control Module) і деякі ін. Будемо дотримуватися загальноприйнятого терміна ECU як «електронний блок управління», доповнюючи його вказівкою приналежності в міру появи необхідності.

Які можуть бути блоки управління:

- Блок управління автономного нагрівника.
- Блок управління АБС гальм з EDS.
- Блок управління системи підтримки безпечної дистанції.
- Передавач системи контролю тиску в шинах, передній лівий.
- Блок управління бортовою мережею.
- Блок управління у двері водія.
- Блок управління доступом і старту.

- Блок управління в комбінації приладів.
- Блок управління електронними приладами на кермовому стовпчику.
- Блок управління телефоном, системою телематик; приймально-передавач для телефону.
- Блок управління двигуном.
- Блок управління Climatronic.
- Блок управління регулюванням сидіння із запам'ятовувальним пристроєм і регулюванням кермового стовпчика.
- Блок управління регулюванням дорожнього просвіту.
- Блок управління коректором фар.
- Блок управління системою контролю тиску в шинах.
- Блок управління 2 бортовою мережею.
- Блок управління MMI передньої інформаційно-командної панелі.
- Діагностичний інтерфейс для шин даних.
- Модуль, приймаче/зчитування, системи антен для доступу без ключа.
- CD-чейнджер.
- CD-ROM-дисковід.
- Блок управління в задніх лівих дверях.
- Блок управління системою Air-Bag.
- Датчик швидкості обертання автомобіля навколо вертикальної осі.
- Блок управління у двері переднього пасажир.
- Блок управління регулюванням сидіння переднього пасажир із запам'ятовувальним пристроєм.
- Блок управління в задніх правих дверях.
- Передавач системи контролю тиску в шинах, задній лівий.
- Радіоприймач стояночного нагрівника.
- Блок управління системою навігації з CD-дисководом.
- Блок управління голосовим введенням.

- Блок управління цифровою звуковою системою; радіомодуль; TV-тюнер; цифрове радіо.
- Передавач системи контролю тиску в шинах, задній правий.
- Блок управління системою полегшення паркування.
- Центральний блок управління системою комфорту.
- Блок управління електричним стояночним "ручним" гальмом.
- Блок управління енергопостачанням (менеджер батареї).

3.3 Розробка функціональної схеми

Універсальний алгоритм

Спосіб діагностики, що викладається, використовує принцип: якщо немає прямих доказів виходу ECU з ладу, то варто почати пошук причини неполадки в системі в припущенні справності ECU. Прямих доказів дефектності блоку управління існує всього два. Або ECU має видимі ушкодження, або проблема йде при заміні ECU на свідомо справний (ну, або переноситься на свідомо справний а/м разом з підозрілим блоком; іноді це робити небезпечно, до того ж тут зустрічається виключення, коли блок управління ушкоджений так, що не здатний працювати у всьому діапазоні експлуатаційного розкиду параметрів різних екземплярів однієї й тої ж системи управління, але на одному із двох а/м все-таки працює).

Діагностика повинна розвиватися в напрямку від простого до складного й згідно з логікою роботи системи управління. Саме тому припущення про дефект ECU варто залишити «на потім». Спочатку розглядаються загальні міркування здорового глузду, потім послідовній перевірці підлягають функції системи управління. Ці функції чітко розділяються на забезпечуючі роботу ECU і на функції ECU, що виконуються. Спочатку повинні перевірятися функції забезпечення, потім – функції виконання. У цьому головну відмінність послідовної перевірки від довільної: вона виконується по пріоритеті функцій.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Відповідно, кожний із цих двох видів функцій може бути представлений своїм списком у порядку убутання значимості для роботи системи управління в цілому.

Діагностика успішна тільки тоді, коли вказує на найважливішу із втрачених або порушених функцій, а не на довільний набір таких. Це істотний момент, тому що втрата однієї функції забезпечення може приводити до неможливості роботи декількох функцій виконання. Останні не будуть працювати, але аж ніяк не будуть втрачені, їхня відмова відбудеться просто в результаті причинно-наслідкових зв'язків. Саме тому такі несправності прийнята називати наведеними.

При непослідовному пошуку наведені несправності маскують справжню причину проблеми (досить характерно для діагностики сканером). Зрозуміло, що спроби боротися з наведеними несправностями «у чоло» ні до чого не приводять, повторне сканування ECU дає колишній результат. Ну а ECU «є предмет темний і науковому дослідженню не підлягає», та й замінити його для проби, як правило, нема чим – от схематичні начерки процесу помилкового вибракування ECU.

Отже, універсальний алгоритм пошуку несправності в системі управління такий:

- візуальний огляд, перевірка найпростіших міркувань здорового глузду;
- сканування ECU, читання кодів несправностей (по можливості);
- огляд ECU або перевірка шляхом заміни (по можливості);
- перевірка функцій забезпечення роботи ECU;
- перевірка функцій виконання ECU.

Важлива роль належить докладному опитуванню власника про те, які зовнішні прояви несправності він спостерігав, як виникла або розвивалася проблема, які дії в цьому зв'язку вже були початі. Якщо проблема в системі управління двигуном, варто приділити увагу питанням про сигналізацію (протиугінну систему), так як електрика додаткових устроїв свідомо менш надійна через спрощені прийоми їхньої установки (наприклад, пайка або стандартні з'єднувачі в призначуваних точках розгалуження й розсічення

штатного проведення при підключенні додаткового джгута, як правило, не застосовуються; причому пайка найчастіше не застосовується свідомо через нібито її нестійкість перед вібрацією, що для якісної пайки, звичайно, не так).

Крім того, необхідно точно встановити, який саме а/м перед вами. Усунення скільки-небудь серйозної несправності в системі управління припускає використання електричної схеми останньої. Електросхеми зведені в спеціальні автомобільні комп'ютерні бази по діагностиці й нині досить доступні, треба лише правильно вибрати потрібну. Звичайно, якщо задати саму загальну інформацію з а/м (відзначимо, що бази по електросхемам не оперують VIN-номерама), розвідувач бази знайде кілька різновидів моделі а/м, і буде потрібно додаткова інформація, що може повідомити власник. Наприклад, назву двигуна завжди записано в техпаспорті – букви перед номером двигуна.

Візуальний огляд відіграє роль найпростіших діагностичних затрат. Нерідко досить прості несправності приводять до складних зовнішніх проявів і змушують вважати складною просту проблему. Тому в процесі попереднього огляду повинно перевірятися:

- наявність палива в бензобаку (якщо підозра на систему управління двигуном);
- відсутність затички у вихлопній трубі (якщо підозра на систему управління двигуном);
- чи затягнуті клеми акумуляторної батареї (АКБ) і їхній стан;
- відсутність видимого ушкодження електропроводки;
- чи добре вставлені (повинні бути защелкнуті й не переплутані) роз'єми проведення системи управління;
- попередні чужі дії по подоланню проблеми;
- дійсність ключа запалювання – для автомобіля зі штатним іммобілайзером (якщо підозра на систему управління двигуном);

Іноді буває корисно оглянути місце установки ECU. Не так уже рідко воно виявляється залито водою, наприклад, після мийки двигуна установкою високого

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

тиску. Вода згубна для ECU негерметичного виконання. Помітимо, що роз'єми ECU також бувають як герметичного, так і простого виконання. Роз'єм повинен бути сухим (припустимо застосовувати як водовідштовхувальні кошти, наприклад, WD-40).

Читання кодів несправностей

Якщо для читання кодів несправностей застосовується сканер або комп'ютер з адаптером, важливо, щоб їхнє з'єднання із цифровою шиною ECU було виконано правильно. Ранні ECU не встановлюють зв'язок з діагностикою, поки не приєднані обидві лінії K і L.

Сканування ECU, або активація самодіагностики автомобіля дозволять швидко визначити нескладні проблеми, наприклад, із числа виявлення несправних датчиків. Причому для ECU однаково, несправні сам датчик або його дроти.

Завжди варто пам'ятати, що формування ECU діагностичних повідомлень, зчитувальних сканером, значно формалізоване й підлегле закладеній програмі самодіагностики. Остання видає найбільш імовірну на думку розроблювача, що написав програму, причину несправності. Але очевидно, що найбільш імовірна причина буде реалізовуватися не щораз.

Ще гірше, коли найбільш імовірну причину алгоритм самодіагностики ігнорує (очевидно, у чинність великої віри розроблювача в надійність того або іншого вузла). Так, наприклад, самодіагностика деяких французьких автомобілів при відсутності пуску двигуна через обрив ланцюга датчика положення коленвала не передбачає тестування цього датчика. І тому навіть дилерський прилад DIAG-2000 у цілому ряді випадків при перевірці системи управління двигуном не знаходить придбаний обрив зазначеного ланцюга.

Виконавчі механізми, наприклад, реле, керовані ECU, перевіряються сканером у режимі примусового включення навантажень. Цей режим називається тестом виконавчих механізмів. Тут знов-таки важливо відрізнити дефект у навантаженні від дефекту в її проведенні.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

По-справжньому повинна насторожувати ситуація, коли спостерігається сканування множинних кодів несправностей. При цьому досить велика ймовірність того, що частина з них відноситься до наведених несправностей. Така вказівка на несправність ECU, як «немає зв'язку», – означає, швидше за все, що ECU знеструмлено або відсутній яке-небудь одне його живлення або заземлення.

Якщо немає у своєму розпорядженні сканер або його еквівалент у вигляді комп'ютера з адаптером ліній K і L, більшу частину перевірок можна зробити вручну (див. розділи «Перевірка функцій...»). Звичайно, це буде повільніше, але при послідовному пошуку й обсяг роботи може бути невеликий.

Огляд і перевірка ECU

У тих випадках, коли доступ до ECU простий, а сам блок може бути легко розкритий, варто оглянути його. От що може спостерігатися в несправному ECU:

- обриви, відшарування струмоведучих доріжок, часто з характерними підпалинами;
- спучені або тріснуті електронні компоненти;
- прогари друкованої плати аж до наскрізних;
- вода;
- окисли білий, синьо-зелений або коричневий кольори;

Як уже було сказано, вірогідно перевірити ECU можна шляхом заміни на свідомо справний. Дуже добре, якщо діагност розташовує перевірочним ECU. Однак варто зважати на ризик вивести цей блок з ладу, адже часто першопричина проблеми – несправність зовнішніх ланцюгів. Тому необхідність мати перевірочні ECU не очевидна, а сам прийом варто застосовувати з великою обачністю. На практиці набагато продуктивніше в початковій фазі пошуку вважати ECU справним уже тільки тому, що його огляд не переконує у зворотному. Буває нешкідливо просто переконатися, що ECU на місці.

Перевірка функцій забезпечення

До функцій забезпечення роботи ECU системи управління двигуном відносяться:

- живлення ECU як електронного устрою;
- обмін з керуючим блоком іммобілайзера – якщо є штатний іммобілайзер;
- запуск і синхронізація ECU від датчиків положення коленвала й/або розподільчого вала;
- інформація з інших датчиків.

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Функціональна схема складається із взаємодії наступних функціональних блоків:

- Блок доступу до функцій читання й стирання кодів несправностей (DTC).
- Блок виведення опису кодів несправностей (DTC definitions).
- Блок підтримки автомобілів обладнаних шиною CAN.
- Блок читання й видалення всіх загальних кодів несправностей (DTC).
- Блок читання й видалення кодів несправностей заданих виробником (manufacture specific DTC).
- Блок відображення поточних параметрів OBDII (ISO 14230) (Live Data у момент виникнення несправності).
- Блок виведення стоп-кадрів помилок OBDII.
- Блок виведення результатів внутрішніх тестів системи самодіагностики.
- Блок визначення заводського номера кузова VIN (для автомобілів з 2002 р.в.).
- Блок визначення статусу значка "Check Engine".

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

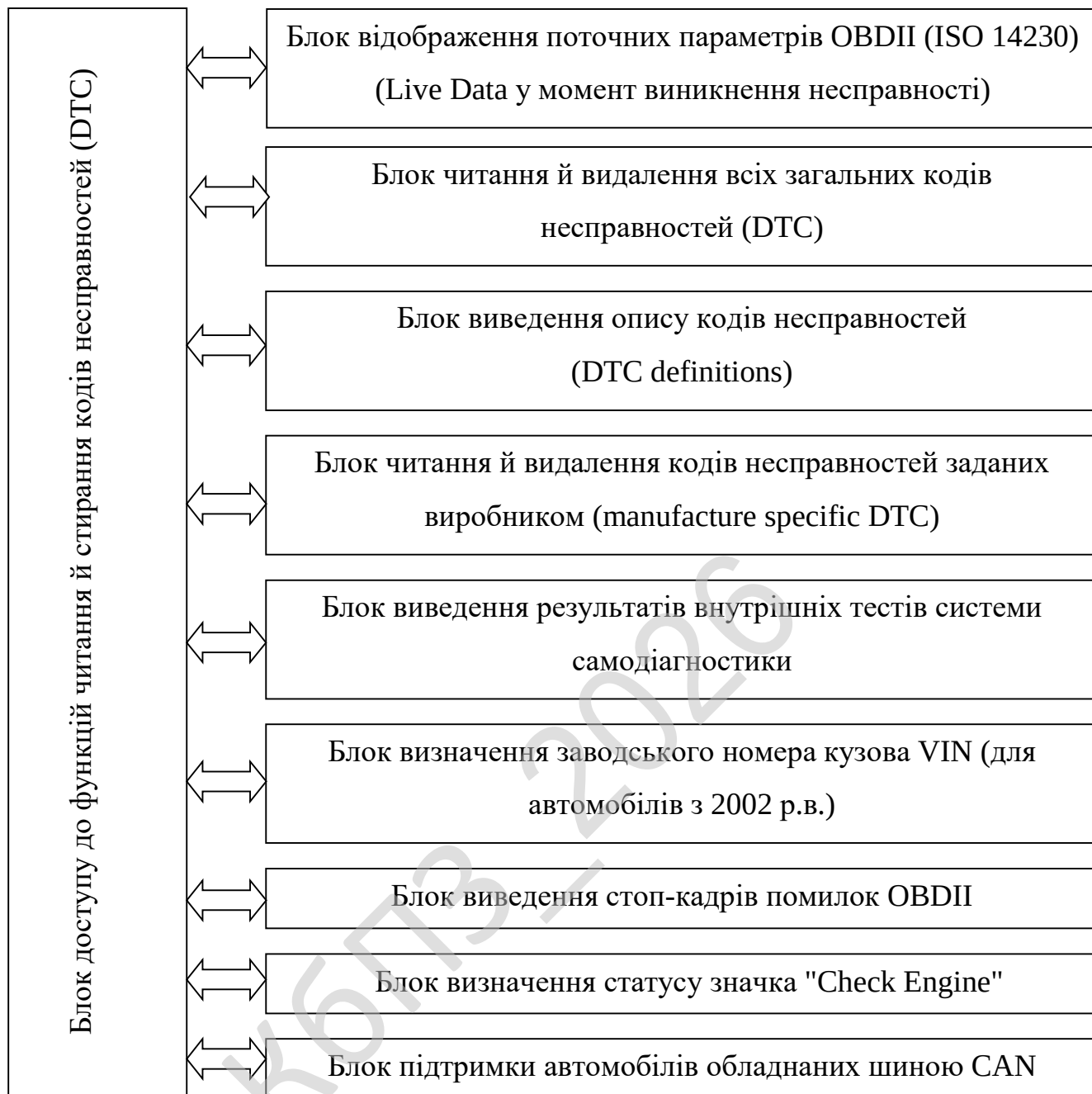


Рисунок 3.2 – Функціональна схема системи

Система функціонує наступним чином. Спершу ЕОМ через кабель з стандартом OBDII (ISO 14230) підключається до бортового комп'ютера автомобіля. Після цього завантажуються стандартні тестові послідовності, які або повертають дані, що системи автомобіля є справними, або повертають коди помилок у роботі того, або іншого блоку автомобіля. Цих кодів помилок дуже

багато, порядку 2000, тому у даній бакалаврській роботі, приведемо перелік тільки основних помилок. До них відносяться наступні нижче перераховані коди несправностей стандарту OBDII.

P0 – 1XX – Вимірники палива й повітря.

P0 – 100 – Несправність ланцюга датчика витрати повітря.

P0 – 101 – Вихід сигналу із припустимого діапазону.

P0 – 102 – Низький рівень вихідного сигналу.

P0 – 103 – Високий рівень вихідного сигналу.

P0 – 105 – Несправність датчика тиску повітря.

P0 – 106 – Вихід сигналу із припустимого діапазону.

P0 – 107 – Низький рівень вихідного сигналу.

P0 – 108 – Високий рівень вихідного сигналу.

P0 – 110 – Несправність датчика температури усмоктуваного повітря.

P0 – 111 – Вихід сигналу із припустимого діапазону.

P0 – 112 – Низький рівень вихідного сигналу.

P0 – 113 – Високий рівень вихідного сигналу.

P0 – 115 – Несправність датчика температури охолодної рідини.

P0 – 116 – Вихід сигналу із припустимого діапазону.

P0 – 117 – Низький рівень вихідного сигналу.

P0 – 118 – Високий рівень вихідного сигналу.

P0 – 120 – Несправність датчика положення дросельної заслінки.

P0 – 121 – Вихід сигналу із припустимого діапазону.

P0 – 122 – Низький рівень вихідного сигналу.

P0 – 123 – Високий рівень вихідного сигналу.

P0 – 125 – Низька температури охолодної рідини для управління по замкнутому контуру..

P0 – 130 – Датчик O2 B1 S1 несправний (Банк1).

P0 – 131 – Датчик O2 B1 S1 має низький рівень сигналу.

P0 – 132 – Датчик O2 B1 S1 має високий рівень сигналу.

РО – 133 – Датчик O2 B1 S1 має повільний відгук на збагачення/збідніння.
 РО – 134 – Ланцюг датчика O2 B1 S1 пасивна.
 РО – 135 – Нагрівач датчика O2 B1 S1 несправний.
 РО – 136 – Датчик O2 B1 S2 несправний.
 РО – 137 – Датчик O2 B1 S2 має низький рівень сигналу.
 РО – 138 – Датчик O2 B1 S2 має високий рівень сигналу.
 РО – 139 – Датчик O2 B1 S2 має повільний відгук на збагачення/збідніння.
 РО – 140 – Ланцюг датчика O2 B1 S2 пасивна.
 РО – 141 – Нагрівач датчика O2 B1 S2 несправний.
 РО – 142 – Датчик O2 B1 S3 несправний.
 РО – 143 – Датчик O2 B1 S3 має низький рівень сигналу.
 РО – 144 – Датчик O2 B1 S3 має високий рівень сигналу.
 РО – 145 – Датчик O2 B1 S3 має повільний відгук на збагачення/збідніння.
 РО – 146 – Ланцюг датчика O2 B1 S3 пасивна.
 РО – 147 – Нагрівач датчика O2 B1 S3 несправний.
 РО – 150 – Датчик O2 Y2 S1 несправний (Банк2).
 РО – 151 – Датчик O2 Y2 S1 має низький рівень сигналу.
 РО – 152 – Датчик O2 Y2 S1 має високий рівень сигналу.
 РО – 153 – Датчик O2 Y2 S1 має повільний відгук на збагачення/збідніння.
 РО – 154 – Ланцюг датчика O2 Y2 S1 пасивна.
 РО – 155 – Нагрівач датчика O2 Y2 S1 несправний.
 РО – 156 – Датчик O2 Y2 S2 несправний.
 РО – 157 – Датчик O2 Y2 S2 має низький рівень сигналу.
 РО – 158 – Датчик O2 Y2 S2 має високий рівень сигналу.
 РО – 159 – Датчик O2 Y2 S2 має повільний відгук на збагачення/збідніння.
 РО – 160 – Ланцюг датчика O2 Y2 S2 пасивна.
 РО – 161 – Нагрівач датчика O2 Y2 S2 несправний.
 РО – 162 – Датчик O2 Y2 S3 несправний.
 РО – 163 – Датчик O2 Y2 S3 має низький рівень сигналу.

- PO – 164 – Датчик O2 Y2 S3 має високий рівень сигналу.
- PO – 165 – Датчик O2 Y2 S3 має повільний відгук на збагачення/збідніння.
- PO – 166 – Ланцюг датчика O2 Y2 S3 пасивна.
- PO – 167 – Нагрівач датчика O2 Y2 S3 несправний.
- PO – 170 – Витік палива з паливної системи блоку №1.
- PO – 171 – Блок циліндрів №1 збіднює (можливо підсмоктування повітря).
- PO – 172 – Блок циліндрів №1 збогачує (можливо неповне закриття форсунки).
- PO – 173 – Витік палива з паливної системи блоку №2.
- PO – 174 – Блок циліндрів №2 збіднює (можливо підсмоктування повітря).
- PO – 175 – Блок циліндрів №2 збогачує (можливо неповне закриття форсунки).
- PO – 176 – Датчик викиду CHx несправний.
- PO – 177 – Сигнал датчика виходить із припустимого діапазону.
- PO – 178 – Низький рівень сигналу датчика CHx.
- PO – 179 – Високий рівень сигналу датчика CHx.
- PO – 180 – Ланцюг датчика температури палива «А» несправний.
- PO – 181 – Сигнал датчика «А» виходить із припустимого діапазону.
- PO – 182 – Низький сигнал датчика температури палива «А».
- PO – 183 – Високий сигнал датчика температури палива «А».
- PO – 185 – Ланцюг датчика температури палива «В» несправний.
- PO – 186 – Сигнал датчика «В» виходить із припустимого діапазону.
- PO – 187 – Низький сигнал датчика температури палива «В».
- PO – 188 – Високий сигнал датчика температури палива «В».
- PO – 190 – Ланцюг датчика тиску палива в паливній рампі несправний.
- PO – 191 – Сигнал датчика виходить із припустимого діапазону.
- PO – 192 – Низький сигнал датчика тиску палива.
- PO – 193 – Високий сигнал датчика тиску палива.
- PO – 194 – Сигнал датчика тиску палива перемежований.

- PO – 195 – Ланцюг датчика температури масла у двигуні несправний.
- PO – 196 – Сигнал датчика виходить із припустимого діапазону.
- PO – 197 – Низький сигнал датчика температури масла.
- PO – 198 – Високий сигнал датчика температури масла.
- PO – 199 – Сигнал датчика температури масла перемежований.
- PO – 2XX –.
- PO – 200 – Ланцюг управління форсункою несправний.
- PO – 201 – Ланцюг управління форсункою циліндра №1 несправний.
- PO – 202 – Ланцюг управління форсункою циліндра №2 несправний.
- PO – 203 – Ланцюг управління форсункою циліндра №3 несправний.
- PO – 204 – Ланцюг управління форсункою циліндра №4 несправний.
- PO – 205 – Ланцюг управління форсункою циліндра №5 несправний.
- PO – 206 – Ланцюг управління форсункою циліндра №6 несправний.
- PO – 207 – Ланцюг управління форсункою циліндра №7 несправний.
- PO – 208 – Ланцюг управління форсункою циліндра №8 несправний.
- PO – 209 – Ланцюг управління форсункою циліндра №9 несправний.
- PO – 210 – Ланцюг управління форсункою циліндра №10 несправний.
- PO – 211 – Ланцюг управління форсункою циліндра №11 несправний.
- PO – 212 – Ланцюг управління форсункою циліндра №12 несправний.
- PO – 213 – Ланцюг управління форсункою холодного старту №1 несправний.
- PO – 214 – Ланцюг управління форсункою холодного старту №2 несправний.
- PO – 215 – Соленоїд вимикання двигуна несправний.
- PO – 216 – Ланцюг контролю часу упорскування несправний.
- PO – 217 – Двигун перебуває в перегрітому стані.
- PO – 218 – Трансмісія перебуває в перегрітому стані.
- PO – 219 – Двигун переключений.
- PO – 220 – Несправність датчика положення дросельної заслінки «В».

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

РО – 221 – Вихід сигналу із припустимого діапазону.

РО – 222 – Низький рівень вихідного сигналу датчика «В».

РО – 223 – Високий рівень вихідного сигналу датчика «В».

РО – 224 – Сигнал датчика «В» перемежований.

РО – 225 – Несправність датчика положення дросельної заслінки «С».

РО – 226 – Вихід сигналу із припустимого діапазону.

РО – 227 – Низький рівень вихідного сигналу датчика «С».

РО – 228 – Високий рівень вихідного сигналу датчика «С».

РО – 229 – Сигнал датчика «С» перемежований.

РО – 230 – Первинний ланцюг бензонасоса (управління реле бензонасосу) несправний.

РО – 231 – Вторинний ланцюг бензонасоса має постійно низький рівень.

РО – 232 – Вторинний ланцюг бензонасоса має постійно високий рівень.

РО – 233 – Вторинний ланцюг бензонасоса має перемежований рівень.

РО – 235 – Ланцюг датчика тиску турбо-наддування «А» несправний.

РО – 236 – Сигнал з датчика турбіни «А» виходить із припустимого діапазону.

РО – 237 – Сигнал з датчика турбіни «А» має постійно низький рівень.

РО – 238 – Сигнал з датчика турбіни «А» має постійно високий рівень.

РО – 239 – Ланцюг датчика тиску турбо-наддування «Б» несправний.

РО – 240 – Сигнал з датчика турбіни «Б» виходить із припустимого діапазону.

РО – 241 – Сигнал з датчика турбіни «Б» має постійно низький рівень.

РО – 242 – Сигнал з датчика турбіни «Б» має постійно високий рівень.

РО – 243 – Соленоїд затвора вихлопних газів турбіни «А» несправний.

РО – 244 – Сигнал соленоїда турбіни «А» виходить із припустимого діапазону.

РО – 245 – Соленоїд вихлопних газів турбіни «А» завжди закритий.

РО – 246 – Соленоїд вихлопних газів турбіни «А» завжди відкритий.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

РО – 247 – Соленоїд вихлопних газів турбіни «В» несправний.

РО – 248 – Сигнал соленоїда турбіни «В» виходить із припустимого діапазону.

РО – 249 – Соленоїд вихлопних газів турбіни «В» завжди закритий.

РО – 250 – Соленоїд вихлопних газів турбіни «В» завжди відкритий.

РО – 251 – Насос упорскування турбіни «А» несправний.

РО – 252 – Сигнал насоса упорскування турбіни «А» виходить із допустимого діапазону.

РО – 253 – Сигнал насоса упорскування турбіни «А» має низький рівень.

РО – 254 – Сигнал насоса упорскування турбіни «А» має високий рівень.

РО – 255 – Сигнал насоса упорскування турбіни «А» перемежований.

РО – 256 – Насос упорскування турбіни «В» несправний.

РО – 257 – Сигнал насоса упорскування турбіни «В» виходить із допустимого діапазону.

РО – 258 – Сигнал насоса упорскування турбіни «В» має низький рівень.

РО – 259 – Сигнал насоса упорскування турбіни «В» має високий рівень.

РО – 260 – Сигнал насоса упорскування турбіни «В» перемежований.

РО – 261 – Форсунка 1-ого циліндра замкнута на землю.

РО – 262 – Форсунка 1-ого циліндра обірвана або замкнута на +12В.

РО – 263 – Драйвер форсунки 1-ого циліндра несправний.

РО – 264 – Форсунка 2-ого циліндри замкнутий на землю.

РО – 265 – Форсунка 2-ого циліндри обірваний або замкнута на +12В.

РО – 266 – Драйвер форсунки 2-ого циліндри несправний.

РО – 267 – Форсунка 3-го циліндри замкнутий на землю.

РО – 268 – Форсунка 3-го циліндри обірваний або замкнута на +12В.

РО – 269 – Драйвер форсунки 3-го циліндри несправний.

РО – 270 – Форсунка 4-ого циліндри замкнутий на землю.

РО – 271 – Форсунка 4-ого циліндри обірваний або замкнута на +12В.

РО – 272 – Драйвер форсунки 4-ого циліндри несправний.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

РО – 273 – Форсунка 5-ого циліндра замкнута на землю.
 РО – 274 – Форсунка 5-ого циліндра обірвана або замкнута на +12В.
 РО – 275 – Драйвер форсунки 5-ого циліндра несправний.
 РО – 276 – Форсунка 6-ого циліндра замкнута на землю.
 РО – 277 – Форсунка 6-ого циліндра обірвана або замкнута на +12В.
 РО – 278 – Драйвер форсунки 6-ого циліндра несправний.
 РО – 279 – Форсунка 7-ого циліндра замкнута на землю.
 РО – 280 – Форсунка 7-ого циліндра обірвана або замкнута на +12В.
 РО – 281 – Драйвер форсунки 7-ого циліндра несправний.
 РО – 282 – Форсунка 8-ого циліндра замкнута на землю.
 РО – 283 – Форсунка 8-ого циліндра обірвана або замкнута на +12В.
 РО – 284 – Драйвер форсунки 8-ого циліндра несправний.
 РО – 285 – Форсунка 9-ого циліндра замкнута на землю.
 РО – 286 – Форсунка 9-ого циліндра обірвана або замкнута на +12В.
 РО – 287 – Драйвер форсунки 9-ого циліндра несправний.
 РО – 288 – Форсунка 10-ого циліндра замкнута на землю.
 РО – 289 – Форсунка 10-ого циліндра обірвана або замкнута на +12В.
 РО – 290 – Драйвер форсунки 10-ого циліндра несправний.
 РО – 291 – Форсунка 11-ого циліндра замкнута на землю.
 РО – 292 – Форсунка 11-ого циліндра обірвана або замкнута на +12В.
 РО – 293 – Драйвер форсунки 11-ого циліндра несправний.
 РО – 294 – Форсунка 12-ого циліндра замкнута на землю.
 РО – 295 – Форсунка 12-ого циліндра обірвана або замкнута на +12В.
 РО – 296 – Драйвер форсунки 12-ого циліндра несправний.
 РО – 3XX – Система запалювання й пропуски.
 РО – 300 – Виявлені випадкові/множинні пропуски запалювання.
 РО – 301 – Виявлені пропуски запалювання в 1-ому циліндрі.
 РО – 302 – Виявлені пропуски запалювання в 2-ому циліндрі.
 РО – 303 – Виявлені пропуски запалювання в 3-ому циліндрі.

- PO – 304 – Виявлені пропуски запалювання в 4-ому циліндрі.
- PO – 305 – Виявлені пропуски запалювання в 5-ому циліндрі.
- PO – 306 – Виявлені пропуски запалювання в 6-ому циліндрі.
- PO – 307 – Виявлені пропуски запалювання в 7-ому циліндрі.
- PO – 308 – Виявлені пропуски запалювання в 8-ому циліндрі.
- PO – 309 – Виявлені пропуски запалювання в 9-ому циліндрі.
- PO – 310 – Виявлені пропуски запалювання в 10-ому циліндрі.
- PO – 311 – Виявлені пропуски запалювання в 11-ому циліндрі.
- PO – 312 – Виявлені пропуски запалювання в 12-ому циліндрі.
- PO – 320 – Ланцюг розподільника запалювання несправний.
- PO – 321 – Сигнал ланцюга розподільника запалювання виходить за допустимі межі.
- PO – 322 – Сигнал ланцюга розподільника запалювання відсутній.
- PO – 323 – Сигнал ланцюга розподільника запалювання перемежований.
- PO – 325 – Ланцюг датчика детонації №1 несправний.
- PO – 326 – Сигнал датчика детонації №1 виходить за припустимі межі.
- PO – 327 – Сигнал датчика детонації №1 має низький рівень.
- PO – 328 – Сигнал датчика детонації №1 має високий рівень.
- PO – 329 – Сигнал датчика детонації №1 перемежований.
- PO – 330 – Ланцюг датчика детонації №2 несправний.
- PO – 331 – Сигнал датчика детонації №2 виходить за припустимі межі.
- PO – 332 – Сигнал датчика детонації №2 має низький рівень.
- PO – 333 – Сигнал датчика детонації №2 має високий рівень.
- PO – 334 – Сигнал датчика детонації №2 перемежований.
- PO – 335 – Датчик положення колінчатого вала «А» несправний.
- PO – 336 – Сигнал датчика «А» виходить за припустимі межі.
- PO – 337 – Сигнал датчика «А» має низький рівень або замкнута на масу.
- PO – 338 – Сигнал датчика «А» має високий рівень або замкнута на 12В.
- PO – 339 – Сигнал датчика «А» перемежований.

PO – 340 – Датчик положення розподільного вала несправний.

PO – 341 – Сигнал датчика виходить за припустимі межі.

PO – 342 – Сигнал датчика має низький рівень або замкнута на масу.

PO – 343 – Сигнал датчика має високий рівень.

PO – 344 – Сигнал датчика перемережований.

PO – 350 – Первинний/вторинний ланцюг котушки запалювання несправні.

PO – 351 – Первинний/вторинний ланцюг котушки запалювання «А» несправні.

PO – 352 – Первинний/вторинний ланцюг котушки запалювання «В» несправні.

PO – 353 – Первинний/вторинний ланцюг котушки запалювання «С» несправні.

PO – 354 – Первинний/вторинний ланцюг котушки запалювання «D» несправні.

PO – 355 – Первинний/вторинний ланцюг котушки запалювання «Е» несправні.

PO – 356 – Первинний/вторинний ланцюг котушки запалювання «F» несправні.

PO – 357 – Первинний/вторинний ланцюг котушки запалювання «G» несправні.

PO – 358 – Первинний/вторинний ланцюг котушки запалювання «H» несправні.

PO – 359 – Первинний/вторинний ланцюг котушки запалювання «I» несправні.

PO – 360 – Первинний/вторинний ланцюг котушки запалювання «J» несправні.

PO – 361 – Первинний/вторинний ланцюг котушки запалювання «K» несправні.

PO – 362 – Первинний/вторинний ланцюг котушки запалювання «L»

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

несправні.

PO – 380 – Свіча накаливання або ланцюг нагрівання несправні.

PO – 381 – Свіча накаливання або індикатор нагрівання несправні.

PO – 385 – Ланцюг датчика положення колінчатого вала «В» несправні.

PO – 386 – Сигнал датчика «В» виходить за припустимі межі.

PO – 387 – Ланцюг датчика обірваний або замкнута на масу.

PO – 388 – Ланцюг датчика замкнутий на один із силових виводів.

PO – 389 – Сигнал датчика «В» перемежований.

PO – 4XX –.

PO – 400 – Система рециркуляції відпрацьованих газів несправний.

PO – 401 – Система рециркуляції відпрацьованих газів неефективна.

PO – 402 – Система рециркуляції відпрацьованих газів надлишкова.

PO – 403 – Ланцюг датчика рециркуляції відпрацьованих газів

несправний.

PO – 404 – Сигнал датчика виходить за припустимі межі.

PO – 405 – Сигнал датчика «А» має низький рівень.

PO – 406 – Сигнал датчика «А» має високий рівень.

PO – 407 – Сигнал датчика «В» має низький рівень.

PO – 408 – Сигнал датчика «В» має високий рівень.

PO – 410 – Система вторинної подачі (упорскування) повітря несправний.

PO – 411 – Помилковий потік проходить через систему вторинної подачі

повітря.

PO – 412 – Клапан системи вторинної подачі повітря «А» несправний.

PO – 413 – Клапан системи вторинної подачі повітря «А» завжди відкритий.

PO – 414 – Клапан системи вторинної подачі повітря «А» завжди закритий.

PO – 415 – Клапан системи вторинної подачі повітря «В» несправний.

PO – 416 – Клапан системи вторинної подачі повітря «В» завжди

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

відкритий.

PO – 417 – Клапан системи вторинної подачі повітря «В» завжди закритий.

PO – 420 – Ефективність системи каталізаторів «В1» нижче поріг.

PO – 421 – Ефективність прогріву каталізатора «В1» нижче поріг.

PO – 422 – Ефективність головного каталізатора «В1» нижче поріг.

PO – 423 – Ефективність нагрівача каталізатора «В1» нижче поріг.

PO – 424 – Температура нагрівача каталізатора «В2» нижче поріг.

PO – 430 – Ефективність системи каталізаторів «В2» нижче поріг.

PO – 431 – Ефективність прогріву каталізатора «В2» нижче поріг.

PO – 432 – Ефективність головного каталізатора «В2» нижче поріг.

PO – 433 – Ефективність нагрівача каталізатора «В2» нижче поріг.

PO – 434 – Температура нагрівача каталізатора «В2» нижче поріг.

PO – 440 – Контроль системи вловлювання пар бензину несправний.

PO – 441 – Система вловлювання пар бензину погано продувається.

PO – 442 – Виявлений невеликий витік у системі вловлювання пар.

PO – 443 – Управління клапаном продувки системи «ЕVAP» несправний.

PO – 444 – Клапан продувки системи «ЕVAP» завжди відкритий.

PO – 445 – Клапан продувки системи «ЕVAP» завжди закритий.

PO – 446 – Управління повітряним клапаном системи «ЕVAP» несправно.

PO – 447 – Повітряний клапан системи «ЕVAP» завжди відкритий.

PO – 448 – Повітряний клапан системи «ЕVAP» завжди закритий.

PO – 450 – Датчик тиску пар бензину несправний.

PO – 451 – Сигнал датчика тиску пар бензину виходить за допустимі діапазон.

PO – 452 – Сигнал датчика тиску пар бензину має низький рівень.

PO – 453 – Сигнал датчика тиску пар бензину має високий рівень.

PO – 454 – Сигнал датчика тиску пар бензину перемижований.

PO – 455 – Виявлена грубий витік у системі вловлювання пар.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

PO – 460 – Ланцюг датчика рівня палива несправний.

PO – 461 – Сигнал датчика рівня палива виходить за припустимі межі.

PO – 462 – Сигнал датчика рівня палива має низький рівень.

PO – 463 – Сигнал датчика рівня палива має високий рівень.

PO – 464 – Сигнал датчика рівня палива перемежований.

PO – 465 – Ланцюг датчика потоку повітря продувки несправний.

PO – 466 – Сигнал датчика потоку повітря продувки виходить за допустимі межі.

PO – 467 – Сигнал датчика потоку повітря продувки має низький рівень.

PO – 468 – Сигнал датчика потоку повітря продувки має високий рівень.

PO – 469 – Сигнал датчика потоку повітря продувки перемежований.

PO – 470 – Датчик тиску вихлопних газів несправний.

PO – 471 – Сигнал датчика тиску виходить за допустимі діапазон.

PO – 472 – Сигнал датчика тиску має низький рівень.

PO – 473 – Сигнал датчика тиску має високий рівень.

PO – 474 – Сигнал датчика тиску перемежований.

PO – 475 – Клапан датчика тиску вихлопних газів несправний.

PO – 476 – Сигнал клапана датчика тиску виходить за допустимі діапазон.

PO – 477 – Сигнал клапана датчика тиску має низький рівень.

PO – 478 – Сигнал клапана датчика тиску має високий рівень.

PO – 479 – Сигнал клапана датчика тиску перемежований.

PO – 500 – Датчик швидкості автомобіля несправний.

PO – 501 – Сигнал датчика швидкості автомобіля виходить за допустимі межі.

PO – 502 – Сигнал датчика швидкості автомобіля має низький рівень.

PO – 503 – Сигнал датчика перемежований або має високий рівень.

PO – 505 – Система підтримки холостого ходу несправний.

PO – 506 – Оберти двигуна під управлінням системи занадто низькі.

PO – 507 – Оберти двигуна під управлінням системи занадто високі.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

PO – 510 – Концевик індикації закритого положення дроселя несправний.
PO – 530 – Датчик тиску холодоагенту кондиціонера несправний.
PO – 531 – Сигнал датчика тиску холодоагенту виходить за допустимі діапазон.

PO – 532 – Сигнал датчика тиску холодоагенту має низький рівень.
PO – 533 – Сигнал датчика тиску холодоагенту має високий рівень.
PO – 534 – Більша втрата холодоагенту в кондиціонері.
PO – 550 – Датчик тиску гідروпідсилювача керма несправний.
PO – 551 – Сигнал датчика тиску виходить за припустимий діапазон.
PO – 552 – Сигнал датчика тиску має низький рівень.
PO – 553 – Сигнал датчика тиску має високий рівень.
PO – 554 – Сигнал датчика тиску перемижований.
PO – 560 – Датчик бортової напруги несправний.
PO – 561 – Бортова напруга нестабільно.
PO – 562 – Бортова напруга має низький рівень.
PO – 563 – Бортова напруга має високий рівень.
PO – 565 – Ланцюг включення «круїз контролю» несправний.
PO – 566 – Ланцюг вимикання «круїз контролю» несправний.
PO – 567 – Ланцюг продовження роботи «круїз контролю» несправний.
PO – 568 – Ланцюг установки швидкості «круїз контролю» несправний.
PO – 569 – Ланцюг підтримки «накату» «круїз контролю» несправний.
PO – 570 – Ланцюг підтримки «розгону» «круїз контролю» несправний.
PO – 571 – Перемикач включення гальм «круїз контролю» несправний.
PO – 572 – Перемикач завжди замкнута.
PO – 573 – Перемикач завжди розімкнута.
PO – 600 – Лінія передачі послідовних даних несправний.
PO – 601 – Помилка контрольної суми внутрішньої пам'яті.
PO – 602 – Програмна помилка контрольного модуля.
PO – 603 – Помилка репрограмуємої пам'яті.

РО – 604 – Помилка оперативного запам'ятовувального пристрою.
 РО – 605 – Помилка постійного запам'ятовувального пристрою.
 РО – 606 – Помилка модуля управління енергозбереженням.
 РО – 700 – Система управління трансмісією несправний.
 РО – 701 – Система управління трансмісією працює невірно.
 РО – 703 – Перемикач карданний вал/гальма несправний.
 РО – 704 – Ланцюг датчика включення зчеплення несправний.
 РО – 705 – Датчик діапазону роботи трансмісії несправний.
 РО – 706 – Сигнал датчика виходить за припустимі межі.
 РО – 707 – Сигнал датчика має низький рівень.
 РО – 708 – має високий рівень.
 РО – 709 – Сигнал датчика перемежований.
 РО – 710 – Датчик температури трансмісійної рідини несправний.
 РО – 711 – Сигнал датчика виходить за припустимі межі.
 РО – 712 – Сигнал датчика має низький рівень.
 РО – 713 – має високий рівень.
 РО – 714 – Сигнал датчика перемежований.
 РО – 715 – Датчик швидкості турбіни несправний.
 РО – 716 – Сигнал датчика виходить за припустимі межі.
 РО – 717 – Сигнал датчика відсутній.
 РО – 718 – Сигнал датчика перемежований.
 РО – 719 – Перемикач карданний вал/гальма замкнуть на масу.
 РО – 720 – Ланцюг датчика «Зовнішньої швидкості» несправний.
 РО – 721 – Сигнал датчика «Зовнішньої швидкості» виходить за допустимі межі.
 РО – 722 – Сигнал датчика «Зовнішньої швидкості» відсутній.
 РО – 723 – Сигнал датчика «Зовнішньої швидкості» перемежований.
 РО – 724 – Перемикач карданний вал/гальма замкнуть на живлення.
 РО – 725 – Ланцюг датчика швидкості обертання двигуна несправний.

РО – 726 – Сигнал датчика виходить за припустимі межі.
 РО – 727 – Сигнал датчика відсутній.
 РО – 728 – Сигнал датчика перемерзований.
 РО – 730 – Передаточне число трансмісії невірне.
 РО – 731 – Передаточне число трансмісії на 1 передачі невірне.
 РО – 732 – Передаточне число трансмісії на 2 передачі невірне.
 РО – 733 – Передаточне число трансмісії на 3 передачі невірне.
 РО – 734 – Передаточне число трансмісії на 4 передачі невірне.
 РО – 735 – Передаточне число трансмісії на 5 передачі невірне.
 РО – 736 – Передаточне число трансмісії на передачі заднього ходу невірне.
 РО – 740 – Ланцюг управління блокуванням диференціала несправний.
 РО – 741 – Диференціал завжди виключений (розблокований).
 РО – 742 – Диференціал завжди включений (заблокований).
 РО – 743 – Зарезервовано.
 РО – 744 – Диференціал стан нестійке.
 РО – 745 – Управління стискаючим соленоїдом несправно.
 РО – 746 – Соленоїд завжди у виключеному стані.
 РО – 747 – Соленоїд завжди у включеному стані.
 РО – 749 – Стан соленоїда хитливий.
 РО – 750 – Соленоїд «А» включення передачі несправний.
 РО – 751 – Соленоїд «А» завжди у виключеному стані.
 РО – 752 – Соленоїд «А» завжди у включеному стані.
 РО – 754 – Стан соленоїда «А» хитливий.
 РО – 755 – Соленоїд «В» включення передачі несправний.
 РО – 756 – Соленоїд «В» завжди у виключеному стані.
 РО – 757 – Соленоїд «В» завжди у включеному стані.
 РО – 759 – Стан соленоїда «В» хитливий.
 РО – 760 – Соленоїд «С» включення передачі несправний.

- PO – 761 – Соленоїд «С» завжди у виключеному стані.
- PO – 762 – Соленоїд «С» завжди у включеному стані.
- PO – 764 – Стан соленоїда «С» хитливий.
- PO – 765 – Соленоїд «Д» включення передачі несправний.
- PO – 766 – Соленоїд «Д» завжди у виключеному стані.
- PO – 767 – Соленоїд «Д» завжди у включеному стані.
- PO – 769 – Стан соленоїда «Д» хитливий.
- PO – 770 – Соленоїд «Е» включення передачі несправний.
- PO – 771 – Соленоїд «Е» завжди у виключеному стані.
- PO – 772 – Соленоїд «Е» завжди у включеному стані.
- PO – 774 – Стан соленоїда «Е» хитливий.
- PO – 780 – Перемикання передач не працює.
- PO – 781 – Перемикання передач із 1-ой на 2-ю не працює.
- PO – 782 – Перемикання передач із 2-й на 3-ю не працює.
- PO – 783 – Перемикання передач із 3-й на 4-ю не працює.
- PO – 784 – Перемикання передач із 4-й на 5-ю не працює.
- PO – 785 – Соленоїд управління синхронізатором несправний.
- PO – 787 – Соленоїд управління синхронізатором завжди виключений.
- PO – 788 – Соленоїд управління синхронізатором завжди включений.
- PO – 789 – Соленоїд управління синхронізатором нестійкий.
- PO – 790 – Ланцюг перемикача режиму руху несправний.
- P1 – 291 – На впуску перегріте повітря.
- P1 – 292 – Тиск газу (бензину)де те в «CN» високе.
- P1 – 293 – Тиск газу (бензину)де те в «CN» низьке.
- P1 – 294 – Холостий хід нестабільний.
- P1 – 295 – На датчику положення дросельної заслінки немає живлення 5В.
- P1 – 296 – На датчику тиску повітря у впускному колекторі немає живлення 5В.
- P1 – 297 – Тиск у датчику мало.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

P1 – 298 – Широко відкритий дросель збіднює.
 P1 – 298 – Виявлена відсутність змін сигналу с.
 P1 – 299 – Потік повітря занадто великий.
 P1 – 390 – Збій за часом синхронізації колінчатого вала.
 P1 – 391 – Провалля сигналу датчика обертання колінчатого вала.
 P1 – 391 – Відсутність сигналу «початок відліку» №1 більше половини часу.
 P1 – 392 – Відсутність сигналу «початок відліку» №2 більше половини часу.
 P1 – 393 – Відсутність сигналу «початок відліку» №3 більше половини часу.
 P1 – 394 – Відсутність сигналу «початок відліку» №4 більше половини часу.
 P1 – 395 – Відсутність сигналу «початок відліку» №5 більше половини часу.
 P1 – 398 – Датчик положення колінчатого вала.
 P1 – 486 – Перетиснений випарний рукав.
 P1 – 487 – Ланцюг високошвидкісного вентилятора №2.
 P1 – 488 – Живлення датчиків 5В відсутнє.
 P1 – 489 – Ланцюг реле високошвидкісного вентилятора.
 P1 – 490 – Ланцюг реле низькошвидкісного вентилятора.
 P1 – 491 – Ланцюг реле радіаторного вентилятора.
 P1 – 492 – Сигнал датчика зовнішньої температури завжди високий.
 P1 – 493 – Сигнал датчика зовнішньої температури завжди низький.
 P1 – 494 – Виявлений витік у ланцюзі перемикача тиску насоса.
 P1 – 495 – Виявлений витік у ланцюзі соленоїда насоса.
 P1 – 496 – Відсутній 5В вихід.
 P1 – 596 – Потужний кроковий перемикач має неправильне початкове положення.

P1 – 598 – Сигнал датчика тиску в кондиціонері завжди низький.

P1 – 599 – Сигнал датчика тиску в кондиціонері завжди високий.

P1 – 698 – Немає кодів повідомлень прийнятих в «trans control mode».

P1 – 699 – Немає кодів повідомлень прийнятих в «powertrain control mode».

P1 – 761 – Керуюча контрольна система.

P1 – 762 – Сигнал датчика тиску GOV зміщений.

P1 – 763 – Сигнал датчика тиску GOV завжди високий.

P1 – 764 – Сигнал датчика тиску GOV завжди низький.

P1 – 765 – Зміна напруги в ланцюзі реле.

P1 – 899 – Перемикач паркування/нейтрал перебуває в помилковому положенні.

P1 – 100 – Сигнал датчика витрати повітря перемежований.

P1 – 101 – Сигнал датчика витрати повітря виходить із допустимого діапазону.

P1 – 112 – Сигнал датчика температури повітря на впуску перемежований.

P1 – 116 – Сигнал датчика температури охолоджуючої рідини виходить із допустимого діапазону.

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів системи, розробленої у результаті виконання бакалаврського проектування, наведена на рисунку 3.3.

Першим процесом який запускається у системі, є процес завантаження вікна діагностики автомобіля.

Він взаємодіє з наступними процесами:

– Процес визначення статусу значка «Check Engine».

– Процес визначення заводського номера кузова.

– Процес запуску виведення внутрішніх тестів системи самодіагностики,

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

який взаємодіє з процесом виведення результатів самодіагностики.

- Процес роботи з кодами несправностей.
- Процес відображення поточних параметрів.



Рисунок 3.3 – Діаграма взаємодії процесів

Процес роботи з кодами несправностей, у свою чергу взаємодіє з наступними процесами: процес читання кодів несправностей, процес читання кодів несправностей заданих виробником, процес виведення опису кодів несправностей, процес виведення стоп-кадрів помилок OBDII.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 наведено блок-схему основної програми. Її робота складається з виконання наступних кроків.

Спершу відбувається виведення основного вікна програми. Після цього користувач обирає, проводити йому діагностику, або ні.

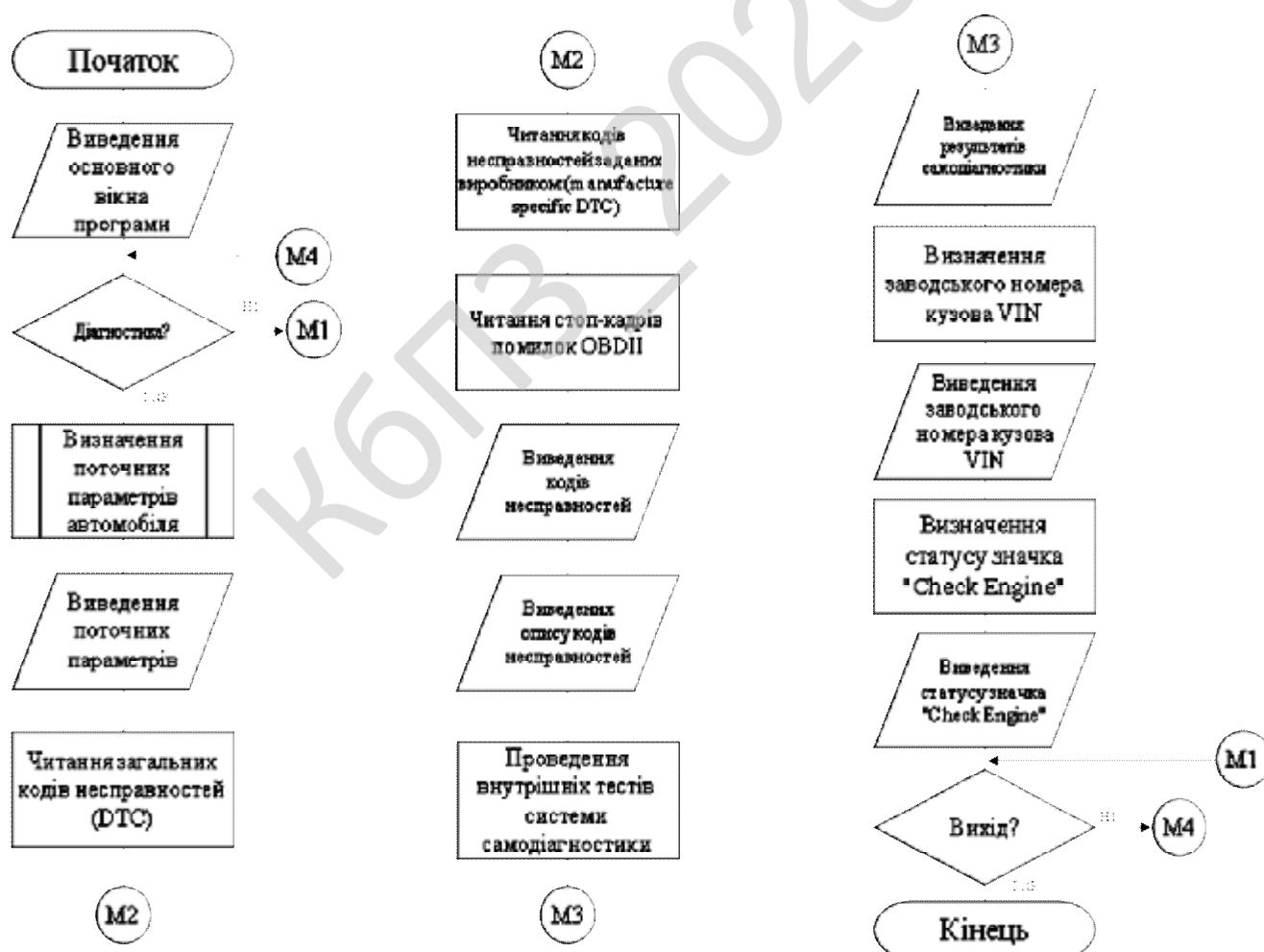


Рисунок 4.1 – Блок-схема роботи основної програми

Якщо він обирає діагностику, то виконується послідовність наступних дій:

- Визначення поточних параметрів автомобіля.
- Виведення поточних параметрів.
- Читання загальних кодів несправностей.
- Читання кодів несправностей заданих виробником.
- Читання стоп-кадрів помилок OBDII
- Виведення кодів несправностей.
- Виведення опису кодів несправностей.
- Проведення внутрішніх тестів системи самодіагностики.
- Виведення результатів самодіагностики.
- Визначення заводського номера кузова.
- Виведення заводського номера кузова.
- Визначення статусу значка «Check Engine».

На рисунку 4.2 зображено блок-схему роботи підпрограми визначення поточних параметрів автомобіля. Вона працює наступним чином.

Спершу відбувається спроба з'єднання з автомобілем. Після цього, якщо сигналу немає, тоді виводиться повідомлення про помилку, й підпрограма завершає свою роботу.

У іншому випадку, тобто, якщо сигнал є, відбувається покрокове виконання наступних ітерацій:

- Отримуються значення температури мотору.
- Отримуються значення оборотів мотора.
- Отримуються значення тиску наддуву.
- Отримуються значення подачі палива.
- Отримуються значення температури палива.
- Отримуються значення температури повітря.
- Отримуються значення атмосферного тиску.
- Отримуються значення пробігу.
- Отримуються значення стану педалі зчеплення.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

- Отримуються значення педалі гальма.
- Отримуються значення педалі газу.
- Отримуються значення температури насосу.
- Отримуються значення затримки ротора.

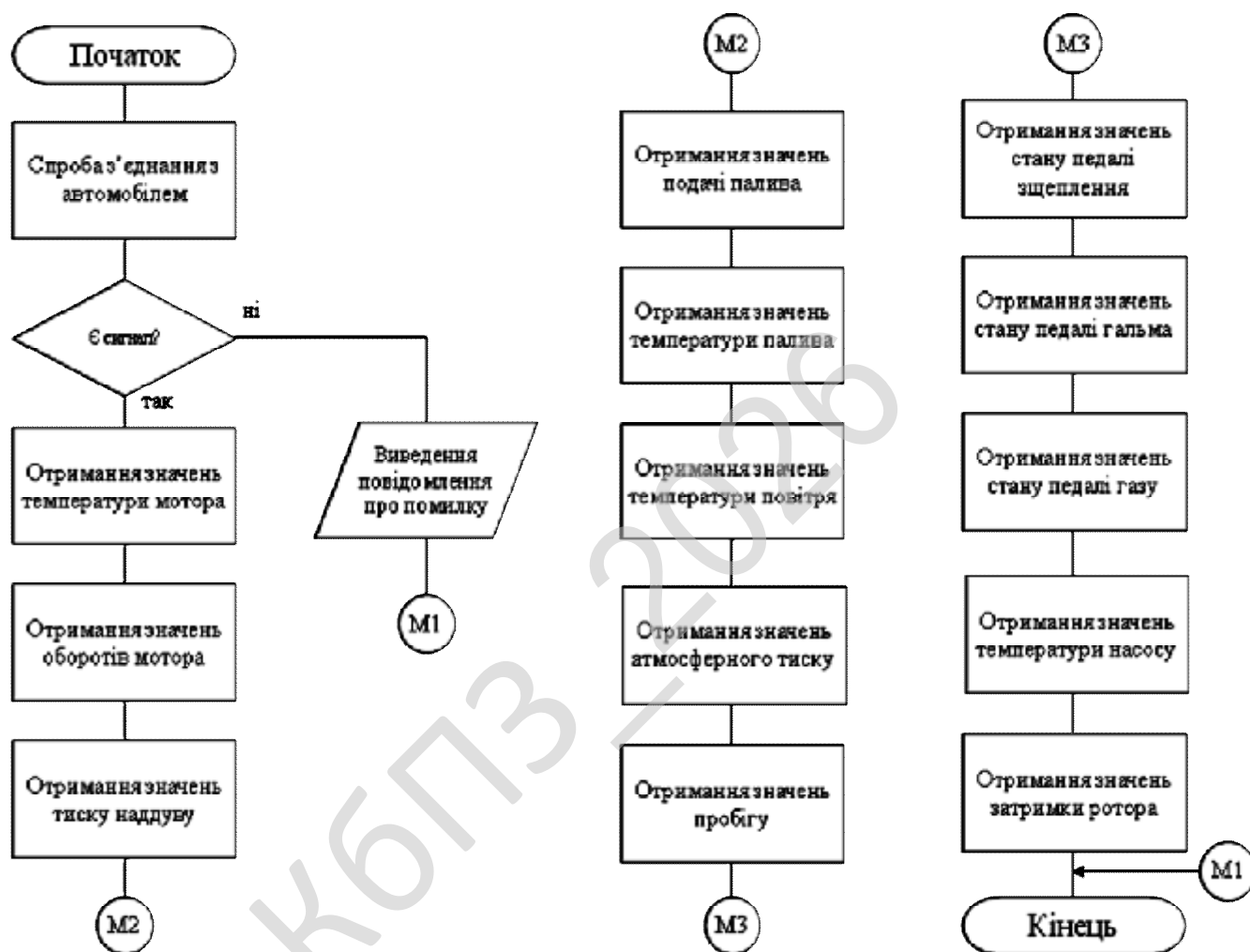


Рисунок 4.2 – Блок-схема роботи підпрограми визначення поточних параметрів автомобіля

Приведемо приклад самодіагностики систем керування на автомобілі OPEL. Іноді, у системі керування упорскуванням автомобілів виникають несправності. Довідатися причину, у більшості випадків, можна самостійно, зчитавши код помилки в накопичувачі ECU через діагностичний роз'єм.

На автомобілі OPEL випуску 86-94 років установлювалися системи діагностики з 10-ти контактним роз'ємом і протоколами діагностики ALDL (Multec) або K-Line (Motronic).

На OPEL випуску 94-96 років установлювалася системи діагностики з 16-ти контактним роз'ємом і протоколом діагностики K-Line. Більше пізні моделі мали систему діагностики, сумісну зі стандартом OBD-II і 16-ти контактним роз'ємом. З 2002 року вводиться CAN протокол на вже стандартному 16-ти контактному OBD-II розніманні.

Варто пам'ятати, що помилка в накопичувачі може мати статус "активної", тобто несправність у системі присутня в цей момент. При цьому буде горіти лампа "check engine" на панелі приладів. І "збереженої" – лампа "check engine" горіти не буде. Якщо ця помилка не виникне повторно протягом 20 запусків двигуна, то вона буде стерта з накопичувача. Так-же накопичувач скидається при знятті клеми акумулятора на 30 секунд. У сучасних системах, помилки можна видалити тільки за допомогою діагностичного встаткування, після усунення їхньої причини.

Розглянемо варіант самодіагностики систем Motronic і Multec з 10 контактним діагностичним роз'ємом.

1. Знаходимо сам роз'єм.

Розташування й призначення його контактів:

A – Маса автомобіля.

B – L-Line Самодіагностика ECU engine.

C – Самодіагностика АКПП.

D – Самодіагностика Trip computer.

E – Канал даних двигуна.

F – "+" Акумуляторної батареї.

G – K-Line канал даних комп'ютерів.

H – Самодіагностика круїзу-контролю.

J – Самодіагностика повного привода.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

К – Самодіагностика ABS.

2. Ставимо дрову перемичку залежно від мети діагностики:

AB – комп'ютер двигуна (Motronic, Multec).

AC – електронна автоматична коробка передач.

AD – бортове табло LCD і бортовий комп'ютер.

AG – стандартна, перешкодозахисна заглушка.

АН – комп'ютер круїз-контролю, пристрою проти викрадення (ATWS).

AJ – електронне керування повним приводом.

AK – комп'ютер ABS.

У сервісних мануалах, ця перемичка називається "діагностичне пристосування КМ 602-2"

3. Включаємо запалювання й зчитуємо flash коди, які нам починає видавати лампа індикації відповідного пристрою. Деякі версії систем можуть не підтримувати самодіагностиків.

Структура показу кодів проста:

Перша серія спалахів з короткими паузами – десятки.

Пауза побільше – роздільник.

Друга серія спалахів з короткими паузами – одиниці.

Довга пауза – роздільник кодів.

x-xx (12).

Кожний цикл починається з коду входу в режим самодіагностики – 12.

Далі видаються коди по зростанню номера помилки. Усе коди повторюються по 3 рази. Після закінчення показу кодів помилок, цикл повторюється заново.

Розшифровка flash кодів помилок наведена в таблицях Flash codes Opel.

Варіант самодіагностики з роз'ємом OBD-II (якщо система його підтримує), мало відрізняється від описаного вище.

1. Знаходимо сам роз'єм.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

В OPEL Vectra, Astra і Zafira він перебуває під захисною кришкою, у салоні під важелем ручного гальма.

В OPEL Omega – у щитку запобіжників салону, під кермовою колонкою.

В OPEL Corsa – у ніші центральної консолі.

2. Ставимо перемичку між контактами 5 і 6.

3. Включаємо запалювання й зчитуємо flash коди, які нам починає видавати лампа індикації. Деякі моделі систем можуть не підтримувати самодіагностики.

Для ранніх систем, принцип зчитування кодів такої-ж. Для більше пізніх, є невеликі відмінності:

10 спалахів лампи відповідає початку коду P0. Далі треба сам 3х значний код.

х-х-xxx (P0113)

Коди повторюються 3 рази і йдуть по зростанню номера помилки. Після закінчення, цикл повторюється заново.

Розшифровка flash кодів помилок стандарту OBD-II наведена в таблицях Flash codes Opel.

Розшифровка flash кодів помилок не OBD-II стандарту наведена в таблицях Flash codes Opel.

Розшифровка flash кодів помилок не OBD-II стандарту для АКПП наведена в таблицях Flash codes Opel, тільки замикати необхідно не 5-6 контакти, а 3-5.

У більш пізніх систем, категорії несправностей розділені, відповідно розділені й лампи індикатори.

Варто пам'ятати, що помилки можуть бути як прямі (точно вказувати на несправність), так і непрямі (вказувати на несправність систем, що перебувають за діапазоном перевірки). Наприклад, помилка лямбда зонда (oxygen sensor), регулятора холостого ходу (IAC) або витратоміра повітря (AFS), може означати так-же й просос неврахованого повітря у впускну систему.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Самі по собі помилки систем не можуть дати повну й точну інформацію про проблеми, більше того – проблеми можуть бути й зовсім без виникнення помилок.

Опис системи

Програмне забезпечення системи мережевих прикінцевих пристроїв автомобіля розробляється як Python застосунок для моніторингу стану автомобільних вузлів, збору телеметрії, виявлення відхилень у роботі пристроїв та формування діагностичного журналу. Система працює з даними від сенсорів, електронних блоків керування, мережевих шлюзів, модулів живлення, температурних датчиків та комунікаційних інтерфейсів автомобіля.

У межах бакалаврської кваліфікаційної роботи система імітує роботу автомобільної мережі та обробляє параметри прикінцевих пристроїв. Python використовується для збору даних, перевірки допустимих значень, збереження подій у SQLite та виведення короткого звіту про технічний стан вузлів.

Основні функції системи наступні:

- Реєстрація прикінцевих пристроїв автомобіля.
- Збір параметрів температури, напруги, навантаження та стану зв'язку.
- Перевірка даних за допустимими межами.
- Виявлення критичних або нестабільних вузлів.
- Збереження діагностичних подій у базі даних.
- Формування звіту про поточний стан автомобільної мережі.

Створені наступні функціональні модулі:

- Модуль пристроїв автомобіля.
- Модуль збору телеметрії.
- Модуль діагностики параметрів.
- Модуль журналювання подій.
- Модуль формування звітів.
- Модуль збереження даних SQLite.

Архітектура системи

Архітектура має модульну структуру, рівень збору даних отримує параметри від прикінцевих пристроїв. Рівень діагностики перевіряє отримані значення та визначає стан вузлів. Рівень збереження фіксує події в базі SQLite. Рівень звітності формує підсумкову інформацію для користувача. Такий підхід спрощує розширення системи новими датчиками, мережевими модулями та діагностичними правилами.

Розглянемо це на частині вихідного коду.

```
import sqlite3
from dataclasses import dataclass
from datetime import datetime

@dataclass
class VehicleEndpoint:
    device_id: str
    name: str
    temperature: float
    voltage: float
    load: float
    online: bool

class VehicleNetworkDatabase:
    def __init__(self, db_name="vehicle_network.db"):
        self.db_name = db_name
        self.create_table()
    def create_table(self):
        connection = sqlite3.connect(self.db_name)
        cursor = connection.cursor()
        #Створення таблиці для збереження діагностичних подій
        cursor.execute("""
            CREATE TABLE IF NOT EXISTS diagnostic_events (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                device_id TEXT NOT NULL,
                device_name TEXT NOT NULL,
                status TEXT NOT NULL,
                message TEXT NOT NULL,
                event_time TEXT NOT NULL
            )
        """)
```

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

```

        connection.commit()
        connection.close()

    def save_event(self, device_id, device_name, status, message):
        connection = sqlite3.connect(self.db_name)
        cursor = connection.cursor()

        #Збереження результату перевірки пристрою в базу даних
        cursor.execute("""
            INSERT INTO diagnostic_events
            (device_id, device_name, status, message, event_time)
            VALUES (?, ?, ?, ?, ?)
        """, (device_id, device_name, status, message,
datetime.now().strftime("%Y-%m-%d %H:%M:%S")))
        connection.commit()
        connection.close()

class VehicleEndpointMonitor:
    def __init__(self, database):
        self.database = database

    def diagnose_device(self, device):
        problems = []
        if not device.online:
            problems.append("пристрій не відповідає в мережі")
        if device.temperature > 85:
            problems.append("перевищено допустиму температуру")
        if device.voltage < 10.5 or device.voltage > 14.8:
            problems.append("напруга знаходиться поза допустимими межами")
        if device.load > 90:
            problems.append("зафіксовано надмірне навантаження")
        if problems:
            status = "ПОМИЛКА"
            message = ", ".join(problems)
        else:
            status = "НОРМА"
            message = "пристрій працює стабільно"

        #Фіксація діагностичного висновку для кожного прикінцевого пристрою
        self.database.save_event(device.device_id, device.name, status, message)
        return status, message

def main():
    database = VehicleNetworkDatabase()
    monitor = VehicleEndpointMonitor(database)

```

```

    devices = [
VehicleEndpoint("ECU001",
    "Блок керування двигуном", 78.5, 12.6, 64.0, True),
VehicleEndpoint("GW002",
    "Мережевий шлюз автомобіля", 91.2, 12.1, 88.0, True),
VehicleEndpoint("SNS003",
    "Температурний датчик салону", 36.4, 9.8, 25.0, True),
VehicleEndpoint("CAM004",
    "Модуль камери заднього огляду", 44.1, 12.4, 93.5, False)
    ]
    for device in devices:
        status, message = monitor.diagnose_device(device)
        print(f"{device.name} | {status} | {message}")

```

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SHACAL-2, який шифрує дані 256-бітними блоками з використанням 512-бітного ключа. Допускається використання ключів менших розмірів (не менш 128 біт), які доповнюються бітовими нулями до 512 біт.

Шифруємий блок даних ділиться на 8 фрагментів по 32 біта (які позначені буквами $A...H$). Алгоритм виконує 64 раунду перетворень, у кожному з яких дані фрагменти обробляються в такий спосіб:

$$T = H_i + S_1(E_i) + Ch(E_i, F_i, G_i) + M_i + K_i,$$

$$H_{i+1} = G_i,$$

$$G_{i+1} = F_i,$$

$$F_{i+1} = E_i,$$

$$E_{i+1} = D_i + T,$$

$$D_{i+1} = C_i,$$

$$C_{i+1} = B_i,$$

$$B_{i+1} = A_i,$$

$$A_{i+1} = T + S_0(A_i) + Maj(A_i, B_i, C_i).$$

де T – тимчасова змінна.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Використовувані функції визначені в такий спосіб:

$$S_0(x) = (x \ggg 2) \oplus (x \ggg 13) \oplus (x \ggg 22),$$

$$S_1(x) = (x \ggg 6) \oplus (x \ggg 11) \oplus (x \ggg 25),$$

$$Ch(x, y, z) = (x \& y) \oplus (x' \& z),$$

$$Maj(x, y, z) = (x \& y) \oplus (x \& z) \oplus (y \& z),$$

де $\ggg$ – операція побітового циклічного зрушення вправо.

Константи, що модифікують, M_i ($i = 0 \dots 63$) наведено нижче (одна за одною від M_0 до M_{63}):

428A2F98	71374491	B5C0FBCF	E9B5DBA5
3956C25B	59F111F1	923F82A4	AB1C5ED5
D807AA98	12835B01	243185BE	550C7DC3
72BE5D74	80DEB1FE	9BDC06A7	C19BF174
E49B69C1	EFBE4786	0FC19DC6	240CA1CC
2DE92C6F	4A7484AA	5CB0A9DC	76F988DA
983E5152	A831C66D	B00327C8	BF597FC7
C6E00BF3	D5A79147	06CA6351	14292967
27B70A85	2E1B2138	4D2C6DFC	53380D13
650A7354	766A0ABB	81C2C92E	92722C85
A2BFE8A1	A81A664B	C24B8B70	C76C51A3
D192E819	D6990624	F40E3585	106AA070
19A4C116	1E376C08	2748774C	34B0BCB5
391C0CB3	4ED8AA4A	5B9CCA4F	682E6FF3
748F82EE	78A5636F	84C87814	8CC70208
90BEFFFA	A4506CEB	BEF9A3F7	C67178F2

Фрагменти розширеного ключа $K_0 \dots K_{63}$ обчислюються в процесі процедури розширення ключа, що виконується в такий спосіб:

Етап 1. 512-бітний вихідний ключ шифрування ділиться на 16 фрагментів по 32 біта $K_0 \dots K_{15} \dots$

Етап 2. Інші фрагменти розширеного ключа $K_{16} \dots K_{63}$ обчислюються з перших 16 фрагментів у такий спосіб:

$$K_i = O_1(K_{i-2}) + K_{i-7} + O_0(K_{i-15}) + K_{i-16},$$

де функції O_0 і O_1 визначені так:

$$O_0(x) = (x \ggg 7) \sqcup (x \ggg 18) \sqcup (x \gg 3),$$

$$O_1(x) = (x \ggg 17) \sqcup (x \ggg 19) \sqcup (x \gg 10),$$

де $\gg$ – операція побітового зрушення (не циклічного) вправо.

Раунди розшифрування алгоритму виконуються у зворотній послідовності:

$$T = A_{i+1} + S_0'(B_{i+1}) + Maj'(B_{i+1}, C_{i+1}, D_{i+1}) + 2,$$

$$H_i = T + S_1'(F_{i+1}) + Ch'(F_{i+1}, G_{i+1}, H_{i+1}) + M_i' + K_i' + 4,$$

$$G_i = H_{i+1},$$

$$F_i = G_{i+1},$$

$$E_i = F_{i+1},$$

$$D_i = E_{i+1} + T' + 1,$$

$$C_i = D_{i+1},$$

$$B_i = C_{i+1},$$

$$A_i = B_{i+1}.$$

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Для установки утиліти запустіть установочний файл відповідно на КПК, CarPC, персональному комп'ютері або ноутбучі й дотримуйтеся інструкцій програми установки. Після установки утиліти натисніть «Налаштування». Уведіть серійний номер адаптера й номер COM порту. У Рядку «Автомобіль» виберіть модель вашого автомобіля. У рядку «Система» виберіть систему (АКПП, ABS і ін.) яку ви збираєтеся діагностувати. Якщо для обраного автомобіля доступна робота тільки з однією системою її назва відобразиться в рядку «Система» сірим кольором. Після натискання «ОК» утиліта настроїться на діагностику обраної системи.

Робота з утилітою

1. Підключити адаптер до автомобіля.
2. Запустити утиліту й виберіть модель автомобіля й діагностуєму систему.
3. Включити запалювання автомобіля не заводячи двигун.
4. Натисніть клавішу «Підключитися». Утиліта встановити зв'язок з адаптером і з електронним блоком управління (ЕБУ) діагностуємої системи автомобіля. Перевірити з'єднання можна нажавши «Інформація», на екрані відобразяться відомості про використовуваний адаптер, бортова напруга автомобіля й ідентифікаційні (паспортні) дані ЕБУ діагностуємої системи*
5. Натисніть «Коди несправностей» на екрані відобразяться коди несправностей діагностуємої системи з розшифровкою українською мовою або повідомлення «Коди несправностей не виявлені».
6. Усунете несправність і натисніть «Скидання кодів». По цій команді коди несправностей зітруться з пам'яті ЕБУ діагностуємої системи.
7. Натисніть клавішу «Відключитися».

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

Якщо несправність не буде усунута, то ЕБУ знову зафіксує відповідний код у своїй пам'яті.

Скріншоти програмного забезпечення зображені на рисунках 5.1 та 5.2

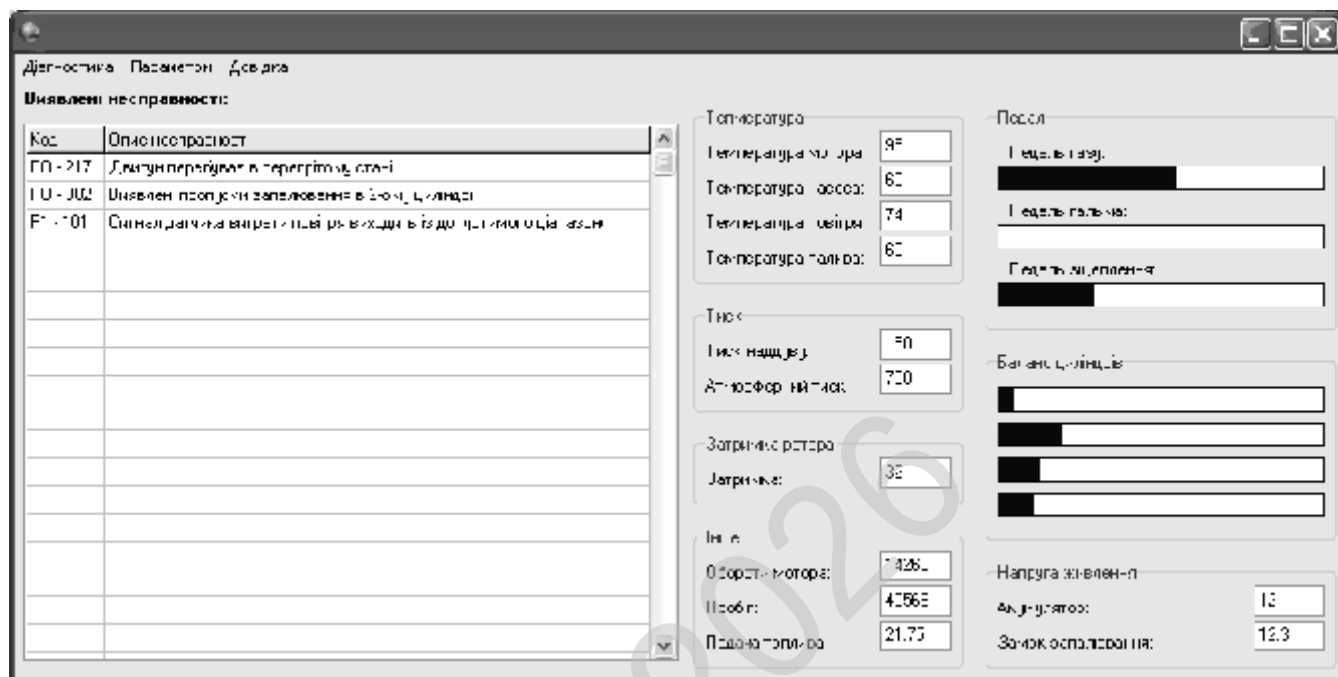


Рисунок 5.1 – Основне вікно програми

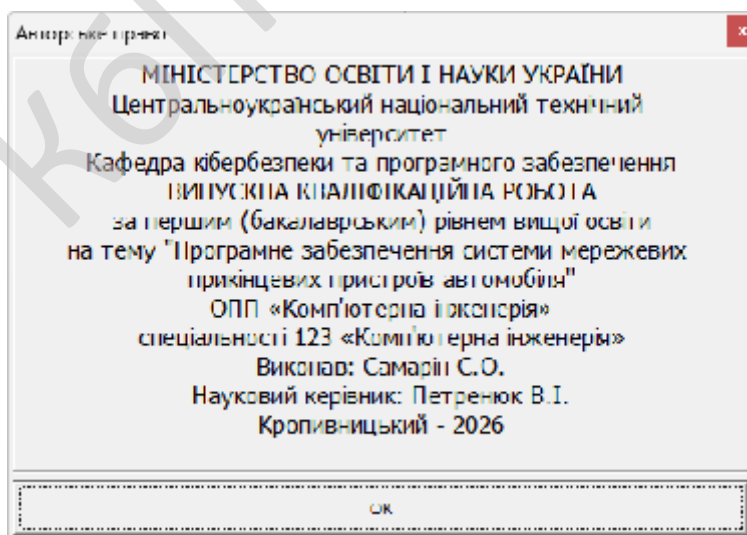


Рисунок 5.2 – Вікно довідки

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи мережевих прикінцевих пристроїв автомобіля.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем мережевих прикінцевих пристроїв автомобіля.
- Досліджена система мережевих прикінцевих пристроїв автомобіля.
- На основі отриманих результатів досліджень створена програмна реалізація системи мережевих прикінцевих пристроїв автомобіля.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання мережевих прикінцевих пристроїв автомобіля.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи мережевих прикінцевих пристроїв автомобіля. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Shacal-2.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

К6ПЗ-2026

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Peter Hoddie, Lizzie Prader «IoT Development for ESP32 and ESP8266 with JavaScript: A Practical Guide to XS and the Moddable SDK» ISBN-13 (pbk): 978-1-4842-5069-3 ISBN-13 (electronic): 978-1-4842-5070-9
2. STM32CubeMX for STM32 configuration and initialization C code generation. User manual. June 2022. 397 p.
3. І.В.Чихіра, А.Г. Микитишин Конспект лекцій з дисципліни «Програмування систем реального часу» / Укладачі : Чихіра І.В., Микитишин А.Г., – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2016. – 76 с.
4. Jack Ganssle and Michael Barr. 2003. Embedded Systems Dictionary. CMP Books.
5. Технології інтернету речей. Навчальний посібник [Електронний ресурс]: / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 12,5 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 271 с.
6. Greg Dunko, Joydeep Misra, Josh Robertson, Tom Snyder “A reference guide to the Internet of Things” / 2017 Bridgera LLC, RIoT.
7. Donald Norris “Programming with STM32. Getting started with the Nucleo Board and C/C++” 416 p. 2018.
8. Neil Kolban “Kolban’s book on ESP32”. Texas, USA. 951 p.
9. Вінтенко Б.Ю., Миронець І.В., Смірнов О.А. «Модель комп’ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». *IV Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи розвитку інформаційних технологій (ІПШРІТ-2025)»* м.Черкаси 25 листопада 2025 року – Черкаси: ЧДТУ.– 2025. – С.101-103.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

10. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.

11. Вінтенко, Б.Ю., Миронець, І.В., Смірнов, О.А., Коваленко, О.В., Усік, П.С., Буравченко, К.О., Лисенко, І.А. «Логіко-структурна модель комп'ютерно-орієнтованої процедури системи підтримки оперативного персоналу АЕС». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С. 413-427, 2025.

12. Смірнова, Т.В. «Дослідження методів, моделей та сучасних ІТ-рішень для підтримки технологічних процесів у критичній інфраструктурі держави». *Кібербезпека: освіта, наука, техніка*. 2025. Том 2 № 30. С.195-208, 2025.

13. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.

14. Kuznetsov, O., Atzeni, G., Arnesano, M., Randieri, C., Smirnov, O. «Secure IoT-based smart wheelchair system: From implementation to security enhancement strategy». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 225–257.

15. Kuznetsov, O., Smirnov, O., Kuznetsova, T., Shaikhanova, A., Svatowsky, I. «Privacy-utility trade-offs in IoT networks: A comparative analysis of differential privacy mechanisms for sensor data aggregation». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 589–622.

16. Вінтенко Б., Смірнов О., Миронець І., Смірнова Т., Смірнов С. «Імітаційна модель шляхів вхідних даних комп'ютерної інтелектуальної системи підтримки оператора енергоблоку АЕС». *Комбінаторні конфігурації та їхні застосування: Матеріали XXVII Міжнародного науково-практичного семінару, присвяченого 125-річчю Національного університету «Запорізька політехніка»*

(Запоріжжя-Кропивницький-Київ, 4-6 червня 2025 р.). Запоріжжя: НУ «Запорізька політехніка», 2025. С.82-91.

17. Al-Azzeh, J., Ayyoub, B., Mesleh, A., Smirnova, T., Gnatyuk, S., Drieiev, O., Smirnov, O., Dorenskyi, O. «Cloud-Based Information System for Evaluating Caverns in the Process of Blasting Metal Surfaces of Details». *International Review on Modelling and Simulations* 18 (1), 2025. pp. 32-42.

18. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В., Коваленко О.В., Мацуї А.М. «Модель шляхів отримання вхідних даних комп’ютерної інтелектуальної системи підтримки оперативного персоналу АЕС». *Центральноукраїнський науковий вісник. Технічні науки*. 2025. Вип. 11(42), ч. II. С.52-62.

19. Вінтенко Б.Ю., Смірнов О.А., Миронець І.В., Смірнова Т.В. «Методи забезпечення відмовостійкості інтелектуальних систем підтримки оператора». *VIII міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології”*, м. Кропивницький. 24-25 квітня 2025 р. – Кропивницький: ЦНТУ. – 2025. – С. 44-46.

20. Вінтенко, Б., Миронець, І., Смірнов, О., Коваленко, А., Коноплицька-Слободенюк, О., Смірнова, Т., Константинова, Л. «Дослідження застосування систем підтримки оперативного персоналу об’єкту критичної інфраструктури при керуванні енергоблоком АЕС з реактором типу ВВЕР-1000». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2024. № 2(26), С. 6-26.

21. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

22. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки

програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

23. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

24. Kuznetsov, O., Kuznetsova, Y., Smirnov, O., Kostenko, O., Zvieriev, V. «Evaluating Hashing Algorithms in the Age of ASIC Resistance». *CEUR Workshop Proceedings*, 2023, 3628, pp. 93-105.

25. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

26. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

27. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

28. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

29. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings* Volume 3156, 2022, Pages 390-399.

30. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

31. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

32. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

33. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

34. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

35. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

36. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and

cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

37. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

38. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

39. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

40. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

41. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

42. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

43. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019,

44. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.517-522.

45. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

46. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

47. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT-2019/* Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

48. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

49. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 347-352.

50. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special

Correlation Properties», *CEUR Workshop Proceedings* Volume 2353, *CEUR Workshop Proceedings* 2019, Pages 618-629.

51. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

52. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

53. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки*. №4. С. 103-110. 2020.

54. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.

					ВКРБ-123.26.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92