

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
« ____ » _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему

**“Програмне забезпечення системи діагностування електронної
системи автомобіля за стандартом ISO 14230”**

Виконав здобувач вищої освіти
IV курсу, групи КМ-19
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Марченко М.С.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Дреєв О.М.
« ____ » _____ 2023 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
« 17 » січня 2023 року

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

Марченку Максиму Сергійовичу

(прізвище, ім'я, по батькові)

- Тема роботи *Програмне забезпечення системи діагностування електронної системи автомобіля за стандартом ISO 14230*
- Керівник роботи *Дреєв Олександр Миколайович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 10-02 від 5.01.2023 року
- Строк подання студентом роботи до захисту *23.05.2023 р.*
- Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи діагностування електронної системи автомобіля за стандартом ISO 14230*
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Призначення та область використання.*
 - Перегляд аналогічних існуючих систем.*
 - Опис і обґрунтування проектних рішень.*
 - Етапи програмування системи.*
 - Впровадження системи в промислову експлуатацію.*
 - Висновки*
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

<i>Структурна схема системи</i>	<i>1 аркуш</i>
<i>Функціональна схема системи</i>	<i>1 аркуш</i>
<i>Діаграма процесів</i>	<i>1 аркуш</i>
<i>Блок-схема алгоритму роботи додатку</i>	<i>2 аркуша</i>

7. Дата видачі завдання « 17 » січня 2023 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	10.03.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2023 р.	
3.	Розробка моделі компонента	20.03.2023 р.	
4.	Розробка структур даних	25.03.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2023 р.	
6.	Програмування алгоритмів	10.04.2023 р.	
7.	Оформлення ПЗ	17.04.2023 р.	
8.	Попередній захист роботи	23.05.2023 р.	

Дата видачі завдання
« 17 » січня 2023 р.

Підпис керівника

Дреєв О.М.
(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2023 р.

Підпис здобувача

Марченко М.С.
(прізвище та ініціали)

АНОТАЦІЯ

Марченко М.С. Програмне забезпечення системи діагностування електронної системи автомобіля за стандартом ISO 14230. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи діагностування електронної системи автомобіля за стандартом ISO 14230.

Метою розробки є програмне забезпечення системи діагностування електронної системи автомобіля за стандартом ISO 14230.

Результат роботи – програмна реалізація системи діагностування електронної системи автомобіля за стандартом ISO 14230.

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 архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі RAD Studio Delphi 10.4.

Ключові слова: комп'ютерна інженерія, електронна система автомобіля, ISO 14230

ABSTRACT

Marchenko M.S. Software of the vehicle electronic system diagnostic system according to the ISO 14230 standard. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this final qualification work for the first (bachelor) level of higher education, software was developed, which is intended for the system of diagnosing the electronic system of the car according to the ISO 14230 standard.

The purpose of the development is the software of the system for diagnosing the electronic system of the car according to the ISO 14230 standard.

The result of the work is the software implementation of the system for diagnosing the car's electronic system according to the ISO 14230 standard.

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 tools are provided.

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the RAD Studio Delphi 10.4 environment.

Keywords: computer engineering, car electronic system, ISO 14230

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	4
1.1 Призначення системи.....	4
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	7
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	7
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	18
2.3 Розгорнута постановка завдання	24
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	26
3.1 Опис функціонування системи	26
3.2 Розробка структурної схеми.....	44
3.3 Розробка функціональної схеми	48
3.4 Розробка діаграми процесів.....	50
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	52
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	52
4.2 Захист розробленого програмного забезпечення.....	65
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	68
6 ОСНОВНІ ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	72

						ВКРБ-123.23.0015.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Марченко М.С.				Програмне забезпечення системи діагностування електронної системи автомобіля за стандартом ISO 14230	Літ.	Аркуш	Аркушів
Перев.	Дресв О.М.					Б	1	78
Н.контр.	Гермак В.С.				ЦНТУ КМ-19			
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

Кафедра _ КБПЗ _ 2023 рік

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Електронний блок керування двигуном (ECU) є центральним контролером і серцем системи керування двигуном. Він контролює подачу палива, керування повітрям, упорскування палива та запалювання. Завдяки масштабованості своєї продуктивності блок керування також може керувати вихлопною системою, а також інтегрувати функції трансмісії та автомобіля. ECU керує всіма типами трансмісії та топологіями, такими як бензин, дизель, CNG, етанол, а також гібридна система та система паливних елементів.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи діагностування електронної системи автомобіля за стандартом ISO 14230.

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

- Огляд існуючих систем діагностування електронної системи автомобіля за стандартом ISO 14230.
- Дослідження системи діагностування електронної системи автомобіля за стандартом ISO 14230.
- Програмна реалізація системи діагностування електронної системи автомобіля за стандартом ISO 14230.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі діагностування електронної системи автомобіля за стандартом ISO 14230.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи діагностування електронної системи автомобіля за стандартом ISO 14230, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

По набору функцій ECU подібні один одному настільки, наскільки подібні відповідні системи управління. Фактичні відмінності можуть бути великі, але питання електроживлення, взаємодії з реле й іншими соленоїдними навантаженнями ідентичні для самих різних ECU. Тому найважливіші дії первинної діагностики різних систем виявляються однаковими. А загальна логіка, що викладається далі, діагностики застосовна до будь-яких автомобільних систем управління.

Розмаїтість систем управління зобов'язано своєю появою на світло частоті модернізації автомобільних агрегатів їхніми виробниками. Так, наприклад, кожний двигун виробляється протягом ряду років, але його система управління модифікується майже щорічно, і вихідна згодом може бути повністю замінена на зовсім іншу. Відповідно, у різні роки той самий двигун може комплектуватися залежно від складу системи управління різними, схожими або не схожими один на одного блоками управління. Нехай механіка такого двигуна добре відома, але часто виявляється, що саме видозмінена система управління приводить до утруднень у діагностиці. Здавалося б, у такій ситуації важливо визначити: а чи справний новий, не знайомий ECU.

Насправді набагато важливіше перебороти спокусу замислюватися на цю тему. Занадто просто засумніватися в справності екземпляра ECU, адже властиво про нього, навіть як про представника відомої системи управління, звичайно мало що відомо. З іншого боку, існують нескладні прийоми діагностики, застосовні в силу своєї простоти однаково успішно до всіляких систем управління. Така універсальність пояснюється тим, що зазначені прийоми опираються на споріднення систем і тестують їхні загальні функції.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Дана перевірка інструментально доступна будь-якому гаражу, і ігнорувати її, посилаючись на застосування сканера, невиправдано. Навпаки, оправданий повторний огляд результатів сканування ECU. Адже те, що читання помилок з ECU досить полегшує діагностику, – розповсюджена омана. Точніше було б сказати, що – так, полегшує пошук одних, але ніяк не допомагає у виявленні інших і утрудняє пошук третіх несправностей. Насправді діагност здатний виявити за допомогою сканера 40...60 % несправностей, тобто цей прилад якимось відслідковує, приблизно, їхню половину. Відповідно близько 50% неполадок сканер або не відслідковує зовсім, або вказує на неіснуючі. Так відбувається, коли несправність не відноситься до електрику, або не укладається в передбачений розроблювачем алгоритм самодіагностики, по якому ECU формує коди помилок, зчитувальні сканером. На жаль, доводиться констатувати, що одного цього буває досить, щоб помилково забракувати ECU.

Близько 7...10% із поступивших на діагностику ECU виявляються справними, і більшість таких випадків – результат висновку про вихід ECU з ладу. Ще порівняно недавно цей відсоток був удвічі-утроє вище. Не буде великим перебільшенням сказати, що за кожним абзацом далі стоїть випадок розгляду з тим або іншим а/м після встановлення справності його ECU, що спочатку був зданий у ремонт як приблизно дефектний.

On-Board Diagnostic (OBD) – самодіагностика бортового встаткування автомобіля, термін, що позначає стандарт діагностики й контролю двигуна автомобіля, також частин шасі, і допоміжних пристроїв. Стандарт регламентує сигнали й распіновку роз'єму діагностики. OBD-II був розроблений Society of Automotive Engineers (SAE) США й затверджений Environmental Protection Agency (EPA) в 1996. (Дещо раніше був впроваджений стандарт OBD-I, але широкого поширення він не одержав). До моменту створення OBD-II існувало три основних протоколи обміну даними між бортовим електронним устаткуванням автомобіля й різних діагностичних сканерів. Властиво всі три ввійшли в OBD-II. Всі Європейські й більшість Азіатських виробників

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

використовували ISO 9141 стандарт (K, L – лінія). General Motors використовував SAE J1850 VPW (Variable Pulse Width Modulation), а Fords – SAE J1850 PWM (Pulse Width Modulation). Ненабагато пізніше з'явився ISO 14230 (удосконалений варіант ISO 9141, відомий як KWP2000).

Європейцями в 2001 був прийнятий EOBD (enhanced) розширений OBD стандарт. Основна перевага – наявність високошвидкісної CAN (Controller Area Network) шини. Назва CAN шини прийшла з комп'ютерної термінології, тому що створювався даний стандарт приблизно в 80-х компаніями BOSCH і INTEL, як комп'ютерний мережний інтерфейс бортових мультипроцесорних систем реального часу. CAN-шина – це двопроводна, послідовна, асинхронна шина з рівноправними вузлами й придушенням синфазних перешкод. CAN характеризується високою швидкістю передачі (набагато більшою, ніж інші протоколи) і високою завадостійкістю. Для порівняння ISO 9141, ISO 14230, SAE J1850 VPW забезпечують швидкість передачі даних 10.4 Kbps, SAE J1850 PWM – 41.6 Kbps, ISO 15765 (CAN) – 250/500 kbit/s.

1.2 Область застосування

Областю застосування розроблювальної системи є діагностичні центри автомобілів або станції технічного обслуговування (СТО), на яких можливо проводити діагностику автомобіля.

Блок керування двигуном керує всіма вимогами до двигуна, розставляє пріоритети, а потім виконує їх. Приклади вимог включають положення педаль акселератора та вимоги до складу суміші вихлопної системи. Крутний момент є ключовим критерієм виконання всіх вимог. Співвідношення повітря-паливо регулюється таким чином, щоб крутний момент був максимально ефективним. Окрім функцій, пов'язаних із спалюванням, електронний блок керування включає функції безпеки, безпеки та діагностики.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

У співпраці з блоками керування транспортним засобом блок керування двигуном здатний підтримувати майбутню технологію програмного перемикавання по повітря (FOTA / SOTA).

Керування двигуном також дозволяє активним системам безпеки водіння, таким як ASR і ESP®, втручатися в крутний момент двигуна.

Переваги продукту:

– Загальна платформа для бензинових, дизельних, альтернативних видів палива та гібридних силових агрегатів.

– Висока продуктивність і масштабованість для підтримки майбутніх вимог ринку та клієнтів.

– Ефективне калібрування даних із застосуванням передових методологій для спрощення архітектури програмного забезпечення та інтеграції функцій на основі моделі та самонавчання.

– Гнучке середовище розробки для ефективного обміну програмним забезпеченням із замовником.

– Індивідуальний, високоефективний перехід на нову платформу та повна відповідність AUTOSAR.

– Апаратний модуль безпеки для покращення кібербезпеки та налаштування захисту.

– Інтеграція додаткових функцій, таких як система обробки вихлопних газів, трансмісія та функції автомобіля.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи діагностування електронної системи автомобіля за стандартом ISO 14230, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.23.0015.00.00.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

Мобільний контролер CM3033

Модуль контролера CM3033 для мобільних транспортних засобів має гнучкі можливості вводу/виводу, включаючи можливості виходу 2А, 5А та 10А з незалежними шинами. Будучи частиною сімейства IQAN-Open, CM3033 можна програмувати за допомогою CODESYS[®], MATLAB[®] або C. Він спілкується за допомогою CAN 2.0B (SAE J1939) з підтримкою діагностики Ethernet.

Переваги CM3033

З'єднання:

- J1939 CAN і Ethernet, протокол UDS.
- П'ять шин CAN, включаючи одну CAN-FD.
- Інтерфейси з рішеннями Parker Mobile IoT.

Розширений і/о:

- Вихід до 10 А.
- 62 програмовані точки введення/виведення.
- Світлодіодні індикатори.

Легко програмувати:

- Опції CODESYS[®] /MATLAB[®] /C.
- Гнучкість розробки додатків.
- Функціональні блоки спрощують розробку.

TOPDON ArtiDiag800BT

Діагностичний сканер ArtiDiag800BT від TOPDON є нашим вибором як найкращий сканер OBD2, оскільки він має широкий спектр функцій, простий у

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

використанні та навіть працює бездротовим способом. Великий сенсорний екран має внутрішню батарею та підключається через ключ (діапазон 30,0 футів), тож ви можете вільно ходити навколо свого автомобіля, користуючись інструментом.

Виберіть цей сканер OBD2, якщо вам потрібен широкий спектр можливостей сканування, включаючи засоби діагностики SRS і ABS. Він сумісний з 96 виробниками транспортних засобів і є хорошим вибором для тих, кому потрібен інструмент із розширеними функціями OBD2.

Ключові особливості:

- Вартість: близько 420 доларів.
- Сумісність з WiFi.
- Автоматично визначає VIN, марку та модель автомобіля.
- Безкоштовні довічні оновлення програмного забезпечення.
- Річна гарантія.

До TOPDON ArtiDiag800BT входять:

- Інструмент сканування ArtiDiag800BT.
- Твердий футляр для транспортування.
- Ключ Bluetooth VCI.
- Зарядний кабель (USB-C).
- Зарядний адаптер (для європейських розеток).

Все, що стосується TOPDON, розроблено для простого використання та поводження. Пристрій поставляється в міцному корпусі ергономічної форми. Ключ Bluetooth[®] вставляється в сам сканер і легко знімається натисканням. Щоб налаштувати TOPDON, потрібне підключення до Інтернету, оскільки вам потрібно буде ввести наданий код активації. Після створення облікового запису TOPDON надає безкоштовні довічні оновлення.

Інтерфейс великого сенсорного екрана простий у навігації, але все ще оснащений такими функціями, як калібрування датчика кута повороту рульового колеса (SAS) і функція скидання сажового фільтра (DPF). Цей інструмент може використовуватися навіть тими, хто не має глибоких знань з обслуговування авто.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Розкладка меню є однією з найпростіших сканерів, які ми протестували, але все ще містить розширені функції, яких бракує простим сканерам, наприклад моніторинг SRS і ABS.

Інструмент сканування CRP129E

Засіб сканування Launch CRP129E посідає друге місце в категорії найкращий сканер OBD2 завдяки його можливостям діагностичного інструменту та широким можливостям за таку ціну. Цей 5,0-дюймовий сенсорний екран оснащено внутрішньою батареєю, тож вам не потрібно тримати автомобіль увімкненим, щоб ним користуватися. Ви також можете побудувати графік живих даних, скинути діагностичні коди несправностей (DTC) і виконати тести викидів.

Одним із плюсів є те, що Launch пропонує безкоштовні довічні оновлення, якщо ви купуєте їх у авторизованого дилера. Launch CRP129E поставляється з протоколами коду даних від майже 60 виробників автомобілів і понад 1000 моделей, тож цей сканер, ймовірно, щось знає про ваш автомобіль.

Ключові особливості:

- Вартість: близько 260 доларів.
- Сумісність з WiFi.
- Може надсилати звіти про двигун електронною поштою.
- Підтримує дев'ять мов.
- Включає скидання масла, калібрування кута повороту керма та скидання положення дросельної заслінки.
- Чуйний сенсорний екран високої чіткості.
- Гарантія п'ять років.

До інструменту сканування Launch CRP129E входить:

- Трубка Creader Professional 129E.
- Діагностичний кабель OBDII.
- Кабель для зарядки 5 В постійного струму.
- Посібник користувача.
- Сумка для перенесення.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Launch має простий у використанні сенсорний інтерфейс і довгий з'єднувальний кабель. Деякі сканери OBD2 мають коротші кабелі, що ускладнює утримання пристрою на колесах після підключення.

Launch містить багато корисних функцій і може виконувати багато діагностичних завдань, таких як графік живих даних і скидання датчиків. Для використання Launch вам знадобиться з'єднання Wi-Fi. Однак це з'єднання не має бути постійним, потрібно лише налаштувати пристрій. Ми також виявили, що Launch запускається трохи довше, ніж деякі сканери, але час запуску все одно відносно швидкий (лише на пару секунд довше, ніж інші сканери, які ми протестували).

Інструмент сканування BlueDriver Bluetooth Pro OBDII

Інструмент сканування BlueDriver Bluetooth® Pro OBDII запускає діагностичний ^{тест} на основі смартфона. Ми вибрали його як найкращий сканер Bluetooth OBD2, оскільки виробник включає регулярні оновлення програмного забезпечення для безкоштовного мобільного додатку, який працює зі смартфонами Android і Apple iPhone.

Ця відносно проста система підключає автомобільний сканер до порту OBD2 вашого автомобіля, який зазвичай розташований під рульовою колонкою. BlueDriver підключається через Bluetooth до програми для смартфона, з якої ви можете переглядати звіти з даними двигуна та систем охолодження. За допомогою цього сканера можна також виконати тест на смог, і він рекомендуватиме можливі ремонти для певних кодів.

Ключові особливості:

- Вартість: близько 120 доларів.
- Швидке та просте підключення Bluetooth.
- Зчитує коди помилок і пропонує ремонт.
- Безкоштовний додаток працює з пристроями iOS і Android.
- Інструмент «встановив і забув» для запису інформації про автомобіль.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

BlueDriver не постачається з жодними аксесуарами, окрім зарядного кабелю, і вам знадобиться смартфон, щоб використовувати цей сканер OBD2.

BlueDriver є одним із найпростіших у використанні сканерів OBD2, які ми протестували. Ви просто завантажуєте додаток для смартфона та підключаєте сканер до свого автомобіля. Діагностика та моніторинг виконуються через додаток із простим інтерфейсом.

Поряд із виконанням базових діагностичних завдань ви можете налаштувати налаштовуваний моніторинг базової інформації, наприклад частоти обертів двигуна та тиску вихлопу.

Сканер Autel MaxiCOM MK808

Ті, кому потрібен сканер OBD2 із безліччю функцій, оцінять сканер Autel MaxiCOM MK808, який дотримується межі між розширеними функціями сканування та бізнес-додатками. Цей гігант із 7,0-дюймовим сенсорним екраном HD має семигодинний час автономної роботи, тож покупці не прив'язані до автомобіля під час роботи.

Autel безкоштовно оновлює пристрій протягом першого року володіння, але він уже містить багато інформації про американські, європейські та азіатські марки автомобілів. Також включено програму, яка зберігає автомобільні та клієнтські дані, тому ви отримуєте сканер OBD2 і бортовий журнал в одному.

Ключові особливості:

- Вартість: близько 480 доларів.
- Одноетапна перевірка систем контролю тиску в шинах (TPMS).
- Автоматичний пошук ідентифікаційного номера автомобіля (VIN).
- Ємнісний сенсорний екран.
- Налаштовувані дані графіка.
- Підтримує карти microSD до 32 Гб.
- Міцний футляр для транспортування.

Autel MaxiCOM MK808 поставляється з кількома аксесуарами, зокрема:

- Кабель Mini USB.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

- USB зовнішній адаптер живлення.
- Основний кабель.
- Чохол для транспортування.

Autel має надійний і розширений інтерфейс користувача, який нагадує робочий стіл ПК. Сенсорний екран простий у навігації, а пристрій має широкий набір функцій, включаючи веб-браузер. Autel може підключатися до Wi-Fi, що необхідно для використання пристрою, принаймні під час запуску. Вам потрібно буде налаштувати обліковий запис Autel, щоб використовувати багато функцій сканера.

Цей сканер OBD2 може бути більшим, ніж вам потрібно, особливо якщо додаткові функції відволікають вас. Посібник користувача досить товстий, і ми не обов'язково рекомендуємо цей пристрій, якщо ви просто намагаєтеся прочитати коди помилок. Однак це чудовий вибір для досвідчених користувачів.

Foxwell NT201

Для більшості автовласників найкращий сканер OBD2 – це щось недороге, що може витягувати коди двигуна OBD2. Foxwell NT201 відповідає цим критеріям і має низьку вартість.

Хоча він не пропонує багато розширених функцій, ми виявили, що NT201 забезпечує збір базових даних, дані двигуна в реальному часі та дані стоп-кадру. За допомогою цього пристрою ви зможете з легкістю читати та очищати власні коди двигуна.

Ключові особливості:

- Вартість: близько 50 доларів.
- Дані в реальному часі та стоп-кадри.
- Функції моніторингу O2.
- Легкий корпус.

Foxwell NT201 не поставляється з жодними додатковими елементами, окрім самого сканера.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Перше, що ми помітили у Foxwell NT201, це його легкість. Це робить його простим у використанні, але також менш довговічним. Здається, що NT201 не витримає багатьох падінь на тверду підлогу. Тим не менш, цей недорогий сканер надзвичайно простий у використанні.

Хоча до комплекту не входить інструкція з експлуатації, користуватися NT201 дуже просто, оскільки немає багатьох кнопок або параметрів меню. Цей пристрій створено, щоб допомогти користувачам перевіряти коди двигунів і проводити тести на викиди. Ми рекомендуємо цей сканер OBD2 тим, хто хоче читати коди своїх двигунів вдома, але не потребує нічого іншого.

Innova CarScan Advisor 5210

Innova, каліфорнійський виробник діагностичних скануючих засобів, пропонує доступний автомобільний діагностичний пристрій, який не тільки очищає коди. Особливістю Innova CarScan Advisor 5210 є його здатність відображати «дані стоп-кадру». Цей параметр відображає інформацію про транспортний засіб, коли спрацьовує код несправності.

Користувачі також можуть оцінити свою антиблокувальну гальмівну систему (ABS) за допомогою цього пристрою, а також зчитувати та гасити індикатор перевірки двигуна однією кнопкою та запускати тести акумулятора та системи зарядки. Щоб отримати більш детальну інформацію про цей чудовий сканер, ознайомтеся з нашим посібником із детальним оглядом сканера Innova OBD2.

Ключові особливості:

- Вартість: близько 100 доларів.
- Кодуйте рівні серйозності, щоб швидко визначити потреби в ремонті.
- Додаток RepairSolutions2 дозволяє підключати смартфон.
- Читання та очищення кодів однією кнопкою.
- Працює з більшістю вітчизняних, азіатських та європейських автомобілів 1996 року випуску та новіших.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Innova CarScan Advisor не містить нічого, крім сканера. Ви повинні отримати посібник користувача онлайн, надавши адресу електронної пошти компанії Innova, яка потім надішле вам копію у форматі PDF.

Innova простий у використанні та налаштуванні, але має принаймні два недоліки: він не має стільки функціональних можливостей, як деякі сканери OBD2, а кабель надзвичайно короткий, тому вам потрібно нахилитися, щоб перевірити екран.

Користувальницький інтерфейс простий у використанні, але пристрій дещо обмежений у функціональності. Він не включає тестування гібридної батареї, а тестування ABS не вдалось, коли ми спробували використати його на Ford Fusion. Однак для простого пристрою, що підключається, це може бути хорошим варіантом. Просто переконайтеся, що він сумісний із вашим автомобілем.

Посібник для покупців: найкращий сканер OBD2

Вибір найкращого сканера OBD2 залежить від необхідної вам інформації про транспортний засіб і рівня ваших механічних знань. Якщо ви хочете зробити простий ремонт або переконатися, що ваш автомобіль пройде перевірку, стандартний сканер OBD2 може стати чудовою інвестицією. Якщо ви хочете виконувати більш складні завдання, наприклад перепрограмувати модуль керування двигуном, вам знадобиться спеціальний сканер.

Механіки, з якими ми спілкуємося, зазвичай кажуть нам, що найважливішою особливістю будь-якого сканера OBD2 є те, що він може зчитувати коди двигуна. Інші особливості дизайну можуть бути приємними, але для багатьох базовий зчитувач коду – це все, що їм потрібно. Зак Салерно, механік, який працює три роки, сказав нам: « Вам не потрібен Maximus 4.0 відразу. Фоксвелл вам підійде. Він прочитає ваші коди, і ви отримаєте уявлення про те, що відбувається. І це найбільша частина, принаймні отримати точку стрибка, щоб зрозуміти, з чого починається ваш діагноз».

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Купити на основі роботи

Найкращий спосіб визначити, який сканер OBD2 вам потрібен, це врахувати тип роботи, яку ви збираєтеся виконувати. Якщо ви берете вихідні, щоб ще раз перевірити оцінку від механіка або дізнатися, чому горить індикатор перевірки двигуна, вам знадобиться лише щось, що може читати коди.

Але майте на увазі, що не всі сканери OBD2 ідеально працюють з усіма марками та моделями транспортних засобів. Якщо у вас старіший або незвичайний автомобіль, вам, можливо, доведеться провести деякі дослідження, щоб дізнатися, які сканери OBD2 найкраще взаємодіють із вашим транспортним засобом.

Якщо ви постійно працюєте з кількома транспортними засобами як роботою чи як хобі, вам може знадобитися сканер, який сумісний з іноземними та вітчизняними автомобілями та регулярно отримує оновлення. Можливість надсилати команди або змінювати налаштування автомобіля корисна, якщо ви працюєте з датчиками температури або моніторами тиску в шинах. Доступ до показань, таких як ABS і інформація про пропуски запалювання, може допомогти краще діагностувати загальний стан автомобіля.

Тільки пам'ятайте: більше функцій і варіантів програмування означають вищу ціну.

Провідний проти Bluetooth

Коли ви шукаєте найкращий сканер OBD2, варто розглянути підключення Bluetooth. Оскільки фізичний шнур зазвичай тримає вас поблизу автомобіля під час діагностики, бездротові сканери набувають популярності через їх простоту та можливість збору даних.

Усе, що вам потрібно, це смартфон із підтримкою Bluetooth і відповідна програма для сканера (яка може бути не безкоштовною), і ви зможете переглядати дані безпосередньо зі свого телефону.

Це може бути чудовим варіантом, якщо ви шукаєте інструмент «встановив і забув», який збирає дані під час водіння, але він також дозволяє користувачам

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

заглядати під капот під час діагностики. Проте деякі сканери вищого класу постачаються з внутрішніми акумуляторами, яких вистачає на години, тому кількість необхідної інформації визначатиме найкращий сканер OBD2 для вашої роботи.

Марка транспортного засобу

Хоча багато кодів OBD2 є універсальними, велика кількість ні. На додаток до загальних кодів, усі автомобілі також мають коди несправностей виробника. Найкращі сканери OBD2 можуть зчитувати коди більшості провідних брендів. Якщо ви плануєте використовувати свій сканер OBD2 з певним транспортним засобом або набором транспортних засобів, переконайтеся, що він здатний зчитувати коди помилок цього виробника.

Вартість сканера OBD2

Сканер OBD2 може коштувати від 30 до 500 доларів США. Якщо ви не професійний механік, вам, ймовірно, не потрібен найдорожчий варіант. Однак найдешевші сканери OBD2 іноді мають приховані витрати, наприклад, ви повинні платити за підписку для доступу до розширених функцій. Ось що ви можете очікувати в кожному ціновому діапазоні:

Менше 60 доларів: ви можете придбати простий ефективний сканер OBD2 приблизно за 60 доларів. Пристрої цього діапазону можуть зчитувати хороший діапазон кодів несправностей OBD і виконувати перевірку викидів. Для деяких сканерів у цьому ціновому діапазоні може знадобитися підписка на оновлення. У цьому ціновому діапазоні є сканери Bluetooth OBD2, але вони або мають складні у використанні програми, або вимагають передплати.

Від 100 до 200 доларів США: сканери цього діапазону пропонують ширший спектр функцій, наприклад дані в реальному часі та моніторинг систем ABS і подушок безпеки. Багато сканерів Bluetooth OBD2 потрапляють у цей ціновий діапазон. Оскільки сканер Bluetooth використовує екран телефону та комп'ютер, він може бути дешевшим, ніж дротовий сканер із подібними

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

функціями. Якщо ви домашній механік, який працює над власним автомобілем, це хороший діапазон цін на сканер OBD2.

Понад 200 доларів США: ці сканери OBD2 зазвичай пропонують широкий спектр функцій, включаючи вдосконалене зчитування коду, SRS, трансмісію та моніторинг ABS. У цьому ціновому діапазоні сканери OBD2 повинні надавати точні дані в реальному часі та включати конструктивні особливості, такі як великий сенсорний дисплей. Ці сканери зазвичай працюють із багатьма виробниками, навіть із спеціалізованими брендами класу люкс, і можуть містити безкоштовні довічні оновлення.

Підсумок: Сканери OBD2

Ми вважаємо, що TOPDON ArtiDiag800BT є найкращим сканером OBD2. Він має широкий спектр діагностичних можливостей, а також пропонує зручний інтерфейс. З цієї причини це хороший вибір для механіка квітучого тіньового дерева. Тим не менш, він може запропонувати більше функцій, ніж буде потрібно деяким користувачам. Якщо ви хочете перевірити основні коди двигуна, Foxwell NT301 може бути кращим вибором (і за нижчою ціною).

Якщо ви шукаєте найбільш багатфункціональний сканер OBD2 із найширшим можливим діапазоном діагностичних можливостей, розгляньте Autel MaxiCOM MK808, хоча цей пристрій коштуватиме чимало, а оновлення не безкоштовні.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент належить й розроблюється Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

- Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

- Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

- Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

- Тип даних Delphi «record» тепер підтримуватиме довільні ініціалізацію, фіналізацію й операції копіювання.

- Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

- Відладник Win 64 (на LLDB) і збирач для C++.

- Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

- Підтримка Metal Driver GPU для macOS і iOS.

- Вбудований Fmxlinux.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API. Реалізація компонента Media Player для macOS тепер використовує Avfoundation. Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

– Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4к моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису `custom managed records`. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільнюються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

Розширена підтримка бібліотек C++

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Cmake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

– Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуються під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємі FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи діагностування електронної системи автомобіля за стандартом ISO 14230.

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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати 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-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Система функціонує наступним чином. Спершу EOM 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.

- PO – 1XX – Вимірники палива й повітря.
- PO – 100 – Несправність ланцюга датчика витрати повітря.
- PO – 101 – Вихід сигналу із припустимого діапазону.
- PO – 102 – Низький рівень вихідного сигналу.
- PO – 103 – Високий рівень вихідного сигналу.
- PO – 105 – Несправність датчика тиску повітря.
- PO – 106 – Вихід сигналу із припустимого діапазону.
- PO – 107 – Низький рівень вихідного сигналу.
- PO – 108 – Високий рівень вихідного сигналу.
- PO – 110 – Несправність датчика температури усмоктуваного повітря.
- PO – 111 – Вихід сигналу із припустимого діапазону.
- PO – 112 – Низький рівень вихідного сигналу.
- PO – 113 – Високий рівень вихідного сигналу.
- PO – 115 – Несправність датчика температури охолодної рідини.
- PO – 116 – Вихід сигналу із припустимого діапазону.
- PO – 117 – Низький рівень вихідного сигналу.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

PO – 118 – Високий рівень вихідного сигналу.
PO – 120 – Несправність датчика положення дросельної заслінки.
PO – 121 – Вихід сигналу із припустимого діапазону.
PO – 122 – Низький рівень вихідного сигналу.
PO – 123 – Високий рівень вихідного сигналу.
PO – 125 – Низька температури охолодної рідини для управління по замкнутому контуру.

PO – 130 – Датчик O2 B1 S1 несправний (Банк1).
PO – 131 – Датчик O2 B1 S1 має низький рівень сигналу.
PO – 132 – Датчик O2 B1 S1 має високий рівень сигналу.
PO – 133 – Датчик O2 B1 S1 має повільний відгук на збагачення/збідніння.
PO – 134 – Ланцюг датчика O2 B1 S1 пасивна.
PO – 135 – Нагрівач датчика O2 B1 S1 несправний.
PO – 136 – Датчик O2 B1 S2 несправний.
PO – 137 – Датчик O2 B1 S2 має низький рівень сигналу.
PO – 138 – Датчик O2 B1 S2 має високий рівень сигналу.
PO – 139 – Датчик O2 B1 S2 має повільний відгук на збагачення/збідніння.
PO – 140 – Ланцюг датчика O2 B1 S2 пасивна.
PO – 141 – Нагрівач датчика O2 B1 S2 несправний.
PO – 142 – Датчик O2 B1 S3 несправний.
PO – 143 – Датчик O2 B1 S3 має низький рівень сигналу.
PO – 144 – Датчик O2 B1 S3 має високий рівень сигналу.
PO – 145 – Датчик O2 B1 S3 має повільний відгук на збагачення/збідніння.
PO – 146 – Ланцюг датчика O2 B1 S3 пасивна.
PO – 147 – Нагрівач датчика O2 B1 S3 несправний.
PO – 150 – Датчик O2 Y2 S1 несправний (Банк2).
PO – 151 – Датчик O2 Y2 S1 має низький рівень сигналу.
PO – 152 – Датчик O2 Y2 S1 має високий рівень сигналу.
PO – 153 – Датчик O2 Y2 S1 має повільний відгук на збагачення/збідніння.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

- PO – 154 – Ланцюг датчика O2 Y2 S1 пасивна.
- PO – 155 – Нагрівач датчика O2 Y2 S1 несправний.
- PO – 156 – Датчик O2 Y2 S2 несправний.
- PO – 157 – Датчик O2 Y2 S2 має низький рівень сигналу.
- PO – 158 – Датчик O2 Y2 S2 має високий рівень сигналу.
- PO – 159 – Датчик O2 Y2 S2 має повільний відгук на збагачення/збідніння.
- PO – 160 – Ланцюг датчика O2 Y2 S2 пасивна.
- PO – 161 – Нагрівач датчика O2 Y2 S2 несправний.
- PO – 162 – Датчик O2 Y2 S3 несправний.
- PO – 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 СНх несправний.
- PO – 177 – Сигнал датчика виходить із припустимого діапазону.
- PO – 178 – Низький рівень сигналу датчика СНх.
- PO – 179 – Високий рівень сигналу датчика СНх.
- 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».

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

- 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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

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 – Трансмсія перебуває в перегрітому стані.

PO – 219 – Двигун перекручений.

PO – 220 – Несправність датчика положення дросельної заслінки «В».

PO – 221 – Вихід сигналу із припустимого діапазону.

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

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

PO – 224 – Сигнал датчика «В» перемежований.

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

PO – 226 – Вихід сигналу із припустимого діапазону.

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

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

PO – 229 – Сигнал датчика «С» перемежований.

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

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

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

PO – 233 – Вторинний ланцюг бензонасоса має перемежований рівень.

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

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

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

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

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

PO – 255 – Сигнал насоса упорскування турбіни «А» перемерзований.

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

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

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

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

PO – 260 – Сигнал насоса упорскування турбіни «В» перемерзований.

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

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

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

- PO – 263 – Драйвер форсунки 1-ого циліндра несправний.
- PO – 264 – Форсунка 2-ого циліндри замкнутий на землю.
- PO – 265 – Форсунка 2-ого циліндри обірваний або замкнута на +12В.
- PO – 266 – Драйвер форсунки 2-ого циліндри несправний.
- PO – 267 – Форсунка 3-го циліндри замкнутий на землю.
- PO – 268 – Форсунка 3-го циліндри обірваний або замкнута на +12В.
- PO – 269 – Драйвер форсунки 3-го циліндри несправний.
- PO – 270 – Форсунка 4-ого циліндри замкнутий на землю.
- PO – 271 – Форсунка 4-ого циліндри обірваний або замкнута на +12В.
- PO – 272 – Драйвер форсунки 4-ого циліндри несправний.
- PO – 273 – Форсунка 5-ого циліндра замкнута на землю.
- PO – 274 – Форсунка 5-ого циліндра обірвана або замкнута на +12В.
- PO – 275 – Драйвер форсунки 5-ого циліндра несправний.
- PO – 276 – Форсунка 6-ого циліндра замкнута на землю.
- PO – 277 – Форсунка 6-ого циліндра обірвана або замкнута на +12В.
- PO – 278 – Драйвер форсунки 6-ого циліндра несправний.
- PO – 279 – Форсунка 7-ого циліндра замкнута на землю.
- PO – 280 – Форсунка 7-ого циліндра обірвана або замкнута на +12В.
- PO – 281 – Драйвер форсунки 7-ого циліндра несправний.
- PO – 282 – Форсунка 8-ого циліндра замкнута на землю.
- PO – 283 – Форсунка 8-ого циліндра обірвана або замкнута на +12В.
- PO – 284 – Драйвер форсунки 8-ого циліндра несправний.
- PO – 285 – Форсунка 9-ого циліндра замкнута на землю.
- PO – 286 – Форсунка 9-ого циліндра обірвана або замкнута на +12В.
- PO – 287 – Драйвер форсунки 9-ого циліндра несправний.
- PO – 288 – Форсунка 10-ого циліндра замкнута на землю.
- PO – 289 – Форсунка 10-ого циліндра обірвана або замкнута на +12В.
- PO – 290 – Драйвер форсунки 10-ого циліндра несправний.
- PO – 291 – Форсунка 11-ого циліндра замкнута на землю.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

- PO – 292 – Форсунка 11-ого циліндра обірвана або замкнута на +12В.
- PO – 293 – Драйвер форсунки 11-ого циліндра несправний.
- PO – 294 – Форсунка 12-ого циліндра замкнута на землю.
- PO – 295 – Форсунка 12-ого циліндра обірвана або замкнута на +12В.
- PO – 296 – Драйвер форсунки 12-ого циліндра несправний.
- PO – 3XX – Система запалювання й пропуски.
- PO – 300 – Виявлені випадкові/множинні пропуски запалювання.
- PO – 301 – Виявлені пропуски запалювання в 1-ому циліндрі.
- PO – 302 – Виявлені пропуски запалювання в 2-ому циліндрі.
- PO – 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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

- 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 перемержований.
- 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 замкнуть на масу.
- PO – 338 – Сигнал датчика «А» має високий рівень або замкнуть на 12В.
- PO – 339 – Сигнал датчика «А» перемержований.
- PO – 340 – Датчик положення розподільного вала несправний.
- PO – 341 – Сигнал датчика виходить за припустимі межі.
- PO – 342 – Сигнал датчика має низький рівень або замкнуть на масу.
- PO – 343 – Сигнал датчика має високий рівень.
- PO – 344 – Сигнал датчика перемержований.
- PO – 350 – Первиний/вториний ланцюг котушки запалювання несправні.
- PO – 351 – Первиний/вториний ланцюг котушки запалювання «А» несправні.
- PO – 352 – Первиний/вториний ланцюг котушки запалювання «В» несправні.
- PO – 353 – Первиний/вториний ланцюг котушки запалювання «С» несправні.
- PO – 354 – Первиний/вториний ланцюг котушки запалювання «D» несправні.
- PO – 355 – Первиний/вториний ланцюг котушки запалювання «Е» несправні.
- PO – 356 – Первиний/вториний ланцюг котушки запалювання «F» несправні.
- PO – 357 – Первиний/вториний ланцюг котушки запалювання «G»

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

несправні.

PO – 358 – Первиний/вториний ланцюг котушки запалювання «Н»

несправні.

PO – 359 – Первиний/вториний ланцюг котушки запалювання «І»

несправні.

PO – 360 – Первиний/вториний ланцюг котушки запалювання «J»

несправні.

PO – 361 – Первиний/вториний ланцюг котушки запалювання «K»

несправні.

PO – 362 – Первиний/вториний ланцюг котушки запалювання «L»

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» перемерзований.

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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

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 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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

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

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

PO – 448 – Повітряний клапан системи «EVAP» 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 перережований.

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

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 перережований.

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 перережований.

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 перережований.

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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

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.

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 перемежований.

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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

- 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 замкнуть.
- PO – 573 – Перемикач завжди розімкнуть.
- 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.
- PO – 604 – Помилка оперативного запам'ятовувального пристрою.
- PO – 605 – Помилка постійного запам'ятовувального пристрою.
- PO – 606 – Помилка модуля управління енергозбереженням.
- PO – 700 – Система управління трансмісією несправний.
- PO – 701 – Система управління трансмісією працює невірно.
- PO – 703 – Перемикач карданний вал/гальма несправний.
- PO – 704 – Ланцюг датчика включення зчеплення несправний.
- PO – 705 – Датчик діапазону роботи трансмісії несправний.
- PO – 706 – Сигнал датчика виходить за припустимі межі.
- PO – 707 – Сигнал датчика має низький рівень.
- PO – 708 – має високий рівень.
- PO – 709 – Сигнал датчика перемежований.
- PO – 710 – Датчик температури трансмісійної рідини несправний.
- PO – 711 – Сигнал датчика виходить за припустимі межі.
- PO – 712 – Сигнал датчика має низький рівень.
- PO – 713 – має високий рівень.
- PO – 714 – Сигнал датчика перемежований.
- PO – 715 – Датчик швидкості турбіни несправний.
- PO – 716 – Сигнал датчика виходить за припустимі межі.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

PO – 717 – Сигнал датчика відсутній.
PO – 718 – Сигнал датчика перережований.
PO – 719 – Перемикач карданний вал/гальма замкнута на масу.
PO – 720 – Ланцюг датчика «Зовнішньої швидкості» несправний.
PO – 721 – Сигнал датчика «Зовнішньої швидкості» виходить за допустимі межі.

PO – 722 – Сигнал датчика «Зовнішньої швидкості» відсутній.
PO – 723 – Сигнал датчика «Зовнішньої швидкості» перережований.
PO – 724 – Перемикач карданний вал/гальма замкнута на живлення.
PO – 725 – Ланцюг датчика швидкості обертання двигуна несправний.
PO – 726 – Сигнал датчика виходить за припустимі межі.
PO – 727 – Сигнал датчика відсутній.

PO – 728 – Сигнал датчика перережований.

PO – 730 – Передаточне число трансмісії невірне.

PO – 731 – Передаточне число трансмісії на 1 передачі невірне.

PO – 732 – Передаточне число трансмісії на 2 передачі невірне.

PO – 733 – Передаточне число трансмісії на 3 передачі невірне.

PO – 734 – Передаточне число трансмісії на 4 передачі невірне.

PO – 735 – Передаточне число трансмісії на 5 передачі невірне.

PO – 736 – Передаточне число трансмісії на передачі заднього ходу невірне.

PO – 740 – Ланцюг управління блокуванням диференціала несправний.

PO – 741 – Диференціал завжди виключений (розблокований).

PO – 742 – Диференціал завжди включений (заблокований).

PO – 743 – Зарезервовано.

PO – 744 – Диференціал стан нестійке.

PO – 745 – Управління стискаючим соленоїдом несправно.

PO – 746 – Соленоїд завжди у виключеному стані.

PO – 747 – Соленоїд завжди у включеному стані.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

- PO – 749 – Стан соленоїда хитливий.
- PO – 750 – Соленоїд «А» включення передачі несправний.
- PO – 751 – Соленоїд «А» завжди у виключеному стані.
- PO – 752 – Соленоїд «А» завжди у включеному стані.
- PO – 754 – Стан соленоїда «А» хитливий.
- PO – 755 – Соленоїд «В» включення передачі несправний.
- PO – 756 – Соленоїд «В» завжди у виключеному стані.
- PO – 757 – Соленоїд «В» завжди у включеному стані.
- PO – 759 – Стан соленоїда «В» хитливий.
- PO – 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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

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.

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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

3.2 Розробка структурної схеми

Структурна схема системи зображена на рисунку 3.1.

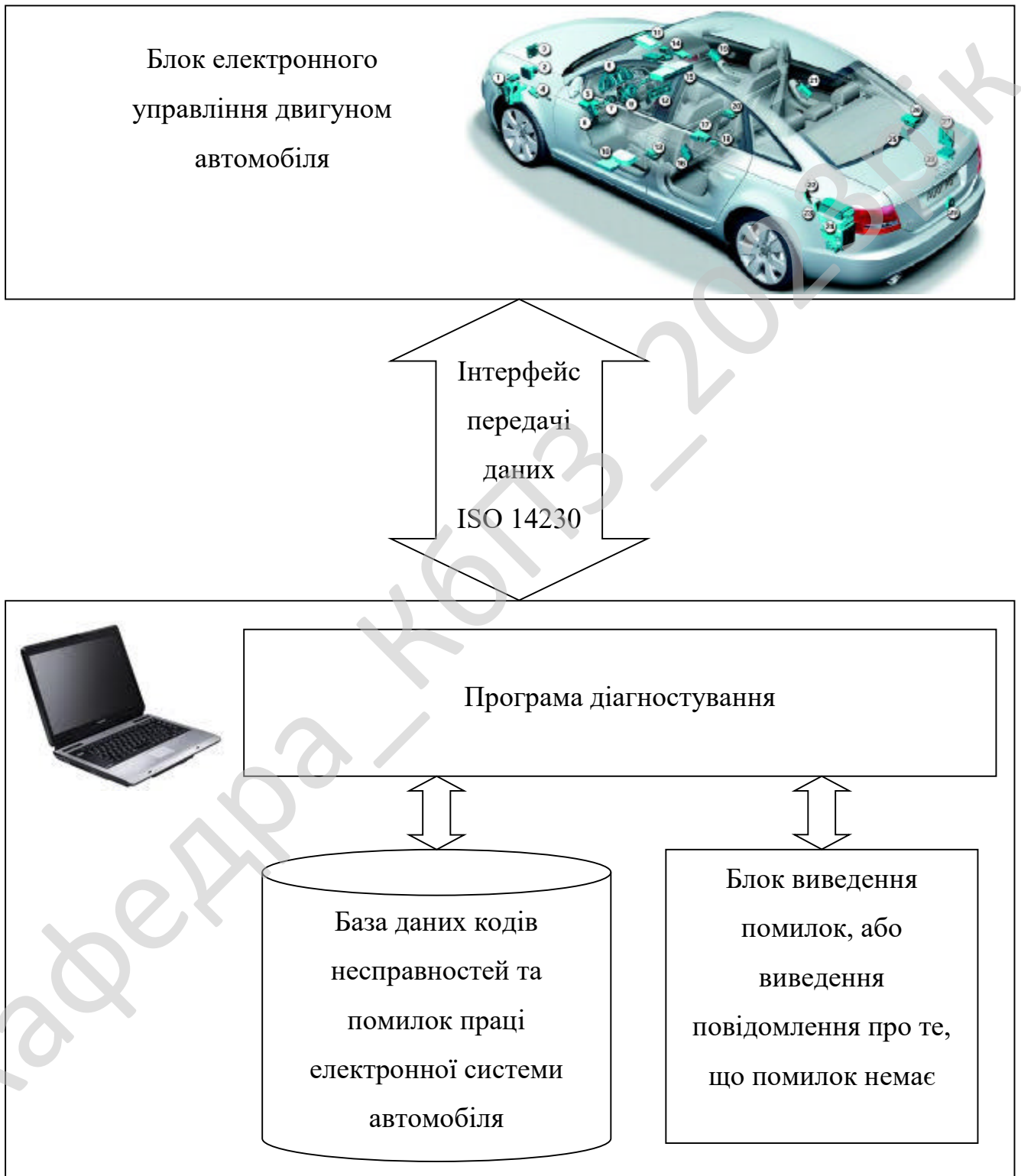


Рисунок 3.1 – Структурна схема системи

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, підвісці, 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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

- 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-дисківід.
- Блок управління в задніх лівих дверях.
- Блок управління системою Air-Bag.
- Датчик швидкості обертання автомобіля навколо вертикальної осі.
- Блок управління у двері переднього пасажира.
- Блок управління регулюванням сидіння переднього пасажира 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-дисківідом.
- Блок управління голосовим уведенням.
- Блок управління цифровою звуковою системою; радіомодуль; TV-тюнер; цифрове радіо.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

- 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).

Як користуватися сканером OBD2

Хоча кожен сканер OBD2 має різні кнопки та інший інтерфейс, багато з них працюють однаково. Щоб скористатися найдосконалішими функціями вашого сканера, вам потрібно буде ознайомитися з посібником користувача. В іншому випадку процес дуже простий.

Підключіть його

Ваш перший крок – знайти порт OBD2. Зазвичай він розташований під кермом і зліва. Слот має форму трапеції і містить 16 штирів. Розташування цього порту має бути описано в посібнику з експлуатації вашого автомобіля.

Запустіть свій двигун

Щоб зчитувати коди несправностей, ваш автомобіль (і його комп'ютер) має бути увімкнено. Після підключення сканера OBD2 запустіть двигун.

Читання та очищення кодів

Найпростіша функція OBD (яку має вміти виконувати кожен сканер OBD2) – читати та очищати коди помилок. Конкретна інформація, що надається, залежить від сканера, який ви купуєте, але будь-який сучасний сканер повинен бути в змозі сказати вам, чому горить індикатор перевірки двигуна.

Сканери OBD2 також можуть очищати коди з комп'ютера вашої системи. Якщо основну проблему не усунути, цей код (і рядок перевірки механізму) зрештою повернеться.

Тест на викиди

Більшість сканерів OBD2 мають функцію перевірки викидів. Запустивши тест на викиди, сканер може допомогти вам визначити, чи пройдете ви державну перевірку до того, як це станеться, потенційно заощадивши вам багато часу. Якщо

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

виникне проблема, ви можете вирішити її, перш ніж вивезти свій автомобіль на техогляд.

Дані в реальному часі

Більшості людей доступ до даних у реальному часі не потрібен. Для тих, хто володіє передовими знаннями в автомобільній галузі, ця інформація може допомогти оптимізувати роботу двигуна та діагностувати певні ремонти. Окрім інформації про двигун, як-от температури, частоти обертання та швидкості впорскування палива, ви також можете контролювати дані ABS та акумулятора.

3.3 Розробка функціональної схеми

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".

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

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

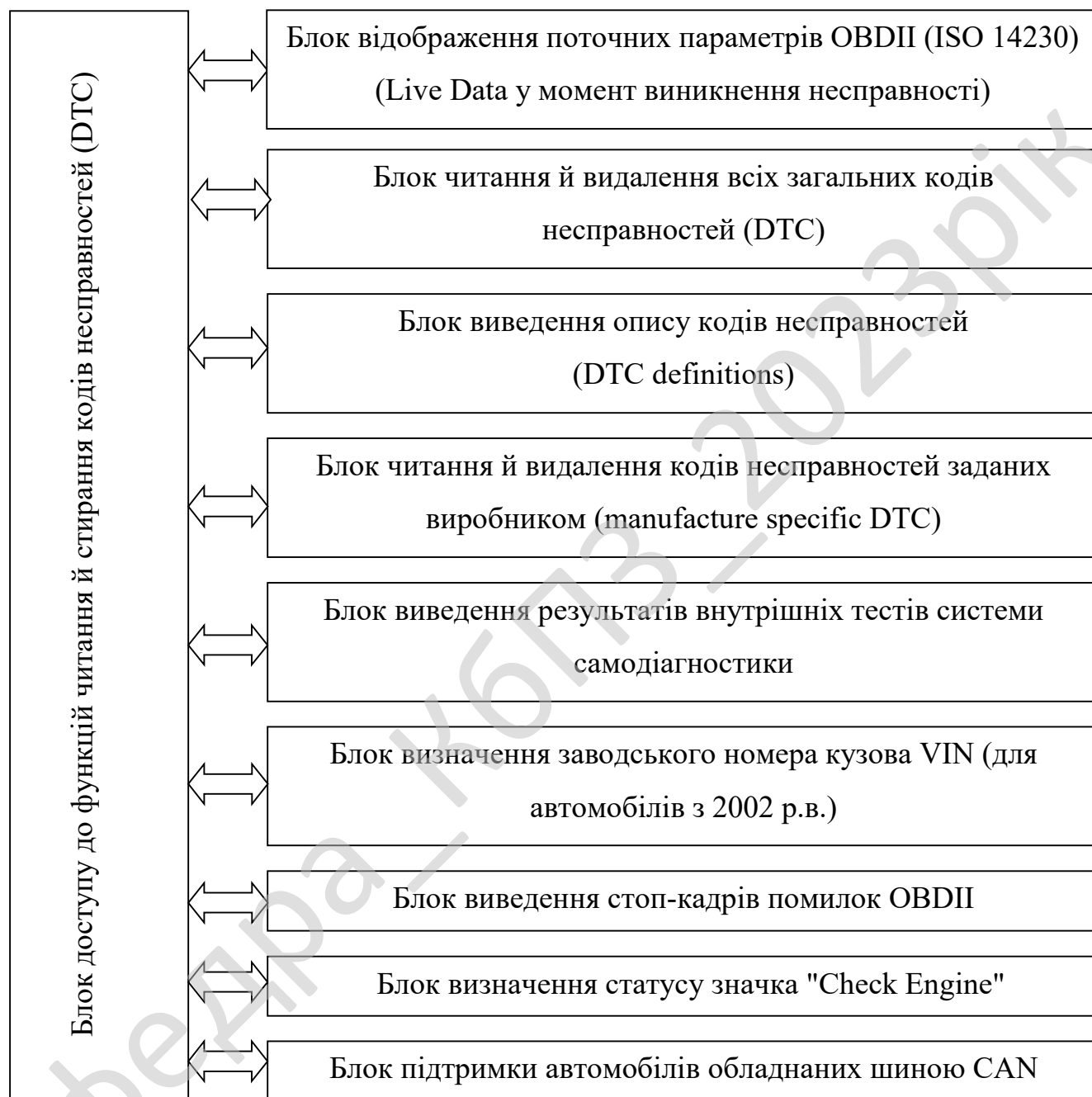


Рисунок 3.2 – Функціональна схема системи

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, 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.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

Кафедра _ КБПЗ _ 2023рік

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ 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.

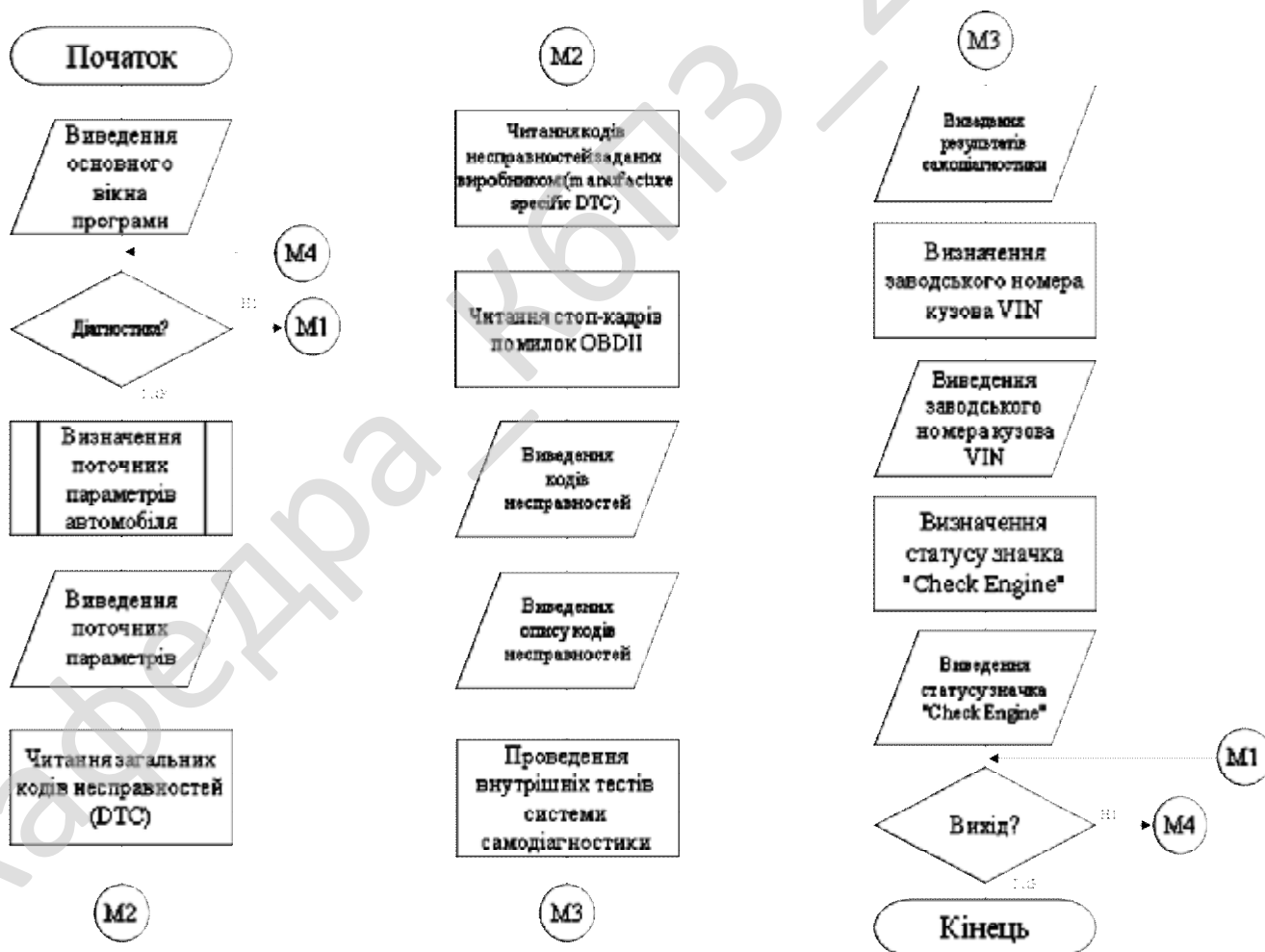


Рисунок 4.1 – Блок-схема роботи основної програми

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-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

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

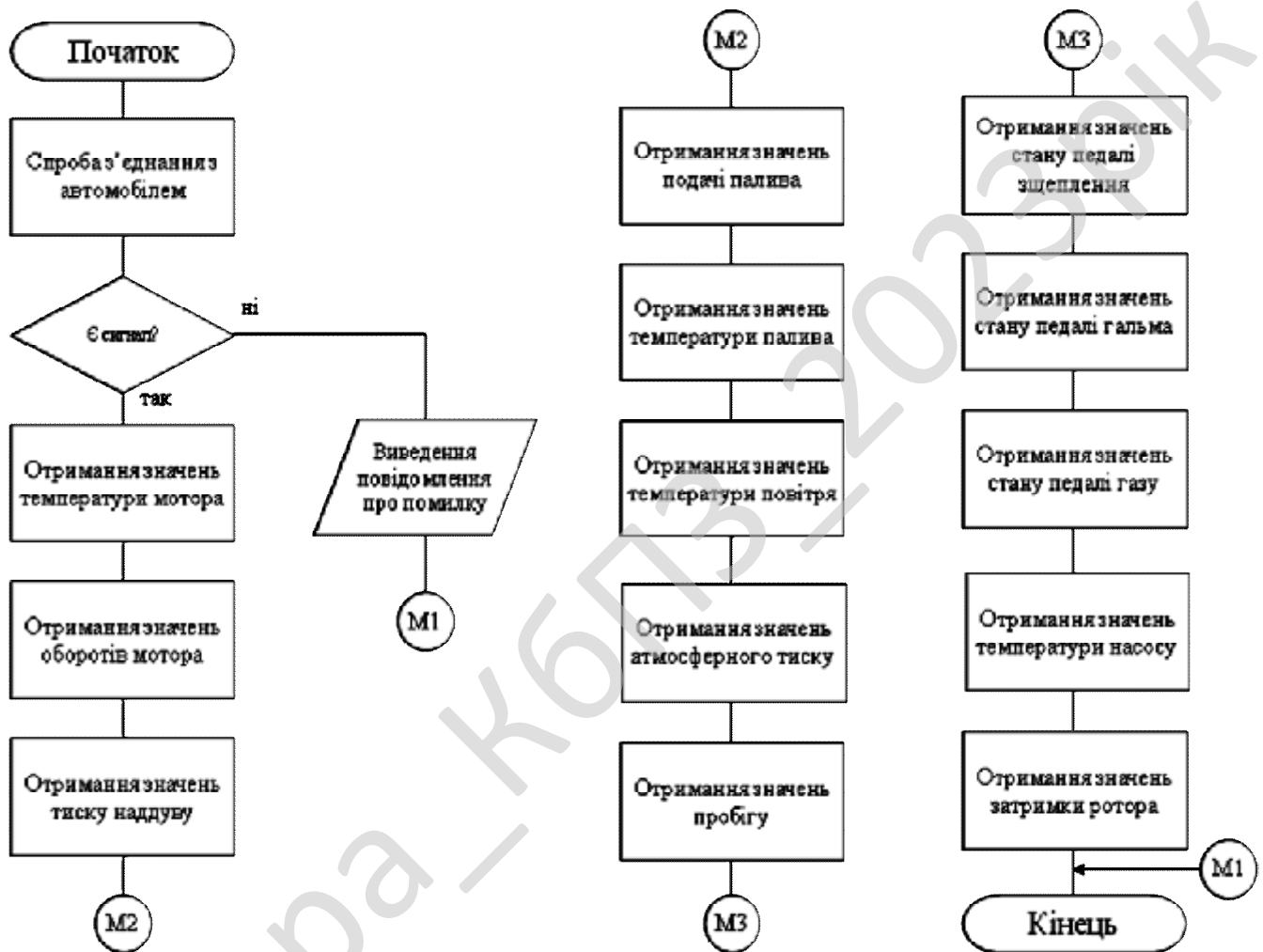


Рисунок 4.2 – Блок-схема роботи підпрограми визначення поточних параметрів автомобіля

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 встановлювалися системи діагностики з 10-ти контактним роз'ємом і протоколами діагностики ALDL (Multec) або K-Line (Motronic).

На OPEL випуску 94-96 років встановлювалася 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 роз'єманні.

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.

В OPEL Kadett, Vectra і Omega він перебуває біля опори стійки, з лівої або правої сторони моторного відсіку. В OPEL Tigra, Corsa і Astra – у щитку запобіжників салону. В OPEL Frontera його ставлять над правою (пасажирської) фарею.

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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

F – "+" Акумуляторної батареї.

G – K-Line канал даних комп'ютерів.

H – Самодіагностика круїзу-контролю.

J – Самодіагностика повного привода.

K – Самодіагностика ABS.

2. Ставимо дротову 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.

AH – 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 KM 602-2"

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

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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

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

1. Знаходимо сам роз'єм.

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.

x-x-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 (указувати на несправність систем, що перебувають

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

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.

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

З портами з-під Win32 працюють так само, як і зі звичайними файлами, використовуючи при цьому всього кілька специфічних функцій WinAPI. Однак комунікаційний порт – це не зовсім звичайний файл. Для нього, наприклад, не можна виконати позиціонування файлового покажчика, або ж створити порт, якщо такий відсутній. Будь-яка робота з портом починається з його відкриття. Для цього використовується файлова функція WinAPI (опис WinAPI-функцій узяті з MSDN (Microsoft Developer Network), отже, приводяться в синтаксисі C):

```
HANDLE CreateFile(  
    LPCTSTR lpFileName,  
    DWORD dwDesiredAccess,  
    DWORD dwShareMode,  
    LPSECURITY_ATTRIBUTES lpSecurityAttributes,  
    DWORD dwCreationDisposition,  
    DWORD dwFlagsAndAttributes,  
    HANDLE hTemplateFile  
);
```

`lpFileName` – покажчик на рядок з нульовим завершальним символом. Звичайно це ім'я файлу, що відкривається, але в нашій випадку це повинне бути назва порту (COM1, COM2, ...).

`dwDesiredAccess` – тип доступу. У нашій випадку повинен бути дорівнює `GENERIC_READ|GENERIC_WRITE`.

`dwShareMode` – параметр спільного доступу. Для комунікаційних портів завжди дорівнює 0.

`lpSecurityAttributes` – атрибут захисту. Для комунікаційних портів завжди дорівнює `NULL`.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

dwCreationDistribution – режим автостворення. Для комунікаційних портів завжди дорівнює OPEN_EXISTING.

dwFlagsAndAttributes – атрибут режиму обробки. Для комунікаційних портів повинен бути дорівнює 0 або FILE_FLAG_OVERLAPPED.

hTemplateFile – описувач файлу-шаблону. Для комунікаційних портів повинен бути дорівнює NULL.

При успішному відкритті порту функція повертає його описувач, а у випадку помилки повертає INVALID_HANDLE_VALUE.

Із всіх параметрів функції CreateFile() особливого пояснення вимагає dwFlagsAndAttributes. Робота з портом може бути організована в синхронному (nonoverlapped) або асинхронному (overlapped) режимах обробки, що й задається цим прапором. При синхронному режимі (коли параметр dwFlagsAndAttributes = 0) тільки один потік додатка може або читати, або писати в порт.

Синхронний режим обробки простий у реалізації. Якщо треба записати дані в порт, то викликаємо функцію запису й очікуємо, поки вона не завершиться. Якщо ж треба читати дані, то викликаємо функцію читання й чекаємо, поки вона не відробить. Для простих завдань синхронний режим обробки цілком підходить, однак у світі Windows він майже завжди приречений на невдачу. Очікування операції читання або записи сприймається користувачем програми як «зависання».

Асинхронний режим (коли параметр dwFlagsAndAttributes = FILE_FLAG_OVERLAPPED) дозволяє робити операції читання й запису в порт паралельно з різних потоків. У той час, поки один потік додатка приймає дані, інший потік може паралельно з першим передавати дані – як при розмові по телефоні, коли ви можете слухати й говорити одночасно. Даний режим обробки більше імпонує ідеї багатозадачності Windows. Але за все треба платити: для реалізації цього режиму обробки потрібно у два рази більше написаного коду, до

того ж, уміння писати багатопоточні програми. Який режим вибрати – вирішуйте самі. Розглянемо більше складний варіант – асинхронну обробку.

На практиці відкриття порту для асинхронного режиму обробки із програми на Delphi виглядає так:

```
hPort := CreateFile('COM1', GENERIC_READ or GENERIC_WRITE, 0, nil,  
    OPEN_EXISTING, FILE_FLAG_OVERLAPPED, 0);  
if hPort = INVALID_HANDLE_VALUE then  
    raise Exception.Create('Помилка відкриття порту');
```

Функція повертає описувач порту (hPort), що нам потім придасться для виклику інших функцій роботи з портом. Якщо в результаті відкриття порту описувач не отриманий, то збуджується виключення з відповідним текстом помилки. Відкривши порт, ми одержуємо його у своє розпорядження. Тепер із цим портом може працювати тільки наша програма (точніше, тільки наш процес).

По закінченні роботи з портом його варто закрити, викликавши функцію:

```
BOOL CloseHandle( HANDLE hObject );
```

Як єдиний параметр треба передати отриманий раніше описувач порту (hPort).

Хоч система при завершенні виконання програми й звільняє всі виділені їй ресурси (у тому числі й порти), гарним тоном програмування вважається власноручне закриття портів. Відкривати/закривати порт начебто нескладно. Крім того, нам буде потрібно програмне налаштування порту. Думаю, усі бачили діалог налаштування послідовного порту в диспетчері пристроїв системи. Всі ці налаштування ми можемо зробити програмно. Для цих цілей використовується функція WinAPI:

```
BOOL SetCommState( HANDLE hFile, LPDCB lpDCB );
```

hFile – описувач відкритого порту.

lpDCB – покажчик на структуру DCB.

Основні параметри послідовного порту описуються структурою DCB. Вона містить масу полів, кожне з яких відповідає певному параметру налаштування порту. Ми розглянемо кілька полів, які нам потрібні:

dwFlags – вироблені дії у вигляді набору прапорів PURGE_TXABORT, PURGE_RXABORT, PURGE_TXCLEAR, PURGE_RXCLEAR.

```
if not PurgeComm(hPort, PURGE_TXCLEAR or PURGE_RXCLEAR) then  
    raise Exception.Create('Помилка, що очищає порт');
```

На цьому підготовча фаза закінчується, і можна приступати безпосередньо до прийому/передачі даних. Прийом даних у нас буде відбуватися по подійній схемі; програма буде очікувати прийом одного або декількох символів (байт). Для перекладу порту в цей режим необхідно викликати функцію SetCommMask() із прапором EV_RXCHAR:

```
if not SetCommMask(hPort, EV_RXCHAR) then  
    raise Exception.Create('Помилка установки маски порту');
```

Прийом і передача даних виконується функціями ReadFile() і WriteFile(), тобто тими ж самими функціями, які використовуються для роботи з дисковими файлами. От їхній опис:

```
BOOL ReadFile(  
    HANDLE hFile,  
    LPVOID lpBuffer,  
    DWORD nNumberOfBytesToRead,  
    LPDWORD lpNumberOfBytesRead,  
    LPOVERLAPPED lpOverlapped  
);  
  
BOOL WriteFile(  
    HANDLE hFile,  
    LPCVOID lpBuffer,  
    DWORD nNumberOfBytesToWrite,  
    LPDWORD lpNumberOfBytesWritten,  
    LPOVERLAPPED lpOverlapped  
);
```

hFile – описувач відкритого порту.

lpBuffer – адреса буфера.

nNumberOfBytesToRead/nNumberOfBytesToWrite – число очікуваних до прийому або призначених для передачі байт.

lpNumberOfBytesRead/lpNumberOfBytesWritten – число фактично прийнятих або переданих байт.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

В будь-якого потоку є головна функція, що починає виконуватися після його створення. В Delphi для потоків існує клас TThread, а його головна процедура називається TThread.Execute().

От так виглядає головна процедура окремого потоку, що очікує появу одного або декількох символів і зчитує їх:

```
procedure TReadThread.Execute;
var
  ComStat: TComStat;
  dwMask, dwError: DWORD;
  OverRead: TOverlapped;
  Buf: array[0..$FF] of Byte;
  dwRead: DWORD;
begin
  OverRead.hEvent := CreateEvent(nil, True, False, nil);
  if OverRead.hEvent = Null then
    raise Exception.Create(' Помилка створення події читання ');
  FreeOnTerminate := True;
  while not Terminated do
  begin
    if not WaitCommEvent(hPort, dwMask, @OverRead) then
    begin
      if GetLastError = ERROR_IO_PENDING then
        WaitForSingleObject(OverRead.hEvent, INFINITE)
      else
        raise Exception.Create('Помилка очікування події порту');
    end;
    if not ClearCommError(hPort, dwError, @ComStat) then
      raise Exception.Create('Помилка очищення порту');
    dwRead := ComStat.cbInQue;
    if dwRead > 0 then
    begin
      if not ReadFile(hPort, Buf, dwRead, dwRead, @OverRead) then
        raise Exception.Create('Помилка читання порту');
      // В Buf перебувають прочитані байти
      // Далі йде обробка прийнятих байтів
    end;
  end; {while}
end;
```

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

У наведеному кодї в потоці крутиться цикл, тим самим ініціюється очікування події порту викликом функції `WaitCommEvent()`, очікування ж самої цієї події задається функцією `WaitForSingleObject()`. Для визначення кількості прийнятих символів використовується функція `ClearCommError()`. Коли кількість прийнятих символів (`dwRead`) відомо, безпосереднє читання символів виконується функцією `ReadFile()`.

Використовуючи вищеописані викладення, я написав на Delphi клас `TComPort` для роботи з COM-портами. Для передачі й прийому даних передбачені окремі вікна. Формат переданих даних – рядок. Прийняті дані представляються у вигляді масиву байт.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм `LOKI_91`. Механізм алгоритму `LOKI_91` подібний `DES` (рисунок 4.3). Блок даних розщеплюється на ліву й праву половини й проходить 16 раундів, що досить нагадує `DES`. У кожному раунді права половина спочатку піддається операції `XOR` із частиною ключа, а потім розширювальній перестановці (таблиця 4.1).

Таблиця 4.1 – Перестановка з розширенням

4	3	2	1	32	31	30	29	28	27	26	25
28	27	26	25	24	23	22	21	20	19	18	17
20	19	18	17	16	15	14	13	12	11	10	9
12	11	10	9	8	7	6	5	4	3	2	1

48-бітовий вихід розділяється на чотири 12-бітових блоки. У кожному блоці виконується така підстановка з використанням S-блоку: береться кожний 12-бітовий вхід, 2 старших і 2 молодших біти використовуються для утворення

номера r , а вісім внутрішніх біт утворять номер s . Вихід S-блоку, O , має наступне значення: $O(r,c) = (c + ((r*17) \oplus 0xff) \& 0xff)^{31} \bmod P_r$.

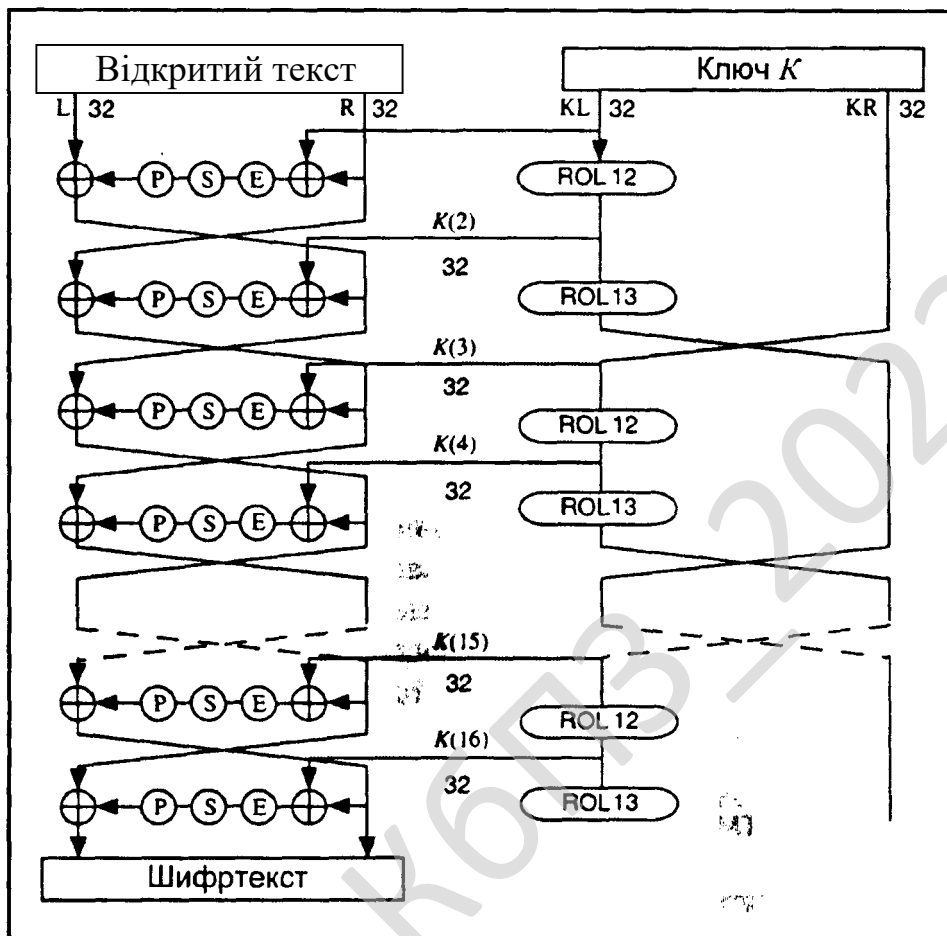


Рисунок 4.3 – Алгоритм LOKI91

Таблиця 4.2 – Значення P_r

r	1	2	3	4	5	6	7	8
P_r	375	379	391	395	397	415	419	425
r	9	10	11	12	13	14	15	16
P_r	433	445	451	463	471	477	487	499

Після цього чотири 8-бітових результати знову поєднуються, утворюючи 32-бітове число, що піддається операції перестановки, описаній в таблиці 3.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Для установки утиліти запустіть установочний файл відповідно на КПК, CarPC, персональному комп'ютері або ноутбуці й додержуйтеся інструкцій програми установки. Після установки утиліти натисніть «Настроювання». Уведіть серійний номер адаптера й номер СОМ 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 російською 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-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи діагностування електронної системи автомобіля за стандартом ISO 14230.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем діагностування електронної системи автомобіля за стандартом ISO 14230.

– Досліджена система діагностування електронної системи автомобіля за стандартом ISO 14230.

– На основі отриманих результатів досліджень створена програмна реалізація системи діагностування електронної системи автомобіля за стандартом ISO 14230.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання діагностування електронної системи автомобіля за стандартом ISO 14230.

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 RAD Studio Delphi 10.4. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

системи діагностування електронної системи автомобіля за стандартом ISO 14230. 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 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 LOKI_91.

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.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. В. Ф. Яковлев. Диагностика электронных систем автомобиля. Солон-пресс, 2003 г. – 272 стр.
2. А. А. Тюнин. Диагностика электронных систем управления двигателями легковых автомобилей. Практическое пособие. СОЛОН-ПРЕСС. 2007 – 352с.
3. В.В. Волгин. Справочник по диагностике неисправностей автомобилей. Атласы автомобилей, 2000. – 96с.
4. Диагностические коды неисправностей 2009. Легион-Автодата. – 1312с.
5. Інформаційні технології. Взаємозв'язок відкритих систем. Базова еталонна модель. Частина 1. Базова модель (ISO/IEC 7498-1:1994, IDT):ДСТУ ISO/IEC 7498-1:2004 – [Чинний від 2006–01–01]. – Київ: Держспоживстандарт України, 2007. – 67 с. – (Національний стандарт України).
6. Карманов И.Н. Измерения, испытания, контроль. Метрология и метрологическое обеспечение: учеб. пособ. / И.Н. Карманов, Н.А. Мещеряков, О.К. Ушаков. – Новосибирск: СГГА, 2006. – 184 с.
7. Каспина Т.В. Экономика и управление приборостроительным производством: учебн. пособ. для высших учебных заведений / Т.В. Каспина, Н.Н. Лямина. – М.: ИЦ "Академия", 2008. – 240 с.
8. Ключев В.В. Неразрушающий контроль и диагностика. Справочник, 2-е изд., перераб. и доп. / В.В. Ключев – М: Машиностроение, 2003. – 656 с.
9. Ключня В.Л. Основы экономической теории / В.Л. Ключня, Н.В. Черченко. – Минск: Минск, 2006. – 238 с.
10. Коваленко А.С. Разработка структуры базы данных интегрированной информационной системы / А.С. Коваленко, А.В. Коваленко // Информационные технологии и защита информации в информационно-коммуникационных

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

системах: монографія / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2015. – С. 54-64.

11. Кожанова А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. Кожанова, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2013. – Вип. 6(113). – С. 255-257.

12. Коваленко А.С. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко ., А.А. Смірнов, А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2014. – Вип. 4(120). – С. 161-164.

13. Коваленко А.С. Підсистема технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко , О.А.Смірнов, О.В. Коваленко // Системи озброєння і військова техніка.– Х.: ХУПС, 2014. – № 1(37). – С. 126-129.

14. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.

15. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

16. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

17. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.

18. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – С. 85-88.

19. Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 1(126). – С. 75-79.

20. Коваленко А.С. Обґрунтування набору даних для оцінки технічного стану інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2015. – Вип. 1(42). – С.39-41.

21. Коваленко А.С. Експертна система технічного діагностування інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2015. – № 1(41). – С. 106-111.

22. Коваленко А.С. Удосконалення методу технічного обслуговування об'єктів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко, О.П. Доренський // Системи озброєння і військова техніка. – Х.: ХУПС, 2016. – № 2(46). – С. 109-114.

23. Коваленко А.С. Метод визначення оптимального комплексу робіт з відновлення працездатності інтегрованої системи технічної діагностики в умовах ресурсних обмежень / А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2016. – Вип. 3(140). – С. 69-72.

24. Kovalenko A.S. Information model and its element for displaying information on technical condition of objects of integrated information system / A.S. Kovalenko, A.A. Smirnov, A.V. Kovalenko, A.P. Dorensky // International Journal of Computational Engineering Research (IJCER). – India: Delhi, 2016. – Volume 6, Issue

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

1. – Р. 21-27.

25. Кожанова А.С. Система технічної діагностики інтегрованих інформаційних систем – обґрунтування необхідності створення, визначення понятійного апарату та напрямів досліджень / А.С. Кожанова, О.А. Смірнов, М.П. Савченко, Д.М. Ізосімов, В.В. Мороз // Створення та модернізація озброєння і військової техніки в сучасних умовах: Тринадцята наук.-техн. конф., 5-6 вер. 2013 р., м. Феодосія: тези доп. – Феодосія: ДНВЦ, 2013. – С. 187-188.

26. Кожанова А.С. Визначення основних напрямків досліджень щодо створення системи технічної діагностики інтегрованих інформаційних систем / А.С. Кожанова, О.А. Смірнов, А.В. Челпанов // Проблемні питання розвитку озброєння та військової техніки Збройних Сил України: IV наук.-техн. конф., 16-20 груд. 2013 р., м. Київ: зб. тез. – Київ: ЦНДІ ОВТ ЗСУ, 2013. – С. 293.

27. Коваленко А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформатика та системні науки : V Всеукр. наук.-практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 2014. – С. 292-294.

28. Коваленко А.С. Задачи распознавания ситуаций в системах организационной стратегии интеграции производства и операций / А.С. Коваленко, А.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVI міжнар. наук.-практ. сем., 11-12 квіт. 2014 р., м. Кіровоград: зб. тез. – Кіровоград: КНТУ, 2014. – С. 53-55.

29. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2014. – С. 241.

30. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп'ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 28-31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 2014. – С. 190-193.

31. Коваленко А.С. Основні складові та функції системи технічної діагностики інтегрованих інформаційних систем / Коваленко А.С. // Інформаційні технології та комп'ютерна інженерія: наук.-практ. конф., 4 груд. 2014 р., м. Кіровоград: зб. тез доп. – Кіровоград: КНТУ, 2014. – С. 236.

32. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2015. – С. 15.

33. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

34. Коваленко А.С. Метод автоматизованої перевірки результатів вимірювання параметрів об'єкті в інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Стратегія якості у промисловості і освіті: XI міжнар. конф., 1 – 5 черв. 2015 р., м. Варна, Болгарія.: зб. матер. – Варна: ТУВ, 2015. – С. 423-426.

35. Коваленко А.С. Обґрунтування необхідності створення розподіленої бази даних для забезпечення захисту рухомих повітряних об'єктів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Перспективні напрями захисту інформації: I всеукр. наук.-практ. конф., 07 вер. 2015 р., м. Одеса: зб. тез доп. – Одеса: ОНАЗ, 2015. – С. 35-39.

36. Коваленко А.С. Розробка інформаційної моделі автоматизованої оцінки технічного стану інтегральної інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформаційні технології та взаємодії (ІТ & І): II

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

між нар. наук.-практ. конф., 3-5 лист. 2015 р., м. Київ: тези доп. – Київ: КНУ ім. Т. Шевченка, 2015. – С. 41-42.

37. Коваленко А.С. Разработка метода усовершенствования технического обслуживания интегрированной информационной системы / А.С. Коваленко, О.А. Смирнов, О.В. Коваленко // Информационные и телекоммуникационные технологии: образование, наука, практика: II междунар. научн.-практ. конф., 3-4 дек. 2015 г., г. Алматы, Казахстан: сб. труд. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – Т.2. – С. 423-427.

38. Королюк Н.А. Оценка временных интервалов работы лица, принимающего решение, на автоматизированном командном пункте / Н.А. Королюк, А.И. Тимочко // Системи обробки інформації. – Х.: ХУПС, 2005. – Вип. 8 (48). – С. 51-54.

39. Костерев В.В. Надёжность технических систем и управление риском: учебн. пособ. / В.В. Костерев. – М.: МИФИ, 2008. – 280 с.

40. Костюков А.В. Підвищення операційної ефективності підприємств на основі моніторингу в реальному часі. / А.В. Костюков, В.М. Костюков. – М.: Машинобудування, 2009. – 192 с.

41. Лазарев А.А. Выбор показателя затрат для анализа сравнительной экономической эффективности техники конечного потребления / А.А. Лазарев, М.В. Бейлин // Сборник научных трудов ХГПУ. – Х.: ХГПУ, 1999. – Вып. 74. – С. 27-29.

42. Ланецкий Б.Н. Основы теории надежности, технического обслуживания и ремонта вооружения и военной техники: Справочные материалы, часть 1. / Б.Н. Ланецкий, А.А. Посудевский. – Харьков: ХВУ, 1993. – 308 с.

43. Ланецкий Б.Н. Основы теории надежности, технического обслуживания и ремонта вооружения и военной техники: Справочные материалы, часть 2. / Б.Н. Ланецкий, А.А. Посудевский. – Харьков: ХВУ, 1993. – 208 с.

44. Лапсарь А.П. Метод оценки состояния сложных технических объектов для синтеза быстродействующих прогнозирующих систем / А.П.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Лапсарь // Измерительная техника. – 2004. – № 2. – С. 7-10.

45. Линейные задачи оптимизации: Учеб. пособие / С.В. Лутманов. – Пермь: ЛИТЕР-А, 2004. – Ч.1. – Линейное программирование. – 128 с.

46. Литвак Б.Г. Экспертные технологии в управлении. Учебное пособие / Б.Г. Литвак. – М.: Дело, 2014. – 318 с.

47. Локазюк В.М. Надійність, контроль, діагностика і модернізація ПК: Посібн. / В.М. Локазюк, Ю.Г. Савченко. – К.: Видавничий центр «Академія», 2004. – 376 с.

48. Лопатников Л. И. Экономико-математический словарь: Словарь современной экономической науки / Л.И. Лопатников. – М.: Дело, 2003. – 520 с.

49. Манухина С.Ю. Инженерная психология и эргономика: хрестоматия / С.Ю Манухина. – М.: Изд. центр ЕАОИ, 2009. –224 с.

50. Мартыненко М.В. Человекомашинные процедуры поддержки организационно–управленческих решений: учеб. пособие СПбГЭТУ / М.В. Мартыненко, О.И. Шеховцов. – СПб, 2012. – 250 с.

					ВКРБ-123.23.0015.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1	Найменування та область застосування.....	2
2	Підстава для розробки.....	2
3	Мета та призначення розробки.....	2
4	Джерела розробки.....	2
5	Технічні вимоги.....	2
5.1	Вміст проекту.....	2
5.2	Показники призначення.....	3
5.3	Вимоги до функціональних характеристик.....	3
5.4	Вимоги до архітектури.....	3
5.5	Вимоги до надійності.....	3
5.6	Умови експлуатації.....	4
5.7	Вимоги до складу та параметрів технічних засобів.....	4
5.8	Вимоги до інформаційної і програмної сумісності.....	4
5.8.1	Обладнання.....	4
5.8.2	Мова програмування.....	4
5.8.3	Вхідні дані.....	5
5.8.4	Вихідні дані.....	5
6	Вимоги до програмної документації.....	5
7	Перелік документів, що розробляються.....	5
8	Етапи розробки.....	6
9	Порядок контролю та приймання.....	6

					ВКРБ-123.23.0015.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Марченко М.С.				<i>Програмне забезпечення системи діагностування електронної системи автомобіля за стандартом ISO 14230</i>	Літ.	Аркуш	Аркушів
Перевірів	Дресв О.М.					Б	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КМ-19			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи діагностування електронної системи автомобіля за стандартом ISO 14230.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 10-02 від 5.01.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи діагностування електронної системи автомобіля за стандартом ISO 14230.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

					ВКРБ-123.23.0015.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- системи діагностування електронної системи автомобіля за стандартом ISO 14230;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРБ-123.23.0015.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище RAD Studio Delphi 10.4.

					ВКРБ-123.23.0015.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		4

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 78 аркушів.

8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

					ВКРБ-123.23.0015.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2023 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 8.06.2023 р.

					ВКРБ-123.23.0015.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
першим (бакалаврським) рівнем вищої освіти

_____ Дреєв О.М.

*Програмне забезпечення системи діагностування електронної системи
автомобіля за стандартом ISO 14230*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 78

Літера: РП

Кропивницький – 2023 року

Diag_ISO_14230.dpr - головний файл проекту

```
program Diag_ISO_14230_;

uses
  Forms,
  umain in 'umain.pas' {FMain},
  common in 'common.pas',
  CPort in 'CPort.pas',
  CPortCtl in 'CPortCtl.pas',
  CPortEsc in 'CPortEsc.pas',
  _ISO_14230_ in '_ISO_14230.pas',
  ubipwm in 'ubipwm.pas' {FBiPWM},
  uwiper in 'uwiper.pas' {FWiper},
  ubridge in 'ubridge.pas' {FBridge},
  ulist in 'ulist.pas' {FList},
  unit_ajoutboitier in 'unit_ajoutboitier.pas' {F_AjoutBoitier},
  unit_configuration in 'unit_configuration.pas' {F_Configuration},
  unit_connection in 'unit_connection.pas' {F_Connection},
  uonoff in 'uonoff.pas' {FOnOff},
  upwm in 'upwm.pas' {FPwm},
  about in 'about.pas' {Fabout},
  uvalue in 'uvalue.pas' {FValue};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TFMain, FMain);
  Application.CreateForm(TFBiPWM, FBiPWM);
  Application.CreateForm(TFWiper, FWiper);
  Application.CreateForm(TFBridge, FBridge);
  Application.CreateForm(TFList, FList);
  Application.CreateForm(TF_AjoutBoitier, F_AjoutBoitier);
  Application.CreateForm(TF_Configuration, F_Configuration);
  Application.CreateForm(TF_Connection, F_Connection);
  Application.CreateForm(TFOnOff, FOnOff);
  Application.CreateForm(TFPwm, FPwm);
  Application.CreateForm(Tabout, Fabout);
  Application.CreateForm(TFValue, FValue);
  Application.Run;
end.
```

umain.pas - основна програма

```

unit umain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, Grids, ComCtrls, unit_connection, CPort,
  common, Buttons, _ISO_14230_, unit_Configuration,
  DateUtils, ImgList, ExtCtrls, StdCtrls, XPMAN, about;
{
uses
  Classes, SysUtils, LResources, Forms, Controls, Graphics, Dialogs, Menus,
  StdCtrls, ExtCtrls, Grids, ComCtrls, unit_connection, CPort,
  common, Buttons, _ISO_14230_, unit_Configuration, EditBtn, u_open, u_save,
  DateUtils, LCLType;
}

const slash = '\';

      Etat_Debut_Connection = 1;
      Etat_Connecte=2;
      Etat_Attente_Reponse =3;

      CT_SensHL = False;
      CT_SensLH = True;

type

  { TFMain }

  TFMain = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    Group_Debug: TGroupBox;
    Image1: TImage;
    ImageList1: TImageList;
    ImageList2: TImageList;
    I_Attente: TImage;
    MainMenu1: TMainMenu;
    Menu_Langues: TMenuItem;
    Menu_Quitter: TMenuItem;
    Menu_Setting: TMenuItem;
    M_Debug: TMemo;
    OpenDialog1: TOpenDialog;
    Panel1: TPanel;
    Panel2: TPanel;
    SaveDialog1: TSaveDialog;
    Splitter1: TSplitter;
    Menu_Connection: TMenuItem;
    T_Attente: TTimer;
    T_StatusIO: TTimer;
    TreeView1: TTreeView;
    IList_Rose: TImageList;
    IListConnect: TImageList;
    IList_Led: TImageList;
    Panel_Status: TPanel;
    L_Titre: TLabel;
    M_Status: TMemo;
    Panel_Command: TPanel;
    GroupBox4: TGroupBox;
    Label3: TLabel;
    Label4: TLabel;

```

```

E_Sid: TEdit;
E_Data: TEdit;
B_Send: TBitBtn;
M_Hist: TMemo;
Panel_Grille: TPanel;
StringGrid1: TStringGrid;
Panel_Grid_Cde: TPanel;
B_Mode_Maitre: TBitBtn;
B_SaveGrid: TBitBtn;
Panel_Liste_Para: TPanel;
Label1: TLabel;
Label2: TLabel;
E_Fichier_Para: TEdit;
B_Rech_Fichier_Para: TButton;
B_Lecture_Liste: TButton;
B_Sauve_Liste: TButton;
GroupBox3: TGroupBox;
StringGrid2: TStringGrid;
B_Export: TButton;
Panel_Parametre: TPanel;
L_Parametre: TLabel;
L_Numero: TLabel;
L_Para_Valeur: TLabel;
B_Para_Lecture: TButton;
B_Para_Ecriture: TButton;
E_Para_Numero: TEdit;
M_Parametre: TMemo;
E_Para: TEdit;
RadioAffVal: TRadioGroup;
Panel_Value: TPanel;
L_Valeur: TLabel;
RadioAffVal1: TRadioGroup;
E_Valeur: TEdit;
M_Valeur: TMemo;
B_Valeur_Ecriture: TButton;
B_Valeur_Lecture: TButton;
XPManifest1: TXPManifest;
T_Command: TTimer;
//   procedure B_DownMouseDown(Sender: TObject; Button: TMouseButton;
//   Shift: TShiftState; X, Y: Integer);
procedure B_ExportClick(Sender: TObject);
procedure B_Lecture_ListeClick(Sender: TObject);
procedure B_Mode_MaitreClick(Sender: TObject);
procedure B_Para_EcritureClick(Sender: TObject);
procedure B_Para_LectureClick(Sender: TObject);
procedure B_Rech_Fichier_ParaClick(Sender: TObject);
procedure B_Sauve_ListeClick(Sender: TObject);
procedure B_SaveGridClick(Sender: TObject);
procedure B_SendClick(Sender: TObject);
//   procedure B_UpMouseDown(Sender: TObject; Button: TMouseButton;
//   Shift: TShiftState; X, Y: Integer);
procedure B_Valeur_EcritureClick(Sender: TObject);
procedure B_Valeur_LectureClick(Sender: TObject);
procedure E_ParaChange(Sender: TObject);
procedure FormClose(Sender: TObject; var CloseAction: TCloseAction);
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);

procedure Menu_ConnectionClick(Sender: TObject);
procedure Menu_QuitterClick(Sender: TObject);
procedure Menu_SettingClick(Sender: TObject);
procedure Panell1Click(Sender: TObject);
procedure RadioAffVal1ChangeBounds(Sender: TObject);
procedure RadioAffValChangeBounds(Sender: TObject);
procedure StringGrid1Click(Sender: TObject);
procedure StringGrid1DrawCell(Sender: TObject; aCol, aRow: Integer;
  aRect: TRect; aState: TGridDrawState);
procedure StringGrid1Resize(Sender: TObject);
procedure T_AttenteTimer(Sender: TObject);

```

```

procedure T_StatusIOTimer(Sender: TObject);
procedure TreeView1Click(Sender: TObject);
procedure Menu_langueAddClick(Sender: TObject);
procedure T_CommandTimer(Sender: TObject);
private
  { private declarations }
  num_image : integer;
  diag_actif : boolean;
  ledverteOn,ledVerteOFF,ledRougeOn,ledamberon,ledamberOff,ledRougeOff :
TBitmap;
  cpt_erreur : integer;
  Grille_Up,nb_Row_visible : integer; // Дозволяє дізнатися діапазон видимих
входів та виходів

  procedure _ISO_14230_Connect(Sender: TObject);
  procedure _ISO_14230_Disconnect(Sender: TObject);
  procedure _ISO_14230_Error(Sender: TObject; Erreur: integer; MsgErreur:
string
  );
  procedure _ISO_14230_ReceiveTrame(Sender: TObject);
  procedure _ISO_14230_Status(Sender: TObject; Messages: string);
  procedure _ISO_14230_Step(Sender: TObject; Messages: string);
public
  { public declarations }
  MainExit : boolean;
  BeginExit : Boolean;
  Variable : array[0..9] of integer;
  Table : array[0..9] of TStringList;
  Type_Connection: integer;
  Debug : integer;
  State : integer;
  TypeAff : integer; // Цей тип визначає видимість на екрані: 0->StringGrid
Vehicule : TStringList;
  MenuList : TStringList;
  ActualMenu : TStringList;
  ExtendedMenu : TStringList;
  ClickList : TStringList;
  ligneMenu : integer;
  Actual_ISO_14230_ : TStringList;
  ColorList : TStringList;
  Execution : boolean;
  TimeRepeat : integer;
  MenuName : string;
  TitleMenu : string;
  Fichier : TStringlist;
  nom_fichier : string;
  CdeReadParam,CdeWriteParam : string;
  CdeValueLength : string;
  CdeValueAction : integer;
  ClickTree : boolean; // при натисканні на дереві, ви не робите якісь речі
TypeCdeActif : integer;
  LigneGrid2 : integer;
  ClickActif : Boolean;
  ClickGlobal : string; // відправлення команди, чи слід активізувати активну
діагностику
  ClickExit : string; // відправлення команди, чи не слід активізувати активну
діагностику
  NoReceive : boolean;
  InIOTimer : boolean;
  ComPort1: TComPort;
  _ISO_14230_ : T_ISO_14230_;
  procedure GlobalReset;
  procedure Search_Item(ItemSearch : string;ListSearch : TStringList;var
searchList: TStringList);
  Procedure Attente_Start;
  Procedure Attente_Stop;
  Function Cde2Send(CDE: string): integer;
  Procedure Create_Menu_Langue;
  Procedure Init_Message;

```



```

        FMain.ISO_14230.DiagnosticSpeed:=Strtoint(texte);
    except
        FMain.ISO_14230.DiagnosticSpeed:=10400;
    end;
end else
    if uppercase(copy(texte,1,8))='SESSION=' then
    begin
        delete(texte,1,8);
        type_diagnosticssession:=hextoDec(texte);
    end else
        if uppercase(copy(texte,1,14))='TESTERPRESENT=' then
        begin
            delete(texte,1,14);
            FMain.ISO_14230.Msg_TesterPresent:=texte;
        end;
    end;
// Меню діагностики ITEM
MenuList.Clear;
    FMain.ExtendedMenu.Clear;
    FMain.Treeview1.Items.Clear;
    Noeud:=FMain.TreeView1.Items.Add(nil, (MenuProg[CMF_Status]));
    Noeud.ImageIndex:=5;
    Noeud:=FMain.TreeView1.Items.Add(nil, (MenuProg[CMF_Command]));
    Noeud.ImageIndex:=1;

    FMain.Panel_grille.Visible:=False;
    FMain.Panel_Parametre.Visible:=False;
    FMain.Panel_Status.Visible:=True;
    MenuI:=1;
    Sous_MenuI:=0;
    FMain.ComPort1.Port:=F_Connection.Combo_Port.Text;
    FMain.M_Status.Clear;
    DateTimeToString(texte, 'h-nn-ss-zzz', Now);
    FMain.M_Status.Lines.Add(texte+(MenuProg[CMF_TestConnect]));
    state:=1; // Спроба підключення до COM-порту
    FMain.Attente_Start;
    FMain.ISO_14230.Connect;
    FMain.Attente_Stop;
    if State<>0 then
    begin
        // З'єднання з COM-портом встановлено
        State:=2;
        FMain.Attente_Start;
        FMain.M_Debug.Lines.Add('Lancement StartDiagnosis');
        FMain.ISO_14230.StartDiagnosis(type_diagnosticssession);
    end;
end;
end else
begin
    FMain.GlobalReset;
end;
finally
    ListConnection.Free;
end;
end;
end;
//*****
//Закриття форми
//*****
procedure TFMain.FormClose(Sender: TObject; var CloseAction: TCloseAction);
begin
    MainExit:=True;
    FMain.Execution:=True;
    FMain.T_StatusIO.Enabled:=False;
    //if FMain.ComPort1.Connected then FMain.ComPort1.Close;
end;

//*****
procedure TFMain.B_Para_LectureClick(Sender: TObject);

```

```

var n_para : integer;
    texte : string;
begin
    try
        n_para:=hexdec (uppercase (FMain.E_Para_Numero.Text));
    except

Application.MessageBox (PCHAR (MenuProg [CMP_NumberNotHex] ), PCHAR (MenuProg [CMP_Error]), 0);
        exit;
    end;
    if n_para<=255 then
        begin
            texte:=FMain.CdeReadParam;
            delete (texte, 1, 1);
            if length (texte)=0 then exit;
            FMain.Cde2Send (texte);
            FMain.ISO_14230.FData [FMain.ISO_14230.SendLength]:=n_para;
            FMain.ISO_14230.SendLength:=FMain.ISO_14230.SendLength+1;
            // Конфігурація системи
            FMain.CdeValueAction:=1;
            FMain.ISO_14230.SendResponse;
            FMain.Attente_Start;
        end else
Application.MessageBox (PCHAR (MenuProg [CMP_ParamOutOfRange] ), PCHAR (MenuProg [CMP_Error]), 0);
        end;
end;

//*****
procedure TFMain.B_Rech_Fichier_ParaClick (Sender: TObject);
var texte : string;
    i : integer;
    erreur : integer;
begin
    erreur:=1;
    //if FOpen.Execute then
    FMain.OpenDialog1.Filter:='Fichier XML|*.xml';
    if FMain.OpenDialog1.Execute then
        begin
            //FMain.E_Fichier_Para.Text:=FOpen.FileName;
            FMain.E_Fichier_Para.Text:=FMain.OpenDialog1.FileName;
            AssignFile (fichier_txt, FMain.E_Fichier_Para.Text);
            reset (fichier_txt);
            FMain.StringGrid2.RowCount:=1;
            FMain.StringGrid2.Cells [0, 0]:=MenuProg [CMP_HexaAdress];
            FMain.StringGrid2.Cells [1, 0]:=MenuProg [CMP_Name];
            FMain.StringGrid2.Cells [2, 0]:=MenuProg [CMP_PhysicalAdress];
            FMain.StringGrid2.Cells [3, 0]:=MenuProg [CMP_Length];
            FMain.StringGrid2.Cells [4, 0]:=MenuProg [CMP_Position];
            FMain.StringGrid2.Cells [5, 0]:=MenuProg [CMP_Fault];
            FMain.StringGrid2.Cells [6, 0]:=MenuProg [CMP_Min];
            FMain.StringGrid2.Cells [7, 0]:=MenuProg [CMP_Max];
            FMain.StringGrid2.Cells [8, 0]:=MenuProg [CMP_DecimalValue];
            For i:=0 to 8 do
                FMain.StringGrid2.ColWidths [i]:=FMain.StringGrid2.Canvas.TextWidth (FMain.StringGrid2.Cells [i, 0]) + 10;
                while eof (fichier_txt)=false do
                    begin
                        readln (fichier_txt, texte);
                        i:=pos (' PARM ID', texte);
                        if i>0 then
                            begin
                                erreur:=0;
                                delete (texte, 1, i+9);
                                i:=pos ('', texte);
                                FMain.StringGrid2.RowCount:=FMain.StringGrid2.RowCount+1;
                                FMain.StringGrid2.Cells [0, FMain.StringGrid2.RowCount-1]:=copy (texte, 1, i-1);

```

```

        delete (texte, 1, i);
        i:=pos ('"', texte);
        FMain.StringGrid2.Cells[1, FMain.StringGrid2.RowCount-
1]:=copy (texte, 1, i-1);
        delete (texte, 1, i);
        i:=pos ('Адреса=', texte);
        delete (texte, 1, i+8);
        i:=pos ('"', texte);
        FMain.StringGrid2.Cells[2, FMain.StringGrid2.RowCount-
1]:=copy (texte, 1, i-1);
        delete (texte, 1, i);
        i:=pos ('Позмип=', texte);
        delete (texte, 1, i+5);
        i:=pos ('"', texte);
        FMain.StringGrid2.Cells[3, FMain.StringGrid2.RowCount-
1]:=copy (texte, 1, i-1);
        delete (texte, 1, i);
        i:=pos ('Позиція=', texte);
        delete (texte, 1, i+9);
        i:=pos ('"', texte);
        FMain.StringGrid2.Cells[4, FMain.StringGrid2.RowCount-
1]:=copy (texte, 1, i-1);
        delete (texte, 1, i);
        i:=pos ('Інше=', texte);
        delete (texte, 1, i+8);
        i:=pos ('"', texte);
        FMain.StringGrid2.Cells[5, FMain.StringGrid2.RowCount-
1]:=copy (texte, 1, i-1);
        delete (texte, 1, i);
        i:=pos ('Мінімум =', texte);
        delete (texte, 1, i+4);
        i:=pos ('"', texte);
        FMain.StringGrid2.Cells[6, FMain.StringGrid2.RowCount-
1]:=copy (texte, 1, i-1);
        delete (texte, 1, i);
        i:=pos ('Максимум =', texte);
        delete (texte, 1, i+4);
        i:=pos ('"', texte);
        FMain.StringGrid2.Cells[7, FMain.StringGrid2.RowCount-
1]:=copy (texte, 1, i-1);
        delete (texte, 1, i);
        i:=pos ('Значення =', texte);
        delete (texte, 1, i+6);
        i:=pos ('"', texte);
        FMain.StringGrid2.Cells[8, FMain.StringGrid2.RowCount-
1]:=copy (texte, 1, i-1);
        delete (texte, 1, i);
        end;
        For i:=0 to 8 do
        begin
            if
(FMain.StringGrid2.ColWidths[i]<FMain.StringGrid2.Canvas.TextWidth(FMain.StringG
rid2.Cells[i, FMain.StringGrid2.RowCount-1])+10)
                then
(FMain.StringGrid2.ColWidths[i]:=FMain.StringGrid2.Canvas.TextWidth(FMain.StringG
rid2.Cells[i, FMain.StringGrid2.RowCount-1])+10);
            end;
        end;
        if FMain.StringGrid2.RowCount>1 then FMain.StringGrid2.FixedRows:=1;
        CloseFile(fichier_txt);
        if erreur=1 then
        begin
Application.MessageBox (PCHAR (MenuProg [CMP_WrongFile]), PCHAR (MenuProg [CMP_Error])
, 0);
            exit;
        end;
    end;
end;
end;

```

```

//*****
procedure TFMMain.B_Sauve_ListeClick(Sender: TObject);
var n_para, val_para : integer;
    texte : string;
    k : integer;
    l, lg : integer; // використовується для значення вмісту октету
begin
    if FMain.StringGrid2.RowCount<2 then exit;
    FMain.TypeCdeActif:=2;
    try
        n_para:=hexdec (uppercase (FMain.StringGrid2.Cells[0,1]));
    except

Application.MessageBox (PCHAR (MenuProg [CMP_NumberNotHex] ), PCHAR (MenuProg [CMP_Erro
r]), 0);
        exit;
    end;
    try
        val_para:=strtoint (FMain.StringGrid2.Cells[8,1]);
    except

Application.MessageBox (PCHAR (MenuProg [CMP_ValueNotHex] ), PCHAR (MenuProg [CMP_Error
]), 0);
        exit;
    end;
    if n_para<=255 then
        begin
            texte:=FMain.CdeWriteParam;
            if texte[1]='L' then l:=0 else l:=1;
            delete (texte, 1, 1);
            FMain.Cde2Send (texte);
            FMain.ISO_14230.FData [FMain.ISO_14230.SendLength]:=n_para;
            FMain.ISO_14230.SendLength:=FMain.ISO_14230.SendLength+1;
            // записати значення в буфер
            try
                texte:=dectohex (val_para);
            except
                texte:='00';
            end;
            if (FMain.StringGrid2.Cells[3,1]='16')
                and (length (texte)=2)
                then texte:='00'+texte;
            if (l=0) and (length (texte)>2) then
                begin
                    texte:=copy (texte, 3, 2)+copy (texte, 1, 2);
                end;
            lg:=(length (texte) div 2);
            for k:=0 to lg-1 do
                begin
                    FMain.ISO_14230.FData [k+FMain.ISO_14230.SendLength]:=hexdec (copy (texte, 1, 2));
                    delete (texte, 1, 2);
                end;

                FMain.ISO_14230.SendLength:=FMain.ISO_14230.SendLength+lg;
                // Конфігурація системи
                //FMain.Timer_Maintient.Enabled:=False;
                FMain.LigneGrid2:=1;
                FMain.ISO_14230.SendResponse;
                FMain.Attente_Start;
            end else
Application.MessageBox (PCHAR (MenuProg [CMP_ParamOutOfRange] ), PCHAR (MenuProg [CMP_Err
or]), 0);
        end;

//*****
procedure TFMMain.B_SaveGridClick(Sender: TObject);
var i, j : integer;

```

```

    texte : string;
begin
  //FSave.RadioGroup1.Visible:=false;
  //FSave.RadioGroup1.ItemIndex:=1;
  FMain.OpenDialog1.Filter:='Fichier Excel (CSV)|*.csv';
  if FMain.OpenDialog1.execute then
    begin
      FMain.nom_fichier:='';
      Fichier.Clear;
      for i:=0 to FMain.StringGrid1.RowCount-1 do
        begin
          texte:='';
          for j:=0 to FMain.StringGrid1.ColCount-1 do
            texte:=texte+FMain.StringGrid1.Cells[j,i]+' ';
          Fichier.Add(texte);
        end;
      texte:=lowercase(FMain.OpenDialog1.FileName);
      i:=pos('.csv',texte);
      if i=0 then texte:=texte+'.csv';
      Fichier.SaveToFile(texte);
    end;
  //FSave.RadioGroup1.Visible:=True;
end;

//*****
procedure TFMain.B_SendClick(Sender: TObject);
var datetxt : string;
    texte : string;
    i : integer;

begin
  DateTimeToString(datetxt,'h : nn m ss sec zzz ms - ',Now);
  FMain.E_Sid.Text:=uppercase(FMain.E_Sid.Text);
  FMain.E_Data.Text:=uppercase(FMain.E_Data.Text);
  try
    FMain.ISO_14230.Sid:=hextohex(trim(FMain.E_Sid.Text));
  except
    FMain.M_Hist.Lines.Add(datetxt+MenuProg[CMPSidError]);
    exit;
  end;
  texte:=FMain.E_Data.Text;
  texte:=trim(texte);
  i:=0;
  while length(texte)>0 do
    begin
      FMain.ISO_14230.FData[i]:=hextohex(copy(texte,1,2));
      i:=i+1;
      delete(texte,1,2);
    end;
  FMain.ISO_14230.SendLength:=i;
  FMain.Execution:=False;
  texte:=dectohex(FMain.ISO_14230.Sid)+' ';
  for i:=0 to FMain.ISO_14230.SendLength-1 do
    texte:=texte+dectohex(FMain.ISO_14230.FData[i]+' ');
  FMain.M_Hist.Lines.Add(datetxt+MenuProg[CMPSend]+' - '+texte);
  FMain.Attente_Start;
  FMain.ISO_14230.SendResponse;
  While (Execution=False) and (MainExit=False) do Application.ProcessMessages;
  if MainExit then Exit;
  texte:=dectohex(FMain.ISO_14230.SidReceive)+' ';
  for i:=0 to FMain.ISO_14230.ReceiveLength-1 do
    texte:=texte+dectohex(FMain.ISO_14230.TrameRx[i]+' ');
  FMain.M_Hist.Lines.Add(datetxt+MenuProg[CMPSReceive]+' - '+texte);
end;

//*****
procedure TFMain.B_UpMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin

```

```

if FMain.Grille_Up>0 then
  begin
    FMain.Grille_Up:=FMain.Grille_Up-1;
    FMain.rafraichir_diag;
  end;
end;
}
//*****
procedure TFMain.B_Valeur_EcritureClick(Sender: TObject);
var val_para : integer;
    texte,textel : string;
    i,j : integer;
    d : array[0..20] of byte;
    nd : integer;
    sens_octet : integer;
    Nb_BitValue : integer;
begin
  if FMain.RadioAffVal.ItemIndex=0 then
    begin
      try
        val_para:=strtoint(FMain.E_Valeur.Text);
      except

Application.MessageBox(PCHAR(MenuProg[ CMP_ValueNotHex ]),PCHAR(MenuProg[ CMP_Error
]),0);
        exit;
      end;
    end else
    begin
      try
        val_para:=hexdec(uppercase(FMain.E_Valeur.Text));
      except

Application.MessageBox(PCHAR(MenuProg[ CMP_ValueNotHex ]),PCHAR(MenuProg[ CMP_Error
]),0);
        exit;
      end;
    end;
    texte:=FMain.CdeWriteParam;
    if texte[1]='L' then sens_octet:=0 else sens_octet:=1;
    delete(texte,1,1);
    if length(texte)=0 then exit;
    Fmain.Cde2Send(texte);
    j:=FMain.ISO_14230.SendLength;
    texte:=dectohex(val_para);
    if Length(CdeValueLength)>0 then
      begin
        try
          Nb_BitValue:=strtoint(CdeValueLength);
          while length(texte)<(Nb_BitValue div 4) do texte:='00'+texte;
        except
          end;
        end;
        nd:=0;
        while length(texte)>0 do
          begin
            textel:=copy(texte,1,2);
            d[nd]:=hexdec(textel);
            nd:=nd+1;
            delete(texte,1,2);
          end;
        FMain.ISO_14230.SendLength:=FMain.ISO_14230.SendLength+nd;
        //j:=2;
        if sens_octet=0 then
          begin
            for i:=nd-1 downto 0 do
              begin
                FMain.ISO_14230.FData[j]:=d[i];
                j:=j+1;

```

```

        end;
    end else
    begin
        for i:=0 to nd-1 do
            begin
                FMain.ISO_14230.FData[j]:=d[i];
                j:=j+1;
            end;
        end;
        // Конфігурація системи
        //FMain.Timer_Maintient.Enabled:=False;
        FMain.CdeValueAction:=2;
        FMain.ISO_14230.SendResponse;
        FMain.Attente_Start;
    end;

//*****
procedure TFMMain.B_Valeur_LectureClick(Sender: TObject);
var texte : string;
begin
    texte:=FMain.CdeReadParam;
    delete(texte,1,1);
    if length(texte)=0 then exit;
    FMain.CdeValueAction:=1;
    FMain.Cde2Send(texte);
    // Конфігурація системи
    FMain.ISO_14230.SendResponse;
    FMain.Attente_Start;
end;

//*****
procedure TFMMain.E_ParaChange(Sender: TObject);
begin

end;

//*****
procedure TFMMain.B_Para_EcritureClick(Sender: TObject);
var n_para,val_para : integer;
    texte,textel : string;
    i,j : integer;
    d : array[0..20] of byte;
    nd : integer;
    sens_octet : integer;
begin
    try
        n_para:=hexdec(uppercase(FMain.E_Para_Numero.Text));
    except

Application.MessageBox(PCHAR(MenuProg[ CMP_NumberNotHex ]),PCHAR(MenuProg[ CMP_Error ]),0);
        exit;
    end;
    if FMain.RadioAffVal.ItemIndex=0 then
        begin
            try
                val_para:=strtoint(FMain.E_Para.Text);
            except

Application.MessageBox(PCHAR(MenuProg[ CMP_ValueNotHex ]),PCHAR(MenuProg[ CMP_Error ]),0);
                exit;
            end;
        end else
        begin
            try
                val_para:=hexdec(uppercase(FMain.E_Para.Text));
            except

```



```

        if i<1 then
FMain.SaveDialog1.FileName:=FMain.SaveDialog1.FileName+'.xml';
        end;
        2 : begin
            i:=pos('.csv',lowercase(texte));
            if i<1 then
FMain.SaveDialog1.FileName:=FMain.SaveDialog1.FileName+'.csv';
            end;
        end;
        if pos('.xml',lowercase(FMain.SaveDialog1.FileName))<1 then
        begin
            AssignFile(fichier_txt,FMain.SaveDialog1.FileName);
            Rewrite(fichier_txt);
            Writeln(fichier_txt,'Таблиця конфігурації'+dateTimeToStr(Now));
            Writeln(fichier_txt,'');
            for i:=0 to FMain.StringGrid2.RowCount-1 do
                begin
                    texte:='';
                    for j:=0 to FMain.StringGrid2.ColCount-1 do
                        begin
                            texte:=texte+FMain.StringGrid2.Cells[j,i]+' ';
                        end;
                    writeln(fichier_txt,texte);
                end;
            Closefile(fichier_txt);
        end else
        begin // sauve xml
            AssignFile(fichier_txt,FMain.SaveDialog1.FileName);
            Rewrite(fichier_txt);
            Writeln(fichier_txt,'<?xml:lang="ru" version="1.0"
standalone="no"?>');
            Writeln(fichier_txt,'<!DOCTYPE ParamEEPROMDescr SYSTEM
"ParamDesc.dtd">');
            Writeln(fichier_txt,'');
            Writeln(fichier_txt,'<ParamEEPROMDescr>');
            Writeln(fichier_txt,'');
            Writeln(fichier_txt,'    <ParamEEPROM-List>');
            Writeln(fichier_txt,'');
            for i:=1 to FMain.StringGrid2.RowCount-1 do
                begin
                    texte:='        <ParamEEPROM-PARM ID="['
+FMain.StringGrid2.Cells[0,i]
+'] '
+FMain.StringGrid2.Cells[1,i]
+' " ADDRESS="'
+FMain.StringGrid2.Cells[2,i]
+' " SIZE="'
+FMain.StringGrid2.Cells[3,i]
+' " POSITION="'
+FMain.StringGrid2.Cells[4,i]
+' " DEFAULT="'
+FMain.StringGrid2.Cells[5,i]
+' " MIN="'
+FMain.StringGrid2.Cells[6,i]
+' " MAX="'
+FMain.StringGrid2.Cells[7,i]
+' " VALUE="'
+FMain.StringGrid2.Cells[8,i]
+' " />'
;

                    writeln(fichier_txt,texte);
                end;
            Writeln(fichier_txt,'');
            Writeln(fichier_txt,'    </ParamEEPROM-List>');
            Writeln(fichier_txt,'');
            Writeln(fichier_txt,'</ParamEEPROMDescr>');
            Writeln(fichier_txt,'');
            Closefile(fichier_txt);
        end;
    end;
end;

```

```

        end;
    end;
end;

{
procedure TFMMain.B_DownMouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
begin
    if FMain.Grille_Up+FMain.nb_Row_visible<FMain.StringGrid3.RowCount then
    begin
        FMain.Grille_Up:=FMain.Grille_Up+1;
        FMain.rafraichir_diag;
    end;
end;
}
//*****
procedure TFMMain.B_Lecture_ListeClick(Sender: TObject);
var n_para : integer;
    texte : string;
begin
    if FMain.StringGrid2.RowCount<2 then exit;
    try
        n_para:=hexdec (uppercase (FMain.StringGrid2.Cells[0,1]));
    except

Application.MessageBox (PCHAR (MenuProg [CMP_NumberNotHex]), PCHAR (MenuProg [CMP_Erro
r]), 0);
        exit;
    end;
    if n_para<=255 then
    begin
        texte:=FMain.CdeReadParam;
        delete (texte, 1, 1);
        FMain.Cde2Send (texte);
        FMain.ISO_14230.FData [FMain.ISO_14230.SendLength]:=n_para;
        FMain.ISO_14230.SendLength:=FMain.ISO_14230.SendLength+1;

        // Конфігурація системи
        //FMain.Timer_Maintient.Enabled:=False;
        FMain.LigneGrid2:=1;
        FMain.TypeCdeActif:=1;
        FMain.ISO_14230.SendResponse;
        FMain.Attente_Start;
    end;
end;

//*****
procedure TFMMain.B_Mode_MaitreClick(Sender: TObject);
begin
    while (FMain.ISO_14230.EndReceive=false)
    and (MainExit=False) do application.ProcessMessages;
    if MainExit=True then Exit;

    FMain.Cde2Send (FMain.ClickExit);
    FMain.Execution:=False;
    FMain.NoReceive:=True;
    FMain.ISO_14230.SendResponse;
    FMain.Attente_Start;
    FMain.B_Mode_Maitre.Visible:=False;
    FMain.ClickActif:=False;
    if FBiPWM.Actif then FBiPwm.Close;
    if FPwm.Actif then FPWM.Close;
    if FBridge.Actif then FBridge.Close;
    if FOnOff.Actif then FOnOff.Close;
    if FValue.Actif then FValue.Close;
    if FWiper.Actif then FWiper.Close;
end;

```

```

//*****
procedure TFMMain.FormCreate(Sender: TObject);
var i,j : integer;
    texte : string;
begin
    MainExit:=False;
    ledverteOn:=Tbitmap.Create;
    ledVerteOFF:=Tbitmap.Create;
    ledRougeOn:=Tbitmap.Create;
    ledRougeOff:=Tbitmap.Create;
    FMain.IList_Led.GetBitmap(0,FMain.ledVerteOFF);
    FMain.IList_Led.GetBitmap(3,FMain.ledVerteON);
    FMain.IList_Led.GetBitmap(1,FMain.ledRougeON);
    FMain.IList_Led.GetBitmap(2,FMain.ledRougeOFF);
    ledamberOn:=Tbitmap.Create;
    ledamberOff:=Tbitmap.Create;
    //ledamberOn.LoadFromLazarusResource('LEDpurpleon');
    //ledamberOff.LoadFromLazarusResource('LEDpurpleoff');
    FMain.ComPort1:=TComPort.Create(Self);
    FMain.ISO_14230_:=T_ISO_14230.Create(Self);
    FMain.ISO_14230.Comport:=FMain.ComPort1;
    FMain.ISO_14230.OnReceiveTrame:=FMain.ISO_14230_ReceiveTrame;
    FMain.ISO_14230.OnError:=FMain.ISO_14230_Error;
    FMain.ISO_14230.OnStep:=FMain.ISO_14230_Step;
    FMain.ISO_14230.OnConnect:=FMain.ISO_14230_Connect;
    FMain.ISO_14230.OnDisconnect:=FMain.ISO_14230_Disconnect;
    FMain.ISO_14230.OnStatus:=FMain.ISO_14230_Status;
    // відновлення поточного каталогу
    texte:=Application.ExeName;
    i:=length(texte);
    while (i>1)and (texte[i]<>'\'') and (texte[i]<>'/'') do i:=i-1;
    Repertoire_courant:=copy(texte,1,i); // copie du repertoire courant avec le
slash de fin
    if FileExists(Repertoire_courant+'_ISO_14230.ini')
        then FMain.ISO_14230.LoadConfiguration(Repertoire_courant+'_ISO_14230.ini')
        else FMain.ISO_14230.SaveConfiguration(Repertoire_courant+'_ISO_14230.ini');
    if not FileExists(Repertoire_courant+'languages\ukrainian.ISO_14230_')
        then
FMain.ISO_14230.SaveMessageTXT(Repertoire_courant+'languages\ukrainian.ISO_14230
_');
    Fichier := TStringList.Create;
    Vehicule:=TStringList.Create;
    MenuList:=TStringList.Create;
    ActualMenu := TStringList.Create;
    Actual_ISO_14230_ := TStringList.Create;
    FMain.ExtendedMenu:=TStringList.Create;
    FMain.ClickList:=TStringList.Create;
    FMain.ColorList := TStringList.Create;
    MenuTextInit;
    if not FileExists(Repertoire_courant+'languages\ukrainian.dia')
        then Writelanguage(Repertoire_courant+'languages\ukrainian.dia');
    FMain.Create_Menu_Langue;
    //
    if FileExists(Repertoire_courant+'Diag_ISO_14230.ini') then
begin
    Fichier.LoadFromFile(Repertoire_Courant+'Diag_ISO_14230.ini');
    for i:=0 to Fichier.Count-1 do
begin
    texte:=fichier.Strings[i];
    if (length(texte)>0) and (copy(texte,1,2)<>'/'')
        then begin
            if copy(texte,1,5)='LANG=' then
begin
                delete(texte,1,5);
                texte:=trim(texte);
            end else
            if copy(texte,1,6)='DEBUG=' then
begin
                delete(texte,1,6);

```

```

        texte:=trim(texte);
        try
            FMain.Debug:=StrToInt(texte);
        except
            FMain.Debug:=0;
        end;
    end;
end;
end;
end;
j:=-1;
i:=0;
while (j=-1) and (i<Nb_Langue) do
    if uppercase(M[i].Caption)=uppercase(texte) then j:=i else i:=i+1;
    if j>-1 then
        begin
            ReadLanguage(Repertoire_courant+'Languages\' +texte+'.dia');
FMain.ISO_14230.LoadMessageTXT(Repertoire_courant+'Languages\' +texte+'.ISO_14230
_');
            M[j].Checked:=True;
            end;
        end else
        begin
            Fichier.Clear;
            Fichier.Add('// Diag_ISO_14230_ конфігураційний файл');
            Fichier.Add('');
            Fichier.Add('LANG=ukrainian');
            Fichier.Add('DEBUG=0');
            Fichier.SaveToFile(Repertoire_Courant+'Diag_ISO_14230.ini');
            j:=-1;
            i:=0;
            while (j=-1) and (i<Nb_Langue) do
                if uppercase(M[i].Caption)='UKRAINIAN' then j:=i else i:=i+1;
            if j>-1 then
                begin
                    M[j].Checked:=True;
                    end;
                end;
            end;
            //
            Fmain.Init_Message;
            //FMain.StringGrid3.RowCount:=0;
            FMain.StringGrid1.RowCount:=0;
            if FMain.Debug=0
                then FMain.Group_Debug.Visible:=False
                else FMain.Group_Debug.Visible:=True;
            FMain.GlobalReset;
        end;

//*****
procedure TFMain.FormDestroy(Sender: TObject);
var i : integer;
begin
    FMain.Vehicule.Free;
    FMain.MenuList.Free;
    Fichier.Free;
    ActualMenu.Free;
    Actual_ISO_14230.Free;
    FMain.ExtendedMenu.Free;
    FMain.ClickList.Free;
    FMain.ColorList.Free;
    ledverteOn.Free;
    ledVerteOFF.Free;
    ledRougeOn.free;
    ledRougeOff.free;
    ledAmberOn.free;
    ledAmberOff.free;
    FMain.ComPort1.Free;
    FMain.ISO_14230.Free;
    //master.Free;

```

```

for i:=0 to 9 do if Table[i]<>nil then Table[i].Free;
end;

//*****
procedure TFMMain.ISO_14230_Connect(Sender: TObject);
var texte,textel : string;
    liste_connection : TStringList;
    ListCommand : TStringList;
    cond : string; // ВИМОГА ПОВТОРИТИ КОМАНДУ
    cde : string; // команда відправки
    i,j,k,l,m,n : integer;
    outdoor : boolean;
    byteRule: string;
    DebutByte,FinByte,DebutBit,FinBit : integer;
    valeur,val_cond : integer;
    f_Byte,f_Bit : string;
    mask : byte;
    noeud : TTreeNode;
    n_ligne,max_val,min_val : integer;
begin
    DateTimeToString(texte,'h:nn.ss.zzz',Now);
    FMain.M_Status.Lines.Add(texte+MenuProg[CMF_ConnectOK]);
    FMain.M_Debug.Lines.Add(texte+MenuProg[CMF_ConnectOK]);
    FMain.Imagel.Picture.Bitmap:=nil;
    FMain.IListConnect.GetBitmap(1,FMMain.Imagel.Picture.Bitmap);
    //FMMain.Timer_Maintient.Enabled:=True;
    State:=Etat_Connecte;
    FMain.Menu_Connection.Caption:=MenuProg[CMF_Deconnecter];
    FMain.Panel_Parametre.Visible:=False;
    FMain.Panel_grille.Visible:=False;
    FMain.Panel_Status.Visible:=True;
    FMain.Panel_Liste_Para.Visible:=False;
    FMain.Configuration.GroupBox2.Enabled:=False;
    // Виконати всі команди пошуку відразу після з'єднання
    { try
        liste_Connection:=TStringList.Create;
        ListCommand:=TStringList.Create;
        FMain.Search_Item('CONNECTION',FMMain.Vehicule,Liste_Connection);
        //Liste_Connection.SaveToFile('c:\Connection.txt');
        i:=0;
        FMain.TypeAff:=-1;
        while i<Liste_Connection.Count do
            begin
                Application.ProcessMessages;
                if MainExit then exit;
                texte:=Liste_Connection.Strings[i];
                if uppercase(copy(texte,1,9))='<COMMAND>' then
                    begin
                        ListCommand.Clear;
                        while (texte<>'</COMMAND>') and (i<Liste_Connection.Count)do
                            begin
                                texte:=Liste_Connection.Strings[i];
                                texte:=trim(texte);
                                ListCommand.Add(texte);
                                if texte<>'</COMMAND>' then i:=i+1;
                            end;
                        //ListCommand.SaveToFile('c:\Command'+inttostr(i)+'.txt');
                        // ми отримуємо все поле COMMAND
                        // Таким чином, ми виконаємо команду
                        j:=0;
                        while j<ListCommand.Count do
                            begin
                                Application.ProcessMessages;
                                if MainExit then exit;
                                texte:=trim(ListCommand.Strings[j]);
                                if copy(texte,1,5)='COND=' then
                                    begin
                                        delete(texte,1,5);
                                        cond:=texte;

```

```

end else
if copy(texte,1,5)='<_ISO_14230_>' then
begin
FMain.Actual_ISO_14230.Clear;
cde:='';
while j<ListCommand.Count do
begin
texte:=trim(ListCommand.Strings[j]);
if copy(texte,1,2)<>'</' then
FMain.Actual_ISO_14230.Add(texte);
if copy(texte,1,4)='CDE=' then cde:=texte;
if copy(texte,1,2)='</' then j:=ListCommand.Count;
j:=j+1;
end;
end;
j:=j+1;
end; // end of while j<ListCommand.Count do
end; // end of if uppercase(copy(texte,1,9))='<COMMAND>' then
if length(cde)>0 then
begin
delete(cde,1,4);
FMain.ISO_14230.Sid:=hexdec(copy(cde,1,2));
delete(cde,1,3);
j:=0;
while length(cde)>0 do
begin
Application.ProcessMessages;
if MainExit then Exit;
FMain.ISO_14230.FData[j]:=hexdec(copy(cde,1,2));
delete(cde,1,3);
j:=j+1;
end;
FMain.ISO_14230.SendLength:=j;
if length(cond)>0 then
begin
outdoor:=false;
// Розрахунок фільтра на дані
j:=pos('/',cond);
ByteRule:=copy(cond,1,j-1);
delete(cond,1,j);
// видаляє невідфільтровані дані
j:=pos('[',ByteRule);
if j>0 then delete(ByteRule,1,j);
j:=pos(']',ByteRule);
if j>0 then delete(ByteRule,j,1);
j:=pos('|',ByteRule);
// Отримує дані з бітів
if j>0 then
begin
f_Byte:=copy(ByteRule,1,j-1);
delete(ByteRule,1,j);
f_bit:=ByteRule;
end else
begin
f_Byte:=ByteRule;
f_bit:='';
end;
j:=pos('-',f_Byte);
if j>0 then
begin
textel:=copy(f_Byte,1,j-1);
try
DebutByte:=Strtoint(textel);
except
DebutByte:=0;
end;
delete(f_byte,1,j);
try
FinByte:=Strtoint(f_Byte);

```

```

except
    FinByte:=0;
end;
end else
begin
    try
        DebutByte:=Strtoint(f_Byte);
    except
        DebutByte:=0;
    end;
    FinByte:=DebutByte;
end;
// отримує площину біт
if length(f_bit)>0 then // відфільтровує біти
begin
    j:=pos('-',f_Bit);
    if j>0 then // є площина
        begin
            textel:=copy(f_Bit,1,j-1);
            try
                DebutBit:=Strtoint(textel);
            except
                DebutBit:=0;
            end;
            delete(f_bit,1,j);
            try
                FinBit:=Strtoint(f_Bit);
            except
                FinBit:=0;
            end;
        end else
        begin //Немає площини
            try
                DebutBit:=Strtoint(f_Bit);
            except
                DebutBit:=0;
            end;
            FinBit:=DebutBit;
        end;
        // відновлення створення маски
        mask:=0;
        for j:=DebutBit to FinBit do
            mask:=mask or (1 shl j);
        end else mask:=$FF;
        texte:=cond;
        delete(texte,1,1);
        try
            val_cond:=strtoint(texte);
        except
            outdoor:=true;
        end;
        // повторення виконання
        sleep(500);
        beginExit:=false;
        While (outdoor=False) and (BeginExit=False) do
        begin
            FMain.M_Debug.Lines.Add('_ISO_14230_Connect');
            Application.ProcessMessages;
            if MainExit then Exit;
            FMain.ISO_14230.SendResponse;
            FMain.Attente_Start;
            Execution:=False;
            While (Execution=False)
                and (MainExit=False) do Application.ProcessMessages;
            if MainExit then Exit;
            // визначення фільтра для байта
            valeur:=0;
            for j:=DebutByte to FinByte do
                begin

```

```

                                valeur:=valeur*256+(FMain.ISO_14230.TrameRx[j] and
mask);
                                end;
                                if copy(cond,1,1)='=' then
                                begin
                                    if valeur=val_cond then outdoor:=true;
                                end else
                                if copy(cond,1,1)='>' then
                                begin
                                    if valeur>val_cond then outdoor:=True;
                                end else
                                if copy(cond,1,1)='<' then
                                begin
                                    if valeur<val_cond then outdoor:=True;
                                end else outdoor:=True; // є помилка

                                end;
                                end else
                                begin
                                    sleep(500);
                                    FMain.ISO_14230.SendResponse;
                                    FMain.Attente_Start;
                                    Execution:=False;
                                    While (Execution=False)
                                        and (MainExit=False)do Application.ProcessMessages;
                                    if MainExit then Exit;
                                end;
                                end;
                                i:=i+1;
                                end; // end of while i<ListConnection.Count do
                                finally
                                    liste_Connection.Free;
                                    ListCommand.Free;
                                end;

                                // Виведення меню
                                i:=0;
                                while i<Vehicule.Count do
                                begin
                                    Application.ProcessMessages;
                                    if MainExit then Exit;
                                    texte:=Vehicule.Strings[i];
                                    trim(texte);
                                    if uppercase(copy(texte,1,4))='ICO=' then
                                    begin
                                        //noeud:=FMain.TreeView1.Items.GetFirstNode;
                                        if noeud<>nil then
                                        begin
                                            delete(texte,1,4);
                                            j:=pos('; ',texte);
                                            try
                                                j:=strtoint(copy(texte,1,j-1));
                                            except
                                                j:=-1;
                                            end;
                                            noeud.ImageIndex:=j;
                                        end;
                                    end else
                                    if uppercase(copy(texte,1,8))='<EXMENU=' then
                                    begin
                                        DebutByte:=i;
                                        FMain.ExtendedMenu.Clear;
                                        delete(texte,2,2);
                                        FMain.ExtendedMenu.Add(texte);
                                        // відновлення розширеного меню, операція повинна бути повторена
                                        outdoor:=False;
                                        i:=i+1;
                                        while outdoor=False do
                                        begin

```

```

Application.ProcessMessages;
if MainExit then Exit;
if i<Vehicule.Count then texte:=Vehicule.Strings[i];
trim(texte);
if (i>=Vehicule.Count)
  or (uppercase(copy(texte,1,8))='<EXMENU=')
  or (uppercase(copy(texte,1,8))='</EXMENU')
  or (uppercase(copy(texte,1,6))='<MENU=') then outdoor:=True;
if not outdoor then
begin
  if (length(texte)>0)
    and (copy(texte,1,2)<>'//') then
    begin
      k:=pos(';',texte);
      if k=0 then FMain.ExtendedMenu.Add(texte) else
FMain.ExtendedMenu.Add(copy(texte,1,k));
      end;
      i:=i+1;
    end else FinByte:=i;
  end;
//FMain.ExtendedMenu.SaveToFile('c:\ExtendedMenu.txt');
// Виводячи розширене меню, видаляємо файл даних параметрів автомобіля
for j:=DebutByte to FinByte do
begin
  FMain.Vehicule.Delete(DebutByte);
end;
// ми повинні відновити умови повторення val_cond
j:=0;
while j<FMain.ExtendedMenu.Count do
begin
  Application.ProcessMessages;
  if MainExit then Exit;
  texte:=FMain.ExtendedMenu.Strings[j];
  if copy(texte,1,9)='EXREPEAT=' then
  begin
    delete(texte,1,9);
    k:=pos(';',texte);
    if k>0 then texte:=copy(texte,1,k-1);
    texte:=trim(texte);
    try
      val_cond:=strtoint(texte); // повторення змінної
    except
      val_cond:=-1;
    end;
    FMain.ExtendedMenu.Delete(j); // Стираємо це попередження
    j:=FMain.ExtendedMenu.Count;
  end;
  j:=j+1;
end;
if copy(FMain.ExtendedMenu.Strings[FMain.ExtendedMenu.Count-
1],1,7)<>'</MENU>'
  then FMain.ExtendedMenu.Add('</MENU>');
// Ми починаємо меню інтеграції файлу автомобіля до пам'яті
if (val_cond>-1) and (FMain.Variable[Val_cond]>0) then
begin
  l:=0;
  n_ligne:=0;
  max_val:=0;
  min_val:=10000;
  m:=-1;
  for j:=1 to FMain.Variable[Val_cond] do
  begin
    for k:=0 to FMain.ExtendedMenu.Count-1 do
    begin
      Application.ProcessMessages;
      if MainExit then Exit;
      texte:=FMain.ExtendedMenu.Strings[k];
      if copy(texte,1,4)='<MEN' then
        begin

```

```

        debutBit:=pos('>',texte);
        if debutBit>0 then texte:=copy(texte,1,debutbit-1);
        texte:=texte+' '+inttostr(j)+'>';
    end else
    if copy(texte,1,4)='CLK=' then
    begin
        m:=pos('EXV',texte);
        textel:=dectoohex(j);
        while m>0 do
            begin
                texte:=copy(texte,1,m-
1)+textel+copy(texte,m+3,length(texte)-3);
                m:=pos('EXV',texte);
            end;
        end else
    if copy(texte,1,4)='CDE=' then
    begin
        debutBit:=pos('; ',texte);
        if debutBit>0 then texte:=copy(texte,1,debutbit-1);
        texte:=texte+' '+dectoohex(j)+' ';
    end else
    if copy(texte,1,3)='LG=' then
    begin
        debutbit:=pos('TAB[ ',texte);
        if debutbit>0 then
            begin
                cond:=copy(texte,1,debutbit+3);
                delete(texte,1,debutbit+3);
                debutbit:=pos(' ',texte);
                cond:=cond+copy(texte,1,debutbit);
                delete(texte,1,debutbit);
                debutbit:=pos(']',texte);
                valeur:=strtoint(copy(texte,1,debutbit-1));
                delete(texte,1,debutbit-1);
                if max_val<valeur then max_val:=valeur;
                if min_val>valeur then min_val:=valeur;
                valeur:=valeur+n_ligne;
                texte:=cond+inttostr(valeur)+texte;
            end;
        end else
    if copy(texte,1,3)='TR=' then
    begin
        debutbit:=pos('TAB[ ',texte);
        if debutbit>0 then
            begin
                cond:=copy(texte,1,debutbit+3);
                delete(texte,1,debutbit+3);
                debutbit:=pos(' ',texte);
                cond:=cond+copy(texte,1,debutbit);
                delete(texte,1,debutbit);
                debutbit:=pos(']',texte);
                valeur:=strtoint(copy(texte,1,debutbit-1));
                delete(texte,1,debutbit-1);
                if max_val<valeur then max_val:=valeur;
                if min_val>valeur then min_val:=valeur;
                valeur:=valeur+n_ligne;
                texte:=cond+inttostr(valeur)+texte;
            end;
        end;
        FMain.Vehicule.Insert(DebutByte+k+(j-
1)*FMain.ExtendedMenu.Count),texte);
    end;
    // розраховуємо l
    // n_ligne => текст у таблицю
    // valeur => читати останнє значення
    n_ligne:=n_ligne+(max_val-min_val)+1;
end;
end;
end;

```

```

        // Ми закінчили підключення, це переводить курсор на початок цієї
області
        i:=DebutByte-1;
    end else
    if uppercase(copy(texte,1,6))='<MENU=' then
        begin
        delete(texte,1,6);
        j:=pos('>',texte);
        if j>0 then
            begin
                MenuList.Add(copy(texte,1,j-1)+'/'+inttostr(i)+'/');
                noeud:=FMain.TreeView1.Items.Add(nil,copy(texte,1,j-1));
            end;
        end else
        if uppercase(copy(texte,1,6))='</MENU' then
            begin
            delete(texte,1,6);
            if MenuList.Count>0 then MenuList.Strings[MenuList.Count-
1]:=MenuList.Strings[MenuList.Count-1]+inttostr(i)+'/';
            end ;
            i:=i+1;
        end;
        //FMain.TreeView1.Enabled:=True;
        FMain.Vehicule.SaveToFile('C:\vehicule.txt');
    }
    FMain.T_Command.Enabled:=True;
end;

//*****
// Роз'єднання
//*****
procedure TFMMain.ISO_14230_Disconnect(Sender: TObject);
begin
    //FMain.Timer_Maintient.Enabled:=False;
    F_Configuration.GroupBox2.Enabled:=True;
    FMain.TreeView1.Enabled:=False;
    FMain.TreeView1.Items.Clear;
    FMain.Menu_Connection.Caption:=MenuProg[CMF_Connector];
    FMain.Imagel.Picture.Bitmap:=nil;
    FMain.IListConnect.GetBitmap(0,FMain.Imagel.Picture.Bitmap);
    F_Configuration.GroupBox2.Enabled:=True;
    State:=0;
    FMain.ComPort1.Close;
    FMain.Attente_Stop;
end;

//*****
//Помилки
//*****
procedure TFMMain.ISO_14230_Error(Sender: TObject; Erreur: integer;
MsgErreur: string);
var texte : string;
begin
    DateTimeToString(texte,'h:nn.ss.zzz',Now);
    execution:=True;
    FMain.M_Status.Lines.Add(texte+' - '+MsgErreur);
    FMain.M_Debug.Lines.Add(texte+' - MainError: '+MsgErreur);
    FMain.Attente_Stop;
    if erreur=Err_5Baud then
        begin
            {FMain.Attente_Stop;

            FMain.cpt_erreur:=FMain.cpt_erreur+1;
            if FMain.cpt_erreur<=3
                then FMain.ISO_14230.StartDiagnosis
                else FMain.GlobalReset;
            }
        end;
    if Erreur=Err_TO then

```

```

begin
  if FMain.State=Etat_Connecte then
    begin
      //FMain.cpt_erreur:=FMain.cpt_erreur+1;
      //if FMain.cpt_erreur<=3
      //  then FMain.ISO_14230.SendResponse
      //  else FMain.Menu_ConnectionClick(nil);
      end;
      F_Configuration.GroupBox2.Enabled:=True;
      FMain.GlobalReset;
    end;

    //FMain.diag_actif:=False;
    //FMain.Attente_Start;
    //FMain.Timer_Maintient.Enabled:=True;
    //FMain.Execution:=True;
end;

//*****
procedure TFMMain.ISO_14230_ReceiveTrame(Sender: TObject);

var i,j,k,l,m,n,k1 : integer;
    texte,datetxt : string;
    octet,octet1 : byte;
    noeud : TTreeNode;
    ByteRule : string;
    RepeatRule : string;
    f_Byte,f_bit : string;
    DebutByte,FinByte : integer;
    DebutBit,FinBit : integer;
    Mask : Byte;
    textel,texte2,texte3 : string;
    value : real;
    Val_Int : integer;
    nb_byteValue : array[0..100] of integer;
    nb_GroupValue : integer;
    ByteValue : array[0..100,0..255] of byte;
    ax,bx : real; // Ce значення розраховується Value=ax*X+bx
    couleur : string;
    IRow,AffRow : integer;
    sens_byte : Boolean;
    etat_timerIO : boolean;

begin
  FMain.cpt_erreur:=0;
  IRow:=-1;
  //FMain.Timer_Maintient.Enabled:=True;
  FMain.Attente_Stop;
  texte:='';
  For i:=0 to FMain.ISO_14230.ReceiveLength-1 do
    texte:=texte+' '+dectohex(FMain.ISO_14230.TrameRx[i]);
  DateTimeToString(datetxt,'h:nn.ss.zzz',Now);
  FMain.M_Debug.Lines.Add(Datetxt+' -Fonction Receive- '+texte);
  FMain.M_Debug.Lines.Add('');
  if (FMain.ClickTree) or (FMain.ISO_14230.SidReceive=$7E) then
    begin
      Execution:=True;
      exit;
    end;
  if FMain.NoReceive then
    begin
      if FOnOff.Actif=True then FOnOff.Enable_Screen;
      if FBiPWM.Actif then FBiPwm.Enable_Screen;
      if FPwm.Actif then FPWM.Enable_Screen;
      if FBridge.Actif then FBridge.Enable_Screen;
      if FValue.Actif then FValue.Enable_Screen;
    end;

```

```

    if FWiper.Actif then FWiper.Enable_Screen;
    FMain.NoReceive:=False;
    Execution:=True;
    Exit;
end;
if State>0 then
begin
    //if (FMain.ISO_14230.Sid=$12) and (FMain.ISO_14230.TrameRx[0]=$01) then
    // begin
    //     Couleur:='zz';
    // end;
    octet:=FMain.ISO_14230.Sid+$40;
    if octet<>FMain.ISO_14230.SidReceive then
        begin
            FMain.M_Debug.Lines.Add(Datetxt+' - octet='+dectohex(octet)+'-
Sid='+dectohex(FMain.ISO_14230.Sid));
            //FMain.Timer_Maintient.Enabled:=False;
            FMain.M_Debug.Lines.Add(Datetxt+MenuProg[CMF_SidError]);
            FMain.M_Status.Lines.Add(Datetxt+MenuProg[CMF_SidError]);
            //FMain.ISO_14230.SendResponse;
            //FMain.Attente_Start;
            Execution:=True;
            if FMain.TypeAff=CT_AFFONEVALUE then
                begin
                    texte:='';
                    for k:=0 to FMain.ISO_14230.ReceiveLength-1 do
                        begin
                            texte:=texte+dectohex(FMain.ISO_14230.TrameRx[k])+ ' ';
                        end;
                    texte:=dectohex(FMain.ISO_14230.SidReceive)+' '+texte+' /
'+MenuProg[CMF_SidError];
                    FMain.M_Valeur.Lines.Add(Datetxt+' - '+texte);
                    FMain.E_Valeur.Text:='';
                end else
                if FMain.TypeAff=CT_AFFONEPARAMETER then
                    begin
                        texte:='';
                        for k:=0 to FMain.ISO_14230.ReceiveLength-1 do
                            begin
                                texte:=texte+dectohex(FMain.ISO_14230.TrameRx[k])+ ' ';
                            end;
                        texte:=dectohex(FMain.ISO_14230.SidReceive)+' '+texte+' /
'+MenuProg[CMF_SidError];
                        FMain.M_Parametre.Lines.Add(Datetxt+' - '+texte);
                        FMain.E_Para.Text:='';
                    end;
                exit;
            end;
            //FMain.Timer_Maintient.Enabled:=True;
            etat_TimerIO:=FMain.T_StatusIO.Enabled;
            FMain.T_StatusIO.Enabled:=False;
            Couleur:='';
            // Ми беремо дані з байта для перегляду
            for i:=0 to Actual_ISO_14230.Count-1 do
                begin
                    texte:=Actual_ISO_14230.Strings[i];
                    if uppercase(copy(texte,1,3))='TR=' then
                        begin
                            sens_byte:=CT_SensHL;
                            delete(texte,1,3);
                            // ми форматуємо вид значення даних на екрані
                            j:=pos('/',texte);
                            ByteRule:=copy(texte,1,j-1);
                            delete(texte,1,j);
                            //delete(ByteRule,1,1);
                            //ByteRule:=copy(ByteRule,1,length(ByteRule)-1);
                            nb_GroupValue:=1;
                            if length(ByteRule)>0 then
                                begin

```

```

// створюємо правило для пошуку даних
if uppercase(ByteRule[1])='R' then
begin
j:=pos('[',ByteRule);
RepeatRule:=copy(ByteRule,2,j-2); // je рйсупйре le debut
de la loi de рйрйtition
delete(ByteRule,1,j-1);
end else
begin
// ми не повторюємо правило
RepeatRule:='';
end;
if uppercase(copy(ByteRule,length(ByteRule),1))='L' then
begin
delete(byteRule,length(byteRule),1);
sens_Byte:=CT_SensLH;
end;
// видаляє невідфільтровані дані
j:=pos('[',ByteRule);
if j>0 then delete(byteRule,1,j);
j:=pos('|',ByteRule);
if j>0 then ByteRule:=copy(ByteRule,1,j-1);
j:=pos('|',ByteRule);
// Отримує дані з бітів
if j>0 then
begin
f_Byte:=copy(ByteRule,1,j-1);
delete(ByteRule,1,j);
f_bit:=ByteRule;
end else
begin
f_Byte:=ByteRule;
f_bit:='';
end;
j:=pos('-',f_Byte);
if j>0 then
begin
textel:=copy(f_Byte,1,j-1);
try
DebutByte:=Strtoint(textel);
except
DebutByte:=0;
end;
delete(f_byte,1,j);
try
FinByte:=Strtoint(f_Byte);
except
FinByte:=0;
end;
end else
begin
try
DebutByte:=Strtoint(f_Byte);
except
DebutByte:=0;
end;
FinByte:=DebutByte;
end;
if DebutByte>FinByte then
begin
j:=DebutByte;
DebutByte:=FinByte;
FinByte:=j;
end;
if FinByte>FMain.ISO_14230.ReceiveLength then
FinByte:=FMain.ISO_14230.ReceiveLength-1;
if DebutByte>FMain.ISO_14230.ReceiveLength then
DebutByte:=FMain.ISO_14230.ReceiveLength;
// отримує площину біт

```

```

if length(f_bit)>0 then // відфільтровує біти
begin
  j:=pos('-',f_Bit);
  if j>0 then // є площина
  begin
    textel:=copy(f_Bit,1,j-1);
    try
      DebutBit:=Strtoint(textel);
    except
      DebutBit:=0;
    end;
    delete(f_bit,1,j);
    try
      FinBit:=Strtoint(f_Bit);
    except
      FinBit:=0;
    end;
  end else
  begin //Немає площини
    try
      DebutBit:=Strtoint(f_Bit);
    except
      DebutBit:=0;
    end;
    FinBit:=DebutBit;
  end;
  // відновлення створення маски
  mask:=0;
  for j:=DebutBit to FinBit do
    mask:=mask or (1 shl j);
  end else
  begin
    mask:=$FF;
    DebutBit:=0;
    FinBit:=0;
  end;
  if DebutBit>FinBit then
  begin
    j:=DebutBit;
    DebutBit:=FinBit;
    FinBit:=j;
  end;
  // значення заповнюються в буфер значення
  if length(RepeatRule)>0 then
  begin
    try
      j:=strtoint(RepeatRule);
    except
      j:=0;
    end;
    nb_GroupValue:=0;
    repeat
      if (j+DebutByte<FMain.ISO_14230.ReceiveLength) then
      begin
        l:=0;
        for k:=j+DebutByte to j+FinByte do
          begin
            if Sens_Byte=CT_SensHL
            then
              ByteValue[nb_GroupValue,l]:=(FMain.ISO_14230.TrameRx[k] and mask) shr DebutBit
            else
              ByteValue[nb_GroupValue,l]:=(FMain.ISO_14230.TrameRx[k] and mask) shr debutBit;
            l:=l+1;
          end;
          nb_ByteValue[nb_GroupValue]:=FinByte-DebutByte+1;
          nb_GroupValue:=nb_GroupValue+1;
        end;
        j:=j+FinByte+1;
      until j>FMain.ISO_14230.ReceiveLength-1;

```

```

end else // end of if length(RepeatRule)>0 then
begin
  if Sens_Byte=CT_SensHL then l:=0 else l:=FinByte-
DebutByte;
  for j:=DebutByte to FinByte do
  begin
    if Sens_Byte=CT_SensHL then
    begin
      ByteValue[0,l] := (FMain.ISO_14230.TrameRx[j] and
mask) shr DebutBit;
      l:=l+1;
    end else
    begin
      ByteValue[0,l] := (FMain.ISO_14230.TrameRx[j] and
mask) shr DebutBit;
      l:=l-1;
    end;
  end;
  nb_ByteValue[0]:=FinByte-DebutByte+1;
  nb_GroupValue:=1;
end; // end of ELSE of if length(RepeatRule)>0 then
end else // end of if length(ByteRule)>0 then
begin
  nb_ByteValue[0]:=FMain.ISO_14230.ReceiveLength;
  nb_GroupValue:=1;
  if Sens_Byte=CT_SensHL then l:=0 else l:=FinByte-DebutByte;
  for j:=0 to nb_byteValue[0]-1 do
  begin
    if Sens_Byte=CT_SensHL then
    begin
      ByteValue[0,l] := (FMain.ISO_14230.TrameRx[j]) shr
DebutBit;
      l:=l+1;
    end else
    begin
      ByteValue[0,l] := (FMain.ISO_14230.TrameRx[j]) shr
DebutBit;
      l:=l-1;
    end;
  end;
end; // end of ELSE if length(ByteRule)>0 then
// ми будемо шукати, якщо є набір значень
ax:=1;
bx:=0;
if length(texte)>0 then
  if uppercase(copy(texte,1,2))='V=' then
  begin
    j:=pos('/',texte);
    textel:=copy(texte,1,j-1);
    delete(texte,1,j);
    delete(textel,1,2);
    j:=pos('|',textel);
    if j>0 then
    begin
      ax:=strtoReel(copy(textel,1,j-1));
      delete(textel,1,j);
      bx:=strtoReel(textel);
    end else
    begin
      ax:=strtoReel(textel);
      bx:=0;
    end;
  end;
end;
if uppercase(copy(texte,1,4))='Значення=' then
begin
  delete(texte,1,4);
  j:=pos('/',texte);
  if j=0 then j:=pos('; ',texte);
  if j>0 then texte2:=copy(texte,1,j-1) else texte2:=texte;

```

```

delete(texte,1,length(texte2)+1);
try
  j:=strtoint(texte2);
except
  j:=0;
end;
FMain.Variable[j]:=0;
for k:=0 to nb_ByteValue[0]-1 do
begin
FMain.Variable[j]:=FMain.Variable[j]*256+ByteValue[0,k];
end;
end;
if uppercase(copy(texte,1,6))='Таблица=' then
begin
delete(texte,1,6);
j:=pos('/',texte);
if j=0 then j:=pos('; ',texte);
if j>0 then texte2:=copy(texte,1,j-1) else texte2:=texte;
delete(texte,1,length(texte2)+1);
try
  j:=strtoint(texte2);
except
  j:=0;
end;
texte2:=' ';
for k:=0 to nb_ByteValue[0]-1 do
begin
//texte2:=texte2+dectohex(ByteValue[0,k])+' ';
if ByteValue[0,k]>=32
then texte2:=texte2+chr(ByteValue[0,k]);
end;
if FMain.Table[j]=nil then
FMain.Table[j]:=TStringList.create;
FMain.Table[j].Add(texte2);

//FMain.Table[j].SaveToFile('c:\table_'+inttostr(j)+'.txt');
end;
if uppercase(copy(texte,1,4))='Колонка=' then
begin
delete(texte,1,4);
j:=pos('/',texte);
texte2:=copy(texte,1,j-1);
delete(texte,1,j);
IRow:=strtoint(texte2);
end;
// ми записуємо значення на екран
for j:=0 to nb_GroupValue-1 do
begin
textel:=texte;
if FMain.TypeAff=CT_AFFMULTIVALUE then
begin
texte2:=' ';
val_int:=0;
for k:=0 to nb_ByteValue[j]-1 do
begin
val_int:=val_int*256+ByteValue[j,k];
end;
if FMain.TypeCdeActif=1 then
FMain.StringGrid2.Cells[FMain.StringGrid2.ColCount-
1,FMain.ligneGrid2]:=inttostr(val_int);
if FMain.ligneGrid2<FMain.StringGrid2.RowCount-1 then
begin
FMain.ligneGrid2:=FMain.ligneGrid2+1;
try
k:=hexdec(uppercase(FMain.StringGrid2.Cells[0,FMain.ligneGrid2]));
except

```

```

Application.MessageBox (PCHAR (MenuProg [CMP_NumberNotHex] ) , PCHAR (MenuProg [CMP_Error] ) , 0);

        exit;
    end;
    if k<=255 then
    begin
        if FMain.TypeCdeActif=1
            then texte:=FMain.CdeReadParam
            else texte:=FMain.CdeWriteParam;
        if texte[1]='L' then l:=0 else l:=1;
        delete(texte,1,1);
        FMain.ISO_14230.Sid:=hexdec (copy (texte,1,2));
        delete(texte,1,3);
        FMain.ISO_14230.SendLength:=0;
        while length(texte)>0 do
            begin

FMain.ISO_14230.FData [FMain.ISO_14230.SendLength] :=hexdec (copy (texte,1,2));

FMain.ISO_14230.SendLength:=FMain.ISO_14230.SendLength+1;
                delete(texte,1,3);
            end;

FMain.ISO_14230.FData [FMain.ISO_14230.SendLength] :=k;

FMain.ISO_14230.SendLength:=FMain.ISO_14230.SendLength+1;
                if FMain.TypeCdeActif=2 then
                    begin

texte:=FMain.StringGrid2.Cells [FMain.StringGrid2.ColCount-1,FMain.ligneGrid2];
                    try
                        texte:=dectohex (strtoint (texte));
                    except
                        texte:='00';
                    end;
                    if
(FMain.StringGrid2.Cells [3,FMain.ligneGrid2]='16')
                        and (length(texte)=2)
                            then texte:='00'+texte;
                    if (l=0) and (length(texte)>2) then
                        begin
                            texte:=copy (texte,3,2)+copy (texte,1,2);
                        end;
                    l:=(length(texte) div 2);
                    for k:=0 to l-1 do
                        begin

FMain.ISO_14230.FData [k+FMain.ISO_14230.SendLength] :=hexdec (copy (texte,1,2));
                            delete(texte,1,2);
                        end;

FMain.ISO_14230.SendLength:=FMain.ISO_14230.SendLength+1;
                            end; // end of if FMain.TypeCdeActif=2 then
                            // Конфігурація системи
                            FMain.ISO_14230.SendResponse;
                            FMain.Attente_Start;
                            exit;
                        end;
                    end; // end of if

FMain.ligneMenu<FMain.StringGrid2.RowCount then
                    exit;
                end else
                    if FMain.TypeAff=CT_AFFONEVALUE then
                        begin
                            texte2:='';
                            val_int:=0;
                            for k:=0 to nb_ByteValue [j]-1 do
                                begin

```

```

        texte2:=texte2+dectohex(ByteValue[j,k])+' ';
        val_int:=val_int*256+ByteValue[j,k];
    end;
    texte2:=dectohex(FMain.ISO_14230.SidReceive)+' '+texte2;
    FMain.M_Valeur.Lines.Add(Datetxt+' - '+texte2);
    if FMain.CdeValueAction=1 then
    begin
        if FMain.RadioAffVal1.ItemIndex=0
        then FMain.E_Valeur.Text:=inttostr(val_int)
        else FMain.E_Valeur.Text:=dectohex(val_int);
    end;
    end else
if FMain.TypeAff=CT_AFFONEPARAMETER then
begin
    texte2:='';
    val_int:=0;
    for k:=0 to nb_ByteValue[j]-1 do
    begin
        texte2:=texte2+dectohex(ByteValue[j,k])+' ';
        val_int:=val_int*256+ByteValue[j,k];
    end;
    texte2:=dectohex(FMain.ISO_14230.SidReceive)+' '+texte2;
    FMain.M_Parametre.Lines.Add(Datetxt+' - '+texte2);
    if FMain.CdeValueAction=1 then
    begin
        if FMain.RadioAffVal.ItemIndex=0
        then FMain.E_Para.Text:=inttostr(val_int)
        else FMain.E_Para.Text:=dectohex(val_int);
    end;
    end else //end of if Fmain.TypeAff=CT_AFFONEPARAMETER then
if Fmain.TypeAff=CT_AFFCONFIRM then
begin
Application.MessageBox(PCHAR(MenuProg[ CMP_MsgConfirm]),PCHAR(MenuProg[ CMP_MsgOK]
),0);

        exit;
    end else // end of if Fmain.TypeAff=CT_AFFCONFIRM then
if FMain.TypeAff=CT_AFFGRID then
begin
    if IRow=-1 then
    begin

//FMain.StringGrid3.RowCount:=FMain.StringGrid3.RowCount+1;
//for k:=0 to FMain.StringGrid3.ColCount-1 do
//
FMain.StringGrid3.Cells[k,FMain.StringGrid3.RowCount-1];
//AffRow:=FMain.StringGrid3.RowCount-1;

FMain.StringGrid1.RowCount:=FMain.StringGrid1.RowCount+1;

FMain.Ajust_StringList(FMain.ColorList.Count+1,FMain.ColorList);
    for k:=0 to FMain.StringGrid1.ColCount-1 do
FMain.StringGrid1.Cells[k,FMain.StringGrid1.RowCount-1]:='';
    AffRow:=FMain.StringGrid1.RowCount-1;
    end else
    begin
        //if IRow>=FMain.StringGrid3.RowCount then
        // begin
        //     FMain.StringGrid3.RowCount:=IRow+1;
        //     for k:=0 to FMain.StringGrid3.ColCount-1 do
        //         FMain.StringGrid3.Cells[k,IRow]:='';
        //     end;
        if IRow>=FMain.StringGrid1.RowCount then
        begin
            FMain.StringGrid1.RowCount:=IRow+1;
            FMain.Ajust_StringList(IRow+1,FMain.ColorList);
            for k:=0 to FMain.StringGrid1.ColCount-1 do
                FMain.StringGrid1.Cells[k,IRow]:='';

```



```

then
FMain.StringGrid1.ColWidths[1]:=FMain.StringGrid1.Canvas.TextWidth(texte3)+32;

//FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]:=FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]+'T'+Texte3+'|';

end else
if uppercase(copy(texte2,1,7))='VAL_TXT' then
begin
texte2:='';
for k:=0 to nb_ByteValue[j]-1 do
begin
if ByteValue[j,k]>31
then
texte2:=texte2+chr(ByteValue[j,k]);
end;

//FMain.StringGrid3.Cells[1,AffRow]:=texte2;
FMain.StringGrid1.Cells[1,AffRow]:=texte2;
if
FMain.StringGrid1.ColWidths[1]<FMain.StringGrid1.Canvas.TextWidth(texte2)+32
then
FMain.StringGrid1.ColWidths[1]:=FMain.StringGrid1.Canvas.TextWidth(texte2)+32;

//FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]:=FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]+'T'+Texte2+'|';

end else
if uppercase(copy(texte2,1,5))='VALUE' then
begin
Value:=0;
for k:=0 to nb_ByteValue[j]-1 do
begin
Value:=Value*256+ByteValue[j,k];
end;
Value:=Value*ax+bx;
texte2:=reeltostr(Value);
FMain.StringGrid1.Cells[1,AffRow]:=texte2;
if
FMain.StringGrid1.ColWidths[1]<FMain.StringGrid1.Canvas.TextWidth(texte2)+32
then
FMain.StringGrid1.ColWidths[1]:=FMain.StringGrid1.Canvas.TextWidth(texte2)+32;

//FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]:=FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]+'V'+Texte2+'|';

end else
if uppercase(copy(texte2,1,7))='VAL_HEX' then
begin
texte2:='';
for k:=0 to nb_ByteValue[j]-1 do
begin
texte2:=texte2+dectohex(ByteValue[j,k])+' ';
end;
FMain.StringGrid1.Cells[1,AffRow]:=texte2;
if
FMain.StringGrid1.ColWidths[1]<FMain.StringGrid1.Canvas.TextWidth(texte2)+32
then
FMain.StringGrid1.ColWidths[1]:=FMain.StringGrid1.Canvas.TextWidth(texte2)+32;

//FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]:=FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]+'H'+Texte2+'|';

end else
if uppercase(copy(texte2,1,4))='IMG_' then
begin

//FMain.StringGrid3.Cells[1,AffRow]:=texte2;

```

```

Val_Int:=0;
for k:=0 to nb_ByteValue[j]-1 do
begin
    Val_Int:=Val_Int*256+ByteValue[j,k];
end;
if Val_int=0 then texte2:=texte2+'0' else
texte2:=texte2+'1';

//FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]:=inttostr(Val_Int);
FMain.StringGrid1.Cells[1,AffRow]:=texte2;
if
FMain.StringGrid1.ColWidths[1]<FMain.StringGrid1.Canvas.TextWidth(texte2)+32
then
FMain.StringGrid1.ColWidths[1]:=FMain.StringGrid1.Canvas.TextWidth(texte2)+32;
//FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]:=FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]+'V'+texte2[length(texte2)]+'|';
end else
if uppercase(copy(texte2,1,5))='Файл(' then
begin
delete(texte2,1,5);
k:=pos(')',texte2);
if k>0 then
begin
    texte2:=copy(texte2,1,k-1);
    texte2:=uppercase(trim(texte2));
    if uppercase(FMain.nom_fichier)<>texte2
then
        if
FileExists(Repertoire_courant+'Vehicule\''+texte2) then
            begin
                FMain.nom_fichier:=texte2;
FMain.Fichier.LoadFromFile(Repertoire_courant+'Vehicule\''+texte2);
end;
Val_Int:=0;
for k:=0 to nb_ByteValue[j]-1 do
begin
Val_Int:=Val_Int*256+ByteValue[j,k];
end;
texte2:=dectohex(Val_Int);
k:=0;
while k<FMain.Fichier.Count do
begin
    m:=pos('; ',Fichier.Strings[k]);
    if copy(Fichier.Strings[k],1,m-
1)=texte2 then
        begin
            texte2:=Fichier.Strings[k];
            delete(texte2,1,m);
            k:=FMain.Fichier.Count;
        end;
        k:=k+1;
    end;
FMain.StringGrid1.Cells[1,AffRow]:=texte2;
if
FMain.StringGrid1.ColWidths[1]<FMain.StringGrid1.Canvas.TextWidth(texte2)+32
then
FMain.StringGrid1.ColWidths[1]:=FMain.StringGrid1.Canvas.TextWidth(texte2)+32;
//FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]:=FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]+'T'+texte2+'|';
end;

```

```

end else
if uppercase(copy(texte2,1,4))='TAB[' then
begin
delete(texte2,1,4);
k:=pos(',',texte2);
if k>0 then
begin
try
m:=strtoint(copy(texte2,1,k-1));
except
m:=-1;
end;
delete(texte2,1,k);
k:=pos(']',texte2);
try
n:=strtoint(copy(texte2,1,k-1));
except
n:=-1;
end;
if (m>-1) and (m<10) then
begin
if FMain.Table[m]<>nil then
begin
if (n>-1) and
(n<FMain.Table[m].Count) then
begin
texte2:=FMain.Table[m].Strings[n];
end else
texte2:=MenuProg[CMPE_ERRORLIGNETABLE];
end else
texte2:=MenuProg[CMPE_TableUnknow];
end else
texte2:=MenuProg[CMPE_ERRORNUMBERTABLE];
end;
FMain.StringGrid1.Cells[1,AffRow]:=texte2;
if
FMain.StringGrid1.ColWidths[1]<FMain.StringGrid1.Canvas.TextWidth(texte2)+32
then
FMain.StringGrid1.ColWidths[1]:=FMain.StringGrid1.Canvas.TextWidth(texte2)+32;

//FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]:=FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]+'T'+texte2+'|';
end else
begin
FMain.StringGrid1.Cells[1,AffRow]:=texte2;
if
FMain.StringGrid1.ColWidths[1]<FMain.StringGrid1.Canvas.TextWidth(texte2)+32
then
FMain.StringGrid1.ColWidths[1]:=FMain.StringGrid1.Canvas.TextWidth(texte2)+32;

//FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]:=FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-
1,AffRow]+'V'+texte2+'|';
end;
end;
end;
if k1>0 then delete(texte1,1,k1) else texte1:='';
until length(texte1)<1;
if FMain.StringGrid1.RowCount>1 then
FMain.StringGrid1.FixedRows:=1;
end;
end; // end of for j:=0 to nb_GroupValue-1 do
//FMain.rafraichir_diag;
end;
end; // end of for i:=0 to Actual_ISO_14230.Count-1 do
FMain.Execution:=True;
if InIOTimer then Exit;

```

```

FMain.M_Debug.Lines.Add('Час повторення: '+inttostr(FMain.TimeRepeat)+'
ms');
if (FMain.TypeAff<>CT_AFFCONFIRM)
  and (FMain.TypeAff<>CT_AFFONEPARAMETER)
  and (FMain.ligneMenu<FMain.ActualMenu.Count) then
begin
  FMain.T_StatusIO.Interval:=100;
  FMain.T_StatusIO.Enabled:=True;
end else
begin
  If (FMain.TimeRepeat>0) and (FMain.TypeAff<>CT_AFFCONFIRM) then
  begin
    FMain.ligneMenu:=0;
    FMain.T_StatusIO.Interval:=FMain.TimeRepeat;
    FMain.T_StatusIO.Enabled:=True;
  end;
end;
end;
end;

//*****
procedure TFMain.ISO_14230_Status(Sender: TObject; Messages: string);
var texte : string;
begin
  DateTimeToString(texte, 'h:nn.ss.zzz', Now);
  FMain.M_Status.Lines.Add(texte+' - '+Messages);
end;

//*****
procedure TFMain.ISO_14230_Step(Sender: TObject; Messages: string);
var texte : string;
begin
  DateTimeToString(texte, 'h:nn.ss.zzz', Now);
  FMain.M_Debug.Lines.Add(texte+' - '+Messages);
  FMain.num_image:=FMain.num_image+1;
  if FMain.num_image>14 then FMain.num_image:=1;
  FMain.I_Attente.Picture.Bitmap:=nil;
  FMain.IList_Rose.GetBitmap(FMain.num_image, FMain.I_Attente.Picture.Bitmap);
end;

//*****
procedure TFMain.Menu_QuitterClick(Sender: TObject);
begin
  FMain.Execution:=True;
  FMain.T_StatusIO.Enabled:=False;
  FMain.Close;
end;

//*****
procedure TFMain.Menu_SettingClick(Sender: TObject);
begin
  F_Configuration.execute;
end;

//*****
procedure TFMain.PanellClick(Sender: TObject);
begin
  FMain.SaveDialog1.FileName:=repertoire_courant+'Debug.txt';
  FMain.SaveDialog1.Filter:='*.txt';
  if FMain.SaveDialog1.Execute then
  begin
    FMain.M_Debug.Lines.SaveToFile(FMain.SaveDialog1.FileName);
  end;
end;

//*****
procedure TFMain.RadioAffVallChangeBounds(Sender: TObject);
var texte : string;
    value : integer;

```

```

begin
  texte:=FMain.E_Valeur.Text;
  texte:=trim(texte);
  try
    if FMain.RadioAffVal.ItemIndex=0 then
      begin
        value:=hextodec(texte);
        texte:=inttostr(value);
      end else
      begin
        value:=strtoint(texte);
        texte:=dectohex(value);
      end;
    FMain.E_Valeur.Text:=texte;
  except
  end;
end;

//*****
procedure TFMMain.RadioAffValChangeBounds(Sender: TObject);
var texte : string;
    value : integer;
begin
  texte:=FMain.E_Para.Text;
  texte:=trim(texte);
  try
    if FMain.RadioAffVal.ItemIndex=0 then
      begin
        value:=hextodec(texte);
        texte:=inttostr(value);
      end else
      begin
        value:=strtoint(texte);
        texte:=dectohex(value);
      end;
    FMain.E_Para.Text:=texte;
  except
  end;
end;

//*****
procedure TFMMain.StringGrid1Click(Sender: TObject);
var aRow,aCol,ligne : integer;
    texte,textel : string;
    i,j,k : integer;
    x,y : integer;
    tx,ty : string;
begin
  // Натиснути кнопку у головному вікні

  aRow:=FMain.StringGrid1.Row;
  aCol:=FMain.StringGrid1.Col;
  if aRow=0 then exit;
  //ligne:=aRow+FMain.Grille_Up; // на наступний рядок
  ligne:=aRow;
  if (FMain.ClickActif=False) and (length(FMain.ClickGlobal)>0) then
    begin
      While (FMain.ISO_14230.EndReceive=False)
        and (MainExit=False) do Application.ProcessMessages;
      if MainExit then Exit;
      FMain.Cde2Send(FMain.ClickGlobal);
      FMain.Attente_Start;
      FMain.Execution:=False;
      FMain.ISO_14230.SendResponse;
      While (not FMain.Execution)
        and (MainExit=False) do application.ProcessMessages;
      if MainExit then Exit;
      FMain.ClickActif:=True;
      FMain.B_Mode_Maitre.Visible:=True;
    end;
end;

```

```

end;
i:=0;
while i<ClickList.Count do
begin
  texte:=ClickList.Strings[i];
  if uppercase(copy(texte,1,4))='CLK=' then
  begin
    delete(texte,1,4);
    j:=pos('/',texte);
    if j>0 then
    begin
      textel:=copy(texte,1,j-1);
      delete(texte,1,j);
      j:=pos(', ',textel);
      if j>0 then
      begin
        tx:=copy(textel,1,j-1);
        delete(textel,1,j);
        ty:=textel;
        try
          if length(tx)>0 then x:=strtoint(tx) else x:=-1;
        except
          x:=-1;
        end;
        try
          y:=strtoint(ty);
        except
          y:=-1;
        end;
        if ((y>=0) and (x>=0) and (x=aCol) and (y=ligne))
          or ((x<0) and (y>=0) and (y=ligne))
          or ((x>=0) and (y<0) and (x=aCol))
          or ((x<0) and (y<0)) then
        begin
          // переходимо до умов описаних низче
          while length(texte)>0 do
          begin
            j:=pos('/',texte);
            if j>0 then
            begin
              textel:=copy(texte,1,j-1);
              delete(texte,1,j);
            end else
            begin
              textel:=texte;
              texte:='';
            end;
            if uppercase(textel)='ONOFF' then
            begin
              while length(texte)>0 do
              begin
                j:=pos('/',texte);
                if j>0 then
                begin
                  textel:=copy(texte,1,j-1);
                  delete(texte,1,j);
                end else
                begin
                  textel:=texte;
                  texte:='';
                end;
                if copy(textel,1,5)='Вихід=' then
                begin
                  delete(textel,1,5);
                  if uppercase(textel)='GLOBAL'
                    then FOnOff.Cde_Exit:=''
                    else FOnOff.Cde_Exit:=textel;
                end else
                if copy(textel,1,3)='ON=' then

```

```

begin
    delete(textel,1,3);
    FOnOff.Cde_ON:=textel;
end else
if copy(textel,1,4)='OFF=' then
begin
    delete(textel,1,4);
    FOnOff.Cde_OFF:=textel;
end else
if copy(textel,1,5)='I'мя=' then
begin
    delete(textel,1,5);
    FOnOff.Caption:=textel;
end else
if copy(textel,1,6)='Опис=' then
begin
    delete(textel,1,6);
    FOnOff.L_Pin.Caption:=textel;
end;
end; // end of while length(texte)>0 do
// ми конфігуруємо вікно
FOnOff.Enable_Screen;
FOnOff.Actif:=True;
FOnOff.Show;
end else // end of if uppercase(textel)='ONOFF' then
if uppercase(textel)='PWM' then
begin
    //CLK=,41/PWM/EXIT=GLOBAL/NAME=Динамічне управління
частотою/TITLE=OUT 4 - CN 5.24/MODE=FREQ/TYPE=16L/CDE=30 04 00 VALUE 00;
    FPWM.diviseur:=1;
    FPWM.Pas:=1;
    FPWM.sens_octet:=CT_SensHL;
    while length(texte)>0 do
        begin
            j:=pos('/',texte);
            if j>0 then
                begin
                    textel:=copy(texte,1,j-1);
                    delete(texte,1,j);
                end else
                begin
                    textel:=texte;
                    texte:='';
                end;
            if copy(textel,1,5)='Вихід =' then
                begin
                    delete(textel,1,5);
                    if uppercase(textel)='GLOBAL'
                        then FPWM.Cde_PWM_Exit:=''
                        else FPWM.Cde_PWM_Exit:=textel;
                    end else
            if copy(textel,1,5)='Тип =' then
                begin
                    delete(textel,1,5);
                    if pos('L',textel)>0 then
                        begin
                            FPWM.sens_octet:=CT_SensLH;
                            delete(textel,pos('L',textel),1);
                        end else
                    if pos('H',textel)>0 then
                        begin
                            delete(textel,pos('H',textel),1);
                        end;
                    try
                        FPWM.lg_data:=strtoint(textel);
                    except
                        FPWM.lg_data:=8;
                    end;
                end else

```

```

if copy(textel,1,4)='CDE=' then
begin
delete(textel,1,4);
FPWM.Cde_PWM:=textel;
end else
if copy(textel,1,4)='DIV=' then
begin
delete(textel,1,4);
FPWM.diviseur:=strtoreel(textel);
end else
if copy(textel,1,4)='Прохід =' then
begin
delete(textel,1,4);
FPWM.Pas:=strtoreel(textel);
end else
if copy(textel,1,5)='Модуль =' then
begin
delete(textel,1,5);
FPWM.Unite:=textel;
end else
if copy(textel,1,5)='Ім'я =' then
begin
delete(textel,1,5);
FPWM.Caption:=textel;
end else
if copy(textel,1,6)='Назва =' then
begin
delete(textel,1,6);
FPWM.L_Pin.Caption:=textel;
end;
end;
FPWM.Calcul_Min_Max;
FPWM.Actif:=True;
FPWM.Enable_Screen;
FPWM.B_Fermer.Caption:=MenuProg[CMF_Close];
FPWM.Show;
end else // end of if uppercase(textel)='PWM' then
if uppercase(textel)='BRIDGE' then
begin
// CLK=,76/BRIDGE/EXIT=GLOBAL/NAME=Управління
мостом n°3/TITLE=OUT 13 &
14/STOP=00/SENS1=01/SENS2=02/DIV=50/TYPE=8/UNIT=Hz/PAS=50/CDE=30 11 EXV SENS
VALUE;
FBridge.diviseur:=1;
FBridge.Pas:=1;
FBridge.sens_octet:=CT_SensHL;
while length(texte)>0 do
begin
j:=pos('/',texte);
if j>0 then
begin
textel:=copy(texte,1,j-1);
delete(texte,1,j);
end else
begin
textel:=texte;
texte:='';
end;
if copy(textel,1,5)='Вихід =' then
begin
delete(textel,1,5);
if uppercase(textel)='GLOBAL'
then FBridge.Cde_Exit:=''
else FBridge.Cde_Exit:=textel;
end else
if copy(textel,1,5)='Тип =' then
begin
delete(textel,1,5);
if pos('L',textel)>0 then

```

```

begin
    FBridge.sens_octet:=CT_SensLH;
    delete(textel,pos('L',textel),1);
end else
if pos('H',textel)>0 then
begin
    delete(textel,pos('H',textel),1);
end;
try
    FBridge.lg_data:=strtoint(textel);
except
    FBridge.lg_data:=8;
end;
end else
if copy(textel,1,4)='CDE=' then
begin
    delete(textel,1,4);
    FBridge.Cde_PWM:=textel;
end else
if copy(textel,1,5)='Умова остановки =' then
begin
    delete(textel,1,5);
    FBridge.Cde_Stop:=textel;
end else
if copy(textel,1,6)='SENS1=' then
begin
    delete(textel,1,6);
    FBridge.Cde_Sens1:=textel;
end else
if copy(textel,1,6)='SENS2=' then
begin
    delete(textel,1,6);
    FBridge.Cde_Sens2:=textel;
end else
if copy(textel,1,4)='DIV=' then
begin
    delete(textel,1,4);
    FBridge.diviseur:=strtoereel(textel);
end else
if copy(textel,1,4)='Прохід =' then
begin
    delete(textel,1,4);
    FBridge.Pas:=strtoereel(textel);
end else
if copy(textel,1,5)='Модуль =' then
begin
    delete(textel,1,5);
    FBridge.Label_Unite:=textel;
end else
if copy(textel,1,5)='Ім'я =' then
begin
    delete(textel,1,5);
    FBridge.Caption:=textel;
end else
if copy(textel,1,6)='Назва =' then
begin
    delete(textel,1,6);
    FBridge.L_Pin.Caption:=textel;
end;
end;
FBridge.Calcul_Mini_Maxi;
FBridge.Actif:=True;
FBridge.Enable_Screen;
FBridge.B_Fermer.Caption:=MenuProg[CMF_Close];
FBridge.Show;
end else // end of if uppercase(textel)='BRIDGE'
then
if uppercase(textel)='WIPER' then
begin

```

```

while length(texte)>0 do
begin
j:=pos('/',texte);
if j>0 then
begin
textel:=copy(texte,1,j-1);
delete(texte,1,j);
end else
begin
textel:=texte;
texte:='';
end;
if copy(textel,1,5)='Вихід =' then
begin
delete(textel,1,5);
if uppercase(textel)='GLOBAL'
then FWiper.Cde_Wiper_Exit:=''
else FWiper.Cde_Wiper_Exit:=textel;
end else
if copy(textel,1,3)='Умова остановки =' then
begin
delete(textel,1,3);
FWiper.Cde_Wiper_Stop:=textel;
end else
if copy(textel,1,4)='SLOW=' then
begin
delete(textel,1,4);
FWiper.Cde_Wiper_Slow:=textel;
end else
if copy(textel,1,4)='FAST=' then
begin
delete(textel,1,4);
FWiper.Cde_Wiper_Fast:=textel;
end else
if copy(textel,1,5)='Ім'я =' then
begin
delete(textel,1,5);
FWiper.Caption:=textel;
end else
if copy(textel,1,6)='Назва =' then
begin
delete(textel,1,6);
FWiper.L_Pin.Caption:=textel;
end;
end;
FWiper.Actif:=True;
FWiper.Enable_Screen;
FWiper.B_Fermer.Caption:=MenuProg[CMF_Close];
FWiper.Show;
end else // end of if uppercase(textel)='WIPER' then
if uppercase(textel)='BIPWM' then
begin
FBiPWM.diviseur_1:=1;
FBiPWM.Pas_1:=1;
FBiPWM.diviseur_2:=1;
FBiPWM.Pas_2:=1;
FBiPWM.sens_octet1:=CT_SensHL;
FBiPWM.sens_octet2:=CT_SensHL;
while length(texte)>0 do
begin
j:=pos('/',texte);
if j>0 then
begin
textel:=copy(texte,1,j-1);
delete(texte,1,j);
end else
begin
textel:=texte;
texte:='';

```

```

end;
if copy(textel,1,5)='Вихід =' then
begin
delete(textel,1,5);
if uppercase(textel)='GLOBAL'
then FBiPWM.Cde_Exit:=''
else FBiPWM.Cde_Exit:=textel;
end else
if copy(textel,1,6)='TYPE1=' then
begin
delete(textel,1,6);
if pos('L',textel)>0 then
begin
FBiPWM.sens_octet1:=CT_SensLH;
delete(textel,pos('L',textel),1);
end else
if pos('H',textel)>0 then
begin
delete(textel,pos('H',textel),1);
end;
try
FBiPWM.lg_data_1:=strtoint(textel);
except
FBiPWM.lg_data_1:=8;
end;
end else
if copy(textel,1,6)='TYPE2=' then
begin
delete(textel,1,6);
if pos('L',textel)>0 then
begin
FBiPWM.sens_octet2:=CT_SensLH;
delete(textel,pos('L',textel),1);
end else
if pos('H',textel)>0 then
begin
delete(textel,pos('H',textel),1);
end;
try
FBiPWM.lg_data_2:=strtoint(textel);
except
FBiPWM.lg_data_2:=8;
end;
end else
if copy(textel,1,4)='CDE=' then
begin
delete(textel,1,4);
FBiPWM.Cde_BiPWM:=textel;
end else
if copy(textel,1,5)='DIV1=' then
begin
delete(textel,1,5);
FBiPWM.diviseur_1:=strto reel(textel);
end else
if copy(textel,1,5)='DIV2=' then
begin
delete(textel,1,5);
FBiPWM.diviseur_2:=strto reel(textel);
end else
if copy(textel,1,5)='PAS1=' then
begin
delete(textel,1,5);
FBiPWM.Pas_1:=strto reel(textel);
end else
if copy(textel,1,6)='UNIT1=' then
begin
delete(textel,1,6);
FBiPWM.Label_Unit1:=textel;
end else

```

```

if copy(textel,1,5)='PAS2=' then
begin
delete(textel,1,5);
FBiPWM.Pas_2:=strtoereel(textel);
end else
if copy(textel,1,6)='UNIT2=' then
begin
delete(textel,1,6);
FBiPWM.Label_Unit2:=textel;
end else
if copy(textel,1,5)='Ім'я =' then
begin
delete(textel,1,5);
FBiPWM.Caption:=textel;
end else
if copy(textel,1,6)='Назва =' then
begin
delete(textel,1,6);
FBiPWM.L_Pin.Caption:=textel;
end;
end;
FBiPWM.Calcul_Min_Max;
FBiPWM.Actif:=True;
FBiPWM.Enable_Screen;
FBiPWM.B_Fermer.Caption:=MenuProg[CMF_Close];
FBiPWM.Show;
end else // end of if uppercase(textel)='BIPWM' then
if uppercase(textel)='VALUE' then
begin
FValue.Cde_Exit:='';
FValue.Cde_Send:='';
FValue.E_Value.Text:='';
FValue.diviseur:=1;
FValue.sens_octet:=CT_SensHL;
while length(texte)>0 do
begin
j:=pos('/',texte);
if j>0 then
begin
textel:=copy(texte,1,j-1);
delete(texte,1,j);
end else
begin
textel:=texte;
texte:='';
end;
//
VALUE/EXIT=GLOBAL/TYPE=8/NAME=Співвідношення/TITLE=xxx/DIV=1/CDE=21 40 VAL 00
if copy(textel,1,5)='Вихід =' then
begin
delete(textel,1,5);
if uppercase(textel)='GLOBAL'
then FValue.Cde_Exit:=''
else FValue.Cde_Exit:=textel;
end else
if copy(textel,1,5)='Тип =' then
begin
delete(textel,1,5);
if pos('L',textel)>0 then
begin
FValue.sens_octet:=CT_SensLH;
delete(textel,pos('L',textel),1);
end else
if pos('H',textel)>0 then
begin
delete(textel,pos('H',textel),1);
end;
try
FValue.lg_data:=strtoint(textel);

```

```

        except
            FValue.lg_data:=8;
        end;
    end else
    if copy(textel,1,4)='CDE=' then
    begin
        delete(textel,1,4);
        FValue.Cde_Send:=textel;
    end else
    if copy(textel,1,4)='DIV=' then
    begin
        delete(textel,1,4);
        FValue.diviseur:=strto reel (textel);
    end else
    if copy(textel,1,5)='Им'я =' then
    begin
        delete(textel,1,5);
        FValue.Caption:=textel;
    end else
    if copy(textel,1,6)='Назва =' then
    begin
        delete(textel,1,6);
        FValue.L_Pin.Caption:=textel;
    end else
    if copy(textel,1,5)='Вибір =' then
    begin
        delete(textel,1,5);
        j:=pos(',',textel);
        try
            FValue.ColGrid:=strto int (copy(textel,1,j-
1));
        except
            FValue.ColGrid:=-1;
        end;
        delete(textel,1,j);
        try
            FValue.RowGrid:=strto int (textel);
        except
            FValue.RowGrid:=-1;
        end;
        if (FValue.ColGrid>-1)and (FValue.RowGrid>-
1) then
            begin
FValue.E_Value.Text:=FMain.StringGrid1.Cells[FValue.ColGrid,FValue.RowGrid];
                end;
                FValue.L_Pin.Caption:=textel;
            end;
        end;

        FValue.Actif:=True;
        FValue.Enable_Screen;
        FValue.B_Fermer.Caption:=MenuProg[ CMP_Close ];
        FValue.Show;
    end else; // end of if uppercase(textel)='VALUE' then
    if uppercase(textel)='LIST' then
    begin
        FLIST.Cde_Exit:='';
        FList.Cde_Send:='';
        FList.E_Value.Text:='';
        while length(texte)>0 do
        begin
            j:=pos('/',texte);
            if j>0 then
            begin
                textel:=copy(texte,1,j-1);
                delete(texte,1,j);
            end else
            begin

```

```

        textel:=texte;
        texte:='';
    end;
    // CLK=,1/LIST/EXIT=/CASE=1,2-9/NAME=Введіть
    дату й час/
    // TITLE=Дата й
    час/COR=0.25|0:1|0:1|0:1|0:0.25|0:1|0:1|0:1|0/
    // TYPE=8:8:8:8:8:8/CDE=2E F9 0B VALUE;
    if copy(textel,1,5)='Вихід =' then
    begin
        delete(textel,1,5);
        if uppercase(textel)='GLOBAL'
            then FList.Cde_Exit:=''
            else FList.Cde_Exit:=textel;
        end else
    if copy(textel,1,5)='Тип =' then
    begin
        delete(textel,1,5);
        FList.type_value:=textel;
    end else
    if copy(textel,1,4)='CDE=' then
    begin
        delete(textel,1,4);
        FList.Cde_Send:=textel;
    end else
    if copy(textel,1,4)='COR=' then
    begin
        delete(textel,1,4);
        FList.correction:=textel;
    end else
    if copy(textel,1,5)='Ім'я =' then
    begin
        delete(textel,1,5);
        FList.Caption:=textel;
    end else
    if copy(textel,1,6)='Назва =' then
    begin
        delete(textel,1,6);
        FList.L_Pin.Caption:=textel;
    end else
    if copy(textel,1,5)='Вибір =' then
    begin
        delete(textel,1,5);

FList.CaseToInteger(textel,FList.ListColMin,FList.ListColMax,FList.ListRowMin,FList.ListRowMax);

FList.StringGrid1.ColCount:=FMain.StringGrid1.ColCount;
        j:=FList.ListRowMax-FList.ListRowMin+2;
        FList.StringGrid1.RowCount:=j;
        for j:=0 to FList.StringGrid1.ColCount-1 do

FList.StringGrid1.Cells[j,0]:=FMain.StringGrid1.Cells[j,0];
        for j:=0 to FList.StringGrid1.ColCount-1 do
            for k:=FList.ListRowMin to
FList.ListRowMax do
                begin
                    FList.StringGrid1.Cells[j,k-
FList.ListRowMin+1]:=FMain.StringGrid1.Cells[j,k];
                end;
            end;
        end;
    end;

    FList.Actif:=True;
    FList.Enable_Screen;
    FList.B_Fermer.Caption:=MenuProg[CMF_Close];
    FList.Show;
    end; // end of if uppercase(textel)='LIST' then
end; // end of while length(texte)>0 do

```



```

if copy(texte,1,3)='GRE' then
begin
  if copy(texte,6,1)='1' then
  begin
    FMain.StringGrid1.Canvas.Draw(bRect.Left,bRect.Top,ledverteon);
    //FMain.StringGrid1.Canvas.Brush.Color:=couleur;
    //FMain.StringGrid1.Canvas.FillRect(aRect);
  end else
  begin
    FMain.StringGrid1.Canvas.Draw(bRect.Left,bRect.Top,ledverteoff);
    //FMain.StringGrid1.Canvas.Brush.Color:=clGreen;
    //FMain.StringGrid1.Canvas.FillRect(aRect);
  end;
end;
end else
begin
  FMain.StringGrid1.Canvas.Font.Color:=clBlack;

FMain.StringGrid1.Canvas.TextOut(aRect.Left,aRect.Top,FMain.StringGrid1.Cells[aC
ol,aRow]);
end;

end;

//*****
procedure TFMain.StringGrid1Resize(Sender: TObject);
begin
  //FMain.rafraichir_diag;
end;

//*****
procedure TFMain.T_AttenteTimer(Sender: TObject);
begin
  FMain.num_image:=FMain.num_image+1;
  if FMain.num_image>14 then FMain.num_image:=1;
  FMain.I_Attente.Picture.Bitmap:=nil;
  FMain.IList_Rose.GetBitmap(FMain.num_image,FMain.I_Attente.Picture.Bitmap);
end;

procedure TFMain.T_CommandTimer(Sender: TObject);
var texte,textel : string;
    liste_connection : TStringList;
    ListCommand : TStringList;
    cond : string; // вимога повторити команду
    cde : string; // команда відправки
    i,j,k,l,m,n : integer;
    outdoor : boolean;
    byteRule: string;
    DebutByte,FinByte,DebutBit,FinBit : integer;
    valeur,val_cond : integer;
    f_Byte,f_Bit : string;
    mask : byte;
    n_ligne,max_val,min_val : integer;
    Noeud : TTreeNode;
begin
  // Виконати всі команди пошуку відразу після з'єднання
  FMain.T_Command.Enabled:=False;
  try
    liste_Connection:=TStringlist.Create;
    ListCommand:=TStringList.Create;
    FMain.Search_Item('CONNECTION',FMain.Vehicule,Liste_Connection);
    //Liste_Connection.SaveToFile('c:\Connection.txt');
    i:=0;
    FMain.TypeAff:=-1;
    while i<Liste_Connection.Count do
    begin
      Application.ProcessMessages;
      if MainExit then exit;
      texte:=Liste_Connection.Strings[i];

```

```

if uppercase(copy(texte,1,9))='<COMMAND>' then
begin
ListCommand.Clear;
while (texte<>'</COMMAND>') and (i<Liste_Connection.Count)do
begin
texte:=Liste_Connection.Strings[i];
texte:=trim(texte);
ListCommand.Add(texte);
if texte<>'</COMMAND>' then i:=i+1;
end;
//ListCommand.SaveToFile('c:\Command'+inttostr(i)+'.txt');
// ми отримуємо все поле COMMAND
// Таким чином, ми виконаємо команду
j:=0;
while j<ListCommand.Count do
begin
Application.ProcessMessages;
if MainExit then exit;
texte:=trim(ListCommand.Strings[j]);
if copy(texte,1,5)='COND=' then
begin
delete(texte,1,5);
cond:=texte;
end else
if copy(texte,1,5)='<_ISO_14230_>' then
begin
FMain.Actual_ISO_14230.Clear;
cde:='';
while j<ListCommand.Count do
begin
texte:=trim(ListCommand.Strings[j]);
if copy(texte,1,2)<>'</' then
FMain.Actual_ISO_14230.Add(texte);
if copy(texte,1,4)='CDE=' then cde:=texte;
if copy(texte,1,2)='</' then j:=ListCommand.Count;
j:=j+1;
end;
end;
j:=j+1;
end; // end of while j<ListCommand.Count do
end; // end of if uppercase(copy(texte,1,9))='<COMMAND>' then
if length(cde)>0 then
begin
delete(cde,1,4);
FMain.ISO_14230.Sid:=hextohex(copy(cde,1,2));
delete(cde,1,3);
j:=0;
while length(cde)>0 do
begin
Application.ProcessMessages;
if MainExit then Exit;
FMain.ISO_14230.FData[j]:=hextohex(copy(cde,1,2));
delete(cde,1,3);
j:=j+1;
end;
FMain.ISO_14230.SendLength:=j;
if length(cond)>0 then
begin
outdoor:=false;
// Розрахунок фільтра на дані
j:=pos('/',cond);
ByteRule:=copy(cond,1,j-1);
delete(cond,1,j);
// видаляє невідфільтровані дані
j:=pos('[',ByteRule);
if j>0 then delete(ByteRule,1,j);
j:=pos(']',ByteRule);
if j>0 then delete(ByteRule,j,1);
j:=pos('|',ByteRule);

```

```

// Отримує дані з бітів
if j>0 then
begin
  f_Byte:=copy(ByteRule,1,j-1);
  delete(ByteRule,1,j);
  f_bit:=ByteRule;
end else
begin
  f_Byte:=ByteRule;
  f_bit:='';
end;
j:=pos('-',f_Byte);
if j>0 then
begin
  textel:=copy(f_Byte,1,j-1);
  try
    DebutByte:=Strtoint(textel);
  except
    DebutByte:=0;
  end;
  delete(f_byte,1,j);
  try
    FinByte:=Strtoint(f_Byte);
  except
    FinByte:=0;
  end;
end else
begin
  try
    DebutByte:=Strtoint(f_Byte);
  except
    DebutByte:=0;
  end;
  FinByte:=DebutByte;
end;
// отримує площину біт
if length(f_bit)>0 then // відфільтровує біти
begin
  j:=pos('-',f_Bit);
  if j>0 then // є площина
  begin
    textel:=copy(f_Bit,1,j-1);
    try
      DebutBit:=Strtoint(textel);
    except
      DebutBit:=0;
    end;
    delete(f_bit,1,j);
    try
      FinBit:=Strtoint(f_Bit);
    except
      FinBit:=0;
    end;
  end else
  begin //Немає площини
    try
      DebutBit:=Strtoint(f_Bit);
    except
      DebutBit:=0;
    end;
    FinBit:=DebutBit;
  end;
  // відновлення створення маски
  mask:=0;
  for j:=DebutBit to FinBit do
    mask:=mask or (1 shl j);
  end else mask:=$FF;
  texte:=cond;
  delete(texte,1,1);

```

```

try
    val_cond:=strtoint(texte);
except
    outdoor:=true;
end;
// повторення виконання
beginExit:=false;
While (outdoor=False) and (BeginExit=False) do
begin
    Application.ProcessMessages;
    if MainExit then Exit;
    FMain.ISO_14230.SendResponse;
    FMain.Attente_Start;
    Execution:=False;
    While (Execution=False)
        and (MainExit=False) do Application.ProcessMessages;
    if MainExit then Exit;
    // визначення фільтра для байта
    valeur:=0;
    for j:=DebutByte to FinByte do
        begin
            valeur:=valeur*256+(FMain.ISO_14230.TrameRx[j] and
mask);

            end;
        if copy(cond,1,1)='=' then
            begin
                if valeur=val_cond then outdoor:=true;
            end else
        if copy(cond,1,1)='>' then
            begin
                if valeur>val_cond then outdoor:=True;
            end else
        if copy(cond,1,1)='<' then
            begin
                if valeur<val_cond then outdoor:=True;
            end else outdoor:=True; // є помилка

        end;
    end else
    begin
        FMain.ISO_14230.SendResponse;
        FMain.Attente_Start;
        Execution:=False;
        While (Execution=False)
            and (MainExit=False) do Application.ProcessMessages;
        if MainExit then Exit;
        end;
        end;
        i:=i+1;
    end; // end of while i<ListConnection.Count do
finally
    liste_Connection.Free;
    ListCommand.Free;
end;
// Виведення меню
i:=0;
while i<Vehicule.Count do
    begin
        Application.ProcessMessages;
        if MainExit then Exit;
        texte:=Vehicule.Strings[i];
        trim(texte);
        if uppercase(copy(texte,1,4))='ICO=' then
            begin
                //noeud:=FMain.TreeView1.Items.GetFirstNode;
                if noeud<>nil then
                    begin
                        delete(texte,1,4);
                        j:=pos('; ',texte);

```

```

    try
      j:=strtoint(copy(texte,1,j-1));
    except
      j:=-1;
    end;
    noeud.ImageIndex:=j;
  end;
end else
if uppercase(copy(texte,1,8))='<EXMENU=' then
begin
  DebutByte:=i;
  FMain.ExtendedMenu.Clear;
  delete(texte,2,2);
  FMain.ExtendedMenu.Add(texte);
  // відновлення розширеного меню, операція повинна бути повторена
  outdoor:=False;
  i:=i+1;
  while outdoor=False do
  begin
    Application.ProcessMessages;
    if MainExit then Exit;
    if i<Vehicule.Count then texte:=Vehicule.Strings[i];
    trim(texte);
    if (i>=Vehicule.Count)
      or (uppercase(copy(texte,1,8))='<EXMENU=')
      or (uppercase(copy(texte,1,8))='</EXMENU')
      or (uppercase(copy(texte,1,6))='<MENU=') then outdoor:=True;
    if not outdoor then
    begin
      if (length(texte)>0)
        and (copy(texte,1,2)<>'//') then
      begin
        k:=pos(';',texte);
        if k=0 then FMain.ExtendedMenu.Add(texte) else
FMain.ExtendedMenu.Add(copy(texte,1,k));
        end;
        i:=i+1;
      end else FinByte:=i;
    end;
    //FMain.ExtendedMenu.SaveToFile('c:\ExtendedMenu.txt');
    // Видалення з розширеного меню файлу з даними автомобіля
    for j:=DebutByte to FinByte do
    begin
      FMain.Vehicule.Delete(DebutByte);
    end;
    // ми повинні відновити умови повторення val_cond
    j:=0;
    while j<FMain.ExtendedMenu.Count do
    begin
      Application.ProcessMessages;
      if MainExit then Exit;
      texte:=FMain.ExtendedMenu.Strings[j];
      if copy(texte,1,9)='EXREPEAT=' then
      begin
        delete(texte,1,9);
        k:=pos(';',texte);
        if k>0 then texte:=copy(texte,1,k-1);
        texte:=trim(texte);
        try
          val_cond:=strtoint(texte); // повторення змінної
        except
          val_cond:=-1;
        end;
        FMain.ExtendedMenu.Delete(j); // Стираємо це попередження
        j:=FMain.ExtendedMenu.Count;
      end;
      j:=j+1;
    end;
  end;
end;

```



```

delete(texte,1,debutbit);
debutbit:=pos(']',texte);
valeur:=strtoint(copy(texte,1,debutbit-1));
delete(texte,1,debutbit-1);
if max_val<valeur then max_val:=valeur;
if min_val>valeur then min_val:=valeur;
valeur:=valeur+n_ligne;
texte:=cond+inttostr(valeur)+texte;
end;
end;
FMain.Vehicule.Insert(DebutByte+k+((j-
1)*FMain.ExtendedMenu.Count),texte);
end;
// розраховуємо l
// n_ligne => текст у таблицю
// valeur => читати останнє значення
n_ligne:=n_ligne+(max_val-min_val)+1;
end;
end;
// Ми закінчили підключення, це переводить курсор на початок цієї
області
i:=DebutByte-1;
end else
if uppercase(copy(texte,1,6))='<MENU=' then
begin
delete(texte,1,6);
j:=pos('>',texte);
if j>0 then
begin
MenuList.Add(copy(texte,1,j-1)+'/'+inttostr(i)+'/');
noeud:=FMain.TreeView1.Items.Add(nil,copy(texte,1,j-1));
end;
end else
if uppercase(copy(texte,1,6))='</MENU' then
begin
delete(texte,1,6);
if MenuList.Count>0 then MenuList.Strings[MenuList.Count-
1]:=MenuList.Strings[MenuList.Count-1]+inttostr(i)+'/';
end ;
i:=i+1;
end;
//FMain.TreeView1.Enabled:=True;
FMain.Vehicule.SaveToFile('C:\vehicule.txt');
FMain.TreeView1.Enabled:=True;
end;

//*****
// Цей таймер використовується для виконання всіх _ISO_14230_ Cde меню, якщо
треба повторити умови, цей таймер є виконуваний іншим разом
//*****
procedure TFMain.T_StatusIOTimer(Sender: TObject);
var i,j : integer;
    texte : string;
    outdoor : boolean;
begin
InIOTimer:=True;
FMain.T_StatusIO.Enabled:=False;
i:=FMain.ligneMenu;
texte:='';
if length(FOnOff.Buffer)>0 then
begin
    texte:=FOnOff.Buffer;
    FOnOff.Buffer:='';
end else
if length(FBiPWM.Buffer)>0 then
begin
    texte:=FBiPWM.Buffer;
    FBiPWM.Buffer:='';
end else

```

```

if length(FPwm.Buffer)>0 then
begin
  texte:=FPWM.Buffer;
  FPWM.Buffer:='';
end else
if length(FBridge.Buffer)>0 then
begin
  texte:=FBridge.Buffer;
  FBridge.Buffer:='';
end else
if length(Fvalue.Buffer)>0 then
begin
  texte:=FValue.Buffer;
  FValue.Buffer:='';
  if texte='#' then
  begin
    // Створюємо запит на читання
    FMain.TreeView1Click(nil);
    exit;
  end;
end else
if length(FList.Buffer)>0 then
begin
  texte:=FList.Buffer;
  FList.Buffer:='';
  if texte='#' then
  begin
    // Створюємо запит на читання
    FMain.TreeView1Click(nil);
    exit;
  end;
end else
if length(Fwiper.Buffer)>0 then
begin
  texte:=FWiper.Buffer;
  FWiper.Buffer:='';
end;
// якщо користувач нажимає на кнопку
if length(texte)>0 then
begin
  While (FMain.ISO_14230.EndReceive=False) and (MainExit=False) do
Application.ProcessMessages;
  if MainExit then Exit;
  FMain.Cde2Send(texte);
  FMain.Attente_Start;
  FMain.Execution:=False;
  //while FMain.Execution= false do Application.ProcessMessages;
  FMain.NoReceive:=True;
  FMain.ISO_14230.SendResponse;
  FOnOff.Buffer:='';
  if (i<FMain.ActualMenu.Count) or
    ((FMain.TimeRepeat>0) and (FMain.TypeAff<>CT_AFFCONFIRM)) then
  begin
    FMain.T_StatusIO.Interval:=100;
    FMain.T_StatusIO.Enabled:=True;
  end;
  InIoTimer:=False;
  exit;
end;
// ми отримуємо поточну команду по протоколу _ISO_14230_
outdoor:=False;
FMain.Actual_ISO_14230.Clear;
if (i<FMain.ActualMenu.Count) then
begin

  // ми отримуємо поточну команду по протоколу _ISO_14230_
  While (i<FMain.ActualMenu.Count) and (OutDoor=False) do
  begin
    Application.ProcessMessages;

```

```

if MainExit then Exit;
texte:=trim(FMain.ActualMenu.Strings[i]);
if uppercase(copy(texte,1,5))='<_ISO_14230_>' then
begin
  i:=i+1;
  while (i<FMain.ActualMenu.Count) and (outdoor=False) do
  begin
    texte:=trim(FMain.ActualMenu.Strings[i]);
    if uppercase(texte)<>'</_ISO_14230_>' then
    begin
      Actual_ISO_14230.Add(texte);
      i:=i+1;
    end else outdoor:=TRUE;
  end;
end;
i:=i+1;
end;
if outdoor=False then
begin
  If (FMain.TimeRepeat>0) and (FMain.TypeAff<>CT_AFFCONFIRM) then
  begin
    FMain.ligneMenu:=0;
    //i:=0;
    FMain.T_StatusIO.Interval:=FMain.TimeRepeat;
    FMain.T_StatusIO.Enabled:=True;
    {
    While (i<FMain.ActualMenu.Count) and (OutDoor=False) do
    begin
      Application.ProcessMessages;
      texte:=trim(FMain.ActualMenu.Strings[i]);
      if uppercase(copy(texte,1,5))='<_ISO_14230_>' then
      begin
        i:=i+1;
        while (i<FMain.ActualMenu.Count) and (outdoor=False) do
        begin
          texte:=trim(FMain.ActualMenu.Strings[i]);
          if uppercase(texte)<>'</_ISO_14230_>' then
          begin
            Actual_ISO_14230.Add(texte);
            i:=i+1;
          end else outdoor:=TRUE;
        end;
      end;
      i:=i+1;
    end;
    }
  end;
  InIOTimer:=False;
  exit;
end;

//FMain.Actual_ISO_14230.SaveToFile('c:\Actual_ISO_14230_'+inttostr(i)+'.txt');
// Записуємо дані з _ISO_14230_
// Посилаємо команду в
outdoor:=False;
j:=0;
while (j<FMain.Actual_ISO_14230.Count) and (outdoor=False) do
begin
  texte:=FMain.Actual_ISO_14230.Strings[j];
  if uppercase(copy(texte,1,4))='CDE=' then
  begin
    outdoor:=True;
  end;
  j:=j+1;
end;
// Виконуємо команду
delete(texte,1,4);
texte:=trim(texte);

```

```

    while (FMain.Execution= false) and (MainExit=False) do
Application.ProcessMessages;
    if MainExit then Exit;
    FMain.ISO_14230.SendLength:=0;
    if FMain.TypeAff=CT_AFFCONFIRM then
        begin
            if MessageDlg(FMain.TitleMenu,mtConfirmation, [mbYes, mbNo],0)= mrYes
then
                begin
                    if length(texte)>0 then
                        begin
                            FMain.Cde2Send(texte);
                        end;
                    FMain.Attente_Start;
                    FMain.Execution:=False;
                    //while FMain.Execution= false do Application.ProcessMessages;
                    FMain.ISO_14230.SendResponse;
                    end;
                    InIOTimer:=False;
                    exit;
                end else
    if FMain.TypeAff=CT_AFFONEPARAMETER then
        begin
            InIOTimer:=False;
            exit;
        end;
        if length(texte)>0 then
            begin
                FMain.Cde2Send(texte);
                FMain.Attente_Start;
                FMain.Execution:=False;
                //while FMain.Execution= false do Application.ProcessMessages;
                FMain.ISO_14230.SendResponse;
            end;
            FMain.ligneMenu:=i;

            //FMain.T_StatusIO.Enabled:=True;
        end else
        begin
            If (FMain.TimeRepeat>0) and (FMain.TypeAff<>CT_AFFCONFIRM) then
                begin
                    FMain.ligneMenu:=0;
                    FMain.T_StatusIO.Interval:=FMain.TimeRepeat;
                    FMain.T_StatusIO.Enabled:=True;
                end;
            end;
            InIOTimer:=False;
        end;

//*****
procedure TFMMain.TreeView1Click(Sender: TObject);
var texte,textel : string;
    i,j,k,l,m,n : integer;
    ligne : integer;
begin
    if FMain.TreeView1.Selected=nil then exit;
    //if FMain.Menu_Connection.Caption<>menuprog[CMF_Deconnecter] then exit;
    FMain.T_StatusIO.Enabled:=False; // stop the engine of _ISO_14230_ command
    texte:=FMain.TreeView1.Selected.Text;
    MenuName:=texte;
    MenuItem:=FMain.TreeView1.Selected.AbsoluteIndex;
    FMain.ClickTree:=True;
    FMain.Panel_Parametre.Visible:=False;
    FMain.Panel_grille.Visible:=False;
    FMain.Panel_Status.Visible:=False;
    FMain.Panel_Liste_Para.Visible:=False;
    FMain.Panel_Command.Visible:=False;
    FMain.Panel_Value.Visible:=False;
    FMain.Grille_Up:=0;

```

```

if MenuI=0 then
begin
// Ми знаходимося в меню «Статус»
FMain.Panel_Parametre.Visible:=False;
FMain.Panel_grille.Visible:=False;
FMain.Panel_Status.Visible:=True;
FMain.Panel_Liste_Para.Visible:=False;
FMain.Panel_Command.Visible:=False;
FMain.ClickTree:=False;
end else // end if MenuI=0
if MENU=1 then
begin
// команда управління ISO 14230
FMain.Panel_Parametre.Visible:=False;
FMain.Panel_grille.Visible:=False;
FMain.Panel_Status.Visible:=False;
FMain.Panel_Liste_Para.Visible:=False;
FMain.Panel_Command.Visible:=True;
FMain.TypeAff:=CT_AFFNONE;
FMain.E_Sid.Text:='';
FMain.E_Data.Text:='';
FMain.M_Hist.Clear;
FMain.ClickTree:=False;
end else
begin
// меню діагностики
ActualMenu.Clear;
FMain.Search_Item('MENU='+MenuName, Vehicule, ActualMenu);
ClickList.Clear;
FMain.Search_Item('CLICK', ActualMenu, ClickList); // On recupère les
ивнієments click sur la grille
// Діагностика помилок
i:=0;
FMain.ClickGlobal:='';
FMain.ClickExit:='';
FMain.B_Mode_Maitre.Visible:=False;
While i<ClickList.Count do
begin
texte:=trim(ClickList.Strings[i]);
if uppercase(copy(texte,1,5))='Вихід =' then
begin
delete(texte,1,5);
FMain.ClickExit:=texte;
FMain.B_Mode_Maitre.Visible:=True;
end else

if uppercase(copy(texte,1,6))='ACTIF=' then
begin
delete(texte,1,6);
FMain.ClickGlobal:=texte;
end;
i:=i+1;
end;
// Шукаємо тип Panel
i:=0;
//ActualMenu.SaveToFile('C:\ActualMenu_'+MenuName+'.txt');
while (i<ActualMenu.Count) do
begin
Application.ProcessMessages;
if MainExit then Exit;
texte:=trim(ActualMenu.Strings[i]);
if uppercase(copy(texte,1,5))='Тип =' then
begin
delete(texte,1,5);
if uppercase(copy(texte,1,4))='GRID' then
begin

FMain.TypeAff:=CT_AFFGRID;
delete(texte,1,5);

```

```

texte:=trim(texte);
//FMain.StringGrid1.Clean;
//FMain.StringGrid3.Clean;
try
  j:=strtoint(texte);
except
  j:=1;
end;
FMain.StringGrid1.ColCount:=j;
//j:=FMain.StringGrid1.ColCount;
//FMain.StringGrid3.ColCount:=j+2;
//j:=FMain.StringGrid3.ColCount;
FMain.Panel_grille.Visible:=True;
FMain.StringGrid1.RowCount:=1;
//FMain.StringGrid3.RowCount:=1;
FMain.Ajust_StringList(1,FMain.ColorList);
// Ми шукаємо рядки, що починаються з LG=
while (i<ActualMenu.Count) do
begin
  //Application.ProcessMessages;
  texte:=trim(ActualMenu.Strings[i]);
  if uppercase(copy(texte,1,3))='LG=' then
  begin
    // Ми записуємо отримані дані у буфер
    delete(texte,1,3); // видаляємо LG=
    delete(texte,1,4); // видаляємо ROW=
    j:=pos('/',texte);
    try
      ligne:=strtoint(copy(texte,1,j-1));
    except
      ligne:=1;
    end;
    if ligne>=FMain.StringGrid1.RowCount then
    begin
      k:=FMain.StringGrid1.RowCount;
      FMain.StringGrid1.RowCount:=ligne+1;
      for l := k to FMain.StringGrid1.RowCount - 1 do
        for m := 0 to FMain.StringGrid1.ColCount-1 do
          FMain.StringGrid1.Cells[m,l]:='';
          FMain.Ajust_StringList(ligne+1,FMain.ColorList);
        end;
      delete(texte,1,j);
      while length(texte)>0 do
      begin
        j:=pos('/',texte);
        if j=0 then j:=pos('; ',texte);
        if j>0 then
        begin
          textel:=copy(texte,1,j-1);
          delete(texte,1,j);
        end else
        begin
          textel:=texte;
          texte:='';
        end;
        if copy(textel,1,3)='CL=' then
        begin
          delete(textel,1,3);
        end;
      end;
      //FMain.StringGrid3.Cells[FMain.StringGrid3.ColCount-2,ligne]:=textel;
      FMain.ColorList.Strings[ligne]:=textel;
    end else // fin if copy(textel,1,3)='CL=' then
    if copy(textel,1,2)='GR' then
    begin
      delete(textel,1,2);
      k:=pos('=',textel);
      try
        l:=strtoint(copy(textel,1,k-1));
      except

```

```

        l:=0;
    end;
    delete(textel,1,k);
    if l<FMain.StringGrid1.ColCount then
    begin
        if uppercase(copy(textel,1,4))='TAB[' then
        begin
            delete(textel,1,4);
            k:=pos(',',textel);
            if k>0 then
            begin
                try
                    m:=strtoint(copy(textel,1,k-1));
                except
                    m:=-1;
                end;
                delete(textel,1,k);
                k:=pos(']',textel);
                try
                    n:=strtoint(copy(textel,1,k-1));
                except
                    n:=-1;
                end;
                delete(textel,1,k);
                if (m>-1) and (m<10) then
                begin
                    if FMain.Table[m]<>nil then
                    begin
                        if (n>-1) and
(n<FMain.Table[m].Count) then
                        begin
                            textel:=FMain.Table[m].Strings[n];
                        end else
                            textel:=MenuProg[ CMP_ERRORLIGNETABLE ];
                        end else
                            textel:=MenuProg[ CMP_TableUnknow ];
                        end else
                            textel:=MenuProg[ CMP_ERRORNUMBERTABLE ];
                        end;
                    FMain.StringGrid1.Cells[l,ligne]:=textel;
                end else
                begin
                    FMain.StringGrid1.Cells[l,ligne]:=textel;
                end;
            end;
        end;
    end; // end of if uppercase(copy(texte,1,3))='LG=' then

        i:=i+1;
    end; // end of while (i<ActualMenu.Count) do
    if FMain.StringGrid1.RowCount>1 then
    FMain.StringGrid1.FixedRows:=1;
    end else
    if uppercase(copy(texte,1,7))='CONFIRM' then
    begin
        FMain.Panel_Parametre.Visible:=False;
        FMain.Panel_grille.Visible:=False;
        FMain.Panel_Status.Visible:=True;
        FMain.Panel_Liste_Para.Visible:=False;
        FMain.Panel_Command.Visible:=False;
        FMain.Panel_Value.Visible:=False;
        FMain.TypeAff:=CT_AFFCONFIRM;
    end else
    if uppercase(copy(texte,1,12))='ONEPARAMETER' then
    begin

```

```

FMain.Panel_Parametre.Visible:=False;
FMain.Panel_grille.Visible:=False;
FMain.Panel_Status.Visible:=False;
FMain.Panel_Liste_Para.Visible:=False;
FMain.Panel_Command.Visible:=False;
FMain.Panel_Value.Visible:=True;
FMain.TypeAff:=CT_AFFONEPARAMETER;
FMain.E_Para.Text:='';
FMain.E_Para_Numero.Text:='';
FMain.M_Parametre.Clear;
FMain.B_Para_Lecture.Caption:='';
FMain.B_Para_Ecriture.Caption:='';
FMain.CdeReadParam:='';
FMain.CdeWriteParam:='';
FMain.CdeValueLength:='';
while (i<ActualMenu.Count) do
begin
  Application.ProcessMessages;
  if MainExit then Exit;
  // шыкаемо CDE_READ,CDE_WRITE,LABEL_READ,LABEL_WRITE
  texte:=trim(ActualMenu.Strings[i]);
  if uppercase(copy(texte,1,11))='LABEL_READ=' then
  begin
    delete(texte,1,11);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.B_Para_Lecture.Caption:=texte;
  end else
  if uppercase(copy(texte,1,12))='LABEL_WRITE=' then
  begin
    delete(texte,1,12);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.B_Para_Ecriture.Caption:=texte;
  end else
  if uppercase(copy(texte,1,9))='CDE_READ=' then
  begin
    delete(texte,1,9);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.CdeReadParam:=texte;
  end else
  if uppercase(copy(texte,1,10))='CDE_WRITE=' then
  begin
    delete(texte,1,10);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.CdeWriteParam:=texte;
  end else
  if uppercase(copy(texte,1,7))='LENGTH=' then
  begin
    delete(texte,1,7);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    CdeValueLength:=texte;
  end;
  i:=i+1;
end;
if length(FMain.CdeReadParam)=0
then FMain.B_Para_Lecture.Visible:=False
else FMain.B_Para_Lecture.Visible:=true;
if length(FMain.CdeWriteParam)=0
then FMain.B_Para_ecriture.Visible:=False
else FMain.B_Para_ecriture.Visible:=true;
end else
if uppercase(copy(texte,1,8))='ONEVALUE' then
begin
  FMain.Panel_Parametre.Visible:=True;
  FMain.Panel_grille.Visible:=False;

```

```

FMain.Panel_Status.Visible:=False;
FMain.Panel_Liste_Para.Visible:=False;
FMain.Panel_Command.Visible:=False;
FMain.Panel_Value.Visible:=False;
FMain.TypeAff:=CT_AFFONEPARAMETER;
FMain.E_Para.Text:='';
FMain.E_Para_Numero.Text:='';
FMain.M_Parametre.Clear;
FMain.B_Para_Lecture.Caption:='';
FMain.B_Para_Ecriture.Caption:='';
FMain.CdeReadParam:='';
FMain.CdeWriteParam:='';
FMain.CdeValueLength:='';
while (i<ActualMenu.Count) do
begin
  Application.ProcessMessages;
  if MainExit then Exit;
  // шыкаемо CDE_READ, CDE_WRITE, LABEL_READ, LABEL_WRITE
  texte:=trim(ActualMenu.Strings[i]);
  if uppercase(copy(texte,1,11))='LABEL_READ=' then
  begin
    delete(texte,1,11);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.B_Para_Lecture.Caption:=texte;
  end else
  if uppercase(copy(texte,1,12))='LABEL_WRITE=' then
  begin
    delete(texte,1,12);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.B_Para_Ecriture.Caption:=texte;
  end else
  if uppercase(copy(texte,1,9))='CDE_READ=' then
  begin
    delete(texte,1,9);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.CdeReadParam:=texte;
  end else
  if uppercase(copy(texte,1,10))='CDE_WRITE=' then
  begin
    delete(texte,1,10);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.CdeWriteParam:=texte;
  end else
  if uppercase(copy(texte,1,7))='LENGTH=' then
  begin
    delete(texte,1,7);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    CdeValueLength:=texte;
  end;
  i:=i+1;
end;
if length(FMain.CdeReadParam)=0
then FMain.B_Valeur_Lecture.Visible:=False
else FMain.B_Valeur_Lecture.Visible:=true;
if length(FMain.CdeWriteParam)=0
then FMain.B_Valeur_ecriture.Visible:=False
else FMain.B_Valeur_ecriture.Visible:=true;
end else
if uppercase(copy(texte,1,10))='MULTIVALUE' then
begin
  FMain.Panel_Parametre.Visible:=False;
  FMain.Panel_grille.Visible:=False;
  FMain.Panel_Status.Visible:=False;
  FMain.Panel_Liste_Para.Visible:=True;

```

```

FMain.Panel_Command.Visible:=False;
FMain.TypeAff:=CT_AFFMultiVALUE;
while (i<ActualMenu.Count) do
begin
  Application.ProcessMessages;
  if MainExit then Exit;
  // шыкаемо CDE_READ,CDE_WRITE,LABEL_READ,LABEL_WRITE
  texte:=trim(ActualMenu.Strings[i]);
  if uppercase(copy(texte,1,11))='LABEL_READ=' then
  begin
    delete(texte,1,11);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.B_Lecture_Liste.Caption:=texte;
  end else
  if uppercase(copy(texte,1,12))='LABEL_WRITE=' then
  begin
    delete(texte,1,12);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.B_Sauve_Liste.Caption:=texte;
  end else
  if uppercase(copy(texte,1,9))='CDE_READ=' then
  begin
    delete(texte,1,9);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.CdeReadParam:=texte;
  end else
  if uppercase(copy(texte,1,10))='CDE_WRITE=' then
  begin
    delete(texte,1,10);
    j:=pos('; ',texte);
    if j>0 then texte:=copy(texte,1,j-1);
    FMain.CdeWriteParam:=texte;
  end;
  i:=i+1;
end;
end;
i:=ActualMenu.Count;
end;
i:=i+1;
end;

// Шыкаемо the title
i:=0;
while (i<ActualMenu.Count) do
begin
  Application.ProcessMessages;
  if MainExit then Exit;
  texte:=trim(ActualMenu.Strings[i]);
  if uppercase(copy(texte,1,6))='Назва =' then
  begin
    delete(texte,1,6);
    texte:=trim(texte);
    FMain.TitleMenu:=texte;
    if FMain.TypeAff=CT_AFFGRID then
    begin
      For k:=0 to FMain.StringGrid1.ColCount-1 do
FMain.StringGrid1.ColWidths[k]:=16;
      if length(texte)>0 then
        if texte[length(texte)]<>'/' then texte:=texte+'/' ;
      j:=0;
      while length(texte)>0 do
      begin
        k:=pos('/',texte);
        if k>0 then
          begin
            if j<FMain.StringGrid1.ColCount then

```

```

begin
    //FMain.StringGrid3.Cells[j,0]:=copy(texte,1,k-1);
    FMain.StringGrid1.Cells[j,0]:=copy(texte,1,k-1);
    if
FMain.StringGrid1.ColWidths[j]<FMain.StringGrid1.Canvas.TextWidth(copy(texte,1,k
-1))+32
        then
FMain.StringGrid1.ColWidths[j]:=FMain.StringGrid1.Canvas.TextWidth(copy(texte,1,
k-1))+32;
            end;
            j:=j+1;
            delete(texte,1,k);
            end else texte:='';
        end;
    end else
if FMain.TypeAff=CT_AFFCONFIRM then
begin
end else
if FMain.TypeAff=CT_AFFONEPARAMETER then
begin
end else
if FMain.TypeAff=CT_AFFMULTIVALUE then
begin
end else
if FMain.TypeAff=CT_AFFONEVALUE then
begin
end;
i:=ActualMenu.Count;
end;
i:=i+1;
end;
FMain.GroupBox2.Caption:=FMain.MenuName;
// Шукано the Repeat time
FMain.TimeRepeat:=0;
i:=0;
while (i<ActualMenu.Count) do
begin
    texte:=trim(ActualMenu.Strings[i]);
    if uppercase(copy(texte,1,7))='REPEAT=' then
begin
    delete(texte,1,7);
    texte:=trim(texte);
    try
        FMain.TimeRepeat:=StrToInt(texte);
    except
        FMain.TimeRepeat:=0;
    end;
    i:=ActualMenu.Count;
end;
i:=i+1;
Application.ProcessMessages;
if MainExit then Exit;
end;

// Використовуємо команди_ISO_14230_
FMain.ClickTree:=False;
FMain.T_StatusIO.Interval:=200;
FMain.ligneMenu:=0;
FMain.T_StatusIO.Enabled:=True;
end; // end if MenuI<>0
end;

//*****
{
procedure TFMain.rafraichir_diag;
var i,j : integer;
    texte : string;
begin
    if FMain.TypeAff=CT_AFFGRID then

```

```

begin
  if FMain.ClickTree then exit;
  FMain.nb_Row_visible:=FMain.StringGrid1.Height div
FMain.StringGrid1.DefaultRowHeight;
  if FMain.nb_Row_visible>FMain.StringGrid3.RowCount
  then FMain.nb_Row_visible:=FMain.StringGrid3.RowCount;
  if FMain.Grille_Up+FMain.nb_Row_visible>FMain.StringGrid3.RowCount
  then FMain.Grille_Up:=FMain.StringGrid3.RowCount-FMain.nb_Row_visible-
1;

  if FMain.StringGrid3.RowCount>FMain.nb_Row_visible
  then FMain.StringGrid1.RowCount:=FMain.nb_Row_visible
  else
  //FMain.StringGrid1.RowCount:=FMain.StringGrid3.RowCount;
  // Передача данных StringGrid3 до StringGrid1
  for i:=0 to FMain.StringGrid1.ColCount-1 do
    begin
      for j:=Grille_up+1 to Grille_up+FMain.StringGrid1.RowCount-1 do
        begin
          texte:=FMain.StringGrid3.Cells[i,j];
          texte:=copy(texte,1,length(texte)-1);
          if (texte<>'IMG_AMBER')
            and (texte<>'IMG_RED')
            and (texte<>'IMG_BLUE')
            and (texte<>'IMG_GREEN')
            then texte:=FMain.StringGrid3.Cells[i,j]
            else texte:='';
          FMain.StringGrid1.Cells[i,j-Grille_Up]:=texte;
          if
FMain.StringGrid1.ColWidths[i]<FMain.StringGrid1.Canvas.TextWidth(texte)+8
            then
FMain.StringGrid1.ColWidths[i]:=FMain.StringGrid1.Canvas.TextWidth(texte)+8;
          end;
        end;
      FMain.StringGrid1.Refresh;
    end;
  end;

end;

//*****
procedure TFMMain.GlobalReset;
var i : integer;
    //noeud : TTreeNode;
begin
  FMain.nom_fichier:='';
  BeginExit:=True;
  //FMain.ISO_14230.Tps_Maintient:=2000;
  FMain.Grille_Up:=0;
  FMain.nb_Row_visible:=FMain.StringGrid1.Height div
FMain.StringGrid1.DefaultRowHeight;
  //FMain.StringGrid3.RowCount:=0;
  FMain.ColorList.Clear;
  Fichier.Clear;
  Vehicule.Clear;
  MenuList.Clear;
  ActualMenu.Clear;
  Actual_ISO_14230.Clear;
  FMain.ExtendedMenu.Clear;
  FMain.Panel_Liste_Para.Visible:=False;
  FMain.Panel_Parametre.Visible:=False;
  FMain.Panel_grille.Visible:=False;
  FMain.Panel_Command.Visible:=False;
  FMain.Panel_Status.Visible:=True;
  FMain.Panel_Value.Visible:=False;
  // nettoyage des panels
  FMain.M_Hist.Clear;
  FMain.M_Parametre.Clear;
  FMain.M_Valeur.Clear;
  FMain.E_Data.Clear;
  FMain.E_Fichier_Para.Clear;

```

```

FMain.E_Para.Clear;
FMain.E_Para_Numero.Clear;
FMain.E_Sid.Clear;
FMain.E_Valeur.Clear;
//
State:=0;
FMain.Attente_Stop;
FMain.ISO_14230.StopDiagnosis;
FMain.Menu_Connection.Caption:=MenuProg[ CMP_Connecter ];
FMain.Image1.Picture.Bitmap:=nil;
FMain.IListConnect.GetBitmap(0,FMain.Image1.Picture.Bitmap);
for i:=0 to 9 do if FMain.Table[i]<>nil then FMain.Table[i].Clear;
FMain.Treeview1.Items.Clear;
//Noeud:=FMain.TreeView1.Items.Add(nil,MenuProg[ CMP_Status ]);
//Noeud.ImageIndex:=5;
//Noeud:=FMain.TreeView1.Items.Add(nil,MenuProg[ CMP_Command ]);
//Noeud.ImageIndex:=1;
FMain.TreeView1.Enabled:=False;
FMain.diag_actif:=False;
FMain.ClickActif:=False;
FMain.ClickGlobal:='';
FMain.ClickExit:='';
FMain.B_Mode_Maitre.Visible:=false;
FMain.NoReceive:=False;
Application.CleanupInstance;

try
if (FBiPWM<>nil) and (FBiPWM.Actif) then FBiPwm.Close;
if (FPWM<>nil) and (FPwm.Actif) then FPWM.Close;
if (FBridge<>nil) and (FBridge.Actif) then FBridge.Close;
if (FOnOff<>nil) and (FOnOff.Actif) then FOnOff.Close;
if (FValue<>nil) and (FValue.Actif) then FValue.Close;
if (FWiper<>nil) and (FWiper.Actif) then FWiper.Close;
//F_Configuration.GroupBox2.Enabled:=True;
except
end;
end;

//*****
// Цей пошук має спеціальний пункт процедури у StringList ListSearch і повертає
всі параметри включно між двома тегами.
//*****
procedure TMain.Search_Item(ItemSearch : string;ListSearch : TStringList;var
searchList: TStringList);
var texte : string;
    i,j : integer;
begin
    searchList.Clear;
    i:=0;
    while i<ListSearch.Count do
        begin
            texte:=ListSearch.Strings[i];
            texte:=trim(texte);
            if length(texte)>0 then
                if copy(texte,1,2)<>'//' then
                    if
uppercase(copy(texte,1,length(ItemSearch)+1))=uppercase('<'+ItemSearch) then
                        begin
                            //i:=i+1;
                            if i<ListSearch.Count then texte:=ListSearch.Strings[i] else
texte:='';
                                while (i<ListSearch.Count) and
(copy(texte,1,length(ItemSearch)+2)<>uppercase('</'+ItemSearch))
and (uppercase(copy(texte,1,7))<>'</MENU>') do
                                    begin
                                        texte:=ListSearch.Strings[i];
                                        texte:=trim(texte);
                                        j:=pos('; ',texte);
                                        if (copy(texte,1,2)<>'//') and (length(texte)>0) then

```

```

begin
    if j>0
    then SearchList.Add(copy(texte,1,j-1))
    else SearchList.Add(texte);
    end;
    i:=i+1;
end;
i:=ListSearch.Count;
end;
i:=i+1;
end;
end;

//*****
procedure TFMain.Ajust_StringList(Count: integer; var Liste: TStringList);
var i : integer;
begin
    if Count>Liste.Count then For i:=1 to Count-Liste.Count do Liste.Add('')
    else
    if Count<Liste.Count then For i:=1 to Liste.Count-Count do
Liste.Delete(Liste.Count-1);
end;

procedure TFMain.Attente_Start;
begin
    FMain.T_Attente.Enabled:=True;
end;

//*****
procedure TFMain.Attente_Stop;
begin
    FMain.T_Attente.Enabled:=False;
end;

//*****
Function TFMain.Cde2Send(CDE: string): integer;
var texte : string;
    erreur: integer;
begin
    while (FMain.Execution=false) and (MainExit=False) do
application.ProcessMessages;
    if MainExit then Exit;
    erreur:=0;
    FMain.ISO_14230.SendLength:=0;
    texte:=CDE;
    try
        FMain.ISO_14230.Sid:=hextodec(copy(texte,1,2));
    except
        FMain.ISO_14230.Sid:=0;
        erreur:=1;
        Result:=Erreur;
        exit;
    end;
    delete(texte,1,3);
    while length(texte)>0 do
        begin
FMain.ISO_14230.FData[FMain.ISO_14230.SendLength]:=hextodec(copy(texte,1,2));
        FMain.ISO_14230.SendLength:=FMain.ISO_14230.SendLength+1;
        delete(texte,1,3);
        end;
        Result:=erreur;
end;

//*****
procedure TFMain.Create_Menu_Langue;
var s : TSearchRec;
    texte : string;
begin

```

```

Nb_Langue:=0;
if FindFirst(Repertoire_courant+'Languages\*.dia',faAnyFile,s)=0 then
begin
  repeat
    M[Nb_Langue]:=TMenuItem.Create(Self);
    m[Nb_Langue].OnClick:=Menu_langueAddClick;
    m[Nb_Langue].Visible:=True;
    m[Nb_Langue].Enabled:=True;
    m[Nb_Langue].Tag:=Nb_Langue;
    texte:=s.Name;
    texte:=copy(texte,1,length(texte)-4);
    m[Nb_Langue].Caption:=texte;
    FMain.Menu_Langues.Add(m[Nb_Langue]);
    Nb_Langue:=Nb_Langue+1;
  until (FindNext(s)<>0) or (Nb_Langue>255);
end;
FindClose(s);
end;

//*****
procedure TFMMain.Init_Message;
begin
  FMain.Menu_Connection.Caption:=MenuProg[CMF_Connector];
  FMain.Menu_Setting.Caption:=MenuProg[CMF_Configuration];
  FMain.Menu_Quitter.Caption:=MenuProg[CMF_Quitter];
  FMain.Menu_Langues.Caption:=MenuProg[CMF_Language];
end;

//*****
procedure TFMMain.Menu_langueAddClick(Sender: TObject);
var texte : string;
    i:integer;
begin
  texte:=TMenuItem(Sender).Caption;
  ReadLanguage(Repertoire_courant+'Languages\'+texte+'.dia');

  FMain.ISO_14230.LoadMessageTXT(Repertoire_courant+'Languages\'+texte+'.ISO_14230
-');
  FMain.Init_Message;
  for i:=0 to Nb_Langue-1 do M[i].Checked:=False;
  i:=TMenuItem(Sender).Tag;
  M[i].Checked:=True;
  F_Configuration.CB_Langue.Text:=texte;
end;

end.

```

unit_configuration.pas - файл конфігурації

```

unit Unit_Configuration;

interface

uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls,
  ExtCtrls, Buttons, Grids, Common;

type
  { TF_Configuration }

TF_Configuration = class(TForm)
  B_Save: TBitBtn;
  B_Exit: TBitBtn;
  CB_Langue: TComboBox;
  GroupBox1: TGroupBox;
  GroupBox2: TGroupBox;
  Label2: TLabel;
  Panell: TPanel;
  Grid_Boitier: TStringGrid;
  procedure B_QuitterClick(Sender: TObject);
  procedure B_SaveClick(Sender: TObject);
  procedure CB_LangueChange(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure Grid_BoitierClick(Sender: TObject);
  procedure Grid_BoitierEditingDone(Sender: TObject);
  procedure Grid_BoitierSelectCell(Sender: TObject; aCol, aRow: Integer;
    var CanSelect: Boolean);
private
  { private declarations }
  oldValue : string;
public
  { public declarations }
  Langue : string;
  sortie : boolean;
  Procedure Init_Langue;

  function execute: boolean;
  Procedure Lecture_ini;
  Procedure Sauve_Ini;
end;

var
  F_Configuration: TF_Configuration;

implementation

{$R *.dfm}
uses uMain;

{ TF_Configuration }
procedure TF_Configuration.B_QuitterClick(Sender: TObject);
begin
  F_Configuration.Close;
end;

procedure TF_Configuration.B_SaveClick(Sender: TObject);
begin
  sortie:=True;
end;

procedure TF_Configuration.CB_LangueChange(Sender: TObject);
var texte : string;

```

```

begin
  texte:=F_Configuration.CB_Langue.Text;
  if FileExists(Repertoire_courant+'Languages\'+'+texte+'.dia') then
    begin
      ReadLanguage(Repertoire_courant+'Languages\'+'+texte+'.dia');
      F_Configuration.Init_Langue;
    end;
  end;

procedure TF_Configuration.FormCreate(Sender: TObject);
var s : TSearchRec;
    texte : string;
begin
  F_Configuration.CB_Langue.Clear;
  if FindFirst(Repertoire_courant+'Languages\*.dia',faAnyFile,s)=0 then
    begin
      repeat
        texte:=s.Name;
        texte:=copy(texte,1,length(texte)-4);
        F_Configuration.CB_Langue.Items.Add(texte);
      until (FindNext(s)<>0) or (Nb_Langue>255);
    end;
  FindClose(s);
  F_Configuration.CB_Langue.Text:='ukrainian';
  //if FileExists(Repertoire_Courant+'Diag_ISO_14230.ini') then Lecture_Ini else
  Sauve_Ini;
end;

procedure TF_Configuration.Grid_BoitierClick(Sender: TObject);
begin
  OldValue:=Grid_Boitier.Cells[Grid_Boitier.Col,Grid_Boitier.Row];
end;

procedure TF_Configuration.Grid_BoitierEditingDone(Sender: TObject);
var ACol,ARow : integer;
    i : integer;
    texte : string;
begin
  aCol:=Grid_Boitier.Col;
  aRow:=Grid_Boitier.Row;
  try
    case aRow of
      1 : begin // com-порт
          texte:=Grid_Boitier.Cells[aCol,aRow];
          delete(texte,1,3); // видаляємо the "COM"
          texte:=trim(texte);
          i:=strtoint(texte);
          FMain.ISO_14230.Comport.Port:=Grid_Boitier.Cells[aCol,aRow];
        end;
      2 : begin // Версія
          F_Configuration.Grid_Boitier.Cells[aCol,aRow]:=OldValue;
        end;
      3 : begin // Швидкість діагностики
          texte:=Grid_Boitier.Cells[aCol,aRow];
          texte:=trim(texte);
          i:=strtoint(texte);
          FMain.ISO_14230.DiagnosticSpeed:=i;
        end;
      4 : begin // Інформаційний режим
          F_Configuration.Grid_Boitier.Cells[aCol,aRow]:=OldValue;
        end;
      5 : begin // Номер спроби
          texte:=Grid_Boitier.Cells[aCol,aRow];
          texte:=trim(texte);
          i:=strtoint(texte);
          FMain.ISO_14230.Tentative:=i;
        end;
      6 : begin // Довжина байту адреси
          texte:=Grid_Boitier.Cells[aCol,aRow];

```

```

        texte:=trim(texte);
        i:=strtoint(texte);
        FMain.ISO_14230.Time_Ad:=i;
    end;
7 : begin // Час між байтами adress та Sync
    texte:=Grid_Boitier.Cells[aCol,aRow];
    texte:=trim(texte);
    i:=strtoint(texte);
    FMain.ISO_14230.Time_Ad_Sync:=i;
end;
8 : begin // Час між байтами Sync та Key1
    texte:=Grid_Boitier.Cells[aCol,aRow];
    texte:=trim(texte);
    i:=strtoint(texte);
    FMain.ISO_14230.Time_Sync_Key1:=i;
end;
9 : begin // Час між байтами Key1 та Key2
    texte:=Grid_Boitier.Cells[aCol,aRow];
    texte:=trim(texte);
    i:=strtoint(texte);
    FMain.ISO_14230.Time_Key1_Key2:=i;
end;
10 : begin // Час між байтами Key2b та Adb
    texte:=Grid_Boitier.Cells[aCol,aRow];
    texte:=trim(texte);
    i:=strtoint(texte);
    FMain.ISO_14230.Time_Key2b_Adb:=i;
end;
11 : begin // Час закінчення
    texte:=Grid_Boitier.Cells[aCol,aRow];
    texte:=trim(texte);
    i:=strtoint(texte);
    FMain.ISO_14230.Time_TO:=i;
end;
12 : begin // Time_tester представлення
    texte:=Grid_Boitier.Cells[aCol,aRow];
    texte:=trim(texte);
    i:=strtoint(texte);
    FMain.ISO_14230.Tps_Maintient:=i;
end;
13 : begin // Чекаємо 1 REQ
    texte:=Grid_Boitier.Cells[aCol,aRow];
    texte:=trim(texte);
    i:=strtoint(texte);
    FMain.ISO_14230.Wait_1Req:=i;
end;
14 : begin // Чекаємо відповідь на запит
    texte:=Grid_Boitier.Cells[aCol,aRow];
    texte:=trim(texte);
    i:=strtoint(texte);
    FMain.ISO_14230.Wait_Response_Request:=i;
end;
end;
except
    F_Configuration.Grid_Boitier.Cells[aCol,aRow]:=OldValue;
end;
end;

procedure TF_Configuration.Grid_BoitierSelectCell(Sender: TObject; aCol,
    aRow: Integer; var CanSelect: Boolean);
begin
    OldValue:=Grid_Boitier.Cells[aCol,aRow];
end;

procedure TF_Configuration.Init_Langue;
var i : integer;
begin
    F_Configuration.Caption:=MenuProg[CMF_Configuration];
    F_Configuration.GroupBox1.Caption:=MenuProg[CMF_DefaultLang];

```

```

F_Configuration.Label2.Caption:=MenuProg[CMF_DefaultLang];
F_Configuration.B_Save.Caption:=MenuProg[CMF_Save];
F_Configuration.B_Exit.Caption:=MenuProg[CMF_Quitter];
F_Configuration.GroupBox2.Caption:=MenuProg[CMF_SetCom];
F_Configuration.Grid_Boitier.RowCount:=15;
F_Configuration.Grid_Boitier.Cells[0,0]:=MenuProg[CMF_Parametre];
F_Configuration.Grid_Boitier.Cells[1,0]:=MenuProg[CMF_Value];
F_Configuration.Grid_Boitier.Cells[0,1]:='COM';
F_Configuration.Grid_Boitier.Cells[0,2]:=menuProg[CMF_Versus];
F_Configuration.Grid_Boitier.Cells[0,3]:=MenuProg[CMF_DiagSpeed];
F_Configuration.Grid_Boitier.Cells[0,4]:=MenuProg[CMF_ModeInformation];
F_Configuration.Grid_Boitier.Cells[0,5]:=MenuProg[CMF_Tentative];
F_Configuration.Grid_Boitier.Cells[0,6]:=MenuProg[CMF_Time_Ad];
F_Configuration.Grid_Boitier.Cells[0,7]:=MenuProg[CMF_Time_Ad_Sync];
F_Configuration.Grid_Boitier.Cells[0,8]:=MenuProg[CMF_Time_Sync_Key1];
F_Configuration.Grid_Boitier.Cells[0,9]:=MenuProg[CMF_Time_Key1_Key2];
F_Configuration.Grid_Boitier.Cells[0,10]:=MenuProg[CMF_Time_Key1_Key2b_Adb];
F_Configuration.Grid_Boitier.Cells[0,11]:=MenuProg[CMF_Time_TO];
F_Configuration.Grid_Boitier.Cells[0,12]:=MenuProg[CMF_testerPresent];
F_Configuration.Grid_Boitier.Cells[0,13]:=MenuProg[CMF_Wait_1Req];
F_Configuration.Grid_Boitier.Cells[0,14]:=MenuProg[CMF_Wait_Res_Req];
for I := 0 to 14 do
  if
F_Configuration.Grid_Boitier.ColWidths[0]<F_Configuration.Grid_Boitier.Canvas.Te
xtWidth(F_Configuration.Grid_Boitier.Cells[0,i])+10
  then
F_Configuration.Grid_Boitier.ColWidths[0]:=F_Configuration.Grid_Boitier.Canvas.T
extWidth(F_Configuration.Grid_Boitier.Cells[0,i])+10;
end;

Function TF_Configuration.execute : boolean;
var i : integer;

begin
  sortie:=False;
  langue:=F_Configuration.CB_Langue.Text;
  if FileExists(Repertoire_Courant+'Diag_ISO_14230.ini') then Lecture_Ini else
Sauve_Ini;
  F_Configuration.Init_Langue;
  F_Configuration.Grid_Boitier.Cells[1,1]:=FMain.ISO_14230.Comport.Port;
  F_Configuration.Grid_Boitier.Cells[1,2]:=FMain.ISO_14230.Version;

F_Configuration.Grid_Boitier.Cells[1,3]:=inttostr(FMain.ISO_14230.DiagnosticSpee
d);

F_Configuration.Grid_Boitier.Cells[1,4]:=FMain.ISO_14230.ModeInfoToStr(FMain.ISO
_14230.Mode_Information);
  F_Configuration.Grid_Boitier.Cells[1,5]:=inttostr(FMain.ISO_14230.Tentative);
  F_Configuration.Grid_Boitier.Cells[1,6]:=inttostr(FMain.ISO_14230.Time_Ad);

F_Configuration.Grid_Boitier.Cells[1,7]:=inttostr(FMain.ISO_14230.Time_Ad_Sync);
F_Configuration.Grid_Boitier.Cells[1,8]:=inttostr(FMain.ISO_14230.Time_Sync_Key1
);
F_Configuration.Grid_Boitier.Cells[1,9]:=inttostr(FMain.ISO_14230.Time_Key1_Key2
);
F_Configuration.Grid_Boitier.Cells[1,10]:=inttostr(FMain.ISO_14230.Time_Key2b_Ad
b);
  F_Configuration.Grid_Boitier.Cells[1,11]:=inttostr(FMain.ISO_14230.Time_TO);

F_Configuration.Grid_Boitier.Cells[1,12]:=inttostr(FMain.ISO_14230.Tps_Maintient
);
  F_Configuration.Grid_Boitier.Cells[1,13]:=inttostr(FMain.ISO_14230.Wait_1Req);

F_Configuration.Grid_Boitier.Cells[1,14]:=inttostr(FMain.ISO_14230.Wait_Response
_Request);
  F_Configuration.ShowModal;

```

```

if sortie then
begin

FMain.ISO_14230.LoadMessageTXT(Repertoire_courant+'Languages\'+F_Configuration.C
B_Langue.Text+'.ISO_14230_');
  FMain.Init_Message;
  FMain.ISO_14230.SaveConfiguration(Repertoire_courant+'_ISO_14230.ini');
  F_Configuration.Sauve_Ini;
end else
begin
  F_Configuration.CB_Langue.Text:=langue;
  ReadLanguage(Repertoire_courant+'Languages\'+langue+'.dia');
  FMain.ISO_14230.LoadConfiguration(Repertoire_courant+'_ISO_14230.ini');
end;
Result:=Sortie;
end;

procedure TF_Configuration.Lecture_ini;
var Fichier : TStringList;
    i : integer;
    texte : string;
begin
try
  Fichier:=TStringList.Create;
  Fichier.LoadFromFile(Repertoire_Courant+'Diag_ISO_14230.ini');
  for i:=0 to Fichier.Count-1 do
    begin
      texte:=fichier.Strings[i];
      if (length(texte)>0) and (copy(texte,1,2)<>'//')
      then begin
        if copy(texte,1,5)='LANG=' then
          begin
            delete(texte,1,5);
            texte:=trim(texte);
            F_Configuration.CB_Langue.Text:=texte;
          end else
            if copy(texte,1,6)='DEBUG=' then
              begin
                delete(texte,1,6);
                texte:=trim(texte);
                try
                  FMain.Debug:=StrToInt(texte);
                except
                  FMain.Debug:=0;
                end;
              end;
            end;
          end;
        end;
      end;
    finally
      fichier.Free;
    end;
  end;

procedure TF_Configuration.Sauve_Ini;
var Fichier : TStringList;
begin
try
  Fichier:=TStringList.Create;
  Fichier.Add('// Diag_ISO_14230_ конфігураційний файл');
  Fichier.Add('');
  Fichier.Add('LANG='+F_Configuration.CB_Langue.Text);
  Fichier.Add('DEBUG='+inttostr(FMain.Debug));
  Fichier.SaveToFile(Repertoire_Courant+'Diag_ISO_14230.ini');
finally
  fichier.Free;
end;
end;
end.

```

unit_connection.pas - файл підключення

```

unit Unit_Connection;

interface

uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, StdCtrls,
  Grids, ExtCtrls, Buttons, CPort, Common;

type
  { TF_Connection }

  TF_Connection = class(TForm)
    B_Quitter: TBitBtn;
    B_Connector: TBitBtn;
    Combo_Port: TComboBox;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    L_PortCOM: TLabel;
    Panell: TPanel;
    StringGrid1: TStringGrid;
    procedure B_ConnectorClick(Sender: TObject);
    procedure B_QuitterClick(Sender: TObject);
  private
    { private declarations }
  public
    { public declarations }
    Valid : boolean;
    Vehicule_Name : string;
    Procedure Init;
    Function execute : boolean;
  end;

var
  F_Connection: TF_Connection;

implementation

{$R *.dfm}

{ TF_Connection }

procedure TF_Connection.B_ConnectorClick(Sender: TObject);
begin
  If F_Connection.StringGrid1.Row<1 then
    begin
      Application.MessageBox(PCHAR(MenuProg[ CMP_NoECUSelect ]), PCHAR(MenuProg[ CMP_Error
      ]), 0);
      exit;
    end else
      Vehicule_Name:=F_Connection.StringGrid1.Cells[0, F_Connection.StringGrid1.Row];
      F_Connection.Valid:=True;
      F_Connection.Close;
    end;

procedure TF_Connection.B_QuitterClick(Sender: TObject);
begin
  F_Connection.Valid:=False;
  F_Connection.Close;
end;

procedure TF_Connection.Init;
var sFileData : TSearchREC;

```

```

begin
  F_Connection.StringGrid1.RowCount:=1;
  if ( FindFirst( Repertoire_courant + 'Vehicule\'+'*.veh', faAnyFile,
sFileData) = 0 ) then
    begin
      repeat
        if ( ( sFileData.Name <> '.' ) and ( sFileData.Name <> '..' ) ) then
          begin

F_Connection.StringGrid1.RowCount:=F_Connection.StringGrid1.RowCount+1;
          F_Connection.StringGrid1.Cells[0,F_Connection.StringGrid1.RowCount-
1]:=sFileData.Name;
          end;
          until ( FindNext( sFileData ) <> 0 );
        end;
      FindClose( sFileData );
      if F_Connection.StringGrid1.RowCount>1 then
        begin
          F_Connection.StringGrid1.FixedRows:=1;
          end;
        EnumComports (F_Connection.Combo_Port.Items);

F_Connection.Combo_Port.ItemIndex:=F_Connection.Combo_Port.Items.IndexOf(Port_Co
m);
      if (F_Connection.Combo_Port.ItemIndex<0)
        and (F_Connection.Combo_Port.Items.Count>0)
        then F_Connection.Combo_Port.ItemIndex:=0
          else F_Connection.Combo_Port.Text:=MenuProg[CMP_NoComPort];
      F_Connection.Caption:=MenuProg[CMP_Connector];
      F_Connection.GroupBox1.Caption:=MenuProg[CMP_ECU];
      F_Connection.StringGrid1.Cells[0,0]:=MenuProg[CMP_ECU];
      F_Connection.B_Connector.Caption:=MenuProg[CMP_Connector];
      F_Connection.B_Quitter.Caption:=MenuProg[CMP_Quitter];
      F_Connection.GroupBox2.Caption:=MenuProg[CMP_Configuration];
      F_Connection.L_PortCOM.Caption:=MenuProg[CMP_Comport];
    end;

function TF_Connection.execute: boolean;
begin
  F_Connection.Init;
  F_Connection.Valid:=False;
  F_Connection.ShowModal;
  Result:=F_Connection.Valid;
end;

end.

```

about.pas - файл довідки

```
unit about;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, jpeg, ExtCtrls;

type
  TFmAbout = class(TForm)
    Memo1: TMemo;
    Button1: TButton;
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FmAbout: TFmAbout;

implementation

{$R *.dfm}

procedure TFmAbout.FormCreate(Sender: TObject);
begin
  Memo1.Clear;
  Memo1.Lines.Add('БАКАЛАВРСЬКА РОБОТА');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('на тему:');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Програмне забезпечення системи діагностування електронної
системи автомобіля за стандартом ISO 14230');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Керівник: Дреев О.М. ');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('Розробив: студент Марченко Максим Сергійович');
  Memo1.Lines.Add(' гр. КМ-19');
  Memo1.Lines.Add('');
  Memo1.Lines.Add('М. Кропивницький 2023');
  Memo1.Lines.Add('');
end;

procedure TFmAbout.Button1Click(Sender: TObject);
begin
  FmAbout.Close;
end;
end.
```