

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

“ ___ ” _____ 2021 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему

**“Дослідження та програмна реалізація автоматизованої
інформаційно-діагностичної системи поїзду”**

Виконав здобувач вищої освіти
II курсу, групи КН-20М-1,4
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Азатьян З.Р.
« ___ » _____ 2021 р.

Керівник проекту
кандидат технічних наук
_____ Костянтин БУРАВЧЕНКО
« ___ » _____ 2021 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 122 “Комп’ютерні науки”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерні науки”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.

Олексій СМІРНОВ
« 6 » вересня 2021 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Азатьяну Захару Романовичу

(прізвище, ім'я, по батькові)

- Тема роботи Дослідження та програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду
- Керівник роботи Буравченко Костянтин Олегович, канд. техн. наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 39-13 від 02.08.2021 року
- Строк подання студентом роботи до захисту 10.12.2021 р.
- Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Призначення та область використання.
 - Економічна ефективність розробленої програми.
 - Перегляд аналогічних існуючих систем.
 - Заходи з охорони праці та техніки безпеки
 - Опис і обґрунтування проектних рішень.
 - Висновки.
 - Етапи програмування системи.
 - Впровадження системи в промислову експлуатацію
 - Наукова новизна
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна	1 аркуш
Структурна схема системи	1 аркуш
Функціональна схема системи	1 аркуш
Діаграма процесів	1 аркуш
Блок-схема алгоритму роботи додатку	2 аркуша
Показники економічної ефективності	1 аркуш

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2021	14.11.2021
Охорона праці	Оришака О.В.	06.10.2021	16.11.2021

7. Дата видачі завдання « 6 » вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2021 р.	
3.	Розробка моделі компонента	20.10.2021 р.	
4.	Розробка структур даних	25.10.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2021 р.	
6.	Програмування алгоритмів	10.11.2021 р.	
7.	Розрахунок економічної ефективності	13.11.2021 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2021 р.	
9.	Оформлення ПЗ	17.11.2021 р.	
10.	Попередній захист роботи	10.12.2021 р.	

Дата видачі завдання
« 6 » вересня 2021 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2021 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Азатьян З.Р. Дослідження та програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для автоматизованої інформаційно-діагностичної системи поїзду.

Метою розробки є дослідження та програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду.

Об'єктом дослідження є процес реалізації автоматизованої інформаційно-діагностичної системи поїзду.

Предметом дослідження є методи реалізації автоматизованої інформаційно-діагностичної системи поїзду.

Методи дослідження базуються на методах теорії автоматизованого управління, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду.

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 XP/Vista/7/8/10.

Програму розроблено в середовищі RAD Studio Delphi 10.4.

Ключові слова: Комп'ютерні науки, автоматизація, інформаційно-діагностична система

ABSTRACT

Azatian Z.R. Research and software implementation of the automated information and diagnostic system of the train. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this final qualification work on the second (master's) level of higher education the software which is intended for the automated information-diagnostic system of the train is developed.

The purpose of development is research and software implementation of the automated information and diagnostic system of the train.

The object of research is the process of realization of the automated information-diagnostic system of the train.

The subject of research is the methods of realization of the automated information-diagnostic system of the train.

Research methods are based on the methods of the theory of automated control, methods of mathematical statistics, methods of software development.

The result is the software implementation of the automated information and diagnostic system of the train.

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.

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on an IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the environment of RAD Studio Delphi 10.4.

Keywords: Computer science, automation, information and diagnostic system

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	8
1.1 Призначення системи.....	8
1.2 Область застосування.....	12
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	14
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	14
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	24
2.3 Розгорнута постановка завдання	30
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	32
3.1 Опис функціонування системи.....	32
3.2 Розробка структурної схеми	50
3.3 Розробка функціональної схеми.....	53
3.4 Розробка діаграми процесів	55
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ	56
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	56
4.2 Захист розробленого програмного забезпечення	70
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	73
6 НАУКОВА НОВИЗНА	76

ВКРМ-122.21.0001.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Азат'ян З.Р.			Дослідження та програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду	Лім.	Аркуш	Аркушіів
Перев.		Буравченко К.О.				М	1	115
Н.контр.		Гермак В.С.			ЦНТУ КН-20М-1,4			
Затв.		Смірнов О.А.						

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	77
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.	77
7.2 Розрахунок трудомісткості розробки програмної продукції	79
7.3 Визначення чисельності виконавців і планового фонду зарплати	81
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника	86
7.5 Визначення собівартості розробки та ціни програмної продукції.	90
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	93
7.7 Визначення експлуатаційних витрат.....	93
7.8 Визначення економічної ефективності програмної продукції.....	95
7.9 Висновок.	97
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	98
8.1 Вступ	98
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером	99
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста .	101
8.4 Розробка заходів з умов поліпшення охорони праці.....	104
8.5 Розрахункова частина	104
8.6 Висновки до розділу.....	106
9 ОСНОВНІ ВИСНОВКИ.....	107
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	109

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

EOM	– електронно-обчислювальна машина
ЕЧТ	– екологічно чистий туалет
IT	– інформаційні технології
ПЕОМ	– персональна електронно-обчислювальна машина
API	– прикладний програмний інтерфейс
GSM/3G/4G/5G	– стандарти мобільного зв'язку
ISO/IEC	– стандарт
ActiveX	– технологія зв'язування й вбудовування об'єктів

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Залізничний транспорт залишається найважливішою складовою частиною транспортної системи України, на його частку доводиться понад 40% пасажирообороту, виконуваного всіма видами суспільного транспорту. Відповідно до прогнозів розвитку народного господарства очікується ріст обсягу пасажирських перевезень. Тим часом, існуючий парк пасажирського рухливого состава багато в чому застарів.

З 1992 р. поставка пасажирських вагонів локомотивної тяги зменшилася до 300 вагонів у рік. У результаті парк вагонів скоротився майже в півтора разу, а його старіння йде значно швидше, ніж відновлення. Сьогодні зношування становить у середньому 50%. Значне число пасажирських вагонів уже виробили свій нормативний термін служби. У сучасний час із загального парку вагонів порядку 1500 вимагають списання по терміну служби [1]. Більше 7 тис. вагонів мають вік понад 20 років, вони застаріли фізично й морально, не відповідають сучасним вимогам перевезення пасажирів. Близько 80% вагонів не обладнані системами кондиціонування повітря, що значно знижує комфортність поїздок. Така ситуація диктує необхідність якнайшвидшого відновлення парку [1-5].

Підвищення ефективності роботи пасажирського комплексу залізниць України ставиться до числа найбільш відповідальних і актуальних завдань, що стоять перед галуззю. Важливу роль у їхньому рішенні покликана зіграти концепція Програми оновлення парку вагонів, яка розрахована до 2028 року.

Основні напрямки Комплексної програми, які повинні забезпечити її ефективність, можна сформулювати в такий спосіб [6]:

– збільшення коефіцієнта використання потужностей вагонних депо й зменшення їхньої кількості за рахунок концентрації потужностей у більш оснащених, як наслідок, зменшення експлуатаційних витрат;

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Робота апробована на LV Науково-технічна конференція здобувачів вищої освіти «Наука – виробництву», 2021, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №12.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

У загальному обсязі роботи залізничного транспорту пасажирські перевезення займають відносно невелику частку – менш 15%. Однак із упевненістю можна стверджувати, що по роботі саме пасажирського комплексу, з огляду на його велике соціальне значення, головним чином формується суспільна думка про залізницю в цілому.

Пасажирські перевезення традиційно збиткові. А останнім часом підсилилася конкуренція з повітряним транспортом і автомобільним загального користування. Статистичні дані свідчать, що з 1996 р. залізничний транспорт перестав бути пріоритетним видом транспорту в пасажирських перевезеннях (рисунок 1.1).

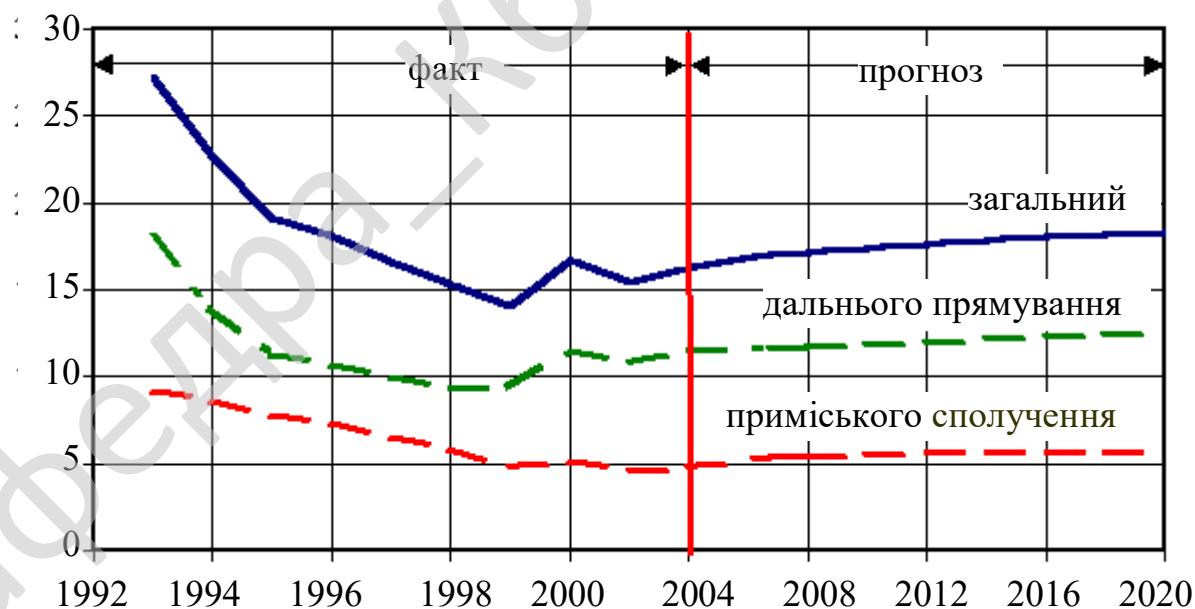


Рисунок 1.1 – Пасажирообіг залізниць

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

8

Сьогодні залізниці відноситься 40,5% виконаного всіма видами суспільного транспорту пасажирообігу (рисунок 1.2).

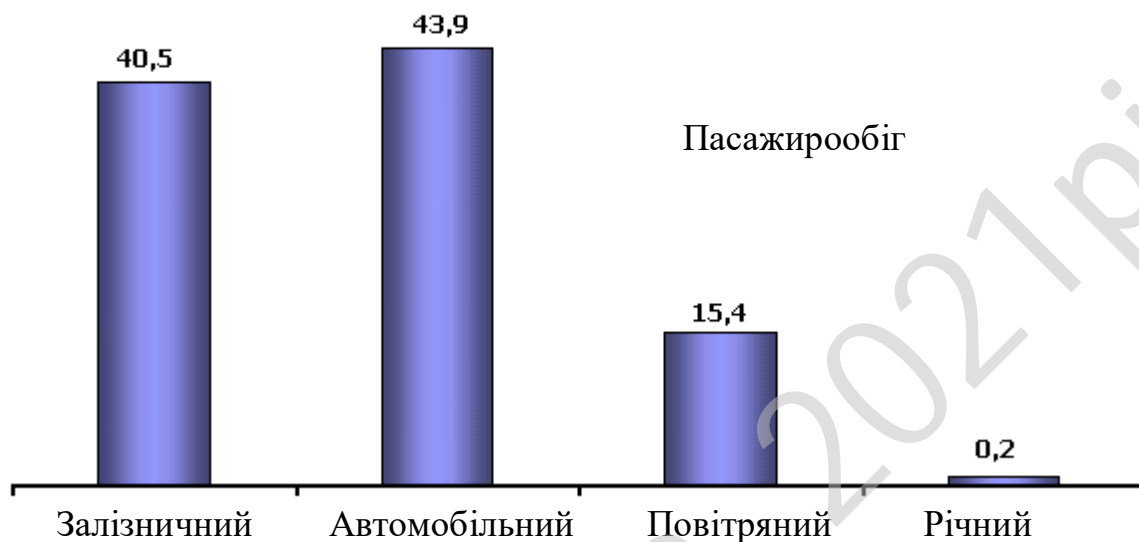


Рисунок 1.2 – Залізниця в транспортній системі загального користування України

Відповідно до прогнозів розвитку народного господарства очікується й останні два роки вже спостерігається ріст обсягу пасажирських перевезень. До 2028 р. прогнозується збільшення пасажирообігу не менш ніж на 13%.

Конкурентоспроможність залізничного транспорту в пасажирських перевезеннях визначається багатьма факторами, такими як швидкість, надійність, ритмічність, безпека, зручність, комфорт, тариф і поруч інших. При цьому технічний і якісний рівень пасажирського рухливого складу відіграє ключову роль.

Зовсім очевидно, що існуючий парк пасажирського рухливого складу застарів. Вирішити проблему його відновлення можна тільки істотним збільшенням обсягів поставок вагонів нової якості.

В останні роки промисловістю виготовлені й пред'явлені на випробування ряд нових зразків пасажирського рухливого складу: швидкісні пасажирські

вагони моделей 61-4170 і 9510, купейні СВ моделі 61-4174, електропоїзда ЕД4, ЕД4МК, ЕД4Е, ЕД9М, ЕД6, ЕТ2А, ЕНЗ, ЕМ2И, ЕС250 (Сокіл), рейковий автобус РА1, дизель-поїзд ДЛ2. І хоча деякі з них ще не доведені до експлуатаційної готовності, проведена робота продемонструвала можливості вітчизняної промисловості й дозволила по-новому оцінити перспективи використання пасажирського рухливого складу в мережі залізниць, застосування в конструкції нових систем і устаткування. На основі цього досвіду й з аналізу закордонних даних саме в останні роки з'явилися нові аргументи "за" або "проти" впровадження на наших залізницях тих або інших технічних рішень.

Закордонний досвід показує, що техніко-економічні й споживчі властивості пасажирського рухливого складу визначаються умовами їхньої експлуатації. Як правило, залізничні компанії купують невеликі по чисельності серії вагонів. Однак ці вагони сконструйовані й обладнані з обліком всіх специфічних вимог замовника. У цих умовах для істотного зменшення капітальних витрат була запропонована й одержала широкий розвиток концепція модульних платформ, коли рухомий склад фактично комбінується з типоразмерного ряду уніфікованих модулів різного призначення. Сьогодні це – типове рішення, пропонуване практично всіма провідними світовими виробниками рухливого складу для перевезення пасажирів.

У нашій країні переважала тенденція створення "універсального" рухливого складу. Ця ідея була привабливою в радянські часи завдяки порівняно низьким витратам на організацію крупносерійного виробництва вагонів. При цьому слід зазначити, що через неефективне або нецільове використання вагонів експлуатаційні витрати значні.

Останнім часом з'явилися нові сегменти ринку пасажирських перевезень, де застосування залізничного транспорту могло б бути ефективним і конкурентоспроможним. Серед них – використання електропоїздів у внутріміському повідомленні, особливо у великих містах, всі частіше зіштовхуються із проблемами транспортного обслуговування населення. Одним з

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

основних критеріїв застосування того або іншого типу рухливого складу на заданому полігоні повинна стати його здатність успішно конкурувати з іншими видами транспорту (автомобільним або авіаційним), тобто мати необхідні параметри й техніко-економічні показники, що забезпечують помітне зниження витрат при експлуатації й достатній привабливості для пасажирів.

Створення економічного й ефективного рухливого складу для регіонального й місцевого повідомлення могло б скласти серйозну конкуренцію автомобільному транспорту загального користування, обсяг перевезень якого в зоні роботи залізниць значний (рисунк 1.3).



Рисунок 1.3 – Розподіл пасажирообігу автомобільного транспорту загального користування в зоні роботи МПЗ

Актуальне створення великих інтермодальних транспортних систем, де різні види транспорту могли б не конкурувати один з одним, а доповнювати й працювати на загальний результат. Основними заходами є узгодження тарифів, розкладів руху різних видів транспорту й розвиток інфраструктури пересадних вузлів для забезпечення швидкої й зручної зміни виду транспорту. При цьому може використовуватися як спеціалізований, так і універсальний залізничний

рухомий склад. Переваги для пасажирів очевидні й можуть бути проілюстровані безліччю прикладів із закордонного досвіду – це зменшення часу й зниження вартості поїздки при підвищенні рівня комфорту, у більшості випадків пасажирообіг залізниць збільшується.

1.2 Область застосування

Для спеціалізованого рухливого складу, розробленого й обладнаного з урахуванням всіх особливостей його експлуатації, характерні низькі експлуатаційні витрати. Уніфікація устаткування зменшує його вартість. У цих умовах розумний компроміс між необхідною спеціалізацією й широкою уніфікацією вагонів стає ключовим фактором зниження вартості життєвого циклу рухливого складу й підвищення ефективності пасажирських перевезень.

Типаж рухливого складу є основним документом, що встановлює вибір основних типів рухливого складу виходячи з потреб залізниць, що визначає їх основні нормативні показники, габарити й вимоги до уніфікації устаткування, безпеки руху, життєзабезпеченню й сумісності із системами залізничного транспорту й визначального пріоритетів у поставках.

Основні завдання, які вирішувалися при розробці нового типу моторвагоного рухливого складу й пасажирських вагонів локомотивної тяги, полягають у визначенні ключових підходів, що забезпечують:

- виконання всіх вимог пасажирів, включаючи безпеку, швидкість і комфорт;
- підвищення експлуатаційної надійності рухливого складу, зниження витрат на технічне обслуговування й ремонт;
- зменшення питомої витрати палива й електроенергії;
- зменшення вартості розробки й виготовлення рухливого складу.

При розробці типу враховувалося, що в даний момент зложилися всі передумови до істотного підвищення якості пасажирського рухливого складу, і що поставлені завдання можуть бути вирішені тільки при використанні сучасних досягнень науки й техніки:

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

– нових технологій виготовлення кузовів полегшеної конструкції зі збільшенням терміну служби до 40 років, обладнаних уніфікованими герметизованими міжвагонними переходами, зчіпними пристроями твердого типу, у тому числі з автоматичним з'єднанням пневматичних магістралей і електричних кіл;

– нових конструкцій екіпажної частини, що дозволяють підвищити швидкість руху, зменшити рівень шуму, динамічний вплив на шлях і поліпшити плавність ходу;

– комплексних бортових мікропроцесорних систем управління, забезпечення безпеки руху, автоведення й діагностики, уніфікованих з аналогічними системами локомотивів;

– нового ефективного гальмового обладнання, що забезпечує максимальну інтеграцію з мікропроцесорною системою управління й широке застосування протиюзного захисту;

– економічних силових установок нового покоління для дизель-поїздів з поліпшеними екологічними характеристиками;

– сучасного тягового електропривода із системами електричного рекуперативного гальмування практично до повної зупинки, у тому числі із застосуванням на дизель-поїздах накопичувачів енергії;

– систем живлення ланцюгів власних потреб із застосуванням високовольтних статичних перетворювачів;

– ефективних систем життєзабезпечення із широким використанням кондиціонування повітря, екологічно чистих туалетів, інформаційних систем, автоматичного контролю посадки-висадження, пристроїв, що забезпечують комфортний і безперешкодний проїзд інвалідів.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Вагон призначений для виконання перевезень пасажирів і персоналу поїзда на електрифікованих ділянках магістральних шляхів колії 1520 мм, з високою й низькою платформою, для швидкостей руху до 160 км/ч. Конструкція вагона повною мірою відповідає вимогам санітарних норм по освітленості, ергономіці, мікроклімату, шуму й вібраціям, вимогам безпеки. Вагон обладнаний апаратурою радіоелектронної інформаційної системи пасажирського рухливого состава, включаючи переговорний пристрій для трансляції відео програм. Система опалення разом із системою вентиляції забезпечує температуру повітря в пасажирських приміщеннях вагона в зимовий період року 20-24°C, літній період року 22-26°C, у туалетах – не нижче 16°C. Вагон має у своєму составі сім пасажирських чотирьохмісних купе з нижніми диванами й радіо купе.

Для забезпечення контролю й управління вагоним устаткуванням поїзда вагон оснащений системою контролю й діагностики управління: ПЕОМ із вбудованою або зовнішніми мережною картою й GSM/3G/4G/5G – модемом.

Зовнішнє фарбування вагона виконується із застосуванням сучасних технологій і матеріалів, що відповідають міжнародним стандартам. Колір зовнішнього фарбування визначається замовником. У вагоні на поперечних перегородках великого коридору встановлені два інформаційних табло, що показують температуру повітря зовні й усередині вагона, зайнятість туалетів, що тече час, додаткову інформацію оперативного, довідкового або комерційного характеру у вигляді рядка, що біжить.

Радіокупе обладнано:

- столом-тумбою для установки радіоапаратури;
- полицею для апаратури;

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

- радіостанцією;
- телевізором;
- апаратурою радіоелектронної інформаційної системи;
- гачками для одягу;
- багажною нішею;
- сходами, що складаються, для доступу до багажної ніші;
- нестационарно встановленим поворотним кріслом;
- багажною полицею;
- нижнім диваном.

Службове відділення обладнане:

- засобами зв'язку, оповіщення, управління;
- кріслом з підлокітниками;
- столиком;
- столом-мийкою для миття посуду;
- шафами для розміщення аптечки, посуду, одягу, сейфа, експлуатаційної документації;
- холодильником.

Таблиця 2.1 – Технічні характеристики:

Маса тари вагона (без екіпірування), т	56,1
Довжина вагона, мм	25500
База вагона, мм	17000
Конструкційна швидкість, км/год.	160
Модель візка	68-4095 і 68-4096
База візка, мм	2500
Ширина кузова зовнішня, мм	3103
Висота осі автотягача від рівня головки рейок, мм	1060
Ширина колії, мм	1520
Кількість місць для сидіння	14
Кількість місць для сидіння провідника	1

Продовження таблиці 2.1

Гальмо	Дисковий пневматичний і електропневматичний, ручний
Електроустаткування	Централізоване від поїзної магістралі напругою 3000 В від статичного перетворювача, від акумуляторної батареї напругою 110 В и від зовнішньої трифазної мережі напругою 380 В 50 Гц із глухозаземленою нейтралю (на стоянці), з пожежогасінням пульта.
Система опалення	рідинна з незамерзаючим теплоносієм
Питома витрата електроенергії, кВт.ч/1000 пасс.км	16,7
Енергоємність поглинаючого апарата кгс.м	4000
Плавність ходу не більше	2,9
Середній коефіцієнт теплопередачі огорожень кузова, не більше Вт/м.К	0,85
Максимальна тривалість проходження до першого екіпірування, годин	24
Система кондиціонування	УКВ-31
Туалет	замкнутого типу з баком-накопичувачем
Ємність системи водопостачання, л	670
Система пожежної сигналізації	УПС-ТМ
Установка пожежогасіння пульта електроустаткування	ОСП-1
Система водяного пожежогасіння	є
Поїзна система зв'язку	
Система управління й діагностики	

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

16

Зовнішнє фарбування вагона виконується із застосуванням сучасних технологій і матеріалів, що відповідають міжнародним стандартам. Колір зовнішнього фарбування вагона визначається замовником.

На вагоні встановлюються алюмопластикові віконні блоки. Трапи робочої сторони обладнаються додатковими підніжками.

Внутрішнє планування приміщень вагона залишається існуючої. Установлюються каркасні металло-стеклопластикові внутрішні двері, виробляється повна заміна елементів меблів купе й службових відділень. Установлюються нові системи опалення й водопостачання, із заміною всіх вузлів санітарно-технічного устаткування.

Модернізація вагона містить у собі:

– устаткування вагона моноблочною установкою кондиціонування повітря з живленням від підвагоного генератора через трифазний перетворювач, що забезпечує частотний пуск і регулювання холодовиробництва;

– установку принципово нової для вагонів даного типу системи електропостачання з живленням від підвагоного генератора змінного струму 110В потужністю 35 кВа;

– установку екологічно чистого туалету (ЕЧТ) з підвагоним баком-накопичувачем;

– модернізовану систему освітлення салону.

Підвагоне електроустаткування виконане у вигляді модулів, установлених у підвагонних ящиках на вкатних візках, що забезпечує заміну блоку, що вийшов з ладу, протягом 20-30 мін. Конструкція високовольтного електроустаткування вагона дозволяє робити вимір опору ізоляції високовольтної магістралі без яких-небудь підготовчих операцій.

Живлення електроугольного казана здійснюється від поїзної магістралі 3 кВ постійного або змінного току.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18



- | | |
|--|---|
| 1 – моноблочна установка кондиювання повітря | 8 – пристрій вступний |
| 2 – воздуховод | 9 – ящик низьковольтної апаратури |
| 3 – бак системи водопостачання | 10 – акумуляторні бокси |
| 4 – ресивер ЕЧТ | 11 – пульт управління електроустановкам |
| 5 – перетворювач трифазний кондиціонера | 12 – бак-накопичувач ЕЧТ |
| 6 – ящик високовольтний | 13 – блок туалетний ЕЧТ |
| 7 – генератор змінного струму | 14 – унітаз |
| | 15 – компресорна установка ЕЧТ |

Рисунок 2.1 – Схема купейного вагона, вид збоку



- | | |
|-------------------------------------|---------------------------------------|
| 1 – тамбур гальмового кінця вагона | 7 – коридор негальмового кінця вагона |
| 2 – котельне відділення | 8 – туалет |
| 3 – службове відділення | 9 – тамбур негальмового кінця вагона |
| 4 – коридор гальмового кінця вагона | 10 – коридор пасажирського приміщення |
| 5 – купе відпочинку провідника | 11 – екологічно чистий туалет |
| I-IX — пасажирські купе | |

Рисунок 2.2 – Схема купейного вагона, вид зверху

Для постачання вагона в шляху проходження живленням 220 В 50 Гц установлений перетворювач.

Система енергопостачання складається з:

- підвагоного високовольтного апаратного ящика 2Я.111.В1.
- високовольтного статичного перетворювача SG 600 380 ВР;
- вхідного реактивного опору SG 600 417;

- трансформатора SG 600 419;
- інвертора статичного трифазного SG 600 388 SP;
- реактивного опору SG 600 41 8;
- акумуляторної батареї 84 ТНЖ-300 110 У (на викатних візках);
- апаратного підвагоного ящика підключення зовнішньої напруги трифазної мережі 380/220В 50 Гц.

На вагоні встановлюються:

моноблочна установка кондиціонування повітря ЛТ УКВ ПВ із екологічно безпечним холодоагентом HC318;

- ЕЧТ "Омега-2" виробництва концерну "Вагонсистем";
- система пожежної сигналізації "Прометей-02" (зі світловою сигналізацією й голосовим оповіщенням) виробництва НПП "Хартрон-Експрес";
- система відеоспостереження (монітор і дві відеокамери);
- інформаційний дисплей;
- світлова індикація зайнятості туалету;
- система виклику провідника в купе;
- світильники перехідних площадок;
- світильники з індивідуальними перетворювачами;
- електрична піч (котельне відділення).

Службове відділення обладнане:

- холодильником;
- мікрохвильовою піччю;
- пультом управління вагоном фірми "PMS", Іспанія;
- пультом управління туалетною системою типу ВТБН.

Усе вищеперераховане обладнання зав'язано у єдину автоматизовану систему, для якої необхідне програмне забезпечення.

Виконання передбачених проектом робіт дозволяє істотно збільшити строк експлуатації вагона, підвищити експлуатаційні характеристики вагона при

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

одночасному зниженні витрат по його технічному обслуговуванню, істотно підвищити комфортність і якість обслуговування пасажирів.

Таким чином виходячи з усього вищеперерахованого автоматизована система повинна включати у себе наступні компоненти:

- управління моноблочною установкою кондиціонування повітря ЛТ УКВ ПВ із екологічно безпечним холодоагентом HC318;
- управління системою пожежної сигналізації "Прометей-02" (зі світловою сигналізацією й голосовим оповіщенням);
- управління системою відеоспостереження (монітор і дві відеокамери);
- управління світловою індикацією зайнятості туалету;
- управління системою виклику провідника в купе;
- управління світильниками перехідних площадок;
- управління електричною піччю (котельне відділення);
- пульт управління вагоном;
- пульт управління туалетною системою типу ВТБН;
- управління автоматизованою діагностикою електрообладнання
- усі дані виносяться на інформаційний дисплей.

Опис автоматизації діагностики електрообладнання купейного вагону

Роботи з діагностики ведуться на підставі спеціально розроблених діагностичних схем для кожного типу вагона залежно від року й заводу побудови. Всі графічні й інші позначення номерів проводів, апаратів, вузлів, елементів відповідають електричним схемам заводів-постачальників. При розробці діагностичних електричних схем урахується побудова електричних принципів і монтажних схем вагонів побудови вітчизняних заводів і колишньої НДР, принципи роботи електроустаткування вагонів і правила їхньої експлуатації. Всі роботи з діагностики електроустаткування вагонів на стоянці й у русі повинні проводитися відповідно до діагностичних схем, рекомендаціями з обліком технічної й електричної безпеки.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Для діагностики використовуються ПЕОМ, яка входить в автоматизовану систему управління купейним вагоном, і цифрові засоби вимірів, які передають дані на ПЕОМ, де вони обробляються й у випадку ві дхилення видається відповідний сигнал, що попереджає про небезпеку.

Діагностика електроустаткування купейного пасажирського вагона з кондиціонуванням повітря

Діагностування на стоянці.

Перед діагностуванням необхідно систему управління заживити напругою акумуляторної батареї шляхом віддалення запобіжника 1e1 (на струм 224 А – не забути поставити після закінчення ремонту на місце) – це необхідно для виключення розряду акумуляторної батареї через якір генератора, і постановки перемички перетином 1,5×2,0 мм (не забути зняти після діагностування й ремонту) між клемми 13 РНГ і 1 РНГ (рисунок 2.3). На цій діагностичній схемі (див. рисунок 2.3) робочими вузлами є: 1 і 4 – підвагоні ящики з резисторами; b2 – пускова кнопка, a1 – рубильник коробки зовнішнього живлення напругою 380В, МБ 1/5 – клемна рейка заднього монтажу в нижній частині розподільної шафи, L2 і L3 – клемні рейки переднього монтажу розподільної шафи.

Дозволяє провести діагностику наступних вузлів і агрегатів:

- Перевірка цілісності ланцюгів порушення.
- Перевірка ланцюгів пускової кнопки двигуна генератора в коробці зовнішнього живлення під вагоном.
- Перевірка ланцюгів порушення генератора.
- Перевірка цілісності діодів.
- Перевірка силових ланцюгів генератора.

Діагностування при русі вагона.

Дозволяє провести діагностику наступних вузлів і агрегатів:

- Перевірка ланцюгів генератора.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

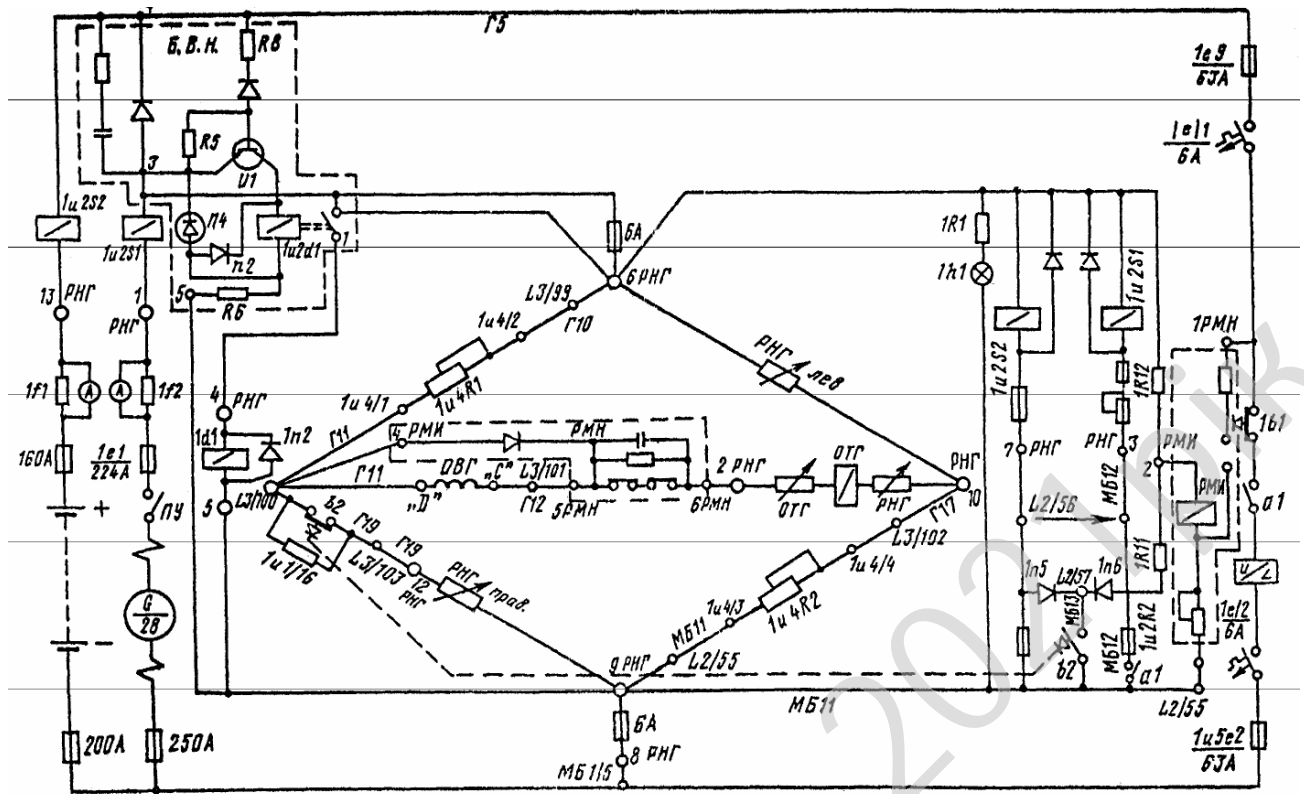


Рисунок 2.3 – Схема діагностування електроустаткування пасажирського купейного вагона з кондиціонуванням повітря

Діагностика електроустаткування купейних вагонів без кондиціонування повітря типу

На вагонах типу 47 Д/К застосовується система електроустаткування ЕВ.10.02.24 В2, а на вагонах типу 61.425 – система ЕВ.10.02.26. Діагностика ведеться по спеціально розробленій діагностичній схемі (рисунок 2.4).

Діагностика електроустаткування на стоянці.

Дозволяє провести діагностику наступних вузлів і агрегатів:

– Подача напруги акумуляторної батареї на обмотки генератора й ланцюга порушення.

– Діагностика тиристорного РНГ типу 2Б.231.

– Перевірка реле Р6.

– Перевірка силових ланцюгів генератора.

Діагностика електроустаткування при русі вагона.

Дозволяє провести діагностику наступних вузлів і агрегатів:

– Діагностика ланцюгів порушення генератора.

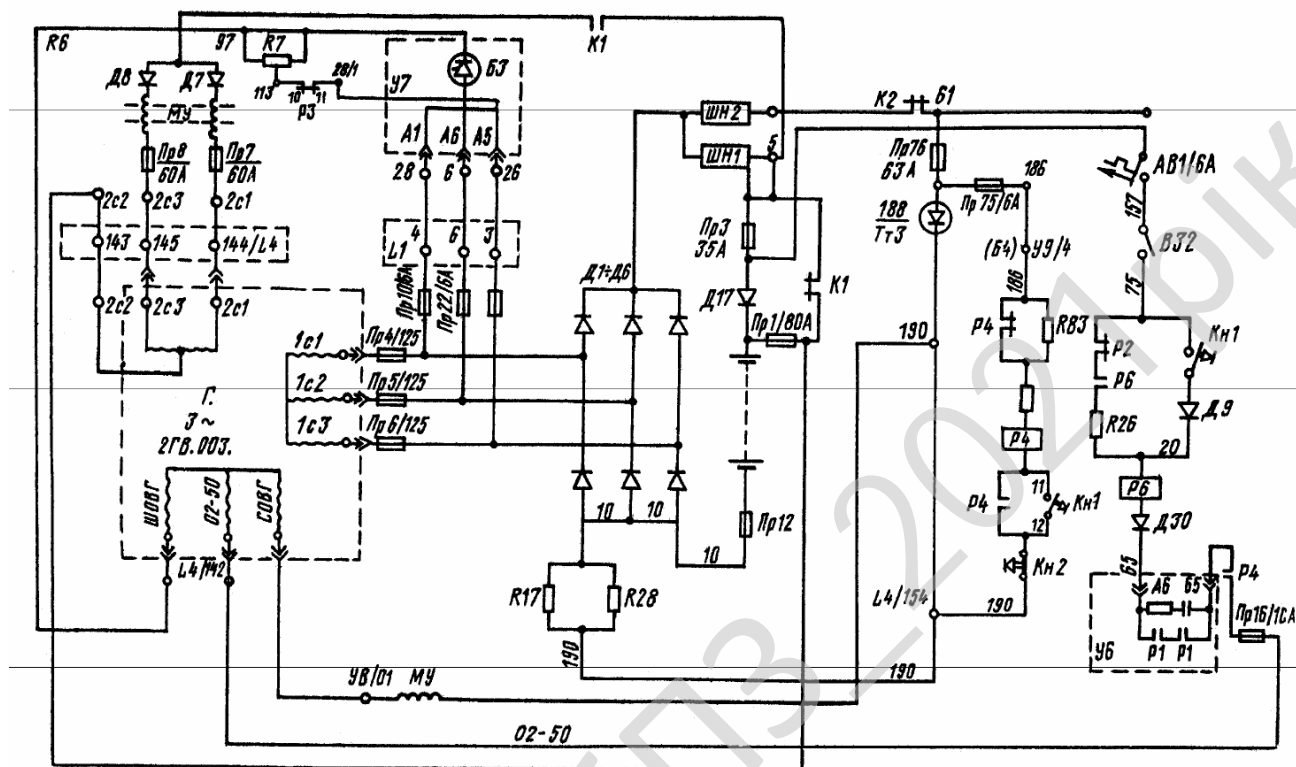


Рисунок 2.4 – Схема діагностування електроустаткування купейних вагонів без кондиціонування повітря типу 47Д/К и відкритих вагонів типу 61-425

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 приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

24

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.

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++

						ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			27

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

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

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 пока зу 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 компонентів на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізуемі 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 стилізуємий FMX компонент TМето 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.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

						ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			29

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 другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для автоматизованої інформаційно-діагностичної системи поїзду.

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

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

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі.

Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів

						ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			30

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) сформуванати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

У промисловій автоматичі звичайно використовують стандарт 4-20 мА для передачі значень. Незважаючи на те, що складні розподілені системи управління грають усе більшу роль в автоматизації виробництва, зв'язок приладів з контролерами по колишньому виглядає як з'єднання " точка-точка". Однак в останні роки все більш широко поширюється концепція впровадження локального інтелекту в промислові прилади й забезпечення зв'язку між рівноправними вузлами мережі. Успіх цього підходу залежить від комунікаційної мережі, що зв'язує пристрої, розташовані на об'єктах управління. Недавно були розроблені кілька запатентованих мереж, широко відомих за назвою "системи fieldbus". Даний розділ присвячений поглибленому порівняльному вивченню трьох перспективних протоколів fieldbus, а саме: Controller Area Network (CAN), Process Fieldbus (PROFIBUS) і Field Instrumentation Protocol (FIP). Досліджуються основні характеристики цих мереж (реакція в реальному масштабі часу, працездатність, придатність для управління процесом). Проведені дослідження показують, що Profibus і FIP – сильні суперники в приладовому оснащенні процесів управління. У той час, як в FIP реалізуються нові концепції, наприклад поділ пропускної здатності шини, шляхом виділення інтервалів часу при використанні синхронізованих тактових генераторів, Profibus ґрунтується на звичайній передачі маркера із циклічним інформаційним обслуговуванням для задоволення потреб у реальному часі. Вивчення витрат на впровадження показує, що протоколи Profibus можуть бути реалізовані з незначним додатковим устаткуванням, оскільки основна передача даних асинхрона й ведеться посимвольно. Що стосується CAN, те вони прекрасно підходять там, де потрібно

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

досить незначний час очікування (порядку 5 мс) і необхідно з'єднати велике число при малих витратах.

Мережа fieldbus зв'язує промислові прилади (сенсори, виконавчі механізми, пристрої вводу-виводу й локальні регулятори) на виробничому рівні. Вона має нові функції, такими, як автоматичне калібрування промислових приладів, самотестування, завантаження на згадку значень параметрів, конфігурування, діагностика в реальному часі, попереднє обчислення вимірюваних величин, моніторинг мережі й т.д. Подібні мережі знижують витрати на кабелі й поліпшують доступ до устаткування. У кожному разі fieldbus повинна забезпечувати виконання критичних вчасно операцій. Отже, інтелектуальні промислові прилади збільшують робоче навантаження fieldbus у міру зростання числа їхніх функцій.

Загальні вимоги до системи fieldbus

Fieldbus – це повністю цифрова двунаправлена багатоточкова комунікаційна система, використовувана для зв'язку приладів на об'єктах із системами в операторській. Одне із самих основних властивостей системи fieldbus полягає в тому, що вона підтримує двунаправлений зв'язок з безліччю змінних величин. Фізично в fieldbus можуть використовуватися три види топологій між'єднань: двухточкова, деревоподібна й багатоточкова. Середнє число приладів, під'єднаних до мережі може коливатися від 50 до 200 при довжині мережі 100 – 1000 м. Крім фізичного з'єднання приладів мережа повинна забезпечувати виконання наступних функцій:

- взаємодія устаткування, що надійшло від різних виготовлювачів;
- просте додавання й видалення пристроїв;
- від'єднання пристрою для проведення ремонту й наступне його включення в мережу, при яких не створюються перешкоди для роботи інших пристроїв і користувальницького завдання, а також не виявляються впливи на динаміку всієї мережі;

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

- сигналізацію про те, що промисловий прилад вийшов з ладу або перебуває в ремонті;
- можливість перевірки всього комплексу устаткування із заданої точки або із всіх точок мережі;
- здатність визначати поточний стан будь-якого під'єданого пристрою за допомогою спеціального пристрою мережі fieldbus;
- можливість пристроїв мережі fieldbus подавати запит на самоконтроль приладу, виявляти в мережі новий пристрій, подавати запит на ідентифікацію приладу;
- просту модифікацію процесу, фізичне додавання або видалення приладів при незначній модифікації користувальницького програмного забезпечення.

На користувальницькому рівні система fieldbus повинна володіти:

- можливістю пересилати циклічні й ациклічні дані;
- двома або чотирма рівнями пріоритетів для повідомлень;
- здатністю: зчитувати й записувати значення змінних у режимах " точка-точка", груповому або широкомовному;
- пускати в хід виконавчі пристрої;
- визначати поточний стан сенсорів і їхнього порушення;
- запам'ятовувати конфігурацію пристроїв;
- проводити точну ідентифікацію приладу;
- синхронізувати роботу двох станцій.

Кожна система fieldbus повинна, у такий спосіб підтримувати загальну Службу повідомлень fieldbus (FMS), щоб забезпечувати виконання перерахованих користувальницьких вимог. Специфікації fieldbus повинні підтримувати також тимчасово під'єдані пристрої. Горизонтальний інформаційний потік в fieldbus характеризується головним чином передачею коротких повідомлень у заданий час. Інтеграція fieldbus у глобальне середовище повинна бути можлива для вертикальних інформаційних потоків, спрямованих на

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

диспетчерський рівень. Більша частина горизонтальних інформаційних потоків буде в основному циклічного типу із часом циклу довжиною 0,25..2 із із типовою довжиною 1..5 байт на одне присоединенное пристрій, із затримками в передачі повідомлень порядку 100 мс.

Система Profibus

Стандарт визначає необхідні функції, що дозволяють пересилати дані між пристроями, виготовленими різними виробниками. Фізичний рівень, рівень каналів передачі даних і управління системою fieldbus для обох рівнів визначені в стандарті DIN 19245-1. Специфікації повідомлень fieldbus (Fieldbus Message Specification- FMS), аналогічні Специфікаціям виробничих повідомлень (Manufacturing Message Specification MMS); інтерфейс нижнього рівня (Lower Layer Interface LLI) і управління системою fieldbus на цьому рівні визначені в DIN 19245-2. Цей стандарт націлений на реалізацію протоколу за допомогою однієї комерційно доступної інтегральної схеми, що містить однокристальний мікроконтролер і внутрішній універсальний синхронний приймачепередатчик, що мінімізує вартість взаєбагато з'єднання пристроїв, розташованих на об'єктах. Мережа містить провідні й ведені станції. Провідна станція може управляти системою й передавати повідомлення, коли вона має право доступу (маркер). На відміну від її ведена станція може лише підтверджувати отримане повідомлення або пересилати інформацію з удаленному запиту. Маркер циркулює по логічному кільцю, утвореному провідними станціями. Таким чином, може бути реалізована або централізована система, або система, що повністю працює в режимі точка-точка, або гібридна. Швидкість передачі лежить у діапазоні від 9,6 Кбіт/с до 2 Мбіт/с. Для критичних вчасно завдань рекомендується система з 32 провідними станціями. Можлива як ациклічна, так і циклічна передача даних з 255 байтами в кадрі.

PROFIBUS-PA і FOUNDATION™ fieldbus мають ряд загальних характеристик (таблиця 3.1, таблиця 3.2):

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

- обидві системи задовольняють вимогам специфікацій фізичного рівня H1 IEC/ISA, які визначають середовище передачі даних;
- обидві системи іскробезпечні й здатні по тим самим проводам передавати як дані, так і електроживлення для підключених до мережі пристроїв, що дозволяє використовувати їх у вибухонебезпечних зонах;
- обидві системи підтримуються міжнародними організаціями, що поєднують як кінцевих користувачів, так і постачальників;
- обидві системи можуть бути розгорнуті як цифрова заміна аналогових каналів 4-20 мА з використанням тих же самих, уже існуючих ліній зв'язку;
- обидві системи підтримують роботу в багатоточечному режимі, завдяки чому знижуються витрати на монтаж і обслуговування кабельного господарства.

Однак між мережними системами є й істотні розходження (таблиця 3.1, таблиця 3.2). Хоча обидві системи здатні управляти подіями в самій мережі, застосовувана в PROFIBUS-PA комунікаційна модель «головний-підлеглий», а також відсутність протоколу системного адміністрування роблять PROFIBUS-PA незадовільним рішенням для управління розподіленими процесами. FOUNDATION™ fieldbus, навпроти, створювалася не тільки для організації обміну цифровою інформацією між керуючим пристроєм мережі й пристроями нижнього рівня (польового устаткування), але й для розподіленого управління, включаючи підтримку функції автоматичного конфігурування (plug-and-play), що істотно розширює границі сумісності устаткування. FOUNDATION™ fieldbus при передачі даних одночасно підтримує маркерний доступ і обмін за розкладом. Таким чином, дані, передані між функціональними блоками прикладної програми, що виконується на різних вузлах мережі, можуть бути точно синхронізовані за часом. Виконання функціонального блоку координується з передачами по шині, тому що кожний пристрій містить синхронізуємий таймер. Таким чином, контур управління, розподілений між декількома пристроями, може завершити операцію в найкоротший час. Це, у свою чергу, приводить до зменшення часу запізнювання й збільшенню швидкодії контуру.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Таблиця 3.1 – Порівняння комунікаційних протоколів

	PROFIBUS-PA	FOUNDATION fieldbus
Фізичний рівень	Стандарт IEC 61158-2	Стандарт IEC 61158-2
Швидкість обміну	31,25 кбіт/с	31,25 кбіт/с
Живлення пристроїв по лінії зв'язку	Так	Так
Використання існуючої кабельної інфраструктури	Так	Так
Робота у вибухонебезпечних зонах	Так	Так
Канальний рівень	802.4 (передача маркера, «ведучий-підлеглий»)	ANSI S50.02-3,4; TS-61158-3,4 (спеціально розроблений для польової шини)
Зв'язок «точка-точка»	Немає	Так
Синхронізація за часом	Немає	Так
Періодичні сеанси обміну	Опитування виконується провідним пристроєм	Планована підписка вузлів на дані, публікуємі іншими вузлами
Прикладний рівень	Розширення DP	Fieldbus Messaging (FMS) — обмін повідомленнями
Функціональні блоки	Типи блоків обмежені профілем пристрою	Повністю визначені й можуть бути розширені виробником пристроїв
Мова опису пристроїв	Немає	Так
Системное адміністрування	Немає	Так
Пошук тегу	Немає	Так
Присвоєння адреси	Немає	Так
Виконання функціонального блоку за розкладом	Немає	Так

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

37

Таблиця 3.2 – Порівняння переваг

Переваги польової шини	PROFIBUS-PA	FOUNDATION fieldbus
Початкова економія на вартості монтажних матеріалів	Так	Так
Ідентифікація пристрою	Так	Так
Діагностична й регламентна інформація	В обмеженому обсязі	Так
Віддалене конфігурування пристроїв	В обмеженому обсязі	Так
Віддалене калібрування	Немає	Так
Управління на рівні датчиків і виконавчих механізмів (польових пристроїв)	Немає	Так
Обробка аварійних подій і тренди	Немає	Так
Розширене подання про контрольований процес	Так, однак імена тегів і параметрів не зберігаються в пристрої	Так
Воля вибору постачальника устаткування	Обмежена	Так
Підтримка декількох провідних вузлів	Обмежена. Додавання наступного провідного пристрою впливає на тривалість циклу опитування	Так. Кількість провідних вузлів не впливає на характеристики шини
Класи вироблених пристроїв	Пристрої цифрового й аналогового вводу/виводу з функціями збору даних	Пристрої цифрового й аналогового вводу/виводу з функціональними блоками

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

38

Прикладний рівень в FOUNDATION™ fieldbus забезпечує підтримку квітованої взаємодії між клієнтом і сервером, що може використовуватися для зміни оператором значень вставок, віддаленого завантаження й налаштування параметрів конфігурації. Крім того, підтримується розсилання оповіщень про аварійні події і їхні підтвердження. Це засновано на тому же прикладному рівні, що використовується в PROFIBUS-FMS. В PROFIBUS-PA один ведучий вузол використовує протокол DP для опитування підлеглих вузлів, що містять функціональні блоки вводу/виводу. Час опитування всіх вузлів мережі залежить від кількості вузлів і ряду інших факторів, тому детермінованим може бути тільки час початку опитування. На прикладному рівні PROFIBUS-PA замість FMS використовує розширення DP, що приводить до обмеження можливостей по віддаленому конфігуруванню, а також по читанню й запису.

Система FIP

FIP являє собою багатопрофільну систему реального часу для управління процесами й комплексними автоматизованими виробництвами (СІМ). Швидкість передачі даних лежить у межах від 31,25 Кбіт/с до 2,5 Мбіт/с. Зв'язок не будується за принципом зв'язку джерела із приймачем. Адреса джерела являє собою ім'я точно ідентифікованого об'єкта. Наприклад, вимірювана змінна процесу це об'єкт. Всі технологічні об'єкти, підключені до мережі, знають і називають об'єкт по його унікальному імені. Арбітр мережі посилає об'єкт у запропонованому порядку в організований список. Система FIP має головним чином періодичний трафік. Аперіодичні інформаційні повідомлення типу подій передаються у вигляді обміну запитами, що супроводжують циклічну передачу даних, з аперіодичним відкриттям вікон аперіодичної передачі даних. FIP це система fieldbus, що функціонує як розподілена база даних реального часу. Часова й просторова несуперечність даних гарантується завдяки локальним зчитуванню й запису даних, про що буде сказано далі.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Система CAN

CAN являє собою протокол послідовного зв'язку, ефективно підтримуючий розподілене управління в реальному часі з дуже високим рівнем захисту. Система має широкий діапазон застосувань: від високошвидкісних мереж до недорогого ущільненого монтажу. Різні підсистеми зв'язуються між собою з допомогою CAN при швидкості передачі 1 Мбіт/с. Інформація посилає по каналі у вигляді повідомлень фіксованого формату.

Вузол CAN не використовує якої-небудь інформації про конфігурацію системи (адреса станції). Змісту повідомлення привласнюється ім'я (ідентифікатор). Ідентифікатор не вказує на саме повідомлення, але описує інформацію, що втримується в ньому. Таким чином, всі вузли мережі можуть вирішувати, фільтруючи повідомлення, чи належно оброблятися на них ця інформація чи ні. Як наслідок з концепції передачі повідомлення, будь-яке число вузлів може одержувати й одночасно відпрацьовувати те саме повідомлення. Отже, погодженість даних у системі досягається шляхом групового використання даних і обробки помилок. Завжди, коли канал вільний, будь-який вузол мережі може почати передачу повідомлення. Конфлікти в системі дозволяються за допомогою поразрядного арбітражу. Під час арбітражу кожний передавач порівнює рівень переданого біта з рівнем біта в каналі. Коли посилає рецесивний рівень, а виявляється домінуючий, блок вважається програвшим арбітраж і повинен бути відкликаний без посилки біта.

Загальне число блоків, що може бути охоплено мережею CAN, обмежено лише часом затримки й електричним навантаженням лінії зв'язку.

Багаторівнева структура

Кожний мережний протокол звичайно порівнюють із багаторівневою ISO-Моделлю й між ними встановлюють відповідність. Систему PROFIBUS можна прямо звести до ISO-Моделі з порожніми рівнями 3. Апаратура, канал передачі даних і управління визначені в розділі 1; FMS, LLI і управління рівнем в у розділі 2. Канальний рівень ділиться на підрівні Medium Access Control -MAC (рівень

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

доступу в середовище) і Fieldbus Logical Control – FLC (логічне управління fieldbus). MAC забезпечує протокол доступу в гібридне середовище. FMS описує об'єкти зв'язку, сервіс і відповідну модель із погляду партнера по комунікації. Основними завданнями LLI є організація відображення FMS і FMA** на FDL***, установлення зв'язку, відключення, диспетчеризації зв'язку й управління потоками, FMA виконує контекстне конфігурування й виправлення помилок.

Система CAN має трьохрівневу структуру: фізичний рівень, рівень пересилання й об'єктний рівень. Рівень пересилання відтворює повідомлення, одержувані на об'єктному рівні, і приймає повідомлення, які варто передати на об'єктний рівень. Рівень пересилання відповідає за бітове тактування й синхронізацію, кадрування повідомлень, арбітраж і т.д. Об'єктний рівень займається фільтрацією повідомлень, а також обробкою статусу й повідомлень.

Система FIP є також трьохрівневою моделлю з фізичним рівнем, рівнем передачі даних і рівнем додатків. Рівень передачі даних відповідає за всі функції управління в реальному часі, а саме: за вибір у реальному часі циклів сканування, підтвердження управління якістю й передачі змінної, зв'язність елементів розподіленої бази, синхронізоване квантування й управління, вибір безлічі диспетчерських послуг без внесення перешкод у трафік реального часу й т.д.

Фізичний рівень

Система PROFIBUS визначає як середовище лише екрановану кручену пару з характеристичним імпедансом 100...130 Ом. Довжина кабелю не перевищує 1200 м. Довжина лінії й число зв'язаних станцій можуть бути збільшені шляхом установки повторювачів (не більше трьох). Кабель шини повинен кінчатися так, як це описано в стандарті EIA RS-485. Кожна станція, призначена для закінчення лінії, повинна забезпечувати напруга +5В на контакті б мережного з'єднання й струм не менш 10 мА.

Максимальна відстань між вузлами може становити 2 км при 256 станціях у мережі. В CAN не визначаються характеристики драйвера/приймача й

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

середовища, що дозволяє оптимізувати відповідно до застосування середовище передачі й реалізацію рівня сигналів.

Метод передачі

У системі PROFIBUS кожний біт кодується без повернення до нуля й передається диференціальною напругою. Під час періоду мовчання незаземлена диференціальна лінія переводиться кінцевим пристроєм в одиницю. Вона передає дані як символ-орієнтовані. Система FIP передає код і інформацію таймера, кодуючи їх за допомогою Manchester II. Швидкості передачі даних визначені рівними 31,25 Кбіт/с, 1 і 2,5 Мбіт/с. Арбітраж в FIP заснований на призначенні ремінного вікна кожному вузлу для періодичних даних і призначенні вікна по запиті для аперіодичних (мал. 2); існує необхідність глобальної синхронізації тактового генератора. Аналогічно в CAN здійснюється передача двох логічних значень, що взаємно доповнюють: рецесивного й доміантного. При одночасній передачі доміантного й рецесивного бітов результатуючий канал буде доміантним. Для апаратної реалізації логічного "1" використовується логічний "0". Біт кодується без повернення до нуля. Час передачі кожного біта ділиться на не перекривають один одного сегменти: синхронізацію, проходження, фази / і 2.

Сегмент синхронізації використовується для синхронізації різних вузлів системи. Передбачається, що фронт імпульсу лежить усередині цього сегмента. Сегмент проходження служить для компенсації часу фізичної затримки. Він дорівнює подвоєній сумі часу проходження сигналу по лінії.

Фазові сегменти використовуються для компенсації фазової помилки фронту імпульсу. Ці сегменти можна вкоротити або подовжити. Рівень каналу зчитується наприкінці фази. Всі контролери CAN синхронізують на старті кадру. Таким чином, необхідний типовий допуск на генератор, що становить 1,58 % при швидкості передачі інформації з каналу, рівної 125 Кбіт/с. Оскільки системи FIP і CAN працюють при глобальній синхронізації тактових генераторів, їм на відміну від PROFIBUS потрібні тверді допуски на частоту тактових генераторів.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Обґрунтування вибору інтерфейсів послідовної передачі даних АСУ

Сучасні системи автоматизації традиційно використовують як обмін даними послідовний спосіб передачі даних. Послідовні інтерфейси відрізняються по швидкості передачі, довжині зв'язку, способі передачі, принципі передачі й топології структури шини. Параметри типових стандартних інтерфейсів послідовної передачі, представлені в таблиці 3.3.

Таблиця 3.3 – Характеристики типових стандартних інтерфейсів послідовної передачі

Інтерфейс	Стандарт	Швидкість передачі	Довжина (м)	Спосіб передачі	Принцип передачі
RS-232	EIA-232-C, CCITT v.24	19,2 Кбод	15	Рівні напруг	дуплекс точка-точка
TTY	DIN 66258-1, DIN 66248-1	19,2 Кбод	1000	Струмова петля	дуплекс точка-точка
RS-422	EIA-422, CCITT v.11	10 Мбод	1000	Різниця напруг	напівдуплекс точка-точка
RS-485	EIA-485, DIN 66259-4	1 Мбод	500	Різниця напруг	напівдуплекс багатоточка

На базі типових стандартних інтерфейсів реалізуються промислові (польові) шини типу CANBUS, VIBUS, PROFIBUS і інші. До шин, застосовуваних у виробництві систем управління купейним вагоном, пред'являються наступні основні вимоги:

- робота устаткування в широкому діапазоні температур;
- перешкодозахищеність трактів передачі даних (спосіб передачі);
- робота устаткування в реальному масштабі часу (швидкість);
- великі відстані об'єктів взаємодії (довжина);
- гнучка структура шин передачі даних (топологія шини).

Перешкодозахищеність трактів передачі даних залежить конкретно від приймачепередатчиків і фізичної лінії (тип кабелю, перетин, хвильовий опір).

Оптимальні дані для польових умов мають приймачепередатчики з диференціальними рівнями сигналів і лінії передачі на кручений парі. Системні магістралі на базі промислових шин повинні забезпечувати своєчасну й підлягаючому розрахункам передачу даних у реальному масштабі часу. Польові шини застосовуються як у централізованих, так і в розподілених системах, де відстані між об'єктами взаємодії можуть становити більше 1000 метрів. Гнучкість структури шин передачі даних припускає використання сегментированих ліній типу "лінія", "дерево", "зірка", "кілеце" на базі багатоточки.

Усім вище перерахованим вимогам цілком задовольняє промислова шина PROFIBUS. Шина PROFIBUS зовсім недавно завоювала ринок промислових систем автоматизації, але вже є загальнопоширеною й визнаною в даній області застосування.

Структура протоколів PROFIBUS орієнтована на вже встановлені національні й міжнародні норми. Так, структура протоколів PROFIBUS базується на семиуровневої моделі взаємодії відкритих систем OSI (Open System Interconnection).

Інтерфейс RS-485

Інтерфейс RS-485 – широко розповсюджений високошвидкісний і завадостійкий промисловий послідовний інтерфейс передачі даних. Практично всі сучасні комп'ютери в промисловому виконанні, більшість інтелектуальних датчиків і виконавчих пристроїв, програмувальні логічні контролери поряд із традиційним інтерфейсом RS-232 містять у своєму складі ту або іншу реалізацію інтерфейсу RS-485.

Інтерфейс RS-485 заснований на стандарті EIA RS-422/RS-485. На жаль, повноцінного еквівалентного українського стандарту не існує, тому в даному розділі пропонуються деякі рекомендації із застосування інтерфейсу RS-485.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

Традиційний інтерфейс RS-232 у промисловій автоматизації застосовується досить рідко. Сигнали цього інтерфейсу передаються перепадами напруги величиною (3...15...15)В, тому довжина лінії зв'язку RS-232, як правило, обмежена відстанню в кілька метрів через низьку завадостійкість. Інтерфейс RS-232 є в кожному PC – сумісному комп'ютері, де використовується в основному для підключення маніпулятора типу “миша”, модему, і рідше – для передачі даних на невелику відстань із одного комп'ютера в іншій. Передача виробляється послідовно, послівно, кожне слово довжиною (5...8...8)біт випереджають стартовим бітом і закінчують необов'язковим бітом парності й стоп-бітами. Інтерфейс RS-232 принципово не дозволяє створювати мережі, тому що з'єднує тільки 2 пристрої (так зване з'єднання “точка – точка”).

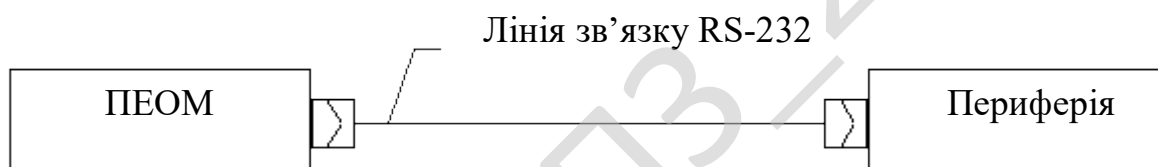


Рисунок 3.1 – З'єднання точка-точка

Сигнали інтерфейсу RS-485 передаються диференціальними перепадами напруги величиною (0,2...8) В, що забезпечує високу завадостійкість і загальну довжину лінії зв'язку до 1 км (і більше з використанням спеціальних пристроїв – повторювачів). Крім того, інтерфейс RS-485 дозволяє створювати мережі шляхом паралельного підключення багатьох пристроїв до однієї фізичної лінії (так звана “мультиплексна шина”).

У звичайному PC-сумісному персональному комп'ютері (не промислового виконання) цей інтерфейс відсутній, тому необхідно спеціальний адаптер – перетворювач інтерфейсу RS-485/232.

Перетворювач інтерфейсу ПІ-485/232, використовується при організації зв'язку між пристроями, обладнаними інтерфейсом RS-232, але, що використовують у якості середовища передачі інтерфейс RS-485.

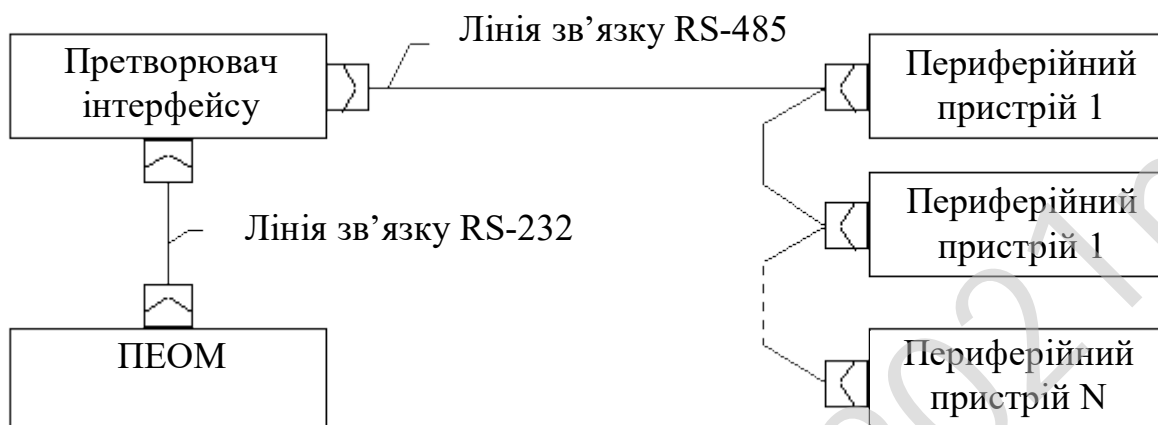


Рисунок 3.2 – Перетворювач інтерфейсу RS-485/232

Деякі технічні дані перетворювача ПІ-485/232:

- взаємне “прозоре” перетворення сигналів інтерфейсів RS-232 і RS-485 з гальванічною ізоляцією між ними;
- управління напрямком передачі здійснюється з боку RS-232 по сигналу RTS;
- вимагає наявності сигналу DTR, використовуваного для живлення перетворювача (на стороні RS-232);
- організація зв'язку між різними пристроями, протокол передачі яких використовує напівдуплексний режим (запит і відповідь передаються по одній фізичній лінії, але в різні проміжки часу);
- індикація стану сигналів інтерфейсу RS-232: Rx (прийом), Tx (передача), RTS (сигнал управління передачею);
- максимальна швидкість обміну – 19200 біт/с.

Сигнал DTR устанавлюється при запуску програмного забезпечення підключеного з боку RS-232 пристрою. Скидання DTR виробляється при

завершенні роботи програмного забезпечення. Сигнал RTS установлюється до початку передачі й скидається після повного її закінчення.

Існують і повністю автоматичні перетворювачі, що не вимагають сигналу управління передавачем, але, як правило, вони вимагають твердої вказівки швидкості обміну й довжини переданого слова (з обліком стартовий, стопових біт і біта парності).

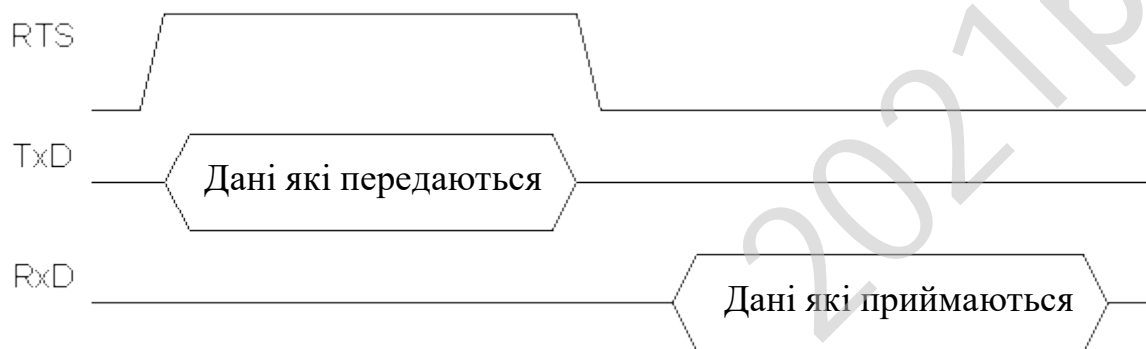


Рисунок 3.3 – Принцип управління напрямком передачі перетворювача ПП-485/232

Пристрої, що підключаються до інтерфейсу RS-485, характеризуються важливим параметром по входу приймачепередатчика: “одиниця навантаження” (“Unit Load” – UL). По стандарті в мережі допускається використання до 32 одиниць навантаження, тобто до 32 пристроїв, кожне з яких навантажує лінію в 1 UL. У цей час існують мікросхеми приймачепередатчиків з характеристикою менш 1 UL, наприклад – 0,25 UL. У цьому випадку кількість фізично підключених до лінії пристроїв можна збільшити, але сумарна кількість UL в одній лінії не повинне перевищувати 32.

Підключення перетворювача ПП-485/232 до порту RS-232 здійснюється так званим “модемним” кабелем. Перетворювач має 9-контактний роз’єм (DB9, гніздо), персональний комп’ютер може мати роз’єми як 9-контактні (DB9, штир), так і 25-контактні (DB25, штир).

Таблиця 3.1 – Для 9-контактного роз'єму розпаювання кабелю здійснюється “один в один” (у дужках зазначені номери контактів)

DB9, штир – до перетворювача	DB9, гніздо – до комп'ютера
GND (5)	GND (5)
Rx (2)	Rx (2)
Tx (3)	Tx (3)
DTR (4)	DTR (4)
DSR (6)	DSR (6)
RTS (7)	RTS (7)
CTS (8)	CTS (8)
RI (9)	RI (9)
DCD (1)	DCD (1)

Цей стандартний кабель виробляється багатьма виготовлювачами.

Перетворювач ПІ-485/232 використовує в кабелі лінії до контактів 2,3,4,5,7.

Таблиця 3.2 – Відповідність контактів роз'ємів DB9 – DB25

Найменування контакту	DB	DB2
	9	5
DCD	1	8
Rx	2	3
Tx	3	2
DTR	4	20
GND (сигнальна)	5	7
DSR	6	6
RTS	7	4
CTS	8	5
RI	9	22

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

48

Пристрої до мережі RS-485 підключаються послідовно, з дотриманням полярності контактів А і В.

Навіть для швидкостей обміну порядку 19200 біт/с кабель уже можна вважати довгою лінією, а будь-яка довга лінія для виключення перешкод від відбитого сигналу повинна бути погоджена на кінцях. Для узгодження використовуються резистори опором 120 Ом (точніше, з опором, рівним хвильовому опору кабелю, але, як правило, використовувані кручені пари мають хвильовий опір близько 120 Ом і точно підбирати резистор немає необхідності) і потужністю не менш 0,25 Вт – так званій “термінатор”. Термінатори встановлюються на обох кінцях лінії зв'язку, між контактами А і В кручений пари. Перетворювач ПІ-485/232 уже має термінатор, і при необхідності його можна включити установкою перемички між контактами 'Т' і 'Т'.

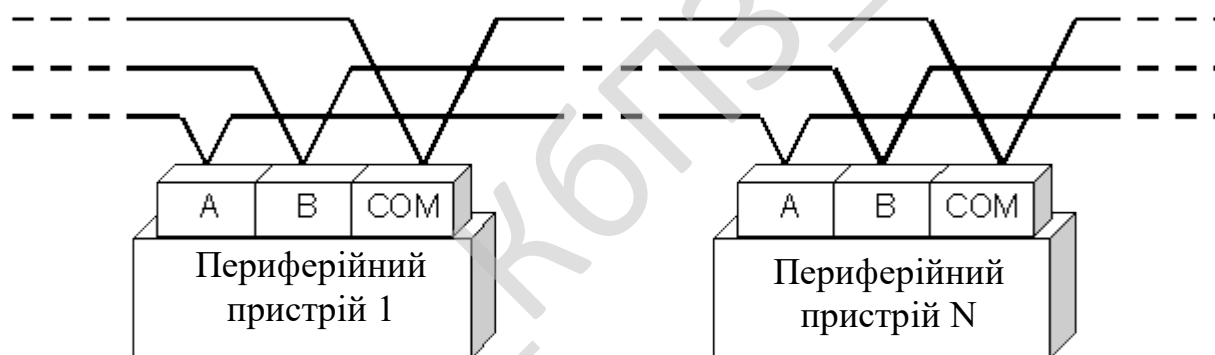


Рисунок 3.4 – Підключення до мережі через RS-485

У мережах RS-485 часто спостерігається стан, коли всі підключені до мережі пристрої перебувають у пасивному стані, тобто в мережі відсутня передача й всі приймачепередатчики “слухають” мережу. У цьому випадку приймачепередатчики не можуть коректно розпізнати ніякого стійкого логічного стану в лінії, а безпосередньо після передачі всі приймачепередатчики розпізнають у лінії стан, що відповідає останньому переданому біту, що еквівалентно перешкоді в лінії зв'язку. На цю проблему не так часто обертають уваги, борючись із її наслідками програмними методами, але проте вирішити її

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

49

апаратно нескладно. Досить за допомогою спеціальних ланцюгів зсуву створити в лінії потенціал, еквівалентний стану відсутності передачі (так званий стан “MARK”: передавач включений, але передача не ведеться). Ланцюга зсуву реалізовані в перетворювачі ПП-485/232, для їхнього підключення досить установити 2 перемички між контактами ‘+V’ і ‘+V’, ‘-V’ і ‘-V’ відповідно. Для коректної роботи ланцюгів зсуву необхідна наявність двох термінаторів у лінії зв'язку.

У мережі RS-485 можлива конфліктна ситуація, коли 2 і більше пристрої починають передачу одночасно. Це відбувається в наступних випадках:

- у момент включення живлення через перехідні процеси пристрою короткочасно можуть перебуває в режимі передачі;
- одне або більше із пристроїв несправно;
- некоректно використовується так званий “мультимастерний” протокол, коли ініціаторами обміну можуть бути кілька пристроїв.

У перших двох випадках швидко усунути конфлікт неможливо, що теоретично може привести до перегріву й виходу з ладу приймачепередатчиків RS-485. На щастя, така ситуація передбачена стандартом і додатковим захистом приймачепередатчика звичайно не потрібно.

В останньому випадку необхідно передбачити програмний поділ каналу між пристроями – ініціаторами обміну, тому що в кожному разі для нормального функціонування лінія зв'язку може одночасно надаватися тільки одному передавачу.

3.2 Розробка структурної схеми

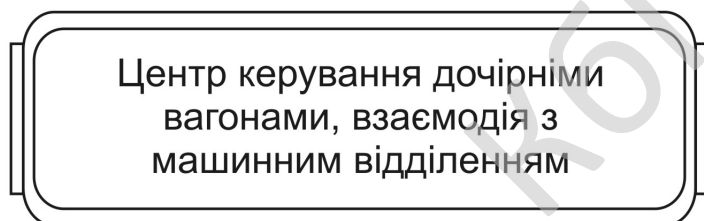
На рисунку 3.5 зображена структурна схема системи. На схемі зображена взаємодія головного вагона (центра керування) і підлеглого вагона, а також підсистеми підлеглого вагона.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

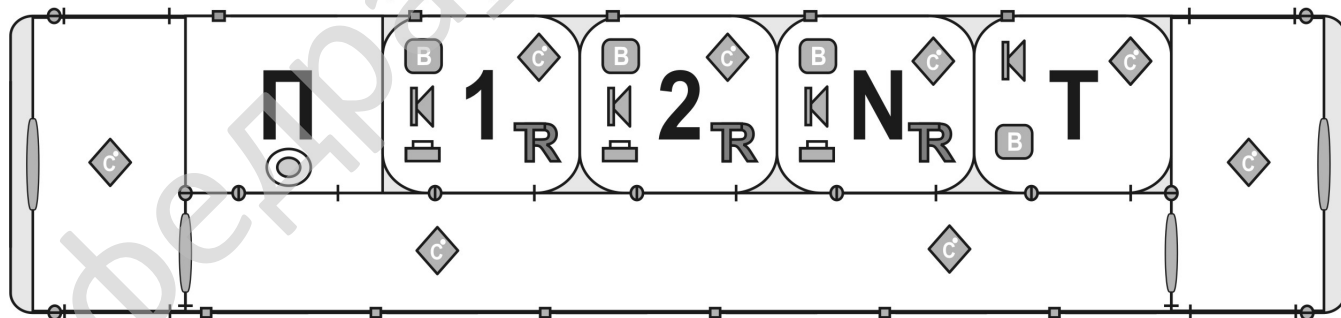
Розглянемо цю схему докладніше:

- система керування вагоном купе провідника підробно розглянута на рисунку 3.6;
- температурний датчик дозволяє уточнювати поточну температуру у вагоні;
- відеосистема дозволяє переглядати поточний відеосигнал з центра керування;
- аудіосистема аудіо сповіщення та радіо;
- кнопка виклику провідника;
- термінал доступу до мережі дозволяє виходити в Інтернет за наявності у пасажирів ноутбука апаратними засобами вагону через (Wi-Fi);
- дверний датчик контролює поточне положення;
- віконний датчик контролює поточне положення;
- динамічне табло показує наступну станцію та рекламні пропозиції.

Головний вагон



Підлеглий вагон (купе)



Умовні позначення

- | | |
|-------------------------------------|-----------------------------|
| R - термінал досту до мережі | - кнопка виклику провідника |
| - система керування вагоном | - дверний датчик |
| - температурний датчик | - віконний датчик |
| - відео система | - динамічне табло |
| - аудіо система | |

Рисунок 3.5 – Структурна схема системи

В свою чергу в купе провідника знаходиться центр керування (рисунок 3.6) який контролює поточні дії у вагоні та відсилає інформацію у центр керування. Розглянемо можливості системи керування у купе бортпровідника:

- перегляд показань температурного датчика;
- дублювання відеота аудіо сигналу;
- комутація з головним вагоном;
- перегляд положення дверних датчиків;
- перегляд положення віконних датчик;
- керувати консоллю та переглядати попередження з центру керування.

◎ Система керування вагону

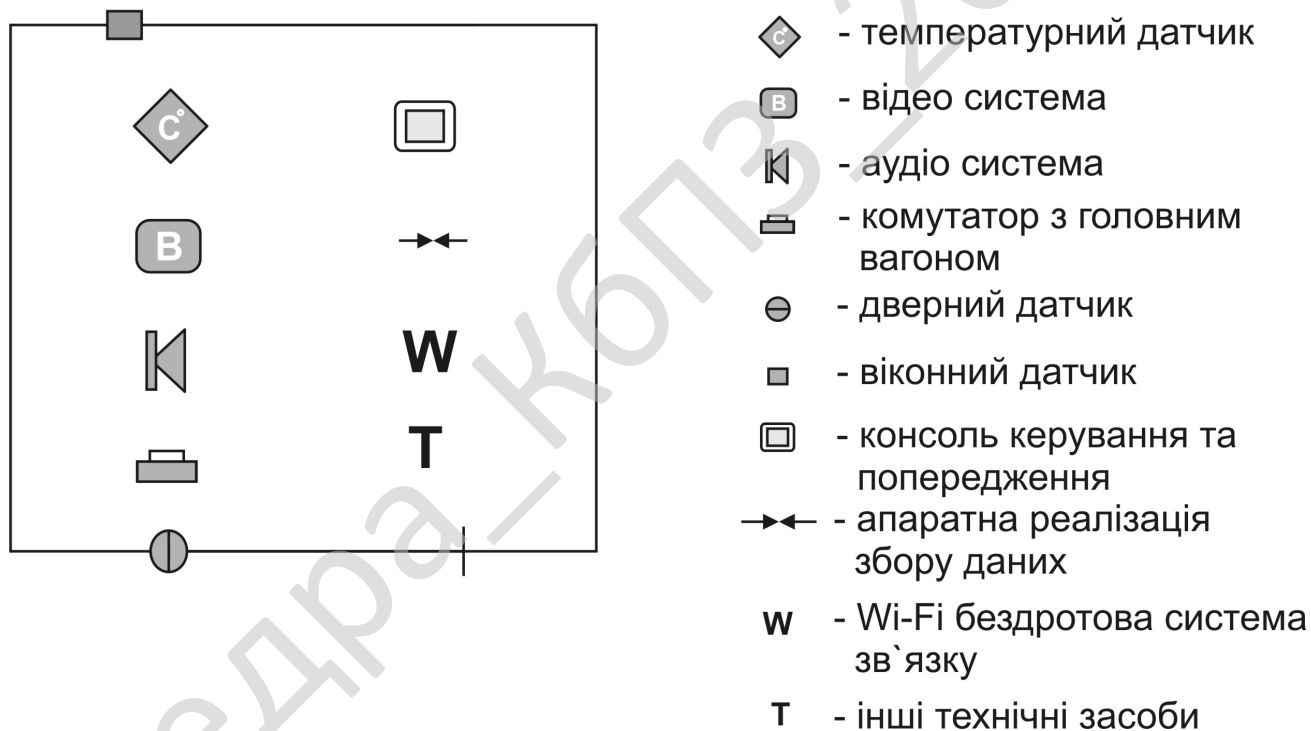


Рисунок 3.6 – Структурна схема системи керування

Також на структурній схемі системи керування відображено що крім системи керування у купе провідника знаходиться апаратна реалізація збору даних, бездротова система зв'язку (Wi-Fi) та реалізація інших технічних засобів.

3.3 Розробка функціональної схеми

Розроблена у магістерській роботі функціональна схема системи зображена на рисунку 3.7.

Автоматизована система моніторингу стану вагону розподіляється на три складові частини це – система керування (апаратна частина), програмне забезпечення (програмна частина) та комутатор з головним вагоном. Провідник являє собою диспетчером якій комплексно розглядує питання та реалізує відповідні дії.

Система керування зв'язана з комутатором. Комутатор контролює наступне: температуру у купе, температуру у тамбурі, стани вікон, стан дверцят, стан аудіо/відео, динамічне табло, спеціальні технічні параметри.

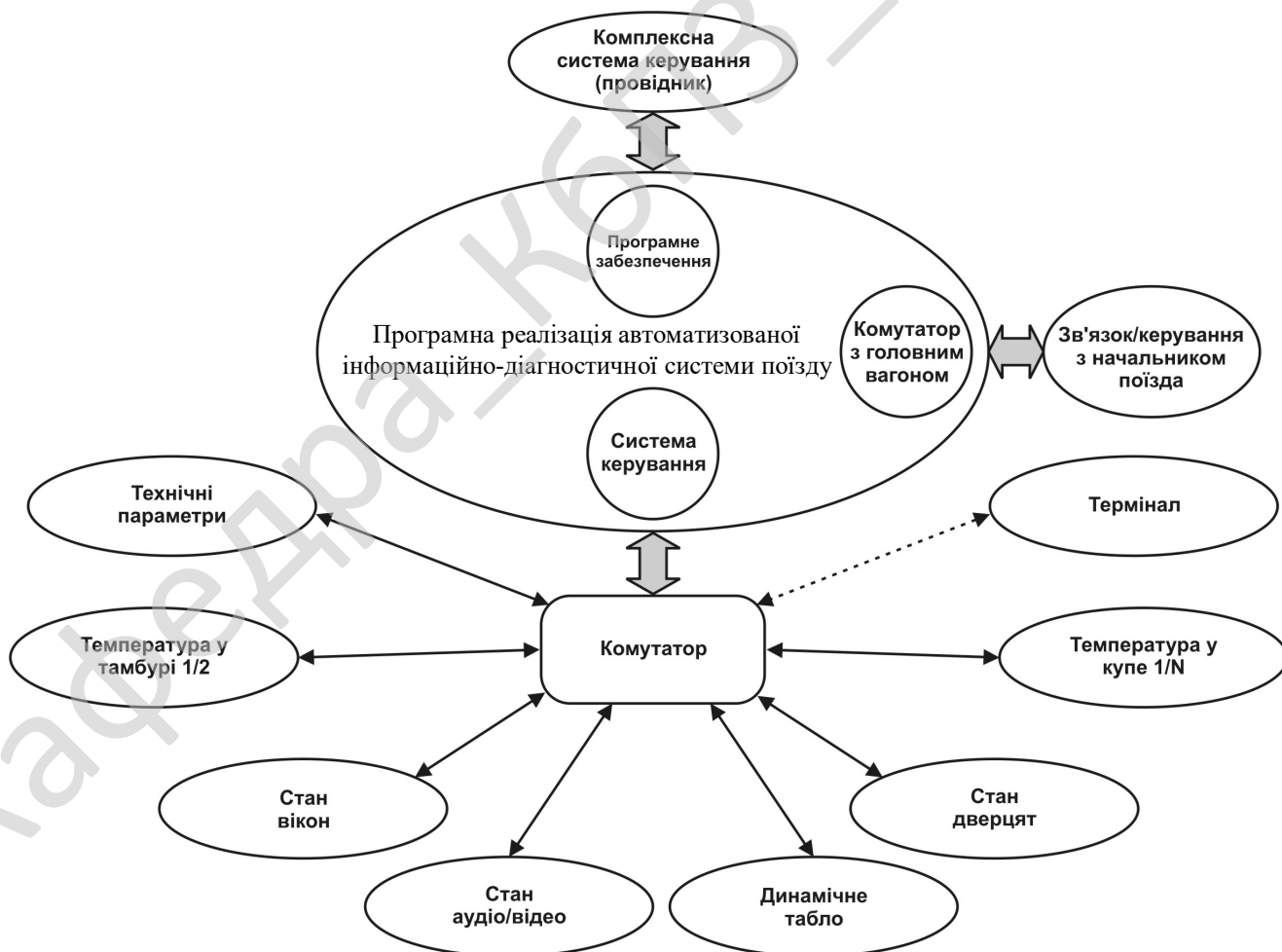
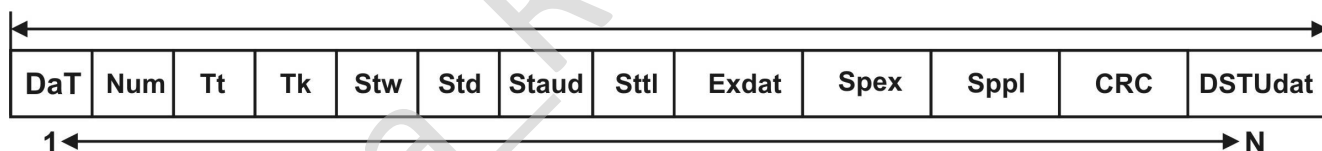


Рисунок 3.7 – Функціональна схема системи

На рисунку 3.8 зображена функціональна схема формату передачі даних.

Пакет складається з наступних полів:

- DaT дата й час заповнення пакета (тип даних TDATETIME);
- Num номер вагона (тип даних INTEGER);
- Tt температура у тамбурі (тип даних REAL);
- Tk температура у купе 1/N (тип даних REAL);
- STw стан вікон (тип даних BOActiveXAN);
- STd стан дверцят (тип даних BOActiveXAN);
- STaud стан аудіо/відео(тип даних потоковий масив);
- STtl стан динамічного табло (тип даних потоковий масив);
- Exdat інші технічні параметри;
- SPvk спец. виклик;
- SPex екстрена ситуація;
- SPpl виклик охорони поїзда (тип даних BOActiveXAN);
- CRC контрольна сума (тип даних INTEGER);
- DSTUdat дані ДСТУ (тип даних потоковий масив).



Формат пакетів центру керування:

1. DaT -Дата й час заповнення пакета.
2. Num - Номер вагона.
3. Tt - Температура у тамбурі.
4. Tk - Температура у купе 1/N.
5. STw - Стан вікон.
6. STd - Стан дверцят.
7. STaud -Стан аудіо/відео.
8. STtl -Стан динамічного табло.
9. Exdat -Інші технічні параметри.
10. SPvk -Спец. виклик.
11. SPex - Екстрена ситуація.
12. SPpl - Виклик охорони поїзда.
13. CRC - Контрольна сума.
14. DSTUdat - Дані ДСТУ.

Рисунок 3.8 – Функціональна схема формату передачі даних

3.4 Розробка діаграми процесів

Розглянемо діаграму процесів рисунку 3.9. На діаграмі зображений процес роботи й взаємодії програми, саме по діаграмі можна зрозуміти як працює розроблена програма в цілому. Починається й закінчується програма в першому блоці, що є основною крапкою звіту в діаграмі, при переміщенні по стрілках можна побачити загальну схему взаємодії блоків і їхнього входження друг у друга. Одним з важливих критеріїв при розробці будь-якого програмного забезпечення це грамотна розробка структури роботи системи потоків і процесів.

Розглядаючи діаграму можна побачити, що чітко простежується те, що взаємодія програми відбувається через блок захисту програми.

Без обробки термінальним вікном бібліотек налаштування й взаємодії із центром керування й бібліотекою алгоритму перетворення ДСТУ не можна організувати канал передачі даних і відправлення пакетів стану вагона. Також видно як комутатор впливає на процес одержання технічних параметрів вагона і його поточного стану.

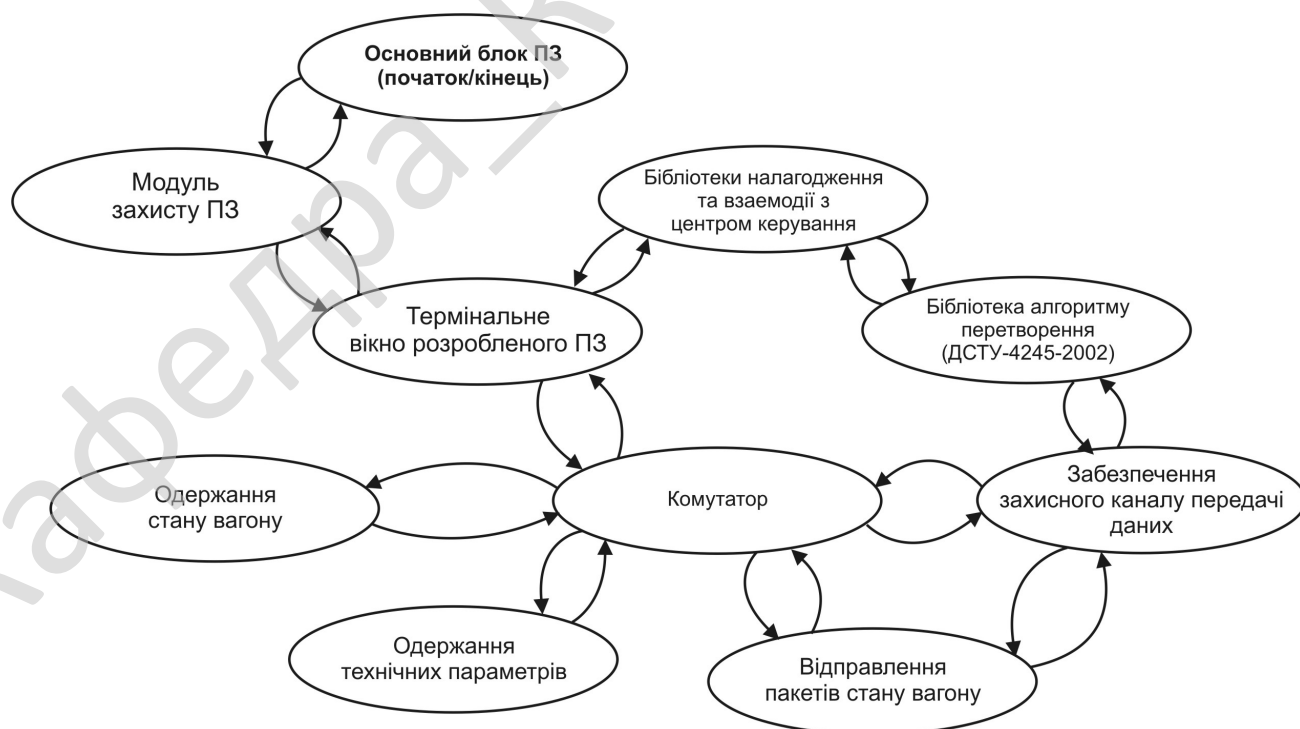


Рисунок 3.9 – Діаграма процесів

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

55

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. Блок-схеми є першоджерелами стратегії розвитку ПЗ. Тому від точності і детальної блок-схеми залежить результат всієї програми. При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єкно-орієнтована, проект, що розробляється, вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу розробці участків кода виведення інформації на екран так як програма буде використовуватися у нестандартному обладнанні.

На рисунку 4.1 зображена основна блок-схема програми. При детальному розгляді програма розбита на дещо важливих блоків, розглянемо складові елементи блок-схеми та їхній опис.

Початкові блоки програми:

- початкова ініціалізація змінних;
- підключення модулів захисту;
- ініціалізація та перевірка терміналу;
- перевірка ініціалізації терміналу;
- ініціалізація апаратної частини взаємодії;
- перевірка ініціалізації апаратної частини.

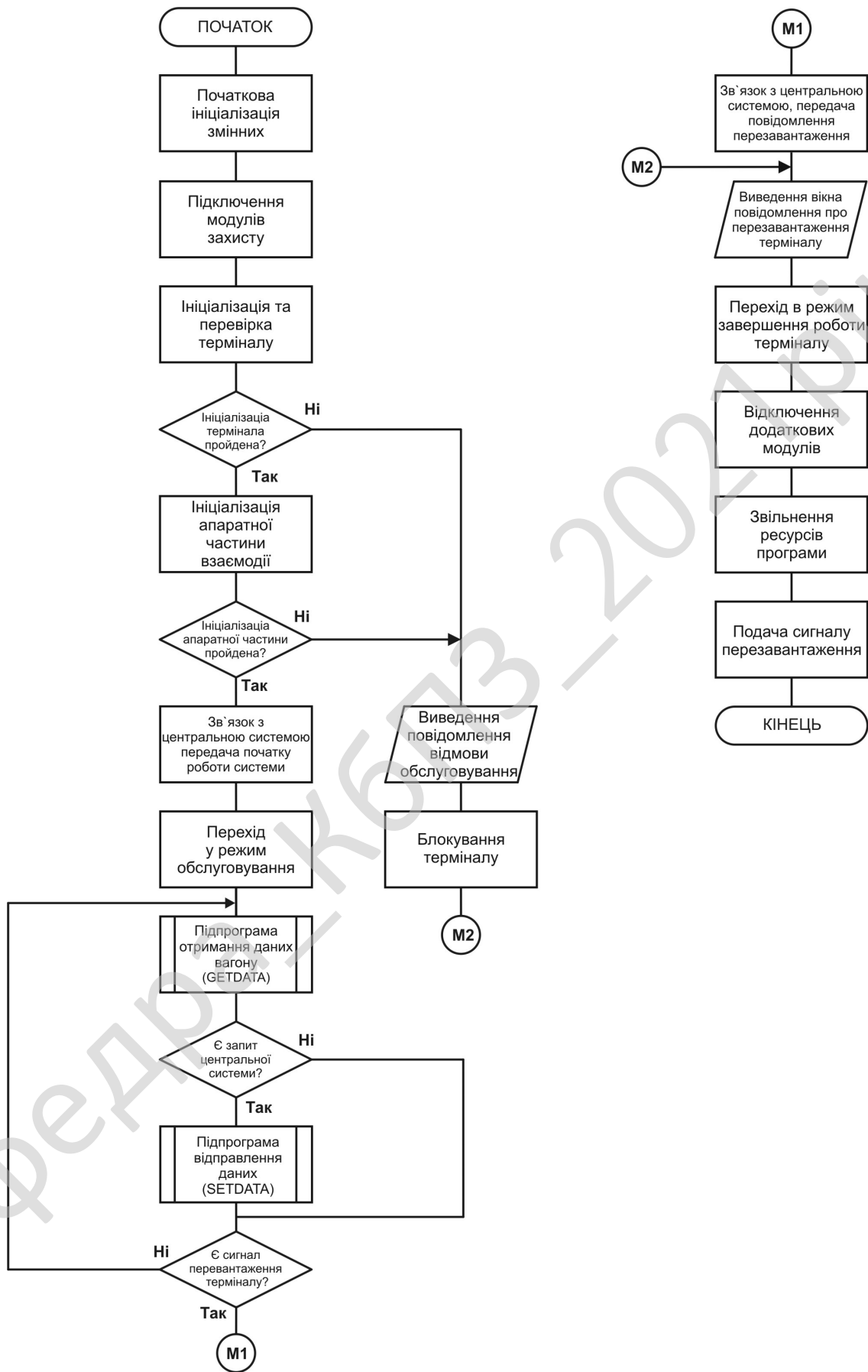


Рисунок 4.1 – Блок схема основного алгоритму програми

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

57

Підпрограма отримання даних вагону
(GETDATA)



Підпрограма відправлення даних
(SETDATA)

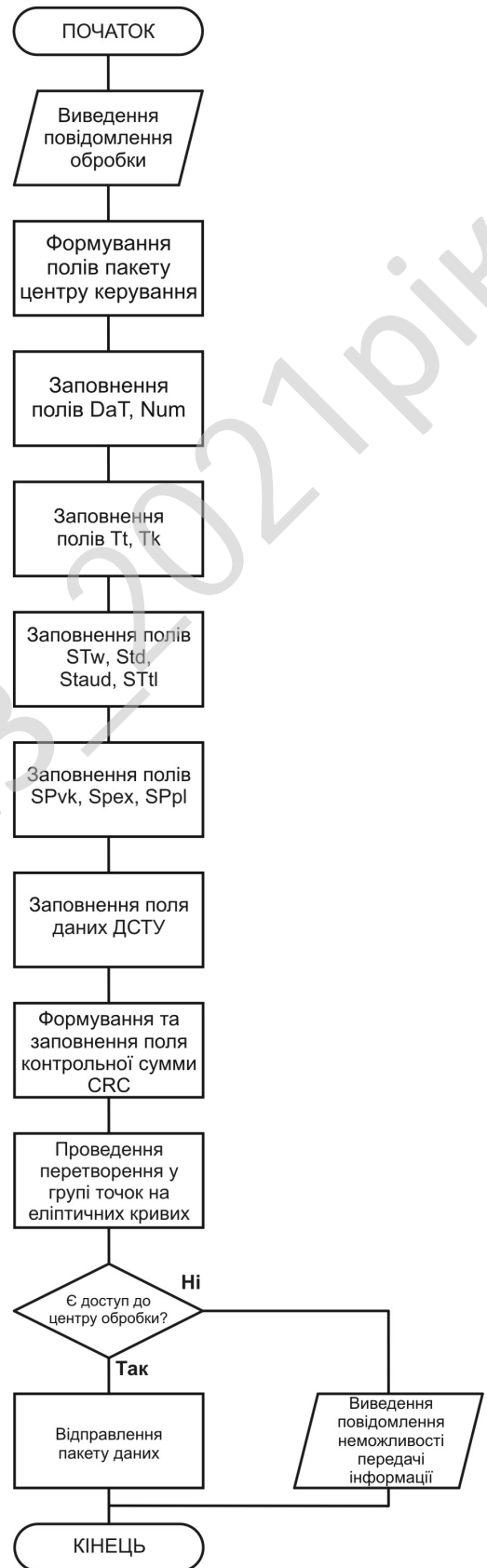


Рисунок 4.2 – Блок схема підпрограм

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

58

Робота програми:

- зв'язок к з центральною системою передача початку роботи системи;
- перехід у режим обслуговування;
- підпрограма отримання даних вагону (GETDATA, рисунок 4.2) ;
- перевірка запиту центральної системи;
- підпрограма відправлення даних (SETDATA, рисунок 4.2) ;
- перевірка сигналу перевантаження терміналу.

Завершення роботи програми:

- зв'язок з центральною системою, передача повідомлення перезавантаження;
- виведення вікна повідомлення про перезавантаження терміналу;
- перехід в режим завершення роботи терміналу;
- відключення додаткових модулів;
- звільнення ресурсів програми;
- подача сигналу перезавантаження.

Розробка програмного встановлення необхідних режимів роботи відеоадаптера

При розробці Windows додатка, іноді доводиться враховувати той факт, що він в майбутньому буде працювати на комп'ютерах з абсолютно різними моніторами й робочими розрішеннями, установленими на відеоадаптері. При розробці магістерської роботи цей факт ураховувався, також як і факт можливого застосування сенсорних моніторів. Тому при розробці магістерської програми були написані функції керування розрішеннями відеоадаптера. Вони ґрунтувалися на API функціях EnumDisplaySettings, що дозволяє одержати список доступних розрішень дисплея, а так само з функції ChangeDisplaySettings для зміни поточного відеорежиму.

Для того, щоб одержати інформацію про всі можливі режими адаптера, необхідно зробити серію викликів функції EnumDisplaySettings. Викликаючи цю

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

функцію в циклі одержуємо доступний режим, доти поки результат функції не ставав відмінним від True.

Дана функція має на вході змінну типу TDevMode, у якій містяться параметри. Сам тип TDevMode має безліч змінних, стосовних до відеоадаптеру. А саме, він містить у собі розрешення відеоадаптера в пікселях dmPelsWidth, dmPelsHeight, розрядність кольору (у бітах на піксель), підтримувана при даному розрешенні dmBitsPerPel, частота відновлення dmDisplayFrequency.

При першому виклику форми використовувався наступний код.

```
procedure TForm1.FormCreate(Sender: TObject);
var
  i      : Integer;
  DevMode : TDevMode;
begin
  i:=0;
  while EnumDisplaySettings(nil,i,DevMode) do begin
    with Devmode do
      ListBox1.Items.Add
        (Format('%dx%d %d Colors',
          [dmPelsWidth,dmPelsHeight,1 shl dmBitsperPel]));
      Inc(i);
    end;
  end;
```

Після даного коду програма одержує всі доступні режими. Для установки необхідного розрешення необхідно викликати функцію ChangeDisplaySettings яка обновляє реєстр і міняє розрешення монітора на задане. Розроблений код виклику функції:

```
procedure TForm1.Video(Sender: TObject);
var
  DevMode      : TDeviceMode;
  liRetVal     : Longint;
begin
  if EnumDisplaySettings
    (nil,ListBox1.ItemIndex,Devmode) then
    liRetVal := ChangeDisplaySettings
      (DevMode, CDS_UPDATEREGISTRY);

  SendMessage (HWND_BROADCAST,
```

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

60

```

WM_DISPLAYCHANGE,
SPI_SETNONCLIENTMETRICS, 0);

end;

```

Функція `ChangeDisplaySettings` повертає значення `longinteger`. Це значення можна використовувати для визначення успішності виконання функції, зрівнявши зі значеннями зі списку констант.

Для відстеження змін дисплея в розробленій магістерській програмі необхідно створити оброблювач для перехоплення повідомлення `WM_DISPLAYCHANGE`.

Звичайно даний прийом використовується у випадку, якщо додаток використовує у своїй роботі графіку, але з огляду на особливості магістерської роботи (застосування нестандартних сенсорних моніторів) цей метод дозволяє повністю розв'язати проблему з відеоадаптером. Код перевірки зміни режиму роботи адаптера:

```

...
type
  TForm1 = class(TForm)
    ListBox1: TListBox;
    ...
  private
    procedure WMDisplayChange(var Message: TMessage);
      message WM_DISPLAYCHANGE;
    ...
  procedure
    TForm1.WMDisplayChange(var Message: TMessage);
begin
  ShowMessage('Відеорежим змінено!');
  inherited;
end;

```

Розробка програмного коду отримання початкової інформації

Перед запуском магістерської програми необхідно перевірити мінімальні вимоги необхідні для роботи програми й зв'язку її із центром керування. Одним з найважливіших початкових параметрів є характеристики підсистеми пам'яті.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Для одержання детальної інформації про стан пам'яті комп'ютера я використовував функцію API GlobalMemoryStatus. У функцію передається змінна типу TMemoryStatus, що являє собою запис, тип якої визначений у такий спосіб:

```

type
    TMemoryStatus = record
        dwLength: DWORD;
        dwMemoryLoad: DWORD;
        dwTotalPhys: DWORD;
        dwAvailPhys: DWORD;
        dwTotalPageFile: DWORD;
        dwAvailPageFile: DWORD;
        dwTotalVirtual: DWORD;
        dwAvailVirtual: DWORD;
    end;

```

Розглянемо поля TMemoryStatus.

1. dwLength, описує довжину запису TMemoryStatus. Поле необхідно ініціалізувати функцією SizeOf до звертання до функції GlobalMemoryStatus.
2. dwMemoryLoad, містить кількість використаної пам'яті у відсотках.
3. dwTotalPhys, містить число байт установленної на комп'ютері ОЗП (фізичної пам'яті).
4. dwAvailPhys – вільна фізична пам'ять (у байтах).
5. dwTotalPageFile і dwAvailPageFile – показують загальний обсяг файлу підкачування й доступний обсяг із цієї величини (у байтах).
6. dwTotalVirtual – загальне число байтів віртуальної пам'яті, використовуваної в процесі, який викликає.
7. AvailVirtual – обсяг цієї пам'яті, ще доступної для процесу, який викликає.

Вихідний код одержання інформації підсистеми пам'яті:

```

function GetRAM: Cardinal;
var MS: TMemoryStatus;
begin
    MS.dwLength:=SizeOf(MS);
    GlobalMemoryStatus(MS);
    Result:=MS.dwTotalPhys;
end;

```

						ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			62

Розглянемо функції які були використані при створенні головного меню магістерської програми:

1. **function** createmenu: hmenu; – У випадку успішного виконання, функція повертає дескриптор створеного меню, інакше повертає 0.

2. **function** createrorupmenu: hmenu; – Як і у випадку з createmenu, після успішного виконання, функція повертає дескриптор створеного меню, інакше повертає 0.

3. **function** appendmenu(hmenu: hmenu; uflags, uidnewitem: uint; lpnewitem: pchar): bool; – Функція appendmenu додає новий елемент (пункт) до кінця зазначеного рядка меню, що розкривається меню або підменю. Цю функцію можна використовувати, щоб визначити зміст, появу і поводження пункту меню.

hmenu – ідентифікатор рядка меню, що розкривається меню або підменю, що буде змінено.

uflags – визначає прапорці, що управляють появою й поводженням нового пункту меню. Цей параметр може бути комбінація значень.

uidnewitem – визначає або ідентифікатор нового пункту меню або, якщо uflags параметр установлений в mf_rorup, дескриптор меню, що розкривається, або підменю.

lpnewitem – визначає зміст нового пункту меню.

4. **function** insertmenu(hmenu: hmenu; uposition, uflags, uidnewitem: uint; lpnewitem: pchar): bool; – Функція insertmenu вставляє новий пункт у меню, переміщаючи інші елементи меню долілиць. Параметри цієї функції ті ж, що й в appendmenu, за одним виключенням. uposition – визначає місце, у яке повинен бути вставлений новий пункт меню.

5. **function** setmenu(hwnd: hwnd; hmenu: hmenu): bool; – Функція setmenu зв'язуємо нове меню з вікном.

hwnd – ідентифікатор вікна, якому повинне бути призначене нове меню.

hmenu – ідентифікатор меню, що повинне бути призначене вікну. Якщо цей параметр нульовий, поточне меню вікна віддаляється.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

6. **function** drawmenubar(hwnd: hwnd): bool; – Функція drawmenubar перемальовує рядок меню зазначеного вікна.

hwnd – ідентифікатор вікна, чий рядок меню має потребу в зміні.

7. **function** enablemenuitem(hmenu: hmenu; uidenableitem, uenable: uint): bool; – Функція enablemenuitem включає/відключає зазначений пункт меню.

hmenu – ідентифікатор меню.

uidenableitem – визначає пункт меню, стан якого потрібно змінити.

uenable – визначає прапорці, що управляють станом пункту меню.

8. **function** checkmenuitem(hmenu: hmenu; uidenableitem, uenable: uint): bool; – Функція checkmenuitem позначає пункт меню або знімає позначку (позначка у вигляді галочки).

hmenu – ідентифікатор меню.

uidcheckitem – визначає пункт меню, чий атрибут позначки повинен бути встановлений у відповідності зі значенням параметра ucheck.

ucheck – визначає прапорці, що управляють станом пункту меню.

Головне меню програми, це рядок, що розташовується у верхній частині форми. Вона складається з пунктів, натискання на кожній з них приведе до розкриття підміню, що належить даному пункту. Це спливаюче меню в windows називається popupmenu. Popupmenu – це меню, що може "спливати" у будь-якій крапці форми.

Розглянемо створений шаблон розроблений при написанні магістерської роботи. Він складається зі списку констант і змінних необхідних для створення головного меню магістерської програми.

```
....  
const  
wndclass = ?twinapiwnd?;  
wndcaption = ?Головне меню?;  
mfile = 100;  
medit = 200;  
mcheck = 300;  
sexit = 101;  
scopy = 201;
```



```

function createmenuitem( hmenu, submenu: hmenu; cap: pchar; _uid, _wid:
uint; sep: boolean ): boolean;
var mi: menuiteminfo;
begin
with mi do
begin
cbsize := sizeof( mi );
fmask := miim_state or miim_type or miim_submenu or miim_id;
if not sep then
ftype := mft_string
else
ftype := mft_separator;
fstate := mfs_enabled;
wid := _wid;
hsubmenu := submenu;
dwitemdata := 0;
dwtypedata := cap;
cch := sizeof( cap );
end;
result := insertmenuitem( hmenu, _uid, false, mi );
end;
....

```

Вхідні параметри функції.

hmenu – меню, у яке додається новий пункт; **submenu** – пов'язане із цим пунктом підменю;

cap – заголовок нового пункту;

_uid – завжди 0 (цей параметр використовується у функції insertmenuitem);

_wid – ідентифікатор, пов'язаний з даним пунктом;

sep – ознака, чи є новий пункт чи роздільником ні.

Магістерський додаток повинний реагувати на вибір того або іншого пункту меню. Розглянемо вихідний код віконної процедури:

```

....
function windowproc( wnd: hwnd; msg: uint; wparam: wparam; lparam: lparam
): lresult; stdcall;
begin
case msg of
wm_destroy: begin
postquitmessage( 0 );

```

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67


```

cbsize := sizeof( wc );
style := cs_hredraw or cs_vredraw;
lpfnwndproc := @windowproc;
cbclsextra := 0;
cbwndextra := 0;
hinstance := hinstance;
hicon := loadicon( 0, idi_application );
hcursor := loadcursor( 0, idc_arrow );
hbrbackground := color_window;
lpzmenuname := @mainmenu;
lpzclassname := wndclass;
end;
// Регіструємо клас в системі
registerclassex( wc );
// Створюємо підменю
submenufile := createpopupmenu;
submenuedit := createpopupmenu;
submenucheck := createpopupmenu;
submenusecondlevel := createpopupmenu;
// Створюємо вікно
wnd := createwindowex( 0, wndclass, wndcaption, ws_overlappedwindow, 10,
10, 300, 100, 0, mainmenu, hinstance, nil );
// Створюємо пункти головного меню
createmenuitem(mainmenu,submenufile,?file?,0,mfile,false);
createmenuitem(mainmenu,submenuedit,?edit?,0,mfile,false);
createmenuitem(mainmenu,submenucheck,?check?,0,mfile, false);
// Підменю для пункту file
createmenuitem( submenufile, 0, ?exit?, 0, sextit, false );
// Підменю для пункту edit
createmenuitem( submenuedit, 0, ?copy?, 0, scopy, false );
createmenuitem( submenuedit, 0, ?cut?, 0, scut, false);
createmenuitem( submenuedit, 0, ??, 0, separator, true );
createmenuitem( submenuedit, 0,?paste?,0,spaste, false);
// Підменю для пункту check->nextmenu
createmenuitem( submenusecondlevel, 0, ?beep?, 0, ssecondlevel, false );
// Підменю для пункту check
createmenuitem( submenucheck, 0, ?select?, 0, sselect, false );
createmenuitem( submenucheck, submenusecondlevel, ?nextmenu?, 0, snextmenu,
false );
// Перерисовуємо меню
drawmenubar( wnd );
// Показуємо вікно

```

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

```

showwindow( wnd, sw_shownormal );
// Цикл обробки повідомлень
while getmessage( msg, 0, 0, 0 ) do
begin
    translatemessage( msg );
    dispatchmessage( msg );
end;
halt( msg.wparam );
end.
.....

```

Розглянемо вищеописаний уривок магістерської роботи. Спочатку створюється головне меню. Показчик на нього привласнюємо полю `lpzmenuname` структури `wc` (раніше воно було дорівнює `nil`). Після реєстрації класу в системі створюємо підменю. При створенні вікна, параметр `hmenu` функції `createwindowex` дорівнює `mainmenu` (`handle` створеного меню, а не `0`, як було в шаблоні). Після створення всіх пунктів, перемальовуємо меню за допомогою функції `drawmenubar`.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використати фіналіста конкурсу AES – шифр Rijndael. Він є нетрадиційним блоковим шифром, оскільки не використовує мережу Фейштеля для криптоперетворень. Алгоритм представляє кожний блок кодуємих даних у вигляді двовимірного масиву байт розміром 4×4 , 4×6 або 4×8 залежно від установленної довжини блоку. Далі на відповідних етапах перетворення відбуваються або над незалежними стовпцями, або над незалежними рядками, або взагалі над окремими байтами в таблиці.

Всі перетворення в шифрі мають строге математичне обґрунтування. Сама структура й послідовність операцій дозволяють виконувати даний алгоритм ефективно як на 16-бітних так і на 64-бітних процесорах. У структурі алгоритму закладена можливість паралельного виконання деяких операцій, що на

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

багатопроесорних робочих станціях може ще підняти швидкість шифрування в 4 рази.

Алгоритм складається з деякої кількості раундів (від 10 до 14 – це залежить від розміру блоку й довжини ключа), у яких послідовно виконуються наступні операції :

ByteSub – Таблична підстановка 8x8 біт (рисунок 4.3).

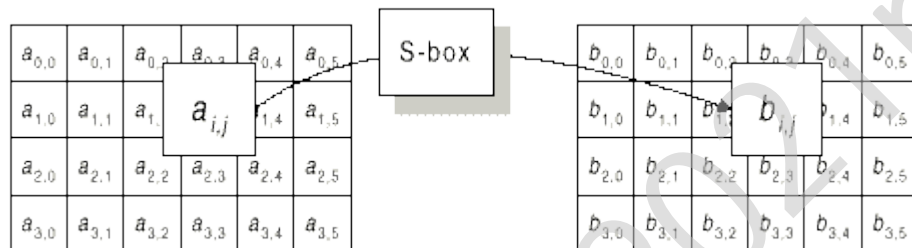


Рисунок 4.3 – Таблична підстановка 8x8 біт

ShiftRow – зрушення рядків у двовимірному масиві на різні зсуви (рисунок 4.4).

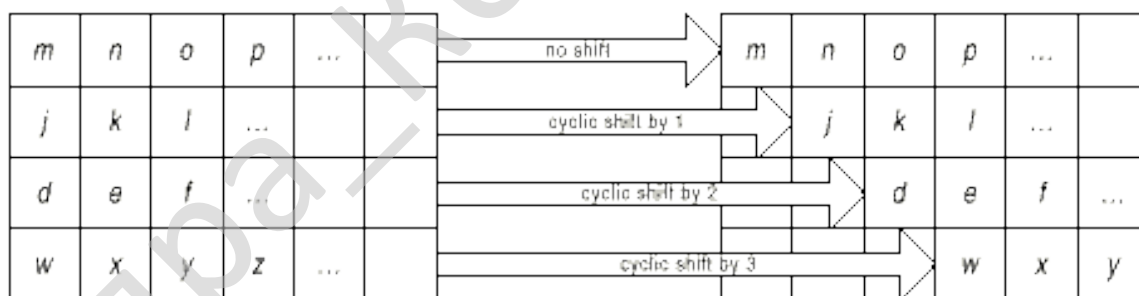


Рисунок 4.4 – Зрушення рядків у двовимірному масиві на різні зсуви

MixColumn – математичне перетворення, що перемішує дані усередині стовпця (рисунок 4.5).

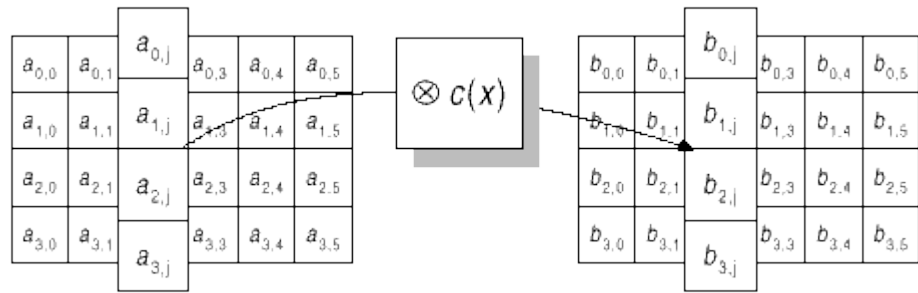


Рисунок 4.5 – Математичне перетворення, що перемішує дані усередині стовпця

AddRoundKey – додавання матеріалу ключа операцією XOR (рисунок 4.6).

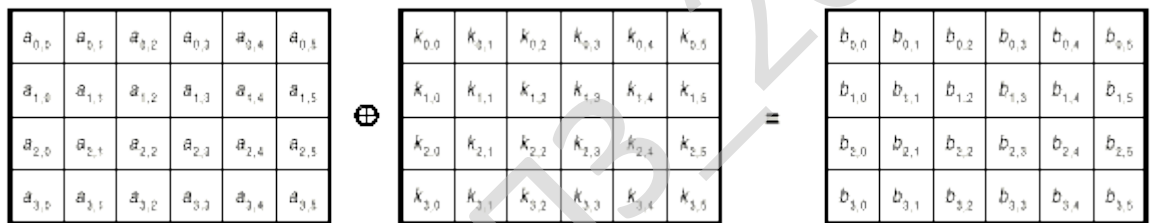


Рисунок 4.6 – Додавання матеріалу ключа операцією XOR

В останньому раунді операція перемішування стовпців відсутня, що робить всю послідовність операцій симетричною.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене програмне забезпечення при запуску в операційній системі запускається в режимі перегляду й тестування. Для повноцінної роботи програмного продукту й впровадження його в експлуатацію, а також виявлення можливих помилок і точного налаштування системи необхідно використовувати тестовий стенд Укрзалізниці. Тільки за допомогою стенда можна з'єднається з температурними датчиками, віконними датчиками, дверними датчиками, аудіо підсистемою, відеопідсистемою й динамічним таблом.

Для налаштування системи на оптимальні умови роботи необхідно виконати наступні кроки:

- встановити розроблене програмне забезпечення на персональний комп'ютер стенду;
- перевірити в операційній системі зв'язок з датчиками: температурними, віконними, дверними;
- забезпечити зв'язок з основними стендовими вузлами.
- На рисунку 5.1 зображене головне вікно програми на ньому відображається інформація про поточний стан вагона. Дані розділені на три групи: верхнє контекстне меню, основне інтерфейсне меню виводу інформації, нижнє меню стану. У свою чергу основне інтерфейсне меню виводу інформації розділено на 6 ділянок:
 - – Температурні дані;
 - – Віконні й дверні дані;
 - – Аудіо підсистема;
 - – Відео підсистема;
 - – Динамічне табло;
 - – Налаштування.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

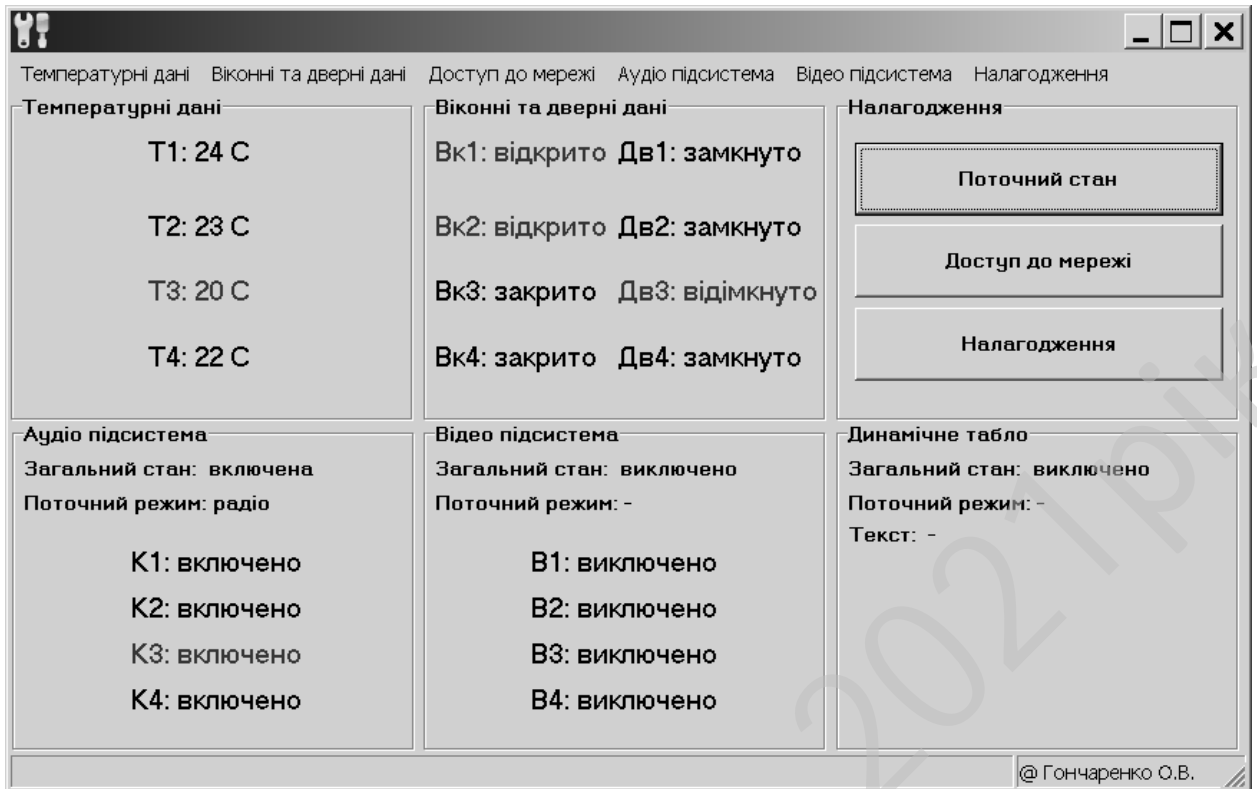


Рисунок 5.1 – Головні вікно програми

Розглянемо відображувані характеристики більш детально.

1. Температурні дані.

Виводиться інформація про поточну інформацію в купі й тамбурі (докладно розглянуто в 4 розділі).

2. Віконні й дверні дані.

Для контролю температури у вагоні (більш актуально в зимовий період) необхідно точно знати поточне положення дверей і вікон. Виводить інформацію про положення дверей і вікон (докладно розглянуто в 4 розділі).

3. Аудіо підсистема.

У нічний час доби необхідно контролювати стан аудіо підсистеми в окремо взятих купе й у цілому для забезпечення тихого періоду доби. Виводить інформацію про стан аудіо підсистеми.

4. Відео підсистема.

Контроль відеодисплеїв у кожному купе вагона.

5. Динамічне табло.

Для завчасного оголошення наступної станції й керування посадкою пасажирів необхідно контролювати поточні дані, які висвітлюються динамічним таблом. Виводить інформацію про стан динамічного табло.

6. Налаштування.

Для налаштування з'єднання з мережею й параметрів виводу на екран даних необхідно нажати на відповідні кнопки.

В умовах магістерської роботи було не повністю проведено тестування розробленої програми так як на це потрібно багато годин, спеціалізоване устаткування та не є головною метою роботи. Мета роботи, а саме – розробка структури програми її модулів, та роботи системи в цілому було виконано в повному обсязі. Ця інформація та авторські дані виводяться у вікні авторського права програми та зображені на рисунку 5.2.

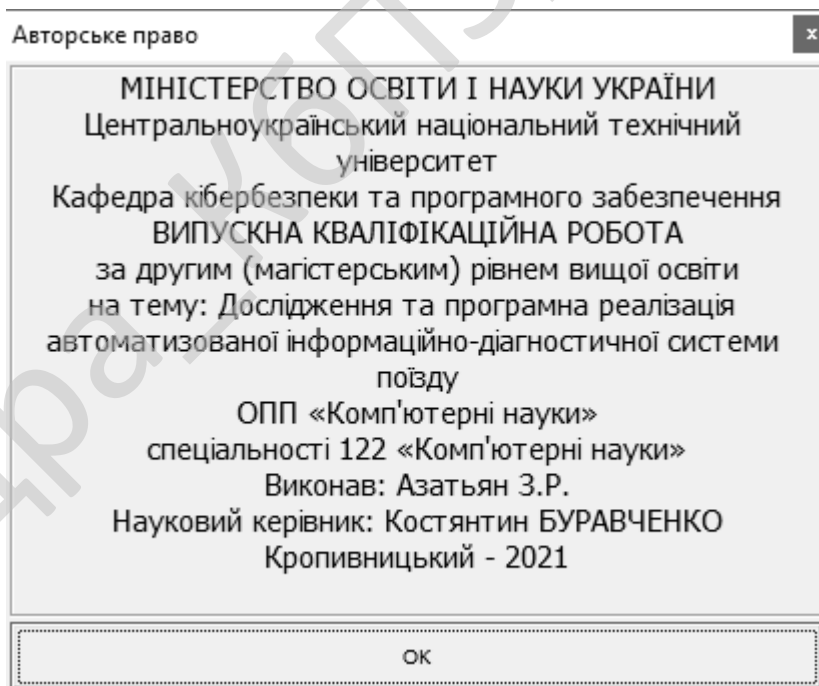


Рисунок 5.2 – Вікно авторського права

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для автоматизованої інформаційно-діагностичної системи поїзду.

Метою розробки є дослідження та програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду.

Об'єктом дослідження є процес реалізації автоматизованої інформаційно-діагностичної системи поїзду.

Предметом дослідження є методи реалізації автоматизованої інформаційно-діагностичної системи поїзду.

Методи дослідження базуються на методах теорії автоматизованого управління, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод реалізації автоматизованої інформаційно-діагностичної системи поїзду.
- Розроблено вітчизняний продукт реалізації автоматизованої інформаційно-діагностичної системи поїзду, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

78

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	100000 (2 ост. цифри № зал*10 ⁵)
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м. п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2}, \quad (7.6)$$

$$\Phi_{op}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2016, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	0,3	0,1
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,1	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,1	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	0,3	
Всього		0,8	

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	0,4	0,1
	Підтримка постійних клієнтів	0,1	
	Оформлення договорів, ведення тендерів	0,1	
	Контроль взаєморозрахунків з постачальниками	0,2	
Всього		0,8	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	0,1	0,1
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,1	
	Розміщення графіки і контенту на Інтернет сторінках	0,1	
Всього		0,8	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	0,2	0,1
	Верстка друкованих видань	0,2	
	Додрукова підготовка макетів	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		0,8	

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

84

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	7925	23775
Продакт-менеджер	0,1	6000	1800
Інженер-програміст	3,8	8000	91200
Інженер - електронщик	1,2	6000	21600
Інженер-системотехнік	0,1	6000	1800
Адміністратор мережі	0,1	6000	1800
Системний програміст	0,1	6000	1800
Дизайнер WEB	0,1	6000	1800
Інженер-верстальник	0,1	6000	1800
Бухгалтер-економіст	0,1	6000	1800
Всього за період розробки	$R_{cn} = 6,7$	-	$\Phi_{роб} = 149175$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{149175}{6,7 \cdot 60} = 301 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

$$B_{y\partial} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

S_y – питома площа на одне робоче місце, m^2 ;

C_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 29 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 29000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 8 \cdot 8 \cdot 29000 = 1858000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 185800 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{nb} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де: C_m – ціна меблів для одного робочого місця, грн.

$$I_{nb} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом Інтернет магазину Компбест за 18.10.21 – джерело <https://compbest.com.ua>.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		10947
Системний блок		Fujitsu P720 Tower
Процесор	Intel Core i3-4130 (2 (4) ядра по 3.40 GHz); Cache Memory 3	-
Системна плата	Fujitsu D3221-A1 Intel Haswell c TDP до 95 Вт	-
Відеокарта	AMD Radeon RX 550 4GB GDDR5 Re Dragon PowerColor (AXRX 550 4GBD5 DH)	-
Жорсткий диск	HDD Seagate Barracuda 750 Gb 7200 32Mb SATAII ST3750528AS (ST3750528AS)	-
Оперативна пам'ять	DIMM 4096Mb DDR3 PC3-10600 CL9 Transcend JetRam, non-Reg., no-ECC , CL 9 (2 модулі)	-
DVD-привод	DVD -RW/+RW , LG SATA SuperMulti Bulk 22x, SecurDisc, black	-
Корпус	GRESSO GE-7525, 500W (120mm big fan), 2xIDE, full-ATX,БЖ 2xSATA, 1xFDD, Air Duct, 2xUSB 2.0, Mic+Audio, silver/black	-

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-122.21.0001.00.00.ПЗ

Арк.

87

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Персональні комп'ютери	15	10947	16420,5	180625,5
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	199177

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1858000	-	-
2. Передавальні пристрої	185800	-	-
Всього по групі	2043800	5	102190
Група 4			
3. Обчислювальна техніка	199177	-	-
Всього по групі	199177	50	99588,5
4. Нематеріальні активи	100000	10	10000
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	143000	20	28600
7. Господарський інвентар	28000	25	7000
Всього по групі	180031	-	37857,75
Разом	$K_p = 2523008$		$A_p = 249636,25$

Примітка: вартість автомобіля Sens (Standard+) взята по даним з автосалону «Кіровоград-Авто», джерело <http://kirovograd-avto.ukravto.ua/catalog/tm-9/model-80/description>, складає 143000 грн.

Згідно виданих викладачем норм приймаємо одну пачку паперу на три місяці розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 103$ грн., визначаємо вартість паперу за період розробки $N_m = 3$ міс:

$$Z_{M1} = C_n \cdot N_m. \quad (7.16)$$

$$Z_{M1} = 103 \cdot 3 = 309 \text{ грн.}$$

Згідно виданих викладачем норм до вартості запам'ятовуваних пристроїв входить вартість CD дисків в кількості, що дорівнює кількості екземплярів програм та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 2 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 2 грн./шт.

$$Z_{M2} = 12 \cdot 100 + 15 = 1215 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де: C_z – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (309 + 1215 + 1702) / 100 = 32 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 629 \cdot 15 \cdot 0,01 = 94 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 100$ прим.):

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 249636 \cdot 3 / (100 \cdot 12) = 624 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	629
2. Додаткова зарплата виконавців	Z_d	63
3. Відрахування на соціальні потреби	C_{oc}	156
4. Загальногосподарські витрати	G_{ocn}	94
5. Витрати на матеріали	Z_M	32
6. Освоєння нових операційних систем, мов програмування	O_n	94
7. Амортизація основних фондів	A_m	624
8. Повна собівартість програмного забезпечення	C_n	1692
9. Плановий прибуток	P_p	846
10. Ціна підприємства $C_n = C_n + P_p$	C_n	2538
11. Податок на додану вартість $ПДВ = 0.01 \cdot N_{об} \cdot C_n$	$ПДВ$	507,6
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	3045,6

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	100
2. Повна собівартість розробленої програми	Грн.	1692
3. Ціна розробленої програми	Грн.	2538
4. Плановий прибуток від реалізації розробленої програми	Грн.	846
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	2523008
7. Загальний прибуток від реалізації програмної продукції	Грн.	89600
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	27170
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,4
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	3046
11. Величина економічного ефекту у користувача програмної продукції	Грн.	44593
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,1

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\delta} - I_n) - E_n(K_n - K_{\delta}), \quad (7.27)$$

де: $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (153720 - 108366) - 0,25 \cdot 3046 = 44593 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{3046}{153720 - 108366} = 0,1 \text{ року.}$$

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини в процесі трудової діяльності.

Охорона праці є складовою частиною безпеки життєдіяльності [3,4].

Законом України “Про охорону праці” [2] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98, та «Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

Умови праці програміста включають наступні фактори:

- параметри монітора (випроміювання);
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення впливу комп'ютера на організм програміста визначемо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Програміст працює з електронно-обчислювальною машиною (ЕОМ) та іншим обладнанням, яке є джерелом небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. Так як програміст постійно перебуває в приміщенні, тому для комфортних умов праці в цьому приміщенні необхідно створити належний мікроклімат.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

- електромагнітні (у т.ч. високочастотні) випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- невідповідність ергономічних показників робочого місця діючим вимогам;

- шуми;
- статичні навантаження на кістково-м'язовий апарат;
- недостатня, або надмірна освітленість робочого місця;
- монотонність праці.

Результати досліджень показали, що найбільшою мірою негативний фізіологічний вплив на операторів ПК пов'язано з дискомфорними зоровими умовами через неправильно спроектованого освітлення: пряма і відбита від екранів бляклість, несприятливий розподіл яскравості в полі зору, невірна орієнтація робочого місця щодо світлоприймачів.

Розташовувати обладнане дисплеєм робоче місце необхідно таким чином, щоб в поле зору оператора не потрапляли вікна або освітлювальні прилади. Вони не повинні перебувати і безпосередньо за спиною оператора. Слід домагатися зменшення відображень на екрані від різних джерел штучного і денного світла. Коли штучне світло змішується з природним, рекомендується використовувати лампи, по спектрального складу найбільш близькі до сонячного світла. Співвідношення яскравості екрана і безпосередніх найближчого оточення не повинно перевищувати 3: 1.

Оптимальні значення температури повітря в приміщенні повинні бути 19-23 С. Швидкість руху повітря не більше 0.1 м / с. Рекомендована відносна вологість повітря 55%.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер HP 1100, електродвигуни вентиляторів ЕОМ.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: наочне знайомство персоналу з шляхами для евакуації людей із приміщення відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання.

8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 2,6 м, довжина – 2,6 м, висота – 3 м.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою:

$$F=ESKZ/n,$$

де:

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

$F=14548$ Лм.

Будемо використовувати лампи TL-F 12 36W LED 6000K IP65, світловий потік яких $F_l = 3000$ Лм.

Число ламп визначається за формулою:

$$N=F/F_l$$

де:

F – світловий потік,

F_l – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо індекс приміщення:

$$N= 14548/ 3000=4,8 \text{ шт.}$$

Приймаємо необхідну кількість ламп 5 шт.

8.6 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для автоматизованої інформаційно-діагностичної системи поїзду.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів реалізації автоматизованої інформаційно-діагностичної системи поїзду.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем реалізації автоматизованої інформаційно-діагностичної системи поїзду.
- Досліджена система реалізації автоматизованої інформаційно-діагностичної системи поїзду.
- На основі отриманих результатів досліджень створена програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання реалізації автоматизованої інформаційно-діагностичної системи поїзду.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

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.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

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. 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 XP/Vista/7/8/10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм AES.

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.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 44593 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,1 роки.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Азатьян З.Р. Дослідження та програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду // Збірник праць молодих науковців ЦНТУ. – Вип. 12. – Кропивницький: ЦНТУ, 2022.
2. Гапеев С.Н. Пассажирские вагоны, электро – дизель – поезд // Железнодорожный транспорт. – 2001. – №3.
3. Комплексная программа реорганизации и развития отечественного локомотиво – и вагоностроения, организации ремонта и эксплуатации пассажирского и грузового подвижного состава на период 2001 – 2010 г. – К., 2001.
4. Назаров О.Н. Типаж и технические требования к перспективному пассажирскому подвижному составу // Железнодорожный транспорт. – 2003. – №2.
5. 4255/ЦВ МПС. Вагоны пассажирские цельнометаллические. Руководство по деповскому ремонту (ДР). – М.: Транспорт, 1986.
6. ЦВ – ЦТВР/4321 МПС. Вагоны пассажирские цельнометаллические. Руководство по капитальному ремонту (КР – 1). – М.: Транспорт, 1985.
7. Ивашов В.А., Орлов М.В. Вагонное хозяйство: Учебник. – Екатеринбург: Издательство УрГАПС, 1998.
8. Шишков А.Д., Дмитриев В.А., Гусаков В.И. Организация, планирование и управление производством по ремонту подвижного состава / Под ред. А.Д. Шишкова. – М.:Транспорт, 1997.
9. Гридюшко В.И., Криворучко Н.З., Бугаев В.П. Вагонное хозяйство. 2 – е. изд. – М.:Транспорт, 1988.
10. Сборник нормативной документации по трудозатратам на все виды ремонта и текущего содержания пассажирских вагонов. МПС ДПС.: – М., 2001.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

11. Романова Т.А., Лавров А.П. Долговечная противокоррозионная защита пассажирских вагонов // Железнодорожный транспорт. – 2003. – №5.
12. ТП – ЦЛПВ – 33/4. Типовой технологический процесс окрашивания пассажирских вагонов.
13. Регламент технической оснащённости производственного подразделения пассажирского вагонного депо № ЦЛПВР – 30. МПС. – М., 1999.
14. Инструкция по осмотру, освидетельствованию, ремонту и формированию вагонных колёсных пар. ЦВ /3429 МПС. – М.: Транспорт, 1977.
15. Инструктивные указания по эксплуатации и ремонту вагонных букс с роликовыми подшипниками. №3 – ЦВРК МПС. – М., 2001.
16. Вагоны пассажирские. Поводок тележки. Руководство по эксплуатации НТ – 0200 РЕ. МПС ДПС. – М., 2000.
17. Вагоны пассажирские. Запорный клапан системы водоснабжения пассажирского вагона постройки Германии. Руководство по эксплуатации НТ – 0497 РЕ. МПС ДПС. – М., 1999.
18. Туровец А.С., Романова Т.А., Караваев И.И. Профилактический уход за пассажирским подвижным составом // Железнодорожный транспорт. – 2003. – №6.
19. Лисевич Т.В., Александров Е.В. Машины вагоноремонтного производства (часть 1): Учебное пособие. – Самара: СамИИТ, 2002.
20. Железнодорожные пассажирские вагонные депо. Нормы проектирования ВСН 02-91/МПС. ГипротрансТЕИ МПС. – М., 1991.
21. Романенко В.М., Коркин В.М., Александров Е.В. Повышение качества транспортного обслуживания пассажиров // Вестник МАНЕБ. Том 9. Вып. 5. – С.-Петербург, 2004.
22. 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.

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

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

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

25. 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.

26. 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.

27. 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.

28. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смирнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.

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

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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 1. – P. 21-27.

36. 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

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

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.

37. 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.

38. 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.

39. 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.

40. 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.

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

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

42. 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.

43. 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.

44. 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.

45. 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.

46. 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.

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

48. Коваленко А.С. Разработка метода усовершенствования технического обслуживания интегрированной информационной системы / А.С. Коваленко,

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		114

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.

49. 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.

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

51. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

52. Про охорону праці: Закон України від 14.10.1992 р. № 2694-ХІІ. Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

53. Щодо організації роботи по вопросам охорони праці та безпеки життєдіяльності у дошкільних навчальних закладах: Лист МОН України від 23.09.2014р. № 1 / 9-482. Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v-482729-14#Text>

54. Вакцинирование, как составляющая безопасности жизнедеятельности во время пандемии Sars-Cov-2 (УДК 331.45, 614.8) / О.В. Орышака, К.Н. Марченко, А.К. Марченко // Журнал “Соціальна фармація в охороні здоров’я” / Національний фармацевтичний університет (НфаУ) – Харків, 2021. – Том 7 № 2 (2021), – С. 27-33. Режим доступу до ресурсу: <https://doi.org/10.24959/sphhcj.21.226>

					ВКРМ-122.21.0001.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		115

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	Економічні вимоги.....	5
8	Вимоги щодо охорони праці.....	5
9	Перелік документів, що розробляються.....	6
10	Етапи розробки.....	6
11	Порядок контролю та приймання.....	6

					ВКРМ-122.21.0001.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Азатьян З.Р.				Літ.	Аркуш	Аркушів
Перевірів	Буравченко К.О.			М			
Н. Контр.	Гермак В.С.				ЦНТУ КН-20М-1,4		
Затв.	Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію автоматизованої інформаційно-діагностичної системи поїзду.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 39-13 від 02.08.2021 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація автоматизованої інформаційно-діагностичної системи поїзду.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) 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;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-122.21.0001.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію автоматизованої інформаційно-діагностичної системи поїзду;
- 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.

					ВКРМ-122.21.0001.00.00.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 XP/Vista/7/8/10 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 XP/Vista/7/8/10.

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.

					ВКРМ-122.21.0001.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2021 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута шкідливі і небезпечні фактори при роботі з комп'ютером.

					ВКРМ-122.21.0001.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 115 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2021 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 21.12.2021 р.

					ВКРМ-122.21.0001.00.00.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
другим (магістерським) рівнем вищої освіти

_____ Буравченко К.О.

*Дослідження та програмна реалізація
автоматизованої інформаційно-діагностичної системи поїзду*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 45

Літера: РП

Кропивницький – 2021 року

Файл проекту програми залізниця

```
program ZALIZNICA;
{Copyright © 2021 CNTU AZATIAN Z.R.}

uses
  {Використовуваємо бібліотеки, та свої допоміжні модулі}
  Forms,
  SysUtils,
  Unit1 in 'Unit1.pas' {Synaser},
  Unit2 in 'Unit2.pas' {About};
  Unit3 in 'Unit3.pas' {Unit3};
  Unit4 in 'Unit4.pas' {LybADD};
  Unit5 in 'Unit5.pas' {indat};

var
  i:integer;
  {$R *.res} {ресурси програми}

Begin
  {виведення початкового вікна програми}
  try
    CM_SPLASH:=TCM_SPLASH.Create(Application);
    CM_SPLASH.Show;
    CM_SPLASH.Update;
    Application.Initialize;
    Application.Title:= 'ZALIZNICA';

    Application.CreateForm(TSynaser, Synaser);
    Application.CreateForm(TUnit3, Unit3);
    Application.CreateForm(TLybADD, LybADD);
    Application.CreateForm(Tindat, indat);
    Application.CreateForm(TAbout, About);
  finally
    for i:=1 to 100 do
      begin
        CM_SPLASH.glProgress1.Percent:=i;
        CM_SPLASH.Update; //Робота початкового вікна
        Sleep(20);
      end;
    CM_SPLASH.Hide;
    CM_SPLASH.free;
  end;
  Application.Run; //Запуск програми
end.
```

Файл завантаження бібліотек

```
unit LybADD;
{Copyright © 2021 CNTU AZATIAN Z.R.}

interface

uses
  Libc, dynlibs;
type
HMODULE = Longint;
function LoadLibrary(ModuleName: PChar): HMODULE;
function FreeLibrary(Module: HMODULE): LongBool;
function GetProcAddress(Module: HMODULE; Proc: PChar): Pointer;
function GetModuleFileName(Module: HMODULE; Buffer: PChar; BufLen: Integer):
Integer;
procedure Sleep(milliseconds: Cardinal);

implementation

function LoadLibrary(ModuleName: PChar): HMODULE;
begin
  Result:= HMODULE(dynlibs.LoadLibrary(ModuleName));
end;

function FreeLibrary(Module: HMODULE): LongBool;
begin
  Result:=dynlibs.UnloadLibrary(pointer(Module));
end;

function GetProcAddress(Module: HMODULE; Proc: PChar): Pointer;
begin
  Result:=dynlibs.GetProcedureAddress(pointer(Module), Proc);
end;

function GetModuleFileName(Module: HMODULE; Buffer: PChar; BufLen: Integer):
Integer;
begin
  Result:=0;
end;

procedure Sleep(milliseconds: Cardinal);
begin
  usleep(milliseconds * 1000);
end;

end.
```

Файл взаємодії

```

unit indat;
{Copyright © 2021 CNTU AZATIAN Z.R.}

interface

uses
  Libc, Windows, SysUtils, Classes;

function TimeZoneBias: integer;
function TimeZone: string;
function Rfc822DateTime(t: TDateTime): string;
function CDateTime(t: TDateTime): string;
function SimpleDateTime(t: TDateTime): string;
function AnsiCDateTime(t: TDateTime): string;
function GetMonthNumber(Value: AnsiString): integer;
function GetTimeFromStr(Value: string): TDateTime;
function GetDateMDYFromStr(Value: string): TDateTime;
function DecodeRfcDateTime(Value: string): TDateTime;
function GetUTTime: TDateTime;
function SetUTTime(Newdt: TDateTime): Boolean;
function GetTick: ULong;
function TickDelta(TickOld, TickNew: ULong): ULong;
function CodeInt(Value: Word): Ansistring;
function DecodeInt(const Value: Ansistring; Index: Integer): Word;
function CodeLongInt(Value: LongInt): Ansistring;
function DecodeLongInt(const Value: Ansistring): LongInt;
function IsIP(const Value: string): Boolean;
function IsIP6(const Value: string): Boolean;
function IPToID(Host: string): string;
function DumpStr(const Buffer: Ansistring): string;
function DumpExStr(const Buffer: Ansistring): string;
procedure Dump(const Buffer: AnsiString; DumpFile: string);
procedure DumpEx(const Buffer: AnsiString; DumpFile: string);
function TrimSPLeft(const S: string): string;
function TrimSPRight(const S: string): string;
function TrimSP(const S: string): string;
function SeparateLeft(const Value, Delimiter: string): string;
function SeparateRight(const Value, Delimiter: string): string;
function GetParameter(const Value, Parameter: string): string;
procedure ParseParametersEx(Value, Delimiter: string; const Parameters: TStrings);
procedure ParseParameters(Value: string; const Parameters: TStrings);
function IndexByBegin(Value: string; const List: TStrings): integer;
function GetEmailAddr(const Value: string): string;
function GetEmailDesc(Value: string): string;
function StrToHex(const Value: Ansistring): string;
function IntToBin(Value: Integer; Digits: Byte): string;
function BinToInt(const Value: string): Integer;
function ParseURL(URL: string; var Prot, User, Pass, Host, Port, Path,
  Para: string): string;
function ReplaceString(Value, Search, Replace: string): string;
function RPosEx(const Sub, Value: string; From: integer): Integer;
function RPos(const Sub, Value: String): Integer;
function FetchBin(var Value: string; const Delimiter: string): string;
function Fetch(var Value: string; const Delimiter: string): string;
procedure StringsTrim(const value: TStrings);
function PosFrom(const SubStr, Value: String; From: integer): integer;
function GetBetween(const PairBegin, PairEnd, Value: string): string;
function CountOfChar(const Value: string; Chr: char): integer;
function UnquoteStr(Value: string; Quote: Char): string;
procedure HeadersToList(const Value: TStrings);
procedure ListToHeaders(const Value: TStrings);
function SwapBytes(Value: integer): integer;
function ReadStrFromStream(const Stream: TStream; len: integer): AnsiString;
procedure WriteStrToStream(const Stream: TStream; Value: AnsiString);

var

```

```

implementation
{=====--//--//--//=====}
function TimeZoneBias: integer;
var
t: TTime_T;
UT: TUnixTime;
begin
var
zoneinfo: TTimeZoneInformation;
bias: Integer;
begin
case GetTimeZoneInformation(Zoneinfo) of
2:
bias:= zoneinfo.Bias + zoneinfo.DaylightBias;
1:
bias:= zoneinfo.Bias + zoneinfo.StandardBias;
else
bias:= zoneinfo.Bias;
end;
Result:= bias * (-1);
end;
{=====--//--//--//=====}
function TimeZone: string;
var
bias: Integer;
h, m: Integer;
begin
bias:= TimeZoneBias;
if bias >= 0 then
Result:= '+'
else
Result:= '-';
bias:= Abs(bias);
h:= bias div 60;
m:= bias mod 60;
Result:= Result + Format('%.2d%.2d', [h, m]);
end;
{=====--//--//--//=====}
function Rfc822DateTime(t: TDateTime): string;
var
wYear, wMonth, wDay: word;
begin
DecodeDate(t, wYear, wMonth, wDay);
Result:= Format('%s, %d %s %s %s', [MyDayNames[DayOfWeek(t)], wDay,
MyMonthNames[1, wMonth], FormatDateTime('yyyy hh:nn:ss', t), TimeZone]);
end;
{=====--//--//--//=====}
function CDateTime(t: TDateTime): string;
var
wYear, wMonth, wDay: word;
begin
DecodeDate(t, wYear, wMonth, wDay);
Result:= Format('%s %2d %s', [MyMonthNames[1, wMonth], wDay,
FormatDateTime('hh:nn:ss', t)]);
end;
{=====--//--//--//=====}
function SimpleDateTime(t: TDateTime): string;
begin
Result:= FormatDateTime('yymmdd hhnss', t);
end;
{=====--//--//--//=====}
function AnsiCDateTime(t: TDateTime): string;
var
wYear, wMonth, wDay: word;
begin
DecodeDate(t, wYear, wMonth, wDay);
Result:= Format('%s %s %d %s', [MyDayNames[DayOfWeek(t)], MyMonthNames[1, wMonth],
wDay, FormatDateTime('hh:nn:ss yyyy ', t)]);
end;

```

```

{=====//--//--//=====}
function DecodeTimeZone(Value: string; var Zone: integer): Boolean;
var
x: integer;
zh, zm: integer;
s: string;
begin
Result:= false;
s:= Value;
if (Pos('+', s) = 1) or (Pos('-',s) = 1) then
begin
if s = '-0000' then
Zone:= TimeZoneBias
else
if Length(s) > 4 then
begin
zh:= StrToIntDef(s[2] + s[3], 0);
zm:= StrToIntDef(s[4] + s[5], 0);
zone:= zh * 60 + zm;
if s[1] = '-' then
zone:= zone * (-1);
end;
Result:= True;
end
else
begin
x:= 32767;
zone:= x * 60;
Result:= True;
end;
end;
end;

function GetTimeFromStr(Value: string): TDateTime;
var
x: integer;
begin
x:= rpos(':', Value);
if (x > 0) and ((Length(Value) - x) > 2) then
Value:= Copy(Value, 1, x + 2);
Value:= ReplaceString(Value, ':', TimeSeparator);
Result:= -1;
try
Result:= StrToTime(Value);
except
on Exception do ;
end;
end;
{=====//--//--//=====}
function GetDateMDYFromStr(Value: string): TDateTime;
var
wYear, wMonth, wDay: word;
s: string;
begin
Result:= 0;
s:= Fetch(Value, '-');
wMonth:= StrToIntDef(s, 12);
s:= Fetch(Value, '-');
wDay:= StrToIntDef(s, 30);
wYear:= StrToIntDef(Value, 1899);
if wYear < 1000 then
if (wYear > 99) then
wYear:= wYear + 1900
else
if wYear > 50 then
wYear:= wYear + 1900
else
wYear:= wYear + 2000;
try

```

```

Result:= EncodeDate(wYear, wMonth, wDay);
except
on Exception do ;
end;
end;
{=====//--//--//=====}
function DecodeRfcDateTime(Value: string): TDateTime;
var
day, month, year: Word;
zone: integer;
x, y: integer;
s: string;
t: TDateTime;
begin
Result:= 0;
if Value = '' then
Exit;
day:= 0;
month:= 0;
year:= 0;
zone:= 0;
Value:= ReplaceString(Value, ' -', ' #');
Value:= ReplaceString(Value, '-', ' ');
Value:= ReplaceString(Value, ' #', ' -');
while Value <> '' do
begin
s:= Fetch(Value, ' ');
s:= uppercase(s);
if DecodetimeZone(s, x) then
begin
zone:= x;
continue;
end;
x:= StrToIntDef(s, 0);
// day or year
if x > 0 then
if (x < 32) and (day = 0) then
begin
day:= x;
continue;
end
else
begin
year:= x;
if year < 32 then
year:= year + 2000;
if year < 1000 then
year:= year + 1900;
continue;
end;
// time
if rpos(':', s) > Pos(':', s) then
begin
t:= GetTimeFromStr(s);
if t <> -1 then
Result:= t;
continue;
end;
if s = 'DST' then
begin
zone:= zone + 60;
continue;
end;
// month
y:= GetMonthNumber(s);
if y > 0 then
month:= y;
end;
if year = 0 then

```

```

year:= 1980;
if (month < 1) or (month > 12) then
month:= 1;
if (day < 1) or (day > 31) then
day:= 1;
Result:= Result + Encodedate(year, month, day);
zone:= zone - TimeZoneBias;
x:= zone div 1440;
Result:= Result - x;
zone:= zone mod 1440;
t:= EncodeTime(Abs(zone) div 60, Abs(zone) mod 60, 0, 0);
if zone < 0 then
t:= 0 - t;
Result:= Result - t;
end;
{=====--//--//--//=====}
function GetUTTime: TDateTime;
var
st: TSystemTime;
begin
GetSystemTime(st);
result:= SystemTimeToDateTime(st);
var
st: SysUtils.TSystemTime;
stw: Windows.TSystemTime;
begin
GetSystemTime(stw);
st.Year:= stw.wYear;
st.Month:= stw.wMonth;
st.Day:= stw.wDay;
st.Hour:= stw.wHour;
st.Minute:= stw.wMinute;
st.Second:= stw.wSecond;
st.Millisecond:= stw.wMilliseconds;
result:= SystemTimeToDateTime(st);
var
TV: TTimeVal;
TZ: Ttimezone;
begin
TZ.tz_minuteswest:= 0;
TZ.tz_dsttime:= 0;
gettimeofday(TV, TZ);
Result:= UnixDateDelta + (TV.tv_sec + TV.tv_usec / 1000000) / 86400;
end;
{=====--//--//--//=====}
function SetUTTime(Newdt: TDateTime): Boolean;
var
st: TSystemTime;
begin
DateTimeToSystemTime(newdt, st);
Result:= SetSystemTime(st);
var
st: SysUtils.TSystemTime;
stw: Windows.TSystemTime;
begin
DateTimeToSystemTime(newdt, st);
stw.wYear:= st.Year;
stw.wMonth:= st.Month;
stw.wDay:= st.Day;
stw.wHour:= st.Hour;
stw.wMinute:= st.Minute;
stw.wSecond:= st.Second;
stw.wMilliseconds:= st.Millisecond;
Result:= SetSystemTime(stw);
var
TV: TTimeVal;
d: double;
TZ: Ttimezone;
begin

```

```

Result:= false;
TZ.tz_minuteswest:= 0;
TZ.tz_dsttime:= 0;
gettimeofday(TV, TZ);
d:= (newdt - UnixDateDelta) * 86400;
TV.tv_sec:= trunc(d);
TV.tv_usec:= trunc(frac(d) * 1000000);
Result:= settimeofday(TV, TZ) <> -1;
end;
{=====--//--//--//=====}
function GetTick: ULong;
var
Stamp: TTimeStamp;
begin
Stamp:= DateTimeToTimeStamp(Now);
Result:= Stamp.Time;
end;

function GetTick: ULong;
begin
Result:= Windows.GetTickCount;
end;
{=====--//--//--//=====}
function TickDelta(TickOld, TickNew: ULong): ULong;
begin
Result:= 0;
if TickOld <> TickNew then
begin
if TickNew < TickOld then
begin
TickNew:= TickNew + ULong(MaxInt) + 1;
TickOld:= TickOld + ULong(MaxInt) + 1;
end;
Result:= TickNew - TickOld;
if TickNew < TickOld then
if Result > 0 then
Result:= 0 - Result;
end;
end;
end;
{=====--//--//--//=====}
function CodeInt(Value: Word): Ansistring;
begin
setlength(result, 2);
result[1]:= AnsiChar(Value div 256);
result[2]:= AnsiChar(Value mod 256);
Result:= AnsiChar(Value div 256) + AnsiChar(Value mod 256)
end;
{=====--//--//--//=====}
function DecodeInt(const Value: Ansistring; Index: Integer): Word;
var
x, y: Byte;
begin
if Length(Value) > Index then
x:= Ord(Value[Index])
else
x:= 0;
if Length(Value) >= (Index + 1) then
y:= Ord(Value[Index + 1])
else
y:= 0;
Result:= x * 256 + y;
end;
{=====--//--//--//=====}
function CodeLongInt(Value: Longint): Ansistring;
var
x, y: word;
begin
// this is fix for negative numbers on systems where longint = integer
x:= (Value shr 16) and integer($ffff);

```

```

y:= Value and integer($ffff);
setlength(result, 4);
result[1]:= AnsiChar(x div 256);
result[2]:= AnsiChar(x mod 256);
result[3]:= AnsiChar(y div 256);
result[4]:= AnsiChar(y mod 256);
end;
{=====--//--//--//=====}
function DecodeLongInt(const Value: Ansistring; Index: Integer): LongInt;
var
x, y: Byte;
xl, yl: Byte;
begin
if Length(Value) > Index then
x:= Ord(Value[Index])
else
x:= 0;
if Length(Value) >= (Index + 1) then
y:= Ord(Value[Index + 1])
else
y:= 0;
if Length(Value) >= (Index + 2) then
xl:= Ord(Value[Index + 2])
else
xl:= 0;
if Length(Value) >= (Index + 3) then
yl:= Ord(Value[Index + 3])
else
yl:= 0;
Result:= ((x * 256 + y) * 65536) + (xl * 256 + yl);
end;
{=====--//--//--//=====}
function IsIP(const Value: string): Boolean;
var
TempIP: string;
function ByteIsOk(const Value: string): Boolean;
var
x, n: integer;
begin
x:= StrToIntDef(Value, -1);
Result:= (x >= 0) and (x < 256);
// X may be in correct range, but value still may not be correct value!
// i.e. "$80"
if Result then
for n:= 1 to length(Value) do
if not (Value[n] in ['0'..'9']) then
begin
Result:= False;
Break;
end;
end;
begin
TempIP:= Value;
Result:= False;
if not ByteIsOk(Fetch(TempIP, '.')) then
Exit;
if not ByteIsOk(Fetch(TempIP, '.')) then
Exit;
if not ByteIsOk(Fetch(TempIP, '.')) then
Exit;
if ByteIsOk(TempIP) then
Result:= True;
end;
{=====--//--//--//=====}
function IsIP6(const Value: string): Boolean;
var
TempIP: string;
s,t: string;
x: integer;

```

```

partcount: integer;
zerocount: integer;
First: Boolean;
begin
TempIP:= Value;
Result:= False;
partcount:= 0;
zerocount:= 0;
First:= True;
while tempIP <> '' do
begin
s:= fetch(TempIP, ':');
if not(First) and (s = '') then
Inc(zerocount);
First:= False;
if zerocount > 1 then
break;
Inc(partCount);
if s = '' then
Continue;
if partCount > 8 then
break;
if tempIP = '' then
begin
t:= SeparateRight(s, '%');
s:= SeparateLeft(s, '%');
x:= StrToIntDef('$' + t, -1);
if (x < 0) or (x > $ffff) then
break;
end;
x:= StrToIntDef('$' + s, -1);
if (x < 0) or (x > $ffff) then
break;
if tempIP = '' then
Result:= True;
end;
end;
{=====--//--//--//=====}
function IPToID(Host: string): string;
var
s, t: string;
i, x: Integer;
begin
Result:= '';
for x:= 1 to 3 do
begin
t:= '';
s:= Fetch(Host, '.');
i:= StrToIntDef(s, 0);
Result:= Result + Chr(i);
end;
i:= StrToIntDef(Host, 0);
Result:= Result + Chr(i);
end;
{=====--//--//--//=====}
function DumpStr(const Buffer: Ansistring): string;
var
n: Integer;
begin
Result:= '';
for n:= 1 to Length(Buffer) do
Result:= Result + ' +#$' + IntToHex(Ord(Buffer[n]), 2);
end;
{=====--//--//--//=====}
function DumpExStr(const Buffer: Ansistring): string;
var
n: Integer;
x: Byte;
begin

```

```

Result:= '';
for n:= 1 to Length(Buffer) do
begin
x:= Ord(Buffer[n]);
if x in [65..90, 97..122] then
Result:= Result + ' +'' + char(x) + ''
else
Result:= Result + ' +#$' + IntToHex(Ord(Buffer[n]), 2);
end;
end;
{=====--//--//--//=====}
procedure Dump(const Buffer: AnsiString; DumpFile: string);
var
f: Text;
begin
AssignFile(f, DumpFile);
if FileExists(DumpFile) then
DeleteFile(DumpFile);
Rewrite(f);
try
Writeln(f, DumpStr(Buffer));
finally
CloseFile(f);
end;
end;
{=====--//--//--//=====}
procedure DumpEx(const Buffer: AnsiString; DumpFile: string);
var
f: Text;
begin
AssignFile(f, DumpFile);
if FileExists(DumpFile) then
DeleteFile(DumpFile);
Rewrite(f);
try
Writeln(f, DumpExStr(Buffer));
finally
CloseFile(f);
end;
end;
{=====--//--//--//=====}
function TrimSPLeft(const S: string): string;
var
I, L: Integer;
begin
L:= Length(S);
I:= 1;
while (I <= L) and (S[I] = ' ') do
Inc(I);
Result:= Copy(S, I, Maxint);
end;
{=====--//--//--//=====}
function TrimSPRight(const S: string): string;
var
I: Integer;
begin
I:= Length(S);
while (I > 0) and (S[I] = ' ') do
Dec(I);
Result:= Copy(S, 1, I);
end;

{=====--//--//--//=====}

function TrimSP(const S: string): string;
begin
Result:= TrimSPLeft(s);
Result:= TrimSPRight(Result);
end;

```

```

{=====--//--//--//=====}
function SeparateLeft(const Value, Delimiter: string): string;
var
x: Integer;
begin
x:= Pos(Delimiter, Value);
if x < 1 then
Result:= Value
else
Result:= Copy(Value, 1, x - 1);
end;
{=====--//--//--//=====}
function SeparateRight(const Value, Delimiter: string): string;
var
x: Integer;
begin
x:= Pos(Delimiter, Value);
if x > 0 then
x:= x + Length(Delimiter) - 1;
Result:= Copy(Value, x + 1, Length(Value) - x);
end;
{=====--//--//--//=====}
function GetParameter(const Value, Parameter: string): string;
var
s: string;
v: string;
begin
Result:= '';
v:= Value;
while v <> '' do
begin
s:= Trim(FetchEx(v, ';', ''));
if Pos(Uppercase(parameter), Uppercase(s)) = 1 then
begin
Delete(s, 1, Length(Parameter));
s:= Trim(s);
if s = '' then
Break;
if s[1] = '=' then
begin
Result:= Trim(SeparateRight(s, '='));
Result:= UnquoteStr(Result, '"');
break;
end;
end;
end;
end;
end;
{=====--//--//--//=====}
procedure ParseParametersEx(Value, Delimiter: string; const Parameters: TStrings);
var
s: string;
begin
Parameters.Clear;
while Value <> '' do
begin
s:= Trim(FetchEx(Value, Delimiter, ''));
Parameters.Add(s);
end;
end;
{=====--//--//--//=====}
procedure ParseParameters(Value: string; const Parameters: TStrings);
begin
ParseParametersEx(Value, ';', Parameters);
end;
{=====--//--//--//=====}
function IndexByBegin(Value: string; const List: TStrings): integer;
var
n: integer;
s: string;

```

```

begin
Result:= -1;
Value:= uppercase(Value);
for n:= 0 to List.Count -1 do
begin
s:= UpperCase(List[n]);
if Pos(Value, s) = 1 then
begin
Result:= n;
Break;
end;
end;
end;
end;
{=====//--//--//=====}
function GetEmailAddr(const Value: string): string;
var
s: string;
begin
s:= SeparateRight(Value, '<');
s:= SeparateLeft(s, '>');
Result:= Trim(s);
end;
{=====//--//--//=====}
function GetEmailDesc(Value: string): string;
var
s: string;
begin
Value:= Trim(Value);
s:= SeparateRight(Value, '');
if s <> Value then
s:= SeparateLeft(s, '')
else
begin
s:= SeparateLeft(Value, '<');
if s = Value then
begin
s:= SeparateRight(Value, '(');
if s <> Value then
s:= SeparateLeft(s, ')')
else
s:= '';
end;
end;
Result:= Trim(s);
end;
{=====//--//--//=====}
function StrToHex(const Value: Ansistring): string;
var
n: Integer;
begin
Result:= '';
for n:= 1 to Length(Value) do
Result:= Result + IntToHex(Byte(Value[n]), 2);
Result:= LowerCase(Result);
end;
{=====//--//--//=====}
function IntToBin(Value: Integer; Digits: Byte): string;
var
x, y, n: Integer;
begin
Result:= '';
x:= Value;
repeat
y:= x mod 2;
x:= x div 2;
if y > 0 then
Result:= '1' + Result
else
Result:= '0' + Result;

```

```

until x = 0;
x:= Length(Result);
for n:= x to Digits - 1 do
Result:= '0' + Result;
end;
{=====//--//--//=====}
function BinToInt(const Value: string): Integer;
var
n: Integer;
begin
Result:= 0;
for n:= 1 to Length(Value) do
begin
if Value[n] = '0' then
Result:= Result * 2
else
if Value[n] = '1' then
Result:= Result * 2 + 1
else
Break;
end;
end;
end;
{=====//--//--//=====}
function ParseURL(URL: string; var Prot, User, Pass, Host, Port, Path,
Para: string): string;
var
x, y: Integer;
sURL: string;
s: string;
s1, s2: string;
begin
Prot:= 'http';
User:= '';
Pass:= '';
Port:= '80';
Para:= '';

x:= Pos('://', URL);
if x > 0 then
begin
Prot:= SeparateLeft(URL, '://');
sURL:= SeparateRight(URL, '://');
end
else
sURL:= URL;
if UpperCase(Prot) = 'HTTPS' then
Port:= '443';
if UpperCase(Prot) = 'FTP' then
Port:= '21';
x:= Pos('@', sURL);
y:= Pos('/', sURL);
if (x > 0) and ((x < y) or (y < 1)) then
begin
s:= SeparateLeft(sURL, '@');
sURL:= SeparateRight(sURL, '@');
x:= Pos(':', s);
if x > 0 then
begin
User:= SeparateLeft(s, ':');
Pass:= SeparateRight(s, ':');
end
else
User:= s;
end;
x:= Pos('/', sURL);
if x > 0 then
begin
s1:= SeparateLeft(sURL, '/');
s2:= SeparateRight(sURL, '/');

```

```

end
else
begin
s1:= sURL;
s2:= '';
end;
if Pos('[', s1) = 1 then
begin
Host:= Separateleft(s1, ']');
Delete(Host, 1, 1);
s1:= SeparateRight(s1, ']');
if Pos(':', s1) = 1 then
Port:= SeparateRight(s1, ':');
end
else
begin
x:= Pos(':', s1);
if x > 0 then
begin
Host:= SeparateLeft(s1, ':');
Port:= SeparateRight(s1, ':');
end
else
Host:= s1;
end;
Result:= '/' + s2;
x:= Pos('?', s2);
if x > 0 then
begin
Path:= '/' + SeparateLeft(s2, '?');
Para:= SeparateRight(s2, '?');
end
else
Path:= '/' + s2;
if Host = '' then
Host:= 'localhost';
end;
{=====--//--//--//=====}
function ReplaceString(Value, Search, Replace: string): string;
var
x, l, ls, lr: Integer;
begin
if (Value = '') or (Search = '') then
begin
Result:= Value;
Exit;
end;
ls:= Length(Search);
lr:= Length(Replace);
Result:= '';
x:= Pos(Search, Value);
while x > 0 do
begin
l:= Length(Result);
SetLength(Result, l + x - 1);
Move(Pointer(Value)^, Pointer(@Result[l + 1])^, x - 1);
Result:=Result+Copy(Value,1,x-1);
l:= Length(Result);
SetLength(Result, l + lr);
Move(Pointer(Replace)^, Pointer(@Result[l + 1])^, lr);
Result:=Result+Replace;
Delete(Value, 1, x - 1 + ls);
x:= Pos(Search, Value);
end;
Result:= Result + Value;
end;
{=====--//--//--//=====}
function RPosEx(const Sub, Value: string; From: integer): Integer;
var

```

```

n: Integer;
l: Integer;
begin
result:= 0;
l:= Length(Sub);
for n:= From - l + 1 downto 1 do
begin
if Copy(Value, n, l) = Sub then
begin
result:= n;
break;
end;
end;
end;
{=====//--//--//=====}
function RPos(const Sub, Value: String): Integer;
begin
Result:= RPosEx(Sub, Value, Length(Value));
end;
{=====//--//--//=====}
function FetchBin(var Value: string; const Delimiter: string): string;
var
s: string;
begin
Result:= SeparateLeft(Value, Delimiter);
s:= SeparateRight(Value, Delimiter);
if s = Value then
Value:= ''
else
Value:= s;
end;
{=====//--//--//=====}
function Fetch(var Value: string; const Delimiter: string): string;
begin
Result:= FetchBin(Value, Delimiter);
Result:= TrimSP(Result);
Value:= TrimSP(Value);
end;
{=====//--//--//=====}
function FetchEx(var Value: string; const Delimiter, Quotation: string): string;
var
n: integer;
b: Boolean;
begin
Result:= '';
b:= False;
n:= 1;
while n <= Length(Value) do
begin
if b then
begin
if Pos(Quotation, Value) = 1 then
b:= False;
Result:= Result + Value[1];
Delete(Value, 1, 1);
end
else
begin
if Pos(Delimiter, Value) = 1 then
begin
Delete(Value, 1, Length(delimiter));
break;
end;
b:= Pos(Quotation, Value) = 1;
Result:= Result + Value[1];
Delete(Value, 1, 1);
end;
end;
Result:= Result;

```

```

end;
{=====--//--//--//=====}
function IsBinaryString(const Value: string): Boolean;
var
n: integer;
begin
Result:= False;
for n:= 1 to Length(Value) do
if Value[n] in [#0..#8, #10..#31] then
begin
Result:= True;
Break;
end;
end;
{=====--//--//--//=====}
function PosCRLF(const Value: AnsiString; var Terminator: AnsiString): integer;
var
p1, p2, p3, p4: integer;
const
t1 = #$0d + #$0a;
t2 = #$0a + #$0d;
t3 = #$0d;
t4 = #$0a;
begin
Terminator:= '';
p1:= Pos(t1, Value);
p2:= Pos(t2, Value);
p3:= Pos(t3, Value);
p4:= Pos(t4, Value);
if p1 > 0 then
Terminator:= t1;
Result:= p1;
if (p2 > 0) then
if (Result = 0) or (p2 < Result) then
begin
Result:= p2;
Terminator:= t2;
end;
if (p3 > 0) then
if (Result = 0) or (p3 < Result) then
begin
Result:= p3;
Terminator:= t3;
end;
if (p4 > 0) then
if (Result = 0) or (p4 < Result) then
begin
Result:= p4;
Terminator:= t4;
end;
end;
{=====--//--//--//=====}
Procedure StringsTrim(const Value: TStrings);
var
n: integer;
begin
for n:= Value.Count - 1 downto 0 do
if Value[n] = '' then
Value.Delete(n)
else
Break;
end;
{=====--//--//--//=====}
function PosFrom(const SubStr, Value: String; From: integer): integer;
var
ls,lv: integer;
begin
Result:= 0;
ls:= Length(SubStr);

```

```

lv:= Length(Value);
if (ls = 0) or (lv = 0) then
Exit;
if From < 1 then
From:= 1;
while (ls + from - 1) <= (lv) do
begin
if CompareMem(@SubStr[1],@Value[from],ls) then
if SubStr = copy(Value, from, ls) then
begin
result:= from;
break;
end
else
inc(from);
end;
end;
{=====//--//--//=====}
function IncPoint(const p: pointer; Value: integer): pointer;
begin
Result:= pointer(integer(p) + Value);
end;
{=====//--//--//=====}
function GetBetween(const PairBegin, PairEnd, Value: string): string;
var
n: integer;
x: integer;
s: string;
begin
Result:= '';
s:= SeparateRight(Value, PairBegin);
x:= 1;
for n:= 1 to Length(s) do
begin
if s[n] = PairBegin then
Inc(x);
if s[n] = PairEnd then
begin
Dec(x);
if x <= 0 then
Break;
end;
Result:= Result + s[n];
end;
end;
{=====//--//--//=====}
function CountOfChar(const Value: string; Chr: char): integer;
var
n: integer;
begin
Result:= 0;
for n:= 1 to Length(Value) do
if Value[n] = chr then
Inc(Result);
end;
end;
{=====//--//--//=====}
function UnquoteStr(Value: string; Quote: Char): string;
var
LText: PChar;
begin
if length(Value) > 1 then
if (Value[1] = Quote) and (Value[Length(value)] <> Quote) then
Value:= Value + Quote;
LText:= PChar(Value);
Result:= AnsiExtractQuotedStr(LText, Quote);
Result:= DequotedStr(Value, Quote);
if Result = '' then
Result:= Value;
end;
end;

```

```

{=====--//--//--//=====}
procedure HeadersToList(const Value: TStrings);
var
n, x: integer;
s: string;
begin
for n:= 0 to Value.Count -1 do
begin
s:= Value[n];
x:= Pos(':', s);
if x > 0 then
begin
s[x]:= '=';
Value[n]:= s;
end;
end;
end;
{=====--//--//--//=====}
procedure ListToHeaders(const Value: TStrings);
var
n, x: integer;
s: string;
begin
for n:= 0 to Value.Count -1 do
begin
s:= Value[n];
x:= Pos('=', s);
if x > 0 then
begin
s[x]:= ':';
Value[n]:= s;
end;
end;
end;
{=====--//--//--//=====}
function SwapBytes(Value: integer): integer;
var
s: string;
x, y, xl, yl: Byte;
begin
s:= CodeLongInt(Value);
x:= Ord(s[4]);
y:= Ord(s[3]);
xl:= Ord(s[2]);
yl:= Ord(s[1]);
Result:= ((x * 256 + y) * 65536) + (xl * 256 + yl);
end;
{=====--//--//--//=====}
function ReadStrFromStream(const Stream: TStream; len: integer): AnsiString;
var
x: integer;
buf: Array of Byte;
begin
SetLength(buf, Len);
x:= Stream.read(buf, Len);
SetLength(buf, x);
Result:= StringOf(buf);
SetLength(Result, Len);
x:= Stream.read(PChar(Result)^, Len);
SetLength(Result, x);
end;
{=====--//--//--//=====}
procedure WriteStrToStream(const Stream: TStream; Value: AnsiString);
var
buf: Array of Byte;
begin
buf:= BytesOf(Value);
Stream.Write(buf, length(Value));
Stream.Write(PChar(Value)^, Length(Value));

```

```
end;
{=====//--//--//=====}
var
n: integer;
begin
for n:= 1 to 12 do
begin
CustomMonthNames[n]:= ShortMonthNames[n];
MyMonthNames[0, n]:= ShortMonthNames[n];
end;
end.
```

Файл Splash вікна

```
unit Unit3;
{Copyright © 2021 CNTU AZATIAN Z.R.}
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, GlPrgrs, ExtCtrls;
```

```
type
```

```
TCM_SPLASH = class(TForm)
```

```
glProgress1: TglProgress;
```

```
Label1: TLabel;
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
CM_SPLASH: TCM_SPLASH;
```

```
implementation
```

```
{$R *.dfm}
```

```
end.
```

Файл роботи з RS-232 портом

```

unit synaser;
{Copyright © 2021 CNTU AZATIAN Z.R.}

interface

uses
  synafpc,
  Libc, KernelIoctl, Types, Windows, Classes, registry, winver, SysUtils, synautil;

const
  CR = #$0d;
  LF = #$0a;
  CRLF = CR + LF;
  cSerialChunk = 8192;
  PortIsClosed = -1;
  SB1 = 0;
  SB1andHalf = 1;
  SB2 = 2;

const
  INVALID_HANDLE_VALUE = THandle(-1);
  CS7fix = $0000020;

type
  TDCB = packed record
    DCBlength: DWORD;
    BaudRate: DWORD;
    Flags: Longint;
    wReserved: Word;
    XonLim: Word;
    XoffLim: Word;
    ByteSize: Byte;
    Parity: Byte;
    StopBits: Byte;
    XonChar: CHAR;
    XoffChar: CHAR;
    MY_Err_orChar: CHAR;
    EofChar: CHAR;
    EvtChar: CHAR;
    wReserved1: Word;
  end;
  PDCB = ^TDCB;

const
  MaxRates = 30;

Rates: array[0..MaxRates, 0..1] of cardinal =
  ((1800, B1800),
  (2400, B2400),
  (4800, B4800),
  (9600, B9600),
  (19200, B19200),
  (38400, B38400),
  (57600, B57600),
  (115200, B115200),
  (230400, B230400),
  (460800, B460800),
  (500000, B500000),
  (576000, B576000));

const
  sOK = 0;
  sMY_Err_ = integer(-1);

type

```

```

THookSerialReason = (
HR_SerialClose,
HR_Connect,
HR_CanRead,
HR_CanWrite,
HR_ReadCount,
HR_WriteCount,
HR_Wait);

THookSerialStatus = procedure(Sender: TObject; Reason: THookSerialReason;
const Value: string) of object;

ESynaSerMY_Err_or = class(Exception)
public
MY_Err_orCode: integer;
MY_Err_orMessage: string;
end;

TBlockSerial = class(TObject)
protected
FOnStatus: THookSerialStatus;
FHandle: THandle;
FTag: integer;
FDevice: string;
FLastMY_Err_or: integer;
FLastMY_Err_orDesc: string;
FBuffer: string;
FRaiseExcept: boolean;
FRecvBuffer: integer;
FSendBuffer: integer;
FModemWord: integer;
FRTSToggle: Boolean;
FDeadlockTimeout: integer;
FInstanceActive: boolean;
FTestDSR: Boolean;
FTestCTS: Boolean;
FLastCR: Boolean;
FLastLF: Boolean;
FMaxLineLength: Integer;
FLNLock: Boolean;
FMaxSendBandwidth: Integer;
FNextSend: ULONG;
FMaxRecvBandwidth: Integer;
FNextRecv: ULONG;
FConvertLineEnd: Boolean;
FATResult: Boolean;
FATTimeout: integer;
FInterPacketTimeout: Boolean;
FComNr: integer;
FPortAddr: Word;
function CanEvent(Event: dword; Timeout: integer): boolean;
procedure DecodeCommMY_Err_or(MY_Err_or: DWord); virtual;
function GetPortAddr: Word; virtual;
function ReadTxEmpty(PortAddr: Word): Boolean; virtual;
procedure SetSizeRecvBuffer(size: integer); virtual;
function GetDSR: Boolean; virtual;
procedure SetDTRF(Value: Boolean); virtual;
function GetCTS: Boolean; virtual;
procedure SetRTSF(Value: Boolean); virtual;
function GetCarrier: Boolean; virtual;
function GetRing: Boolean; virtual;
procedure DoStatus(Reason: THookSerialReason; const Value: string); virtual;
procedure GetComNr(Value: string); virtual;
function PreTestFailing: boolean; virtual;
function TestCtrlLine: Boolean; virtual;

procedure DcbToTermios(const dcb: TDCB; var term: termios); virtual;
procedure TermiosToDcb(const term: termios; var dcb: TDCB); virtual;

```

```

function ReadLockfile: integer; virtual;
function LockfileName: String; virtual;
procedure CreateLockfile(PidNr: integer); virtual;
procedure LimitBandwidth(Length: Integer; MaxB: integer; var Next: ULong); virtual;
procedure SetBandwidth(Value: Integer); virtual;
public
TermiosStruc: termios;
constructor Create;
destructor Destroy; ovMY_Err_ide;
class function GetVersion: string; virtual;
procedure CloseSocket; virtual;
procedure Connect(comport: string); virtual;
procedure SetCommState; virtual;
procedure GetCommState; virtual;
function SendBuffer(buffer: pointer; length: integer): integer; virtual;
procedure SendByte(data: byte); virtual;
procedure SendString(data: string); virtual;
procedure SendInteger(Data: integer); virtual;
procedure SendBlock(const Data: string); virtual;
procedure SendStreamRaw(const Stream: TStream); virtual;
procedure SendStream(const Stream: TStream); virtual;
procedure SendStreamIndy(const Stream: TStream); virtual;
function RecvBuffer(buffer: pointer; length: integer): integer; virtual;
function RecvBufferEx(buffer: pointer; length: integer; timeout: integer): integer;
virtual;
function RecvBufferStr(Length: Integer; Timeout: Integer): string; virtual;
function RecvPacket(Timeout: Integer): string; virtual;
function RecvByte(timeout: integer): byte; virtual;
function RecvTerminated(Timeout: Integer; const Terminator: string): string;
virtual;
function Recvstring(timeout: integer): string; virtual;
function RecvInteger(Timeout: Integer): Integer; virtual;
function RecvBlock(Timeout: Integer): string; virtual;
function WaitingData: integer; virtual;
function WaitingDataEx: integer; virtual;
function SendingData: integer; virtual;
procedure Flush; virtual;
procedure Purge; virtual;
function CanRead(Timeout: integer): boolean; virtual;
function CanWrite(Timeout: integer): boolean; virtual;
function CanReadEx(Timeout: integer): boolean; virtual;
function ModemStatus: integer; virtual;
procedure SetBreak(Duration: integer); virtual;
function ATCommand(value: string): string; virtual;
function ATConnect(value: string): string; virtual;
function SerialCheck(SerialResult: integer): integer; virtual;
procedure ExceptCheck; virtual;
procedure SetSynaMY_Err_or(MY_Err_Number: integer); virtual;
procedure RaiseSynaMY_Err_or(MY_Err_Number: integer); virtual;
function cponComportAccessible: boolean; virtual;
procedure cponReleaseComport; virtual;
property Device: string read FDevice;
property LastMY_Err_or: integer read FLastMY_Err_or;
property LastMY_Err_orDesc: string read FLastMY_Err_orDesc;
property ATResult: Boolean read FATResult;
property RTS: Boolean write SetRTSF;
property CTS: boolean read GetCTS;
property DTR: Boolean write SetDTRF;
property DSR: boolean read GetDSR;
property Carrier: boolean read GetCarrier;
property Ring: boolean read GetRing;
property InstanceActive: boolean read FInstanceActive;
property MaxSendBandwidth: Integer read FMaxSendBandwidth Write FMaxSendBandwidth;
property MaxRecvBandwidth: Integer read FMaxRecvBandwidth Write FMaxRecvBandwidth;
property MaxBandwidth: Integer Write SetBandwidth;
property SizeRecvBuffer: integer read FRecvBuffer write SetSizeRecvBuffer;
published
class function GetMY_Err_orDesc(MY_Err_orCode: integer): string;
property Tag: integer read FTag write FTag;

```

```

property Handle: THandle read FHandle write FHandle;
property LineBuffer: string read FBuffer write FBuffer;
property RaiseExcept: boolean read FRaiseExcept write FRaiseExcept;
property OnStatus: THookSerialStatus read FOnStatus write FOnStatus;
property TestDSR: boolean read FTestDSR write FTestDSR;
property TestCTS: boolean read FTestCTS write FTestCTS;
property MaxLineLength: Integer read FMaxLineLength Write FMaxLineLength;
property DeadlockTimeout: Integer read FDeadlockTimeout Write FDeadlockTimeout;
property LNLock: Boolean read FLNLock write FLNLock;
property ConvertLineEnd: Boolean read FConvertLineEnd Write FConvertLineEnd;
property AtTimeout: integer read FATimeout Write FATimeout;
property InterPacketTimeout: Boolean read FInterPacketTimeout Write
FInterPacketTimeout;
end;
function GetSerialPortNames: string;
implementation

constructor TBlockSerial.Create;
begin
inherited create;
FRaiseExcept:= false;
FHandle:= INVALID_HANDLE_VALUE;
FDevice:= '';
FComNr:= PortIsClosed;
FInstanceActive:= false;
Fbuffer:= '';
FRTSSToggle:= False;
FMaxLineLength:= 0;
FTestDSR:= False;
FTestCTS:= False;
FDeadlockTimeout:= 30000;
FLNLock:= True;
FMaxSendBandwidth:= 0;
FNextSend:= 0;
FMaxRecvBandwidth:= 0;
FNextRecv:= 0;
FConvertLineEnd:= False;
SetSynaMY_Err_or(sOK);
FRecvBuffer:= 4096;
FLastCR:= False;
FLastLF:= False;
FATimeout:= 1000;
FInterPacketTimeout:= True;
end;

destructor TBlockSerial.Destroy;
begin
CloseSocket;
inherited destroy;
end;
class function TBlockSerial.GetVersion: string;
begin
Result:= 'SynaSer 6.3.5';
end;

procedure TBlockSerial.CloseSocket;
begin
if FHandle <> INVALID_HANDLE_VALUE then
begin
Purge;
RTS:= False;
DTR:= False;
FileClose(integer(FHandle));
end;
if InstanceActive then
begin
if FLNLock then
cpomReleaseComport;
FInstanceActive:= false

```

```

end;
Fhandle:= INVALID_HANDLE_VALUE;
FComNr:= PortIsClosed;
SetSynaMY_Err_or(sOK);
DoStatus(HR_SerialClose, FDevice);
end;

procedure TBlockSerial.GetComNr(Value: string);
begin
FComNr:= PortIsClosed;
if pos('COM', uppercase(Value)) = 1 then
FComNr:= StrToIntDef(copy(Value, 4, Length(Value)-3), PortIsClosed+1)-1;
if pos('/DEV/TTYS', uppercase(Value)) = 1 then
FComNr:= StrToIntDef(copy(Value, 10, Length(Value)-9), PortIsClosed-1);
end;

procedure TBlockSerial.SetBandwidth(Value: Integer);
begin
MaxSendBandwidth:= Value;
MaxRecvBandwidth:= Value;
end;

procedure TBlockSerial.LimitBandwidth(Length: Integer; MaxB: integer; var Next: ULONG);
var
x: ULONG;
y: ULONG;
begin
if MaxB > 0 then
begin
y:= GetTick;
if Next > y then
begin
x:= Next - y;
if x > 0 then
begin
DoStatus(HR_Wait, IntToStr(x));
sleep(x);
end;
end;
Next:= GetTick + Trunc((Length / MaxB) * 1000);
end;
end;

procedure TBlockSerial.Config(baud, bits: integer; parity: char; stop: integer;
softflow, hardflow: boolean);
begin
FillChar(dcb, SizeOf(dcb), 0);
dcb.DCBLength:= SizeOf(dcb);
dcb.BaudRate:= baud;
dcb.ByteSize:= bits;
case parity of
'N', 'n': dcb.parity:= 0;
'O', 'o': dcb.parity:= 1;
'E', 'e': dcb.parity:= 2;
'M', 'm': dcb.parity:= 3;
'S', 's': dcb.parity:= 4;
end;
dcb.StopBits:= stop;
dcb.XonChar:= #17;
dcb.XoffChar:= #19;
dcb.XonLim:= FRecvBuffer div 4;
dcb.XoffLim:= FRecvBuffer div 4;
dcb.Flags:= GONCHARENKO_Binary;
if softflow then
dcb.Flags:= dcb.Flags or GONCHARENKO_OutX or GONCHARENKO_InX;
if hardflow then
dcb.Flags:= dcb.Flags or GONCHARENKO_OutxCtsFlow or GONCHARENKO_RtsControlHandshake
else
dcb.Flags:= dcb.Flags or GONCHARENKO_RtsControlEnable;

```

```

dcb.Flags:= dcb.Flags or GONCHARENKO_DtrControlEnable;
if dcb.Parity > 0 then
dcb.Flags:= dcb.Flags or GONCHARENKO_ParityCheck;
SetCommState;
end;

procedure TBlockSerial.Connect(comport: string);
var
CommTimeouts: TCommTimeouts;
begin
if InstanceActive then
begin
RaiseSynaMY_Err_or(MY_Err_AlreadyInUse);
Exit;
end;
FBuffer:= '';
FDevice:= comport;
GetComNr(comport);
SetLastMY_Err_or(sOK);
SetLastMY_Err_or(sOK);
_MY_Err_no_location^:= sOK;
if FComNr <> PortIsClosed then
FDevice:= '/dev/ttyS' + IntToStr(FComNr);
if FLNLock then
if not cpomComportAccessible then
begin
RaiseSynaMY_Err_or(MY_Err_AlreadyOwned);
Exit;
end;
FHandle:= THandle(Libc.open(pchar(FDevice), O_RDWR or O_SYNC));
SerialCheck(integer(FHandle));
if FLastMY_Err_or <> sOK then
if FLNLock then
cpomReleaseComport;
ExceptCheck;
if FLastMY_Err_or <> sOK then
Exit;
if FComNr <> PortIsClosed then
FDevice:= '\\.\COM' + IntToStr(FComNr + 1);
FHandle:= THandle(CreateFile(PChar(FDevice), GENERIC_READ or GENERIC_WRITE,
0, nil, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL or FILE_FLAG_OVERLAPPED, 0));
SerialCheck(integer(FHandle));
ExceptCheck;
if FLastMY_Err_or <> sOK then
Exit;
SetCommMask(FHandle, 0);
SetupComm(Fhandle, FRecvBuffer, 0);
CommTimeOuts.ReadIntervalTimeout:= MAXWORD;
CommTimeOuts.ReadTotalTimeoutMultiplier:= 0;
CommTimeOuts.ReadTotalTimeoutConstant:= 0;
CommTimeOuts.WriteTotalTimeoutMultiplier:= 0;
CommTimeOuts.WriteTotalTimeoutConstant:= 0;
SetCommTimeOuts(FHandle, CommTimeOuts);
FPortAddr:= GetPortAddr;
SetSynaMY_Err_or(sOK);
if not TestCtrlLine then
begin
SetSynaMY_Err_or(MY_Err_NoDeviceAnswer);
FileClose(integer(FHandle));
if FLNLock then
cpomReleaseComport;
Fhandle:= INVALID_HANDLE_VALUE;
FComNr:= PortIsClosed;
end
else
begin
FInstanceActive:= True;
RTS:= True;
DTR:= True;

```

```

Purge;
end;
ExceptCheck;
DoStatus(HR_Connect, FDevice);
end;

function TBlockSerial.SendBuffer(buffer: pointer; length: integer): integer;
var
  Overlapped: TOverlapped;
  x, y, MY_Err_: DWord;
begin
  Result:= 0;
  if PreTestFailing then
  Exit;
  LimitBandwidth(Length, FMaxSendBandwidth, FNextsend);
  if FRTSToggle then
  begin
  Flush;
  RTS:= True;
  end;
  result:= FileWrite(integer(FHandle), Buffer^, Length);
  serialcheck(result);
  FillChar(Overlapped, Sizeof(Overlapped), 0);
  SetSynaMY_Err_or(sOK);
  y:= 0;
  if not WriteFile(FHandle, Buffer^, Length, DWord(Result), @Overlapped) then
  y:= GetLastMY_Err_or;
  if y = MY_Err_OR_IO_PENDING then
  begin
  x:= WaitForSingleObject(FHandle, FDeadlockTimeout);
  if x = WAIT_TIMEOUT then
  begin
  PurgeComm(FHandle, PURGE_TXABORT);
  SetSynaMY_Err_or(MY_Err_Timeout);
  end;
  GetOverlappedResult(FHandle, Overlapped, Dword(Result), False);
  end
  else
  SetSynaMY_Err_or(y);
  ClearCommMY_Err_or(FHandle, MY_Err_, nil);
  if MY_Err_ <> 0 then
  DecodeCommMY_Err_or(MY_Err_);
  if FRTSToggle then
  begin
  Flush;
  CanWrite(255);
  RTS:= False;
  end;
  ExceptCheck;
  DoStatus(HR_WriteCount, IntToStr(Result));
  end;

  procedure TBlockSerial.SendByte(data: byte);
  begin
  SendBuffer(@Data, 1);
  end;

  procedure TBlockSerial.SendString(data: string);
  begin
  SendBuffer(Pointer(Data), Length(Data));
  end;

  procedure TBlockSerial.SendInteger(Data: integer);
  begin
  SendBuffer(@data, SizeOf(Data));
  end;

  procedure TBlockSerial.SendBlock(const Data: string);
  begin

```

```

SendInteger(Length(data));
SendString(Data);
end;

procedure TBlockSerial.SendStreamRaw(const Stream: TStream);
var
si: integer;
x, y, yr: integer;
s: string;
begin
si:= Stream.Size - Stream.Position;
x:= 0;
while x < si do
begin
y:= si - x;
if y > cSerialChunk then
y:= cSerialChunk;
Setlength(s, y);
yr:= Stream.read(Pchar(s)^, y);
if yr > 0 then
begin
SetLength(s, yr);
SendString(s);
Inc(x, yr);
end
else
break;
end;
end;

procedure TBlockSerial.SendStreamIndy(const Stream: TStream);
var
si: integer;
begin
si:= Stream.Size - Stream.Position;
si:= Swapbytes(si);
SendInteger(si);
SendStreamRaw(Stream);
end;

procedure TBlockSerial.SendStream(const Stream: TStream);
var
si: integer;
begin
si:= Stream.Size - Stream.Position;
SendInteger(si);
SendStreamRaw(Stream);
end;

function TBlockSerial.RecvBuffer(buffer: pointer; length: integer): integer;
begin
Result:= 0;
if PreTestFailing then
Exit;
LimitBandwidth(Length, FMaxRecvBandwidth, FNextRecv);
result:= FileRead(integer(FHandle), Buffer^, length);
serialcheck(result);
var
Overlapped: TOverlapped;
x, y, MY_Err_: DWord;
begin
Result:= 0;
if PreTestFailing then
Exit;
LimitBandwidth(Length, FMaxRecvBandwidth, FNextRecv);
FillChar(Overlapped, Sizeof(Overlapped), 0);
SetSynaMY_Err_or(sOK);
y:= 0;
if not ReadFile(FHandle, Buffer^, length, Dword(Result), @Overlapped) then

```

```

y:= GetLastMY_Err_or;
if y = MY_Err_OR_IO_PENDING then
begin
x:= WaitForSingleObject(FHandle, FDeadlockTimeout);
if x = WAIT_TIMEOUT then
begin
PurgeComm(FHandle, PURGE_RXABORT);
SetSynaMY_Err_or(MY_Err_Timeout);
end;
GetOverlappedResult(FHandle, Overlapped, Dword(Result), False);
end
else
SetSynaMY_Err_or(y);
ClearCommMY_Err_or(FHandle, MY_Err_, nil);
if MY_Err_ <> 0 then
DecodeCommMY_Err_or(MY_Err_);
ExceptCheck;
DoStatus(HR_ReadCount, IntToStr(Result));
end;

function TBlockSerial.RecvBufferEx(buffer: pointer; length: integer; timeout:
integer): integer;
var
s: string;
rl, l: integer;
ti: ULong;
begin
Result:= 0;
if PreTestFailing then
Exit;
SetSynaMY_Err_or(sOK);
rl:= 0;
repeat
ti:= GetTick;
s:= RecvPacket(Timeout);
l:= System.Length(s);
if (rl + l) > Length then
l:= Length - rl;
Move(Pointer(s)^, IncPoint(Buffer, rl)^, l);
rl:= rl + l;
if FLastMY_Err_or <> sOK then
Break;
if rl >= Length then
Break;
if not FInterPacketTimeout then
begin
Timeout:= Timeout - integer(TickDelta(ti, GetTick));
if Timeout <= 0 then
begin
SetSynaMY_Err_or(MY_Err_Timeout);
Break;
end;
end;
until False;
delete(s, l, l);
FBuffer:= s;
Result:= rl;
end;

function TBlockSerial.RecvBufferStr(Length: Integer; Timeout: Integer): string;
var
x: integer;
begin
Result:= '';
if PreTestFailing then
Exit;
SetSynaMY_Err_or(sOK);
if Length > 0 then
begin

```

```

Setlength(Result, Length);
x:= RecvBufferEx(PChar(Result), Length , Timeout);
if FLastMY_Err_or = sOK then
SetLength(Result, x)
else
Result:= '';
end;
end;

function TBlockSerial.RecvPacket(Timeout: Integer): string;
var
x: integer;
begin
Result:= '';
if PreTestFailing then
Exit;
SetSynaMY_Err_or(sOK);
if FBuffer <> '' then
begin
Result:= FBuffer;
FBuffer:= '';
end
else
begin
Sleep(0);
x:= WaitingData;
if x > 0 then
begin
SetLength(Result, x);
x:= RecvBuffer(Pointer(Result), x);
if x >= 0 then
SetLength(Result, x);
end
else
begin
if CanRead(Timeout) then
begin
x:= WaitingData;
if x = 0 then
SetSynaMY_Err_or(MY_Err_Timeout);
if x > 0 then
begin
SetLength(Result, x);
x:= RecvBuffer(Pointer(Result), x);
if x >= 0 then
SetLength(Result, x);
end;
end
else
SetSynaMY_Err_or(MY_Err_Timeout);
end;
end;
ExceptCheck;
end;

function TBlockSerial.RecvByte(timeout: integer): byte;
begin
Result:= 0;
if PreTestFailing then
Exit;
SetSynaMY_Err_or(sOK);
if FBuffer = '' then
FBuffer:= RecvPacket(Timeout);
if (FLastMY_Err_or = sOK) and (FBuffer <> '') then
begin
Result:= Ord(FBuffer[1]);
System.Delete(FBuffer, 1, 1);
end;
ExceptCheck;

```

end;

```

function TBlockSerial.RecvTerminated(Timeout: Integer; const Terminator: string):
string;
var
x: Integer;
s: string;
l: Integer;
CorCRLF: Boolean;
t: string;
tl: integer;
ti: ULong;
begin
Result:= '';
if PreTestFailing then
Exit;
SetSynaMY_Err_or(sOK);
l:= system.Length(Terminator);
if l = 0 then
Exit;
tl:= 1;
CorCRLF:= FConvertLineEnd and (Terminator = CRLF);
s:= '';
x:= 0;
repeat
ti:= GetTick;
s:= s + RecvPacket(Timeout);
if FLastMY_Err_or <> sOK then
Break;
x:= 0;
if Length(s) > 0 then
if CorCRLF then
begin
if FLastCR and (s[1] = LF) then
Delete(s, 1, 1);
if FLastLF and (s[1] = CR) then
Delete(s, 1, 1);
FLastCR:= False;
FLastLF:= False;
t:= '';
x:= PosCRLF(s, t);
tl:= system.Length(t);
if t = CR then
FLastCR:= True;
if t = LF then
FLastLF:= True;
end
else
begin
x:= pos(Terminator, s);
tl:= 1;
end;
if (FMaxLineLength <> 0) and (system.Length(s) > FMaxLineLength) then
begin
SetSynaMY_Err_or(MY_Err_MaxBuffer);
Break;
end;
if x > 0 then
Break;
if not FInterPacketTimeout then
begin
Timeout:= Timeout - integer(TickDelta(ti, GetTick));
if Timeout <= 0 then
begin
SetSynaMY_Err_or(MY_Err_Timeout);
Break;
end;
end;
until False;

```

```

if x > 0 then
begin
Result:= Copy(s, 1, x - 1);
System.Delete(s, 1, x + tl - 1);
end;
FBuffer:= s;
ExceptCheck;
end;

function TBlockSerial.RecvString(Timeout: Integer): string;
var
s: string;
begin
Result:= '';
s:= RecvTerminated(Timeout, #13 + #10);
if FLastMY_Err_or = sOK then
Result:= s;
end;

function TBlockSerial.RecvInteger(Timeout: Integer): Integer;
var
s: string;
begin
Result:= 0;
s:= RecvBufferStr(4, Timeout);
if FLastMY_Err_or = 0 then
Result:= (ord(s[1]) + ord(s[2]) * 256) + (ord(s[3]) + ord(s[4]) * 256) * 65536;
end;

function TBlockSerial.RecvBlock(Timeout: Integer): string;
var
x: integer;
begin
Result:= '';
x:= RecvInteger(Timeout);
if FLastMY_Err_or = 0 then
Result:= RecvBufferStr(x, Timeout);
end;

procedure TBlockSerial.RecvStreamRaw(const Stream: TStream; Timeout: Integer);
var
s: string;
begin
repeat
s:= RecvPacket(Timeout);
if FLastMY_Err_or = 0 then
WriteStrToStream(Stream, s);
until FLastMY_Err_or <> 0;
end;

procedure TBlockSerial.RecvStreamSize(const Stream: TStream; Timeout: Integer; Size:
Integer);
var
s: string;
n: integer;
begin
for n:= 1 to (Size div cSerialChunk) do
begin
s:= RecvBufferStr(cSerialChunk, Timeout);
if FLastMY_Err_or <> 0 then
Exit;
Stream.Write(Pchar(s)^, cSerialChunk);
end;
n:= Size mod cSerialChunk;
if n > 0 then
begin
s:= RecvBufferStr(n, Timeout);
if FLastMY_Err_or <> 0 then
Exit;

```

```

Stream.Write(Pchar(s)^, n);
end;
end;

procedure TBlockSerial.RecvStreamIndy(const Stream: TStream; Timeout: Integer);
var
x: integer;
begin
x:= RecvInteger(Timeout);
x:= SwapBytes(x);
if FLastMY_Err_or = 0 then
RecvStreamSize(Stream, Timeout, x);
end;

procedure TBlockSerial.RecvStream(const Stream: TStream; Timeout: Integer);
var
x: integer;
begin
x:= RecvInteger(Timeout);
if FLastMY_Err_or = 0 then
RecvStreamSize(Stream, Timeout, x);
end;

function TBlockSerial.WaitingData: integer;
begin
serialcheck(ioctl(integer(FHandle), FIONREAD, @result));
if FLastMY_Err_or <> 0 then
Result:= 0;
ExceptCheck;
end;

function TBlockSerial.WaitingData: integer;
var
stat: TComStat;
MY_Err_: DWORD;
begin
if ClearCommMY_Err_or(FHandle, MY_Err_, @stat) then
begin
SetSynaMY_Err_or(sOK);
Result:= stat.cbInQue;
end
else
begin
SerialCheck(sMY_Err_);
Result:= 0;
end;
ExceptCheck;
end;

function TBlockSerial.WaitingDataEx: integer;
begin
if FBuffer <> '' then
Result:= Length(FBuffer)
else
Result:= Waitingdata;
end;

function TBlockSerial.SendingData: integer;
begin
SetSynaMY_Err_or(sOK);
Result:= 0;
end;

function TBlockSerial.SendingData: integer;
var
stat: TComStat;
MY_Err_: DWORD;
begin

```

```

SetSynaMY_Err_or(sOK);
if not ClearCommMY_Err_or(FHandle, MY_Err_, @stat) then
serialcheck(sMY_Err_);
ExceptCheck;
result:= stat.cbOutQue;
end;

procedure TBlockSerial.DcbToTermios(const dcb: TDCB; var term: termios);
var
n: integer;
x: cardinal;
begin
cfmakeraw(term);
term.c_cflag:= term.c_cflag or CREAD;
term.c_cflag:= term.c_cflag or CLOCAL;
term.c_cflag:= term.c_cflag or HUPCL;
if (dcb.flags and GONCHARENKO_RtsControlHandshake) > 0 then
term.c_cflag:= term.c_cflag or CRTSCTS
else
term.c_cflag:= term.c_cflag and (not CRTSCTS);
if (dcb.flags and GONCHARENKO_OutX) > 0 then
term.c_iflag:= term.c_iflag or IXON or IXOFF or IXANY
else
term.c_iflag:= term.c_iflag and (not (IXON or IXOFF or IXANY));
term.c_cflag:= term.c_cflag and (not CSIZE);
case dcb.bytesize of
5:
term.c_cflag:= term.c_cflag or CS5;
6:
term.c_cflag:= term.c_cflag or CS6;
7:
term.c_cflag:= term.c_cflag or CS7fix;
8:
term.c_cflag:= term.c_cflag or CS8;
end;
if (dcb.flags and GONCHARENKO_ParityCheck) > 0 then
term.c_cflag:= term.c_cflag or PARENB
else
term.c_cflag:= term.c_cflag and (not PARENB);
case dcb.parity of
1:
term.c_cflag:= term.c_cflag or PARODD;
2:
term.c_cflag:= term.c_cflag and (not PARODD);
end;
if dcb.stopbits > 0 then
term.c_cflag:= term.c_cflag or CSTOPB
else
term.c_cflag:= term.c_cflag and (not CSTOPB);
x:= 0;
for n:= 0 to Maxrates do
if rates[n, 0] = dcb.BaudRate then
begin
x:= rates[n, 1];
break;
end;
cfsetospeed(term, x);
cfsetispeed(term, x);
end;

procedure TBlockSerial.TermiosToDcb(const term: termios; var dcb: TDCB);
var
n: integer;
x: cardinal;
begin
dcb.baudrate:= 0;
x:= cfgetospeed(term);
for n:= 0 to Maxrates do
if rates[n, 1] = x then

```

```

begin
dcb.baudrate:= rates[n, 0];
break;
end;
if (term.c_cflag and CRTSCTS) > 0 then
dcb.flags:= dcb.flags or GONCHARENKO_RtsControlHandshake or GONCHARENKO_OutxCtsFlow
else
dcb.flags:= dcb.flags and (not (GONCHARENKO_RtsControlHandshake or
GONCHARENKO_OutxCtsFlow));
if (term.c_cflag and IXOFF) > 0 then
dcb.flags:= dcb.flags or GONCHARENKO_OutX or GONCHARENKO_InX
else
dcb.flags:= dcb.flags and (not (GONCHARENKO_OutX or GONCHARENKO_InX));
case term.c_cflag and CSIZE of
CS5:
dcb.bytesize:= 5;
CS6:
dcb.bytesize:= 6;
CS7fix:
dcb.bytesize:= 7;
CS8:
dcb.bytesize:= 8;
end;
if (term.c_cflag and PARENB) > 0 then
dcb.flags:= dcb.flags or GONCHARENKO_ParityCheck
else
dcb.flags:= dcb.flags and (not GONCHARENKO_ParityCheck);
dcb.parity:= 0;
if (term.c_cflag and PARODD) > 0 then
dcb.parity:= 1
else
dcb.parity:= 2;
if (term.c_cflag and CSTOPB) > 0 then
dcb.stopbits:= 2
else
dcb.stopbits:= 0;
end;

procedure TBlockSerial.SetCommState;
begin
DcbToTermios(dcb, termiosstruc);

SerialCheck(tcsetattr(integer(FHandle), TCSANOW, termiosstruc));

SerialCheck(tcsetattr(integer(FHandle), TCSANOW, @termiosstruc));

ExceptCheck;
end;

procedure TBlockSerial.SetCommState;
begin
SetSynaMY_Err_or(sOK);
if not windows.SetCommState(Fhandle, dcb) then
SerialCheck(sMY_Err_);
ExceptCheck;
end;

procedure TBlockSerial.GetCommState;
begin
SerialCheck(tcgetattr(integer(FHandle), termiosstruc));
ExceptCheck;
TermiostoDCB(termiosstruc, dcb);
end;

procedure TBlockSerial.GetCommState;
begin
SetSynaMY_Err_or(sOK);
if not windows.GetCommState(Fhandle, dcb) then
SerialCheck(sMY_Err_);

```

```

ExceptCheck;
end;

procedure TBlockSerial.SetSizeRecvBuffer(size: integer);
begin

SetupComm(Fhandle, size, 0);
GetCommState;
dcb.XonLim:= size div 4;
dcb.XoffLim:= size div 4;
SetCommState;

FRecvBuffer:= size;
end;

function TBlockSerial.GetDSR: Boolean;
begin
ModemStatus;
Result:= (FModemWord and TIOCM_DSR) > 0;
Result:= (FModemWord and MS_DSR_ON) > 0;
end;

procedure TBlockSerial.SetDTRF(Value: Boolean);
begin
ModemStatus;
if Value then
FModemWord:= FModemWord or TIOCM_DTR
else
FModemWord:= FModemWord and not TIOCM_DTR;
ioctl(integer(FHandle), TIOCMSET, @FModemWord);
if Value then
EscapeCommFunction(FHandle, SETDTR)
else
EscapeCommFunction(FHandle, CLRDR);
end;

function TBlockSerial.GetCTS: Boolean;
begin
ModemStatus;
Result:= (FModemWord and TIOCM_CTS) > 0;
Result:= (FModemWord and MS_CTS_ON) > 0;
end;

procedure TBlockSerial.SetRTSF(Value: Boolean);
begin

ModemStatus;
if Value then
FModemWord:= FModemWord or TIOCM_RTS
else
FModemWord:= FModemWord and not TIOCM_RTS;
ioctl(integer(FHandle), TIOCMSET, @FModemWord);

if Value then
EscapeCommFunction(FHandle, SETRTS)
else
EscapeCommFunction(FHandle, CLRRTS);
end;

function TBlockSerial.GetCarrier: Boolean;
begin
ModemStatus;
Result:= (FModemWord and TIOCM_CAR) > 0;
Result:= (FModemWord and MS_RLSD_ON) > 0;
end;

function TBlockSerial.GetRing: Boolean;
begin

```

```

ModemStatus;
Result:= (FModemWord and TIOCM_RNG) > 0;
Result:= (FModemWord and MS_RING_ON) > 0;
end;

```

```

function TBlockSerial.CanEvent(Event: dword; Timeout: integer): boolean;
var
ex: DWord;
y: Integer;
Overlapped: TOverlapped;
begin
FillChar(Overlapped, Sizeof(Overlapped), 0);
Overlapped.hEvent:= CreateEvent(nil, True, False, nil);
try
SetCommMask(FHandle, Event);
SetSynaMY_Err_or(sOK);
if (Event = EV_RXCHAR) and (Waitingdata > 0) then
Result:= True
else
begin
y:= 0;
if not WaitCommEvent(FHandle, ex, @Overlapped) then
y:= GetLastMY_Err_or;
if y = MY_Err_OR_IO_PENDING then
begin
WaitForSingleObject(Overlapped.hEvent, Timeout);
SetCommMask(FHandle, 0);
GetOverlappedResult(FHandle, Overlapped, DWord(y), True);
end;
Result:= (ex and Event) = Event;
end;
finally
SetCommMask(FHandle, 0);
CloseHandle(Overlapped.hEvent);
end;
end;

```

```

function TBlockSerial.CanRead(Timeout: integer): boolean;
var
FDSet: TFDSet;
TimeVal: PTimeVal;
TimeV: TTimeVal;
x: Integer;
begin
TimeV.tv_usec:= (Timeout mod 1000) * 1000;
TimeV.tv_sec:= Timeout div 1000;
TimeVal:= @TimeV;
if Timeout = -1 then
TimeVal:= nil;
FD_ZERO(FDSet);
FD_SET(integer(FHandle), FDSet);
x:= Select(integer(FHandle) + 1, @FDSet, nil, nil, TimeVal);
SerialCheck(x);
if FLastMY_Err_or <> sOK then
x:= 0;
Result:= x > 0;
ExceptCheck;
if Result then
DoStatus(HR_CanRead, '');
end;

```

```

function TBlockSerial.CanRead(Timeout: integer): boolean;
begin
Result:= WaitingData > 0;
if not Result then
Result:= CanEvent(EV_RXCHAR, Timeout);

```

```

if Result then
DoStatus(HR_CanRead, '');
end;

function TBlockSerial.CanWrite(Timeout: integer): boolean;
var
FDSet: TFDSet;
TimeVal: PTimeVal;
TimeV: TTimeVal;
x: Integer;
begin
TimeV.tv_usec:= (Timeout mod 1000) * 1000;
TimeV.tv_sec:= Timeout div 1000;
TimeVal:= @TimeV;
if Timeout = -1 then
TimeVal:= nil;
FD_ZERO(FDSet);
FD_SET(integer(FHandle), FDSet);
x:= Select(integer(FHandle) + 1, nil, @FDSet, nil, TimeVal);
SerialCheck(x);
if FLastMY_Err_or <> sOK then
x:= 0;
Result:= x > 0;
ExceptCheck;
if Result then
DoStatus(HR_CanWrite, '');
end;

function TBlockSerial.CanWrite(Timeout: integer): boolean;
var
t: ULong;
begin
Result:= SendingData = 0;
if not Result then
Result:= CanEvent(EV_TXEMPTY, Timeout);
if Result and (Win32Platform <> VER_PLATFORM_WIN32_NT) then
begin
t:= GetTick;
while not ReadTxEmpty(FPortAddr) do
begin
if TickDelta(t, GetTick) > 255 then
Break;
Sleep(0);
end;
end;
if Result then
DoStatus(HR_CanWrite, '');
end;

function TBlockSerial.CanReadEx(Timeout: integer): boolean;
begin
if Fbuffer <> '' then
Result:= True
else
Result:= CanRead(Timeout);
end;

procedure TBlockSerial.EnableRTSToggle(Value: boolean);
begin
SetSynaMY_Err_or(sOK);

FRTSToggle:= Value;
if Value then
RTS:=False;

if Win32Platform = VER_PLATFORM_WIN32_NT then
begin
GetCommState;

```

```

if value then
dcb.Flags:= dcb.Flags or GONCHARENKO_RtsControlToggle
else
dcb.flags:= dcb.flags and (not GONCHARENKO_RtsControlToggle);
SetCommState;
end
else
begin
FRTSToggle:= Value;
if Value then
RTS:=False;
end;
end;

procedure TBlockSerial.Flush;
begin
SerialCheck(tcdrain(integer(FHandle)));
SetSynaMY_Err_or(sOK);
if not Flushfilebuffers(FHandle) then
SerialCheck(sMY_Err_);
ExceptCheck;
end;

procedure TBlockSerial.Purge;
begin
SerialCheck(ioctl(integer(FHandle), TCFLSH, TCIOFLUSH));
FBuffer:= '';
ExceptCheck;
end;

procedure TBlockSerial.Purge;
var
x: integer;
begin
SetSynaMY_Err_or(sOK);
x:= PURGE_TXABORT or PURGE_TXCLEAR or PURGE_RXABORT or PURGE_RXCLEAR;
if not PurgeComm(FHandle, x) then
SerialCheck(sMY_Err_);
FBuffer:= '';
ExceptCheck;
end;

function TBlockSerial.ModemStatus: integer;
begin

SerialCheck(ioctl(integer(FHandle), TIOCMGET, @Result));

SetSynaMY_Err_or(sOK);
if not GetCommModemStatus(FHandle, dword(Result)) then
SerialCheck(sMY_Err_);

ExceptCheck;
FModemWord:= Result;
end;

procedure TBlockSerial.SetBreak(Duration: integer);
begin

SerialCheck(tcsendbreak(integer(FHandle), Duration));

SetCommBreak(FHandle);
Sleep(Duration);
SetSynaMY_Err_or(sOK);
if not ClearCommBreak(FHandle) then
SerialCheck(sMY_Err_);
end;

```

```

procedure TBlockSerial.DecodeCommMY_Err_or(MY_Err_or: DWord);
begin
if (MY_Err_or and DWord(CE_FRAME)) > 1 then
FLastMY_Err_or:= MY_Err_Frame;
if (MY_Err_or and DWord(CE_OVMY_Err_UN)) > 1 then
FLastMY_Err_or:= MY_Err_OvMY_Err_un;
if (MY_Err_or and DWord(CE_RXOVER)) > 1 then
FLastMY_Err_or:= MY_Err_RxOver;
if (MY_Err_or and DWord(CE_RXPARITY)) > 1 then
FLastMY_Err_or:= MY_Err_RxParity;
if (MY_Err_or and DWord(CE_TXFULL)) > 1 then
FLastMY_Err_or:= MY_Err_TxFull;
end;

```

```

function TBlockSerial.PreTestFailing: Boolean;
begin
if not FInstanceActive then
begin
RaiseSynaMY_Err_or(MY_Err_PortNotOpen);
result:= true;
Exit;
end;
Result:= not TestCtrlLine;
if result then
RaiseSynaMY_Err_or(MY_Err_NoDeviceAnswer)
end;

```

```

function TBlockSerial.TestCtrlLine: Boolean;
begin
result:= ((not FTestDSR) or DSR) and ((not FTestCTS) or CTS);
end;

```

```

function TBlockSerial.ATCommand(value: string): string;
var
s: string;
ConvSave: Boolean;
begin
result:= '';
FAtResult:= False;
ConvSave:= FConvertLineEnd;
try
FConvertLineEnd:= True;
SendString(value + #$0D);
repeat
s:= RecvString(FAtTimeout);
if s <> Value then
result:= result + s + CRLF;
if s = 'OK' then
begin
FAtResult:= True;
break;
end;
if s = 'MY_Err_OR' then
break;
until FLastMY_Err_or <> sOK;
finally
FConvertLineEnd:= Convsave;
end;
end;

```

```

function TBlockSerial.ATConnect(value: string): string;
var
s: string;
ConvSave: Boolean;
begin
result:= '';
FAtResult:= False;

```

```

ConvSave:= FConvertLineEnd;
try
FConvertLineEnd:= True;
SendString(value + #$0D);
repeat
s:= RecvString(90 * FAtTimeout);
if s <> Value then
result:= result + s + CRLF;
if s = 'NO CARRIER' then
break;
if s = 'MY_Err_OR' then
break;
if s = 'BUSY' then
break;
if s = 'NO DIALTONE' then
break;
if Pos('CONNECT', s) = 1 then
begin
FAtResult:= True;
break;
end;
until FLastMY_Err_or <> sOK;
finally
FConvertLineEnd:= Convsave;
end;
end;

function TBlockSerial.SerialCheck(SerialResult: integer): integer;
begin
if SerialResult = integer(INVALID_HANDLE_VALUE) then
result:= GetLastMY_Err_or
result:= GetLastMY_Err_or
result:= __MY_Err_no_location^
else
result:= sOK;
FLastMY_Err_or:= result;
FLastMY_Err_orDesc:= GetMY_Err_orDesc(FLastMY_Err_or);
end;

procedure TBlockSerial.ExceptCheck;
var
e: ESynaSerMY_Err_or;
s: string;
begin
if FRaiseExcept and (FLastMY_Err_or <> sOK) then
begin
s:= GetMY_Err_orDesc(FLastMY_Err_or);
e:= ESynaSerMY_Err_or.CreateFmt('Communication MY_Err_or %d: %s', [FLastMY_Err_or,
s]);
e.MY_Err_orCode:= FLastMY_Err_or;
e.MY_Err_orMessage:= s;
raise e;
end;
end;

procedure TBlockSerial.SetSynaMY_Err_or(MY_Err_Number: integer);
begin
FLastMY_Err_or:= MY_Err_Number;
FLastMY_Err_orDesc:= GetMY_Err_orDesc(FLastMY_Err_or);
end;

procedure TBlockSerial.RaiseSynaMY_Err_or(MY_Err_Number: integer);
begin
SetSynaMY_Err_or(MY_Err_Number);
ExceptCheck;
end;

procedure TBlockSerial.DoStatus(Reason: THookSerialReason; const Value: string);
begin

```

```

if assigned(OnStatus) then
OnStatus(Self, Reason, Value);
end;

function TBlockSerial.LockfileName: String;
var
s: string;
begin
s:= SeparateRight(FDevice, '/dev/');
result:= LockfileDirectory + '/LCK..' + s;
end;

procedure TBlockSerial.CreateLockfile(PidNr: integer);
var
f: TextFile;
s: string;
begin
s:= IntToStr(PidNr);
while length(s) < 10 do
s:= ' ' + s;
try
AssignFile(f, LockfileName);
try
Rewrite(f);
writeln(f, s);
finally
CloseFile(f);
end;
s:='chmod a+rw ' + LockfileName;
Libc.system(pchar(s));
Libc.__system(pchar(s));
except
end;
end;

function TBlockSerial.ReadLockfile: integer;
var
f: TextFile;
s: string;
begin
AssignFile(f, LockfileName);
Reset(f);
try
readln(f, s);
finally
CloseFile(f);
end;
Result:= StrToIntDef(s, -1)
end;

function TBlockSerial.cpmComportAccessible: boolean;
var
MyPid: integer;
Filename: string;
begin
Filename:= LockfileName;
MyPid:= Libc.getpid;
if not DirectoryExists(LockfileDirectory) then
CreateDir(LockfileDirectory);
if not FileExists (Filename) then
begin
CreateLockfile(MyPid);
result:= true;
exit;
end;
if Libc.getsid(ReadLockfile) = -1 then
begin
DeleteFile(Filename);
CreateLockfile(MyPid);

```

```
result:= true;
exit;
end;
result:= false
end;
procedure TBlockSerial.cpomReleaseComport;
begin
DeleteFile(LockfileName);
end;
{=====--//--//--//=====}
function GetSerialPortNames: string;
var
reg: TRegistry;
l, v: TStringList;
n: integer;
begin
l:= TStringList.Create;
v:= TStringList.Create;
reg:= TRegistry.Create;
try
reg.RootKey:= HKEY_LOCAL_MACHINE;
reg.OpenKey('HARDWARE\DEVICEMAP\SERIALCOMM', false);
reg.GetValueNames(l);
for n:= 0 to l.Count - 1 do
v.Add(reg.ReadString(l[n]));
Result:= v.CommaText;
finally
reg.Free;
l.Free;
v.Free;
end;
end;
end.
```

Кафедра КБПЗ – 2021 рік

Файл авторського права

```
unit About;
{Copyright © 2021 CNTU AZATIAN Z.R.}

interface
uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls, Buttons,
ExtCtrls;
type
  TAboutBox = class(TForm)
    Panel1: TPanel;
    ProgramIcon: TImage;
    ProductName: TLabel;
    Version: TLabel;
    Copyright: TLabel;
    Comments: TLabel;
    OKButton: TButton;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure OKButtonClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  AboutBox: TAboutBox;
implementation
uses IqrForm;
{$R *.dfm}
procedure TAboutBox.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action:=caNone;
end;
procedure TAboutBox.OKButtonClick(Sender: TObject);
begin
  MainForm.show;
  AboutBox.hide;
end;
end.
```