

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

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
« ____ » _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи цифрового
телебачення DVB-T2”

Виконав здобувач вищої освіти
II курсу, групи КІ-22М-1
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Мироненко О.І.
« ____ » _____ 2023 р.

Керівник проекту
доктор філософії (PhD)
_____ Усік П.С.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 "Інформаційні технології"
Спеціальність 123 "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2023 року

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

Мироненку Олександр Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи цифрового телебачення DVB-T2

2. Керівник роботи Усік Павло Сергійович, доктор філософії (PhD)

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 34-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту 10.12.2023 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи цифрового телебачення DVB-T2

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

6. 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.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 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	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2023 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Мироненко О.І. Дослідження та програмна реалізація системи цифрового телебачення DVB-T2. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи цифрового телебачення DVB-T2.

Метою розробки є дослідження та програмна реалізація системи цифрового телебачення DVB-T2.

Об'єктом дослідження є процес цифрового телебачення DVB-T2.

Предметом дослідження є методи цифрового телебачення DVB-T2.

Методи дослідження базуються на методах цифрового телебачення, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи цифрового телебачення DVB-T2.

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

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

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

Програму розроблено в середовищі Delphi 10.4.1.

Ключові слова: комп'ютерна інженерія, DVB-T2

ABSTRACT

Myronenko O.I. Research and software implementation of the DVB-T2 digital television system. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software designed for the DVB-T2 digital television system was developed.

The purpose of the development is research and software implementation of the DVB-T2 digital television system.

The object of research is the DVB-T2 digital television process.

The subject of the study is the methods of DVB-T2 digital television.

Research methods are based on digital television methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the DVB-T2 digital television system.

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

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

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

The program was developed in the Delphi 10.4.1 environment.

Keywords: computer engineering, DVB-T2

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	13
2.3 Розгорнута постановка завдання	19
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	21
3.1 Опис функціонування системи	21
3.2 Розробка структурної схеми.....	33
3.3 Розробка функціональної схеми	42
3.4 Розробка діаграми процесів.....	44
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	46
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	46
4.2 Захист розробленого програмного забезпечення.....	61
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	62
6 НАУКОВА НОВИЗНА	64

						ВКРМ-123.23.0016.00.00.ПЗ		
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Мироненко О.І.				Дослідження та програмна реалізація системи цифрового телебачення DVB-T2	Літ.	Аркуш	Аркушів
Перев.	Усік П.С.					М	1	105
Н.контр.	Коваленко А.С.				ЦНТУ КІ-22М-1			
Затв.	Смірнов О.А.							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	65
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	65
7.2 Розрахунок трудомісткості розробки програмної продукції.....	67
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	69
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	73
7.5 Визначення собівартості розробки та ціни програмної продукції.....	77
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	81
7.7 Визначення експлуатаційних витрат.....	81
7.8 Визначення економічної ефективності програмної продукції.....	83
7.9 Висновок.....	85
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	86
8.1 Вступ.....	86
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером	87
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	88
8.4 Розробка заходів з умов поліпшення охорони праці.....	91
8.5 Розрахункова частина	92
9 ОСНОВНІ ВИСНОВКИ.....	97
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	99

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

TV (Tele Vision) – телевізійна модуляція;
FM (Frequency Modulation) – частотна модуляція;
ОЗП – оперативний пристрій, що запам'ятовує;
ОС – операційна система;
ТЗ – технічна задача;
НС – надзвичайна ситуація;
ВДТ – відео дисплейний термінал;
BT-8x8 (BT-848, BT-878) – відео чіпсети TV-тюнерів;
JTV (Jet TV), Fly2000TV – програма перегляду TV, FM сигналу;
МП – мікропроцесор;
ВІС – великі інтегральні схеми;
САУ – система автоматичного управління;
ЕОМ – електронна обчислювальна машина;
Bt (Brooktree) – назва чіпсету;
CC (Closed Captioning) – закінчення захвату;
GPIO (General Purpose Input/Output) – головний інтерфейс вводу/виводу
GPO (GPIO Output Enable) – GPIO дозвіл виходу;
I2C Inter (Integrated Circuit Interface) – вбудований цикловий інтерфейс;
PAL (Phase Alternation Line) – Станція альтернативної лінії;
RGB (Red, Green, Blue) – червоний, зелений, синій;
SCL (Serial Clock) – серійні часи;
SECAM – інший ТБ формат відмінний від PAL.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Наземне телебачення DVB-T2 – це технологія, яка революціонізувала спосіб отримання та споживання телевізійного контенту.

DVB-T2 – це друге покоління стандарту DVB-T (цифрове відеомовлення – наземне), який використовується для наземного телевізійного мовлення. DVB-T2 пропонує значно покращену продуктивність та ефективність у порівнянні зі своїм попередником, забезпечуючи передачу більшої кількості каналів вищої якості на тому самому діапазоні частот.

DVB-T2 використовує передові методи кодування та модуляції, такі як HEVC (високоєфективне кодування відео) та OFDM (мультиплексування з ортогональним частотним поділом каналів), щоб забезпечити високу якість відео та аудіо, а також кращу стійкість до перешкод. Таким чином, навіть якщо ви живете в районі з поганим покриттям, ви все одно зможете насолоджуватися кристально чистим відео та аудіо.

DVB-T2 – це майбутнє ефірного телебачення з кількох причин. По-перше, він пропонує набагато кращу якість зображення та звуку порівняно зі стандартом DVB-T. Завдяки використанню кодування HEVC DVB-T2 здатний передавати відео Full HD і навіть 4K, чого неможливо досягти за допомогою стандарту DVB-T.

По-друге, DVB-T2 більш ефективний з точки зору використання діапазону частот. Завдяки вдосконаленим методам модуляції DVB-T2 може передавати більше даних на одному діапазоні, що означає, що телевізійні станції можуть пропонувати більше каналів і послуг.

По-третє, DVB-T2 є більш гнучким і перспективним. Стандарт здатний підтримувати різноманітні формати та стандарти, такі як IPTV, VoD (відео на вимогу) та інтерактивні послуги, відкриваючи двері до нових можливостей та послуг для глядачів.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи цифрового телебачення DVB-T2.

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

- Огляд існуючих систем цифрового телебачення DVB-T2.
- Дослідження системи цифрового телебачення DVB-T2.
- Програмна реалізація системи цифрового телебачення DVB-T2.

Об'єктом дослідження є процес цифрового телебачення DVB-T2.

Предметом дослідження є методи цифрового телебачення DVB-T2.

Методи дослідження базуються на методах цифрового телебачення, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод цифрового телебачення DVB-T2.
- Розроблено вітчизняний продукт цифрового телебачення DVB-T2, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі цифрового телебачення DVB-T2.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи цифрового телебачення DVB-T2, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для реалізації програмного забезпечення цифрового телебачення DVB-T2. Стандарт DVB-T2 – це друге покоління європейського стандарту ефірного цифрового віщання DVB-T. Він покликаний як мінімум на 30% поліпшити ємність телевізійних мереж у порівнянні з DVB-T при тій же інфраструктурі мережі й частотних ресурсів.

Серед переваг переходу на DVB-T2 можна виділити наступні:

- збільшення кількості каналів трансльованого пакета;
- можливість організації «місцевого» віщання;
- можливість розвитку телебачення високої чіткості;
- вивільнення ефірних частот.

Завдяки переходу на стандарт DVB-T2 з'явилася можливість розширити состав уже першого мультиплексу. Це означає, що й состав другого мультиплексу буде розширений. А виходить, телеглядачі можуть одержати 32 канали без абонентської плати вже в першому й другому мультиплексах.

Це дозволило приступитися до розгляду питання про те, щоб у третій мультиплекс цифрового ефірного віщання були включені канали у форматі ТВЧ (телебачення високої чіткості).

Крім того, застосування стандарту DVB-T2 в абонентських пристроях створює технологічну основу для надання через мережі цифрового ефірного телемовлення додаткових послуг, зокрема, доступу до державних послуг в електронному виді, оперативного оповіщення населення в надзвичайних ситуаціях і інших. У перспективі можливе впровадження нової інтерактивної технології HbbTV, завдяки якій можливості звичного телевізора впритул наблизяться до можливостей комп'ютера.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1.2 Область застосування

Областю застосування є мережа цифрового ефірного телемовлення.

Перехід на ефірне телебачення DVB-T2 має багато переваг. По-перше, ви отримаєте доступ до більшої кількості каналів вищої якості. DVB-T2 пропонує відео у форматі Full HD і навіть 4K, що є значним покращенням у порівнянні зі стандартною чіткістю, яку пропонує DVB-T.

По-друге, DVB-T2 більш стійкий до перешкод, що означає, що ви насолоджуватиметеся кращим прийомом, навіть якщо живете в районі з поганим покриттям.

По-третє, DVB-T2 пропонує багато додаткових функцій і послуг, таких як IPTV, VoD та інтерактивні служби, які можуть значно покращити ваші враження від перегляду телепередач.

Щоб використовувати ефірне телебачення DVB-T2, потрібен декодер DVB-T2. Цей декодер – це пристрій, який приймає сигнал DVB-T2 і перетворює його на відео та аудіо, які можна відтворювати на телевізорі.

У Deko Electronics ми пропонуємо широкий вибір декодерів DVB-T2, які прості у використанні та пропонують чудову якість зображення та звуку. Наші декодери, такі як декодер DVB-T2 DekoTV Pro HEVC H.265, декодер DVB-T2 DekoTV Pro2 2USB і декодер наземного телебачення DVB-T2 HEVC H.265. Mini оснащені найновішими технологіями, такими як кодування HEVC, модуляція OFDM і підтримка IPTV, що гарантує вам найкраще враження від перегляду телепередач.

Таким чином, ефірне телебачення DVB-T2 – це технологія майбутнього, яка пропонує кращу якість зображення та звуку, більше каналів і послуг, а також кращу стійкість до перешкод. Якщо ви хочете скористатися цими перевагами, вам слід розглянути можливість переходу на наземне телебачення DVB-T2. І якщо вам потрібен декодер DVB-T2, Deko Electronics тут, щоб допомогти.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи цифрового телебачення DVB-T2, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Згідно офіційної інформації, в Україну рекомендовано для перегляду цифрового ефірного ТБ дві моделі тюнерів:

- STRONG SRT 8500.
- Trimax TR2012-HD.

Обидві моделі тюнерів заявляють підтримку HD, обидві мають функції медіаплеєра, ціна приблизно однакова – в районі 600 грн.

Розглянемо тюнер STRONG SRT 8500. До речі, дисплей на передній панелі – це не дисплей, в усякому разі ніяких написів від нього домогтися не вдалося ні в якому режимі. Тільки зелений / червоний світлодіод (включено / виключено). На задній стінці тюнера – роз'єми.

Не дуже зрозуміло, чому USB зробили ззаду – флешку туди вставляти буде незручно. В іншому теж все очікувано: вхід антени, радіочастотний вихід тюнера, аудіо / відео тюльпани RCA, USB, HDMI і коаксіальний S / PDIF.

Отже, приєднати тюнер до ТБ можна аж трьома способами: HDMI, тюльпани RCA і через антенний вхід, тобто фактично тюнер можна підключити до будь-якого телевізора, в тому числі до дуже старим.

До слова, шнура HDMI і шнура для підключення до антенного входу ТБ в комплекті немає. Але якщо HDMI придбати немає проблем, то шнур для з'єднання радіочастотного виходу тюнера з антенним входом ТБ (якщо раптом знадобиться) доведеться або робити самостійно, або добре шукати. До речі, на гладкій поверхні тюнер тримається дивно добре, не совається. Ще один цікавий

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

недолік: ніякого управління, окрім кнопки включення, на плеєрі немає – тільки пультом. Якщо пульт зламався, сіли батареї – вибачте, ТБ сьогодні не для вас.

Підключення до старого телевізора і якість прийому

Для прийому ТБ я в селі використовую дуже популярну конструкцію – так звана «польська антена».

Умови прийому – жахливі: до базової станції передавальної антени ТБ-центру близько 30 кілометрів, прямої видимості немає – яма.

Прийом сильно залежить від погоди, часу доби і ще незрозуміло від чого. Ніякі експерименти по заміні антени, винесення її на щоглу, додаванню підсилювачів та інше практично нічого не дають.

Тепер пробуємо цю ж антену на цифровий тюнер: підключаємо антену до тюнера, тюнер через дзвіночки – до телевізора.

Після включення він приблизно хвилину завантажується, потім пропонує початкові налаштування і пошук каналів. Знайдено 32 канали, якість у всіх – ідеал.

З 32 каналів – 1 в тестовому режимі, 3 обласних, решта – загальноукраїнські. В процесі настройки було ясно, що канали йдуть на 4-х частотах ефіру (при покупці тюнера продавець назвав їх «транспондерами» – по аналогії з супутниковим ТБ).

HD каналів в ефірі поки що немає, але, за чутками, будуть.

Якість прийому дуже порадувало – незважаючи на явно неякісний прийом, дуже зрідка (1-2 рази на годину) на деяких каналах пропадало зображення і звук на дві-три секунди.

Після кількох днів перегляду з'ясувалося, що часом пропадає частина каналів – мабуть, погіршуються умови прийому на одному з транспондерів.

Варіант усунення цієї проблеми: замість ширококутної і незрозуміло якої якості «польської» антени зробити антену конкретно для прийому транспондерів, благо номери каналів відомі.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Перше загальне враження – дуже позитивне: просто, швидко, недорого, прекрасна якість прийому.

З додаткових переваг:

– Список каналів і швидкий вибір доступний з меню, це зручно.

– Гучність регулюється на пульті тюнера, пульт від ТБ потрібен фактично тільки для включення.

– Приймається додаткова інформація, деякі канали транслюють свою телепрограму.

Недолік:

– Канали перемикаються кілька секунд (мабуть, буферизація).

Після прочитання інструкції виявив ще один бонус: з тюнера можна взяти живлення антени прямо з гнізда, що я і зробив, від'єднавши роз'єм з живленням і прикривши на його місце звичайний:

Хоча – знову ж таки після кількох днів перегляду – я, напевно, зроблю живлення антени через зовнішній БП – на ньому можна регулювати напругу, часто це дозволяє поліпшити якість прийому.

Очікувався ще один бонус. На пульті є кнопка «Запис», продавець сказав, що можна записувати. Це – неправда, записувати ефір цим тюнером не можна, і чи буде ця функція в майбутніх прошивках – велике питання.

Підключення тюнера до моніторів і підтримувані дозволи

До монітора Dell U2312, який підтримує 1920×1080, підключення ніяких особливостей не має – встромили HDMI-шнур і все працює.

Взагалі тюнер підтримує такі варіанти розрішення: 1080i, 576p, 576p, 720p, 720i, 1080p

При першому включенні тюнер виставляє дозвіл 1080i, на пульті є кнопка, що дозволяє міняти дозвіл одним натисканням на неї – при кожному натисканні дозвіл змінюється в циклі по порядку, зазначеному вище: 1080i -> 576p -> 576i -> ...

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Теоретично цей ряд розрешеннів дозволяє підключити тюнер до будь-якого ТБ з HDMI-входом, практично до будь-якого монітора і без проблем домогтися зображення.

Проте змусити працювати тюнер з монітором Acer X203W (1680×1050) не вдалося – на всіх перемиканнях розрешеннів монітор намагається показувати шматок картинки, потім пише, що режим не підтримується.

Масштабування зображення

Тут є всі варіанти. Стандартний ефірний сигнал із співвідношенням сторін 4:3 можна показувати на моніторі 16:9 і розтягнутим, і масштабованим (як на фото вище), і з темними смугами зліва / справа.

Аналогічно фільм з флешки можна комфортно дивитися і на широкоформатному моніторі 16:9, і на старому телевізорі 4:3.

Тюнер як медіаплеєр

Відео

При навігації по файлах на носії виявилось, що є попередній перегляд.

Першим ділом я перевінив відтворення. Mkv – чудово грає 720p, зі звуком.

Виявився нюанс: зображення дуже сильно розтягується у висоту. Допомагає перемикання режиму екрану в 4:3.

Але далі захоплення кілька згасли – дуже багато. Avi йдуть без звуку. Причина начебто в тому, що виробником заявлена підтримка звуку тільки в mp3 і wma, а в дуже багатьох фільмах звукова доріжка у форматі aac.

До речі, виробник спочатку заклав можливість перепрошивки з USB-носія (є пункт прямо в меню).

Фото

Дивитися цілком можна, при навігації є мініатюри і попередній перегляд.

До речі, і мініатюри, і попередній, і перегляд самих фото прив'язані до встановленого режиму масштабування.

Кардрідер зі вставленою карткою від фотоапарата девайс розуміє без проблем, фотографії 8Мп показує.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Ніяких операцій із зображенням при перегляді – зум, поворот – зробити не можна. Якщо є повернені на 90 градусів зображення – насолоджуйтеся на боці, дорогі товариші. Навігація по файлах носія в будь-якому режимі – фото, відео – в тюнері зроблена не дуже зручно. Мультизадачності немає – тобто включити музику і запустити слайд-шоу не вийде (можливо, в нових прошивках буде можна). Витягування «на гарячу» USB-носія тюнер розуміє коректно, з носієм також нічого не відбувається.

Порівняння з супутниковим ТБ

Я у свій час використовував супутникову систему ТБ – дві тарілки, три супутника, два тюнера (на два ТБ). Рішення дорожче приблизно в 4 рази (за нинішніми цінами), і порівнювати супутникову систему з цим тюнером можна тільки за якістю зображення – воно однакове.

Але канали зовсім різні – на цьому тюнері, зрозуміло, немає українських каналів, а серед безкоштовних супутникових каналів зовсім мало українських.

Всього з усіх супутників можна було реально дивитися 30-40 українських та українських каналів (нагадаю – тюнер дає 32 каналу при нормальному прийомі всіх ефірних транспондерів). Є, звичайно, НТВ + з його пакетами, але це інша цінова категорія – і з підключення, і абонплата вельми відчутна.

Порівняння з IPTV

Якщо планується дивитися ТБ тільки на комп'ютері – тюнер вам не потрібен. В Інтернеті можна знайти плейлисти каналів і спокійно дивитися через Інтернет. Є варіанти, що працюють навіть зі швидкостями каналу менш 3 Мбіт/с.

IPTV через Інтернет показує дещо гірше (особливо потоки з невеликим бітрейтом), більше лага, але не вимагає ніяких зусиль і додаткових девайсів, + дозволяє дивитися у вікні.

Якщо в мережі провайдера є безкоштовне мовлення ТБ, то потрібно подумати, що краще, де більше каналів і пр.

Якість – однакова, найпростіша приставка до ТБ для прийому IPTV коштує приблизно стільки ж, скільки цей тюнер.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Порівняння з кабельним ТБ

За кількістю каналів – на кабельному, звичайно ж, їх більше в рази. Якість зображення тюнера – таке ж, як у цифрового кабельного (з чого б воно було іншим) і краще, ніж у аналогового кабельного.

Але врахуйте витрати: тюнер потрібен буде на кожен телевізор.

Загальне резюме тюнера

Якщо розглядати тюнер як:

- Пристрій, що підвищує якість прийому з поганою антеною і додає каналів: хороший, придатний девайс.
- Програваач кіно: не годиться (можливо, це часово, до виходу нових прошивок).
- Переглядач фото: 50/50.
- Заміна кабельного: не годиться.
- Засіб заощадити на платежах за кабельне ТБ: окупиться за рік-два.
- Заміна супутникового: 50/50, можливо як доповнення до нього.
- Замість IPTV: якщо на комп'ютері, то не годиться, якщо на ТБ: 50/50.

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.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

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

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

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

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

RAD Studio 10.4 Короткий огляд:

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

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

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom managed records. Більш швидке виконання паралельних завдань на сучасних

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

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-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

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 Snake.
- 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-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Змінені стилі VCL для High DPI

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

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

Нові High DPI стилі й стилізація окремих VCL компонент

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

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізовані 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 TМемо 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-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

КБГПЗ-2023

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

DVB-T2 – цифрова наземна система передачі, розроблена в рамках проекту DVB. Він представляє найновіші методи модуляції та кодування, щоб забезпечити високоефективне використання цінного наземного спектру для доставки аудіо-, відео- та послуг передачі даних на фіксовані, портативні та мобільні пристрої.

DVB-T2 використовує модуляцію OFDM із великою кількістю піднесучих, що забезпечує надійний сигнал. Як і DVB-T, DVB-T2 пропонує низку різних режимів, що робить його дуже гнучким стандартом. Для виконання виправлення помилок DVB-T2 використовує те саме кодування, яке було вибрано для DVB-S2. Поєднання кодування перевірки парності з низькою щільністю (LDPC) і кодування Бозе-Чаудхурі-Хоккенгама (BCH) забезпечує чудову продуктивність у середовищі з високим рівнем шуму та перешкод, створюючи дуже надійний сигнал.

Доступні кілька варіантів щодо таких аспектів, як кількість несучих, розмір захисного інтервалу та пілот-сигнали. Таким чином, накладні витрати можуть бути мінімізовані для будь-якого цільового каналу передачі. Нова техніка під назвою «повернені сузір'я» забезпечує значну додаткову надійність у складних каналах. Для досягнення необхідних умов прийому (наприклад, кімнатна антена/антена на даху) надається механізм окремого регулювання надійності кожної наданої послуги в межах каналу. Цей самий механізм дозволяє адаптувати передачі, щоб дозволити приймачу економити енергію, декодуючи лише одну програму, а не цілий комплекс програм.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Ключові характеристики DVB-T2:

- Щонайменше на 30-50% вища пропускна здатність і покращена продуктивність SFN порівняно з DVB-T.
- Надійність трансмісії залежно від послуги.
- Передача програм для мобільних і стаціонарних приймачів.
- Масове повторне використання інфраструктури DVB-T.
- Зменшення відношення пікової до середньої потужності і, отже, експлуатаційних витрат на кінці передавача.

Покращення, надані DVB-T2:

- Модуляція OFDM з додатковими режимами IFFT.
- Ефективний захист від помилок за допомогою кодування LDPC.
- Передача та інтеграція концепції кадру основної смуги DVB-S2.
- Обернені сузір'я із затримкою Q.
- Передача MISO.
- Зменшення крест-фактора.

DVB-T2 є наступним розвитком стандартів наземного цифрового відеомовлення. Він спирається на технологію та успіх DVB-T, щоб забезпечити додаткові можливості та функції відповідно до розвитку ринку DTT або цифрового наземного телебачення.

Хоча дехто може розглядати DVB-T2 як конкурента існуючому стандарту DVB-T, це не так. Планується, що два стандарти співіснують протягом багатьох років, а DVB-T2 надасть додаткові функції та послуги.

Основи DVB-T2

Стандарт DVB-T2 використовує мультиплекс з ортогональним частотним розподілом як основне середовище радіопередачі. Ця форма передачі є особливо надійною та дозволяє приймати сигнали даних (у цьому випадку телевізійні дані) за наявності певних перешкод або відсутності каналів у результаті таких ефектів, як багатопроменевість.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

що поєднання цих двох методів забезпечує чудову продуктивність за наявності високих рівнів шуму та перешкод.

Як і раніше, доступно кілька параметрів у таких областях, як кількість несучих, розміри захисного інтервалу та пілот-сигнали, так що накладні витрати можна мінімізувати для будь-якого даного каналу передачі.

Незважаючи на те, що DVB-T2 являє собою наступну еволюцію для цифрового наземного телебачення, планується, що він буде працювати поряд з поточним стандартом DVB-T протягом багатьох років і розвинути перехід на DVB-T2. Ця еволюція має відбуватися приблизно так само, як це відбулося між DVB-S і DVB-S2.

Оскільки DVB-T2 пропонує додаткові можливості, це дозволить мовникам пропонувати нові та захоплюючі послуги, щоб гарантувати, що вони зможуть утримати своїх глядачів. Спираючись на успіх існуючих цифрових телевізійних послуг, DVB-T2 неодмінно отримає значний рівень поширення протягом найближчих років.

Що таке OFDM: Ортогональне частотне мультиплексування

OFDM, мультиплексування з ортогональним частотним поділом використовує кілька близьких несучих, кожна з яких має низьку швидкість передачі даних для стійкого зв'язку.

Мультиплексування з ортогональним частотним поділом – це форма сигналу або модуляції, яка забезпечує деякі значні переваги для каналів передачі даних.

Відповідно, OFDM, ортогональне частотне мультиплексування, використовується для багатьох найновіших широкосмугових і високошвидкісних бездротових систем, включаючи Wi-Fi, стільниковий зв'язок і багато іншого.

Той факт, що OFDM використовує велику кількість несучих, кожна з яких передає дані з низькою швидкістю передачі даних, означає, що він дуже стійкий до вибіркового завмирання, перешкод і ефектів багатопроменевого поширення, а також забезпечує високий ступінь спектральної ефективності.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Ранні системи, що використовують OFDM, виявили, що обробка, необхідна для формату сигналу, була відносно високою, але з прогресом у технології OFDM створює невеликі проблеми з точки зору необхідної обробки.

Розвиток OFDM

Використання OFDM і модуляції з кількома несучими в цілому вийшло на перший план в останні роки, оскільки це забезпечує ідеальну платформу для бездротової передачі даних.

Однак концепція технології OFDM була вперше досліджена в 1960-х і 1970-х роках під час дослідження методів зменшення перешкод між близько розташованими каналами. На додаток до цих інших вимог, необхідних для досягнення безпомилкової передачі даних за наявності перешкод і умов вибіркового розповсюдження.

Спочатку використання OFDM вимагало великих рівнів обробки і, відповідно, не було життєздатним для загального використання.

Одними з перших систем, які прийняли OFDM, були цифрове мовлення – тут OFDM зміг забезпечити високонадійну форму транспортування даних за різними умовами шляху сигналу.

Одним із прикладів є цифрове радіо DAB, яке було представлено в Європі та інших країнах. Норвезька радіомовна корпорація NRK запустила першу послугу 1 червня 1995 року. OFDM також використовувався для цифрового телебачення.

Пізніше обчислювальна потужність зросла в результаті зростання рівнів інтеграції, що дозволило розглянути OFDM для систем мобільного зв'язку 4G, які почали розгортатися приблизно з 2009 року. Також OFDM було прийнято для Wi-Fi та низки інших бездротових систем передачі даних.

Що таке OFDM?

OFDM є формою модуляції з кількома несучими. Сигнал OFDM складається з ряду близько розташованих модульованих несучих. Коли модуляція будь-якої форми – голосу, даних тощо застосовується до несучої, тоді бічні смуги

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

розповсюджуються в обидві сторони.

Необхідно, щоб приймач міг приймати весь сигнал, щоб мати можливість успішно демодулювати дані. У результаті, коли сигнали передаються близько один до одного, вони повинні бути розташовані так, щоб приймач міг розділити їх за допомогою фільтра, і між ними повинна бути захисна смуга. З OFDM це не так.

Незважаючи на те, що бічні смуги від кожної несучої перекриваються, вони все одно можуть прийматися без перешкод, які можна очікувати, оскільки вони ортогональні одна одній. Це досягається тим, що інтервал між несучими дорівнює зворотній величині періоду символу.

Традиційна вибірка сигналів if на різних каналах

Щоб побачити, як працює OFDM, необхідно подивитися на приймач. Це діє як банк демодуляторів, перетворюючи кожен несучу на DC. Результуючий сигнал інтегрується по періоду символу для регенерації даних із цього носія. Цей же демодулятор також демодулює інші несучі.

Оскільки інтервал між несучими дорівнює зворотній величині періоду символу, це означає, що вони матимуть цілу кількість циклів у періоді символу, а їхній внесок дорівнюватиме нулю – іншими словами, внеску перешкод немає.

Основна концепція OFDM, ортогонального частотного мультиплексування

Однією з вимог систем передачі та прийому OFDM є те, що вони повинні бути лінійними. Будь-яка нелінійність спричинить інтерференцію між несучими в результаті інтермодуляційних спотворень. Це призведе до появи небажаних сигналів, які можуть спричинити перешкоди та погіршити ортогональність передачі.

Що стосується обладнання, яке буде використовуватися, високе відношення піку до середнього в системах з декількома несучими, такими як OFDM, вимагає кінцевого РЧ-підсилювача на виході передавача, щоб мати можливість впоратися з піками, тоді як середня потужність набагато нижча, і це

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

призводить до до неефективності.

У деяких системах піки обмежені. Хоча це вносить спотворення, що призводить до більш високого рівня помилок даних, система може покладатися на виправлення помилок для їх усунення.

Дані з OFDM

Традиційний формат для надсилання даних через радіоканал полягає в тому, щоб надсилати їх послідовно, один біт за іншим. Це залежить від одного каналу, і будь-які перешкоди на одній частоті можуть порушити всю передачу.

OFDM використовує інший підхід. Дані передаються паралельно через різні несучі в межах загального сигналу OFDM. Будучи поділеним на кілька паралельних «підпотоків», загальна швидкість передачі даних дорівнює вихідному потоку, але для кожного з підпотоків набагато нижча, а символи рознесені далі один від одного в часі.

Це зменшує перешкоди між символами та полегшує точне отримання кожного символу, зберігаючи однакову пропускну здатність.

Нижча швидкість передачі даних у кожному потоці означає, що перешкоди від відображень набагато менш критичні. Це досягається шляхом додавання в систему часу захисної смуги або захисного інтервалу.

Захисний час забезпечує вибірку даних лише тоді, коли сигнал стабільний і не надходять нові затримані сигнали, які можуть змінити час і фазу сигналу. Цього можна досягти набагато ефективніше в субпоточі з низькою швидкістю передачі даних.

Захисний інтервал для сигналів OFDM

Розподіл даних через велику кількість несучих у сигналі OFDM має деякі додаткові переваги. Нульові сигнали, викликані ефектами багатопроменевого поширення або перешкодами на даній частоті, впливають лише на невелику кількість несучих, решта приймаються правильно. Використовуючи методи кодування помилок, що означає додавання додаткових даних до переданого сигналу, це дає змогу відновити багато або всі пошкоджені дані в приймачі. Це

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

можна зробити, оскільки код виправлення помилок передається в іншій частині сигналу.

Ключові особливості OFDM

Схема OFDM відрізняється від традиційної FDM наступними взаємопов'язаними способами:

- Кілька несучих (так звані піднесучі) передають потік інформації.
- Піднесучі ортогональні одна одній.
- До кожного символу додається захисний інтервал, щоб мінімізувати розкид затримки каналу та міжсимвольні перешкоди.

Переваги та недоліки OFDM

Переваги OFDM

OFDM використовувався в багатьох бездротових системах з високою швидкістю передачі даних через численні переваги, які він надає.

- **Стійкість до вибіркового завмирання:** одна з головних переваг OFDM полягає в тому, що він більш стійкий до частотно-селективного завмирання, ніж системи з однією несучою, оскільки він розділяє загальний канал на кілька вузькосмугових сигналів, на які впливають індивідуально як на підканали з плоским завмиранням.

- **Стійкість до перешкод:** перешкоди, що з'являються на каналі, можуть бути обмежені смугою пропускання, і таким чином не впливатимуть на всі підканали. Це означає, що не всі дані втрачаються.

- **Ефективність використання спектру.** Значною перевагою OFDM є те, що він ефективно використовує доступний спектр.

- **Стійкість до ISI:** ще одна перевага OFDM полягає в тому, що він дуже стійкий до міжсимвольних і міжкадрових перешкод. Це є результатом низької швидкості передачі даних на кожному з підканалів.

- **Стійкість до вузькосмугових ефектів:** використовуючи адекватне каналне кодування та чергування, можна відновити символи, втрачені через частотну вибіркковість каналу та вузькосмугові перешкоди. Не всі дані

						ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			28

Схема, яка використовується для надання можливості множинного доступу для таких програм, як стільниковий зв'язок, коли використовуються технології OFDM.

– VOFDM: векторний OFDM. Ця форма OFDM використовує концепцію технології MIMO. Він розробляється компанією CISCO Systems. MIMO розшифровується як Multiple Input Multiple output і використовує кілька антен для передачі та прийому сигналів, щоб можна було використовувати ефекти багатопроменевості для покращення прийому сигналу та покращення підтримуваної швидкості передачі.

– WOFDM: широкосмуговий OFDM. Концепція цієї форми OFDM полягає в тому, що вона використовує достатньо великий відстань між каналами, щоб будь-які частотні помилки між передавачем і приймачем не впливали на продуктивність. Особливо це стосується систем Wi-Fi.

Кожна з цих форм OFDM використовує ту саму основну концепцію використання близько розташованих ортогональних несучих, кожна з яких передає сигнали з низькою швидкістю передачі даних. Під час фази демодуляції дані потім об'єднуються для отримання повного сигналу.

OFDM, мультиплексування з ортогональним частотним поділом, набуло значної присутності на ринку бездротового зв'язку. Поєднання високої ємності даних, високої спектральної ефективності та його стійкості до перешкод у результаті багатопроменевого ефекту означає, що він ідеально підходить для додатків з великим об'ємом даних, які стали головним чинником сучасної комунікаційної сцени.

OFDM Синхронізація часу та частоти

Для оптимальної демодуляції OFDM потрібна точна синхронізація часу та частоти, інакше рівень перешкод зростає та виникають інші проблеми з демодуляцією.

Однією з ключових вимог для оптимальної демодуляції сигналів OFDM є точна синхронізація частоти та часу.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

На щастя, хорошу синхронізацію OFDM і частоту відносно легко отримати, оскільки легко вивести точні сигнали.

Погана синхронізація часу та частоти призводить до значного погіршення якості сигналу, і в результаті цей аспект сигнального ланцюга є ключовим для оптимальної продуктивності.

Основи синхронізації OFDM

OFDM пропонує багато переваг щодо стійкості до вицвітання, віддзеркалень тощо. OFDM також пропонує високий рівень ефективності спектру. Однак, щоб отримати винагороду, необхідно, щоб система OFDM працювала правильно, і для досягнення цього необхідно, щоб синхронізація OFDM була ефективною.

Існує ряд областей, у яких синхронізація OFDM критична для роботи системи:

- Синхронізація OFDM з точки зору зміщення частоти: необхідно, щоб частоти точно відстежувалися, щоб забезпечити підтримку ортогональності.
- Синхронізація OFDM з точки зору точності годин: необхідно, щоб вибірка відбувалася в правильний інтервал часу, щоб забезпечити синхронізацію вибірок і мінімізацію помилок даних.

Для того, щоб забезпечити оптимальну роботу системи OFDM, необхідно переконатися, що існують схеми, які гарантують, що синхронізація OFDM знаходиться в необхідних межах.

Синхронізація зміщення частоти OFDM

Особливо важливо, щоб демодулятор в приймачі OFDM міг точно синхронізуватися з несучими в сигналі OFDM. Зміщення можуть виникати з ряду причин, включаючи будь-які частотні помилки між передавачем і приймачем, а також в результаті доплерівських зсувів, якщо є рух між передавачем і приймачем.

Якщо частотна синхронізація порушується, то ортогональність несучих знижується в процесі демодуляції, а частота помилок зростає. Відповідно,

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

важливо підтримувати ортогональність, щоб зменшити кількість помилок і підтримувати ефективність зв'язку.

Спочатку подивіться, як має відбуватися вибірка. Коли демодулятор синхронізований, усі внески від інших несучих сумарно дорівнюють нулю, як показано. Таким чином, усі носії ортогональні, а частота помилок мінімальна.

Демодуляція OFDM, яка показує, як мінімізуються перешкоди від інших несучих в центрі несучої частоти

Якщо виникає ситуація, коли синхронізація OFDM для частотних аспектів є поганою, тоді демодулятор центруватиме свої зразки подалі від піку сигналу, а також у точці, де внески інших сигналів не дорівнюють нулю. Це призведе до погіршення якості сигналу, що, у свою чергу, може призвести до збільшення кількості бітових помилок.

Демодуляція OFDM, де перешкоди високі через погану частотну синхронізацію

Синхронізація зміщення годинника OFDM

Також необхідно підтримувати синхронізацію OFDM з точки зору годинника. Посилення, якщо синхронізація годинника є неточною, дискретизація буде зміщена, і знову ортогональність буде зменшена, а помилки даних зростуть.

При розгляді синхронізації OFDM з огляду на зсув тактової частоти інтервал несучої, що використовується в приймачі для дискретизації отриманого сигналу, базуватиметься на внутрішній тактовій частоті. Якщо це відрізняється від того, що використовується в передавачі, буде виявлено, що навіть якщо перша несуча в мультиплексі є правильною, тоді буде зростати розбіжність з кожною несучою, віддаленою від першої. Навіть невеликі рівні розбіжностей призведуть до збільшення рівня помилок.

Демодуляція OFDM із зміщенням синхронізації годинника

При використанні OFDM необхідно переконатися, що синхронізація як для часу, так і для частоти є точною. Забезпечуючи точну синхронізацію, можна виконати оптимальну демодуляцію сигналу. Будь-яке зміщення призводить до

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

того, що приймач починає вловлювати небажані сигнали перешкод. На щастя, відносно легко отримати точні сигнали синхронізації, оскільки вони доступні з мережі, а короткочасну синхронізацію можна створити всередині.

3.2 Розробка структурної схеми

Розробники заклали як мінімум 30-відсоткове поліпшення всіх значимих показників щодо попередньої версії стандарту – DVB-T.

Окрім нових застосовуваних технологій для передачі інформації в новій версії стандарту існують і деякі відмінності в організації мереж синхронного одночастотного мовлення (SFN).

У стандарті DVB-T2 є два режими роботи: так звані System A і System B. Перший відрізняється від DVB-T використанням більш високих порядків модуляцій і додаткових нових алгоритмів корекції помилок.

Крім того, даний режим не передбачає роботу мережі в SFN-режимі (окремий випадок – робота в System B з одним PLP).

System B є більш складним режимом роботи DVB-T2 передавачів. Його реалізація вимагає використання більш складних технологій і пристроїв (multi-PLP потоків, T2-Gateway).

Саме він передбачає створення SFN-мереж. Специфікою є і те, що для організації SFN-мереж вже не потрібно такий спеціальний пристрій, як SFN-адаптер (MIP-інсертер).

Його функції включає в себе DVB-T2 Gateway – пристрій, на вхід якого подається(-ються) MPEG-2 TS, а на вихідному інтерфейсі T2 Modulator Interface (T2-MI) формується вихідний multi-PLP потік.

T2-MI – це інтерфейс, призначений для передачі інформації від T2 Gateway до модуляторам. Він розроблений спеціально для того, щоб модулятор міг працювати з технологією multi-PLP. T2-MI-пакети містять в собі транспортні потоки і всю необхідну інформацію для роботи модуляторів.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

У певних місцях (L1 signalling) T2-МІ-пакетів міститься вся необхідна інформація для передавачів по режимам і параметрам передачі, причому параметри можна вказувати для кожного передавача окремо.

Наприклад, передавати значення тимчасової затримки прямо в потоці, змінюючи їх на «льоту».

Генерація потоку T2-МІ відбувається наступним чином: кожен раз, коли розташування ВВ-кадру (BaseBand Frame) визначено, він може бути вставлений в T2-МІ-пакет з відповідною інформацією в заголовку і негайно відправлений через T2-МІ інтерфейс.

ВВ-Frame – це основна одиниця в логічній кадровій структурі DVB-T2. Пакетовані потоки розбиваються на ВВ-кадри (синхронно або асинхронно).

Кожен ВВ-кадр може містити ціле число пакетів, або ж пакет може бути розбитий на кілька ВВ-кадрів. Тема ВВ-кадру містить інформацію про довжину пакету і позиції першого пакету, що дозволяє реконструювати їх на приймальному кінці.

Розмір кожного ВВ-кадру, що містить будь наповнення та/або внутрішню сигналізацію, постійний для даного PLP (Physical Layer Pipe – технологія, що дозволяє одночасно передавати кілька видів інформації з окремими для кожного параметрами перешкодозахищеності і швидкості), залежить від кодової швидкості LDPC і від довжини використаних FEC-блоків.

Як тільки розташування ВВ-кадрів визначено для всіх PLP, чий кадр перемеження (Interleaving Frames) вставлені в даний T2-кадр, в пакеті генерується і передається T2-МІ інформація, що описує цей кадр L1-сигналізація.

Там же міститься й інформація для синхронізації. Якщо синхронізація не потрібна (для MFN), у відповідних осередках передаються нулі.

У DVB-T2 збільшене число піднесущих (32К проти 8к), що дозволяє продовжити захисний інтервал до 532 Мкс (тоді як в DVB-T – 224 Мкс).

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Відповідно, збільшується і максимальна відстань між сусідніми передавачами в SFN. Теоретично воно дорівнює приблизно 160 км проти 67 у попередньому стандарті.

Також немає вже максимально допустимої затримки в поширенні сигналу в мережі передавачів в 1с через прив'язку до сигналу в 1 PPS.

Вся система може синхронізуватися з глобального часу (GMT), і затримка може бути більше.

У той же час збільшення числа піднесучих до 32К призводить до зменшення «відстані» між ними, а це, у свою чергу, висуває досить високі вимоги до стабільності частоти передавача при роботі в SFN-режимі в System B – до 0,5 Гц.

Додатково стандарт DVB-T2 при роботі в System B, завдяки використанню системи MISO (яка, у свою чергу, використовує коди Alamauti), дозволяє ефективно боротися з завмираннями сигналу в SFN-мережах (вхідний сигнал на двох сусідніх передавачах попередньо спотворюється таким чином, що в точці прийому без праці можна відновлювати вихідний сигнал, а умови прийому поліпшуються).

Коротко підсумуємо відмінності в організації SFN-мереж в новому стандарті від попереднього.

1. Відпала необхідність у SFN-адаптер. Його функції вже закладені в T2 Gateway.
2. Можлива затримка в мережі стала більше секунди.
3. Максимально допустима відстань між сусідніми передавачами зросла з 67 до 160 км.
4. Зросли вимоги до стабільності частоти сигналу передавачів: $\pm 0,5$ Гц.
5. Завдяки системі MISO поліпшуються умови прийому в зонах з однаковими відстанями до сусідніх передавачів, що зумовлює виникнення селективних за частотою завмирань.

						ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			35

Таким чином, використовуючи перераховані вище особливості, можна створювати більш ефективні SFN-мережі і використовувати їх більш гнучко.

Структурна схема розробленої системи зображена на рисунку 3.1. На ній показано структуру системи.

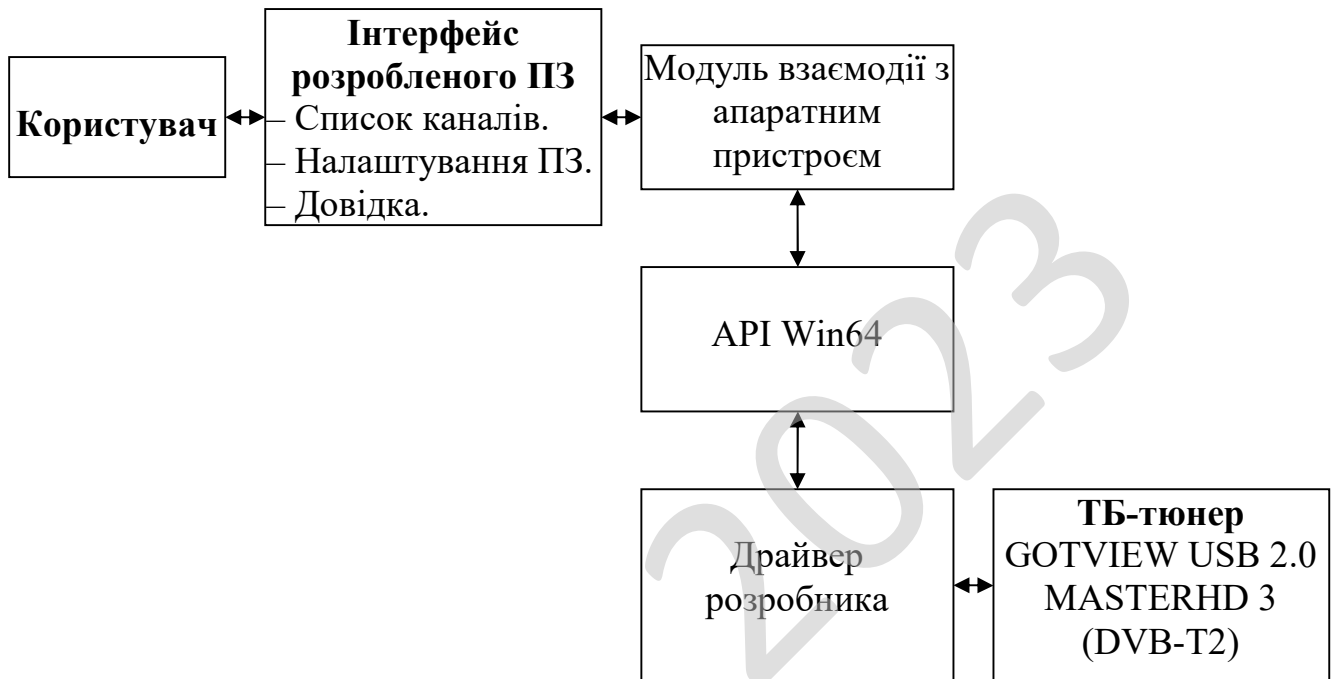


Рисунок 3.1 – Структурна схема роботи системи

Користувач використовує розроблене ПЗ під керуванням ОС Windows 8 через інтерфейс розробленого ПЗ. Розроблена ПЗ має список каналів, налаштування ПЗ та довідкову інформацію.

Через модуль взаємодії з апаратним пристроєм (Application programming interface) а саме підсистему Digital Video Broadcasting Second Generation Terrestrial (DVB-T2) та драйвер розробника проходить взаємодія з ТБ-тюнером GOTVIEW USB 2.0 MASTERHD 3.

GOTVIEW USB 2.0 MASTERHD

ТБ-тюнер USB 2.0 представляє собою рішення по прийому й запису відкритих цифрових теле- і радіоканалів DVB-T2, DVB-T і DVB-C стандартів, включаючи FULL HD канали. GOTVIEW USB 2.0 MASTERHD 3 є одним з

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

перших тюнерів в Україні з одночасною підтримкою цих трьох стандартів віщання. Тюнер виконаний у звичному дизайні USB стик GOTVIEW, аналогічному моделям GOTVIEW USB 2.0 Hybrid MasterHD і GOTVIEW USB 2.0 Master, але його технічна начинка більше функціональна й унікальна. Тюнер дозволяє приймати ефірне цифрове DVB-T2, DVB-T, цифрове, кабельне ТБ DVB-C.

Наявність автодетекта відеостандарту в тюнері Gotview USB 2.0 MasterHD з необхідно для використання тюнера в таких програмах, як Windows Media Center, які не можуть самі перемикаєти стандарти. Будь-який канал можна настроїти індивідуально.

Важливо, що для стандарту DVB-T2 реалізована підтримка однопотокowego (single) PLP і багатопоточного (multiple) PLP режимів віщання, що дозволить стабільно приймати DVB-T2 сигнал у всіх регіонах України.

У багатьох регіонах України вже запущене ефірне цифрове віщання в DVB-T2.

Якщо цікаві передачі йдуть одночасно на різних каналах і хочеться подивитися їх всі, то й це не проблема. Можна записати весь транспортний потік і потім дивитися той канал, на якому віщалась цікава передача. Досить у плеєрі вибрати потрібний ідентифікатор каналу.

Як селектор каналів використаний новий напівпровідниковий чипсет компанії Silabs Si2148.

На ринку ТБ тюнерів існує тверда конкуренція між SiLabs, NXP, MaxLinear і SONY. Ключовими особливостями селектора SiLabs є те, що він не вимагають зовнішнього балансування, котушок індуктивності в ланцюгах відстеження сигналу й у фільтруючих вузлах блока живлення.

Продуктивність даного селектора перевищує аналогічні характеристики тюнерів на основі MOPLL (гетеродин з ФАПЧ).

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

транспортного потоку в ISO/IEC 13818-1 Transport Stream, для одночасного запису всіх каналів, трансльованих на даній частоті.

9. Повністю русифікований інтерфейс.

10. Індивідуальні налаштування для кожного каналу: гучність, яскравість, контрастність, насиченість.

11. Функція відкладеного перегляду або зрушення за часом (TimeShift) з можливістю прискореного й уповільненого перегляду.

12. Можливість налаштування до 255 ТВ каналів.

13. Наявність вибору дозволу для попереднього й повноекраного режимів перегляду.

14. Захват кадрів ТВ вікна у форматі JPG по таймеру або по натисканню кнопки.

15. Опціональне відключення звуку під час запуску програми, пошуку або перемикання каналів.

16. Захват окремих кадрів.

17. Диспетчер відкладених завдань для запису відео в заданий час.

18. Гнучке налаштування захвата.

19. Відеозахват з автоматичним формуванням файлів заданого розміру (наприклад, по 700 Мб для подальшого запису на CD-DVD диски).

20. Можливість автоматичної зміни імені файлу.

21. Сканування каналів у будь-якому зазначеному діапазоні, з будь-яким заданим кроком.

22. Повна інформація про захват:

- розмір і тривалість захопленого фрагмента;
- вільне місце на диску;
- час можливого запису, виходячи з наявного вільного простору на диску;
- кількість захоплених і пропущених кадрів;
- частота кадрів;

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

на парність (LDPC) і кодування Боуза-Чоудхурі-Хоквінгема (BCH) забезпечує дуже стійкий сигнал і чудову якість в умовах з високим рівнем шумів і перешкод.

Є декілька опцій таких параметрів, як кількість несучих, тривалість захисного інтервалу та розміщення пілот-сигналів. Це дозволяє знизити до мінімуму частку службової інформації для будь-якого заданого каналу передачі. Новий метод, названий «поворот сигнального сузір'я» забезпечує істотний приріст стійкості у складних ефірних умовах. Для забезпечення необхідних умов прийому (наприклад, кімнатна антена / антена на даху), передбачено механізм роздільної настройки стійкості сигналу в межах каналу для кожної надаваної служби. Такий самий механізм дозволяє настроїти передачі так, щоб дати можливість приймачу економити енергію за допомогою декодування тільки однієї програми, а не всього пакета програм.

Ключові особливості характеристик DVB-T2:

- Збільшена не менше ніж на 30% пропускна здатність і покращені характеристики SFN, порівняно з DVB-T.
- Обумовлена стійкість передачі.
- Передача програм на мобільні та стаціонарні приймачі.
- Широке використання інфраструктури DVB-T.
- Зниження експлуатаційних витрат на стороні передачі за рахунок зменшення відношення пікової потужності до середньої потужності.

Покращення, передбачені DVB-T2:

- Модуляція OFDM з додатковими режимами IFFT.
- Кодування LDPC забезпечує ефективний захист від помилок.
- Використання та інтеграція базової структури кадру DVB-S2.
- Поворот сигнального сузір'я з Q-затримкою.
- Передача MISO.
- Зменшення пік-фактора.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

3.3 Розробка функціональної схеми

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

Функціонально система складається з наступних блоків:

- Блок інтерфейсу користувача.
- Блок роботи з DVB-T2, H.264 і DVB-S2, тобто можна дивитися HD-канали.

- Блок перемикання каналів.

- Планувальник завдань для запису за розкладом.

- Блок підтримки телетексту й множинних звукових доріжок.

- Блок підтримки API плагинів.

- Блок підтримки MPEG і TS форматів.

- Блок підтримки інтегрованої функції EPG.

- Функція зрушення в часі (timeshifting).

- Блок роботи пультами winlirc.

EPG

Electronic Program Guide (EPG) або Electronic Service Guide (ESG) – інтерактивна послуга в області цифрового телебачення або радіомовлення, що забезпечує гнучкість у керуванні цифровим контентом.

Електронний телегід (EPG) являє собою екранне меню, що відображає розклад теле– або радіопрограм з можливістю інтерактивної навігації контенту за часом, назві, каналу, жанру й т.д. за допомогою пульта дистанційного керування.

Електронний телегід (EPG) є у всіх ресиверах цифрового телебачення, а також у всіх сучасних телевізорах що мають убудований тюнер (декодер) цифрового телебачення.

Можливі функції EPG: вибір контенту по жанрах, напрямкам, алфавіту, рейтингу та ін.; відображення рівнів доступу до розділів або каналів, паролі або політика «батьківського контролю» (Parental Control); надання інтерфейсу для

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

доступу до сервісу PVR/DVR (відеозапис) і TimeShift (зрушення в часі) для запису по заданих інтервалах часу обраних ТБ каналів через EPG, редагування часових інтервалів перегляду; дозволяє користувачеві встановлювати нагадування про передачу або про подію; здійснює попередній перегляд ТБ каналів прийнятих при адресному розсиланні у вигляді виводу відео в розмірі 1/8 або 1/16 екрана зі звуком або без нього; навігація по розділах меню для одержання розширеної інформації про телепередачу, фільм, радіопередачу; дозволяє виводити екранну клавіатуру й застосовувати її для набору символів; вибір формату телемовлення, якщо він доступний (SD та HD відео, бітрейт для аудіо); вивід списку доступного відеоконтенту для сервісів VoD (відео за запитом).

Інформація, що доставляється електронному телегiдовi може бути зiбрана прийомним устаткуванням з кожного переданого каналу або передаватися в окремому каналi. Для полегшення першого методу Європейським Телекомунікаційним Інститутом Стандартів (ETSI) був виданий стандарт ETS 300 707.

H.264

H.264/MPEG-4 AVC – міжнародний стандарт відеокомпресії. Стандарт було розроблено групою фахівців організації ITU (Study Group 16, фахівці з відео-кодування) під назвою H.26L. У 2001 році група ITU об'єдналась з MPEG-Visual та розробку стандарту було продовжено в команді Joint Video Team (JVT). Метою проекту була розробка методів стиснення відео-даних, що гарантували б, у порівнянні з існуючими, щонайменше вдвічі менший бітрейт зі збереженням рівня якості як для мобільних приладів, так і для телебачення. У 2003 році обидві організації затвердили стандарт. Назва стандарту ITU: H.264. Стандарт ISO/IEC має назву MPEG-4/AVC (Advanced Video Coding) і є десятою частиною стандарту MPEG-4 (MPEG-4/Part 10, ISO/IEC 14496-10). Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

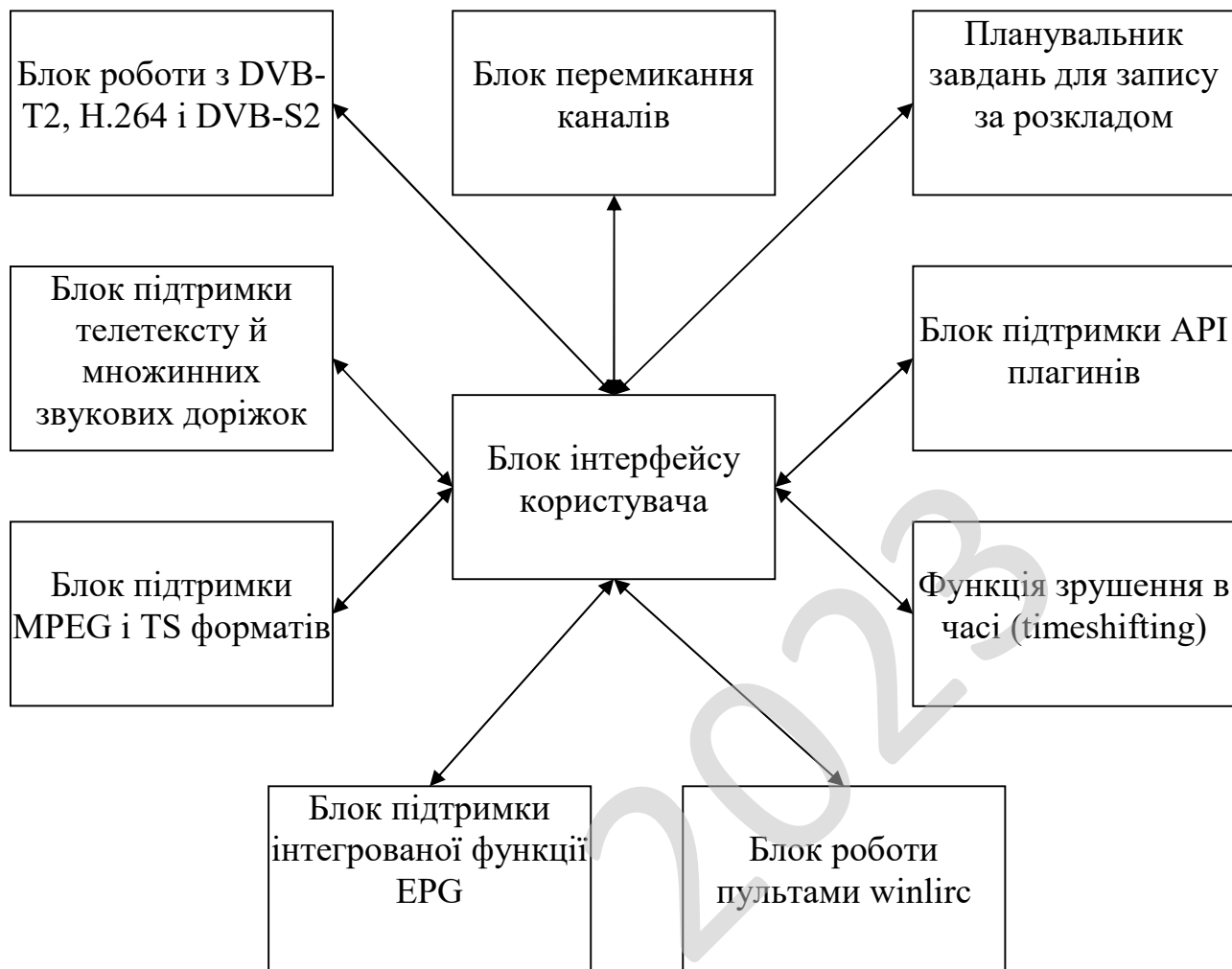


Рисунок 3.2 – Функціональна схема системи

3.4 Розробка діаграми процесів

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

Після початку роботи розробленого ПЗ ми потрапляємо до інтерфейсу ПЗ звідки може перейти до довідки ПЗ, налаштування ПЗ, списку каналів, бібліотеки DVB-T2 та через обробник помилок потрапити до підключення до апаратного пристрою, аналізу відеопотоку та до кінцевого отримання відеопотоку стандарту DVB-T2.

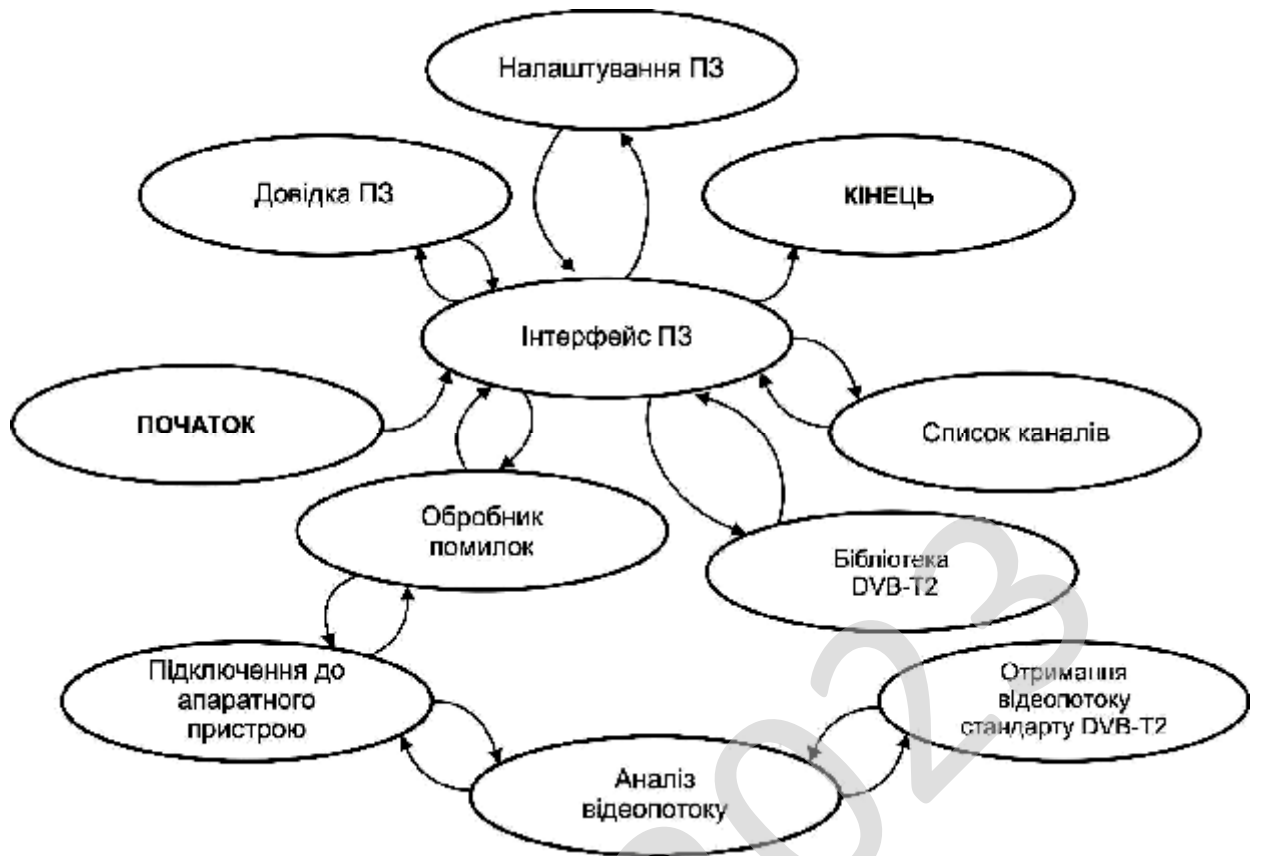


Рисунок 3.3 – Діаграма взаємодії процесів

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

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

- Ініціалізація ПЗ та підключення бібліотеки DVB-T2.
- Підключення бібліотеки обробки відеопотоку, спроба доступу до файлу ресурсів.

- Є доступ до файлу ресурсів?
- Спроба доступу до ТВ-тюнера.
- Отримання характеристик ТВ-тюнера.
- Є доступ до ТВ-тюнера ?
- Перевірка підтримки протоколу DVB-T2.
- Плата підтримує протокол?
- Початкове налаштування пакету DVB-T2.
- Виведення вікна перегляду відеопотоку.
- Очікування запиту користувача.
- Запит WM_CLOSE?
- Є дія?
- Запит перегляд відеопотоку?
- Підпрограма перегляду відеопотоку.
- Виведення потоку трансляції.
- Запит – перемикання каналу?
- Відправлення запиту перемикання каналу.
- Запит WM_CLOSE?

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

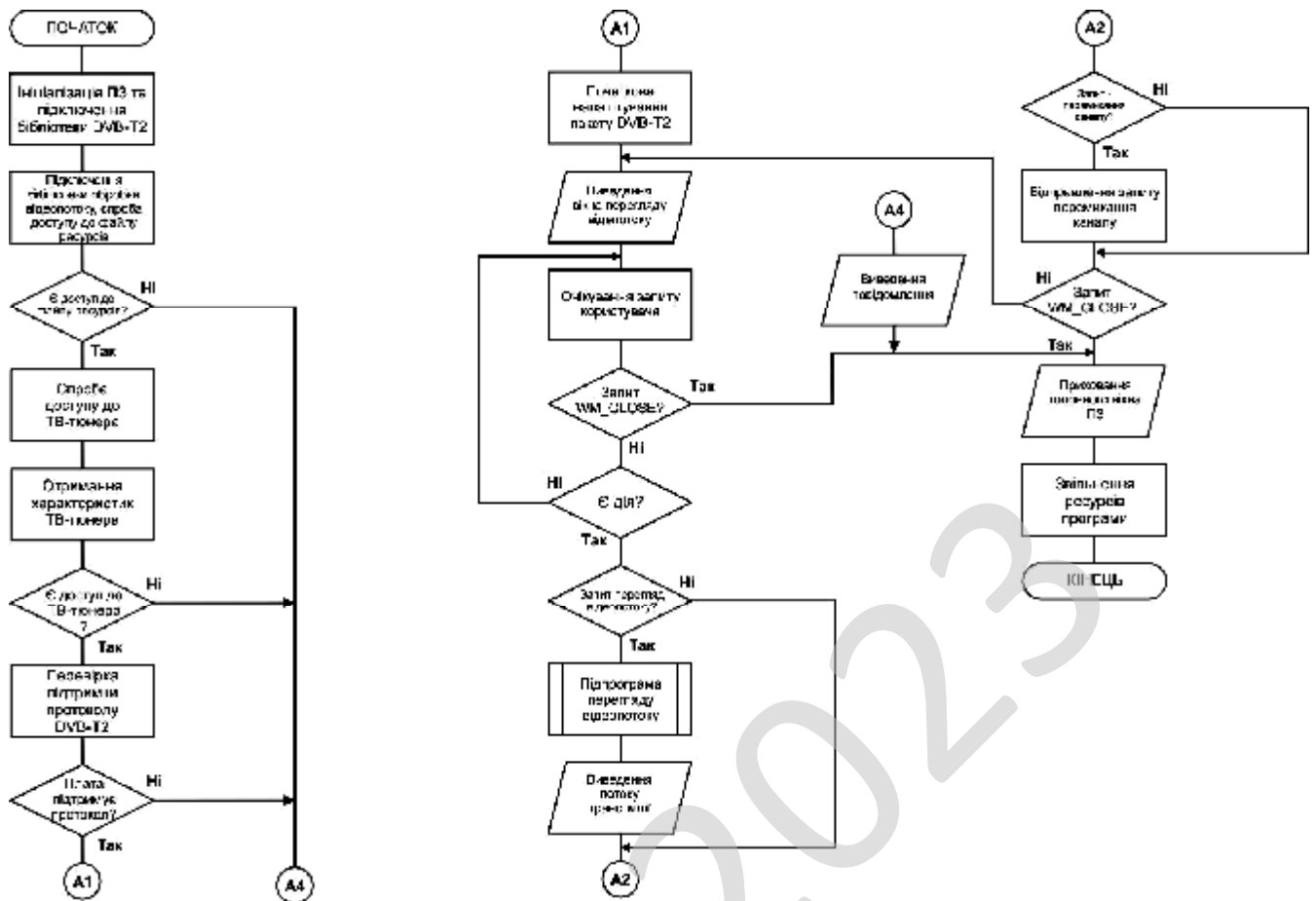


Рисунок 4.1 – Блок-схема основної програми

- Приховання головного вікна ПЗ.
- Звільнення ресурсів програми.

На рисунку 4.2 наведено блок-схему підпрограми перегляду відеопотоку. Її робота складається з виконання наступних кроків.

- Запит масиву каналів.
- Масив змінено?
- Створення та формування списку каналів.
- Оновлення списку каналів.
- Запит потоку DVB-T2.
- Перевірка структури T2-кадру.
- Спроба доступу до потоку DVB-T2.
- Є доступ до потоку DVB-T2?

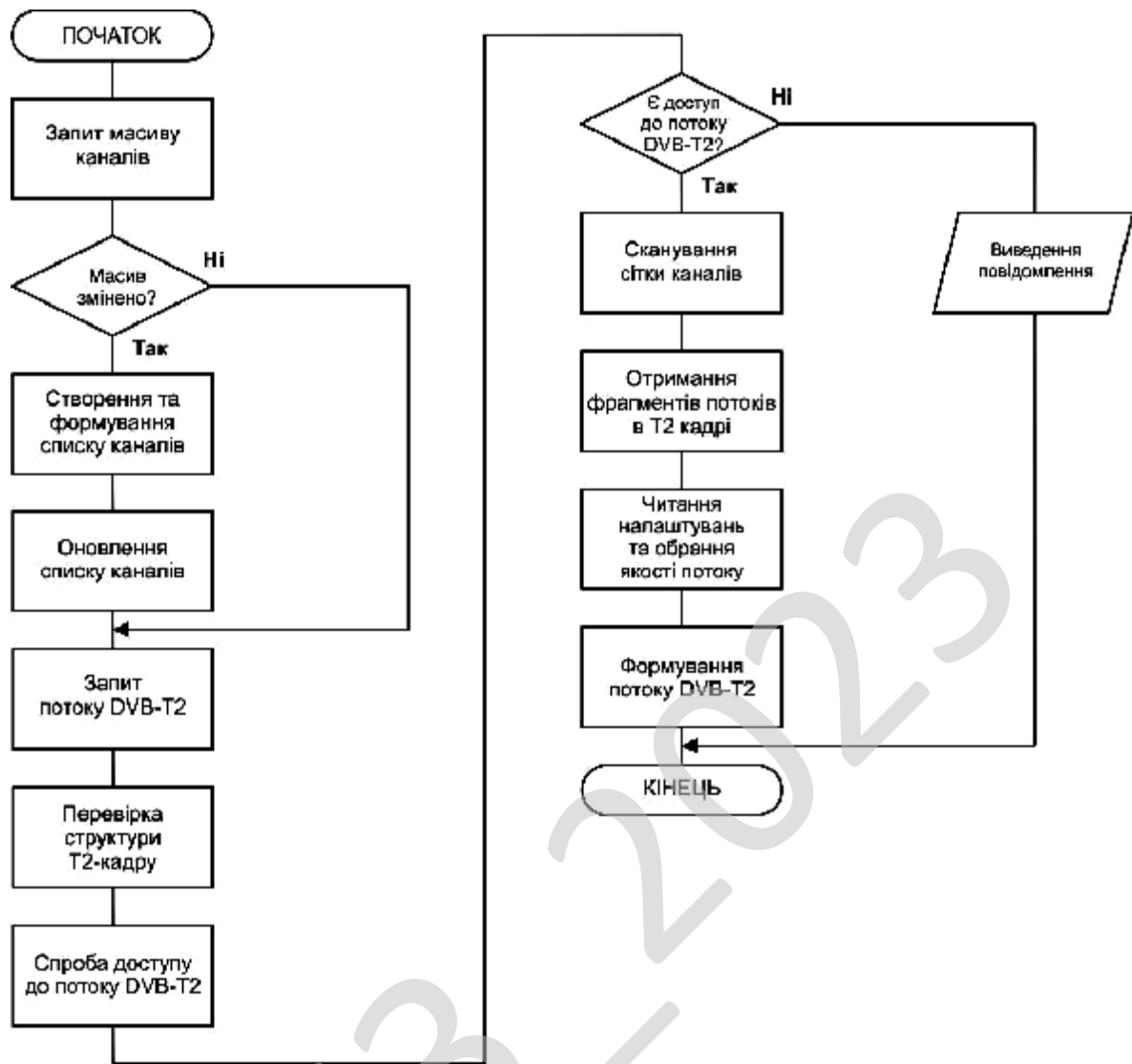


Рисунок 4.2 – Блок-схема підпрограми

- Сканування сітки каналів.
- Отримання фрагментів потоків в T2 кадрі.
- Читання налаштувань та обрання якості потоку.
- Формування потоку DVB-T2.

Розглянемо опис алгоритмів функціонування системи

Етапи програмування DirectX. DirectDraw може виявитися досить каверзним у використанні. На перший вигляд він здається простим; існує тільки кілька класів і вони не мають великої кількості методів.

Переключення на повно віконний режим. Наступне, що необхідно зробити, – це переключити дисплей у режим перелистування сторінок. Коли ви це робите, стає видимим тільки ПЗ.

Воно займає весь екран. Будь-які інші додатки Windows, що знаходяться в режимі виконання, подібні Windows Explorer, продовжують працювати і можуть записувати вихідні дані на те, що вони вважають екраном. Ви не бачите, як виглядають вихідні дані, тому що інші додатки усе ще використовують GDI для вихідних даних, якому нічого не відомо про DirectDraw. Але вам зовсім немає необхідності турбуватися про це. GDI буде продовжувати безтурботно писати в екранну пам'ять, хоча ви не зможете побачити його вихідні дані.

Шляхом переключення в спеціальний режим відображення даних проводиться займання всього екрану.

Як правило, ви можете запускати безліч регулярних додатків середовища Windows у той самий час; їхні вікна перекриваються і завдяки GDI справи йдуть прекрасно. Але що відбудеться, якщо ви спробуєте запустити два і більш повно віконних DirectDraw програм у той самий час? Відповідь – тільки одному дозволений доступ до повного екрана.

DirectDraw керує цим, припускаючи, що ви маєте винятковий доступ до екранної карти перед зміною режиму. Ви зробите це установкою кооперативного рівня об'єкта DirectDraw у Exclusive. DirectDraw підтримує ексклюзивний рівень доступу тільки для одного додатка одночасно. Якщо ви спробуєте одержати ексклюзивний доступ і який-небудь інший додаток уже його має, виклик не вдасться.

Подібним же чином, якщо ви спробуєте змінити режими відображення даних без придбання ексклюзивного доступу, цей виклик не вдасться.

Таким чином, спробуйте одержати ексклюзивний доступ і потім переключите режими відображення.

Тут необхідно відзначити, що ви повинні надати описувач вікна SetCooperativeLevel. DirectDraw змінює розміри цього вікна автоматично, так що

воно заповнює екран у новому режимі відображення даних. Ви повинні передати описувач форми в SetCooperativeLevel.

Через те, що описувач вікна не був створений до часу виклику OnCreate, ви повинні все це зробити і події OnShow.

```
procedure TForm1.FormShow (Sender: TObject);
begin
if DirectDraw.SetCooperativeLevel(
    Handle,DDSCliEXCLUSIVE or DDSCliFULLSCREEN
    ) <> DDiOK
then
    Raise Exception.Create ( 'Unable to acquire exclusive full-screen access');
    if DirectDraw. SetDisplayMode( 640, 480, 8 ) <> DDiOK
then
    Raise Exception. Create ( 'Unable to set new display mode' ) ;
end;
```

Поки все в порядку. Нічого тут складного немає. Якщо запустити приклад прямо зараз, ваш додаток переключить режими і ви побачите, як форма заповнить екран.

Якщо ви пересунете змініте її розміри, ви побачите за ним Delphi. Ви усе ще дивитесь на поверхню висновку GDі. GDі може благополучно виводити дані в цих різних режимах, так що ви побачите свої звичайні додатки Windows так довго, скільки ця поверхня буде залишатися на передньому плані. Але через те, що ви створюєте додаток з мигтючими сторінками, це не зовсім те, що нам потрібно.

Поверхні відображення. Тепер можна створювати поверхні відображення. У DirectDraw поверхню відображення являє собою лінійну область екранної пам'яті, до якої можна одержати безпосередній доступ для маніпуляцій.

Поверхня відображення, що ви бачите на екрані, називається основною поверхнею. Вона представляє пам'ять видимого кадрового буфера на карті відображення. Ви також можете мати невидимі поверхні, що визначаються як поза екрані, чи оверлейні поверхні.

Подібне може існувати або в регулярній системній пам'яті, або у поза екранній області пам'яті на самій графічній карті.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Для того, щоб створити ситуацію з мигтючими сторінками, необхідні основна поверхня і, принаймні, одна поза екрана поверхня для здійснення відображення. Для того, щоб поза екрана поверхня могла з'являтися і зникати на екрані, вона повинна знаходитися у відео пам'яті.

Проте, DirectDraw намагається створити поверхні у відеопам'яті за замовчуванням, тому немає необхідності починати що-небудь спеціально.

Існує спосіб для створення основної поверхні й однієї і більш змінних поверхонь у те саме час – за рахунок створення комплексної поверхні.

Ще один аспект у створенні комплексної (складений) поверхні полягає в тім, що ви можете звільнити всі поверхні в комплексному ланцюзі змінних поверхонь шляхом вивільнення самої комплексної поверхні. Для приклада створимо комплексну поверхню за допомогою однієї допоміжної буферної поверхні.

Будь-які фонові поверхні, що створюються як частину комплексної, відомі як неявні поверхні.

Існує велике число операцій, що ви не зможете здійснити за допомогою неявних поверхонь, наприклад, від'єднати їх від основної чи поверхні звільнити їхній незалежно від основної поверхні.

Однак, комплексні поверхні набагато простіше створювати, тому що DirectDraw створює фонові буфери і з'єднує їх з основною поверхнею. У цьому зв'язку я повинен затронати питання складності DirectDraw, оскільки необхідно заповнювати поле і запис TDDSurfaceDesc.

```
{ заповнити описатель DirectDrawSurface перед створенням поверхні }
    FillChar (DDSsurfaceDesc, sizeof (DDSsurfaceDesc), 0);
    with DDSurfaceDesc do begin
        dwSize := sizeof ( DDSurfaceDesc );
        dwFlags := DDSdiCAPS or DDSdiBACKBUFFERCOUNT;
        ddSCaps.dwCaps := DDSCAPSiCOMPLEX or DDSCAPSiFLi or
            DDSCAPSiPRIMARYSURFACE;
        dwBackBufferCount := 1;
    end ;
{ створити комплексну поверхню }
    if DirectDraw.CreateSurface (DDSsurfaceDesc, PrimarySurface, Ni) <> DDIOk
```

```
then
  Raise Exception.Create ('CreateSurface failed');
```

DDSurface – це локальна перемінна. Поле dwFlags повідомляє DirectDraw, яке з полів, що залишилися, є дійсним.

Це саме те, що дозволяє ігнорувати більшість з них. У цьому зв'язку ви тільки використовуєте ddSCaps і dwBackBufferCount. (ddSCaps повинний завжди бути заповненим).

```
procedure TForm1.FormKeyDown (Sender: TObject; var Key: Word; Shift:
                                TShiftState);
begin
  // якщо натиснуті клавіші Escape чи F12 припинити роботу ПЗ
  case Key of
    VK_ESCAPE, VK_F12 : Close;
  end;
end;
```

Малювання на поверхні DirectDraw

Існує два способи малювати на поверхні DirectDraw. Ви можете одержати покажчик безпосередньо на область пам'яті поверхні і безпосередньо нею маніпулювати.

Це дуже потужний спосіб, але вимагає написання спеціального коду і часто для швидкості – на асемблері. Усе-таки вам рідко прийдеться це робити, тому що DirectDraw може створювати контекст пристрою (DC), сумісний з GDі.

Це означає, що ви можете малювати на ній, використовуючи стандартні виклики GDі, а також будь-який DC. Однак, виклики GDі досить стомлюючі, і Delphi уже включає DC у свій клас TCanvas. Таким чином, у прикладі я створюю TCanvas і використовую його для полегшення собі життя.

Усе, що необхідно зробити, – створити об'єкт TCanvas і викликати метод GetDC поверхні. Потім ви призначаєте DC Canvas.Handle, переконавши, що ви по завершенню переустановили Handle у нуль.

Створення канви і розміщення контекстів пристроїв вимагає пам'яті і ресурсів. Контексти пристрою являють собою особливо поганий ресурс. Для того, щоб зробити код необхідно використовувати блоки try...finally.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

Він просто заповнює основну поверхню блакитним кольором і виводить текст "Primary surface" (Основна поверхня) у центрі ліворуч. Фоновий буфер зафарбовується в червоний колір і містить текст "Back buffer" (Фоновий буфер) у центрі праворуч.

```
// поверхні відображення
    procedure TForm1.DrawSurfaces ;
    var DC      : HDC ;
        ARect  : TRect ;
        DDCanvas : TCanvas ;
        ATopPos : Integer ;
    begin
//Заповнити основну поверхню червоним, а фоновий буфер - блакитним кольором
// і вивести на обидва деякий текст.
        DDCanvas := TCanvas.Create ;
        try
// first output to the primary surface
            if PrimarySurface.GetDC( DC ) = DD_OK then
                try
                    ARect := Rect( 0, 0, 640, 480 ) ;
                    with DDCanvas do begin
                        Handle := DC ;
// вивести полотно на DC
                        Brush.Color := clRed ;
                        FillRect( ARect ) ;
                        Brush.Style := bsClear ;
// прозорий фон для тексту
                        Font.Name := 'Arial' ;
                        Font.Size := 24 ;
                        Font.Color := clWhite ;
                        ATopPos := ( 480 - TextHeight( 'A' ) ) div 2 ;
                        TextOut( 10, ATopPos, 'Primary surface' ) ;
                    end ;
                finally
// переконалися, що все акуратно завершено і DC звільнений
                    DDCanvas.Handle := 0 ;
                    PrimarySurface.ReleaseDC( DC ) ;
                end ;
// працюємо з фоновим буфером
            if BackBuffer.GetDC( DC ) = DD_OK then
                try
```

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

```
        with DDCanvas do begin
            Handle := DC ;
// вивести полотно на DC
            Brush.Color := clBlue ;
            FillRect( ARect ) ;
            Brush.Style := bsClear ;
// прозорий фон для тексту
            Font.Name := 'Arial' ;
            Font.Size := 24 ;
            Font.Color := clWhite ;
            TextOut( 630 - TextWidth( 'Back buffer' ), ATopPos, 'Back buffer' ) ;
            end ;
        finally
            DDCanvas.Handle := 0 ;
            BackBuffer.ReleaseDC( DC ) ;
        end ;
        finally
            DDCanvas.Free ;
        end ;
    end ;
```

Помітьте, що я не використовую тут `MakeItSo` при виклику `GetDC`. На це мається дві причини.

По-перше, поверхні повинні бути відновлені безпосередньо перед викликом `DrawSurface`.

По-друге, `DrawSurface` звичайно викликається `MakeItSo`, і ви хочете уникнути невизначеної рекурсії. Якщо виклик `GetDC` не вдається, `Windows` не створює будь якого висновку.

Украй небажано, щоб `GetDC` не удався. Якщо ви хочете одержати повідомлення про помилку, коли подібне відбудеться, ви можете перевірити на успішність і викликати виключення.

Ініціалізація `Direct Audio`. Перед тим як ми зможемо програти звук, необхідно ініціалізувати `Direct Audio`. Це робиться в три етапи.

1. Ініціалізація `COM`.

На відміну від інших компонентів `Direct`, `Direct Audio` – чистий `(D)COM`.

Одна річ повинна бути відзначена: після запиту `set_freq`, потрібно затримати якийсь час (100ms, перед читанням статусу радіо (`Get_stereo`)). Якщо радіо не набудовано, `get_stereo` буде повертати 0, інакше це поверне цінність відмінний від нуля.

```
пошук станції 100.0 100.2
----->-----*****-----
Є станція десь 100.0~100.2 MHz
```

Вищезгаданий підхід не може розташовувати кращої радіостанції, тому що це зупиниться негайно, коли `get_stereo` повертаються 1.

Подібно вищезгаданому числу, функція `seraching` зупиниться в 100.00 MHz замість кращої станції 100.1 MHz.

Тому, протягом шукаючого режиму, коли кожний знаходить, радіо знаходиться в режимі стерео, кожний повинний продовжувати набудовувати, щоб з'ясувати те, що радіочастота не набудована (мається 100.2MHz), після вищезгаданого два кроки, ми можемо укласти, що радіостанція пошуку повинна зупинити в $100.1\text{MHz} = (100.0+100.2) / 2$.

Програмування TV пульта DVB-T2 тюнера.

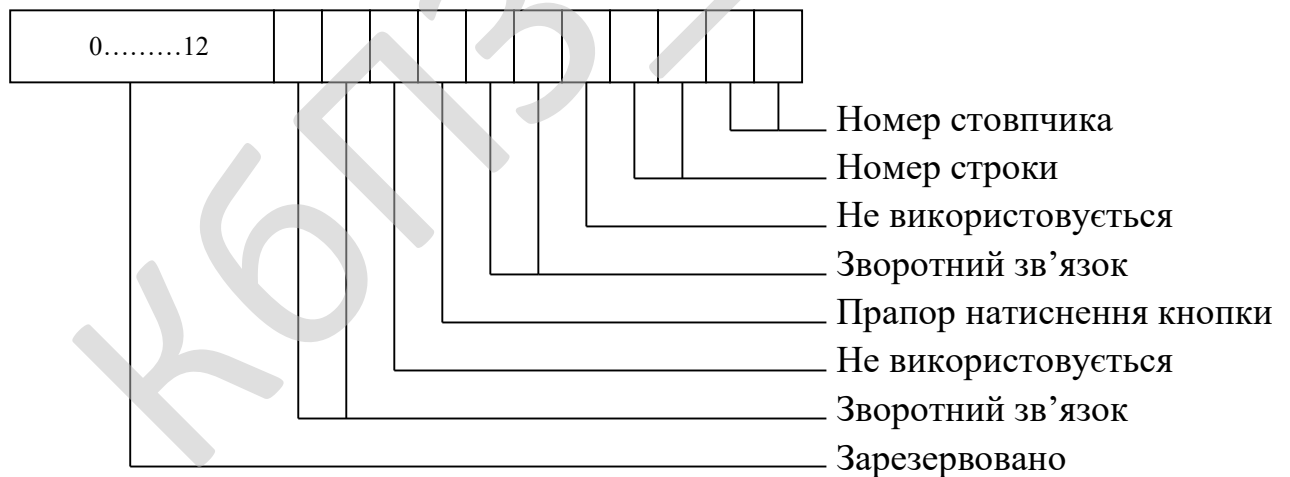


Рисунок 4.3 – Інформаційний GPI_DATA порт

Таблиця 4.2 – Діапазони каналів кабельного телебачення для відео формату

PAL-B

Канал	Відео потік	Аудіо потік	Діапазон розтягу	Канал	Відео потік	Аудіо потік	Діапазон розтягу
1	2	3	4	5	6	7	8
2	55.25	59.75	54-60	36	603.25	607.75	602-608
3	61.25	65.75	60-66	37	609.25	613.75	608-614
4	67.25	71.75	66-72	38	615.25	619.75	614-620
5	77.25	81.75	76-82	39	621.25	625.75	620-626
6	83.25	87.75	82-88	40	627.25	631.75	626-632
7	175.25	179.75	174-180	41	633.25	637.75	626-632
8	181.25	185.75	180-186	42	639.25	643.75	638-644
9	187.25	191.75	186-192	43	645.25	649.75	644-650
10	193.25	197.75	192-198	44	651.25	655.75	650-656
11	199.25	203.75	198-204	45	657.25	661.75	656-662
12	205.25	209.75	204-210	46	663.25	667.75	662-668
13	211.25	215.75	210-216	47	669.25	673.75	668-674
14	471.25	475.75	470-476	48	675.25	679.75	674-680
15	477.25	481.75	476-482	49	681.25	685.75	680-686
16	483.25	487.75	482-488	50	687.25	691.75	686-692
17	489.25	493.75	488-494	51	693.25	697.75	692-698
18	495.25	499.75	494-500	52	699.25	703.75	698-704
19	501.25	505.75	500-506	53	705.25	709.75	704-710
20	507.25	511.75	506-512	54	711.25	715.75	710-716
21	513.25	517.75	512-518	55	717.25	721.75	716-722
22	519.25	523.75	518-524	56	723.25	727.75	722-728
23	525.25	529.75	524-530	57	729.25	733.75	728-734
24	531.25	535.75	530-536	58	735.25	739.75	734-740
25	537.25	541.75	536-542	59	741.25	745.75	740-746
26	543.25	547.75	542-548	60	747.25	751.75	746-752
27	549.25	553.75	548-554	61	753.25	757.75	752-758
28	555.25	559.75	554-560	62	759.25	763.75	758-764
29	561.25	565.75	560-566	63	765.25	769.75	764-770
30	567.25	571.75	566-572	64	771.25	775.75	770-776
31	573.25	577.75	572-578	65	777.25	781.75	776-782
32	579.25	583.75	578-584	66	783.25	787.75	782-788
33	585.25	589.75	584-590	67	789.25	793.75	788-794
34	591.25	595.75	590-596	68	795.25	799.75	794-800
35	597.25	601.75	596-602	69	801.25	805.75	800-806

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм Khufu. Khufu – це 64-бітовий блоковий шифр. 64-бітовий відкритий текст спочатку розщеплюється на дві 32-бітові половини, L і R . Над обома половинами й певними частинами ключа виконується операція XOR. Потім, аналогічно DES, результати проходять деяку послідовність раундів. У кожному раунді молодший значущий байт L використовується як вхід S-блоку. У кожного S-блоку 8 вхідних біт і 32 вихідних біта. Далі обраний в S-блоці 32-бітовий елемент піддається операції XOR з R . Потім L циклічно зрушується на число, кратним восьми біткам, L і R міняються місцями, і раунд завершується. Сам S-блок не статичний, він міняється кожні вісім раундів. Нарешті, по закінченні останнього раунду, над L і R виконується операція XOR з іншими частинами ключа, і половини поєднуються, утворюючи блок шифртексту.

Хоча частини ключа використовуються для операції XOR із блоком шифрування на початку й кінці виконання алгоритму, головне призначення ключа – генерація S-блоків. Ці S-блоки секретні, по суті, це частина ключа. Повний розмір ключа алгоритму Khufu дорівнює 512 біт (64 байт), алгоритм надає спосіб генерації S-блоків по ключу.

					VKPM-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		61

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено головне вікно програми. З нього видно, що інтерфейс користувача програми складається з таких логічних блоків:

- Меню (розділи налаштування та про програму).
- Блоку відображення скорегованого відеосигналу.
- Блоку обрання каналу.
- Функціональні клавіші (Підключити, Відключити, Сканування).



Рисунок 5.1 – Головне вікно програми

На рисунку 5.2 зображено форму авторського права, де відображені дані розробника.

Було обрано Shareware умову розповсюдження. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (неповнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання. Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно.

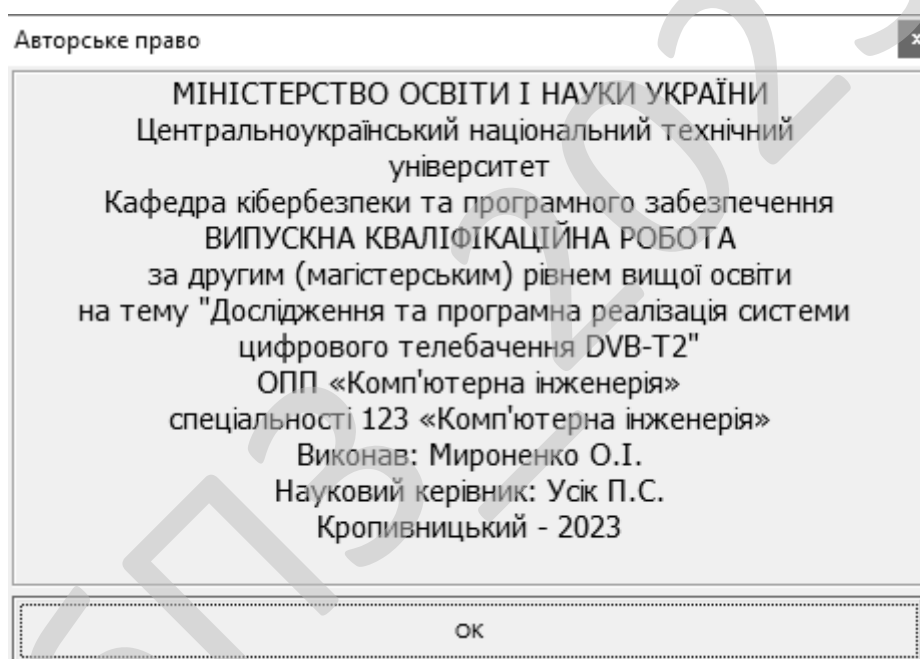


Рисунок 5.2 – Довідка

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи цифрового телебачення DVB-T2.

Метою розробки є дослідження та програмна реалізація системи цифрового телебачення DVB-T2.

Об'єктом дослідження є процес цифрового телебачення DVB-T2.

Предметом дослідження є методи цифрового телебачення DVB-T2.

Методи дослідження базуються на методах цифрового телебачення, методах математичної статистики, методах розробки програмного забезпечення.

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

- Удосконалено метод цифрового телебачення DVB-T2.
- Розроблено вітчизняний продукт цифрового телебачення DVB-T2, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

Продовження табл. 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 ПО для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	40000
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	55
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.:

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

де A – коефіцієнт Боєма, $A=2,45$; $Size$ – загальний об'єм відлагодженого програмного коду, тис. рядків; B – показник ступеня, що визначається співвідношенням

$$B = 1,01 + 0,001 \sum W_i \quad (7.2)$$

де W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \Pi V_j, \quad (7.3)$$

де ΠV_j – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33+0,2(B-1,01)} S, \quad (7.4)$$

де C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 150 = 252 \text{ люд/день}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

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	90	10	900	15
Монітор	60	10	600	10
Клавіатура	30	10	300	5
Маніпулятор «мишка»	30	10	300	5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	2	240	4
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	3	90	1,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	300	750	12,5
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	56,66

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:

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

$$\Phi_{op}^c = \frac{56,66 \cdot 3}{1,2} = 141,65 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Продовження таблиці 7.4

Посада	Вид роботи	Час	Кількість штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців:

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	21000	63000
Продакт-менеджер	0,25	15000	11250
Інженер-програміст	5,3	18000	286200
Інженер-електронщик	0,3	15000	13500
Інженер-системотехнік	0,25	16000	12000
Адміністратор мережі	0,5	15000	22500
Системний програміст	0,25	15000	11250
Дизайнер WEB	0,25	16000	12000
Інженер-верстальник	0,25	14000	10500
Бухгалтер-економіст	0,5	17000	25500
Всього за період розробки	$R_{cn}=8,85$	-	$\Phi_{роб}=467700$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{сд} = \frac{467700}{8,85 \cdot 60} = 881 \text{ грн.}$$

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-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

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

де R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць. S_y – питома площа на одне робоче місце, m^2 ; $C_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\delta} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

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

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

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

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

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 по прайсу Інтернет магазину Компбест за 30.10.23 – джерело <https://compbest.com.ua>.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Монітор	Samsung S22R350FHI (LS22R350FHIXCI) / 21.5" (1920x1080) IPS LED / HDMI, VGA	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струменевий	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

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

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 випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	10947	8757,6	96333,6
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	114885,1

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	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	114885	-	-
Всього по групі	114885	50	57442,5
Група 5			
4. Вимірювальні пристрої	5190	-	-
5. Господарський інвентар	28000	-	-
Всього по групі	33190	25	8297,5
Нематеріальні активи			
6. Нематеріальні активи	40000	10	4000
Разом	$K_p = 1596075$		$A_p = 140140$

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців

$$z_o = \frac{z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де N_e – Кількість екземплярів програм, шт.

$$Z_o = 881 \cdot 293 / 40 = 6453 \text{ грн}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де H_q – норматив додаткової зарплати, %

$$Z_d = 6453 \cdot 10 \cdot 0,01 = 645 \text{ грн}$$

Відрахування на соціальні потреби за нормативом $H_c=22\%$ від суми основної та додаткової зарплати

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де H_c – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 22(6453+645) = 1562 \text{ грн}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_g=15\%$ від основної зарплати

$$G_{ocn} = Z_o \cdot H_g \cdot 0,01, \quad (7.14)$$

де H_g – загальногосподарські витрати, %

$$G_{ocn} = 6453 \cdot 15 \cdot 0,01 = 968 \text{ грн}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де Z_{M1} – вартість паперу, грн., Z_{M2} – вартість запам'ятовуючих пристроїв, грн., Z_{M3} – вартість фарби, картриджів, тонеру, грн., N_e – кількість екземплярів програм, шт.

Згідно норм n_{mic} приймаємо 1/6 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n=210$ грн., визначаємо вартість паперу за період розробки $N_m=3$ міс:

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

$$Z_{M1} = 210 \cdot 3 \cdot 1/6 = 105 \text{ грн.}$$

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості 10 примірників:

$$Z_{M2} = \sum C_{d.}, \quad (7.17)$$

де C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 49,2 грн/шт., DVD-R LG 4,7Gb, 16x speed Cake box – 49,2 грн/шт.

$$Z_{M2} = 49,2 \cdot 10 = 492 \text{ грн.}$$

Згідно виданих 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.

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

$$Z_M = (105 + 492 + 1702) / 40 = 57 \text{ грн.}$$

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, %

$$O_n = 6453 \cdot 15 \cdot 0,01 = 968 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 40$ прим.)

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

де A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 140140 \cdot 3 / (40 \cdot 12) = 876 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_M + O_n + A_m. \quad (7.21)$$

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

$$C_n = 6453 + 645 + 1562 + 968 + 57 + 968 + 876 = 11529 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_p) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 55%

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де P_n – рівень рентабельності, %

$$P_p = 0,01 \cdot 55 \cdot 11529 = 6341 \text{ грн.}$$

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. Основна зарплата виконавців	Z_o	6453
2. Додаткова зарплата виконавців	Z_d	645
3. Відрахування на соціальні потреби	C_{oc}	1562
4. Загальногосподарські витрати	Γ_{ocn}	968
5. Витрати на матеріали	Z_M	57
6. Освоєння нових операційних систем, мов програмування	O_n	968
7. Амортизація основних фондів	A_M	876
8. Повна собівартість програмного забезпечення	C_n	11529
9. Плановий прибуток	P_p	6341
10. Ціна підприємства $C_n = C_n + P_p$	C_n	17870
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{oc} \cdot C_n$	$ПДВ$	3574
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	24211

Витрати на оплату праці:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де T_p – кількість годин обслуговування системи за рік, год.; Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 180 годин на рік до 80 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 180 \cdot 160 \cdot 1,1 \cdot 1,22 = 38650 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 80 \cdot 160 \cdot 1,1 \cdot 1,22 = 17178 \text{ грн.}$$

Витрати на електроенергію визначаються з урахуванням спожитої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$).

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

$$Z_{ел \text{ баз}} = 0,475 \cdot 7200 \cdot 3,8 = 12996 \text{ грн}$$

$$Z_{ел \text{ нов}} = 0,475 \cdot 6900 \cdot 3,8 = 12455 \text{ грн}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	50	–	24211	–	12105,5
Всього відрахувань	-	–	24211	–	12105,5

$$T_{cn} = \frac{24211}{51646 - 41738} = 2,4 \text{ років}$$

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.	40
2. Повна собівартість розробленої програми	Грн	11529
3. Ціна розробленої програми	Грн.	17870
4. Плановий прибуток від реалізації розробленої програми	Грн.	6341
5. Рентабельність програмної продукції	%	55
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1596075
7. Загальний прибуток від реалізації програмної продукції	Грн.	253640
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	218605
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років.	1,6
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	24211
11. Величина економічного ефекту у користувача програмної продукції	Грн.	3855
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	2,4

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.

КБПЗ-2023

					VKPM-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Найявний в даний час в нашій країні комплекс розроблених організаційних заходів та технічних засобів захисту, накопичений передовий досвід роботи ряду обчислювальних центрів показує, що є можливість домогтися значно більших успіхів у справі усунення впливу на працюючих небезпечних і шкідливих виробничих факторів. Проте стан умов праці та його безпеки в ряді обчислювальних центрів (ОЦ) та підприємств ще не задовольняють сучасним вимогам. Оператори ЕОМ, оператори підготовки даних, програмісти та інші працівники ОЦ та підприємств ще стикаються з впливом таких фізично небезпечних і шкідливих виробничих факторів, як підвищений рівень шуму, підвищена температура зовнішнього середовища, відсутність або недостатня освітленість робочої зони, електричний струм, статична електрика і інші.

Багато працівників ОЦ та підприємств пов'язані з впливом таких психофізичних факторів, як розумова перенапруга, перенапруження зорових і слухових аналізаторів, монотонність праці, емоційні перевантаження. Вплив зазначених несприятливих факторів призводить до зниження працездатності, викликане розвиваються втому. Поява і розвиток втоми пов'язане зі змінами, які виникають під час роботи в центральній нервовій системі, з гальмівними процесами в корі головного мозку. Наприклад сильний шум викликає труднощі з розпізнаванням колірних сигналів, знижує швидкість сприйняття кольору, гостроту зору, зорову адаптацію, порушує сприйняття візуальної інформації, зменшує на 5 – 12% продуктивність праці. Тривала дія шуму з рівнем звукового тиску 90 дБ знижує продуктивність праці на 30 – 60%.

Медичні обстеження працівників ОЦ та підприємств показали, що крім зниження продуктивності праці високі рівні шуму призводять до погіршення

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

слуху. Тривале перебування людини в зоні комбінованого впливу різних несприятливих факторів може призвести до професійного захворювання. Аналіз травматизму серед працівників ВЦ показує, що в основному нещасні випадки відбуваються від впливу фізично небезпечних виробничих факторів при заправці носія інформації на обертний барабан при знятому кожусі, під час співробітниками невластивих їм робіт. На другому місці випадки, пов'язані з дією електричного струму.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ЕОМ) та інше обладнання є джерелами безпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

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;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

машин» [5], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»). Таним чином можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

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 Ia. 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 Ia, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Ia			Фактичні		
	Температура, °C	Воло- гість,%	Швидкість повітря, м/с	Температура, °C	Воло- гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23	40-55	0,1
Тепла	23-25	50-70	0,1	24-25	50-65	0,11

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.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [1], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп’ютером, згідно ДБН В.2.5-28:2018 [1], можна віднести до роботи з малою точністю (найменший розмір об’єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об’єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об’єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого

типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [1], Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

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: регулярні періодичні наочні огляди персоналом шляхів 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'язково наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

8.5 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток $80 \cdot 50 \cdot 6$ мм., (згідно з ДСТУ 8769:2018 «Кутики сталеві гарячекатані нерівнополічні. Сортамент») довжиною $L=1,7$ м., та горизонтальний електрод – металева полоса з перетином $60 \cdot 5$ мм. Напруга – $220/380$ В. Розрахункова схема розташування заземлюючих електродів – у ряд.

Розрахунок проведемо за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта – чорнозем, нижнього шару ґрунта – глина (питомий опір $\rho_2 = 40$ Ом·м). Умовна товщина верхнього шару ґрунта: $H=0,5$ м. Відстань між вертикальними заземлювачами (електродами) $A=1,7$ м. Глибина закладення горизонтального контура заземлення $t=0,6$ м. Опір заземлювача, який нормується: $R_{3H} = 4$ Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Розрахунок

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,6+1,7/2=1,45 \text{ м.}$$

Розрахунковий питомий опір ґрунта (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунта):

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

$$R = (R_0 \cdot R_{II}) / (R_0 \cdot \eta_{II} + N \cdot R_{II} \cdot K_{ев}) =$$

$$= (21,9 \cdot 23,5) / (21,9 \cdot 0,75 + 9 \cdot 23,5 \cdot 0,8) = 3,48 \text{ Ом.}$$

де $\eta_{II} = 0,75$ – табличне значення коефіцієнта екранування з'єднуючої полоси [10].

Умова $R \leq R_{3H}$ виконується ($3,48 \leq 4$).

Так як при 7 вертикальних електродах R суттєво менше R_{3H} , зменшимо кількість вертикальних електродів N до 6 і виконаємо перерахунок. У результаті остаточно отримали: $R = 3,98 \text{ Ом.}$ при кількості вертикальних електродів $N=6$.

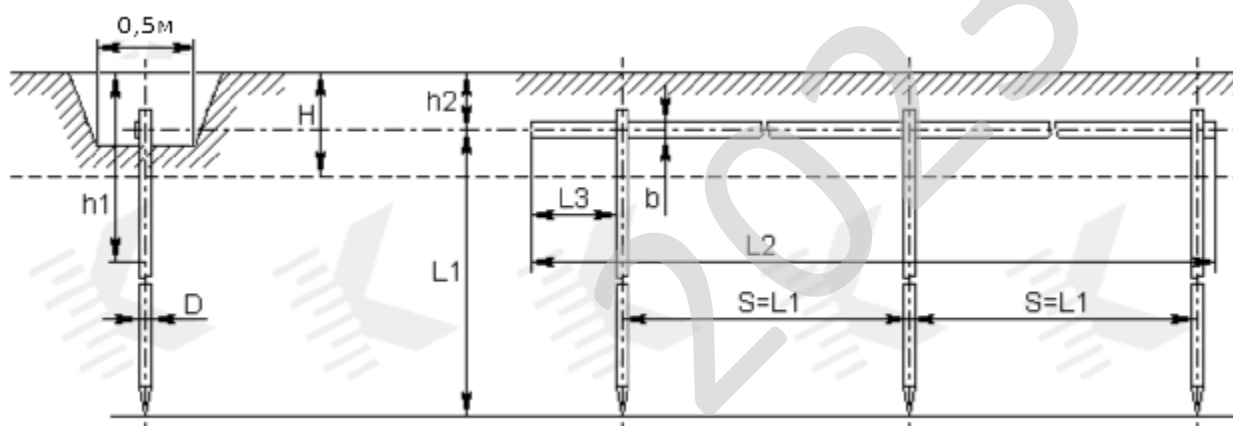


Рисунок 8.1 – Схема штучного заземлення.

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 умов праці на робочому місці програміста, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи, розроблені заходи з умов поліпшення охорони праці.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

Список використаних джерел інформації

1. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ> (дата звернення 19.10.23).
2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98> (дата звернення 19.10.23).
3. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12> (дата звернення 19.10.23).
4. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.
5. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508> (дата звернення 19.10.23).
6. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99> (дата звернення 19.09.23).
7. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення 19.09.23).
8. Оришака О.В. Охорона праці в галузі та цивільний захист / О.В Оришака, Г.П. Горбачова, О.М. Мезенцева, К.М. Марченко, К.О. Буравченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький: Видавець Лисенко В.Ф., 2019. – 226 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/9258> (дата звернення 19.09.23).

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

9. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. – 19 с. [Електронний ресурс]. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення 19.09.23).

10. Охорона праці. Ч. 1. Захисне заземлення : метод. вказ. до викон. розрахунків з викор. персон. ЕОМ IBM сумісного типу / Кіровоград. ін-т с.-г. машинобуд. ; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград : КІСМ, 1997. – 20 с. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/4358> (дата звернення 19.09.23).

11. Наказ Міністерства регіонального розвитку, будівництва та житлово-комунального господарства України 31.10.2016 «Про затвердження ДБН В.1.1-702016» – Режим доступу до ресурсу: <https://ips.ligazakon.net/document/fn025551> (дата звернення 19.09.23).

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи цифрового телебачення DVB-T2.

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 цифрового телебачення DVB-T2.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем цифрового телебачення DVB-T2.
- Досліджена система цифрового телебачення DVB-T2.
- На основі отриманих результатів досліджень створена програмна реалізація системи цифрового телебачення DVB-T2.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання цифрового телебачення DVB-T2.

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-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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

Програма реалізована на мові високого рівня Delphi 10.4.1. 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 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм Khufu.

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 3855 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 2,4 роки.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мироненко О.І. Дослідження та програмна реалізація системи цифрового телебачення DVB-T2 // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.

2. Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2). – DVB Document A122, June 2008.

3. Implementation guidelines for a second generation digital terrestrial television broadcasting system (DVB-T2). – DVB Document A133, February 2009.

4. Peter Hoddie, Lizzie Prader «IoT Development for ESP32 and ESP8266 with JavaScript: A Practical Guide to XS and the Moddable SDK» ISBN-13 (pbk): 978-1-4842-5069-3 ISBN-13 (electronic): 978-1-4842-5070-9

5. STM32CubeMX for STM32 configuration and initialization C code generation. User manual. June 2022. 397 p.

6. І.В.Чихіра, А.Г. Микитишин Конспект лекцій з дисципліни «Програмування систем реального часу» / Укладачі : Чихіра І.В., Микитишин А.Г., – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя , 2016. – 76 с.

7. Jack Ganssle and Michael Barr. 2003. Embedded Systems Dictionary. CMP Books.

8. Технології інтернету речей. Навчальний посібник [Електронний ресурс]: / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 12,5 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 271 с.

9. Greg Dunko, Joydeep Misra, Josh Robertson, Tom Snyder “A reference guide to the Internet of Things” / 2017 Bridgera LLC, RIoT.

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

10. Donald Norris “Programming with STM32. Getting started with the Nucleo Board and C/C++” 416 p. 2018.
11. Neil Kolban “Kolban’s book on ESP32”. Texas, USA. 951 p.
12. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
13. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.
14. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.
15. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.
16. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.
17. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.
18. Smirnov O., Kuznetsov A., Pushkar’ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In:

Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

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

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

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

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

23. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

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

25. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE*

					BKPM-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

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

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

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

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

30. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019*, Pages 618-629.

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

32. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

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

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

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

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

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

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

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

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

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

41. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

42. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія.* – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

43. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки.* № 2(33). с. 161-172, 2019.

44. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

45. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

46. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

47. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

48. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

49. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

50. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

51. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

					ВКРМ-123.23.0016.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

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-123.23.0016.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Мироненко О.І.				Літ.	Аркуш	Аркушів
Перевірів	Усік П.С.						
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-22М-1		
Затв.	Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи цифрового телебачення DVB-T2.

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. № 34-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи цифрового телебачення DVB-T2.

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-123.23.0016.00.00.Т3	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 системи цифрового телебачення DVB-T2;
- 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-123.23.0016.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Delphi 10.4.1.

					ВКРМ-123.23.0016.00.00.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 2023 року.

8 Вимоги щодо охорони праці

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

					ВКРМ-123.23.0016.00.00.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 – 105 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 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 22.12.2023 р.

					ВКРМ-123.23.0016.00.00.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

_____ Усік П.С.

*Дослідження та програмна реалізація
системи цифрового телебачення DVB-T2*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 45

Літера: РП

DVB-T2.PAS: ФАЙЛ ПРОГРАМИ

```

unit DVB-T2;
// Дослідження та програмна реалізація системи цифрового телебачення
DVB-T2
// Мироненко Олександр Ігорович
// 2023

Interface // Інтерфейсна частина

uses
  GOTVIEW_USB_2_MASTERHD_3I2c,
  PlataCopInterface,
  PlataCopRegisters,
  PlataCopGeneral,
  S11935Registers,
  Stv0299bRegisters,
  Windows;
const
  CDVB-T2BurstA = $01;
  CDVB-T2BurstB = $02;
  CDVB-T2Command = $04;
type
  TDVB-T2Data = array[0..5] of Byte;
function PlataCopUsDelay(UsDelay : Dword): Boolean;
function PlataCopReadEeprom (Handle:THandle; Address : Word; var Data:
array of Byte): Boolean;
function PlataCopWriteEeprom (Handle : THandle; Address:Word; Data:
array of Byte): Boolean;
function PlataCopEepromChecksum (Data : array of Byte): Byte;
function PlataCopCheckQpskResetValues (Handle: THandle): Boolean;
function PlataCopWriteToSynthesizer (Handle: THandle; AddressIndex:
Byte; Data: array of Byte): Boolean;
function PlataCopDetectSynthesizerAddress (Handle: THandle): Byte;
function PlataCopQpskSetAGCInversion(Handle: THandle; Inversion:
Boolean): Boolean;
function PlataCopQpskSetInversion(Handle: THandle; Inversion:
Boolean): Boolean;
function PlataCopQpskGetLockDetectorPercentage(Handle: THandle; var
Percentage: Byte): Boolean;
function PlataCopLowBandLNB(Handle: THandle): Boolean;
function PlataCopHighBandLNB (Handle: THandle): Boolean;
function PlataCopDVB-T2Command (Handle: THandle; Repeats: Byte;
CommandType: Byte; CommandLength: Byte; CommandData: TDVB-T2Data):
Boolean;

implementation

uses
  Math,
  SyncObjs,
  SysUtils;

var
  HighPerformanceAvailable: Boolean;
  HighPerformanceFrequency: TLargeInteger;
  I2cLock:TCriticalSection;
function PlataCopEepromChecksum(Data: array of Byte): Byte;
var
  Loop: Integer;
begin
  Result := 0;

```

```

    for Loop := 0 to High(Data) do
        Result := Result xor Data[Loop];
    end;

function PlataCopReadFromQpsk(Handle: THandle; AccessRegister: Byte;
var Data: Byte): Boolean;
begin
    I2cLock.Acquire;
    try
        Result := PlataCopI2cReadBytes(Handle, CPlataCopI2cChannelFrontEnd,
            CStv0299bDemodulatorReadAddress, AccessRegister, Data);
    finally
        I2cLock.Release;
    end;
end;

function PlataCopWriteDefaultsToQpsk(Handle: THandle; TunerType: Byte):
Boolean;
var
    Loop: Byte;
begin
    Result := False;
    for Loop := Low(CStv0299bDefaultsTBMU24112) to
High(CStv0299bDefaultsTBMU24112) do
        begin
            if CStv0299bDefaultsTBMU24112[Loop, 0] <> $FF then
if not PlataCopWriteToQpsk(Handle, CStv0299bDefaultsTBMU24112[Loop, 0],
CStv0299bDefaultsTBMU24112[Loop, 1]) then
                Exit;
            end;
            Result := True;
        end;
    end;

function PlataCopCheckQpskDefaultValues(Handle: THandle; TunerType:
Byte): Boolean;
var
    Loop: Byte;
    Data: Byte;
begin
    Result := False;
    for Loop := Low(CStv0299bDefaultsTBMU24112) to
High(CStv0299bDefaultsTBMU24112) do
        begin
            if CStv0299bDefaultsTBMU24112[Loop, 0] <> $FF then
                begin
if not PlataCopReadFromQpsk(Handle, CStv0299bDefaultsTBMU24112[Loop,
0], Data) then
                    Exit;
                    if Data <> CStv0299bDefaultsTBMU24112[Loop, 1] then
                        Exit;
                    end;
                end;
            end;
            Result := True;
        end;

function PlataCopCheckQpskResetValues(Handle: THandle): Boolean;
var
    Loop: Byte;
    Data: Byte;
begin
    Result := False;
    for Loop := Low(CStv0299bResets) to High(CStv0299bResets) do
        begin

```

```

    if CStv0299bResets[Loop, 0] <> $FF then
    begin
        if not PlataCopReadFromQpsk(Handle, CStv0299bResets[Loop, 0],
Data) then
            Exit;
            if Data <> CStv0299bResets[Loop, 1] then
                Exit;
            end;
        end;
        Result := True;
    end;

function PlataCopDVB-T2WaitIdle(Handle: THandle): Boolean;
var
    Data : Byte;
    Empty: Boolean;
    Retry: Integer;
begin
    Result := False;
    Retry := 100;
    repeat
        if not PlataCopReadFromQpsk(Handle, CStv0299bDVB-T2Status, Data)
then
            Exit;
            Empty := ((Data and CStv0299bFifoMask) = CStv0299bFifoEmpty);
            if not Empty then
                begin
                    PlataCopUsDelay(1000);
                    Dec(Retry);
                end;
            until Empty or (Retry = 0);
            if Retry = 0 then
                Exit;
            Result := True;
        end;

function PlataCopDVB-T2WaitFree(Handle: THandle): Boolean;
var
    Data : Byte;
    Full : Boolean;
    Retry: Integer;
begin
    Result := False;
    Retry := 20;
    repeat
        if not PlataCopReadFromQpsk(Handle, CStv0299bDVB-T2Status, Data)
then
            Exit;
            Full := ((Data and CStv0299bFifoFull) = CStv0299bFifoFull);
            if Full then
                begin
                    PlataCopUsDelay(1000);
                    Dec(Retry);
                end;
            until (not Full) or (Retry = 0);
            if Retry = 0 then
                Exit;
            Result := True;
        end;

function PlataCopDVB-T2SendByte(Handle: THandle; Data: Byte): Boolean;
begin
    Result := False;

```

```

// Wait for space in fifo
if not PlataCopDVB-T2WaitFree(Handle) then
  Exit;
// Запис даних до FIFO
if not PlataCopWriteToQpsk(Handle, CStv0299bDVB-T2Fifo, Data) then
  Exit;
Result := True;
end;

function PlataCopSendDVB-T2Msg(Handle: THandle; CommandType: Byte;
  CommandLength: Byte; CommandData: TDVB-T2Data; Delay: Boolean):
Boolean;
var
  Loop: Byte;
begin
  Result := False;
  case (CommandType and (CDVB-T2BurstA or CDVB-T2BurstB or CDVB-
T2Command)) of
    $00: begin
      Result := True;
      Exit;
    end;
    CDVB-T2Command: if CommandLength = 0 then
      begin
        Result := True;
        Exit;
      end;
    else Exit;
  end;
  if not PlataCopDVB-T2WaitIdle(Handle) then
    Exit;
  if not PlataCopWriteToQpsk(Handle, CStv0299bDVB-T2,
CStv0299bModulationDVB-T2) then
    Exit;
  if Delay then
    PlataCopUsDelay(16000);
  if ((CommandType and CDVB-T2Command) = CDVB-T2Command) then
    begin
      for Loop := 0 to CommandLength-1 do
        begin
          if not PlataCopDVB-T2SendByte(Handle, CommandData[Loop]) then
            Exit;
          end;
          if not PlataCopDVB-T2WaitIdle(Handle) then
            Exit;
        end;
      end;
    end;
  if ((CommandType and (CDVB-T2BurstA or CDVB-T2BurstB)) <> 0) then
    begin
      if not PlataCopWriteToQpsk(Handle, CStv0299bDVB-T2,
CStv0299bModulationUnmodulated) then
        Exit;
      if (CommandType and CDVB-T2BurstA) <> 0 then
        PlataCopDVB-T2SendByte(Handle, $00)
      else
        PlataCopDVB-T2SendByte(Handle, $FF);
      if not PlataCopDVB-T2WaitIdle(Handle) then
        Exit;
    end;
  Result := True;
end;

```

```

function PlataCopDVB-T2Command(Handle: THandle; Repeats: Byte;
CommandType: Byte; CommandLength: Byte; CommandData: TDVB-T2Data):
Boolean;
var
  DVB-T2Original: Byte;
  RepeatCommands: Byte;
begin
  Result := False;
  if not PlataCopReadFromQpsk(Handle, CStv0299bDVB-T2, DVB-T2Original)
then
  Exit;
  if ((CommandType and CDVB-T2Command) = CDVB-T2Command) and
(CommandLength > 0) then
  begin
    CommandData[0] := $E0;
    if not PlataCopSendDVB-T2Msg(Handle, CDVB-T2Command, CommandLength,
CommandData, True) then
      Exit;
    RepeatCommands := Repeats;
    while RepeatCommands > 0 do
    begin
      PlataCopUsDelay(100000);
      CommandData[0] := $E1;
      if not PlataCopSendDVB-T2Msg(Handle, CDVB-T2Command, CommandLength,
CommandData, True) then
        Exit;
      Dec(RepeatCommands);
    end;
  end;
  if (CommandType and (CDVB-T2BurstA or CDVB-T2BurstB)) <> 0 then
  begin
    if (CommandType and CDVB-T2BurstA) <> 0 then
    begin
      if not PlataCopSendDVB-T2Msg(Handle, CDVB-T2BurstA, 0,
CommandData, True) then
        Exit;
      end
    else
      if not PlataCopSendDVB-T2Msg(Handle, CDVB-T2BurstB, 0,
CommandData, True) then
        Exit;
      end;
    PlataCopUsDelay(16000);
    if (DVB-T2Original and not CStv0299bModulationModeMask) =
CStv0299bModulationContinuous then
      PlataCopHighBandLNB(Handle)
    else
      PlataCopLowBandLNB(Handle);
    Result := True;
  end;

function PlataCopDVB-T2Command2(Handle: THandle; CommandType: Byte;
CommandLength: Byte; CommandData: TDVB-T2Data): Boolean;
var
  DVB-T2Original: Byte;
begin
  Result := False;
  if not PlataCopReadFromQpsk(Handle, CStv0299bDVB-T2, DVB-T2Original)
then
  Exit;
  if ((CommandType and CDVB-T2Command) = CDVB-T2Command) and
(CommandLength > 0)
  then

```

```

    if not PlataCopSendDVB-T2Msg(Handle, CDVB-T2Command, CommandLength,
        CommandData, True) then
        Exit;
    if (CommandType and (CDVB-T2BurstA or CDVB-T2BurstB)) <> 0 then
        begin
    if (CommandType and CDVB-T2BurstA) <> 0 then
        begin
    if not PlataCopSendDVB-T2Msg(Handle, CDVB-T2BurstA, 0, CommandData,
        True)
            then
                Exit;
            end
            else if not PlataCopSendDVB-T2Msg(Handle, CDVB-T2BurstB, 0,
CommandData,
                True) then
                    Exit;
            end;
        PlataCopUsDelay(16000);
        if (DVB-T2Original and CStv0299bModulationModeMask) =
            CStv0299bModulationContinuous then
            PlataCopHighBandLNB(Handle)
        else
            PlataCopLowBandLNB(Handle);
        Result := True;
    end;

function ExtractSpacedPart(var Extracted: ShortString): ShortString;
var
    SpacePos: Integer;
begin
    Result := '';
    Extracted := Trim(Extracted);
    SpacePos := Pos(' ', Extracted);
    if SpacePos <> 0 then
        begin
            Result := Copy(Extracted, 1, SpacePos - 1);
            Delete(Extracted, 1, SpacePos);
        end
    else
        begin
            Result := Extracted;
            Extracted := '';
        end;
    end;

function ExtractFromDVB-T2Setting(
    Setting: ShortString;
    var Satellite: ShortString;
    var SLOF: Integer;
    var Polarity: Char;
    var LOF: Integer;
    var DVB-T2: ShortString): Boolean;
var
    ProcessString: ShortString;
    Error: Integer;
begin
    Result := True;
    Satellite := '';
    SLOF := 0;
    Polarity := ' ';
    LOF := 0;
    DVB-T2 := '';
    Satellite := ExtractSpacedPart(Setting);

```

```

ProcessString := ExtractSpacedPart(Setting); // SLOF
Val(ProcessString, SLOF, Error);
if (ProcessString = '') or (Error <> 0) then
begin
    SLOF := 0;
    Result := False;
end;
ProcessString := ExtractSpacedPart(Setting);
if Length(ProcessString) <> 1 then
begin
    Polarity := ' ';
    Result := False;
end
else
begin
    Polarity := ProcessString[1];
    if not ((Polarity) in ['V', 'v', 'H', 'h']) then
        Result := False;
end;
ProcessString := ExtractSpacedPart(Setting);
Val(ProcessString, LOF, Error);
if (ProcessString = '') or (Error <> 0) then
begin
    LOF := 0;
    Result := False;
end;
DVB-T2 := Setting; // DVB-T2
end;

function CreateDVB-T2Setting(
    Satellite: ShortString;
    SLOF: Integer;
    Polarity: Char;
    LOF: Integer;
    DVB-T2: ShortString): ShortString;
begin
    Result := format('%s %5.5d %s %5.5d %s', [Satellite, SLOF, Polarity,
    LOF,
    DVB-T2]);
end;

function PlataCopDVB-T2CommandVdr(Handle: THandle; CommandData:
ShortString):
    Boolean;
var
    Satellite: ShortString;
    Slof: Integer;
    Polarity: Char;
    Lof: Integer;
    DVB-T2Command: ShortString;
    ProcessString: ShortString;
    Error: Integer;
    Value: Integer;
    DVB-T2Data: TDVB-T2Data;
    DVB-T2Index: Integer;
    EndDetected: Boolean;
begin
    Result := True;
if ExtractFromDVB-T2Setting(CommandData, Satellite, Slof, Polarity,
Lof,
DVB-T2Command) then
while (DVB-T2Command <> '') and (Result = True) do
begin

```



```

end;
Ord([''):
begin
  // DVB-T2 команди
  Delete(ProcessString, 1, 1);
  if ProcessString = '' then
    ProcessString := ExtractSpacedPart(DVB-T2Command);
    DVB-T2Index := Low(DVB-T2Data);
    EndDetected := False;
    repeat
      if DVB-T2Index > High(DVB-T2Data) then
        begin
          Result := False;
          Exit;
        end;
      if Length(ProcessString) <> 2 then
        begin
          Result := False;
          Exit;
        end;
      // Конвертація
      Val('$' + ProcessString, DVB-T2Data[DVB-T2Index], Error);
      // Помилка конвертації
      if Error <> 0 then
        begin
          Result := False;
          Exit;
        end;
      Inc(DVB-T2Index);
      if not EndDetected then
        begin
          ProcessString := ExtractSpacedPart(DVB-T2Command);
          if (Length(ProcessString) = 1) and (ProcessString[1] = ']')
            then
              begin
                ProcessString := '';
              end;
          if Length(ProcessString) > 2 then
            begin
              if ProcessString[3] <> ']' then
                begin
                  Result := False;
                  Exit;
                end;
              Delete(ProcessString, 3, 1);
              EndDetected := True;
            end;
          end
        else
          ProcessString := '';
          until (ProcessString = '');
      // Відправлення даних до DVB-T2Data
      if not PlataCopSendDVB-T2Msg(Handle, CDVB-T2Command, DVB-T2Index,
        DVB-T2Data, False) then
        begin
          Result := False;
          Exit;
        end;
      end;
    else
      Result := False;
    end;
  end;
end;

```

```

end
else
Result := False;
end;

function PlataCopReadFromSynthesizer(Handle: THandle; AddressIndex:
Byte; var Data: Byte): Boolean;
var
  AByte      : Byte;
begin
  Result := False;
  if AddressIndex > High(CS11935SynthesizerWriteAddress) then
    Exit;
  I2cLock.Acquire;
  try
    Exit;
    Result := PlataCopI2cReadBytes(Handle, CPlataCopI2cChannelFrontEnd,
      CS11935SynthesizerReadAddress[AddressIndex], 0, Data);
  finally
    I2cLock.Release;
  end;
end;
end;

function PlataCopDetectSynthesizerAddress(Handle: THandle): Byte;
var
  Address      : Byte;
  Success      : Boolean;
ReturnedByte: Byte;
begin
  Address := 0;
  repeat
    Success := PlataCopReadFromSynthesizer(Handle, Address,
ReturnedByte);
    if not Success then
      Inc(Address);
  until Success or (Address > 3);
  if Success then
    Result := Address
  else
    Result := 255;
end;

function PlataCopQpskSetSymbolRate(Handle: THandle; SymbolRateKS:
Dword): Boolean;
var
  Data: Dword;
  AClc: Byte;
  BClc: Byte;
begin
  Result := False;
  if (SymbolRateKS * 1000) > CStv0299bMasterClock then
    Exit;
  AClc := $04;
  BClc := $11;
  if SymbolRateKS < 30000 then
    begin
      AClc := $06;
      BClc := $13;
    end;
  if SymbolRateKS < 14000 then
    begin
      AClc := $07;
      BClc := $13;
    end;
end;

```

```

end;
if SymbolRateKS < 7000 then
begin
  AClc := $07;
  BClc := $0F;
end;
if SymbolRateKS < 3000 then
begin
  AClc := $07;
  BClc := $0B;
end;
if SymbolRateKS < 1500 then
begin
  AClc := $07;
  BClc := $07;
end;
AClc := AClc or $B0;
BClc := BClc or $40;
if not PlataCopWriteToQpsk(Handle, CStv0299bAclc, AClc) then
  Exit;
if not PlataCopWriteToQpsk(Handle, CStv0299bBclc, BClc) then
  Exit;
Data := Round((SymbolRateKS * 1000) / CStv0299bSetSymbolrateUnits);
Data := Data shl 4;
if not PlataCopWriteToQpsk(Handle, CStv0299bSfrH, (Data shr 16) and
$FF) then
  Exit;
if not PlataCopWriteToQpsk(Handle, CStv0299bSfrM, (Data shr 8) and
$FF) then
  Exit;
if not PlataCopWriteToQpsk(Handle, CStv0299bSfrL, Data and $F0) then
  Exit;
Result := True;
end;

function PlataCopQpskGetLockDetectorPercentage(Handle: THandle; var
Percentage: Byte): Boolean;
var
  Data: Byte;
  Temp: Integer;
begin
  Result := False;
  if not PlataCopReadFromQpsk(Handle, CStv0299bCldi, Data) then
    Exit;
  if (Data and $80) <> 0 then
    Temp := Data - 256
  else
    Temp := Data;
  Temp := Temp + 128;
  Temp := (Temp * 100) div 255;
  Percentage := Temp;
  Result := True;
end;

function PlataCopQpskGetSignalPower(Handle: THandle;
var StrengthPercentage: Double;
var StrengthDecibel : Double): Boolean;
var
  DataMSB : Byte;
  DataLSB : Byte;
  Data : Integer;
begin
  Result := False;

```

```

if not PlataCopReadFromQpsk(Handle, CStv0299bAgc2I1, DataMSB) then
  Exit;
if not PlataCopReadFromQpsk(Handle, CStv0299bAgc2I2, DataLSB) then
  Exit;
Data := (DataMSB shl 8) or DataLSB;
StrengthPercentage := 100 - ((Data * 100) / $FFFF);
if StrengthPercentage <> 0 then
  StrengthDecibel := (Log10(4) * 20) + Log10(StrengthPercentage/100)
* 20
else
  StrengthDecibel := 0;
  Y
  Result := True;
end;

```

```

function PlataCopQpskGetInputLevel(Handle: THandle; var InputLevelDbm:
Integer;
  var InputLevelPercentage: Byte): Boolean;
var
  Data: Byte;
  NewData: Integer;
  Calc: Integer;
begin
  Result := False;
  // Читання змінної
if not PlataCopReadFromQpsk(Handle, CStv0299bAgc1I, Data) then
  Exit;
if Data > 128 then
  NewData := Data - 256
else
  NewData := Data;
NewData := NewData - 50;
if NewData > 0 then
  NewData := 0;
Calc := 100 - Abs((NewData * 2) div 3);
if Calc < 0 then
  Calc := 0;
InputLevelPercentage := Calc;
NewData := NewData div 3;
NewData := NewData - 20;
InputLevelDbm := NewData;
Result := True;
end;

```

```

function PlataCopQpskGetNoiseIndicator(Handle: THandle; var Noise:
Double): Boolean;
var
  DataMSB: Byte;
  DataLSB: Byte;
  Data : Word;
begin
  Result := False;
  if not PlataCopReadFromQpsk(Handle, CStv0299bNirH, DataMSB) then
    Exit;
  if not PlataCopReadFromQpsk(Handle, CStv0299bNirL, DataLSB) then
    Exit;
  Data := (DataMSB shl 8) or DataLSB;
  Noise := (-0.0017 * Data) + 19.02;
  Result := True;
end;

```

```

function PlataCopQpskSetAGCInversion(Handle: THandle; Inversion:
Boolean): Boolean;

```

```

var
  OriginalData: Byte;
  Data        : Byte;
begin
  Result := False;
  if not PlataCopReadFromQpsk(Handle, CStv0299bAgc1R, OriginalData)
  then
    Exit;
  Data := OriginalData and not CStv0299bAGCInversion;
  if Inversion then
    Data := Data or CStv0299bAGCInversion;
  if Data <> OriginalData then
    if not PlataCopWriteToQpsk(Handle, CStv0299bAgc1R, Data) then
      Exit;
  Result := True;
end;

function PlataCopQpskSetInversion(Handle: THandle; Inversion: Boolean):
Boolean;
var
  OriginalData: Byte;
  Data        : Byte;
begin
  Result := False;
  if not PlataCopReadFromQpsk(Handle, CStv0299bIoCfg, OriginalData)
  then
    Exit;
  Data := OriginalData and not CStv0299bInversion;
  if Inversion then
    Data := Data or CStv0299bInversion;
  if Data <> OriginalData then
    if not PlataCopWriteToQpsk(Handle, CStv0299bIoCfg, Data) then
      Exit;
  Result := True;
end;

function PlataCopQpskGetCarrierDeviation(Handle: THandle; var
DeviationKHz: Integer): Boolean;
var
  DataM: Byte;
  DataL: Byte;
  Temp: Int64;
begin
  Result := False;
  if not PlataCopReadFromQpsk(Handle, CStv0299bCfrM, DataM) then
    Exit;
  if not PlataCopReadFromQpsk(Handle, CStv0299bCfrL, DataL) then
    Exit;
  Temp := (DataM shl 8) or DataL;
  if (DataM and $80) <> 0 then
    Temp := Temp - 65536;
  Temp := Temp * (CStv0299bMasterClock div 1000);
  DeviationKHz := (Temp div 65536);
  Result := True;
end;

function PlataCopLowBandLNB(Handle: THandle): Boolean;
var
  Data: Byte;
begin
  Result := False;
  if not PlataCopReadFromQpsk(Handle, CStv0299bDVB-T2, Data) then
    Exit;

```

```
Data := Data and not CStv0299bModulationMask;  
Data := Data or CStv0299bModulation0;  
if not PlataCopWriteToQpsk(Handle, CStv0299bDVB-T2, Data) then  
  Exit;  
Result := True;  
end;  
end.
```

K6П3-2023

ПРОГРАМА ОРГАНІЗАЦІЇ ЗВ`ЯЗКУ

```
program DVB-T2;  
// Дослідження та програмна реалізація системи цифрового телебачення  
DVB-T2  
// Мироненко Олександр Ігорович  
// 2023  
Uses // підключення стандартних бібліотек та розроблених модулів  
GOTVIEW_USB_2_MASTERHD_3, Forms,  
tDVB-T2 in 'DVB-T2.pas' {frmDVB-T2},  
tMain in 'Main.pas' {frmOne};  
  
{$R *.RES} // ресурси  
  
begin  
  Application.Title := 'DVB-T2';  
  Application.Initialize;  
  Application.CreateForm(TfrmOne, frmOne);  
  Application.CreateForm(TfrmDVB-T2, frmDVB-T2);  
  Application.Run;  
end.
```

КБПЗ - 2023

GOTVIEW_USB_2_MASTERHD_3.PAS: ФАЙЛ ПРОГРАМИ

```

unit GOTVIEW_USB_2_MASTERHD_3;
// Дослідження та програмна реалізація системи цифрового телебачення
DVB-T2
// Мироненко Олександр Ігорович
// 2023

Interface // Інтерфейсна частина

uses Windows; // Підключення бібліотеки форми

type
  Pvoid = Pointer;
  PPhysicalMemoryAddress = ^PhysicalMemoryAddress;
  PhysicalMemoryAddress = LARGE_INTEGER;
  TGOTVIEW_USB_2_MASTERHD_3GetDriverVersion = record
    MajorVersion: Word; // Версія
    MinorVersion: Word;
    Build: Dword; // час створення
  end;
  TGOTVIEW_USB_2_MASTERHD_3GetDmaStatus = record
    Interrupts: Longint; // переривання
    Isr: Dword; //статус регістр
    VirtualAddress: Pvoid; // Віртуальна адреса
    PhysicalAddress: PhysicalMemoryAddress;
    Size: Dword;
    FifoOverflows: Dword;
  end;
  TGOTVIEW_USB_2_MASTERHD_3DmaBuffer = record
    Identifier: Longint; // ідентифікатор
    VirtualAddress: Pvoid; // віртуальна адреса
    PhysicalAddress: PhysicalMemoryAddress; // фізична адреса
    Size: Dword; // розмір
  end;
  TGOTVIEW_USB_2_MASTERHD_3TransferBuffer = record
    Identifier: Longint; // ідентифікатор
    TransferAddress: Pchar; // показчик на буфер
    SourceIndex: Dword; // Індекс
    TargetIndex: Dword;
    TransferLength: Dword; // кількість байтів
  end;
  TGOTVIEW_USB_2_MASTERHD_3FifoTransferBuffer = record
    Identifier: Longint; // Ідентифікатор
    NumberOfBuffers: Dword;
    TransferAddress: Pchar; // Показчик
    TransferLength: Dword;
    IsValid: Boolean;
    Overflows: Dword;
    Irqs: Dword;
    AllOverflows: Dword;
  end;
  TGOTVIEW_USB_2_MASTERHD_3IrqBuffer = record
    Irqs: Dword;
    IrqsWhenActive: Dword;
    IrqBufferingIsActive: Boolean; // Прапор переривання
    UseNotificationEvent: Boolean;
    UseSignaling: Boolean;
    UseFifo: Boolean;
    IrqAutoDisable: Boolean;
    SignalAndValue: Dword;
    SignalOrValue: Dword;

```

```

SignalXorValue: Dword;
FifoBufferPreviousIndex: Byte;
FifoBufferFirstIndex: Byte; // Буфер FIFO - перше входження
FifoBufferLastIndex: Byte; // Буфер FIFO - останнє входження
FifoBufferCirculations: Dword;
FifoOverflows: Dword;
end;
TGOTVIEW_USB_2_MASTERHD_3IrqTransferBuffer = record
  Identifier: Longint; // Ідентифікатор
  Information: TGOTVIEW_USB_2_MASTERHD_3IrqBuffer; // Опис структури
end;
const
  CGOTVIEW_USB_2_MASTERHD_3DriverName =
  '\\.\GOTVIEW_USB_2_MASTERHD_31';
  CIoctlGOTVIEW_USB_2_MASTERHD_3GetDriverVersion = $222000;
  CIoctlGOTVIEW_USB_2_MASTERHD_3GetDmaStatus = $222004;
  CIoctlGOTVIEW_USB_2_MASTERHD_3ReadFromGOTVIEW_USB_2_MASTERHD_3Register
  = $222008; // Читання реєстру GOTVIEW_USB_2_MASTERHD_3
  CIoctlGOTVIEW_USB_2_MASTERHD_3WriteToGOTVIEW_USB_2_MASTERHD_3Register =
  $22200C; // Запис реєстру GOTVIEW_USB_2_MASTERHD_3
  CIoctlGOTVIEW_USB_2_MASTERHD_3GenerateManualNotification = $222023;
  CIoctlGOTVIEW_USB_2_MASTERHD_3AllocateDma = $222018;
  CIoctlGOTVIEW_USB_2_MASTERHD_3ReleaseDma = $22201C;
  CIoctlGOTVIEW_USB_2_MASTERHD_3ReadFromDma = $222020;
  CIoctlGOTVIEW_USB_2_MASTERHD_3WriteToDma = $222024;
  CIoctlGOTVIEW_USB_2_MASTERHD_3AllocateFifo = $222028;
  CIoctlGOTVIEW_USB_2_MASTERHD_3ReleaseFifo = $22202C;
  CIoctlGOTVIEW_USB_2_MASTERHD_3ReadFromFifo = $222030;
  CIoctlGOTVIEW_USB_2_MASTERHD_3WriteToIrqHandling = $222034;
  CIoctlGOTVIEW_USB_2_MASTERHD_3ReadFromIrqHandling = $222038;
implementation
end.

```

ONE.PAS: ФАЙЛ ПРОГРАМИ

```

unit One;
// Дослідження та програмна реалізація системи цифрового телебачення
DVB-T2
// Мироненко Олександр Ігорович
// 2023

Interface // Інтерфейсна частина

uses
  IdGlobal, IdUDFClient,
  Forms, ExtCtrls, StdCtrls, Mask, Controls, Classes, Gauges, ComCtrls,
  Messages, Buttons, Graphics, Menus, Dialogs, Grids, Outline,
  DirOutln,
  FileCtrl, Spin, Windows;
const
  WM_REMOTE = WM_USER + 1;
  WM_SIGNALLING = WM_USER + 10;
  MessageRight: array[0..3] of Char = 'RIGH';
  MessageRecord: array[0..3] of Char = 'RECO';
  MessageRecordOn: array[0..3] of Char = 'REON';
  MessageRecordOff: array[0..3] of Char = 'REOF';
  MessageEPGShort: array[0..3] of Char = 'EPGS';
  MessageEPGCurrent: array[0..3] of Char = 'EPGC';
  MessageVideo: array[0..3] of Char = 'VIDE';
type
  TfrmOne = class(TForm)
    tmrUpdate: TTimer;
    pgInfo: TPageControl;
    chkEPGShortNameOnly: TCheckBox;
    rgOperation: TRadioGroup;
    Label19: TLabel;
    txtPacketSyncErrors: TLabel;
    chkAutoECM: TCheckBox;
    tmrStateMachineAutoECM: TTimer;
    chkRecordPat: TCheckBox;
    pnlVideo: TPanel;
    pnlBorder: TPanel;
    Settings1: TMenuItem;
    mnuTV: TMenuItem;
    mnuRadio: TMenuItem;
    mnuData: TMenuItem;
    cmbServices: TComboBox;
    tmrStateMachineKeys: TTimer;
    tmrInitStart: TTimer;
    chkRecordEmm: TCheckBox;
    chkRecordMandatory: TCheckBox;
    chkRecordCat: TCheckBox;
    pnlControl: TPanel;
    imgBackgroundButtons: TImage;
    imgRecord2: TImage;
    imgRecord: TImage;
    txtFiltersDefinedPlugin3: TLabel;
    txtFiltersDefinedPlugin4: TLabel;
    Label2: TLabel;
    txtFiltersPlugin1: TLabel;
    txtFiltersPlugin2: TLabel;
    tsTuning: TTabSheet;
    tsProgram: TTabSheet;
    tsMessages: TTabSheet;
    mmoDriver: TMemo;
    cmbTeletextPids: TComboBox;
  end;

```

```

Label16: TLabel;
mskPmt: TMaskEdit;
chkRecordTransponder: TCheckBox;
Label28: TLabel;
txtVideoPackets: TLabel;
stbStatus1: TStatusBar;
stbStatus2: TStatusBar;
chkEpg: TCheckBox;
tbFile: TTrackBar;
mnuMain: TMainMenu;
mnuExit: TMenuItem;
Processrawdatastream1: TMenuItem;
ProcessRecordedTSFile: TMenuItem;
mnuPlugins: TMenuItem;
mnuDummyPlugins: TMenuItem;
txtFiltersPlugin3: TLabel;
txtFiltersPlugin4: TLabel;
mnuDisabled: TMenuItem;
btnDisableEnable: TButton;
mnuDirectShow: TMenuItem;
SetProcessDelay: TMenuItem;
SetProcessDelayTo5ms: TMenuItem;
SetProcessDelayTo10ms: TMenuItem;
SetProcessDelayTo15ms: TMenuItem;
SetProcessDelayTo20ms: TMenuItem;
SetProcessDelayTo50ms: TMenuItem;
SetProcessDelayTo0ms: TMenuItem;
txtNativeVideoSize: TLabel;
Label3: TLabel;
procedure SetProcessDelayToMsClick(Sender: TObject);
procedure mnuDirectShowOptionsClick(Sender: TObject);
procedure SetupDirectShowExit(Sender: TObject);
procedure mnuSetupCardClick(Sender: TObject);
procedure SetupCardExit(Sender: TObject);
procedure mnuSetupFilesClick(Sender: TObject);
procedure SetupFilesExit(Sender: TObject);
procedure mnuSetupDVB-T2Click(Sender: TObject);
procedure SetupDVB-T2Exit(Sender: TObject);
procedure mnuSetupMiscellaneousClick(Sender: TObject);
procedure SetupMiscellaneousExit(Sender: TObject);
procedure mnuSetupRemoteClick(Sender: TObject);
private
FFullScreenMode: Boolean;
FFullScreenForm: TForm;
FStateMachineAutoEcm: Integer;
FRichViewEventStart: array of TDateTime;
FRichViewEventEnd: array of TDateTime;
FSetupDirectShow: TfrmSetupDirectShow;
procedure WMUser(var Msg: TMessage); message WM_USER;
procedure WMRemote(var Msg: TMessage); message WM_REMOTE;
procedure WMSignalling(var Msg: TMessage); message WM_SIGNALLING;
procedure FullScreenMode(FullScreen: Boolean);
procedure RichViewJump(Sender: TObject; id: Integer);
function StartRecordingProgram: Byte;
function StartRecording(Pids: array of TPidList): Byte;
function GetRecordingPids(var PidsRecord: TPidLists): Boolean;
function TransponderDisplay: Boolean;
procedure ReadFavourites;
procedure ReadLists;
procedure ReadTransponders(TransponderFile: string);
procedure EventUpdate;
procedure TransponderChanged;
procedure TransponderChangedTuner;

```

```

    procedure ProgramChanged(UseCurrentSettings: Boolean);
    procedure PidChanged;
    procedure Scan;
    procedure VolumeUpdate(Sender: TObject);
    procedure DirectShowFiltersAddToMenu;
    procedure DirectShowGraphsAddToMenu;
public
    { Public declarations }
    PrevSliderPosition: Integer;
end;
var
    frmOne: TfrmOne;

implementation
{$R *.DFM}
uses
    adCpuUsage, DvbDirectShow2, DvbDirectShow, DvbStreamBuffering,
    FlexCopGeneral, FlexCopDvbStreamBuffering, FlexCopInterface,
    FlexCopI2c, FlexCopIoControl, FlexCopRegisters,
    GOTVIEW_USB_2_MASTERHD_3Interface, GOTVIEW_USB_2_MASTERHD_3I2c,
    GOTVIEW_USB_2_MASTERHD_3IoControl, GOTVIEW_USB_2_MASTERHD_3Registers,
    Stv0299bRegisters, Tsa5059Registers, Registry, SyncObjs, SysUtils;
const
    Version = $0007;
    SubVersion = '3';
    Build = $20060427;
type
    TOperation = (otNoOp, otNormal, otError); // Тип обладнання
using
    TDataThread = class(TThread)
    private
        FHasStopped: Boolean; // Прапор роботи
        FProgrammedStop: Boolean; // Якщо true проходе завершення роботи
        FPacketBuffers: Word; // Номер буфера
        FFileStream: TFileStream;
        FFileStreamSize: Int64;
        FFileStreamStop: Boolean; stopped
        FFileStreamDelay: Word;
        FOperation: TOperation; // Тип операції
        FStreamData: PDvbTransportPackets; // поточний показчик
        procedure DecodeTransportStreamData;
        procedure DescrambleFFDeCsa;
    protected
        procedure Execute; override;
    end;
    TRcThread = class(TThread)
    private
        HasStopped: Boolean; // прапор
        ProgrammedStop: Boolean;
    protected
        procedure Execute; override;
    end;
    TDFilterProc = procedure(hFilter: DWORD; Len: Cardinal; Buf:
PByteArray);
    cdecl;
    TFilter = record
        Name: string; // ім'я фільтру
        Active: Boolean; // прапор активності фільтру
        Pid: Word;
        FilterId: Word; // ID фільтра
        CallbackFunction: TDFilterProc; // Функція обратного виклику
    end;
    PTFilters = ^TFilters;

```

```

TFilters = record
  Filters: array[1..255] of TFilter; // масив
  CrossReference: array[0..$1FFF] of Byte;
  Active: Word; // Число активного фільтру
  ActiveText: string; // Текстовий запис фільтра
end;
procedure GetAndWriteDVB-T2Setup(SetupForm: TfrmSetupDVB-T2;
ForceWrite:
  Boolean); forward;
procedure SetDVB-T2Setup(SetupForm: TfrmSetupDVB-T2); forward
  procedure GetAndWriteMiscellaneousSetup(SetupForm:
TfrmSetupMiscellaneous;
  ForceWrite: Boolean); forward;
procedure SetMiscellaneousSetup(SetupForm: TfrmSetupMiscellaneous);
forward
  procedure GetAndWriteRemoteSetup(SetupForm: TfrmSetupRemote;
ForceWrite:
  Boolean); forward;
procedure SetRemoteSetup(SetupForm: TfrmSetupRemote); forward
  procedure GetAndWriteRecordings(SetupForm: TfrmRecording; ForceWrite:
  Boolean); forward;
procedure GetAndWriteDirectShowSetup(SetupForm: TfrmSetupDirectShow;
ForceWrite:
  Boolean); forward;
procedure SetDirectShowSetup(SetupForm: TfrmSetupDirectShow); forward
  procedure GetAndWriteCardSetup(SetupForm: TfrmSetupCard; ForceWrite:
Boolean);
  forward;
procedure SetCardSetup(SetupForm: TfrmSetupCard); forward
  procedure GetAndWriteFilesSetup(SetupForm: TfrmSetupFiles;
ForceWrite:
  Boolean); forward;
procedure SetFilesSetup(SetupForm: TfrmSetupFiles); forward
function SetRecordings(SetupForm: TfrmRecording): Boolean; forward
  procedure ReadRecordings; forward;
procedure WriteRecordings; forward;
const
  D_SCAN_CURRENT_CAT = $01020031;
  D_START_OSD = $01020040;
  D_OSD_DRAWBLOCK = $01020041;
  D_OSD_SETFONT = $01020042;
  D_OSD_TEXT = $01020043;
  D_SEND_OSD_KEY = $01020044;
  D_STOP_OSD = $01020049;
  D_DVB_COMMAND = $01020060;
  D_GET_VERSION = $01020100;
  D_GET_TRANSPONDER = $01020000;
  D_SET_TRANSPONDER = $01020001;
  D_GET_PROGRAM = $01020010;
  D_SET_PROGRAM = $01020011;
  D_RESCAN_PROGRAM = $01020012;

type
  PDVB_COMMAND = ^TDVB_COMMAND;
  TDVB_COMMAND = record
    CmdLength: Word;
    CmdBuffer: array[0..31] of Byte;
  end;
  PProgramNumberParam = ^TProgramNumberParam;
  TProgramNumberParam = record
    RealNumber: Integer;
    VirtNumber: Integer;
  end;
end;

```

```

PStartFilterParam = ^TStartFilterParam;

TStartFilterParam = record
  DLLIdentifier: Word;
  Filter_ID: Word;
  Pid: Word;
  Name: array[0..31] of Byte;
  CallAddress: LongInt;
  Running_ID: Integer;
end;

TPIDFilters = record
  FilterName: array[0..4] of Char;
  FilterId: Byte;
  PID: Word;
end;

TCA_System82 = record
  CA_Type: Word;
  ECM: Word;
  EMM: Word;
  Provider_Id: Dword;
end;

PProgram82 = ^TProgram82;

TProgram82 = record
  Name: array[00..29] of Char;
  Provider: array[00..29] of Char;
  Country: array[00..29] of Char;
  Freq: Dword;
  PType: Byte;
  Voltage: Byte;
  Afc: Byte;
  DVB-T2: Byte;      Symbolrate: Word;
  Qam: Word;         Fec: Word;
  Norm: Byte;        Tp_id: Word;
  Video_pid: Word;   Audio_pid: Word;
  TeleText_pid: Word; PMT_pid: Word;
  PCR_pid: Word;     ECM_pid: Word;
  SID_pid: Word;     AC3_pid: Word;
  TVType: Byte;      ServiceTyp: Byte;
  CA_ID: Byte;        Temp_Audio: Word;
  FilterNr: Word;     Filters: array[00..31] of TPIDFilters;
  CA_Nr: Word;         CA_System82: array[0..31] of TCA_System82;
  CA_Country: array[0..5] of Char;
  Marker: Byte;      Link_TP: Word;
  Link_SID: Word;    PDynamic: Byte;
  Extern_Buffer: array[00..15] of Char;
end;

TDInitPluginProc =
  procedure(MDInstance: LongWord; MDWnd: HWND;
    Log_Set: Bool; DLL_ID: Integer; HotKey: PChar; Vers: PChar;
    var ReturnValue: Integer); cdecl;
TDGetPluginName =
  procedure(Buf: PByteArray); cdecl;
TDProcessPluginMenuCommandProc =
  procedure(MenuID: integer); cdecl;
TDProcessChannelChange =
  procedure(prg: TProgram82); cdecl;
TDExitPluginProc =
  procedure(MDInstance: LongWord; MDWnd: HWND; Log_Set: Bool); cdecl;
TDProcessHotKeyProc =

```

```

    procedure; cdecl;
TDFilterCloseProc =
    procedure(MDInstance: LongWord); cdecl;
TDProcessRecPlayProc =
    procedure(Mode: Integer); cdecl;
TMdPlugin = record
    DllIdentifier: THandle; // Показчик на DLL
    Name: string; // Ім'я
    HotKey: Char;
    PacketSize188: Boolean; // розмір пакету
    Filters: TFilters;
    PluginMenu: HMENU; // меню плагіна
    GetPluginName: TDGetPluginName;
    InitPlugin: TDInitPluginProc;
    ProcessPluginMenuCommand: TDProcessPluginMenuCommandProc;
    ProcessChannelChange: TDProcessChannelChange;
    ExitMDPlugin: TDExitPluginProc;
    ProcessHotKey: TDProcessHotKeyProc;
    ProcessFilterClose: TDFilterCloseProc;
    ProcessRecPlay: TDProcessRecPlayProc;
end;
TSetKeyProc = procedure(Key: PByteArray; KeyStruct: PByteArray);
cdecl;
TCSADecryptProc = procedure(KeyStruct: PByteArray; Encrypted:
PByteArray);
cdecl;

TFFDeCsaCluster = packed record
    StartBuffer: Pointer;
    EndBuffer: Pointer;
end;
PFFDeCsaClusters = ^TFFDeCsaClusters;
TFFDeCsaClusters = array[0..CDvbPacketVSync] of TFFDeCsaCluster;
TFFSetKeyProc = function(Even, Odd, KeySet: PByteArray): Integer;
stdcall;
TFFDecryptProc = function(Cluster: PFFDeCsaClusters; KeySet:
PByteArray):
    Integer; stdcall;
TFFKeySizeProc = function: Integer; stdcall;
TFFParallelismProc = function: Integer; stdcall;
TRecording = (rtIdle, rtStartRecording, rtRecording,
rtStopRecording);
const
    COsdSizeBitmapHeader = sizeof(BITMAPINFO) + 256 * sizeof(TRGBQuad);
    COsdTransparentColor = $00112211;
    COsdTransparentColorRGB: TRGBQuad = (
        rgbBlue: ((COsdTransparentColor shr 0) and $FF);
        rgbGreen: ((COsdTransparentColor shr 8) and $FF);
        rgbRed: ((COsdTransparentColor shr 16) and $FF);
        rgbReserved: 0);
    COsdWidth = 500;
    COsdHeight = 500;
type
    PTMainWindowOsdData = ^TMainWindowOsdData;
    TMainWindowOsdData = array[0..(COsdWidth * COsdHeight) - 1] of
TRGBQuad;
    BitmapDef = record
        Header: array[0..COsdSizeBitmapHeader] of Byte;
        Info: PBitmapInfo;
        Handle: HBITMAP;
        DC: HDC;
        Data: PTMainWindowOsdData;
end;
end;

```

```

var
  OsdBitmaps: array[0..4] of BitmapDef;
  OsdTimeout: Integer;
var
  AppName: string;
  ExeDirectory: string;
  GUDPClient : TIdUDPClient;
  GUDPBuffer : TIdBytes;
  GUDPPointer : PByte;
  GUDPLastIndex: Integer;
  GUDPIndex : Integer;
  IsSlave: Boolean;
  ThreadHandle: THandle;
  PacketThread: TDataThread;
  RemoteThread: TRCThread;
  CardHandle: THandle;
  KeySwapUpDown: Boolean;
type
  TCsaMethod = (cmNone, cmInternal, cmCsa, cmFfCsa);
var
  CsaMethod: TCsaMethod; // метод CSA
  CsaName: string; // поточний CSA
  CsaDLLIdentifier: Integer;
  SetKeyProc: TSetKeyProc;
  CSADecryptProc: TCSADecryptProc;
  KeyStruct: array[0..800] of Byte;
  Key: array[0..15] of Byte;
  KeysOdd: TBlock;
  KeysEven: TBlock;
  KeysPreviousOdd: TBlock;
  KeysPreviousEven: TBlock;
  FFSetKeyProc: TFFSetKeyProc;
  FFDecryptProc: TFFDecryptProc;
  FFKeySizeProc: TFFKeySizeProc;
  FFParallelismProc: TFFParallelismProc;
  FFCsaKeys: PByteArray;
  FFCsaKeySize: Integer;
  FFCsaClusters: TFFDeCsaClusters;

procedure OsdFillColor(Index: Byte; Color: TRGBQuad);
var
  Width: Integer;
  height: Integer;
begin
  if (Index > High(OsdBitmaps)) then
    Exit;
  for Width := 0 to COsdWidth - 1 do
    for Height := 0 to COsdHeight - 1 do
      OsdBitmaps[Index].Data[Width * COsdWidth + Height] := Color;
end;

procedure OsdClear;
begin
  if (DsOptions and CDirectShowMethodAlternative) <> 0 then
    DvbDirectShow2.DirectShowBlendImage(OsdBitmaps[0].DC, Rect(0, 0,
COsdWidth,
COsdHeight), 0.0, COsdTransparentColor)
  else
    DvbDirectShow.DirectShowGraph.DirectShowBlendImage(OsdBitmaps[0].DC,
Rect(0,
0, COsdWidth,
COsdHeight), 0.0, COsdTransparentColor);

```

```

end;

procedure OsdShowInfo;
var
  DisplayStr: string;
begin
  SetBkColor(OsdBitmaps[1].DC, clWhite);
  SetTextColor(OsdBitmaps[1].DC, clBlack);
  OsdFillColor(1, COsdTransparentColorRGB);
  DisplayStr := format(' %4.4d - %s ', [Favourite + 1, ProgramName]);
  TextOut(OsdBitmaps[1].DC, 200, 250, @DisplayStr[1],
Length(DisplayStr));
  if (DsOptions and CDirectShowMethodAlternative) <> 0 then
    DvbDirectShow2.DirectShowBlendImage(OsdBitmaps[1].DC, Rect(0, 0,
COsdWidth,
COsdHeight), 1.0, COsdTransparentColor)
  else

DvbDirectShow.DirectShowGraph.DirectShowBlendImage(OsdBitmaps[1].DC,
Rect(0,
0, COsdWidth,
COsdHeight), 1.0, COsdTransparentColor);
end;
function EnableShutdownPrivileges: Boolean;
var
  ProcessHandle: THandle;
  TokenHandle: THandle;
  Token: TTokenPrivileges;
  Luid: TLargeInteger;
  PreviousState: TTokenPrivileges;
  ReturnLength: Dword;
  VersionInfo: TOSVersionInfo;
begin
  Result := False;
  VersionInfo.dwOSVersionInfoSize := SizeOf(TOSVersionInfo);
  if not GetVersionEx(VersionInfo) then
    Exit;
  if not ((VersionInfo.dwPlatformId = VER_PLATFORM_WIN32_NT) and
(VersionInfo.dwMajorVersion >= 4)) then
    begin
      Result := True;
      Exit;
    end;
  ProcessHandle := GetCurrentProcess;
  if ProcessHandle <> 0 then
    begin
      if OpenProcessToken(ProcessHandle, TOKEN_ADJUST_PRIVILEGES or
TOKEN_QUERY,
TokenHandle) then
        begin
          if LookupPrivilegeValue('', 'SeShutdownPrivilege', Luid) then
            begin
              Token.PrivilegeCount := 1;
              Token.Privileges[0].Luid := Luid;
              Token.Privileges[0].Attributes := SE_PRIVILEGE_ENABLED;
              if AdjustTokenPrivileges(TokenHandle, False, Token,
SizeOf(PreviousState), PreviousState, ReturnLength) then
                Result := True
            end;
          end;
        end;
    end;
end;
function ChangeTimeSeparator(ToChange: string): string;

```

```

var
  ThePos: Integer;
begin
  if TimeSeparator <> ':' then
  begin
    ThePos := Pos(':', ToChange);
    while ThePos <> 0 do
    begin
      ToChange[ThePos] := TimeSeparator;
      ThePos := Pos(':', ToChange);
    end;
  end;
  Result := ToChange;
end;

procedure Shutdown;
var
  Flags: Integer;
begin
  Flags := EWX_POWEROFF or EWX_FORCE or EWX_FORCEIFHUNG;
  if EnableShutdownPrivileges then
    ExitWindowsEx(Flags, 0);
end;

procedure ToLog(LogString: string; Level: Byte);
var
  NewLog: string;
begin
  if ((Level and $80) <> 0) and
    Assigned(frmOne) and
    Assigned(frmOne.mmoDriver) then
    frmOne.mmoDriver.Lines.Add(LogString);
  if (Level and $0F) > LogLevel then
    Exit;
  if not Assigned(LogStream) then
    Exit;
  NewLog := FormatDateTime('YYYYMMDD"T"HHMMSS" "', Now) + LogString +
    #13#10;
  LogStream.Write(NewLog[1], Length(NewLog));
end;

procedure FilterUpdateCrossReference(Filter: PTFilters);
var
  FilterIndex: Integer;
begin
  Filter.Active := 0;
  Filter.ActiveText := '';
  for FilterIndex := Low(Filter.CrossReference) to
  High(Filter.CrossReference)
  do
    Filter.CrossReference[FilterIndex] := 0;
  for FilterIndex := Low(Filter.Filters) to High(Filter.Filters) do
    if Filter.Filters[FilterIndex].Active then
    begin
      Filter.CrossReference[Filter.Filters[FilterIndex].Pid] :=
FilterIndex;
      Inc(Filter.Active);
      if Filter.ActiveText = '' then
        Filter.ActiveText := format('%d',
[Filter.Filters[FilterIndex].Pid])
      else
        Filter.ActiveText := Filter.ActiveText + ' ' + format('%d',
[Filter.Filters[FilterIndex].Pid]);
    end;
  end;
end;

```

```

    end;
end;

procedure FilterRemoveByName(Filter: PTFilters; Name: string);
var
    FilterIndex: Integer;
begin
    ToLog('FilterRemoveByName: [' + Name + ']', $5);
    FilterLock.Acquire;
    try
        for FilterIndex := Low(Filter.Filters) to High(Filter.Filters) do
            begin
                Name := LowerCase(Name);
                if Name = 'video' then
                    PidVideo := 0;
                if Name = 'audio' then
                    PidAudio := 0;
                if Name = 'pmt' then
                    PidPmt := 0;
                if Name = 'pcr' then
                    PidPcr := 0;
                if Name = 'ecm' then
                    PidEcm := 0;
                if Name = 'teletext' then
                    PidTeletext := 0;
                if Name = 'subtitle' then
                    PidSubtitle := 0;
                if LowerCase(Filter.Filters[FilterIndex].Name) = Name then
                    begin
                        Filter.Filters[FilterIndex].Active := False;
                        FilterUpdateCrossReference(Filter);
                    end;
            end;
        finally
            FilterLock.Release;
        end;
    end;

function DvbSetFilter(Filter: PTFilters; Pid: Word; FilterId: Word;
FilterProc:
    Pointer; Name: PChar): Dword; stdcall;
var
    FilterIndex: Integer;
    Valid: Boolean;
    PString: string;
begin
    ToLog(format('DvbSetFilter %d (%s).', [Pid, Name]), $04);
    PString := Name;
    if PString <> '' then
        FilterRemoveByName(Filter, PString);
    Result := 0;
    if Pid > High(Filter.CrossReference) then
        Exit;
    FilterLock.Acquire;
    try
        Valid := False;
        // Пошук
        if Filter.CrossReference[Pid] <> 0 then
            begin
                FilterIndex := Filter.CrossReference[Pid];
                Valid := True;
            end
        else

```

```

begin
  FilterIndex := Low(Filter.Filters);
  repeat
    if Filter.Filters[FilterIndex].Active then
      Inc(FilterIndex)
    else
      Valid := True;
    until Valid or (FilterIndex > High(Filter.Filters));
  end;
  if Valid then
    begin
      Filter.Filters[FilterIndex].Pid := Pid;
      Filter.Filters[FilterIndex].FilterId := FilterId;
      Filter.Filters[FilterIndex].CallBackFunction := FilterProc;
      Filter.Filters[FilterIndex].Name := Name;
      if LowerCase(Name) = 'video' then
        PidVideo := Pid;
      if LowerCase(Name) = 'audio' then
        PidAudio := Pid;
      if LowerCase(Name) = 'pmt' then
        PidPmt := Pid;
      if LowerCase(Name) = 'pcr' then
        PidPcr := Pid;
      if LowerCase(Name) = 'ecm' then
        PidEcm := Pid;
      if LowerCase(Name) = 'teletext' then
        PidTeletext := Pid;
      if LowerCase(Name) = 'subtitle' then
        PidSubtitle := Pid;
      Filter.Filters[FilterIndex].Active := True;
      FilterUpdateCrossReference(Filter);
      Result := FilterIndex;
    end;
  finally
    FilterLock.Release;
  end;
end;

procedure MdPluginExit(Plugin: Word);
var
  WriteLog: Boolean;
  Index: Integer;
begin
  ToLog(format('MdPlugExit: [%d]', [Plugin]), $5);
  try
    WriteLog := True;
    if MdPlugin[Plugin].DllIdentifier <> 0 then
      if Assigned(@MdPlugin[Plugin].ExitMdPlugin) then
        begin
          FilterLock.Acquire;
          try
            // Видалення фільтрів
            for Index := Low(AppFilters.Filters) to
High(AppFilters.Filters) do
              MdPlugin[Plugin].Filters.Filters[Index].Active := False;
              FilterUpdateCrossReference(@MdPlugin[Plugin].Filters);
            finally
              FilterLock.Release;
            end;
          MdPlugin[Plugin].ExitMdPlugin(HInstance, frmOne.Handle,
WriteLog);
        end;
      except

```

```

    end;
end;

procedure CleanupMdPlugins;
var
    Index: Word;
begin
    ToLog('CleanupMdPlugins', $5);
    try
        try
            if MdPlugins > 0 then
                for Index := 0 to MdPlugins - 1 do
                    MdPluginExit(Index);
                if MdPlugins > 0 then
                    for Index := 0 to MdPlugins - 1 do
                        DestroyMenu(MdPlugin[Index].PluginMenu);
                    finally
                        MdPlugins := 0;
                    end;
                except
                    end;
            end;
end;

procedure DOnCloseFilters;
var
    Index: Word;
begin
    ToLog('DOnCloseFilters', $5);
    try
        if MdPlugins > 0 then
            for Index := 0 to MdPlugins - 1 do
                if MdPlugin[Index].DllIdentifier <> 0 then
                    if Assigned(@MdPlugin[Index].ProcessFilterClose) then
                        MdPlugin[Index].ProcessFilterClose(HInstance);
            except
                end;
        end;
end;

procedure DOnChannelChange(Prg: TProgram82);
var
    Index: Word;
begin
    ToLog('DOnChannelChange', $5);
    try
        if MdPlugins > 0 then
            for Index := 0 to MdPlugins - 1 do
                if MdPlugin[Index].DllIdentifier <> 0 then
                    if Assigned(@MdPlugin[Index].ProcessChannelChange) then
                        MdPlugin[Index].ProcessChannelChange(Prg);
            except
                end;
        end;
end;

procedure SetCSAKeys(Command: array of Byte);
var
    i: Integer;
    j: Integer;
    Change: Boolean;
begin
    if CsaMethod = cmNone then
        Exit;
    if Command[4] = 1 then
        OddKeyFound := True

```

```

else
  EvenKeyFound := True;
for i := 0 to 3 do
begin
  Key[(Command[4]) * 8 + i * 2 + 0] := Command[6 + i * 2 + 1];
  Key[(Command[4]) * 8 + i * 2 + 1] := Command[6 + i * 2 + 0];
  if CsaMethod in [cmInternal, cmFfCsa] then
  begin
    if Command[4] = 0 then
      for j := 0 to 3 do
      begin
        KeysEven[j * 2 + 0] := Command[6 + j * 2 + 1];
        KeysEven[j * 2 + 1] := Command[6 + j * 2 + 0];
      end
    else
      for j := 0 to 3 do
      begin
        KeysOdd[j * 2 + 0] := Command[6 + j * 2 + 1];
        KeysOdd[j * 2 + 1] := Command[6 + j * 2 + 0];
      end;
    end;
  end;
end;
if OddKeyFound and EvenKeyFound then
begin
  KeyFound := True;
  if CsaMethod = cmInternal then
  begin
    Change := False;
    for j := 0 to 7 do
    begin
      if KeysOdd[j] <> KeysPreviousOdd[j] then
        Change := True;
      if KeysEven[j] <> KeysPreviousEven[j] then
        Change := True;
    end;
    if Change then
    begin
      CSASetKeys(KeysOdd, KeysEven);
      for j := 0 to 7 do
      begin
        KeysPreviousOdd[j] := KeysOdd[j];
        KeysPreviousEven[j] := KeysEven[j];
      end;
    end;
  end;
else
  begin
    if CsaMethod = cmCsa then
      SetKeyProc(@Key[0], @KeyStruct[0]);
    if CsaMethod = cmFfCsa then
      FFSetKeyProc(@KeysEven[0], @KeysOdd[0], FFCsaKeys);
  end;
end;
end;

procedure TfrmOne.WMCommand(var Msg: TWMMCommand);
var
  Index: Word;
begin
  ToLog(format('WMCommand: %d', [Msg.ItemId]), $5);
  try
    if MdPlugins > 0 then
      for Index := 0 to MdPlugins - 1 do

```

```

        if MdPlugin[Index].DllIdentifier <> 0 then
            if Assigned(@MdPlugin[Index].ProcessPluginMenuCommand) then
                MdPlugin[Index].ProcessPluginMenuCommand(Msg.ItemId);
        except;
        end;
        inherited;
    end;
end;

procedure UpdateProgram82(Program82: PProgram82);
var
    Index: Integer;
    EmmCaSystemId: Word;
    EmmCaPid: Word;
    EmmIndex: Word;
    DummyP: TDvbPids;
    DummyW: Word;
    Language: TDvbLanguages;
    PidsEcm: TDvbPids;
    CasId: TDvbPids;
    ValidCount: Integer;
    Nothing: Boolean;
    LogString: string;
    PrgName: string;
    Error: Integer;
    ProcessString: string;
begin
    StrPCopy(Program82.Name, Copy(ProgramName, 1,
    SizeOf(Program82.Name)));
    Program82.Freq := Frequency;
    Program82.Video_pid := PidVideo;
    Program82.Audio_pid := PidAudio;
    Program82.Teletext_pid := PidTeletext;
    Program82.PCR_pid := PidPcr;
    Program82.PMT_pid := PidPmt;
    Program82.ECM_pid := PidEcm;
    Program82.CA_ID := 0;
    Program82.SID_pid := ServiceNumber;
    Program82.CA_NR := 0;
    if TransponderHasChanged = $0000 then
        begin
            ToLog('UpdateProgram82 (not using realtime data yet)', $3);
            Exit;
        end;
    LogString := 'UpdateProgram82 (' + ProgramName + '): ECM: ';
    ValidCount := 5;
    Nothing := True;
    repeat
        if DvbGetProgramInfo($FF, ServiceNumber, DummyW, DummyW, DummyP,
        DummyP,
        DummyP, DummyP, Language, Language, PidsEcm, CasId, PrgName) then
            begin
                if PrgName <> '' then
                    ProgramName := PrgName;
                    StrPCopy(Program82.Name, Copy(ProgramName, 1,
                    SizeOf(Program82.Name)));
                    Index := 0;
                    while ((PidsEcm[Index] <> 0) and (Index <=
                    High(Program82.CA_System82)))
                    do
                        begin
                            Program82.CA_System82[Index].CA_Type := CasId[Index];
                            Program82.CA_System82[Index].ECM := PidsEcm[Index];
                            Program82.CA_System82[Index].EMM := 0;

```

```

        Program82.CA_System82[Index].Provider_Id := ServiceNumber;
        LogString := Logstring + format('%d ', [PidsEcm[Index]]);
        Inc(Index);
    end;
    Program82.CA_NR := Index;
    ValidCount := 1;
    Nothing := False;
end
else
    Sleep(10);
    Dec(ValidCount);
until ValidCount = 0;
if Nothing then
    LogString := LogString + 'none ';
    LogString := LogString + ' EMM: ';
    Nothing := True;
    if frmOne.chkEmmManual.Checked then
    begin
        if Program82.CA_NR > 0 then
        begin
            Nothing := False;
            ProcessString := frmOne.edtEmmPids.Text;
            ProcessString := Trim(ProcessString);
            while ProcessString <> '' do
            begin
                ProcessString := Trim(ProcessString);
                Val(ProcessString, EmmCaPid, Error);
                if Error <> 1 then
                begin
                    if Error = 0 then
                        ProcessString := ''
                    else
                        Delete(ProcessString, 1, Error - 1);
                        Index := Program82.CA_NR;
                        Program82.CA_System82[Index].ECM := 0;
                        Program82.CA_System82[Index].Provider_Id := 0;
                        Program82.CA_NR := Index + 1;
                        Program82.CA_System82[Index].EMM := EmmCaPid;
                        LogString := Logstring + format('%d ', [EmmCaPid]);
                    end
                else
                    Delete(ProcessString, 1, 1);
                end;
            end;
        end;
    end
else
    begin
        frmOne.edtEmmPids.Text := '';
        if Program82.CA_NR > 0 then
        begin
            Nothing := False;
            EmmIndex := 0;
            while (DvbFilterGetEmm(EmmIndex, EmmCaSystemId, EmmCaPid) and
                (Program82.CA_NR <= High(Program82.CA_System82))) do
            begin
                Index := Program82.CA_NR;
                Program82.CA_System82[Index].ECM := 0;
                Program82.CA_System82[Index].Provider_Id := 0;
                Program82.CA_NR := Index + 1;
                Program82.CA_System82[Index].CA_Type := EmmCaSystemId;
                Program82.CA_System82[Index].EMM := EmmCaPid;
                LogString := Logstring + format('%d ', [EmmCaPid]);
            end;
        end;
    end;
end

```



```

if PidPMT <> PProgram82(Msg.LParam)^.PMT_pid then
begin
  PidPMT := PProgram82(Msg.LParam)^.PMT_pid;
  frmOne.mskPmt.Text := format('%4.4d', [PidPMT]);
  Change := True;
end;
if PidEcm <> PProgram82(Msg.LParam)^.ECM_pid then
begin
  PidEcm := PProgram82(Msg.LParam)^.ECM_pid;
  frmOne.cmbEcmPids.Text := format('%4.4d', [PidEcm]);
  Change := True;
end;
if Change then
  SendMessage(frmOne.Handle, WM_REMOTE,
    Integer(MessageSettings), 0);
end;
end;
D_GET_PROGRAM_NUMBER:
begin
  PProgramNumberParam(Msg.LParam)^.RealNumber := ServiceNumber;
  PProgramNumberParam(Msg.LParam)^.VirtNumber := ServiceNumber;
end;
D_START_FILTER:
begin
  StartFilterParam := PStartFilterParam(Msg.LParam);
  if (StartFilterParam^.DLLIdentifier >= MdPlugins) then
    Exit;
  StartFilterParam^.Running_ID :=
((StartFilterParam^.DLLIdentifier and
  $FF) shl 8)
  or
  DvbSetFilter(@MdPlugin[StartFilterParam^.DLLIdentifier].Filters,
  StartFilterParam^.Pid, StartFilterParam^.Filter_ID,
  Pointer(StartFilterParam^.CallAddress),
  PChar(@(StartFilterParam^.Name[0])));
end;
D_STOP_FILTER:
begin
  Plugin := (Msg.LParam and $FF00) shr 8;
  if Plugin >= MdPlugins then
    Exit;
  DvbStopFilter(@MdPlugin[Plugin].Filters, Msg.LParam and $FF);
end;
D_GET_VERSION:
begin
  StrPCopy(PChar(Msg.LParam), 'MD-API Version 01.02 Root
254376');
end;
D_DVB_COMMAND:
begin
  if PDVB_COMMAND(msg.LParam)^.CmdLength = 7 then
    for Index := 4 to 13 do
      SAA_COMMAND[Index] :=
PDVB_COMMAND(Msg.LParam)^.CmdBuffer[Index];
      SetCSAKeys(SAA_COMMAND);
    end;
end;
end;

procedure TfrmOne.WMSignalling(var Msg: TMessage);
begin
  case Msg.WParam of
    CSIGNAL_EVENT:

```

```

begin
  if (Msg.LParam = ServiceNumber) or (ServiceNumber = 0) then
    begin
      ToLog(format('WMSignalling - Event (Id: %d)', [Msg.LParam]),
$5);
      UpdateEvent := True;
    end
  else
    ToLog(format('WMSignalling - Event (Id: %d) - not handled',
      [Msg.LParam]), $5);
  end;
CSignalPat:
begin
  ToLog(format('WMSignalling - PAT (Version: %d)', [Msg.LParam]),
$5);
  UpdatePAT := True;
end;
CSignalCat:
begin
  ToLog(format('WMSignalling - CAT (Version: %d)', [Msg.LParam]),
$5);
  UpdateCAT := True;
end;
CSignalPmt:
begin
  if (Msg.LParam = ServiceNumber) or (ServiceNumber = 0) then
    begin
      ToLog(format('WMSignalling - PMT (Id: %d)', [Msg.LParam]),
$5);
      UpdatePmt := True;
      UpdatePat := True;
    end
  else
    ToLog(format('WMSignalling - PMT (Id: %d) - not handled',
      [Msg.LParam]), $5);
  end;
CSignalPid: case Msg.LParam of
  CPidNIT:
    begin
      ToLog('WMSignalling - NIT', $5);
      UpdateProgram := True;
    end;
  else
    ToLog(format('WMSignalling - PID: %d', [Msg.LParam]), $5);
  end;
else
  ToLog(format('WMSignalling - [%d / %d]', [Msg.WParam, Msg.LParam]),
$5);
end;
end;

```

```

procedure TDataThread.DescrambleFfDeCsa;

```

```

var

```

```

  StreamPacket: Word;
  Pid: Word;
  Plugin: Word;
  Processed: Integer;
  ProcessIndex: Integer;
  ProcessIt: Boolean;

```

```

begin

```

```

  if (not FilteringOff) and
    (not FiltersOff) then
    begin

```

```

if (CsaMethod = cmFfCsa) then
begin
  ProcessIndex := 0;
  for StreamPacket := 0 to CDvbPacketVSync - 1 do
  begin
    if DvbGetPid(@FStreamData[StreamPacket], Pid) then
    begin
      ProcessIt := False;
      if MdPlugins > 0 then
        for Plugin := 0 to MdPlugins - 1 do
          if MdPlugin[Plugin].Filters.CrossReference[Pid] <> 0 then
          begin
            ProcessIt := True;
            Break;
          end;
        if not ProcessIt then
          if AppFilters.CrossReference[Pid] <> 0 then
            ProcessIt := True;
        if Processit and ((FStreamData[StreamPacket][3] and $80) <>
0) then
        begin
          if Pid <> 0 then
          begin
            if Pid = PidVideo then
              VideoCheckScrambled := True;
            if Pid = PidAudio then
              AudioCheckScrambled := True;
          end;
        end
      else
        ProcessIt := False;
      if ProcessIt then
      begin
        FFCsaClusters[ProcessIndex].StartBuffer :=
          @FStreamData[StreamPacket][0];
        FFCsaClusters[ProcessIndex].EndBuffer :=
          @FStreamData[StreamPacket][187];
        Inc(ProcessIndex);
      end;
    end;
  end;
  FFCsaClusters[ProcessIndex].StartBuffer := nil;
  FFCsaClusters[ProcessIndex].EndBuffer := nil;
  if KeyFound then
  repeat
    try
      try
        Processed := FFDecryptProc(@FFCsaClusters, FFCsaKeys);
      finally
        Set8087CW(Default8087CW);
      end;
    except
      Processed := 0;
    end;
  until Processed = 0;
  end;
end;
end;

procedure TfrmOne.ReadSettings;
var
  Default: string;
  Value: Integer;

```

```

Error: Integer;
Hour: Word;
Minute: Word;
Second: Word;
Msec: Word;
TheTime: TDateTime;
DeltaTime: TDateTime;
CheckTime: TDateTime;
begin
  ToLog('ReadSettings', $5);
  Default := GetParameter('Settings', 'FormTop', '-4');
  Val(Default, Value, Error);
  if Error = 0 then
    Self.Top := Value;
  Default := GetParameter('Settings', 'FormLeft', '-4');
  Val(Default, Value, Error);
  if Error = 0 then
    Self.Left := Value;
  Default := GetParameter('Settings', 'FormWidth', '1032');
  Val(Default, Value, Error);
  if Error = 0 then
    Self.Width := Value;
  Default := GetParameter('Settings', 'FormHeight', '748');
  Val(Default, Value, Error);
  if Error = 0 then
    Self.Height := Value;
  Default := GetParameter('Settings', 'Transponder', '0');
  Val(Default, Transponder, Error);
  Error := -1;
  if ParameterFavourite <> '' then
    begin
      Val(ParameterFavourite, Value, Error);
      cmbLists.ItemIndex := Value;
      ParameterFavourite := '';
    end;
  if Error <> 0 then
    begin
      Default := GetParameter('Settings', 'FavouriteList', '0');
      Val(Default, Value, Error);
      cmbLists.ItemIndex := Value;
    end;
  Default := GetParameter('Settings', 'Service', '0');
  cmbServices.Text := Default;
  Default := LowerCase(GetParameter('Settings', 'TransponderDisplay',
'No'));
  if Default = 'yes' then
    chkTransponderDisplay.Checked := True
  else
    chkTransponderDisplay.Checked := False;
  chkTransponderDisplayClick(nil);
  Default := LowerCase(GetParameter('Settings', 'AutoECM', 'Yes'));
  if Default = 'yes' then
    chkAutoECM.Checked := True
  else
    chkAutoECM.Checked := False;
  if ParameterShutdown <> '' then
    begin
      try
        ParameterShutdown := ChangeTimeSeparator(ParameterShutdown);
        TheTime := StrToTime(ParameterShutdown) +
TimeCorrectionParameters;
        dtShutdown.Time := TheTime;
        ParameterShutdown := '';

```

```

        chkShutdown.Checked := True;
        Error := 0;
    except
    end;
end;
if Error <> 0 then
begin
    Default := GetParameter('Settings', 'ShutdownTime', '00' +
TimeSeparator +
    '00' + TimeSeparator + '00');
    try
        Default := ChangeTimeSeparator(Default);
        TheTime := StrToTime(Default);
        dtShutdown.Time := TheTime;
    except
        dtShutdown.Time := dtStopRecording.Time + EncodeTime(0, 1, 0, 0);
    end;
end;
if ParameterExit <> '' then
begin
    try
        ParameterExit := ChangeTimeSeparator(ParameterExit);
        TheTime := StrToTime(ParameterExit) + TimeCorrectionParameters;
        dtExit.Time := TheTime;
        dtExit.Enabled := True;
        ParameterExit := '';
    except
    end;
end;
ReadRecordings;
end;

procedure TfrmOne.VolumeUpdate(Sender: TObject);
begin
    if not Mixer.VolumeAvailable then
    begin
        imgVolumeUp.Visible := False;
        imgVolumeDown.Visible := False;
        imgVolumeOn.Visible := False;
        imgVolumeOff.Visible := False;
    end
    else
    begin
        if Mixer.VolumeMute then
        begin
            imgVolumeOn.Visible := True;
            imgVolumeOff.Visible := False;
        end
        else
        begin
            imgVolumeOn.Visible := False;
            imgVolumeOff.Visible := True;
        end;
    end;
end;

procedure TfrmOne.imgVolumeOffClick(Sender: TObject);
var
    Mute: Boolean;
begin
    if not Mixer.VolumeAvailable then
        Exit;
    Mute := False;

```

```

    if Sender = nil then
        Mute := Mixer.VolumeMute;
    if Sender = imgVolumeOff then
        Mute := True;
    if Sender = imgVolumeOn then
        Mute := False;
    Mixer.VolumeMute := Mute;
    VolumeUpdate(imgVolumeOff);
end;

procedure TfrmOne.imgVolumeUpClick(Sender: TObject);
begin
    imgVolumeOffClick(imgVolumeOn);
    Mixer.VolumeUp;
end;

procedure TfrmOne.imgVolumeDownClick(Sender: TObject);
begin
    imgVolumeOffClick(imgVolumeOn);
    Mixer.VolumeDown;
end;

procedure TfrmOne.btnPositionerStopClick(Sender: TObject);
var
    DVB-T2Data: TDVB-T2Data;
begin
    DVB-T2Data[0] := 0;
    DVB-T2Data[1] := $30;
    DVB-T2Data[2] := $60;
    if UseFlexCop then
        FlexCopDVB-T2Command(CardHandle, DVB-T2Repeats, CDVB-T2Command, 3,
            DVB-T2Data)
    else
        GOTVIEW_USB_2_MASTERHD_3DVB-T2Command(CardHandle, DVB-T2Repeats,
            CDVB-T2Command, 3,
            DVB-T2Data);
end;

// спос позиції
procedure TfrmOne.btnPositionerResetClick(Sender: TObject);
var
    DVB-T2Data: TDVB-T2Data;
begin
    DVB-T2Data[0] := 0;
    DVB-T2Data[1] := $30;
    DVB-T2Data[2] := $00;
    if UseFlexCop then
        FlexCopDVB-T2Command(CardHandle, DVB-T2Repeats, CDVB-T2Command, 3,
            DVB-T2Data)
    else
        GOTVIEW_USB_2_MASTERHD_3DVB-T2Command(CardHandle, DVB-T2Repeats,
            CDVB-T2Command, 3,
            DVB-T2Data);
end;

procedure TfrmOne.btnPositionerGotoClick(Sender: TObject);
var
    DVB-T2Data: TDVB-T2Data;
begin
    DVB-T2Data[0] := 0;
    DVB-T2Data[1] := $30;
    DVB-T2Data[2] := $6B;
    DVB-T2Data[3] := spnSatellitePosition.Value;

```

```

    if UseFlexCop then
        FlexCopDVB-T2Command(CardHandle, DVB-T2Repeats, CDVB-T2Command, 4,
            DVB-T2Data)
    else
        GOTVIEW_USB_2_MASTERHD_3DVB-T2Command(CardHandle, DVB-T2Repeats,
            CDVB-T2Command, 4,
            DVB-T2Data);
end;

procedure TfrmOne.btnPositionerStoreClick(Sender: TObject);
var
    DVB-T2Data: TDVB-T2Data;
begin
    DVB-T2Data[0] := 0;
    DVB-T2Data[1] := $30;
    DVB-T2Data[2] := $6A;
    DVB-T2Data[3] := spnSatellitePosition.Value;
    if UseFlexCop then
        FlexCopDVB-T2Command(CardHandle, DVB-T2Repeats, CDVB-T2Command, 4,
            DVB-T2Data)
    else
        GOTVIEW_USB_2_MASTERHD_3DVB-T2Command(CardHandle, DVB-T2Repeats,
            CDVB-T2Command, 4,
            DVB-T2Data);
    if spnSatellitePosition.Value < spnSatellitePosition.MaxValue then
        spnSatellitePosition.Value := spnSatellitePosition.Value + 1;
end;

procedure TfrmOne.btnPositionerGoEastClick(Sender: TObject);
var
    DVB-T2Data: TDVB-T2Data;
begin
    DVB-T2Data[0] := 0;
    DVB-T2Data[1] := $30;
    DVB-T2Data[2] := $68;
    DVB-T2Data[3] := $00;
    if UseFlexCop then
        FlexCopDVB-T2Command(CardHandle, DVB-T2Repeats, CDVB-T2Command, 4,
            DVB-T2Data)
    else
        GOTVIEW_USB_2_MASTERHD_3DVB-T2Command(CardHandle, DVB-T2Repeats,
            CDVB-T2Command, 4,
            DVB-T2Data);
end;

procedure TfrmOne.btnPositionerStepEastClick(Sender: TObject);
var
    DVB-T2Data: TDVB-T2Data;
begin
    DVB-T2Data[0] := 0;
    DVB-T2Data[1] := $30;
    DVB-T2Data[2] := $68;
    DVB-T2Data[3] := $FF;
    if UseFlexCop then
        FlexCopDVB-T2Command(CardHandle, DVB-T2Repeats, CDVB-T2Command, 4,
            DVB-T2Data)
    else
        GOTVIEW_USB_2_MASTERHD_3DVB-T2Command(CardHandle, DVB-T2Repeats,
            CDVB-T2Command, 4,
            DVB-T2Data);
end;

procedure TfrmOne.btnPositionerSetEastLimitClick(Sender: TObject);

```

```

var
  DVB-T2Data: TDVB-T2Data;
begin
  DVB-T2Data[0] := 0;
  DVB-T2Data[1] := $30;
  DVB-T2Data[2] := $66;
  if UseFlexCop then
    FlexCopDVB-T2Command(CardHandle, DVB-T2Repeats, CDVB-T2Command, 3,
      DVB-T2Data)
  else
    GOTVIEW_USB_2_MASTERHD_3DVB-T2Command(CardHandle, DVB-T2Repeats,
      CDVB-T2Command, 3,
      DVB-T2Data);
end;

procedure TfrmOne.btnPositionerGoWestClick(Sender: TObject);
var
  DVB-T2Data: TDVB-T2Data;
begin
  DVB-T2Data[0] := 0;
  DVB-T2Data[1] := $30;
  DVB-T2Data[2] := $69;
  DVB-T2Data[3] := $00;
  if UseFlexCop then
    FlexCopDVB-T2Command(CardHandle, DVB-T2Repeats, CDVB-T2Command, 4,
      DVB-T2Data)
  else
    GOTVIEW_USB_2_MASTERHD_3DVB-T2Command(CardHandle, DVB-T2Repeats,
      CDVB-T2Command, 4,
      DVB-T2Data);
end;

procedure TfrmOne.btnPositionerStepWestClick(Sender: TObject);
var
  DVB-T2Data: TDVB-T2Data;
begin
  DVB-T2Data[0] := 0;
  DVB-T2Data[1] := $30;
  DVB-T2Data[2] := $69;
  DVB-T2Data[3] := $FF;
  if UseFlexCop then
    FlexCopDVB-T2Command(CardHandle, DVB-T2Repeats, CDVB-T2Command, 4,
      DVB-T2Data)
  else
    GOTVIEW_USB_2_MASTERHD_3DVB-T2Command(CardHandle, DVB-T2Repeats,
      CDVB-T2Command, 4,
      DVB-T2Data);
end;

procedure TfrmOne.btnPositionerSetWestLimitClick(Sender: TObject);
var
  DVB-T2Data: TDVB-T2Data;
begin
  DVB-T2Data[0] := 0;
  DVB-T2Data[1] := $30;
  DVB-T2Data[2] := $67;
  if UseFlexCop then
    FlexCopDVB-T2Command(CardHandle, DVB-T2Repeats, CDVB-T2Command, 3,
      DVB-T2Data)
  else
    GOTVIEW_USB_2_MASTERHD_3DVB-T2Command(CardHandle, DVB-T2Repeats,
      CDVB-T2Command, 3,
      DVB-T2Data);

```

```

end;

procedure TfrmOne.btnPositionerDisableLimitsClick(Sender: TObject);
var
  DVB-T2Data: TDVB-T2Data;
begin
  DVB-T2Data[0] := 0;
  DVB-T2Data[1] := $30;
  DVB-T2Data[2] := $63;
  if UseFlexCop then
    FlexCopDVB-T2Command(CardHandle, DVB-T2Repeats, CDVB-T2Command, 3,
      DVB-T2Data)
  else
    GOTVIEW_USB_2_MASTERHD_3DVB-T2Command(CardHandle, DVB-T2Repeats,
      CDVB-T2Command, 3,
      DVB-T2Data);
end;

procedure TfrmOne.mnuScrambledClick(Sender: TObject);
begin
  mnuScrambled.Checked := not mnuScrambled.Checked;
  ReadFavourites;
  UpdateSettings(nil);
end;

procedure TfrmOne.mnuDataClick(Sender: TObject);
begin
  mnuData.Checked := not mnuData.Checked;
  ReadFavourites;
  UpdateSettings(nil);
end;

procedure TfrmOne.mnuRadioClick(Sender: TObject);
begin
  mnuRadio.Checked := not mnuRadio.Checked;
  ReadFavourites;
  UpdateSettings(nil);
end;

procedure TfrmOne.mnuTVClick(Sender: TObject);
begin
  mnuTV.Checked := not mnuTV.Checked;
  ReadFavourites;
  UpdateSettings(nil);
end;

procedure TfrmOne.mnuDisabledClick(Sender: TObject);
begin
  mnuDisabled.Checked := not mnuDisabled.Checked;
  ReadFavourites;
  UpdateSettings(nil);
end;

procedure TfrmOne.mnuSetupFilesClick(Sender: TObject);
begin
  if not Assigned(FSetupFiles) then
    begin
      FSetupFiles := TfrmSetupFiles.Create(Application);
      // встановлення налаштувань
      SetFilesSetup(FSetupFiles);
    end;
  FSetupFiles.ExitNotify := SetupFilesExit;
  FSetupFiles.Show;

```

```

end;

procedure TfrmOne.SetupFilesExit(Sender: TObject);
begin
  GetAndWriteFilesSetup(FSetupFiles, False);
  if FSetupFiles.SettingChanged then
  begin
    ReadLists;
    ReadFavourites;
  end;
  FSetupFiles := nil;
end;

procedure TfrmOne.mnuSetupDVB-T2Click(Sender: TObject);
begin
  if not Assigned(FSetupDVB-T2) then
  begin
    FSetupDVB-T2 := TfrmSetupDVB-T2.Create(Application);
    SetDVB-T2Setup(FSetupDVB-T2);
  end;
  FSetupDVB-T2.ExitNotify := SetupDVB-T2Exit;
  FSetupDVB-T2.Show;
end;

procedure TfrmOne.SetupDVB-T2Exit(Sender: TObject);
begin
  GetAndWriteDVB-T2Setup(FSetupDVB-T2, False);
  FSetupDVB-T2 := nil;
end;

procedure TfrmOne.mnuSetupMiscellaneousClick(Sender: TObject);
begin
  if not Assigned(FSetupMiscellaneous) then
  begin
    FSetupMiscellaneous := TfrmSetupMiscellaneous.Create(Application);
    // встановлення всіх налаштувань
    SetMiscellaneousSetup(FSetupMiscellaneous);
  end;
  FSetupMiscellaneous.ExitNotify := SetupMiscellaneousExit;
  FSetupMiscellaneous.Show;
end;

procedure TfrmOne.SetupMiscellaneousExit(Sender: TObject);
begin
  GetAndWriteMiscellaneousSetup(FSetupMiscellaneous, False);
  FSetupMiscellaneous := nil;
end;

procedure TfrmOne.mnuSetupRemoteClick(Sender: TObject);
begin
  if not Assigned(FSetupRemote) then
  begin
    FSetupRemote := TfrmSetupRemote.Create(Application);
    SetRemoteSetup(FSetupRemote);
  end;
  FSetupRemote.ExitNotify := SetupRemoteExit;
  FSetupRemote.Show;
end;

procedure TfrmOne.SetupRemoteExit(Sender: TObject);
begin
  GetAndWriteRemoteSetup(FSetupRemote, False);
  FSetupRemote := nil;
end;

```

```
end;

procedure TfrmOne.imgRecordListClick(Sender: TObject);
var
  ItemsAvailable: Boolean;
begin
  if not Assigned(FRecordings) then
  begin
    FRecordings := TfrmRecording.Create(Application);
    ItemsAvailable := SetRecordings(FRecordings);
  end
  else
    ItemsAvailable := True;
  if not ItemsAvailable then
  begin
    if Assigned(FRecordings) then
      FreeAndNil(FRecordings);
    Exit;
  end;
  FRecordings.ExitNotify := RecordingsExit;
  FRecordings.Show;
end;
end.
```

K6713-2023