

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф. О.А.Смірнов
« 11 » січня 2021 року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Коваленку Віталію Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмне забезпечення системи керування сервоприводами та виведення контенту Digital Signage

керівник роботи Петренюк Володимир Ілліч, канд. фіз.-мат. наук, доцент

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

затверджені наказом вищого навчального закладу № 189-02 від 28.12.2020 року

2. Строк подання студентом роботи до захисту 22.05.2021 р.

3. Мета та завдання кваліфікаційної бакалаврської роботи: Метою розробки є програмне забезпечення системи керування сервоприводами та виведення контенту Digital Signage

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

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

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

6. Дата видачі завдання « 11 » січня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів кваліфікаційної бакалаврської роботи	Примітка
1.	Аналіз існуючих систем	10.03.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2021 р.	
3.	Розробка моделі компонента	20.03.2021 р.	
4.	Розробка структур даних	25.03.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2021 р.	
6.	Програмування алгоритмів	10.04.2021 р.	
7.	Оформлення ПЗ	17.04.2021 р.	
8.	Попередній захист роботи	14.05.2021 р.	

Студент _____

(підпис)

_____ (прізвище та ініціали)

Керівник роботи _____

(підпис)

_____ (прізвище та ініціали)

АНОТАЦІЯ

Коваленко В.М. Програмне забезпечення системи керування сервоприводами та виведення контенту Digital Signage. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній кваліфікаційній бакалаврській розроблено програмне забезпечення, яке призначено для системи керування сервоприводами та виведення контенту Digital Signage.

Метою розробки є програмне забезпечення системи керування сервоприводами та виведення контенту Digital Signage.

Результат роботи – програмна реалізація системи керування сервоприводами та виведення контенту Digital Signage.

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

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

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10.

Програму розроблено в середовищі RAD Studio Delphi 10.4.

Ключові слова: комп'ютерна інженерія, Digital Signage

ABSTRACT

Kovalenko V.M. Servo control system and Digital Signage content output software. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this bachelor's qualification, software has been developed that is designed for the Servo drive control system and Digital Signage content output system.

The purpose of the development is the software of the servo drive control system and Digital Signage content output.

The result is a software implementation of the servo control system and Digital Signage content output.

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

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

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

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

Keywords: computer engineering, Digital Signage

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми кваліфікаційної бакалаврської роботи.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування	16
2.3 Розгорнута постановка завдання	22
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	24
3.1 Опис функціонування системи	24
3.2 Розробка структурної схеми.....	30
3.3 Розробка функціональної схеми	34
3.4 Розробка діаграми процесів	36
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ	38
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	38
4.2 Захист розробленого програмного забезпечення.....	57
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	59
6 ОСНОВНІ ВИСНОВКИ	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	63

КБР-123.21.0052.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Коваленко В.М.			Програмне забезпечення системи керування сервоприводами та виведення контенту Digital Signage	Лім.	Аркуш	Аркушіів
Перев.		Петренко В.І.				Б	1	69
Н.контр.		Гермак В.С.			ЦНТУ КІ-19-2СК			
Затв.		Смірнов О.А.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ВКЗ	–	відеоконференцзв'язок
ПЗ	–	програмне забезпечення
CMP	–	Codian Management Platform
DNS	–	Domain Name System
GMS	–	Global Management System
GRE	–	Generic Routing Encapsulation
ISDN	–	Integrated Services Digital Network
MCU	–	Multipoint Control Unit
MPLS	–	Multiprotocol Label Switching
MXM	–	Media eXchange Manager
SSL	–	Secure Socket Layer
TLS	–	Transport Layer Security
TMS	–	Tandberg Management Suite
VPN	–	Virtual Private Network
USENET		User Network

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ВСТУП

Актуальність теми. Ефективність використання систем Digital Signage у рекламній індустрії, торгівлі, банківській сфері й багатьох інших областях уже доведена.

З тих пор, як рекламні агентства й роздрібні мережі почали активно впроваджувати системи Digital Signage, була проведена безліч досліджень їх ефективності.

От що вдалося з'ясувати психологам і маркетологам:

- 60% покупок відбувається імпульсивно;
- завдяки Digital Signage імовірність покупки зростає на 20-120%;
- 77% аудиторії звертають увагу на динамічне зображення;
- при перегляді відео час очікування біжить на 35% швидше;
- 42% аудиторії воліють купувати в тих магазинах, де демонструється цифрове відео;
- 76% покупців прагнуть користуватися інтерактивними системами;
- 30% споживачів вважають, що цифрові меню мотивують до здійснення покупки.

Ефективність Digital Signage більше не зазнає сумніву, і при впровадженні мова йде про те, як прискорити повернення інвестицій і досягти максимальних показників ефективності. Цього можна добитися двома способами: запропонувати більш якісний, таргетований контент або підсилити вплив за рахунок технічних інновацій.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи керування сервоприводами та виведення контенту Digital Signage.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

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

– Огляд існуючих систем керування сервоприводами та виведення контенту Digital Signage.

– Дослідження системи керування сервоприводами та виведення контенту Digital Signage.

– Програмна реалізація системи керування сервоприводами та виведення контенту Digital Signage.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі керування сервоприводами та виведення контенту Digital Signage.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи керування сервоприводами та виведення контенту Digital Signage, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Оскільки цифрові засоби відображення інформації й реклами усе більше впливають на наше життя й відкривають можливості для бізнесу, можливості систем Digital Signage також розвиваються зі звичайного показу цифрових слайдів і відео, переміщаючи контент у більш просунуту площину, витягаючи, з'єднуючи й комбінуючи контент із різних джерел, взаємодіючи з ним, відображає його в режимі реального часу.

Усі ми знаємо Digital Signage як цифрову інсталяцію, яка відтворює й відображає відео або інший мультимедійний контент в інформаційних або рекламних цілях. Ми бачимо її всюди. Ми бачили рекламу на цифрових світлодіодних екранах у місті, меню в кафе й ресторанах швидкого харчування на РК-екранах, електронне табло з розкладом на вокзалах і аеропортах, купуючи квитки в кіно й усе це завдяки Digital Signage.

Існують нескінченні варіанти її використання різноманітні потреби, що забезпечують, бізнесу в повідомленні інформації до аудиторії. Фактично, очікується, що ринок цифрових вивісок виросте з 20,8 млрд. USD в 2020 році до 29,6 млрд. USD до 2024 року, що вказує на величезний вплив і потенційні можливості.

У цій роботі ми розглянемо визначення, область застосування й потенціал Digital Signage. Давайте спробуємо розібратися із цим. Вікіпедія пише, що Digital Signage – це підкатегорія електронних екранів, яка, у свою чергу, визначається як застосовувана в них технологія відображення, наприклад LCD, LED, проекція й «електронний папір». Це досить даремно. Екран – це набагато більше, ніж просте встаткування. Розуміння сучасного Digital Signage вимагає більш глибокого розгляду його ролі, функцій і технічних можливостей.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1.2 Область застосування

Digital Signage – є повною системою створення й відтворення контенту для локальної мережі й каналів закритого корпоративного віщання. Це гнучкий і простий в освоєнні інструмент для створення, планування й передачі якісного контенту прямо з настільного ПК.

Пакет продуктів Digital Signage має сучасні технологічні характеристики, які сприяють швидкому створенню, розгортанню й моніторингу цифрової інформації через мережі віщання на дисплеї з динамічним зображенням, які ідеальні для використання в різних областях, таких як комерція, роздрібна торгівля, промисловість, готельно-туристичний бізнес і ін.

Короткий перелік можливостей

Керування:

- Web-інтерфейс.
- Централізоване керування мережею плеєрів з будь-якої крапки усередині мережі.
- Канали зв'язку – інтернет, VPN-мережа, GRPS, WI-FI.
- Керування мережею – 1 людина.
- Поділ рівня доступу й ролей користувачів.
- Моніторинг системи віщання 24/7.
- On-line відновлення контенту.
- Захист контенту від несанкціонованого доступу.
- Локалізований інтерфейс.
- Технічна підтримка в Україні й у світі (по телефону й e-mail).

Відтворення:

- Надійне відтворення 24/7.
- Два незалежні канали віщання з одного плеєра.
- Музичний канал.
- Трансляція на вертикальних екранах.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		6

- Керування екранами.
- Відтворення всіх основних форматів медіаданих.

Створення контенту:

- Вбудований модуль створення динамічного контенту.
- Мінімальний трафік.
- Автоматична публікація матеріалів (текст і графіка) з web-ресурсів.
- Робота з більшим обсягом форматів медіаданих.

Гнучке формування зон на екрані:

- Екран може бути поділений на будь-яка кількість зон.
- Зони можуть розташовуватися одна поверх іншої.
- Для кожної зони індивідуальний плей-аркуш.

Рядок, що біжить:

- Текст може рухатися в кожному напрямку (горизонтально й вертикально).
- Інформація для рядка, що біжить, може надходити з інтернету із сайту компанії або інших ресурсів, забиратися з текстового файлу.

Екстрені повідомлення:

- Можливість моментальної заміни контенту.
- Автоматична робота із системою аварійного оповіщення.

Використання зовнішніх джерел даних

- Текстові дані.
- Бази даних (1С, SQL і т.д.).
- Відео й телевізійні трансляції.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи керування сервоприводами та виведення контенту Digital Signage, є актуальною задачею, яка потребує вирішення у даній кваліфікаційній бакалаврській роботі.

					КБР-123.21.0052.00.00.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 кваліфікаційної бакалаврської роботи

AG Neovo

Цей виробник представив тільки засоби відображення. Для виводу інформації в холах і коридорах ТЦ рекомендований інформаційний стенд DF-55B, а для по будови відеостіни – панелі PN-46D. Засобом відображення стенда DF-55У служить двосторонній дисплей, причому інформація на обох сторонах може бути різної – для кожного монітора є свій вхід HDMI. Подвійні рознімання стандарту HDMI спрощують інтеграцію із промисловим ПК або медіаплеєром, для установки якого є спеціальний відсік у підставі стенда.

Auvix

Компанія Auvix запропонувала відразу чотири варіанти рішення для стельових дисплеїв і три – для відеостіни. Представляємо рішення для стельових екранів.

1. Бюджетний варіант:

- бюджетні Lcd-панелі 46" (Samsung, LG, Philips, Panasonic, NEC);
- плеєри Iadea XMP-6200, установлювані на кожний екран (або один на два екрани, за підтримки функції прохідного цифрового сигналу в Lcd-панелі);
- безкоштовне ПЗ в комплекті із плеєрами Iadea XMP-6200 (статистика показів не збирається).

Таке рішення підходить для невеликих ТЦ зі скромним бюджетом, воно дозволяє вирішувати базові завдання, але має ряд обмежень. Наприклад, немає можливості синхронізувати відтворення контенту; не саме зручне ПЗ для роботи з контентом.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

(або плеєри) показує індивідуальний контент. Такий контент може створюватися централізовано, крім того, право створення контенту може бути надане магазину, де встановлений даний екран. Якщо магазин сам управляє своїм контентом, то він може міняти тільки ту частину плей-аркуша (або зони), яка йому відведена.

NEC

У якості окремо висячих РК-дисплеїв для побудови системи DS замовникові рекомендовані дисплеї серії V – наприклад, NEC V423, NEC V463 і NEC V552. Відзначимо ряд реалізованих у цих дисплеях функцій і технологій, що забезпечують високу надійність і, як наслідок, швидке повернення інвестицій. По-перше, це система триурівневого захисту від перегріву працездатність, що гарантує дисплеїв при високих температурах, що й запобігає передчасний вихід з ладу при досягненні критичних рівнів перегріву. По-друге, датчик зовнішньої освітленості, що адаптує яскравість дисплея до рівня освітленості, що в остаточному підсумку дозволяє заощадити на витратах на електроенергію. По-третє, мережне рознімання на дисплеї для віддаленого моніторингу й керування.

Найголовнішою перевагою зазначених дисплеїв є наявність опціонального слота для установки медіаплеєра NEC OPS. На медіаплеєрі є ряд програмних продуктів, завдяки яким системи DS можна відразу використовувати (запуск «out-of-box»). Крім того, завжди є можливість установити додаткові комерційні продукти сторонніх розроблювачів, просто скачавши їх з Google Play.

NEC Display Solutions тісно співробітничает з компанією Smartsign, яка пропонує програмні продукти, спеціально розроблені для рішень з медіаплеєрами NEC OPS. Доставка контенту від центрального сервера на віддалені дисплеї із установленим медіаплеєром здійснюється по локальній мережі або Wi-Fi (передбачена також можливість хмарного й локального розміщення). Для зв'язку із сервером Smartsign на медіаплеєр NEC OPS необхідно встановити додаток Smartsign Android Player (доступно безкоштовно в Google Play). Усе подальше керування системою зводиться до роботи з ПЗ через Web-інтерфейс (для хмарного рішення) або локально на сервері. ПЗ українізоване й, як затверджують

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

мережі Ethernet. Одна з особливостей рішення «Колан» – використання комутаторів мультимедіаконтента (Smart Matrix Multimedia, SMM). Вони ухвалюють і передають сигнали по звичайних каналах локальної мережі, але « усередині» реалізується функціональність матричних відеокмутаторів, здатних комутирувати відеосигнали й управляти їхнім відображенням. Зокрема, такий пристрій здатний формувати відеостіни.

За словами представників компанії «Колан», на відміну від звичайних медіаплеєрів, які здатні транслювати тільки заздалегідь підготовлені файли з відео або світлинами, пропоновані ними контент-плеєри вміють поєднувати на одному екрані в реальному часі й виводити в заданій послідовності й з необхідними ефектами різноманітні вихідні дані: відео, картинки, тексти, флеш-анімацію, Rss-стрічки, Pdf-файли, ТБ-канали й «живе» відео, Html-документи й багато чого іншого. Система керування контент-плеєрами (медіасервери TNT SX) дозволяє створювати безліч різних розкладів трансляції контенту, а також поєднувати контент-плеєри в групи для централізованого контролю й керування ними.

Фахівці компанії «Колан» виділяють наступні ключові особливості й переваги свого рішення: гнучкість системи комутації, більш низька вартість (у порівнянні з розв'язками інших виробників), а також можливість досить просто створити свої відеоролики. Остання забезпечується завдяки наявності бібліотеки з більшим числом шаблонів різної тематики й достатньо простого у використанні вбудованого редактора контенту.

КРОК

Для рішення завдання замовника найкраще підійдуть дисплеї Samsung DB48D (48") із вбудованим медіаплеєром другого покоління Samsung Smart Signage Platform (SSSP).

Але, поряд із вбудованими плеєрами SSSP, він включив у проект і окремі плеєри на основі міні-ПК AOpen. Вирішальними факторами вибору (вбудований або зовнішній плеєр) є передбачуваний медіаконтент і його складність. SSSP-

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		14

елементами системи DS в одному або декількох ТЦ (без обмежень) з одного центру керування. Для організації збору статистики показів комплекс програм DISE доповнюється спеціальним розширенням Monitoring Add-on, яке здатне збирати параметри, що настраюються, статистики й надавати її в зручній формі для звітності перед клієнтами даної системи.

На кожний комп'ютер у системі (рекомендовані пристрої АOpen) встановлюється ПЗ DISE Replay, після чого вони управляються з єдиного центру за допомогою DISE Server. Комплекс програм DISE Server передбачає можливість контролю доступу до елементів системи DS, таким чином, власникам торговельних крапок при необхідності може бути надана можливість самостійно управляти контентом на виділеному дисплеї.

Питання створення контенту на етапі запуску компанія «Моушн В'ю» за узгодженням із замовником готова побрати на себе, так само як і запропонувати послуги своєї дизайнерської групи для подальшої роботи з контентом. Однак, на думку фахівців компанії, для реалізації незалежного керування й організації ефективної роботи із системою DS замовникові було б вигідніше побрати в штат власного дизайнера. Для цих цілей у пропозиції «Моушн В'ю» виділені пункти по навчанню персоналу.

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.

					КБР-123.21.0052.00.00.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.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

Універсальний інсталятор для установки Online і Offline

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

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

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на кваліфікаційну бакалаврську роботу, реалізації підлягає програмне забезпечення, яке призначено для системи керування сервоприводами та виведення контенту Digital Signage.

В процесі розробки кваліфікаційної бакалаврської роботи необхідно виконати наступний обсяг роботи:

- а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;
- б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;
- в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

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.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

От деякі приклади найпоширеніших типів Digital Signage, заснованих на сценаріях використання:

- Загальнодоступний – навігація й інформація (новини, погода, розклад, дорожній трафік і т.д.)
- Корпоративний – інформування персоналу, обмін знаннями, бронювання переговорних і конференц-залів, координація нарад і зустрічей, навчання
- Продажі – демонстрація й реклама продуктів і послуг
- Туризм – туристичні продукти, пропозиція готелів, інформування гостей про послуги й дозвіллі
- Роздрібна торгівля – системи самообслуговування, «живі» торговельні полки, демонстрація продукції, «розумне» дзеркало
- Ресторани – цифрове меню, кіоски самообслуговування, розвага й інформування відвідувачів
- Маркетинг – акції, створення атмосфери, гейміфікація
- Музеї – мультимедійні стенди, інтерактивна експозиція, каси, цифрові путівники й електронні таблички
- Утвір – цифровий розклад, інформування й навчання
- Виставки – інтерактивні каталоги продуктів і послуг, реєстрація відвідувачів

Що таке встаткування Digital Signage?

Digital Signage більше не є просто відеоекраном, на якому відображається контент по певному плей-листу. Ми живемо в епоху, коли можна по своєму бажанню вибирати з безлічі варіантів устаткування, комбінуючи його для досягнення своїх цілей. Деякі передові розв'язки дозволяють, навіть не володіючи спеціальними технічними знаннями, впроваджувати сучасні технології з мінімальними зусиллями й досвідом.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

От короткий огляд деяких з найбільш популярних технологій.

Сенсорні мультитач-екрани

Сенсорні екрани дозволяють за допомогою дотику до поверхні екрана, одним або декільком пальцями або іншими предметами, взаємодіяти з відображуваним на екрані контентом. Доторкатися до екрана можуть декілька людей, що приводить до безлічі незалежних взаємодій, що відбуваються паралельно.

Дві найбільш популярні технології:

Ємнісна – сенсорна технологія, яка виявляє у своєму полі що-небудь струмопровідне. Вбудовані в скло датчики, визначають місце розташування потоку струму, який потім визначається як подія торкання. Найбільш популярна форма відома як проекційна ємнісна, вона застосовується у всіх мобільних телефонах і планшетах. Ємнісні дисплеї вважаються найточнішою сенсорною технологією й, таким чином, золотим стандартом, коли екран захищений від впливу погодних умов.

Інфрачервона технологія випромінює сітку невидимого інфрачервоного світла над поверхнею екрана. Коли об'єкт стикається з екраном, інфрачервоний промінь перерветься, що приведе до визначення місця торкання. На відміну від проекційно-ємнісних дисплеїв, на поверхню яких необхідно нанести сенсорний провідний матеріал, сенсорні дисплеї з інфрачервоною технологією можуть працювати з кожною екранною поверхнею.

Beacon маяки

Маяки – це електронні пристрої з низьким енергоспоживанням, що передають унікальний ідентифікатор або Url-адресу. Привласнюючи маяк окремим елементам або відправляючи повідомлення, можна створити інформаційний контекстний міст між дисплеями й фізичним продуктом або мобільним пристроєм. Приклад використання маяків містить у собі безконтактний маркетинг і автоматизований путівник.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

RFID, NFC і інші зчитувачі (наприклад, QR або AR код, чип, пін-код і т. д.)

Усі ці пристрої фіксують унікальні ідентифікатори, які можна використовувати для відтворення на екрані відповідної інформації. Наприклад, піднесіть до зчитувачу вашу членську картку, і призначена саме для вас інформація відобразиться на екрані.

Підключені об'єкти

Підключені об'єкти – це підключені до Інтернету пристрою й доступні через деякий API, що дозволяє легко передавати інформацію в/из їх за допомогою інших цифрових сервісів. У результаті Інтернет речей (IoT) стає усе більш розповсюдженим. Представте, що холодильник пише вам, що у вас скінчилося молоко. Ця технологія може підключити рішення Digital Signage до датчиків, лампочок, камер, телевізорів, виконавчих пристроїв і т.д. Наприклад коли людина входить у приміщення, можуть автоматично зашторитися вікна, приглушити світло, увімкнутися відеопроєктор і запуснитися відеоролик у відповідності з перевагами глядача, орієнтуючись на його підлогу, вік і інші дані, які він повідомив про себе раніше. А по закінченню перегляду, системою може бути запропоновано пройти короткий тест, щоб одержати зворотний зв'язок, повернувши приміщення у вихідний стан чекаючи наступного відвідувача.

Що таке програмне забезпечення Digital Signage?

Якщо ви коли-небудь гуглили «програмне забезпечення для Digital Signage» і звернули увагу на те, що всі приблизно говорять про одне й те саме. По правді говорячи, непросто дати всеосяжне визначення програмному забезпеченню Digital Signage, оскільки багато варіантів програмного забезпечення пропонують різні можливості й, отже, підходять для різних цілей.

Однак можна спробувати піти на компроміс, описавши, як повинне виглядати ідеальний програмний рішення для Digital Signage. Воно повинен дозволити своїм користувачам легко управляти контентом і підключеними пристроями Digital Signage, максимально автоматизувати процес відтворення

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

контенту, з можливістю інтеграції зовнішніх даних, устаткування й сторонніх сервісів, реалізації інтерактивності, а також збору аналітичних даних.

Воно повинен не залежати від використовуваного мультимедійного формату, розміру й типу екрана, дозволяти створювати й використовувати будь-який дизайн, графічний макет або сценарій відтворення, бачити результат відразу, до публікації на екранах. Це розширює можливості дистанційної роботи й керування.

Строго говорячи, програмне забезпечення Digital Signage складається із чотирьох основних компонентів:

- Створення й редагування контенту.
- Поширення, планування й керування відтворенням контенту.
- Керування пристроєм відтворення.
- Моніторинг.

Яке майбутнє в Digital Signage?

Сьогодні інновації Digital Signage в основному відбуваються у світі програмного забезпечення. От три області, за якими потрібно стежити:

Інтерактивність без доторкань

Майбутнє покоління Digital Signage містить у собі інтерактивність. На додаток до вищезгаданих технологій пристроїв програмне забезпечення для Digital Signage дозволить нам працювати практично з будь-яким інтерактивним підходом, який ви можете собі представити, включаючи:

- Доповнена реальність.
- Розпізнавання мови.
- Датчики руху.
- Матеріальні об'єкти.
- Взаємодії по демографічних ознаках.
- Дистанційні дії.
- Розпізнавання жестів.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Це всебічна взаємодія людини з машиною будь-якого роду, доступне для тих, хто прагне впливати на аудиторію через унікальний контент, що підходить для конкретного середовища й цільової аудиторії.

Підключення до світу

Більшість представлених на ринку програмних рішень для Digital Signage являють собою систему керування контентом (CMS), пропонуючи спеціальний користувацький інтерфейс (UI), що дозволяє користувачам завантажувати контент і управляти їм, який потім публікується на медіаплеєрі.

CMS нікуди не подінуться, але майбутні для Digital Signage за продуктами, які можуть витягати контент із будь-якої кількості сторонніх джерел. Незалежно від того, чи є контент локальним або хмарним, перевага цього підходу полягає в тому, що він розширює обсяг усілякого контенту й типу інформації, яка може бути відображена на екранах.

Деякі приклади зовнішніх джерел, які можуть служити постачальниками й споживачами даних в Digital Signage:

- Автономні джерела: Microsoft Excel, Бази даних.
- Хмарні джерела: Google Sheets, Dropbox, Google.
- Автономна CMS: Airtable, Wordpress, Directus.
- Соціальні мережі: Facebook, Instagram, Twitter, Rss-канали, зображення Flickr і т.д.
- Сторонні сервіси, додатки й пристрою: IFTTT і Zapier.

Будь-яке джерело даних, бізнес-логіка або пристрій, доступне через API.

Можливості – це дані

Як згадувалося раніше, рішення для Digital Signage уже здатні витягати інформацію з різних джерел для підвищення рівня взаємодії із цільовою аудиторією. Аудиторія виражає свої переваги й інтерес за допомогою персоналізованих взаємодій, забезпечуваних такими технологіями пристроїв, як веб-камери, що визначають стать й вік, зчитувачі RFID / NFC або QR-коди.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		29

Тепер представте, які можливості відкриються, коли Digital Signage зможе забезпечувати всі можливі взаємодії, скомбінованих з окремих елементів, миттєво адаптувати контент і вишиковувати сценарії під конкретного користувача, враховуючи напрямок погляду, психо-фізичний стан, настрій, очікування, соціальний стан, фінансові можливості, поведінка в суспільстві й багато чого іншого – у комбінації з конкретними обставинами, місцем розташування й навколишнім середовищем.

Digital Signage більше не буде просто засобом відображення контенту. Технологія дозволить управляти аудиторією, стане інформаційним центром, який збирає інформацію про аудиторію і її поведінці, миттєво адаптуючи контент залежно від конкретних умов.

Результатом впровадження Digital Signage є не тільки привабливий, різноманітний, цікавий і корисний контент, але й вікно можливостей для вашого бізнесу, що забезпечує приріст нової аудиторії і її лояльність.

3.2 Розробка структурної схеми

Персоналізований добір контенту. Системи розпізнавання образів, що вбудовуються в дисплеї, уже зараз уможливають облік соціологічних (підлога, вік) і поведінкових (напрямок переміщення, рух очей) параметрів при видачі контенту. Далі ці системи будуть тільки удосконалюватися.

Підвищення інтерактивності. Одержуючи від користувача зворотний зв'язок, система може підбирати цільовий контент із урахуванням конкретних запитів.

Удосконалювання систем керування контентом. Приклади корисних функцій: зонування віщання й реклами, підлаштування контенту під зміну параметрів зовнішнього середовища, обробка даних, що надходять від систем окулографії (або eye-tracking, відстеження рухів око).

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Нові види медіаносіїв. Удосконалювання характеристик і здешевлення екранів, розробка нових видів: прозорих світлодіодних дисплеїв, екранів складної форми, рухливих екранів.

Механізовані екрани

Відразу обмовимося, що такого терміна, як механізований екран (або механізована система екранів), поки не існує, оскільки на українському ринку аудіовізуальних систем такого продукту ще немає.

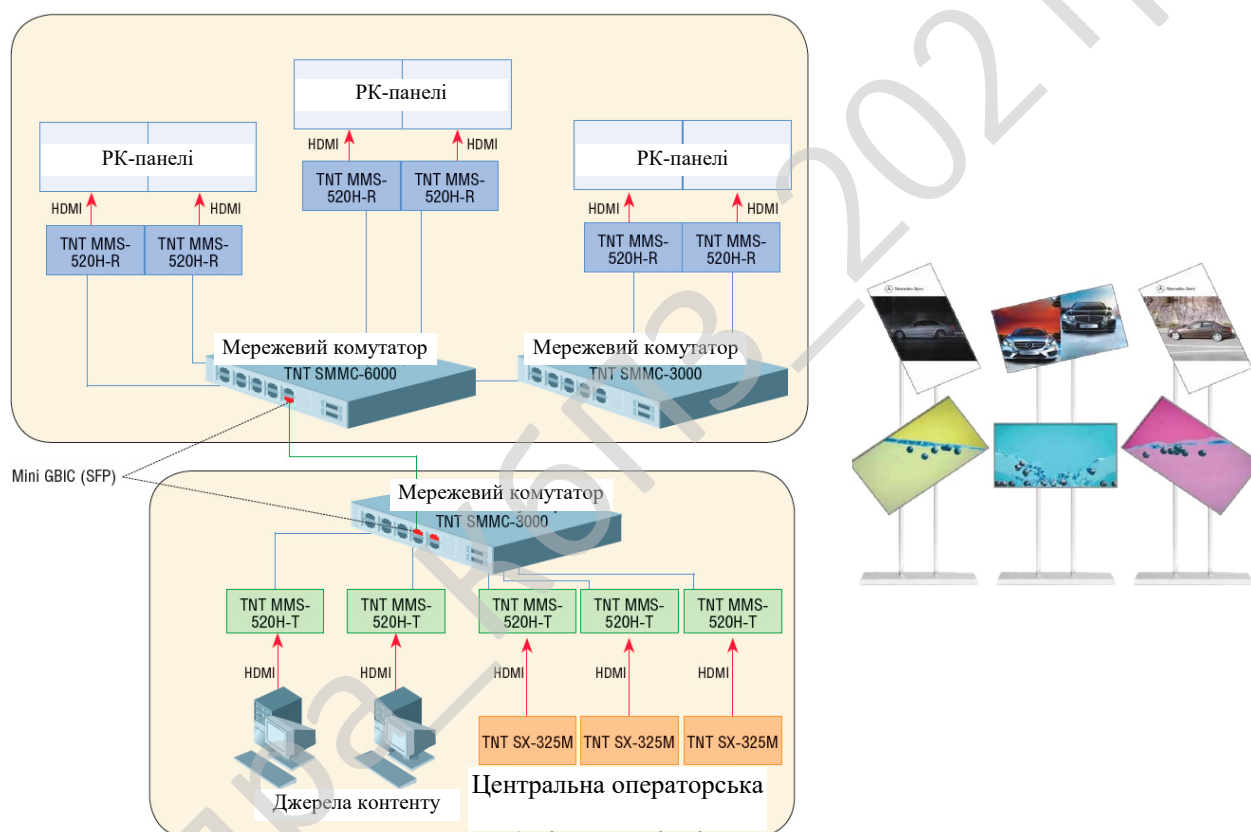


Рисунок 3.1 – Структурна схема системи

Що це таке – механізована система екранів? Мова йде про одне або декількох жорстко з'єднаних між собою екранах (наприклад, світлодіодних), які приводяться в рух сервоприводом. Відео обробляється таким чином, щоб разом з рухом екранів створювався потрібний візуальний ефект. Переміщенням дисплеїв і видачею контенту управляє спеціальне програмне забезпечення.

сервоприводом і обертається незалежно від інших «поверхів» щодо загальної осі – центральної колони. За допомогою спеціального ПЗ можна управляти швидкістю й положенням будь-якої призми, синхронізуючи їх з відеороликом.

У підставу й кришку можуть бути вбудовані стаціонарні LED-екрани, на які буде виводитися інформація, пов'язана зі змістом відео.

Обертові вежі здатні працювати в чотирьох режимах: «шпалери», повноекранний режим, групування й вільний рух.

Шпалери. На всіх трьох гранях демонструється те саме зображення.

Повноекранний режим. На кожній грані може бути своє, окреме зображення, але призми рухаються як одне ціле.

Групування. Дві сусідні призми легко поєднуються в групу, що дозволяє демонструвати два різні ролики.

Вільний рух. Усі призми обертаються незалежно одна від іншої, і при необхідності на різних гранях демонструються різні ролики. У цьому режимі можливе створення візуальних ефектів.

«Танець» таких екранів – заворожливе видовище.

Обертові екрани. З погляду механіки й обробки відео ця схема найпростіша. LED-дисплей монтується на нерухливу підставку або стіну за допомогою кронштейна, що забезпечує його обертання на 360° у вертикальній площині. Екран повертається щодо центральної крапки (вона може й не збігатися із крапкою перетинання його головних діагоналей) згідно із заданим алгоритмом: з постійної або мінливою кутовою швидкістю, із зупинками в певних положеннях. Відео й обертання екрана синхронізуються (кілька екранів можна об'єднати в систему), за рахунок чого досягаються цікаві візуальні ефекти.

Ковзний екран. Ковзний екран являє собою великий LED-дисплей у боксі, поверх якого переміщається одна або декілька світлодіодних панелей менших розмірів. Зображення на панелях «накочується» на більшу відеосцену.

Ковзна панель приводиться в рух серводвигуном із планетарною й черв'ячної/ рейковою передачею, керування яким здійснюється із ПЛК.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		33

Координатами панелі й часом її переміщення можна управляти, синхронізуючи ці параметри з відео на стаціонарному екрані.

Ковзні екрани працюють у трьох режимах.

Шпалери. Поверх статичної картинки на стаціонарному екрані переміщається зображення, як форма рухливої панелі правило логотип.

Слайд-синхронізація. Створюється ефект синхронізованого руху на стаціонарному екрані й панелі.

Рекламний щит. Залежно від відеоконтенту основного екрана міняється видача рекламного контенту.

3.3 Розробка функціональної схеми

Програмне забезпечення для Digital Signage останнього покоління, що забезпечує:

- Можливість об'єднання відеомоніторів по всій країні в єдину мережу із централізованим керуванням віщанням;
- Моніторинг стану пристроїв у реальному часі;
- Віддалене відновлення контенту через інтернет;
- Автоматичний підрахунок статистики показів (з можливістю доступу рекламодавців до неї в онлайн-режимі);
- Можливість додавання рядка, що біжить, і віджетів (час, погода, курси валют, пробки й т.п.);
- Можливість використання каналів стільниковому зв'язку або Wi-Fi для відновлення контенту.

Сервіс передбачає використання широкого спектра недорогого встаткування на ОС Android і Windows, доступного в будь-яких магазинах електроніки, це дозволяє вишиковувати мережі Digital Signage будь-якого масштабу в стислий термін і з мінімальними фінансовими витратами.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		34



Рисунок 3.2 – Функціональна схема системи

Підтримуються відеомонітори будь-якого формату, будь те звичайний планшет, телевизор, відео-стіна, зовнішній LED-екран, відеостенд або плазмабокс.

Підключення пристроїв до мережі інтернет можливо будь-якими можливими способами: Ethernet, Wi-Fi, 3G, 4G, EDGE.

Сервіс простий і зручний на будь-якому етапі використання, починаючи від підключення, закінчуючи вивантаженням звітів зі статистикою. Для роботи не потрібно кваліфікованого персоналу:

- Контент-менеджер завантажує відеоролики на хмарний диск.
- Реєструє екрани в системі.
- Становить плейлисти.
- Запускає трансляцію.
- Відслідковує статистику показів і стан компонентів системи.
- Реєструє додаткових користувачів з потрібними правами (наприклад, тільки на перегляд статистики).

Відмінні риси:

- Централізоване керування медіаконтентом.
- Зручне планування ефіру як окремого екрана, так і всієї мережі.

- Потоки зовнішні по відношенню до системи сутності.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Потоки даних гібридні між елементами трьох попередніх типів.

Відповідно до документації основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Для схематичного представлення системи що розробляється необхідно спочатку представити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи в цілому у подальшому. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі. Розроблена діаграма взаємодії процесів системи в подальшому уточнюється шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

Таким чином у результаті після розгляду, вищеописаної системи, схеми структурної, функціональної, діаграми взаємодії процесів перейдемо до опису та розгляду блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо алгоритм роботи основної програми. Його блок-схема зображена на рисунку 4.1, з рисунку видно, що після запуску програми відбуваються наступні дії:

- Виведення вікна керування сервоприводами та виведення контенту.
- Складання списку апаратних пристроїв у мережі типу Digital Signage.
- Побудова топології мережі та знаходження пристроїв.
- Сканування пристроїв на статус активний.
- Виведення на екран схеми даних.
- Аналіз отриманих статистичних даних.
- Виведення результатів.
- Запит коду пристрою.
- Підпрограма формування коду пристрою.
- Визначення типу відповіді.
- Виведення на екран результатів.
- Запит відключення пристрою.
- Підпрограма відключення пристрою.
- Виведення на екран звіту.
- Запит кінцевий – завершення роботи після циклу перевірок ПЗ (Вихід).

Робота підпрограми зображено на рисунку 4.2, з рисунку видно, що після виклику відбуваються наступні дії:

- Запит перевірки, знайдено аномальне поведження пристроїв?
- Виділення пристроїв з аномальним трафіком.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

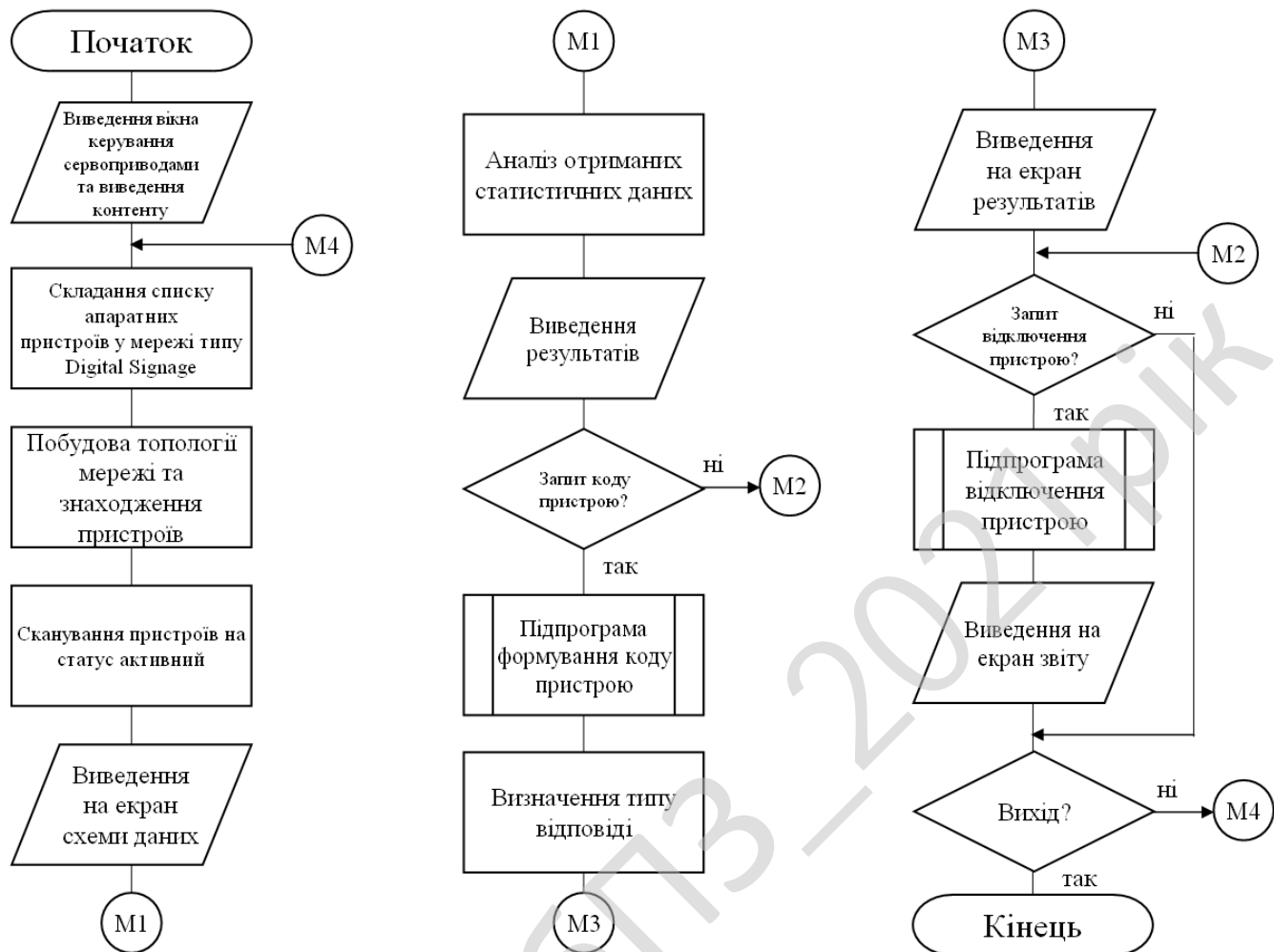


Рисунок 4.1 – Блок-схема основної програми

- Переадресація аномального трафіку, відключення пристрою.
- Багатоступінчастий аналіз аномального трафіку.
- Відділення джерел працюючих у штатному режимі.
- Створення фільтрів для аномального трафіку пристроїв.
- Ввімкнення фільтрації трафіку.
- Виведення інформації про перенаправлення та фільтрацію.

Розглянемо частину коду який призначено для захисту розробленого програмного забезпечення за допомогою алгоритму DES, більш детально теоретичну основу розглянуто у розділі 4.3.

Вим.	Арк.	№ докум.	Підпис	Дата

КБР-123.21.0052.00.00.ПЗ

Арк.

39

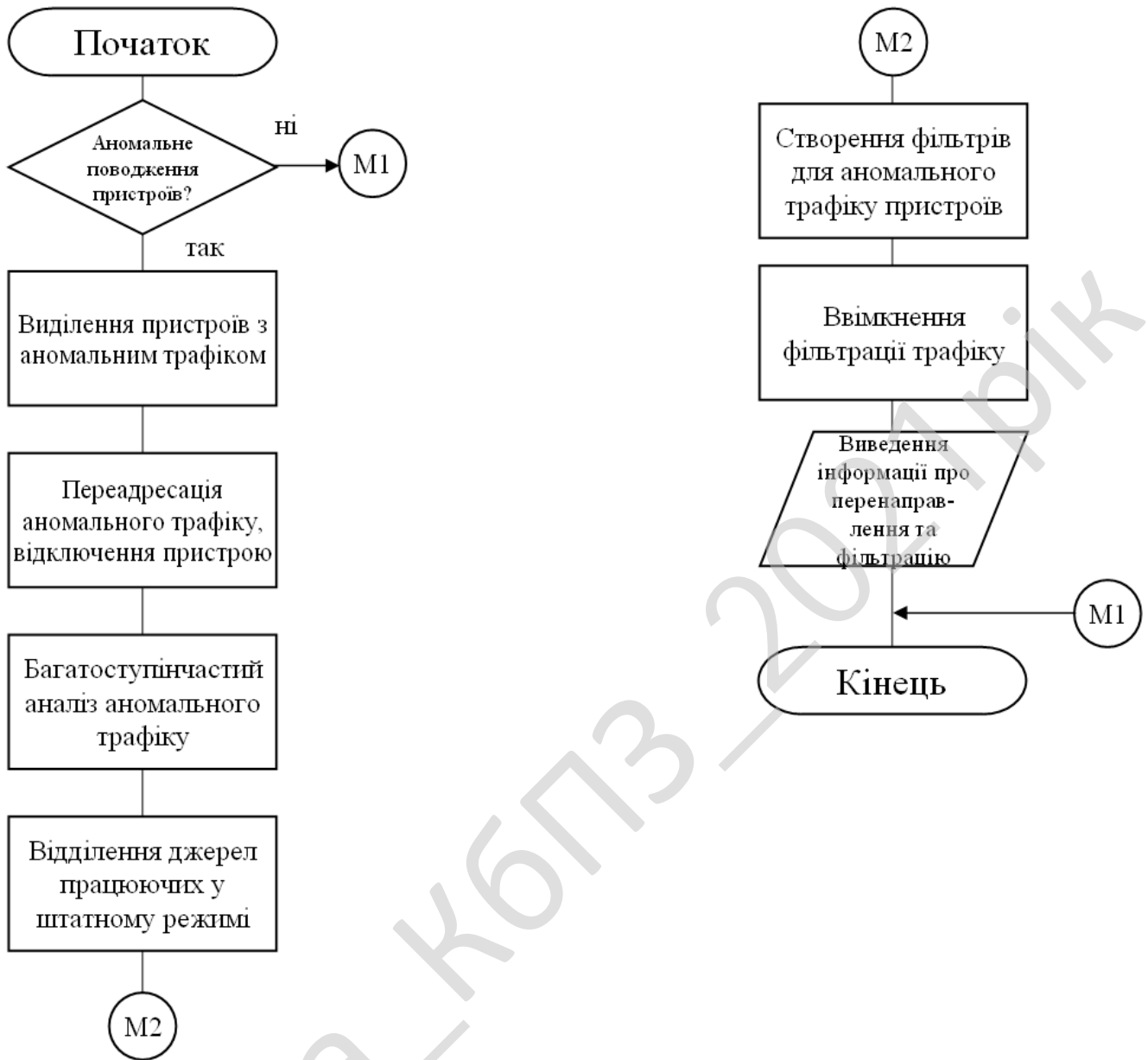


Рисунок 4.2 – Блок-схема роботи підпрограми

Вихідний код наступний:

```

// Модуль алгоритму DES
unit Unit1;

// інтерфейс на частина
interface

// підключення
uses
  
```

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

КБР-123.21.0052.00.00.ПЗ

Арк.

40

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ImgList, ActnMan, ActnColorMaps, ToolWin, ActnCtrls,
ActnMenus, XPStyleActnCtrls, ActnList, ExtCtrls;

// оголошення типів даних та ін..

type

TForm1 = class(TForm)

ActionManager1: TActionManager;

ActionMainMenuBar1: TActionMainMenuBar;

XPColorMap1: TXPColorMap;

Edit1: TEdit;

Action1: TAction;

Action2: TAction;

Action3: TAction;

Action4: TAction;

OpenDialog1: TOpenDialog;

SaveDialog1: TSaveDialog;

OpenDialog2: TOpenDialog;

SaveDialog2: TSaveDialog;

ImageList1: TImageList;

Image1: TImage;

Button1: TButton;

Button2: TButton;

Mem1: TMemo;

procedure Action1Execute(Sender: TObject);

procedure Action2Execute(Sender: TObject);

procedure Action3Execute(Sender: TObject);

procedure Action4Execute(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

// об'ява змінних

var

Form1: TForm1;

fp1,fp2:file;

ip1: array[1..64] of integer=(58,50,42,34,26,18,10,2,
60,52,44,36,28,20,12,4,
62,54,46,38,30,22,14,6,
64,56,48,40,32,24,16,8,

Вим.	Арк.	№ докум.	Підпис	Дата

КБР-123.21.0052.00.00.ПЗ

Арк.

41

```

57,49,41,33,25,17,9,1,
59,51,43,35,27,19,11,3,
61,53,45,37,29,21,13,5,
63,55,47,39,31,23,15,7);
ip2: array[1..64] of integer=(40,8,48,16,56,24,64,32,
39,7,47,15,55,23,63,31,
38,6,46,14,54,22,62,30,
37,5,45,13,53,21,61,29,
36,4,44,12,52,20,60,28,
35,3,43,11,51,19,59,27,
34,2,42,10,50,18,58,26,
33,1,41,9,49,17,57,25);

i,j,k,flag:integer;
t: array[1..64] of byte;
l: array[1..32] of byte;
r: array[1..32] of byte;
tmp: array[1..64] of byte;
pass: array[1..32] of byte;
temp: byte;

// реалізаційна частина
implementation

uses Unit2;
// підключення файлу ресурсів
{$R *.dfm}

procedure TForm1.Action1Execute(Sender: TObject);
begin
Form1.Close;
end;

procedure TForm1.Action2Execute(Sender: TObject);
begin
Form2.ShowModal;
end;

// Реалізація при натисненні через діалогову форму
procedure TForm1.Action3Execute(Sender: TObject);
begin
if Length(Edit1.Text)>1 then
begin

```

Вим.	Арк.	№ докум.	Підпис	Дата

КБР-123.21.0052.00.00.ПЗ

Арк.

42

```

    i:=1;
    flag:=0;
// основний цикл функції
    while i<33 do
        begin
            for j:=1 to Length(Edit1.Text) do
                begin
                    for k:=0 to 7 do
                        begin
// ЗДВИГ
                            pass[i]:=((byte(Edit1.Text[j]) shl k) and $80) shr 7;
                            inc(i);
                            if i>32 then
                                break;
                            end;
                            if i>32 then
                                break;
                            end;
                        end;
                    end;
                end;
            end;
        OpenDialog1.FileName:='';
        OpenDialog1.InitialDir:=GetCurrentDir;
        if OpenDialog1.Execute then
            begin
                AssignFile(fp1,OpenDialog1.FileName);
                Reset(fp1, SizeOf(byte));
                SaveDialog1.FileName:=OpenDialog1.FileName+'.des';
                SaveDialog1.InitialDir:=GetCurrentDir;
                if SaveDialog1.Execute then
                    begin
// підключення файлу
                        AssignFile(fp2,SaveDialog1.FileName);
                        Rewrite(fp2, SizeOf(byte));
                        while not eof(fp1) do
                            begin
                                for i:=1 to 8 do
                                    begin
                                        if not eof(fp1) then
                                            BlockRead(fp1, temp, SizeOf(byte))
                                        else
                                            begin
                                                temp:=0;
                                                inc(flag);
                                            end;
                                    end;
                                end;
                            end;
                    end;
                end;
            end;
        end;
    end;

```

Вим.	Арк.	№ докум.	Підпис	Дата

КБР-123.21.0052.00.00.ПЗ

Арк.

43

```

        end;
        for k:=0 to 7 do
            begin
                tmp[(i-1)*8+k+1]:=((temp shl k) and $80) shr 7;
            end;
        end;
end;
// цикл з 1 до 64 елементу
for i:=1 to 64 do
    t[i]:=tmp[ip1[i]];
for i:=1 to 32 do
    begin
        l[i]:=t[i];
        r[i]:=t[i+32];
    end;
// цикл з 1 до 16 елементу
for i:=1 to 16 do
    begin
        for j:=1 to 32 do
            begin
                tmp[j]:=l[j];
                l[j]:=r[j];
            end;
        for j:=1 to 32 do
            begin
// застосування xor
                r[j]:=l[j] xor pass[j];
                r[j]:=r[j] xor tmp[j];
            end;
        end;
        for i:=1 to 32 do
            begin
                t[i]:=l[i];
                t[i+32]:=r[i];
            end;
        for i:=1 to 8 do
            begin
                for k:=0 to 7 do
                    begin
                        temp:=temp shl 1;
                        temp:=temp+(t[ip2[(i-1)*8+k+1]] and $1);
                    end;
                BlockWrite(fp2, temp, SizeOf(byte));

```

Вим.	Арк.	№ докум.	Підпис	Дата

КБР-123.21.0052.00.00.ПЗ

Арк.

44

```

        end;
        end;
        BlockWrite(fp2, flag, SizeOf(byte));
        CloseFile(fp2);
    end;
    CloseFile(fp1);
end;
end
// в іншому випадку виводимо помилку - Пароль занадто короткий
else
    begin
        MessageDlg('Пароль занадто короткий!', mtError, [mbOk], 0);
        Edit1.Clear;
        ActiveControl:=Edit1;
    end;
end;

procedure TForm1.Action4Execute(Sender: TObject);
begin
    if Length(Edit1.Text)>1 then
        begin
            i:=1;
            flag:=0;
            // основний цикл
            while i<33 do
                begin
                    for j:=1 to Length(Edit1.Text) do
                        begin
                            for k:=0 to 7 do
                                begin
                                    pass[i]:=((byte(Edit1.Text[j]) shl k) and $80) shr 7;
                                    inc(i);
                                    if i>32 then
                                        break;
                                end;
                            if i>32 then
                                break;
                            end;
                        end;
                    end;
                end;
            OpenDialog2.FileName:='';
            // діалогова форма
            OpenDialog2.InitialDir:=GetCurrentDir;

```

Вим.	Арк.	№ докум.	Підпис	Дата

КБР-123.21.0052.00.00.ПЗ

Арк.

45

```

if OpenFileDialog2.Execute then
begin
AssignFile(fp1,OpenDialog2.FileName);
Reset(fp1, SizeOf(byte));
SaveDialog2.FileName:=OpenDialog2.FileName;
SaveDialog2.InitialDir:=GetCurrentDir;
if SaveDialog2.Execute then
begin
// підключення файлу діалога
AssignFile(fp2,SaveDialog2.FileName);
Rewrite(fp2, SizeOf(byte));
while not eof(fp1) do
begin
for i:=1 to 8 do
begin
if not eof(fp1) then
BlockRead(fp1, temp, SizeOf(byte))
else
begin
flag:=temp;
end;
for k:=0 to 7 do
begin
tmp[(i-1)*8+k+1]:=((temp shl k) and $80) shr 7;
end;
end;
// цикл з 1 до 64 елементу
for i:=1 to 64 do
t[ip2[i]]:=tmp[i];
for i:=1 to 32 do
begin
l[i]:=t[i];
r[i]:=t[i+32];
end;
for i:=1 to 16 do
begin
for j:=1 to 32 do
begin
tmp[j]:=r[j];
r[j]:=l[j];
end;
for j:=1 to 32 do

```

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

```

        begin
            l[j]:=r[j] xor pass[j];
            l[j]:=l[j] xor tmp[j];
        end;
    end;
end;
// цикл з 1 до 32 елементу
for i:=1 to 32 do
    begin
        t[i]:=l[i];
        t[i+32]:=r[i];
    end;
for i:=1 to 64 do
    tmp[ip1[i]]:=t[i];
for i:=1 to 8 do
    begin
        for k:=0 to 7 do
            begin
                temp:=temp shl 1;
                temp:=temp+(tmp[(i-1)*8+k+1] and $1);
            end;
            BlockWrite(fp2, temp, SizeOf(byte));
        end;
    end;
    Seek(fp2, FilePos(fp2)-flag-8);
    Truncate(fp2);
    CloseFile(fp2);
end;
// закриття файлу
CloseFile(fp1);
end;
end
else
    begin
        MessageDlg('Пароль занадто короткий!', mtError, [mbOk], 0);
        Edit1.Clear;
        ActiveControl:=Edit1;
    end;
end;
end.

```

						КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			47

Для збереження файлів було використано MongoDB – документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць.

MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Код MongoDB написаний на мові C++ і поширюється в рамках ліцензії AGPLv3.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій.

У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу.

Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин.

При розробці автори виходили з необхідності спеціалізації баз даних, завдяки чому їм вдалося відійти від принципу «один розмір під усе». За рахунок мінімізації семантики для роботи з транзакціями з'являється можливість вирішення цілого ряду проблем, пов'язаних з нестачею продуктивності, причому горизонтальне масштабування стає простішим. Використовувана модель документів зберігання даних (JSON/BSON) простіше кодується, простіше управляється (у тому числі за рахунок застосування так званого безсхемного

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

від очікуваного показника. У 1990 Г. Марковіцу присуджена Нобелівська премія за розробку теорії і практики оптимізації портфеля фондових активів.

Етапи ризик-менеджменту

У ризик-менеджменті прийнято виділяти декілька ключових етапів:

– на першому етапі відбувається виявлення ризику з супутньою оцінкою вірогідності його реалізації і масштабу наслідків;

– на другому етапі здійснюється розробка ризик-стратегії з метою зниження вірогідності реалізації ризику і мінімізації можливих негативних наслідків;

– на третьому етапі вибираються методи і інструменти управління виявленим ризиком;

– на четвертому етапі проводиться безпосереднє управління ризиком;

– на завершальному етапі оцінюються досягнуті результати і коректується ризик-стратегія.

За ключовий етап ризик-менеджменту вважається етап вибору методів і інструментів управління ризиком.

Методи і інструментарій ризик-менеджменту

Базовими методами ризик-менеджменту є відмова від ризиків, зниження, передача і ухвалення.

Ризик-інструментарій значно ширший. Він включає політичні, організаційні, правові, економічні, соціальні інструменти, причому ризик-менеджмент як система допускає можливість одночасного застосування декількох методів і інструментів ризик-управління.

Найбільш часто вживаним інструментом ризик-менеджменту є страхування. Страхування припускає передачу відповідальності за відшкодування передбачуваного збитку сторонній організації (страхової компанії).

Прикладами інших інструментів можуть бути відмова від надмірно ризикової діяльності (метод відмови), профілактика або диверсифікація (метод

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		52

зниження), аутсорсинг витратних ризикових функцій (метод передачі), формування резервів або запасів (метод ухвалення).

Розглянемо протокол SNMP (Simple Network Management Protocol, простий протокол керування мережею) – це протокол керування мережами зв'язку на основі архітектури TCP/IP.

На основі концепції TMN в 1980–1990 р. різними органами стандартизації був вироблений ряд протоколів керування мережами передачі даних з різним спектром реалізації функцій TMN. До одного з типів таких протоколів керування належить Simple Network Management Protocol (SNMP).

SNMP – це технологія, покликана забезпечити керування й контроль за пристроями й програмами в мережі зв'язку шляхом обміну керуючою інформацією між агентами, що розташовуються на мережних пристроях, і менеджерами, розташованими на станціях керування. SNMP визначає мережу як сукупність мережних керуючих станцій й елементів мережі (головні машини, шлюзи й маршрутизатори, термінальні сервери), які спільно забезпечують адміністративні зв'язки між мережними керуючими станціями й мережними агентами. SNMP різних версій присвячений цілий ряд рекомендацій IETF (RFC).

Зазвичай при використанні SNMP присутні керовані та керуючі системи. До складу керованої системи входить компонент, який називається агентом, який відправляє звіти керуючій системі. По суті SNMP агенти передають управлінську інформацію на керуючі системи як змінні (такі як «вільна пам'ять», «ім'я системи», «кількість працюючих процесів» тощо).

Керуюча система може отримати достовірну інформацію через операції протоколу GET, GETNEXT і GETBULK. Агент може самостійно без запиту надсилати дані, використовуючи операцію протоколу TRAP або INFORM. Управляючі системи можуть також відправляти конфігураційні оновлення або контролюючі запити, використовуючи операцію SET для безпосереднього управління системою. Операції конфігурування та управління використовуються

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

тільки тоді, коли потрібні зміни у мережній інфраструктурі. Операції моніторингу зазвичай виконуються на регулярній основі.

Змінні, доступні через SNMP, організовані в ієрархії. Ці ієрархії та інші метадані (такі як тип і опис змінної) описуються Базами Керуючої Інформації (Management Information Bases (MIBs)).

SNMP не визначає, яку інформацію (які змінні) керована система повинна надавати. Навпаки, SNMP використовує розширювану модель, в якій доступна інформація визначається Базами Керуючої Інформації (MIB – Management Information Base).

Бази Керуючої Інформації описують структуру керуючої інформації пристроїв. Вони використовують ієрархічний адресний простір імен, що містить унікальний ідентифікатор об'єкта (object identifier (OID)).

Грубо кажучи, кожен унікальний ідентифікатор об'єкта ідентифікує змінну, яка може бути прочитана чи встановлена через SNMP. MIB'и використовують нотацію, визначену в ASN.1.

Ієрархія MIB може бути зображена як дерево з безіменним коренем, рівні якого приписані різними організаціями. На найвищому рівні MIB OID'и належать різним організаціям, що займаються стандартизацією, в той час як на нижчих рівнях OID'и виділяються асоційованими організаціями. Ця модель забезпечує управління на всіх шарах мережної моделі OSI, адже MIB'и можуть бути визначені для будь-яких типів даних і операцій.

Керований об'єкт – це одна з будь-якого числа характеристик, специфічних для керованого пристрою. Керований об'єкт включає в себе один або більше екземплярів об'єкта (що ідентифікуються за OID), які насправді є змінними.

Існує два типи керованих об'єктів: Скалярні об'єкти визначають єдиний екземпляр об'єкта; Табличні об'єкти визначають множинні, пов'язані екземпляри об'єктів які групуються в таблицях MIB.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Прикладом керованого об'єкта може бути atInput, який є скалярним об'єктом що містить єдиний екземпляр об'єкта, ціле число, яке показує загальну кількість вхідних пакетів AppleTalk на мережний інтерфейс маршрутизатора.

Ідентифікатор об'єкта (OID) унікально ідентифікує керований об'єкт в ієрархії MIB.

В телекомунікаціях і комп'ютерних мережах, ASN.1 є стандартною гнучкою нотацією для опису структур даних, що служать для кодування, передачі і декодування даних. ASN.1 являє собою набір правил для опису структури об'єктів, незалежних від специфічних для обладнання методик кодування, і формальну нотацію, яка дозволяє уникнути неоднозначностей.

ASN.1 це єдиний ISO та ITU-T стандарт, спочатку визначений у 1984 році як частина стандарту CCITT X.409: 1984. Пізніше, в 1988 році, завдяки його широкому застосуванню, він був виділений в окремий стандарт X.208. Значно переглянута версія 1995 року описана в X.680.

Адаптована підмножина ASN.1 Структура Керуючої Інформації (SMI) описана в протоколі SNMP для визначення наборів пов'язаних MIB об'єктів, також званих MIB модулями.

SNMP працює на прикладному рівні TCP/IP (сьомий рівень моделі OSI). Агент SNMP отримує запити по UDP-порту 161. Менеджер може посилати запити з будь-якого доступного порту джерела на порт агента. Відповідь агента буде відправлений назад на порт джерела на менеджері. Менеджер отримує повідомлення (Traps і InformRequests) по порту 162. Агент може генерувати повідомлення з будь-якого доступного порту. При використанні TLS або DTLS запити виходять по порту 10161, а пастки відправляються на порт 10162.

У SNMPv1 зазначено п'ять основних протокольних одиниць обміну (protocol data units – PDU). Ще дві PDU, GetBulkRequest і InformRequest, були введені в SNMPv2 і перенесені в SNMPv3.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Нижче перераховані сім протокольних одиниць обміну SNMP:

1. **GetRequest.** Запит від менеджера до об'єкту для отримання значення змінної або списку змінних. Необхідні змінні вказуються в полі `variable bindings` (розділ поля `values` при цьому не використовується). Отримання значень зазначеної змінної повинно бути виконано агентом як Атомарна операція. Менеджеру буде повернений **Response** (відповідь) з поточними значеннями.

2. **SetRequest.** Запит від менеджера до об'єкту для зміни змінної або списку змінних. Зв'язані змінні вказуються в тілі запиту. Зміни всіх зазначених змінних повинні бути виконані агентом як атомарна операція. Менеджеру буде повернений **Response** з (поточними) новими значеннями змінних.

3. **GetNextRequest.** Запит від менеджера до об'єкту для виявлення доступних змінних і їх значень. Менеджеру буде повернений **Response** зі зв'язаними змінними для змінної, яка є наступною в базі МІВ в лексикографічному порядку. Обхід всієї бази МІВ агента може бути проведений ітераційним використанням **GetNextRequest**, починаючи з **OID 0**. Рядки таблиці можуть бути прочитані, якщо вказати в запиті **OID**-и колонок в пов'язаних змінних.

4. **GetBulkRequest.** Покращена версія **GetNextRequest**. Запит від менеджера до об'єкту для численних ітерацій **GetNextRequest**. Менеджеру буде повернений **Response** з декількома пов'язаними змінними, обійденими починаючи зі пов'язаної змінної (змінних) в запиті. Специфічні для **PDU** поля `non-repeaters` і `max-repetitions` використовуються для контролю за поведінкою відповіді. **GetBulkRequest** був введений в **SNMPv2**.

5. **Response.** Повертає зв'язані змінні і значення від агента менеджеру для **GetRequest**, **SetRequest**, **GetNextRequest**, **GetBulkRequest** і **InformRequest**. Повідомлення про помилки забезпечуються полями статусу помилки і індексу помилки.

Ця одиниця використовується як відповідь і на **Get-**, і на **Set -**запити, в **SNMPv1** називається **GetResponse**.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

6. Trap. Асинхронне повідомлення від агента – менеджера. Включає в себе поточне значення sysUpTime, OID, що визначає тип trap (пастки), і необов'язкові зв'язані змінні. Адресація одержувача для пасток визначається за допомогою змінних trap-конфігурації в базі MIB. Формат trap-повідомлення був змінений в SNMPv2 і PDU перейменували в SNMPv2-Trap.

7. InformRequest. Асинхронне повідомлення від менеджера менеджера або від агента менеджера. Повідомлення від менеджера менеджера були можливі вже в SNMPv1 (за допомогою Trap), але SNMP зазвичай працює на протоколі UDP, в якому доставка повідомлень не гарантована, і не повідомляється про втрачені пакетах. InformRequest виправляє це зворотним відправленням підтвердження про отримання. Одержувач відповідає Response-му, що повторює всю інформацію з InformRequest. Цей PDU був введений в SNMPv2.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму DES.

DES є класичною мережею Фейштеля із двома гілками. Дані шифруються 64-бітними блоками, використовуючи 56-бітний ключ. Алгоритм перетворить за кілька раундів 64-бітний вхід в 64-бітний вихід. Довжина ключа дорівнює 56 бітам.

Процес шифрування складається із чотирьох етапів. На першому з них виконується початкова перестановка (IP) 64-бітного вихідного тексту (забілювання), під час якої біти переставляються у відповідності зі стандартною таблицею.

Наступний етап складається з 16 раундів однієї й тої ж функції, що використовує операції зрушення й підстановки. На третьому етапі ліва й права половини виходу останньої (16-й) ітерації міняються місцями.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Нарешті, на четвертому етапі виконується перестановка IP^{-1} результату, отриманого на третьому етапі. Перестановка IP^{-1} інверсна початковій перестановці.



Рисунок 4.3 – Загальна схема DES

Праворуч на рисунку показаний спосіб, яким використовується 56-бітний ключ. Спочатку ключ подається на вхід функції перестановки.

Потім для кожного з 16 раундів підключ K_i є комбінацією лівого циклічного зрушення й перестановки. Функція перестановки та сама для кожного раунду, але підключи K_i для кожного раунду виходять різні внаслідок повторюваного зрушення біт ключа.

середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

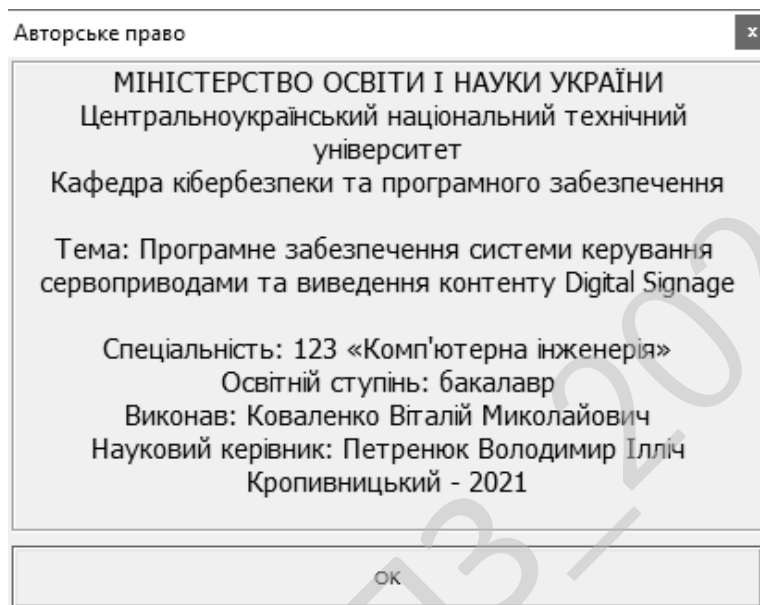


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання кваліфікаційної бакалаврської роботи, призначено для системи керування сервоприводами та виведення контенту Digital Signage.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем керування сервоприводами та виведення контенту Digital Signage.

– Досліджена система керування сервоприводами та виведення контенту Digital Signage.

– На основі отриманих результатів досліджень створена програмна реалізація системи керування сервоприводами та виведення контенту Digital Signage.

Розроблені під час виконання кваліфікаційної бакалаврської роботи алгоритми дозволяють успішно вирішувати завдання керування сервоприводами та виведення контенту Digital Signage.

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

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

Програма реалізована на мові високого рівня RAD Studio Delphi 10.4. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи керування сервоприводами та виведення контенту Digital Signage. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід,

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

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.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм DES.

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.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

17. Сафронов И.К. Практикум по информатике.– СПб.: БХВ-петербург, 2006.– 432 с.
18. Соловьева А.Н. Программное обеспечение в видеосюжетах + CD.– М.: Издательство Форум, 2007.– 432 с.
19. Videoconferencing Cookbook (version 4.1). March, 2005
<http://www.vide.net/cookbook/cookbook.en/>.
20. Айфичер Э. Цифровая обработка сигналов. Практический подход/ Э. Айфичер – М.: «Вильямс», 2004. – 992 с.
21. Амиантов И.Н. Избранные вопросы статистической теории связи/ Амиантов И.Н. – М.: Советское радио. – 1971. – 416 с.
22. Артеменко М.Е., Касымов Р.Р. Прогнозирование временных рядов для задач управления с использованием аппарата нейронных сетей и вейвлет анализа / М.Е. Артеменко, Р.Р. Касымов // Холодильна техніка і технологія.– 2010. – №3. – С. 74-76.
23. Артеменко М.Е, Касымов Р.Р. Использование вейвлетов для повышения качества прогноза телекоммуникационного трафика/ М.Е. Артеменко, Р.Р. Касымов // Зв'язок. – 2010. – №3.- С. 57-60.
24. Баранник В.В. Обоснование базовой технологии компрессии изображений с заданным качеством визуализации / В.В. Баранник, И.Е. Рогоза, А.А. Красноручкий // Інформаційно-керуючі системи на залізничному транспорті. – 2013. – Вип. 1. – С. 17-23.
25. Баранник В.В. Теоретичні основи та методи стиску зображень в телекомунікаційних системах на підставі біноміально-поліадичного представлення: Автореф. дис. д.т.н.: 05.12.02 / В.В. Баранник //.– Харків, Українська державна академія залізничного транспорту 2006. – 36 с.
26. Бараннік В.В. Метод стиснення НН-квдратури вейвлет-перетвореного зображення на основі поліадичного кодування/ В.В. Бараннік, А.В. Ширяєв, П.Н. Гуржий // Наукоємні технології, 2011. № 1-2 (9-10) – С. 69-72.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

О.А. Дробот, Д.В. Симоненко // Збірник наукових праць Харківського університету Повітряних Сил. Вип. 1(13), 2007. – С. 66-69.

47. Лемешко А.В. Тензорная модель многопутевой маршрутизации с гарантиями качества обслуживания одновременно по множеству разнородных показателей [Электронный ресурс] / А.В. Лемешко, О.Ю. Евсева // Проблемы телекоммуникаций. – 2012. – № 4 (9). – С. 16 – 31. – Режим доступа: http://pt.journal.kh.ua/2012/4/1/124_lemeshko_tensor.pdf.

48. Лемешко А.В. Модель отказоустойчивой маршрутизации многоадресных и широковещательных потоков в MPLS-сети / Лемешко А.В., Арус К.М. // Системи обробки інформації. – 2013. – №9 (116). – С. 160 – 163.

49. Малла С. Вейвлеты в обработке сигналов: пер. с англ./ С. Малла – М.: Мир, 2005. – 671 с.

50. Марчук Г.И. Методы вычислительной математики / Г.И. Марчук – М.: Наука, 1989. – 608 с.

51. Муранов О.С. Експериментальні дослідження механізмів прогнозування пульсацій пакетного трафіку / Муранов О.С. // Защита информации : Сб. научн. трудов Национального авиационного университета. – К. : НАУ, 2008. – Специальный выпуск. – С.137-142.

52. Муранов О.С. Підвищення якості технології адаптивного управління трафіком пакетних телекомунікаційних мереж: автореф. дис. на здобуття наук. ступеня кандидата техн. наук: спец. 05.13.06 – інформаційні технології / О.С. Муранов. – Київ: Національний авіаційний університет, 2010. – 29 с.

53. Новиков Ю.В. Локальные сети: архитектура, алгоритмы, проектирование / Ю.В. Новиков, С.В. Кондратенко – М.: ЭКОМ, 2000. – 312 с.

54. Остерлох Х. Маршрутизация в IP-сетях. Принципы, протоколы, настройка / Х. Остерлох – С.Пб.: ВHV-С.Пб., 2002. – 512 с.

55. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 2-е изд./ В.Г. Олифер, Н.А. Олифер – СПб: Питер, 2005. – 864 с.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

56. Пантелеев А.В. Методы оптимизации в примерах и задачах / А.В. Пантелеев, Т.А. Летова. – М.: Высшая школа, 2002. – 544 с.
57. Приоров А.Л. Обработка и передача мультимедийной информации: учебное пособие / А.Л. Приоров, В.В. Хрящев – Ярослав. гос. ун-т им. П.Г. Демидова. – Ярославль: ЯрГУ, 2010. – 188 с.
58. Прокис Дж. Цифровая связь. Пер. с англ./Под ред. Д. Д. Кловского. М.: Радио и связь, 2000. – 800 с.
59. Руккас К.М. Поточковая модель сети MPLS. / К.М. Руккас, К.А. Овчинников // Системи обробки інформації. Випуск 3(101) Том 2. – Х.: ХУПС. – 2012. – С. 211-214
60. Рябий М.О. Технологія підвищення рівня стиснення цифрових зображень на основі сплайн-моделі: автореф. дис. на здобуття наук. ступеня кандидата техн. наук: спец. 05.13.06 – інформаційні технології / М.О. Рябий // Київ, Національний авіаційний університет, 2013. – 20 с.
61. Семенов С.Г. Модели и методы управления ресурсами в информационно-телекоммуникационных системах / С.Г. Семенов, А.А. Смирнов, Е.В. Мелешко – Х.: Вид-во Віровець А.П. «Апостроф», – 2012. – 212 с.
62. Сергиенко А.Б. Цифровая обработка сигналов / А.Б. Сергиенко – СПб.: Питер, 2002. – 608 с.
63. Свами М.Н. Графы, сети и алгоритмы: пер. с англ. / М.Н. Свами, К. Тхуласираман; под ред. В.А. Горбатова. – М.: Мир, 1984. – 454 с.
64. Семенов А.Б. Структурированные кабельные системы / А.Б. Семенов, С.К. Стрижаков, И.Р. Сунчелей – 3-е изд. – М.: “Компьютер-Пресс”, 2001. – 608с.
65. Семенов С.Г. Анализ методов прогнозирования в телекоммуникационных сетях автоматизированных систем управления / С.Г. Семенов // Збірник наукових праць «Системи управління, навігації та зв'язку», – К.:ЦНДІ навігації і управління, – 2008.-Вип. 2(6). – С.134-137.

					КБР-123.21.0052.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					КБР-123.21.0052.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Коваленко В.М.				Програмне забезпечення системи керування сервоприводами та введення контенту Digital Signage	Літ.	Аркуш	Аркушів
Перевірів	Петренко В.І.					Б	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КІ-19-2СК			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи керування сервоприводами та виведення контенту Digital Signage.

2 Підстава для розробки

Підставою для розробки служить завдання на кваліфікаційну бакалаврську роботу, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 189-02 від 28.12.2020 року).

3 Мета та призначення розробки

Метою кваліфікаційної бакалаврської роботи є розробка програмного забезпечення системи керування сервоприводами та виведення контенту Digital Signage.

4 Джерела розробки

Джерелом цієї кваліфікаційної бакалаврської роботи 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;

					КБР-123.21.0052.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

– системи керування сервоприводами та виведення контенту Digital Signage;

– 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.

					КБР-123.21.0052.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10 із сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows XP/Vista/7/8/10.

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище RAD Studio Delphi 10.4.

					КБР-123.21.0052.00.00.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 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 69 аркушів.

					КБР-123.21.0052.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8 Етапи розробки

8.1 Збір і обробка інформації по темі кваліфікаційної бакалаврської роботи. Постановка задачі на виконання кваліфікаційної бакалаврської роботи (складання ТЗ).

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень кваліфікаційної бакалаврської роботи.

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

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання кваліфікаційної бакалаврської роботи на попередній захист 22.05.2021 р.

11.2 Подання кваліфікаційної бакалаврської роботи на захист 1.06.2021 р.

					КБР-123.21.0052.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник кваліфікаційної бакалаврської роботи

_____ Петренюк В.І.

*Програмне забезпечення системи керування сервоприводами та виведення
контенту Digital Signage*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 16

Літера: РП

Кропивницький – 2021 року

Digital_Signage.dpr - головний файл проекту

```
program Digital_Signage;
//підключення бібліотек
uses
  Forms,
  main in 'main.pas' {MainForm},
  option in 'option.pas' {OptionForm},
  prog in 'prog.pas' {ProgForm},
  about in 'about.pas' {AboutForm};

{$R *.res}

begin
  Application.Initialize;
  Application.Title := 'Обробка відеоінформації з Web-камери';
  Application.CreateForm(TMainForm, MainForm); // створення головного вікна
  програми
  Application.CreateForm(TOptionForm, OptionForm); // створення вікна «Параметри
  відео»
  Application.CreateForm(TProgForm, ProgForm); // створення вікна «Параметри
  програми»
  Application.CreateForm(TAboutForm, AboutForm); // створення вікна довідки
  Application.Run;
end.
```

about.pas - довідка

```
unit frmAbout;
interface
uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls, Buttons,
ExtCtrls;

type
  TAboutBox = class(TForm)
    Panell: TPanel;
    ProgramIcon: TImage;
    ProductName: TLabel;
    Version: TLabel;
    Copyright: TLabel;
    Comments: TLabel;
    Memol: TMemo;
    OKButton: TButton;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var AboutBox: TAboutBox;

implementation
{$R *.dfm}
procedure TAboutForm.FormCreate(Sender: TObject);
begin
  Memol.Clear;
  Memol.Lines.Add('БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА');
  Memol.Lines.Add('');
  Memol.Lines.Add('на тему:');
  Memol.Lines.Add('');
  Memol.Lines.Add('Програмне забезпечення системи керування сервоприводами та
виведення контенту Digital Signage');
  Memol.Lines.Add('');
  Memol.Lines.Add('');
  Memol.Lines.Add('Керівник: Петренко В.І. ');
  Memol.Lines.Add('');
  Memol.Lines.Add('Розробив: студент Коваленко Віталій Миколайович');
  Memol.Lines.Add('                гр. КІ-19-2СК');
  Memol.Lines.Add('');
  Memol.Lines.Add('');
  Memol.Lines.Add('м. Кропивницький 2021');
  Memol.Lines.Add('');
  Memol.Lines.Add('');
end;
procedure TAboutForm.Button1Click(Sender: TObject);
begin
  AboutForm.Close;
end;
end.
```

main.pas - основна програма

```

unit main;

interface

uses
  Windows, Messages, SysUtils, jpeg ,Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls, DSPack, DSUtil, DirectShow9, ComCtrls, ExtCtrls, IniFiles,
  XPMan,
  ToolWin, ActnMan, ActnCtrls, ActnMenus, XPStyleActnCtrls, ActnList,
  ImgList, OleCtrls, Buttons, StrUtils,
  ScktComp, Sockets, AppEvnts;

type
  TMainForm = class(TForm)
    CaptureGraph: TFilterGraph;
    VideoSourceFilter: TFilter;
    StatusBar: TStatusBar;
    MenuBar: TActionMainMenuBar;
    ActionManager: TActionManager;
    VideoSend: TAction;
    Exit: TAction;
    OnTop: TAction;
    Help: TAction;
    About: TAction;
    OptionProg: TAction;
    SampleGrabber: TSampleGrabber;
    ImageList: TImageList;
    Timer: TTimer;
    Size: TAction;
    ApplicationEvents: TApplicationEvents;
    Image: TImage;
    bt2: TImage;
    bt1: TImage;
    VideoWindow2: TVideoWindow;
    procedure FormDestroy(Sender: TObject);
    function IniGetStringValue(
      TheIniFile: string;
      IniSection: string;
      StringName: string;
      DefaultString: string): string;
    function IniSetStringValue(
      TheIniFile: string;
      IniSection: string;
      StringName: string;
      StringValue: string): Boolean;
    procedure OnTopExecute(Sender: TObject);
    procedure VideoSendExecute(Sender: TObject);
    procedure OptionVideoExecute(Sender: TObject);
    procedure ExitExecute(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure OptionProgExecute(Sender: TObject);
    procedure AboutExecute(Sender: TObject);

    procedure SendBitmap(hTargetWnd : HWND; Btm: TBitmap);
    procedure SendText (hTargetWnd : HWND; Text: string);
    procedure TimerTimer(Sender: TObject);
    procedure CreateProgressDialog;
    procedure Wait(tim: integer);
    function SetVideoParams(CB_B2: ICaptureGraphBuilder2; Category: TGUID;
      fSource: IBaseFilter): HRESULT;
    procedure SizeExecute(Sender: TObject);
    procedure HelpExecute(Sender: TObject);
    procedure ApplicationEventsHint(Sender: TObject);
  end;

```

```

private
  { Private declarations }
  procedure SendCopyData(hTargetWnd: HWND; ACopyDataStruct:TCopyDataStruct);
  procedure WMCopyData(var Msg: TWMCopyData); message WM_COPYDATA;
public
  { Public declarations }
  Format, Mode, Bit, Kadr, arch : integer;
  gladco: string;
  Kadr2: REAL;
end;

var
  MainForm: TMainForm;
  CapEnum: TSysDevEnum;
  VideoMediaTypes: TEnumMediaType;
  Writer: IFileSinkFilter;
  PinList: TPinList;
  CompFilter : TFilterList;
  CapFilters : TSysDevEnum;
  SendBt: TBitmap;
  first: boolean =true ;

implementation

uses option, prog, about;

{$R *.dfm}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////// Функції й процедури для передачі даних по з'єднанню ////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// передача бітмапа
procedure TMainForm.SendBitmap(hTargetWnd : HWND; Btm: TBitmap);
var
  ms: TMemoryStream;
  JpegIm: TJpegImage;
  MyCopyDataStruct: TCopyDataStruct;
begin
  ms := TMemoryStream.Create;
  JpegIm := TJpegImage.Create;
  try
    JpegIm.Assign (Btm);
    JpegIm.CompressionQuality := arch;
    JpegIm.Compress;
    JpegIm.SaveToStream (ms);
    with MyCopyDataStruct do
      begin
        dwData := 1;
        cbData := ms.Size;
        lpData := ms.Memory;
      end;
    SendCopyData (hTargetWnd, MyCopyDataStruct);
  finally
    ms.Free;
    JpegIm.Free;
  end;
end;

// передача тексту
procedure TMainForm.SendText( hTargetWnd : HWND; Text: string);
var
  MyCopyDataStruct: TCopyDataStruct;
begin
  with MyCopyDataStruct do
    begin

```

```

        dwData := 0;
        cbData := StrLen(PChar(Text)) + 1;
        lpData := PChar(Text)
    end;
    SendCopyData(hTargetWnd, MyCopyDataStruct);
end;

// для передачі даних
procedure TMainForm.SendCopyData(hTargetWnd: HWND;
ACopyDataStruct:TCopyDataStruct);
begin
    if hTargetWnd <> 0 then
        SendMessage(hTargetWnd, WM_COPYDATA, Longint(Handle),
Longint(@ACopyDataStruct))
// else ShowMessage('Заданий Handle не знайдений');
end;

// для прийому бітмапа
procedure TMainForm.WMCopyData(var Msg: TWMCopyData);
var
    sText: array[0..99] of Char;
    ms: TMemoryStream;
    JpegIm: TJpegImage;
begin
    case Msg.CopyDataStruct.dwData of
        0: { текст }
            begin
                StrLCopy(sText, Msg.CopyDataStruct.lpData, Msg.CopyDataStruct.cbData);
            end;
        1: { картинка }
            begin
                ms := TMemoryStream.Create;
                JpegIm := TJpegImage.Create;
                try
                    with Msg.CopyDataStruct^ do
                        ms.Write(lpdata^, cbdata);
                        ms.Position := 0;
                        JpegIm.LoadFromStream(ms);
                        try
                            begin
                                Height:=JpegIm.Height+69;
                                Width:=JpegIm.Width;
                                Image.Picture.Graphic.Assign(JpegIm);
                            end;
                        except
                            end;
                        finally
                            ms.Free;
                            JpegIm.Free;
                        end;
                    end;
                end;
            end;
    end;
end;

////////////////////////////////////
//////////////////////////////////// Функції й процедури для роботи із графікою
////////////////////////////////////
////////////////////////////////////
function TMainForm.SetVideoParams(CB_B2: ICaptureGraphBuilder2; Category: TGUID;
fSource: IBaseFilter): HRESULT;
var
    StreamConf: IAMStreamConfig;
    PAMT: PAMMediaType;
begin
    Result:= E_FAIL;
    StreamConf:= nil;
    PAMT:= nil;
    try
        Result:= CB_B2.FindInterface(@Category, @MEDIATYPE_Video, fSource,

```

```

IID_IAMStreamConfig, StreamConf);
if Assigned(StreamConf) then
begin
  StreamConf.GetFormat(PAMT);
  if Assigned(PAMT) then
  begin
    if PAMT.cbFormat= sizeof(TVideoInfoHeader) then
    begin
      PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biWidth:= 640;
      PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biHeight:= 480;
      PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biBitCount:= 24;
      PVIDEOINFOHEADER(PAMT^.pbFormat)^.AvgTimePerFrame:= 10000000 div Kadr;
      //fps
      with PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader do
        PAMT^.lSampleSize := ((biWidth + 3) and (not (3))) * biHeight *
biBitCount
          shr 3;

PVIDEOINFOHEADER(PAMT^.pbFormat)^.bmiHeader.biSizeImage:=PAMT^.lSampleSize;
      end;
      Result:= StreamConf.SetFormat(PAMT^)
      end;
      end;
      result:= S_OK;
    except
    on E: Exception do
    MessageBox(0, PChar(E.Message), '', MB_OK or MB_ICONERROR);
    end;
    StreamConf:= nil;
    if Assigned(PAMT) then
    DeleteMediaType(PAMT);
end;
end;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///////////////// Функції й процедури для роботи з файлами й директоріями //////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// одержати значення з файлу
function TMainForm.IniGetStringValue(
  TheIniFile: string;
  IniSection: string;
  StringName: string;
  DefaultString: string): string;
var
  TheIni: TIniFile;
begin
  TheIni := TIniFile.Create(TheIniFile);
  try
    Result :=
    TheIni.ReadString(
    IniSection,
    StringName,
    DefaultString);
    if Result = '' then
    Result := DefaultString;
  finally
    TheIni.Free;
  end;
end;
end;

// записати значення у файл
function TMainForm.IniSetStringValue(
  TheIniFile: string;
  IniSection: string;
  StringName: string;
  StringValue: string): Boolean;
var

```

```
TheIni: TIniFile;
begin
  TheIni := TIniFile.Create(TheIniFile);
  try
    try
      TheIni.WriteString(
        IniSection,
        StringName,
        StringValue);
      Result := True;
    except
      Result := False;
    end;
  finally
    TheIni.Free;
  end;
end;

////////////////////////////////////
////////////////////////////////////  Основний текст програми  //////////////////////////////////////
////////////////////////////////////

procedure TMainForm.Wait(tim: integer);
var
  whevent:THandle;
begin
  whevent:=createevent(nil,true,false,'et');
  while(true)do
  begin
    WaitForSingleobject(whevent,tim);
    Application.processmessages;
  end;
end;

procedure TMainForm.FormDestroy(Sender: TObject);
begin
  SendBt.Destroy;
  CaptureGraph.ClearGraph;
  CapEnum.Free;
  VideoMediaTypes.Free;
end;

procedure TMainForm.OnTopExecute(Sender: TObject);
begin
  // відображення програми поверх всіх вікон
  if OnTop.Checked then MainForm.FormStyle:=fsStayOnTop
  else MainForm.FormStyle:=fsNormal;
end;

procedure TMainForm.VideoSendExecute(Sender: TObject);
begin
  VideoWindow2.FilterGraph:=CaptureGraph;
  CapEnum:= TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
  CaptureGraph.ClearGraph;
  CaptureGraph.Active := false;
  VideoSourceFilter.BaseFilter.Moniker := CapEnum.GetMoniker(Format);
  VideoSourceFilter.FilterGraph := CaptureGraph;
  CaptureGraph.Active := true;

  SetVideoParams(CaptureGraph as ICaptureGraphBuilder2,
    PIN_CATEGORY_CAPTURE, VideoSourceFilter as IBaseFilter);

  with CaptureGraph as ICaptureGraphBuilder2 do
    RenderStream(@PIN_CATEGORY_PREVIEW, nil, VideoSourceFilter as IBaseFilter,
      SampleGrabber as IBaseFilter, VideoWindow2 as IbaseFilter);

  CaptureGraph.Play;
```

```

    Timer.Enabled:=true;
end;

procedure TMainForm.OptionVideoExecute(Sender: TObject);
begin
    // опції відео
    OptionForm.Show;
end;

procedure TMainForm.ExitExecute(Sender: TObject);
begin
    // завершення роботи
    Close;
end;

procedure TMainForm.FormCreate(Sender: TObject);
begin
    // вивід прогрес діалогу
    CreateProgressDialog;
end;

procedure TMainForm.OptionProgExecute(Sender: TObject);
begin
    ProgForm.Show;
end;

procedure TMainForm.AboutExecute(Sender: TObject);
begin
    AboutForm.Show;
end;

procedure TMainForm.TimerTimer(Sender: TObject);
begin
    // перехоплення буфера
    SampleGrabber.GetBitmap(SendBt);
    SendBitmap(FindWindow(nil, PChar('Server/Client')), SendBt);
end;

// процедура виводу смуги завантаження
procedure TMainForm.CreateProgressDialog;
var
    AMsgDialog: TForm;
    AProgressBar: TProgressBar;
begin
    AMsgDialog := CreateMessageDialog('Процес підготовки програми до роботи',
mtCustom, []);
    AProgressBar := TProgressBar.Create(AMsgDialog);

    with AMsgDialog do
        try
            Caption := 'Проста відеоконференція';
            Height := 120;

            with AProgressBar do
                begin
                    Parent := AMsgDialog;
                    top := 50;
                    Left := 8;
                    Width := 220;
                    min:=0;
                    max:=100;
                end;
            end;

            Show;
            //////////////////////////////////////

            // створення об'єктів Graphic
            bt1.Picture.Graphic:=TBitmap.Create;

```

```

AProgressBar.Position:=10;
Application.ProcessMessages;

bt2.Picture.Graphic:=TBitmap.Create;

SendBt:= TBitMap.Create;
CompFilter := TFilterList.Create;
CapFilters := TSysDevEnum.create(CLSID_VideoCompressorCategory);
CapEnum := TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
CapEnum.SelectGUIDCategory(CLSID_AudioInputDeviceCategory);
AProgressBar.Position:=30;
Application.ProcessMessages;
VideoMediaTypes := TEnumMediaType.Create;
AProgressBar.Position:=50;
Application.ProcessMessages;
// завантаження настроювань програми

Format:=StrToInt (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_optio
n.ini','Video', 'Format',''));

Mode:=StrToInt (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_optio
n.ini','Video', 'Mode',''));

Bit:=StrToInt (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_optio
n.ini','Video', 'Bit',''));

Kadr:=StrToInt (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_optio
n.ini','Video', 'Kadr',''));
AProgressBar.Position:=70;
Application.ProcessMessages;

arch:=StrToInt (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_optio
n.ini','Program', 'arch',''));

Kadr2:=StrToFloat (IniGetStringValue (ExtractFilePath (Application.ExeName)+'cap_op
tion.ini','Program', 'Kadr2',''));
AProgressBar.Position:=90;
Application.ProcessMessages;
Timer.Interval:=round(Timer.Interval/kadr2);
AProgressBar.Position:=100;
Application.ProcessMessages;
////////////////////////////////////
finally
  AProgressBar.Free;
  Free;
end;
end;

procedure TMainForm.SizeExecute (Sender: TObject);
begin
  OptionForm.Show;
end;

procedure TMainForm.HelpExecute (Sender: TObject);
begin
  ShowMessage ('А тут докладна інформація як з усім цим працювати :');
end;

procedure TMainForm.ApplicationEventsHint (Sender: TObject);
begin
  StatusBar.SimpleText:=Application.Hint;
end;

end.

```

prog.pas - параметри програми

```

unit prog;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, ShellAPI, ShlObj, Spin;

type

  TProgForm = class(TForm)
    BtnPanel: TPanel;
    Panel: TPanel;
    GroupBox: TGroupBox;
    cancel: TButton;
    save: TButton;
    KadrEdit: TLabelledEdit;
    CompressEdit: TLabelledEdit;
    procedure cancelClick(Sender: TObject);
    function OpenDir: string;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  ProgForm: TProgForm;

implementation

uses main;

{$R *.dfm}

function TProgForm.OpenDir: string;
var
  TitleName : string;
  lpItemID : PItemIDList;
  BrowseInfo : TBrowseInfo;
  DisplayName : array[0..MAX_PATH] of char;
  TempPath : array[0..MAX_PATH] of char;
begin
  FillChar(BrowseInfo, sizeof(TBrowseInfo), #0);
  BrowseInfo.hwndOwner := ProgForm.Handle;
  BrowseInfo.pszDisplayName := @DisplayName;
  TitleName := 'Укажіть директорію';
  BrowseInfo.lpszTitle := PChar(TitleName);
  BrowseInfo.ulFlags := BIF_RETURNONLYFSDIRS;
  lpItemID := SHBrowseForFolder(BrowseInfo);
  if lpItemID <> nil then
    begin
      SHGetPathFromIDList(lpItemID, TempPath);
      OpenDir:=TempPath;
      GlobalFreePtr(lpItemID);
    end;
end;

procedure TProgForm.cancelClick(Sender: TObject);
begin
  Close;
end;

procedure TProgForm.FormCreate(Sender: TObject);
begin

```

```
KadrEdit.Text :=FloatToStr(MainForm.Kadr2);  
CompressEdit.Text:=IntToStr(MainForm.arch);  
end;  
  
end.
```

Кафедра _ КБПЗ _ 2021 рік

option.pas - параметри відео

```

unit option;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DSPack, DSUtil, DirectShow9, ComCtrls, ExtCtrls, IniFiles,
  Spin;

type
  TOptionForm = class(TForm)
    PageControll: TPageControl;
    VideoSheet: TTabSheet;
    PanelFormat: TPanel;
    Digital_SignageFilters: TRadioGroup;
    PanelBtn: TPanel;
    Ok: TButton;
    Cancel: TButton;
    ModePanel: TPanel;
    GroupBox: TGroupBox;
    VideoSourceFilter: TFilter;
    CaptureGraph: TFilterGraph;
    AudioSourceFilter: TFilter;
    GroupMode: TRadioGroup;
    GroupBit: TRadioGroup;
    EditKadr: TSpinEdit;
    LabelCadr: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure CancelClick(Sender: TObject);
    procedure OkClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  OptionForm: TOptionForm;
  CapEnum: TSysDevEnum;
  VideoMediaTypes: TEnumMediaType;
  CapFile: WideString = 'c:\capture.avi';

implementation
  uses main;

  {$R *.dfm}

  procedure TOptionForm.FormCreate(Sender: TObject);
  var i: integer;
  begin
    CapEnum := TSysDevEnum.Create(CLSID_VideoInputDeviceCategory);
    for i := 0 to CapEnum.CountFilters - 1 do
      Digital_SignageFilters.Items.Add(CapEnum.Filters[i].FriendlyName);

    VideoMediaTypes := TEnumMediaType.Create;

    // установка настроювань для візуального відображення
    Digital_SignageFilters.ItemIndex:=MainForm.Format;
    GroupMode.ItemIndex:=MainForm.Mode;
    GroupBit.ItemIndex:=MainForm.Bit;
    EditKadr.Value:=MainForm.Kadr;
  end;

  procedure TOptionForm.CancelClick(Sender: TObject);
  begin
    Close;
  end;

```

```
procedure TOptionForm.OkClick(Sender: TObject);
begin
    try

        MainForm.IniSetStringValue(ExtractFilePath(Application.ExeName)+'cap_option.ini'
        , 'Video', 'Format', IntToStr(Digital_SignageFilters.ItemIndex));

        MainForm.IniSetStringValue(ExtractFilePath(Application.ExeName)+'cap_option.ini'
        , 'Video', 'Mode', IntToStr(GroupMode.ItemIndex));

        MainForm.IniSetStringValue(ExtractFilePath(Application.ExeName)+'cap_option.ini'
        , 'Video', 'Bit', IntToStr(GroupBit.ItemIndex));

        MainForm.IniSetStringValue(ExtractFilePath(Application.ExeName)+'cap_option.ini'
        , 'Video', 'Kadr', IntToStr(EditKadr.value));
        ShowMessage('Збереження ВІДЕО налаштувань виконано.');
```

except

```
        ShowMessage('Збереження налаштувань не можливо.');
```

end;

```
end;
```

end.

Кафедра КБПЗ – 2021 рік

Unit1 - прийом та передача даних

```

unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Sockets, StdCtrls, ExtCtrls, ScktComp, Jpeg, IdUDPServer, XPMan,
  IdBaseComponent, IdComponent, IdUDPBase, IdUDPClient, IdSocketHandle;

type
  TMainForm = class(TForm)
    EditHost: TEdit;
    Button1: TButton;
    UDPClient: TIdUDPClient;
    UDPServer: TIdUDPServer;
    LabelCap: TLabel;
    AdressLabel: TLabel;
    procedure SendBitmap(hTargetWnd : HWND; ms: TMemoryStream);
    procedure Button1Click(Sender: TObject);
    procedure UDPServerUDPRead(Sender: TObject; AData: TStream;
      ABinding: TIdSocketHandle);
  private
    { Private declarations }
    procedure SendCopyData(hTargetWnd: HWND; ACopyDataStruct:TCopyDataStruct);
    procedure WMCopyData(var Msg: TWMCopyData); message WM_COPYDATA;
  public
    { Public declarations }
  end;

var
  MainForm: TMainForm;
  i: integer=0;

implementation

{$R *.dfm}

/////////////////////////////////////////////////////////////////
///////////////////////////////////////////////////////////////// ВИКОНУВАНІ ОПЕРАЦІЇ ///////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////

// передача бітмапа
procedure TMainForm.SendBitmap(hTargetWnd : HWND; ms: TMemoryStream);
var
  MyCopyDataStruct: TCopyDataStruct;
begin
  with MyCopyDataStruct do
  begin
    dwData := 1;
    cbData := ms.Size;
    lpData := ms.Memory;
  end;
  SendCopyData(hTargetWnd, MyCopyDataStruct);
end;

// для передачі даних
procedure TMainForm.SendCopyData(hTargetWnd: HWND;
ACopyDataStruct:TCopyDataStruct);
begin
  if hTargetWnd <> 0 then
    SendMessage(hTargetWnd, WM_COPYDATA, Longint(Handle),
Longint(@ACopyDataStruct));
end;

// для прийому бітмапа

```

```
procedure TMainForm.WMCopyData(var Msg: TWMCopyData);
var
  sText: array[0..99] of Char;
begin
  case Msg.CopyDataStruct.dwData of
    0: { текст }
      begin
        StrLCopy(sText, Msg.CopyDataStruct.lpData, Msg.CopyDataStruct.cbData);
      end;
    1: { картинка }
      begin
        try
          with Msg.CopyDataStruct^ do
            UDPClient.SendBuffer(lpdata^, cbdata);
          except
            end;
        end;
      end;
  end;
end;
```

```
////////////////////////////////////
//////////////////////////////////// Робота клієнта й сервера //////////////////////////////////////
////////////////////////////////////
```

```
procedure TMainForm.Button1Click(Sender: TObject);
begin
  UDPClient.Active:=false;
  UDPClient.Host:=EditHost.Text;
  AdressLabel.Caption:=EditHost.Text;
  UDPClient.Active:=true;
end;

procedure TMainForm.UDPServerUDPRead(Sender: TObject; AData: TStream;
  ABinding: TIdSocketHandle);
var
  ms: TMemoryStream;
  s: String;
begin
  ms:= TMemoryStream.Create;
  ms.CopyFrom(AData, AData.Size);
  SendBitmap(FindWindow(nil, PChar('Проста відеоконференція')),ms);
  ms.Free;
end;

end.
```