

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

”Допущено до захисту”

Завідувач кафедри кібербезпеки
та програмного забезпечення

д.т.н., професор

_____ Олексій СМІРНОВ

“ ____ ” _____ 20__ р.

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

**“ Дослідження та програмна реалізація системи кібербезпеки
для охоронної системи”**

Виконав здобувач вищої освіти

II курсу, групи _____

ОПП «Комп’ютерні науки»

спеціальності 123 «Комп’ютерна Інженерія»

_____ Нетребенко В.В.

« ____ » _____ 20__ р.

Керівник проекту

кандидат технічних наук, доцент

_____ Віктор БОСЬКО

« ____ » _____ 20__ р.

Рецензент _____

м. Кропивницький

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

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
Олексій СМІРНОВ
« 06 » вересня 2021 року

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

Нетребенку Владиславу Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи кібербезпеки для охоронної системи

2. Керівник роботи Босько Віктор Васильович, кандидат техн. наук, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу №42-13 від 02.08.2021 року

3. Строк подання роботи до захисту 15.12.2021 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи кібербезпеки для охоронної системи

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

1. Призначення та область використання. 7. Економічна ефективність

2. Перегляд аналогічних існуючих систем. розробленої програми.

3. Опис і обґрунтування проектних рішень. 8. Заходи з охорони праці та техніки

4. Етапи програмування системи. безпеки.

5. Впровадження системи в промислову експлуатацію. 9. Висновки.

експлуатацію.

6. Наукова новизна

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

Наукова новизна 1 аркуш

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

Діаграма процесів 1 аркуш

Показники економічної ефективності 1 аркуш

6. Консультанти по роботі, із зазначенням розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В., к.т.н., доцент	05.09.2021	12.11.2021
Охорона праці	Оришака О.В., к.т.н., доцент	06.09.2021	13.11.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2021 р.	
3.	Розробка моделі компонента	20.10.2021 р.	
4.	Розробка структур даних	25.10.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2021 р.	
6.	Програмування алгоритмів	10.11.2021 р.	
7.	Розрахунок економічної ефективності	13.11.2021 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2021 р.	
9.	Оформлення ПЗ	17.11.2021 р.	
10.	Попередній захист роботи	04.12.2021 р.	

Дата видачі завдання

« 06 » вересня 2021 р.

Підпис керівника

(прізвище та ініціали)

Завдання прийнято до виконання

« 06 » вересня 2021р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Нетребенко В.В. Дослідження та програмна реалізація системи кібербезпеки для охоронної системи.

123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній магістерській роботі розроблено програмне забезпечення, яке призначено для охоронної системи об'єкта з впровадженням підсистеми кібербезпеки для захисту системи від зовнішніх впливів.

Метою розробки є програмне забезпечення системи кібербезпеки охоронної системи.

Об'єктом дослідження є процес забезпечення управлінням насосної станції.

Предметом дослідження є методи забезпечення управління охоронними системами з впровадженням підсистем кібербезпеки.

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

Результат роботи – програмна реалізація охоронної системи з підсистемою кібербезпеки.

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 C++ та PHP.

Ключові слова: комп'ютерна інженерія, мікропроцесор, кібербезпека, охоронна система.

ABSTRACT

Netrebenko V.V. Research and software implementation of cybersecurity system for security system.

123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi 2021.

In this master's thesis, software is developed that is designed for the security system of the object with the introduction of a cybersecurity subsystem to protect the system from external influences.

The purpose of the development is the software of the cybersecurity system.

The object of research is the process of providing control of the pumping station.

The subject of the research is the methods of ensuring the management of security systems with the introduction of cybersecurity subsystems.

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

The result is a software implementation of a security system with a cybersecurity subsystem.

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 PC with Windows XP / Vista / 7/8/10.

The program is developed in C ++ and PHP.

Keywords: Keywords: computer engineering, microprocessor, cyber security, security system.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми магістерської роботи	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	41
2.3 Розгорнута постановка завдання	43
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	44
3.1 Опис функціонування системи.....	44
3.2 Розробка структурної схеми	45
3.3 Розробка функціональної схеми.....	46
3.4 Розробка діаграми процесів.....	47
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ ...	49
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	62
4.2 Захист розробленого програмного забезпечення	65
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	69
6 НАУКОВА НОВИЗНА	75
7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ... 76	76
7.1 Техніко-економічне обґрунтування теми магістерської роботи	76

						ВКРМ-123.21.0033.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Петребенко В.В.				Дослідження та програмна реалізація системи кібербезпеки для охоронної системи	Лім.	Аркуш	Аркушів
Перев.	Босько В.В.					Б	1	113
Н.контр.	Гермак В.С.				ЦНТУ КІ-20М			
Затв.	Смірнов О.А.							

7.2 Розрахунок трудомісткості розробки програмної продукції.....	78
7.3 Визначення чисельності виконавців і планового фонду зарплати	80
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань.....	84
7.5 Визначення собівартості розробки та ціни програмної продукції	89
7.6 Визначення об'єму капітальних вкладень у споживача продукції	93
7.7 Визначення експлуатаційних витрат	93
7.8 Визначення економічної ефективності програмної продукції.....	95
7.9 Висновки	96
8 ЗАХОДИ ЩОДО ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ.....	97
8.1 Вступ	97
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером	98
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста... ..	99
8.4 Розробка заходів з умов поліпшення охорони праці	102
8.5 Розрахункова частина	103
9 ОСНОВНІ ВИСНОВКИ.....	106
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	108

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І
ТЕРМІНІВ**

ОС	–	Охоронна система
ПЗ	–	Програмне забезпечення
BIOS	–	Basic Input-Output System
МК	–	Мікроконтролер
RAID	–	Redundant Array of Inexpensive Disks
ТРМ	–	Універсальний вимірювач
МДВВ	–	Модуль дискретного вводу/вивода
ПКП	–	Програмний контрольно-програмуємий
ПЛК	–	Програмований логічний контролер
СКД	–	Система контролю доступу
ПЗО	–	пристрій зв'язку з об'єктом
АЦП	–	аналогово-цифровий перетворювач
АСУ	–	автоматизована система управління

ВКРМ-123.21.0033.00.00.ПЗ

Арк.

3

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

ВСТУП

Актуальність теми. Відеоспостереження – це процес, реалізований з використанням технічних рішень, призначених для візуального огляду або охорони територій, об'єктів, об'єктів. За останні роки в цій сфері відбулися найбільш суттєві зміни та впровадження новітніх наукових методів аналізу відеоінформації, підвищення швидкості та якості запису. Система відеомоніторингу складається з камер та монітора (комплексу моніторів), за допомогою якого здійснюють візуальний огляд території, що охороняється. Зображення також записується на потрібний носій. Сьогодні ринок все більше захоплюють цифрові камери, які поєднують прийнятну ціну і високий технічний рівень. Цифрові камери забезпечують краще зображення, мають широкий спектр функцій, а зображення можна записати безпосередньо на жорсткий диск комп'ютера або до хмари. Установка цифрових систем охоронного відеоспостереження на різних типах об'єктів здійснюється при правильному виборі відеокамери та іншого обладнання, де витрати зазвичай відіграють велику роль для замовника. Ціни на відеокамери сьогодні варіюються в залежності від характеристик відеокамери та виробників. Хороших дешевих відеокамер немає. Установка якісної відеокамери забезпечить хорошу роботу системи відеоспостереження, а термін служби відеокамери буде значно довшим.

Відеоспостереження, як і цифрова система, являє собою програмно-технічний комплекс, призначений для організації технологічних і охоронних телевізійних систем. Це дозволяє організувати відеоспостереження, як за місцевими, так і територіально розташованими об'єктами. Сучасні цифрові системи відеомоніторингу дозволяють запрограмувати реагування сигналізаційних пристроїв у разі виникнення тривоги. Приховане відеоспостереження призначене для місць, де встановлена відеокамера не повинна бути помітна. Установка такої відеокамери проводиться таким чином,

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

що при уважному огляді на місце її встановлення можна позначити лише невелику крапку. При використанні такого типу відеоспостереження можливе встановлення відеокамер у внутрішніх об'єктах, але бажано, щоб вони були стаціонарними. Бездротове відеоспостереження сконструйовано таким чином, що замість кабелю, встановленого між відеокамерою та відеоприймачем, використовується пара (передавач/приймач), яка передає/приймає відеосигнал по ефірі. Будуючи проект, встановлюючи різні системи, будь то відеоспостереження чи інша охоронна система, потрібно звернути увагу на безліч факторів, які будуть грати роль в роботі будь-якої системи відеоспостереження.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи кібербезпеки для охоронної системи об'єкта.

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

- Огляд існуючих систем захисту об'єктів.
- Дослідження системи захисту.
- Програмна реалізація системи кібербезпеки для захисту охоронної системи від взлому.

Об'єктом дослідження є процес захисту об'єкта від крадіжок та кібератак на систему з метою взлому охоронної системи.

Предметом дослідження є методи реалізації охоронних систем та систем кіберзахисту.

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

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

- Удосконалено систему кіберзахисту для охоронної системи об'єкта.
- Проведено огляд технологій кіберзахисту охоронних систем;

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

– Розроблено вітчизняний продукт та алгоритми які дозволяють успішно вирішувати задачі захисту автоматизованої охоронної системи від зовнішніх впливів, що дозволило отримати економічний ефект від впровадження.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі для реалізації систем охорони з сучасним кіберзахистом системи від взлому.

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

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи кіберзахисту для охоронної системи, є актуальною задачею, яка потребує вирішення у даній магістерській роботі.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Останнім часом питання безпеки в компаніях і в повсякденному житті стає все більш актуальним. Це пов'язано із погіршенням криміногенної ситуації, проявами зловживання владою, вимагання, корупції тощо. Тому керівники багатьох компаній і бізнес-структур замислюються про встановлення систем безпеки в офісах, складах і виробничих приміщеннях. Система об'єктного відеоспостереження є одним з найпопулярніших засобів серед існуючих систем безпеки. Це особливо важливо і необхідно на тих об'єктах, де з тих чи інших причин неможливо забезпечити безперервну безпеку. Система відеомоніторингу дає можливість безперервно спостерігати за всім периметром будівлі, як всередині, так і зовні, що дозволяє охоронцям якомога швидше реагувати на протиправні дії, зафіксовані відеокамерою.

Наразі компанія вживає заходів щодо покращення стану систем відеоспостереження. Ця система повинна контролювати роботу персоналу і одночасно виконувати функцію пожежної сигналізації та системи безпеки.

По-перше, на заводах і підприємствах, які займають великі площі, розміщення систем відеоспостереження дозволить значно знизити витрати на найм і оплату охоронців. Установка такого пристрою дозволить заощадити не тільки гроші, але і контроль над усією територією компанії. Системи відеоспостереження підвищують ефективність у надзвичайних ситуаціях.

По-друге, системи відеоспостереження часто відлякують потенційних зловмисників. Знижується ризик крадіжки, пошкодження майна та проникнення в будівлю. Якщо є порушення, зловмисників можна знайти за відео, яке передається з камер відеоспостереження. Однак така функція

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

можлива тільки в тому випадку, якщо система відеоспостереження розроблена і встановлена професіоналами.

1.2 Область застосування

Комплексний захист власного житла неможливий без швидкого реагування на зовнішнє втручання. Ви можете досягти цього, лише придбавши комплект відеомоніторингу у професійної компанії. Умови експлуатації в приватному будинку встановлюють основні параметри використовуваних камер і моніторів:

а) Естетика. Це найважливіший критерій при виборі способу моніторингу свого будинку. Останнім часом з'явилося багато цифрових бездротових рішень, які дозволяють значно підвищити конфіденційність і зручність розміщення системи відеоспостереження в будинках різної конфігурації.

б) Компактність. Цей параметр більш актуальний для робочого місця оператора. Оператор відеоспостереження може своєчасно реагувати на події, що відбуваються.

в) Доступність. Завдяки розвитку цифрових технологій для приватного користування стали доступні системи відеоспостереження високої чіткості з можливістю запису в нічний час.

Відеоспостереження в офісах, холах і конференц-залах

Особливий тип систем управління, де важлива якість зображення та звуку. У деяких випадках для спостереження за діями співробітників у робочий час може знадобитися приховане відеоспостереження.

Такі системи відеоспостереження стануть ефективним засобом стимулювання якості роботи.

Відеоспостереження для будівель

Цей ефективний спосіб огляду оточення, фіксації проїзду та проїзду автомобілів може бути частиною комплексної системи безпеки об'єктів будь-якої складності. Сучасні відеореєстратори дозволяють не тільки розпізнавати номери

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

автомобілів і обличчя людей, а й автоматично порівнювати їх з наявними в базі даних. Результатом такого порівняння може стати відкриття бар'єру або, навпаки, заклик до захисту. Придбати відеоспостереження такого рівня може бути непросто, але при комплексному підході до безпеки будівництва та будівництва ці інвестиції окупаються. Для великої системи відеоспостереження на перший план зазвичай виходять такі характеристики:

- а) функціональність;
- б) універсальність;
- в) за будь-якої погоди;
- г) Вміння працювати в темряві.

Всім цим вимогам відповідає нове покоління цифрових камер з інфрачервоним підсвічуванням.

Приховане відеоспостереження та відеоспостереження в темних місцях

Для цього використовуються спеціалізовані відеокамери, відносно компактні, оснащені інфрачервоним освітленням. Такий спосіб освітлення дозволяє домогтися гарного освітлення при цьому, не розкриваючи положення камери. Це основний спосіб освітлення відеокамери в темряві, поширений сьогодні.

Такі відеокамери можна використовувати для відеоспостереження в під'їзді, підвалі або на горищі.

Бездротова система відеоспостереження

У цьому випадку, якщо вам потрібно забезпечити автономність або мобільність реєстраторів, використовуйте системи відеоспостереження, які використовують бездротові мережі передачі даних Wi-Fi, 3G, 4G і GPRS. Це значно спрощує встановлення комплексної системи безпеки, дозволяє розташовувати відеокамери в нестандартних місцях, забезпечуючи чудовий огляд околиць. Мобільні системи відеоспостереження також широко поширені в дорожньому русі та всередині приміщень, де не допускається зберігання

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

додаткових кабелів. Таким чином, оператор відеоспостереження отримує таке ж чітке зображення на моніторі, як і при використанні звичайних камер. Затримки передачі сигналу можуть виникати, коли камера знаходиться далеко від центру обробки даних.

Виходячи з вищесказаного, розробка програмного забезпечення кібербезпеки для системи безпеки є актуальним завданням, яке потребує вирішення в даній магістерській роботі.

Кафедра КБПЗ – 2021 рік

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми магістерської роботи

Системи відеоспостереження призначені для візуального спостереження за об'єктом, що охороняється, за допомогою відеокамер. Сучасні цифрові системи відеоспостереження відрізняються функціональністю, невеликими розмірами та універсальністю. Вони дозволяють швидко встановити відеоспостереження на об'єкт будь-якої складності, і багатьма такими системами може керувати пересічний користувач ПК.

Створені системи відеоспостереження реалізують такі функції:

- відеоспостереження за всіма входами та виходами в будівлю та околиці;
- нічний перегляд інфрачервоних відеокамер;
- відеозапис;
- дистанційний моніторинг (також в Інтернеті);
- перегляд зображень з відеокамер, як на сенсорних панелях управління,

так і в телемережі будинку.

Загальна схема розподіленої системи цифрового відеоспостереження

- моніторинг відеокамер для всіх входів і виходів будівлі та оточення;
- нічний перегляд інфрачервоних відеокамер;
- відеозапис;
- дистанційний моніторинг (у тому числі через Інтернет);
- Перегляд зображень із відеокамер як на сенсорних панелях керування, так і в мережі домашнього телебачення.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

яку точку запису в налаштуваннях. Такі камери відеоспостереження здійснюють автоматичний моніторинг за заданим сценарієм.

Види систем оповіщення

Види сигналізації

Сигналізація - отримання, обробка, передача в заданій формі споживачем технічними засобами інформації про несанкціоновану подію на об'єктах, що охороняються.

Основні функції сигналізації забезпечуються різними технічними засобами. Для виявлення використовуються сповіщувачі, а приймальні та керуючі пристрої та периферійні пристрої використовуються для обробки, запису інформації та генерації контрольних тривог.

Пожежна сигналізація призначена для раннього виявлення точки займання та створення контрольних сигналів для систем пожежної сигналізації та автоматичного пожежогасіння.

Крім цих функцій, пожежна сигналізація повинна складатися з команд для інтеграції автоматичного протипожежного та димового обладнання, систем пожежної сигналізації, технологічного, електричного та іншого інженерного обладнання будівель.

Системи безпеки виконують завдання своєчасного сповіщення служби безпеки (у нашому випадку власника) про несанкціоноване проникнення або спробу проникнення осіб у будинок або його окремі приміщення із зазначенням дати, місця та часу порушення. запас міцності.

Системи охорони та протипожежного захисту поєднують в собі всі особливості охоронно-пожежної сигналізації.

Пожежна сигналізація

Пожежна сигналізація - це сукупність технічних засобів, призначених для виявлення, обробки, передачі в заданій формі пожежної сигналізації, спеціальної інформації та/або видачі команд на включення системи пожежної сигналізації,

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

видачі контрольних сигналів в автоматичних системах пожежогасіння та системах планування які використанні у разі пожеж.



Рисунок 2.2 – Пожежні оповіщувачі

Після отримання сигналу пожежної тривоги диспетчерська система видає такі команди:

- включення системи пожежної сигналізації;
- включення системи пожежогасіння;
- відключення вентиляції;
- включення систем димовидалення;
- включення повітряної підтримки в протипожежні частини будівлі;
- опускання ліфтів на евакуаційні поверхи та блокування їх на цих поверхах тощо.

У системі пожежної сигналізації використовуються сповіщувачі (датчики), які реагують на дим, тепло або полум'я. Повідомлення про виникнення горіння також може бути видано в систему за допомогою ручного пожежного сповіщувача, який зазвичай знаходиться на шляхах евакуації з приміщення і активується безпосередньо вручну.

інших причин, що зменшує витрати на технічне обслуговування під час експлуатації. Крім того, адресно-аналогові системи дозволяють змінювати фіксований поріг чутливості сповіщувачів у програмі без переривання пожежної сигналізації, якщо їх необхідно адаптувати до умов експлуатації на місці.

Переваги використання адресно-аналогових систем пожежної сигналізації:

- автоматичне виявлення пожежі та тривоги;
- автоматичне повідомлення системи пожежної сигналізації працівникам, персоналу та відвідувачам про факт (загрозу) пожежі;
- визначення видів вогню - відкритий вогонь, дим;
- контроль інтенсивності задимлення та поширення вогню;
- дистанційне оповіщення про пожежу (передавання текстових і голосових повідомлень на мобільні та стаціонарні телефони, звукові оповіщення тощо);
- одночасний контроль великих і розподілених будівель (багатопверхових будинків, будівельних комплексів);
- можливість використання прихованих пожежних сповіщувачів у потоці та у вентиляційних коробах;
- автоматичне реагування системи на прийом пожежного сигналу;
- сучасні алгоритми управління вогнем та звітування;
- можливість повної інтеграції з іншими системами безпеки, взаємодії з інженерними системами.

Сучасні комерційні об'єкти мають велику площу та поверховість, тому обладнати їх адресною системою пожежної сигналізації просто необхідно, адже лише тоді власники можуть розраховувати на мінімальний збиток у разі пожежі.

Для забезпечення повної безпеки об'єкта необхідно виявити всі можливі фактори загрози. Факторами загрози можуть бути: вторгнення, небажане вилучення об'єктів, приховане видалення інформації (аудіо-відеозапис, комп'ютерні атаки тощо), технологічні аварії (витік води, газу тощо), пожежа.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Виходячи з цього, оптимізувати заходи безпеки, проектувати комплексні системи безпеки з необхідними компонентами: системи контролю та управління доступом; інженерний захист будівлі; захист інформації.

Системи оповіщення поділяються на типи за ступенем інтеграції різних типів датчиків, контрольно-вимірювальних приладів і периферійних пристроїв.

Існує кілька основних принципів виявлення пожежі, які ґрунтуються на розпізнаванні різних його характеристик (димоутворення, теплоутворення, інфрачервоне випромінювання тощо) і використовуються в сучасних системах пожежної сигналізації. Існує кілька типів пожежних сповіщувачів, включаючи іонізаційні та оптичні димові сповіщувачі, теплові та комбіновані сповіщувачі, світлові, теплові кабелі та системи раннього виявлення диму на пробах повітря.

Електроконтактні сповіщувачі є найпростішим типом сповіщувачів безпеки. Вони являють собою тонкий металевий провідник (фольга, дріт), спеціально прикріплений до об'єкта або конструкції, що захищається.

Вони призначені для захисту будівельних конструкцій (скла, дверей, люків, воріт, неінвестиційних перегородок, стін тощо) від несанкціонованого проникнення через них шляхом руйнування.

Магнітноконтактні (контактні) сповіщувачі призначені для блокування відкривання різних будівельних конструкцій (дверей, вікон, лазів, воріт тощо). Магнітно-контактний детектор складається з магнітно-герметичного детектора керований контакт (геркон) і магніт у пластиковому або металевому немагнітному корпусі.

Магніт встановлюється на рухомій (відкривається) частині будівельної конструкції (дверне полотно, стулку тощо), а магнітокерований контакт - на нерухомій (дверна коробка, віконна рама тощо).

Електричні контактні сповіщувачі, такі як кінцеві вимикачі колії, використовуються для блокування великих конструкцій, що відкриваються (розсувні та розпашні ворота), які мають значний люфт.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Ударні сповіщувачі призначені для блокування різних скляних конструкцій (вікон, вікон, вітражів тощо) від руйнування. Сповіщувачі складаються з блоку обробки сигналів (BOS) і 5 ... 15 датчиків розбиття скла (DRS). Розташування компонентів детектора (BOS і DRS) визначається кількістю, взаємним розташуванням і площею заблоковані склопакети.

П'єзоелектричні сповіщувачі призначені для блокування від руйнування будівельних конструкцій (стіни, підлоги, стелі тощо) та окремих об'єктів (сейфи, металеві шафи, банкомати тощо). При визначенні кількості сповіщувачів даного типу та місця їх встановлення на споруді, що захищається, необхідно враховувати, що їх можна використовувати при 100 або 75% охопленні.

заблокований простір. Площа кожної незахищеної ділянки закритої поверхні не повинна перевищувати 0,1 м².

Оптоелектроніку поділяють на активну і пасивну. Активні оптико-електронні сповіщувачі генерують тривожне повідомлення, коли відбитий потік змінюється (однопозиційні детектори) або зупиняється (змінюється) потік, який отримує (два детектора позиції) енергію інфрачервоного випромінювання, викликаного переміщенням порушника в зоні виявлення. Зона виявлення таких сповіщувачів має форму променевого бар'єру, утвореного одним або кількома паралельними вузькоспрямованими променями, розташованими у вертикальній площині. Зони виявлення різних сповіщувачів зазвичай відрізняються довжиною і кількістю променів. Конструктивно активні оптикоелектронні сповіщувачі зазвичай складаються з двох окремих блоків: блоку випромінювання (БВ) і блоку приймача (БП), розташованих на робочій відстані (діапазоні).

Активні оптико-електронні сповіщувачі використовуються для захисту внутрішнього та зовнішнього периметра, вікон, вікон та доступів до окремих об'єктів (сейфів, музейних експонатів тощо).

Найпоширенішими є пасивні оптикоелектронні сповіщувачі, оскільки за допомогою спеціально розроблених оптичних систем (лінз Френеля) можна легко

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

і швидко отримувати зони виявлення різної форми та розміру та використовувати їх для захисту приміщень.

будь-якої конфігурації, будівельних конструкцій та індивідуальних об'єктів .

Принцип дії сповіщувачів заснований на реєстрації різниці між інтенсивністю інфрачервоного випромінювання, що виходить від тіла людини, і температурою навколишнього середовища. Чутливим елементом сповіщувачів є піроелектричний перетворювач (піроприймач), який

фокусує інфрачервоне випромінювання за допомогою дзеркала або оптична система лінз (найпоширенішою є друга).

Зона виявлення детектора - це просторова дискретна система, що складається з елементарних чутливих зон у вигляді променів, розташованих в один або кілька шарів, або у вигляді тонких широких пластин, розміщених у вертикальній площині (наприклад, «завіса» або «шарнір»). .

Завдяки можливості створення зон виявлення різної конфігурації, пасивні інфрачервоні оптоелектронні сповіщувачі мають універсальне застосування і можуть використовуватися для блокування об'ємів просторів, точок концентрації, коридорів, внутрішніх периметрів, проходів між полицями, віконних і дверних прорізів тощо. підлоги, стелі, невеликі кімнати, склади тощо.

Ємнісні сповіщувачі призначені для блокування металевих шаф, сейфів, окремих предметів, створення захисних бар'єрів. Принцип дії заснований на зміні електричної ємності чутливого елемента (антени) при наближенні або дотику до особи, яка охороняється. У цьому випадку захищений об'єкт необхідно встановити на підлогу з хорошим ізоляційним покриттям або на ізоляційний ущільнювач. Допускається підключення до одного сповіщувача в кімнаті кілька металевих сейфів або шаф. Кількість елементів, що підключаються, залежить від їх місткості, конструктивних елементів приміщення і уточнюється при монтажі.

Звукові (акустичні) сповіщувачі призначені для блокування скляних конструкцій (вікон, вікон, вітражів тощо) від розбиття. Принцип дії цих

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

сповіщувачів заснований на безконтактному методі акустичного контролю руйнування склотканини, що виникає при його руйнуванні коливаннями в діапазоні частот звуку і поширюється в повітрі.

При встановленні сповіщувача всі захищені ділянки скляної конструкції повинні знаходитися в межах його безпосереднього огляду.

Ультразвукові детектори призначені для блокування об'єму закритих приміщень. Принцип дії заснований на реєстрації збурень у полі пружних хвиль ультразвукового діапазону, створюваних спеціальними випромінювачами, при переміщенні в полі виявлення людини. Зона виявлення детектора має форму еліпсоїда обертання або форму сльози.

Через низьку шумостійкість в даний час практично не використовується.

Радіохвилі призначені для захисту об'єму внутрішніх приміщень, внутрішнього і зовнішнього периметрів, окремих будівель і будівельних конструкцій, відкритих приміщень. Принцип дії радіохвильових сповіщувачів заснований на реєстрації несправностей у НВЧ електромагнітних хвилях, що передаються передавачем і реєструються приймачем сповіщувача під час руху людини в зоні виявлення. Зона виявлення (як і при ультразвуку) має форму обертового еліпсоїда або форму сльози. Зони виявлення різних сповіщувачів відрізняються лише розміром.

Радіохвильові сповіщувачі бувають одно- та двопозиційні. Однопозиційні сповіщувачі використовуються для захисту об'єму внутрішніх і зовнішніх приміщень, двопозиційні – для захисту периметрів.

Вибираючи, встановлюючи та експлуатуючи радіохвилі, слід пам'ятати про одну з їхніх особливостей. Для мікрохвильових електромагнітних хвиль деякі будівельні матеріали та конструкції не є перешкодою (тінь) і проникають вільно, з деяким ослабленням. Тому зона виявлення радіохвильового сповіщувача в деяких випадках може виходити за межі приміщення, що охороняється, що може викликати помилкові тривоги. До таких матеріалів і конструкцій належать, наприклад, тонкі гіпсокартонні перегородки, вікна, дерев'яні та пластикові двері

						ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			20

тощо. Тому радіохвилі не повинні фокусуватися на віконних прорізах, тонких стінах та перегородках, за якими в період захисту можуть переміщатися великі предмети та люди. Не рекомендується використовувати їх поблизу потужного радіопередавального обладнання.

Вони об'єднані комбінацією двох сповіщувачів, заснованих на різних фізичних принципах виявлення, з'єднаних конструктивно і схематично в одному корпусі. Причому вони схематично уніфіковані за схемою «а», тобто тільки при спрацьовуванні обох сповіщувачів формується тривожне повідомлення. Найпоширеніша комбінація інфрачервоних пасивних і радіодетекторів.

Комбіновані охоронні сповіщувачі мають дуже високу стійкість до шуму і використовуються для захисту ділянок об'єктів зі складними перешкодами, де використання інших типів сповіщувачів неможливе або малоефективне.

Комбінованими є два сповіщувачі, засновані на різних фізичних принципах виявлення, конструктивно об'єднані в одному корпусі. Кожен сповіщувач працює незалежно від іншого і має власну зону виявлення та власний вихід для підключення до петлі сигналізації. Найбільш поширене поєднання інфрачервоного пасивного і звукового. Є й інші комбінації.

Сповіщувачі сигналізації призначені для того, щоб вручну або автоматично подавати сигнал на службу внутрішньої безпеки будівлі або в поліцію у разі можливого злочинного нападу на співробітників, клієнтів або відвідувачів будівлі.

В якості сповіщувачів тривоги використовуються різні ручні та ножні кнопки та педалі на основі магнітних та електричних контактних сповіщувачів. Як правило, такі сповіщувачі блокуються в натиснутому стані і повернутися в вихідне положення можна тільки за допомогою ключа.

Для цих же цілей розробляються і застосовуються спеціальні міні-сигналізатори. Вони включають приймач, який підключається до АТС або обміну,

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

і кілька портативних брелоків для бездротової передачі тривожних повідомлень. Деякі брелоки включають датчик падіння.

Радіус дії таких систем становить від кількох десятків до кількох сотень метрів.

Особливе місце серед сповіщувачів тривоги займають сповіщувачі пасток. Вони призначені для того, щоб бити тривогу, коли ви намагаєтеся вкрасти гроші або вкрасти захищений об'єкт, незалежно від діяльності персоналу. Вони імітують пачку грошей у банківській тарі об'ємом 100 банкнот, в яку вмонтовано магніт і в спеціальній підставці, на якій розміщена пачка, магнітний датчик (геркон). При знятті (переміщенні) імітаційного пакета грошей з підставки контакти магнітного датчика розмикаються і на панель сигналізації об'єкта надходить тривожне повідомлення. Існують подібні сповіщувачі-пастки, в які разом з магнітом вбудований спеціальний патрон, що містить кольоровий (помаранчевий) дим, об'ємом 5 м³. димовий склад розпилюють із затримкою часу (3 хв) після спрацювання магнітного датчика.

Димові пожежні сповіщувачі

Димлення є найбільш характерною ознакою пожежі, оскільки майже всі види пожеж супроводжуються утворенням великої кількості невловимих частинок диму.

Тому найчисельнішою та найпоширенішою групою пожежних сповіщувачів є димові сповіщувачі, які реалізують різні принципи виявлення часток диму залежно від їх розміру, кольору тощо.

Пожежні сповіщувачі є основними елементами систем автоматичної пожежної та охоронної сигналізації.

За способом управління пожежники поділяються на ручні та автоматичні. Ручні пожежні сповіщувачі не мають функції виявлення джерела займання, їх завдання – відправити тривожне повідомлення в електричну ланцюг шлейфа сигналізації після того, як людина виявить пожежу та активує її, натиснувши відповідну кнопку запуску.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Автоматичні пожежні сповіщувачі працюють без втручання людини. З їх допомогою виявляється займання за одним або кількома аналізованими ознаками і спрацьовує пожежна тривога при досягненні контрольованого фізичного параметра заданого значення. В якості контрольованих параметрів можуть виступати підвищена температура повітря, виділення продуктів згорання, турбулентний потік гарячих газів, електромагнітне випромінювання тощо. За основними ознаками пожежі сповіщувачі поділяються на теплові, димові, полум'яні, газові та комбіновані. Можна використовувати й інші знаки вогню. Комбіновані реагують на два або більше параметрів, що характеризують зовнішній вигляд джерела займання.

Теплові можна використовувати методом створення аналізованого сигналу, що дозволяє їм реагувати не тільки на підвищення абсолютного значення температури вище максимально встановленого порогу, але і на швидкість збільшення її граничного значення. Тому за характером реакції на зміну керованого характеру їх поділяють на максимальні, диференціальні та максимально-диференціальні. Димові сповіщувачі поділяються на оптоелектронні та іонізаційні.

За способом живлення пожежники поділяються на:

- живлення від шлейфа сигналізації від ПКП або обладнання ПКП;
- живлення від окремого зовнішнього джерела живлення;
- живлення від вбудованого внутрішнього джерела живлення (окремі пожежні сповіщувачі).

Зона виявлення сповіщувача - це зона біля сповіщувача, в якій гарантується його робота в разі пожежі. Цей параметр найчастіше виражається в одиницях площі (м²), контрольованих детектором з необхідною надійністю. Зі збільшенням висоти установки сповіщувача контрольована зона зменшується. Якщо висота установки вище зазначеного максимуму, ефективне виявлення пожежі сповіщувачем не гарантується.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

Захищена зона призначена для сповіщувачів світла максимальна дальність виявлення полігону відкритого вогню та кут огляду залежно від конструкції оптичної системи.

Пожежні сповіщувачі повинні забезпечувати надійне виявлення пожежі в певних охоронюваних зонах. Для цього при виборі необхідно враховувати ймовірний характер пожежі та процес розвитку основних факторів пожежі з часом: підвищення температури, концентрації диму, світлового випромінювання в різних місцях приміщення.

Залежно від виду та кількості горючих матеріалів у вогні може бути перевага в одному або кількох символах.

Найчастіше запалення супроводжується виділенням диму на ранній стадії, тому в більшості випадків краще використовувати димові сповіщувачі. При виборі диму необхідно враховувати, що іонізуючий (радіоізотопний) і оптоелектронний дим мають різну чутливість до продуктів згоряння, частинки диму яких мають різний колір і розмір. Оптоелектронні точкові сповіщувачі краще реагують на легкий дим, характерний для целюлозовмісних матеріалів, а також на дим, що складається з дрібних аерозольних частинок. Іонізатори мають відносно більшу чутливість до продуктів горіння, вони виділяють чорний дим з більшими частинками (наприклад, при горінні гуми).

Місця, де відкрите полум'я, швидше за все, швидко з'явиться під час пожежі, найкраще обладнати сповіщувачами світла.

Температуру доцільно встановлювати якомога раніше в тих випадках, коли при пожежі забезпечується значна потужність вогню, а при пожежі відбувається інтенсивне виділення тепла.

Розрахунок загальної кількості сповіщувачів та визначення їх розташування установка повинна проводитися з урахуванням особливостей приміщення, а також вимог нормативно-технічної документації. Нормативно-технічна документація містить відповідні документи, що регламентують загальні питання проектування та монтажу систем, систем пожежної автоматики пожежні

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

та охоронні сигналізації та комплекси, а також експлуатаційна документація на відповідний тип сповіщувача.

Все більшого поширення набувають пожежні сповіщувачі, створені з використанням елементної бази четвертого покоління: спеціалізованих контролерів і мікропроцесорів.

Загальною особливістю таких сповіщувачів з розширеними тактико-технічними можливостями є використання для спільної роботи лише спеціальних пристроїв (бірж), що входять до системи охоронно-пожежної сигналізації підприємства.

Використання комп'ютерних технологій дає змогу створювати адресні та адресні аналогові пожежні сповіщувачі, які передають інформацію про своє місцезнаходження на панель управління ЦП, що забезпечує точне відтворення зображення та аналіз процесу виникнення та розвитку пожежі. Вони виконуються автоматично або за запитом з центру управління працездатністю та передають дані про робочі параметри в цифровому вигляді.

У таких сповіщувачах при необхідності можна регулювати чутливість при зміні умов навколишнього середовища.

Детектори аналогового типу також можуть передавати інформацію про рівень контрольованого параметра. Асортимент сповіщувачів також розширюється за рахунок використання нових технологій.

Наприклад, сучасний закордонний лінійний термічний (кабельного типу) фіксує різницю між нормальною і підвищеною температурою, що дозволяє сформувавши тривогу перед пожежею (димом або пожежею) при перегріванні обстежуваного об'єкта. Сигнал передається в аналоговій формі від сповіщувача на спеціальний пульт управління, що дозволяє визначити відстань до перегрітої області. Такі сповіщувачі можна ефективно використовувати для контролю об'єктів з електрообладнанням, приміщень зі стелями, кабельних трас і каналів.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Іонізаційний димовий сповіщувач

Детектори диму іонізації використовують здатність іонів повітря, щоб залучити частки диму. В електричному полі вимірювальної камери детектора, повітря іонізується за допомогою слабого радіоактивного джерела. Іонізовані, позитивно і негативно заряджені молекули газу рухаються до протилежно заряджених електродів під дією електричного поля. У той же час, електричний струм проходить через вимірювальну камеру, величина якого залежить від кількості і швидкості іонів. В процесі рекомбінації заряду позитивних і негативних іонів при їх русі в камері, число іонів відповідають за перенесення заряду зменшується. Потік вимірювальної камери стабілізується на рівні деякого кінцевого значення, відповідного інший режим роботи детектора. У міру того як частки диму увійти в простір між електродами відкритої вимірювальної камери детектора, вони починають заважати вільному руху іонів. Деякі з іонів, присутніх вступають в контакт з більш важкими частинками диму і залишаються на їх поверхні. Це збільшує рівень заряду рекомбінації і висока інерція цих частинок диму фактично позбавляє їх мобільність і запобігає потраплянню заряду електродів. Це призводить до зменшення струму вимірювальної камери, яка служить в якості критерію для прийняття рішення, чи слід видавати сигнал тривоги на детектор.

Іонізаційні димові сповіщувачі підходять для раннього виявлення пожеж, частинок диму будь-якого розміру та кольору.

Димовий пожежний сповіщувач, заснований на принципі ослаблення світлового потоку

Цей принцип виявлення заснований на зміні інтенсивності світла при проходженні диму. У вимірювальній камері детектора на певній відстані один від одного знаходяться джерело світла і фотоприймач. При відсутності диму в камері випромінювання, яке пропускає джерело світла, майже досягає фотодетектора, який формує певний сигнал S_1 , що відповідає наступному стану детектора. У випадку, якщо частинки диму потрапляють у вимірювальну камеру і потрапляють

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

між джерелом світла і фотодетектором, вимірюваний сигнал зменшується до відповідного значення S_2 , яке фіксується і оцінюється блоком обробки сигналів для прийняття рішення про подачу тривоги. Це зниження сигналу викликано двома явищами. Частинки диму поглинають частину світла. Друга частина розсіяна, тобто відмінна від основного напрямку руху. Ослаблення радіації – це сума поглинання і розсіювання світла. Величина цього ослаблення істотно залежить від відношення розміру часток диму до використовуваної довжини хвилі. Використання сучасного джерела світла зі спектром у видимому та ближньому інфрачервоному діапазоні дозволяє реалізувати описаний вище принцип загасання для лінійних димових сповіщувачів.

Лінійний детектор диму включає приймач, який генерує модульований інфрачервоний промінь, фокусований оптичною системою передавача. При відсутності диму в контрольованій зоні більша частина інфрачервоного випромінювання потрапляє в відбивач, розташований перед сповіщувачем, заломлюється, тим же шляхом повертається до сповіщувача і фокусується на фотодетекторі. Отриманий сигнал відповідає наступному стану детектора.

Якщо в зоні, контрольованій сповіщувачем, з'являється дим, частина ІЧ-випромінювання або поглинається, або розсіюється частинками диму на шляху до відбивача і назад. Таким чином, до приймача потрапляє лише незначна частина ІЧ-випромінювання, що значно зменшує випромінюваний ним сигнал. Це зниження сигналу служить критерієм для прийняття рішення, чи запускати тривогу.

Димові сповіщувачі, засновані на описаному вище принципі, виявляють усі частинки диму, які можуть спричинити послаблюючий ефект, тобто світлі та темні, великі та малі. Тому вони підходять для раннього виявлення пожеж, частинок диму будь-якого розміру та кольору.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Димовий пожежний сповіщувач, заснований на принципі розсіювання світла

За цим принципом всі компоненти системи детектування розміщуються в вимірвальній камері детектора так, щоб світло від джерела не потрапляло до приймача. Це створює мінімальний сигнал S_1 , що відповідає наступному стану детектора. Тільки якщо частинки диму присутня в оптичному каналі вимірвальної камери, частина розсіяного світла надходить в приймач і викликає сигнал, щоб збільшити до величини S_2 , який записується і оцінюється блоком обробки сигналу, щоб прийняти рішення про тривогу .

Щільність диму та оптичні властивості частинок диму мають вирішальний вплив на збільшення сигналу. Великі частинки диму мають набагато більшу здатність до розсіювання світла, ніж дрібні частинки. Крім того, інтенсивність розсіювання зменшується в залежності від відношення розміру частинок до використовуваної довжини хвилі. Тому розмір частинок диму має вирішальне значення для цього принципу виявлення. Крім того, інтенсивність розсіювання частково зменшується за рахунок поглинання світла частинками диму. З цієї причини частинки сажі або чорного диму мають набагато меншу інтенсивність розсіювання, ніж білий дим.

Інтенсивність світлорозсіювання значною мірою залежить від кута, під яким вимірюється розсіяне світло. Тому існують детектори, які використовують як пряме, так і зворотне розсіювання.

Світлорозсіюючі димові сповіщувачі в основному виявляють видимі білі частинки і тому підходять для тих видів пожеж, які характеризуються наявністю білого диму.

Найбільш популярними за частотою використання є охоронна сигналізація з додаванням дешевих функцій пожежної сигналізації.

Призначення таких систем – захист великих площ і будинків (котеджів, підприємств, заводів тощо).

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

даних дозволяє швидко знайти працівника в районі в останній точці переходу, де він пред'явить посвідчення особи.

У кожній точці доступу може бути кілька часових поясів (тимчасові обмеження доступу). Наприклад, працівникові можна дозволити змінюватися лише з 09:00 до 20:00.

Центральним пристроєм системи є контролер, який отримує інформацію про ідентифікатори та керує виконавчими механізмами відповідно до системних алгоритмів.



Рисунок 2.5 – Система контролю доступу

Як потужні пристрої системи контролю доступу можуть використовуватися електромеханічні або електромагнітні замки різних типів, турнікети, автоматичні двері тощо. Об'єктом доступу може бути не тільки людина, а й автомобіль, на якому закріплено спеціальний пристрій. Приводами в даному випадку є ворота та автоматичні приводи воріт.

Поточний стан системи, список повідомлень, звітність, система графічних планів тощо. захищений об'єкт доступний оператору з графічного інтерфейсу програмного забезпечення системи контролю доступу.

Системи контролю доступу можуть бути локальними та розподіленими.

Локальні системи дозволяють виконувати функції контролю доступу на одному територіально розмежованому об'єкті охорони. Як правило, такі системи

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

базуються на єдиному контрольно-приймальному пристрої та мають одне або кілька автоматизованих робочих місць оператора для керування системою.

Локальна система може включати, наприклад, контролер, який реалізує функції контролю входу в будівлю (вхідні турнікети), приміщення, паркінг, а також дозволяє керувати ліфтом.

Розподілені системи використовуються на кількох територіально розосереджених пристроях безпеки. У таких системах є основний контролер – контролер, що дозволяє централізовано керувати всією системою (локальними будівельними контролерами), створювати єдину базу даних працівників і подій, звітувати про всі об'єкти охорони. Використання головного контролера надає можливість централізовано визначати права доступу співробітників у всіх філіях організації, а також визначати місцезнаходження співробітника в будь-якому з відділень.

Сучасні системи контролю доступу можуть обмінюватися даними через мережі Ethernet, що дозволяє використовувати існуючі канали передачі даних для з'єднання філій компанії в одну систему.

Огляд існуючих систем відеоспостереження

Система відеоспостереження «Патріот»

Patriot Video Surveillance System – це проста у використанні та дуже надійна система відеоспостереження.

Характерною особливістю цієї системи відеоспостереження є - швидкий пошук змін всередині, а також вакансій і мультисервера.

Система відеомоніторингу «Патріот» також поставляється у вигляді повністю підготовлених рішень - відеореєстраторів «Патріот».

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31



Рисунок 2.6 – Система відео спостереження «Патріот»

Використовуючи цифрову систему відеоспостереження «Патріот», ви зможете наступне:

З цифровою системою відеоспостереження Patriot ви можете:

- захист території компанії від проникнення сторонніх осіб;
- протидіяти зловживанням, змові чи недбалості персоналу;
- контролювати роботу працівників, час їх приходу та відходу з роботи;
- підвищення дисципліни працівників;
- ефективно контролювати клієнтів;
- об'єктивно аналізувати обставини «нестандартних» ситуацій;
- підвищення надійності та об'єктивності збору та зберігання інформації.

Система відеоспостереження Патріот дозволяє наступне:

перегляд поточної інформації та архівування за період, який вас цікавить (до 16 каналів одночасно з різних відеосерверів);

– використовувати оптимальний режим відображення (реальний, повільний рух, прискорення, стоп - кадр), варіативність зображення (посилення, регулювання різкості і т.д.);

– отримати доступ до даних і керувати системою в будь-який зручний для вас час (за допомогою ПК, КПК, мобільного телефону);

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

- встановити обмеження та пріоритети доступу до системи для різних користувачів;
- використовувати систему як охорону з функцією виявлення руху;
- надсилати повідомлення про реєстрацію тривожних подій електронною поштою, пейджером або мобільним телефоном (SMS).

Для досягнення вищезазначених цілей ми пропонуємо цифрову систему відеоспостереження та моніторингу на базі системи Patriot Pro – як універсальний, якісний та надійний засіб моніторингу та збору інформації у вашій компанії. З його допомогою можна легко організувати спостереження, реєстрацію інформації, її архівування і, при необхідності, передачу на великі відстані по телефонних лініях, стільникових мережах, а також локальних і глобальних мережах (Інтернет). Основним завданням системи відеоспостереження найчастіше є моніторинг і швидке реагування на дії порушників. Однак розвиток сучасних технологій відсунув цю функцію на другий план, ставши не більше ніж однією з можливостей системи.

Окрім оперативного моніторингу та виявлення протиправних дій замовників та персоналу, автоматично створюється доказова база для розслідування нестач, крадіжок, злочинних дій, спорів.

При цьому можливий швидкий пошук та передача окремих фрагментних файлів на носій інформації. Завдяки притаманній цій системі багатофункціональності, а також гнучкості та різноманітності налаштувань, її можна використовувати як для вирішення вузьких конкретних завдань, так і як універсальний комплекс для спостереження, збору та зберігання інформації. В результаті кошти, вкладені в систему, можуть використовуватися більш раціонально і швидше повертатися.

Загалом система відеоспостереження складається з двох частин: відеосервера і відеоклієнта.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Відеосервер - це комп'ютерна система, оснащена картою захоплення зображень, до якої під'єднані відеокамери та встановлено спеціальне програмне забезпечення відеосервера.

Відеоклієнт - це будь-який комп'ютер із програмним забезпеченням відеоклієнта, підключеним до мережі. Мобільний відеоклієнт використовується для перегляду відеоспостереження на мобільному телефоні, смартфоні або комунікаторі.

Страж SMS 8x6 GPS

Система "Страж SMS 8x6 GPS" (рисунок 2.1) призначена для контролю приміщень і автомобілів, та для керування об'єктами.



Рисунок 2.7 - Страж SMS 8x6 GPS

Основними характеристиками цієї моделі є:

Збільшена кількість входів і виходів (7 сенсорних входів, вхід охоронного сигналу / 4 контрольованих виходу, вихід індикатора стану виходу, сирена на виході).

– Використання практично будь-яких типів датчиків (аналогових, цифрових (контактних)).

– Керування за допомогою сигналів DTMF, що дозволяє керувати системою не тільки за допомогою SMS-повідомлень, а й з будь-якого телефону, який підтримує тональний набір, в режимі реального часу.

– Вилучено перевірку та поповнення SIM-карти пристрою.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

– Можливість «інтелектуального GPS-контролю» - сигналізація при зміні GPS-координат.

– Наявність автоматичного внутрішнього підігріву дозволяє працювати при низьких температурах.

– Живлення від 7,5U до 30V DC або від мережі ~ 220V (при використанні мережевого адаптера).

– Температурний діапазон -40C .. + 40C.

Основні переваги «SMS 8x6 GPS Guard»:

– Простота програмування та адміністрування.

– Тривожна інформація про зміну статусу вводів об'єкта передається у вигляді SMS-повідомлень і дзвінків з цього пристрою (до 3-х телефонних номерів).

– Можливі SMS на запит з телефонів абонентів для отримання інформації:

- стан коштів на рахунку;

- стан обладнання (стан захисту, живлення, контрольних входів і виходів);

- про GPS-координати пристрою (якщо підключено GPS-приймач).

– Самою охороною та зовнішніми пристроями, підключеними до нього за командою Dtm, можна дистанційно керувати викликом об'єкта (керування б виходами потужних пристроїв).

– Можливість підключення мікрофона для дистанційного прослуховування кімнати при включенні датчика (протягом 4 хвилин).

– Підтримка мобільних мереж GSM.

– Вбудована літій-іонна батарея (до 2 днів роботи в разі відключення зовнішнього живлення).

– Відсутня міцний зв'язок між настанням події та контролем учасників дії.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

– Довгострокова операція являє собою режим мікро-споживання , що дозволяє працювати від повністю заряджений акумулятор в протягом декількох місяців.

– Контроль споживання струму зовнішніми модулями та їх автоматичне відключення у разі проблем з ними.

Як працює система

При запуску одного з контрольованих датчиків пристрій здійснює набір номера та надсилає SMS-повідомлення на список телефонів (до 3 номерів), попередньо записаних на SIM-карту. Ви можете слухати кімнату протягом чотирьох хвилин (лише будильник).

підключені датчики можуть бути:

– аналогові (випромінювальна напруга з інтервалу 0..15 В) - датчики руху, датчики витоку газу, датчики пожежного диму тощо;

– контакт (цифровий) - кнопка, геркон, датчик пожежі, датчик гучності (IC sensor), акустичний датчик.

Робота датчиків контролюється незалежно один від одного, відповідно до логічних налаштувань, розташованих на SIM-карті. Приклад підключення одного з датчиків (датчика розбиття скла Pygonix) до системи «SMS 8x6 GPS Guard» показано на рисунку 2.8.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

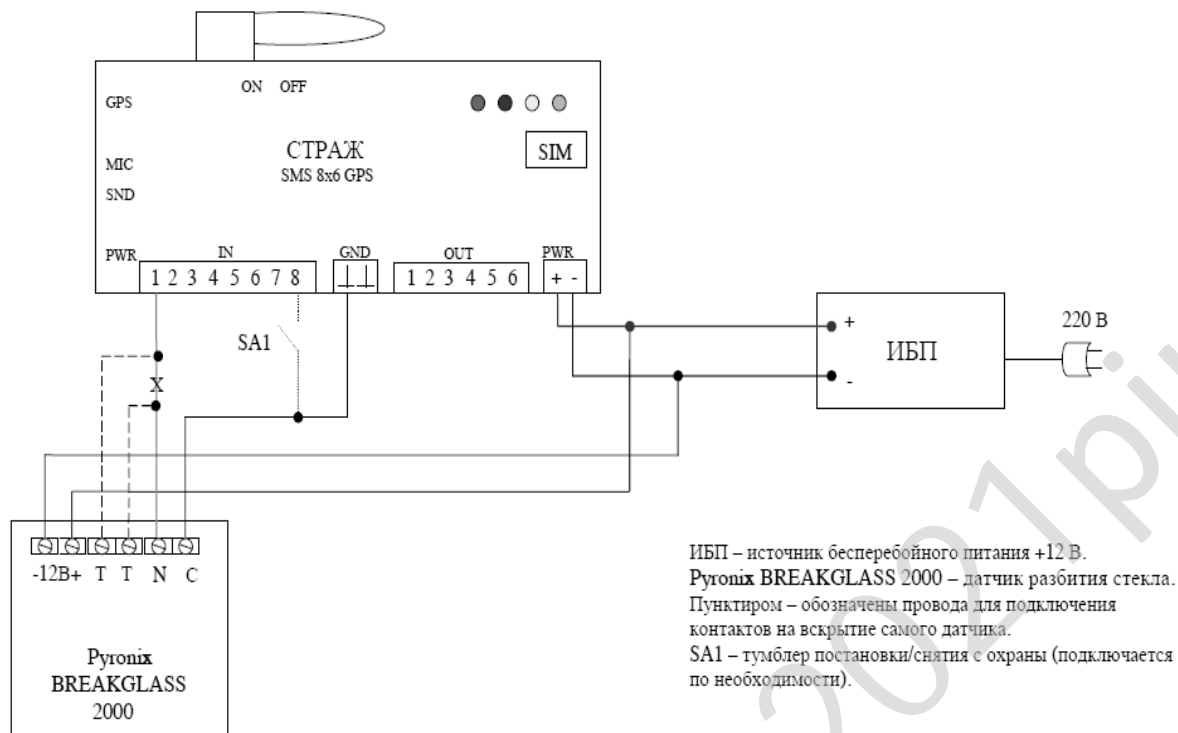


Рисунок 2.8 - Підключення датчика розбиття скла Rygonix до Страж 8x6 GPS

Також за допомогою SMS 8x6 GPS Guards ви можете керувати 6 різними потужними пристроями за допомогою команд DTMF. Наприклад, віддалено деактивувати кімнату.

Виходами для приводів є електронні вимикачі з максимальною напругою перемикання 30 В і максимальним струмом 0,5 А. Низькоенергетичні приводи можна підключати безпосередньо до цих виходів, а більш потужні - тільки через реле. Рейнджер стежить за станом кожного з цих виходів.

Віддалений моніторинг стану системи

За станом системи можна в будь-який момент проконтролювати, зателефонувавши на її номер і відправивши спеціальну команду з DTMF-сигналами. Система надсилає контрольне SMS-повідомлення. Таким чином ви завжди будете в курсі стану системи безпеки і будете негайно проінформовані в разі непередбаченої ситуації. Під час роботи системи постійно контролюються

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

ВКРМ-123.21.0033.00.00.ПЗ

Арк.

37

вихідні стани датчиків, вихідні стани виконавчих механізмів, наявність мережі GSM та напруга живлення.

Система живлення

Рейнджер автоматично заряджає вбудований акумулятор при підключенні до нього зовнішнього джерела живлення (від 7,5 В до 30 В постійного струму або від мережі ~ 220 В (при використанні мережевого адаптера)). У разі відмови зовнішнього джерела живлення система живиться від вбудованого Li-Ion акумулятора до 2 днів. Схема системи розрахована на роботу при негативних температурах навколишнього середовища (для захисту автомобіля та неопалюваних приміщень). Слід мати на увазі, що автоматичний внутрішній обігрів працює тільки при наявності зовнішнього джерела живлення.

Visonic Ga-1rs



Рисунок 2.9 - Visonic GA-1RS

Технічні характеристики пристрою

Технічні властивості пристрою вбудований GSM модем (900/1800); передача цифрових тривожних повідомлень (наприклад, Contact ID) на телефонний приймач (наприклад, RC12 / RC112); передача голосових повідомлень на 4 стаціонарні або мобільні телефони користувача; відправка SMS-повідомлень на 4 мобільні телефони користувач

управління охоронною системою, приладами X10 та прослуховуванням обстановки в приміщенні;

програмування маршрутизації повідомлень з арт-панелі SSB-003;

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

режими роботи: резервний канал GSM, основний канал GSM, тільки GSM, тільки SMS;

вибір групи відправлення SMS (сигнали, несправності, видалення/підготовка тощо);

передача повідомлення у разі втрати зв'язку модуля з панелі арт. SSB-003;

відображення поточного рівня сигналу від базової станції RSSI;

Powermax + блок живлення та/або 9В змінного струму, вбудований акумулятор;

заряджати акумулятор від арт-панелі SSB-003 або адаптера змінного струму;

споживання струму 30 мА / максимум 400 мА (у режимі повідомлення);

підключення до арт-панелі SSB-003 через RS232 (максимальна відстань 12 м);

діапазон робочих температур 0 ... 49С;

розміри 185x108x43 мм;

вага 500 грам.

дистанційне програмування та зчитування панелі SSB - 003 через GSM адаптер;

Страж-gsm-m

«Страж-GSM-M» (рисунок 2.9) призначений для охорони квартир, офісів, дач, автомобілів і інших об'єктів охорони й здійснює оповіщення через мережу стільникового зв'язку стандарту GSM за допомогою дзвінка на задані телефонні номери (до 3-х номерів).

Як працює охоронна система

Фізично «Гард-GSM-M» є модифікованим мобільним телефоном «Motorola M3x88», оснащеним входом для підключення датчиків (в упаковці є один геркон). Можна використовувати будь-який тип датчика з нормально закритим контактним виходом. При цьому вся функціональність телефону збережена.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

Для підвищення ефективності системи безпеки рекомендуємо звернути особливу увагу на конфіденційність місцезнаходження пристрою. Також слід переконатися, що рівень отриманого сигналу достатній на місці установки «Варта GSM-M».

Після запуску (отвори) датчика, пристрій негайно набирає введені номери телефонів в інтервалі 2 хв. В режимі розмови для кожного номера складає 30 секунд. Наявність функції Caller ID передбачається на комутованих телефонів. визначення AON за номером телефону «Варта GSM-M» вказує на те, що сигналізація спрацювала. Оскільки немає необхідності відповідати на виклик, баланс на «Варта GSM-M» не поширюється на абонентську плату, за винятком. «GSM-M Guard» має ряд унікальних функцій, яких немає в інших системах безпеки:

Практично необмежений обсяг час роботи.

Установка систем надзвичайно проста: просто підключіть датчики до телефону, увімкніть телефон і запрограмуйте необхідні номери телефону та текст попередження. Без зайвих проводів.

Телефон має вбудований акумулятор, тому пошкодження джерела живлення 220 В (або відключення автомобільного акумулятора, якщо він використовується як автосигналізація) не вплине на надійність оповіщення.

GSM-M Guard дуже економний завдяки тому, що вам не потрібно приймати вхідні дзвінки. Ви можете стежити за станом кількох об'єктів, оскільки «GSM-M Guard» ідентифікує об'єкт за номером встановленого на ньому мобільного телефону.

Установка системи

Ця система безпеки проста в установці. Телефон монтується в затишному місці, щоб була доступна телефонна клавіатура (з якої програмується, встановлюється і деактивується система), встановлюється необхідна кількість датчиків і підключається до телефонної розетки. Телефон підключається до мережі 220 В через блок живлення, що постачається. Номер та сповіщення

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

запрограмовані.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Програмне забезпечення включає в себе набір комп'ютерних програм, описів та інструкцій щодо їх використання на комп'ютері. Програмне забезпечення поділяється на два комплекси: загальний (операційні системи, операційні оболонки, компілятори, інтерпретатори, програмне забезпечення для розробки додатків, СУБД, мережеві програми тощо) і спеціальний (набір програм, призначених для виконання конкретних завдань у функціональних підсистемах, і приклади елементів керування)

Windows / 7/8/10 є найпоширенішою для сучасних комп'ютерів сьогодні. Кожен з них потребує певних обчислювальних ресурсів, що накладає певні обмеження на характеристики комп'ютера. Для використання мережевої версії потрібні додаткові стандартні програми.

Розробка програмного забезпечення реалізована на мові високого рівня Delphi.

Delphi - це комбінація кількох важливих технологій:

- Високопродуктивний компілятор машинного коду.
- Об'єктно-орієнтована модель компонентів.
- Візуальне проектування додатків з прототипів програмного забезпечення.
- Масштабовані будівельні інструменти.

Компілятор, вбудований у Delphi, забезпечує високу продуктивність, необхідну для створення клієнт-серверних програм. Він пропонує простоту розробки та швидке тестування готового програмного блоку, типового для мов четвертого покоління (4GL), при цьому забезпечує якість коду, типову для компілятора 3GL.

Проектування в Delphi не дуже відрізняється від проектування в середовищі інтерпретатора, але після компіляції ми отримуємо код, який виконується в 10-20

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

Додаток містить описи всіх процедур і функцій, що містяться в програмі.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на магістерську роботу, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки для охоронної системи.

В процесі розробки кваліфікаційної магістерської 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;

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) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Інтеграція систем відеоспостереження та пожежної сигналізації може здійснюватися на одному або кількох рівнях роботи комбінованої системи ОПС. На малюнку 3.1 показана розроблена узагальнена діаграма, що показує можливі можливості такого інтегрування. Інтеграція на рівні отримання інформації про стан будівлі може бути реалізована шляхом використання багатофункціональних сповіщувачів – охоронних та пожежних, які реагують на виникнення симптомів пожежі та несанкціоноване проникнення. Такі детектори зазвичай використовують ультразвуковий або оптоелектронний принцип дії. З окремих сповіщувачів можна збирати модульні блоки різного призначення.



Рисунок 3.1 – Узагальнена схема інтеграції системи відеонагляду та пожежно-охоронної сигналізації.

одного з чотирьох захисних контурів (за 3,5 хвилини роботи сирени) сканер перемикає ланцюг захисту на зворотний сигнал, тобто якщо цей контур раніше спрацював розмиканням, тепер буде реагувати на зміну стану (коротке замикання). При необхідності схема виконує це перемикання до трьох разів при появі імпульсу на виході лічильника.

ІЧ-датчик - призначений для виявлення несанкціонованого проникнення в захищену зону.

Датчик температури - призначений для виявлення займання, що супроводжується наявністю низької концентрації диму в салоні

3.3 Розробка функціональної схеми

На рисунку 3.3 зображена функціональна схема системи. Вхідні дані це дані з датчиків.

- Д0-Д31 -Датчики, сповіщувачі.
- MUX – мультиплексом
- MCS51 – мікроконтролер сімейства 51.
- К - ключ-кнопка включення системи.

використовувані датчики:

- Сповіщувач інфрачервоний з релейним виходом (цифровим).

Кількість: 32 шт.

Як чутливий елемент використовується кабель зв'язку.

Використовувані мультиплексори:

два цифрових мультиплексора на шістнадцять інформаційних входів.

- використовуваний мікроконтролер - Мікроконтролер сімейства МК51.
- використовуваний РКІ - Два семисегментних індикатора.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

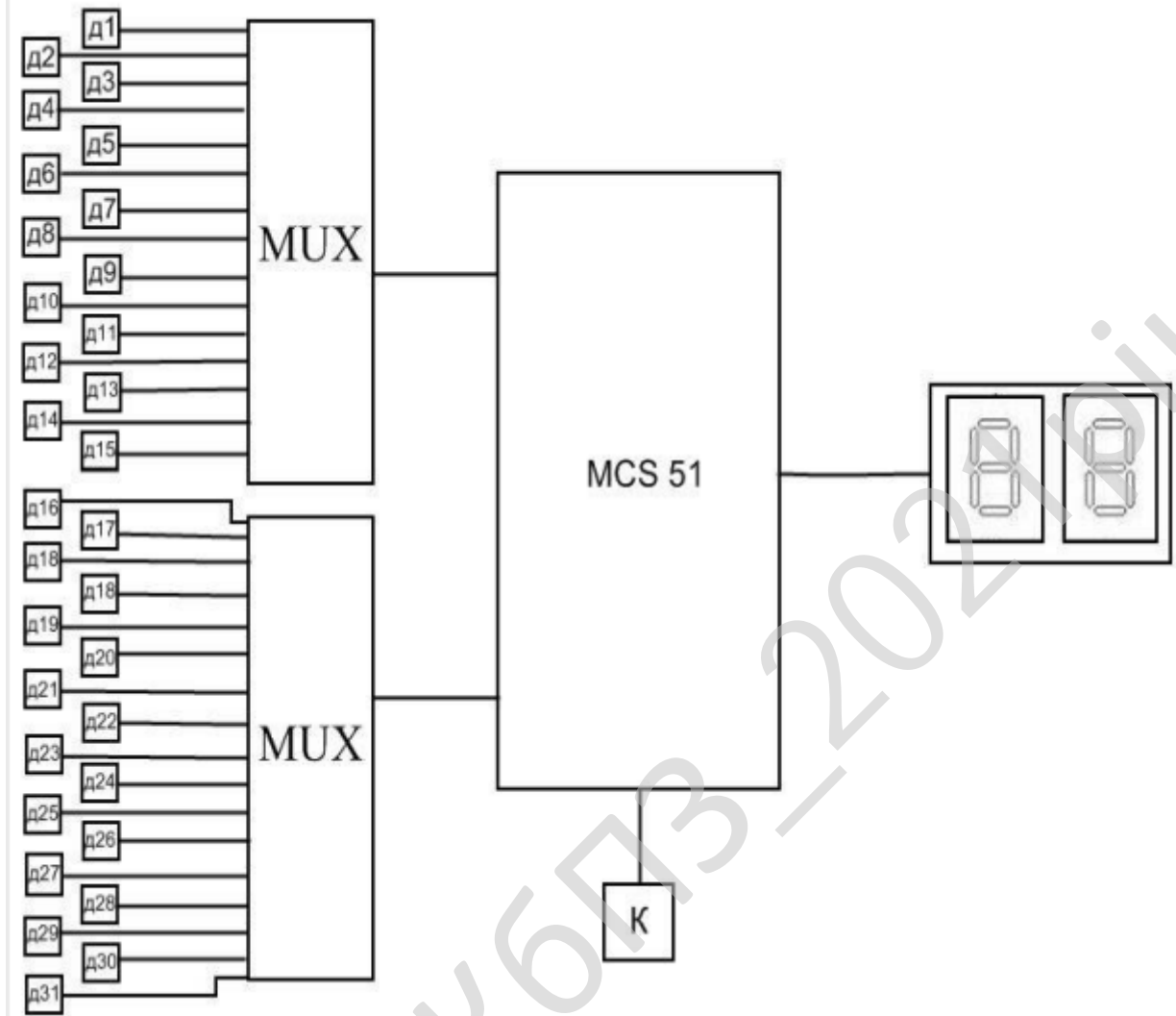


Рисунок 3.2 – Функціональна схема роботи системи

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3.

Після початку роботи над розробленим програмним забезпеченням ми потрапляємо в програмний інтерфейс, де потрапляємо в налаштування програмного забезпечення, бібліотеку швидкого сканування відеопотоку і через підключення камер і датчиків отримуємо зображення для упаковки вхідного зображення. матриця. Крім того, ми можемо дійти до зменшення деталей зображення та матриці авторизованого коду доступу. Від матриці авторизованого

коду доступу до налаштування матриці авторизованого коду доступу і далі від матриці до матриці порівняння з матрицею коду та остаточної авторизації доступу.

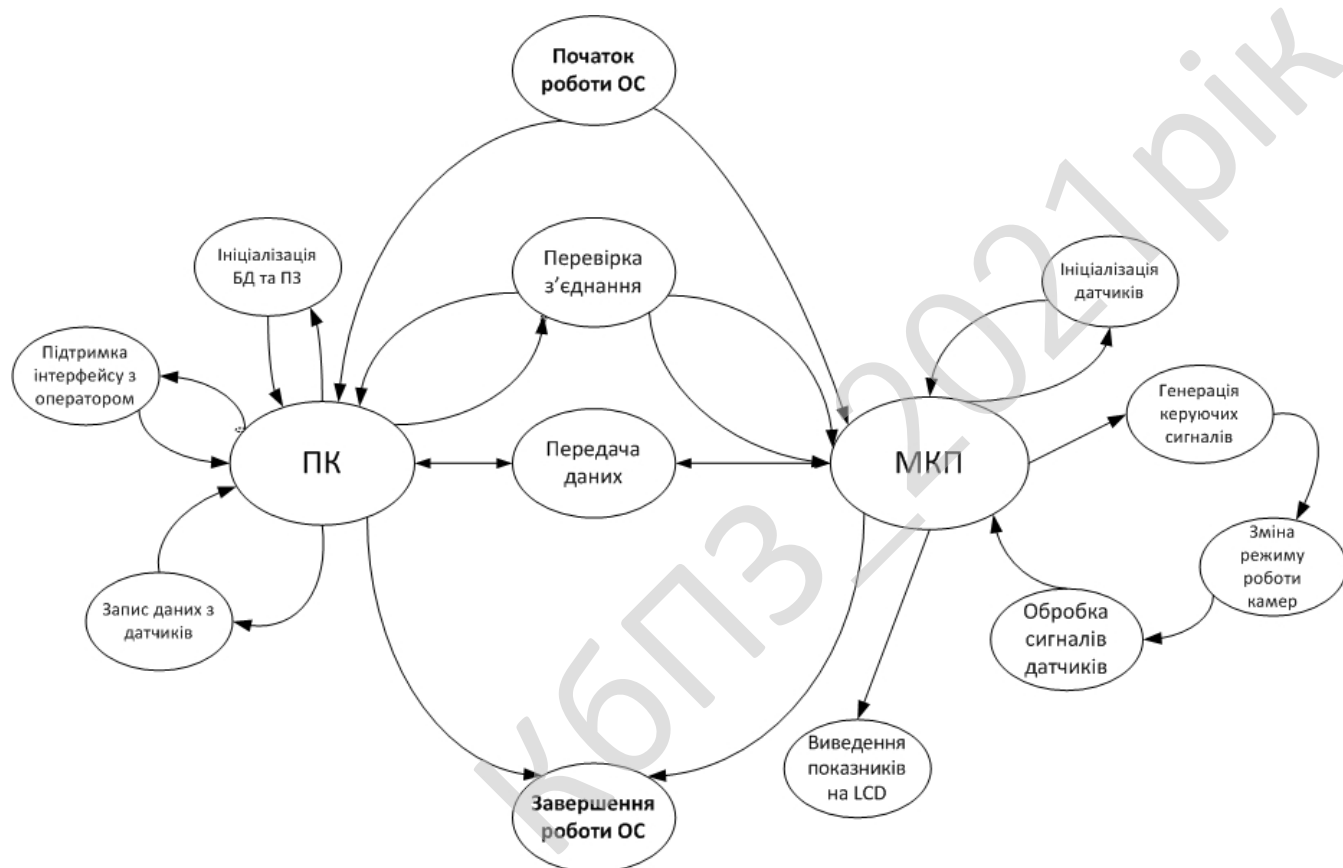


Рисунок 3.3 – Діаграма взаємодії процесів

Після розгляду опису системи, структурної, функціональної схеми системи та схеми взаємодії процесів переходимо до опису блок-схем основної програми та підпрограм, які використовуються для реалізації системи.

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

Розробка блок-формувача часових інтервалів

Блок часового інтервалу (рис. 4.1) складається з тригера на елементах DD1.1. DD1.3; генератор імпульсів DD3.1, DD3.2; лічильник імпульсів DD5 селектор часового інтервалу (12 і 6 с) на логічних елементах мікросхем DD6, DD3, DD7; обмежувач часу звукового сигналу на лічильнику DD2; тригер на елементах DD4 для забезпечення режиму очікування на початок першого інтервалу часу (12 с). Мікросхема 561LE5 вибирається в якості тригера, а мікросхема 561LE5 вибирається як генератор імпульсів D3.1, D3.2.

У момент подачі ланцюга імпульс, що формується схемою С3-Р3, забезпечує початкове встановлення нуля лічильників DD2 і DD5 (на виході DD2/7 буде логічна «1», тобто напруга живлення). У цьому випадку контакти мікросхеми будуть: DD4 / 3 - «1»; DD5 / 11 - «1»; DD1 / 1 - «1»; DD1 / 2 - «1» DD1 / 3 - «0»; DD6 / 10 - «1»; DD7 / 9 - «0». В якості лічильників виберемо мікросхеми 561PE11 і 561PE16.

За датчиком F1 (логіка «0» на входах DD4 / 13 і DD1 / 9) з'являється логіка «1» на виході DD4 / 11 (на DD4 / 10 — «0», що дозволяє спрацьовувати лічильник DD5). Генератор (імпульси на DD3/3 з частотою приблизно 500 Гц) і відповідний йому лічильник DD5 до часу (12 с) до появи логічного нуля на DD6/10 (логіка «1» на DD1/3 - що зупиняє генератор). Схема переходить в режим ЗАХИСТ. Якщо одночасно працює датчик F1 - тригер включає елементи DD1.1. DD1.3 (в кінці з'являється DD1/4 "1", на DD1/3 - "0"), що дозволяє генератору DD5 працювати.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

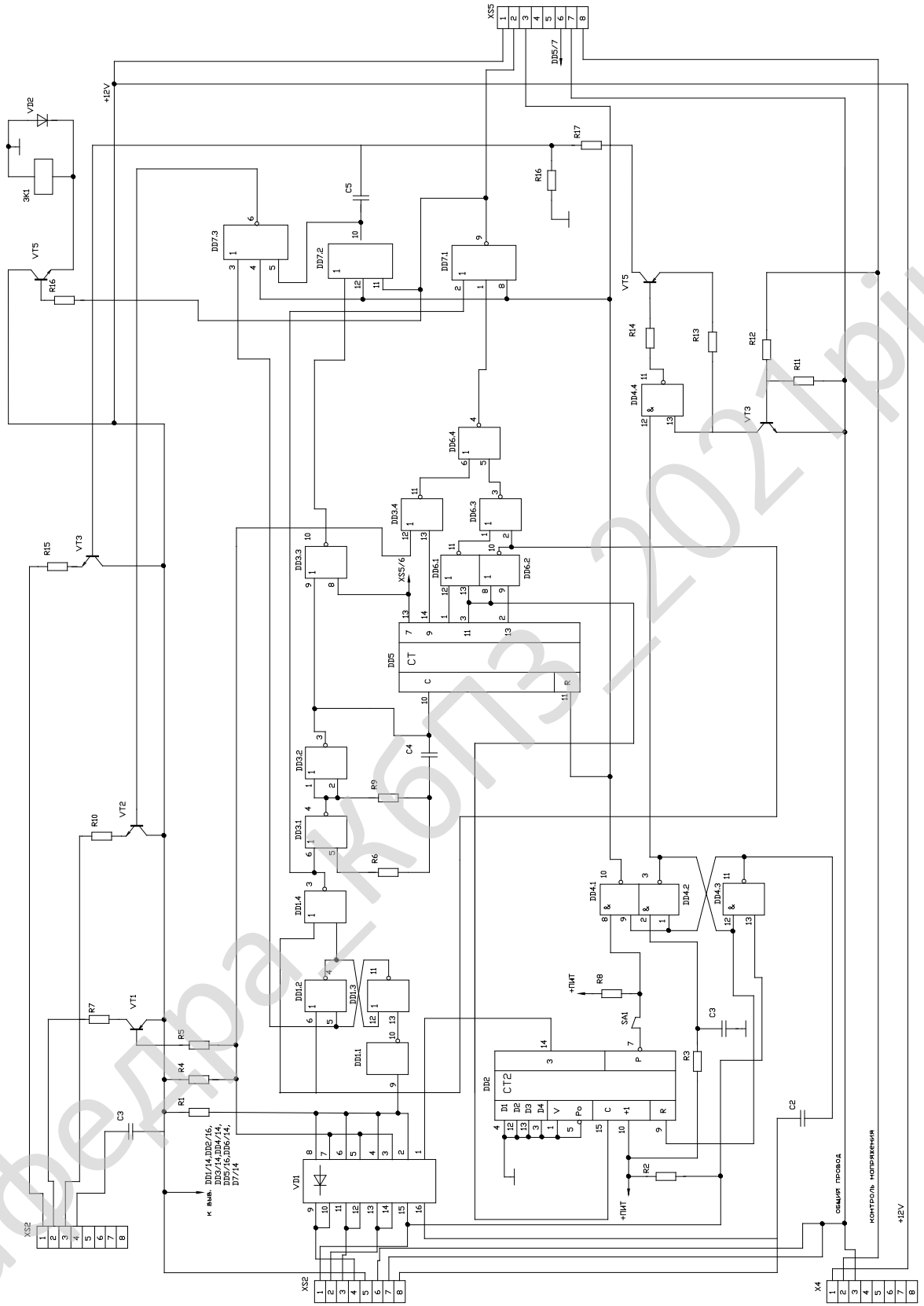


Рисунок 4.1 - Формувач часових інтервалів

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

ВКРМ-123.21.0033.00.00.ПЗ

Арк.

50

При спрацьовуванні будь-якого іншого датчика тригер на елементах DD1.1... DD1.3 також перемикається, але сигнал тривоги з'явиться без затримки і буде перервано.

Лічильник DD2 дозволяє обмежити час роботи звукового оповіщення. Коли на DD2 / 7 з'являється логічний «0» (з активованим SA1) і логічний «1» на DD4 / 10, цей рівень відключає роботу DD5 і передачу сигналів на вихід DD7 / 9.

Рівень заряду акумулятора контролюється транзистором VT3. Працює в режимі мікроструму, завдяки чому має великий коефіцієнт посилення і перемикається із замкнутого на розімкнений при зміні напруги в ланцюзі управління на 0,1В. Вибравши резистор R11, необхідно переконатися, що транзистор VT3 закритий при напрузі джерела G1 9 В або менше (логіка «1» на вході DD4 / 6). Зелений світлодіод буде горіти постійно - це означає, що причину падіння напруги необхідно усунути. Світлодіод гасне, коли ланцюг переходить в режим ЗАХИСТ (DD4 / 5 - логічний "0") - це виключає розряд батареї через струм, що проходить через світлодіод. Подвійний світлодіод HL1 можна замінити будь-якими двома звичайними, але з іншим кольором світіння. Зелений світлодіод також використовується для індикації режимів роботи - миготіння). В даному випадку, щоб зменшити споживання струму ланцюга, напруга подається на неї короткими імпульсами з виходу DD7 / 10. Через інерційність зору воно невидиме.

Налаштування блоку часового інтервалу починається з установки порога блокування транзистора VT3 на напругу 9В в ланцюзі акумулятора резистором R12 (напруга рухається від регульованого джерела живлення). Потім логіку схеми перевіряють, як описано. При необхідності можна налаштувати частоту опору R9 тактового генератора для отримання інтервалів часу 6 і 12 або 8 і 16 секунд.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Розробка сканера охоронних датчиків

Датчик безпеки сканер (рисунок 4.2), зібраний на двох мікросхемах. Чотирирозрядний послідовно-паралельний регістр 2DD1 використовується для зберігання початкового стану датчиків безпеки.

Запис в регістр здійснюється з появою імпульсу на вході 2DD1 / 6 - це вперше відбувається при запуску датчика F1 (перемикається тригер на DD4.2, DD4.3). Управління регістру 2DD1 вибрано таким чином, щоб на його виходах був зворотний до входів сигнал (у вихідному стані на висновках 13,15,14 і 1 буде логіка «1»). Логічні елементи 2DD2 забезпечують виходи «1». Це еквівалентно підключенню до ємнісних нормально відкритих датчиків HSZ.

Наявність виходів з кожного регістра тригера дає можливість перетворити послідовний код на вході D в паралельний, який видається з виходів Q0 -Q3. В одному випадку IP типу IP2 можна організувати восьмирозрядний регістр з послідовним введенням інформації та паралельним зчитуванням. Для цього просто встановіть перемички між контактами 6 і 14, 1 і 9, 10 і 15.

В якості датчиків, встановлених на дверях, вікнах та інших місцях, можуть використовуватися як загальноприйняті, промислового виробництва (СМК-1, ДМК) при відкриванні, так і будь-які інші (ультразвукові, ємнісні, інфрачервоні та ін.) з релейним виходом під час роботи. Багато датчиків, замкнених по колу, можна підключити до однієї запобіжної петлі, тому відкриття кожного з них призведе до розмикання ланцюга.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

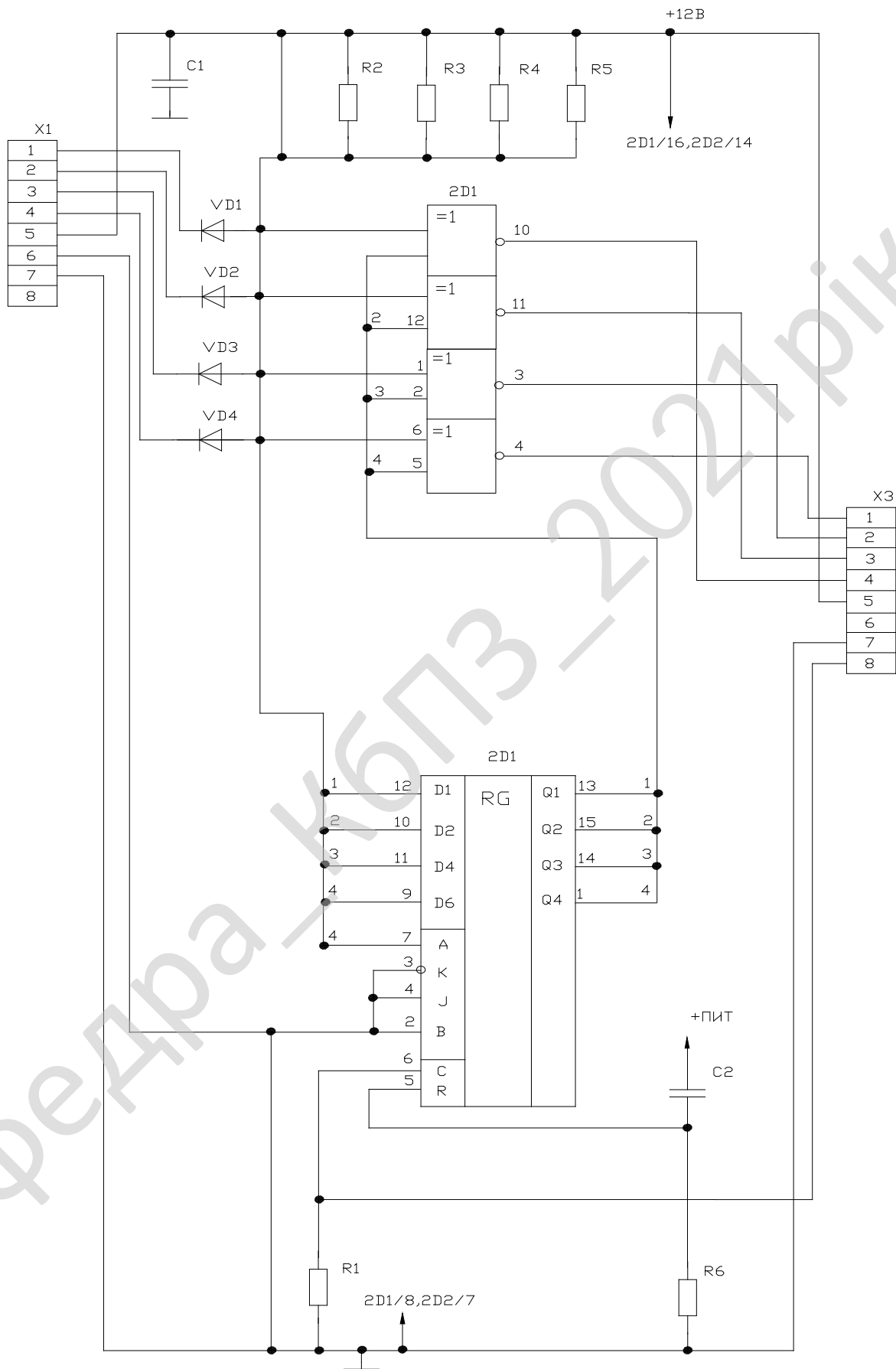


Рисунок 4.2 - Сканер охоронних датчиків

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

ВКРМ-123.21.0033.00.00.ПЗ

Арк.

53

буде перервано, оскільки до входу DD3 / 12 і DD3 / 11 застосовується логіка «0».
будуть імпульси.

Лічильник DD2 дозволяє обмежити час роботи звукового оповіщення. Коли на DD2 / 7 з'являється логічний «0» (з активованим SA1) і логічний «1» на DD4 / 10, цей рівень відключає роботу DD5 і передачу сигналів на вихід DD7 / 9.

Напруга акумулятора контролюється транзистором VT3. Працює в режимі мікроструму, завдяки чому має великий коефіцієнт посилення і перемикається із замкнутого на розімкнений при зміні напруги в ланцюзі управління на 0,1В. Вибравши резистор R11, необхідно переконатися, що при напрузі живлення G1 9В або менше транзистор VT3 закритий (логіка «1» на вході DD4/6). Зелений світлодіод буде горіти постійно - це означає, що причину падіння напруги необхідно усунути. Світлодіод гасне, коли ланцюг переходить в режим ЗАХИСТ (DD4 / 5 - логічний "0") - це виключає розряд батареї через струм, що проходить через світлодіод. Подвійний світлодіод HL1 можна замінити будь-якими двома звичайними, але з іншим кольором світіння. Зелений світлодіод також використовується для індикації блимання режимів). В даному випадку, щоб зменшити споживання струму ланцюга, напруга подається на неї короткими імпульсами з виходу DD7 / 10. Через інерційність зору воно невидиме.

Запис в регістр здійснюється з появою імпульсу на вході 2DD1 / 6 - це вперше відбувається при запуску датчика F1 (перемикається тригер на DD4.2, DD4.3). Управління регістром 2DD1 вибрано таким чином, щоб на його виходах був сигнал, зворотний до входу (у вихідному стані на висновках 13,15,14 і 1 буде логічна «1») датчиків.

Наявність виходів з кожного регістра тригера дає можливість перетворити послідовний код на вході D в паралельний, який видається з виходів Q0 -Q3. В одному випадку IP типу IP2 можна організувати восьмирозрядний регістр з послідовним введенням інформації та паралельним зчитуванням. Для цього просто встановіть перемички між контактами 6 і 14, 1 і 9, 10 і 15.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

В якості датчиків, що встановлюються на дверях, вікнах та інших місцях, можуть використовуватися як загальноприйняті, промислового виробництва (СМК-1, ДИМК) при відкриванні, так і будь-які інші (ультразвукові, ємнісні, інфрачервоні та ін.) з релейним виходом під час роботи. Багато датчиків із замкнутим контуром можуть бути підключені до однієї петлі безпеки, щоб ланцюг розривався, коли кожен відкривається.

Зчитувач ключів

Зчитувач ключів із сенсорною пам'яттю використовується для встановлення / видалення системи безпеки, розробленої в режимі безпеки. Touch Memory (буквально контактує з пам'яттю) — електронний ключ-ідентифікатор («планшет», брелок у побуті), носій даних для автоматичної ідентифікації унікального коду (рисунок 3.6).

Ключ Touch Memory являє собою мікросхему, розміщену в міцному герметичному корпусі з нержавіючої сталі, що забезпечує підвищену стійкість до впливу навколишнього середовища. Верхня кришка електрично ізольована від решти циліндра поліпропіленовим корпусом і є контактом для передачі даних. Нижня частина циліндра має фланець для полегшення складання і є загальним контактом.

Клавіша Touch Memory в парі ключ-драйвер є пасивним елементом. Це означає, що ключ не має внутрішнього живлення.

Кнопки сенсорної пам'яті не мають двох однакових продуктів (64-розрядний реєстраційний номер).

Touch Memory виробляється Dallas Semiconductor, США (зараз Maxim Integrated Products).

Сенсорна пам'ять Dallas DS1990A

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56



Рисунок 4.3 - Ключ Touch Memory

Як зчитувач ключів використовується стандартна кнопка запиту даних з вбудованим світлодіодом КС-05 (рисунок 3.7).

Технічні характеристики КС-05

Призначення	системи контролю доступом
Матеріал	метал
Виконання	накладне
Колір	чорний
Контакти	НР
Розміри	Ш38x10 мм



Рисунок 4.4 - Зчитувач ключів КС-05

Вибір датчиків

У системі розробки використовуються лише дискретні датчики: інфрачервоні датчики руху, датчик пожежного диму та верхній магнітний контактний датчик для входних дверей.

Інфрачервоний датчик руху Pyronix Colt XS:

Сповіщувачі Pyronix Colt XS призначені для виявлення несанкціонованого проникнення в захищені зони.

Використовується для установки в охоронних і протипожежних системах [OPS].

Запатентовані технології, використані при розробці та створенні сповіщувача Pyronix Colt XS, значно підвищили завадостійкість: досягти оптимального співвідношення чутливості до стійкості до некоректної роботи.

Крім того, слід зазначити, що дальність дії Pyronix Colt XS становить 15 метрів.

Функціональні параметри Pyronix Colt XS:

- Екрановане реле
- Герметична оптика
- Варіанти лінз: №1, №2, №3, №4
- Фільтр білого світла
- Завадостійкість до радіовипромінювання
- Світлодіодна індикація спрацьовування
- Перемичка відключення індикації
- Низьке енергоспоживання
- температурна компенсація, що попереджає, [АТС]
- Спрацьовування по імпульсу будь-якої полярності
- Аналоговий підрахунок імпульсів [1-й, 2-й або 3-й імпульс]
- Антистатичний корпус із товщиною стінки 3 мм.
- Фіксація лицьової кришки гвинтом
- Регулювання дальності дії від 5 до 15 м.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

- Технологічний паз для укладання сполучних проводів
- Поворотний кронштейн Pyronix EB1 у комплекті
- 5 років заводської гарантії

Технічні параметри Pyronix Colt XS:

- Тип: ІЧ пасивний
- Монтаж: Настінний
- Максимальна дальність дії: 15 м
- Час видачі тривожного повідомлення: >1 сек
- Навантаження на контакти реле: до 100 В, до 75 мА
- Навантаження на вихід самоохорони: до 12 В, до 50 мА
- Електроживлення: 9 - 16В DC
- Енергоспоживання в черговому режимі: 15 мА
- Енергоспоживання в режимі спрацьовування: 15 мА
- Розміри : 52x61x40 мм
- Температурний діапазон: -30 ... +70 °С
- Вага: 72 г
- Колір приладу: Білий
- Матеріал корпусу: Пластик
- Обробка корпусу: Полірування.

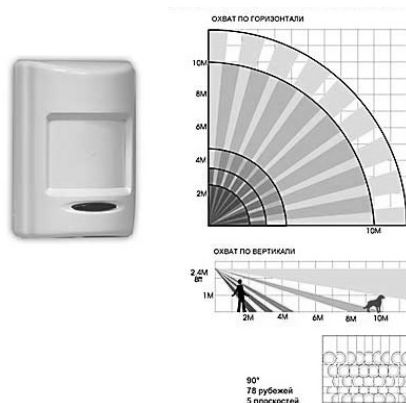


Рисунок 4.5 - Зовнішній вигляд датчика та схема його зони виявлення

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Сповіс­тувач пожежний оптико-електронний димовий ИП 212-41М

Датчик призначений для виявлення пожеж, які супроводжуються низькою концентрацією диму в салоні. Сповіс­тувач IP 212-41 (ДИП-41) працює з пристроями, що приймають і контролюють російського і зарубіжного виробництва з постійною або змінною напругою в контурі від 7,5 до 30В. Сповіс­тувач формує сигнал «ПОЖАЖЯ», зменшуючи внутрішній опір не більше ніж до 500 Ом. Скидання зі стану «ПОЖЕМ» здійснюється шляхом відключення напруги живлення не менше ніж на 1,5 с. Сповіс­тувач IP 212-41 (DIP-41) має вбудований оптичний індикатор активності та забезпечує можливість підключення пристрою дистанційної системи оптичної безпеки (ВУОС).

Функціональні параметри ИП 212-41М

- площа, що захищається - до 100 кв.м.
- світлодіодний індикатор стану; горить постійно в режимі тривоги.
- Чутливість (поріг) сповіс­тувача відповідає концентрації диму, що забезпечує ослаблення світлового потоку в межах 0,05...0,2 дБ/м.
- Інерція сповіс­тувача не перевищує 5 с.
- Сигнал FAULT генерується при відключенні сповіс­тувача від розетки приймача режиму та контуру керування.
- Можна підключити дистанційну оптичну сигналізацію (VUOS).
- Розширений температурний діапазон від -25 °С до +55 °С дозволяє встановлювати ці пожежні сповіс­тувачі в неопалюваних приміщеннях.
- Ступінь захисту IP30.
- При дотриманні правил технічного обслуговування середній термін служби становить не менше 10 років.
- Середній час роботи в разі відмови з урахуванням технічного обслуговування - 60 000 годин.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Основні параметри

- Установка: стеля
- Живлення: 9 .. 30 В постійного струму
- Споживання електроенергії в режимі очікування: 50 мкА
- Споживання електроенергії в робочому режимі: <30 мА
- Індикація тривоги: червоний світлодіод
- Розміри: Ø 105 X 60 мм
- Температурний діапазон: -25...+55 °С
- Вага: 210 г

Колір обладнання: білий

автономні сповіщуючі поєдні димові оптико-електронні



Рисунок 4.6 - Зовнішній вигляд ИП 212-41М

Вибір мікроконтролера для центрального блоку

Мікроконтролер - мікросхема, призначена для управління електронними пристроями. Ці чіпи є основою для побудови вбудованих систем, їх можна знайти в багатьох сучасних пристроях, таких як телефони, пральні машини тощо. На сьогоднішній день існує понад 200 модифікацій мікроконтролерів. Популярністю серед розробників користуються 8-розрядні мікроконтролери PIC від Microchip Technology і AVR від Atmel, 16-розрядні MSP430 від TI і ARM. У той час як 8 - розрядні універсальні процесори були повністю замінені більш сучасними

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

моделями, 8-розрядні мікроконтролери продовжують широко використовуватися. Існує велика кількість додатків, які не вимагають високої продуктивності.

Пристрій розробки не буде потребувати високопродуктивного мікроконтролера (наприклад, 16-розрядного MSP430), оскільки кількість завдань у системі оповіщення не буде великою.

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо алгоритм роботи основної програми. Його блок-схема зображена на рисунку 4.7.

З рисунку видно, що після запуску програми спочатку відбувається вивід основного вікна програми. Потім запуск алгоритму програмного продукту

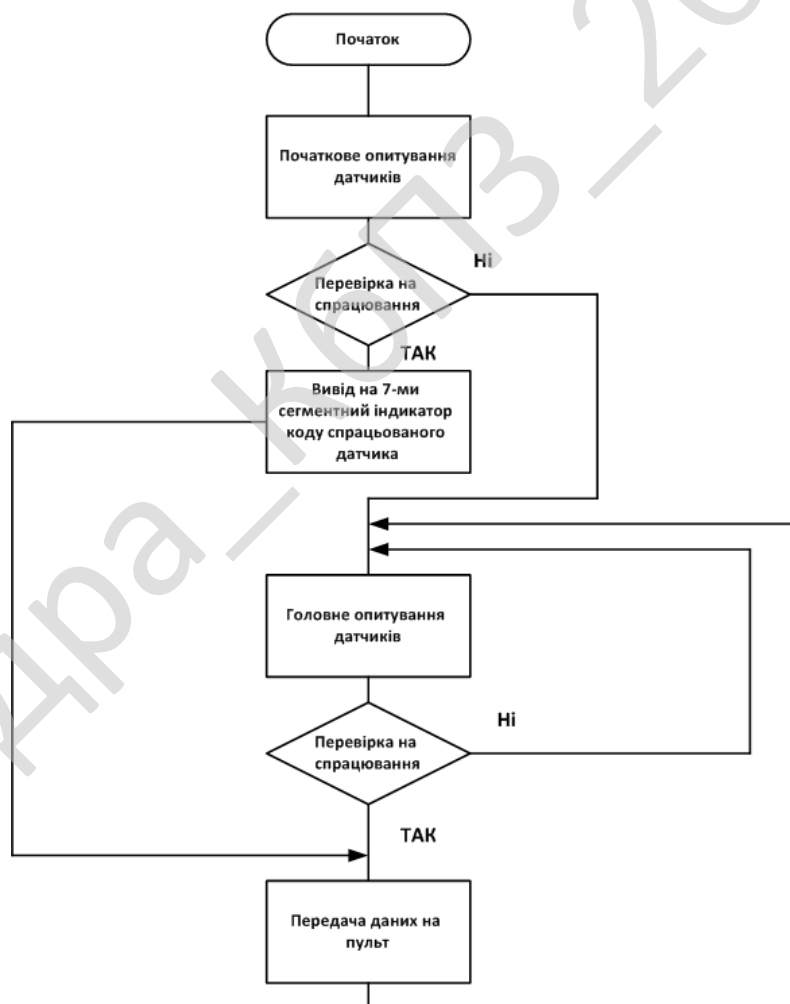


Рисунок 4.7 – Блок-схема основної програми

На малюнку 4.8 представлена блок-схема підпрограми основного пункту спостереження системи безпеки. Її робота складається з наступних етапів:

- введення даних авторизації
- перевірка полів введення
- Перевірте дані, наявні в базі даних
- Запис інформації в базу даних
- Помилка доступу до виведення. Скасування реєстрації в системі
- Перехід до розділу профілю ACS
- Робота з ОС ACS
- Вимкнення ОС.

					VKPM-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

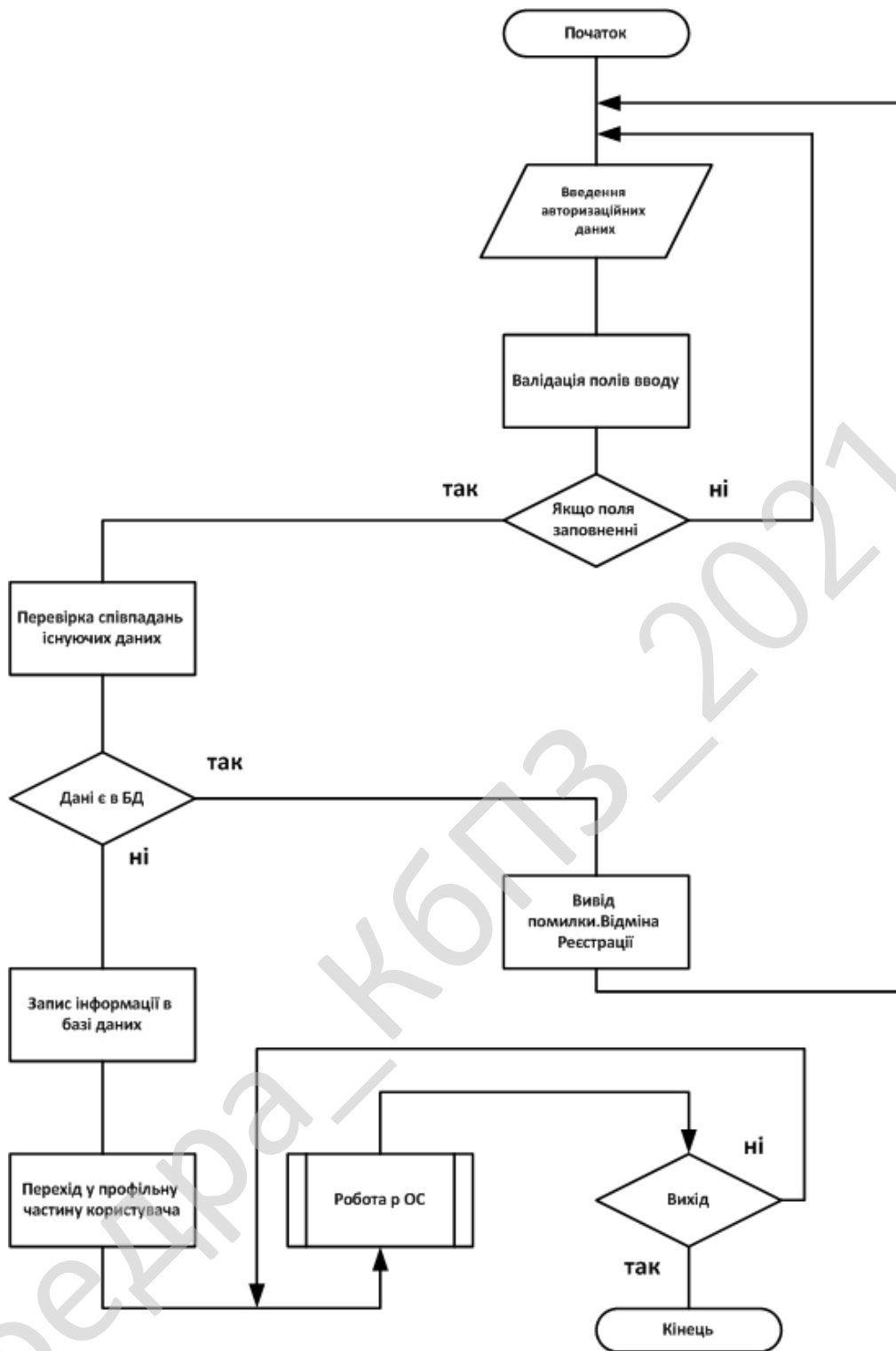


Рисунок 4.2 – Блок-схема підпрограми доступу оператора до АСУ роботи охоронної системи

4.2 Захист розробленого програмного забезпечення

Дані в програмі захищені алгоритмом Md5. Він отримує повідомлення будь-якої довжини на вході і створює 128-бітний підсумок повідомлення на виході. Алгоритм складається з наступних кроків:

1. Додайте пропущені біти. Повідомлення доповнюється так, що його довжина дорівнює 448 за модулем 512 (довжина 448 за модулем 512). Це означає, що довжина доданого повідомлення на 64 біти менша, ніж кратна 512. Додавання виконується завжди, навіть якщо повідомлення має необхідну довжину. Наприклад, якщо довжина повідомлення становить 448 біт, воно доповнюється 512 бітами до 960 біт. Таким чином, кількість доданих бітів коливається від 1 до 512.

Додавання складається з одиниці, за якою слідує необхідна кількість нулів.

2. Додайте довжину. До результату першого кроку додається 64-бітове представлення довжини початкового (перед додаванням) повідомлення в бітах. Якщо початкова довжина перевищує 264, використовуються тільки останні 64 біти. Таким чином, поле містить довжину початкового повідомлення за модулем 264.

Перші два кроки створюють повідомлення, кратне 512 бітам. Це розширене повідомлення представлено у вигляді послідовності 512-бітових блоків Y_0, Y_1, \dots, Y_{l-1} із загальною довжиною розширеного повідомлення, що дорівнює $L * 512$ біт. Таким чином, довжина отриманого розширеного повідомлення кратна шістнадцяти 32-бітовим словам.

3. Ініціалізація буфера MD. Алгоритм Md5 використовує 128-бітовий кеш для зберігання проміжних продуктів і кінцевих результатів хеш-функції. Буфер може бути представлений у вигляді чотирьох 32-розрядних регістрів (A, B, C, D). Ці регістри були ініціалізовані такими шістнадцятковими числами:

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

A = 01234567

B = 89abcdef

C = Fedcba98

D = 76543210

4. Послідовна обробка 512-розрядних (16-слівних) блоків. Основою алгоритму Md5 є модуль, що складається з чотирьох циклічних налаштувань, які називаються Hmd5. Чотири цикли мають схожу структуру, але кожен цикл використовує свою елементарну логічну функцію, позначену ff, fg, fh і fi.

У кожному циклі поточні входи щойно обробленого 512-бітового блоку Y q і 128-бітове значення буфера ABCD, яке є середнім поділом, і змінюють вміст цього буфера. Кожен цикл також використовує четверту частину 64-елементної таблиці T [1. 64], на основі функції sin. Позначений i-й елемент T, T [i] має значення, рівне всій частині $232 * \text{abs}(\sin(i))$ і наводиться в радіанах. Оскільки $\text{abs}(\sin(i))$ є числом від 0 до 1, кожен елемент T є цілим числом, яке може бути представлено 32 бітами. У таблиці наведено «випадковий» набір 32-бітових значень, які повинні усунути будь-яку закономірність у вхідних даних.

Щоб отримати Mdq +1, вихід чотирьох циклів дорівнює 232 з Mdq. Додавання виконується незалежно для кожного з чотирьох слів у кеші.

5) Вихід Md5. Після того, як усі L 512-бітові блоки були оброблені, вихід L-ї фази є 128-бітним підсумком повідомлення.

Логіка кожного з чотирьох циклів виконання одного 512-бітового блоку більш детально обговорюється нижче. Кожен цикл складається з 16 кроків, які працюють з буфером ABCD.

$AB + C_{lss}(A + f(B, C, D) + X[k] + T[i]),$

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

На кожному кроці змінюється лише одне з чотирьох слів кешу ABCD. Таким чином, кожне слово кешу змінюється 16 разів, а потім 17 разів у кінці, щоб отримати остаточний результат цього блоку.

Кафедра КБПЗ – 2021 рік

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Введення в експлуатацію системи захисту означає виконання заходів із захисту інформації, передбачених технічним проектом на ці роботи, встановлення обладнання для обробки інформації, технічних засобів, а також заходів технічного захисту інформації мають залучати ліцензовані організації.

Монтажна організація або замовник здійснює закупівлю сертифікованих TSOI і спеціальну перевірку несертифікованих TSOI для виявлення будь-яких вбудованих електронних пристроїв збирання інформації ("закладок") та їх спеціальне дослідження.

За результатами спеціальних досліджень TSOI уточнюються заходи захисту інформації. При необхідності будуть внесені відповідні зміни в технічний проект, які будуть узгоджені з проектною організацією та замовником.

Виконується закупівля сертифікованих технічних, програмних та програмно-технічних засобів захисту інформації та їх встановлення відповідно до технічного проекту.

Служба безпеки (спеціаліст) організовує контроль заходів щодо захисту інформації, наданої технічним проектом.

У період встановлення та монтажу TSOI та засобів захисту інформації особливу увагу необхідно приділяти забезпеченню режиму об'єкта, що охороняється.

Деякі заходи щодо організації контролю в цей період включають:

- перед монтажем необхідно провести негласний огляд всіх змонтованих конструкцій, особливо монтажного обладнання, на наявність різноманітних міток та їх взаємних відмінностей, а також заставного обладнання.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

- виникає необхідність організації регулярних оглядів приміщень у вечірній час за відсутності будівельників з метою виявлення підозрілих ділянок та місць;

- організувати контроль за ходом усіх видів будівельних робіт біля будинку та в будинку. Основна функція перевірки залежить від підтвердження правильності технології будівництва та монтажу та відповідності їх технічного проекту;

- організувати огляд міст і будівель, які за технологією закриті іншими будівлями. Такі перевірки можуть бути організовані законно під легендою про необхідність контролю якості монтажу та матеріалів або таємно;

- повинні бути ретельно перевірені схеми підключення та кількість прокладених згідно з технічним проектом провідників. Особливу увагу слід приділити етапу введення кабельного зв'язку та кабелів у зону забороненого використання. Усі запасні проводи та кабелі, що зберігаються, повинні бути прокладені на плані з початковою та кінцевою точками.

Перед встановленням у відведених приміщеннях та об'єктах інформатизації меблів та предметів внутрішнього технічного обладнання та засобів оргтехніки перевіряють відсутність заставного обладнання. При цьому доцільно перевіряти технічні засоби на рівень перешкоджаючого електромагнітного випромінювання. Такий огляд підійде в спеціально обладнаному приміщенні або проміжному сховищі.

Після встановлення та встановлення технічних засобів захисту проводиться їх дослідна експлуатація в комплексі з іншими технічними та програмними засобами з метою контролю їх працездатності в рамках предмета інформатизації та детальної опрацювання технологічного процесу обробки. (передача) інформації.

За результатами дослідної експлуатації проводяться приймальні випробування засобів захисту з реєстрацією відповідного закону.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Після введення ЗЗІ в експлуатацію проводиться паспортизація інформаційних об'єктів та заповідних територій відповідно до вимог безпеки. це процедура офіційного підтвердження ефективності впровадженого на об'єкті комплексу заходів та засобів захисту інформації.

У разі потреби за рішенням керівника організації може бути прийнято рішення про пошук засобів електронного запису інформації («заставних об'єктів»), можливо, що здійснюється на заповідних територіях, що здійснюється організаціями, які мають відповідні ліцензії СБУ України. .

Протягом операційного циклу необхідно періодично проводити спеціальні перевірки обраних територій та інформаційних об'єктів. Спеціальні опитування слід проводити під заголовком персоналу або відсутності (допускається присутність обмеженої кількості керівників і працівників охорони).

При включенні живлення або натисканні кнопки Reset пристрій перемикається в режим, в якому перебував востаннє – «очікування включення» або «постановка на охорону». Однак, якщо телефон не підключений або з ним є якісь проблеми, світлодіод буде дуже рідко блимати (приблизно 0,5 с, ~ 5 с паузи), доки телефон не буде підключено або проблема не буде вирішена.

У режимі «очікування охорони» пристрій чекає, поки зчитувач підбере ключ. Якщо ключ піднятий, він знаходиться в пам'яті, всі зони закриті, світлодіод блимає тричі і пристрій переходить в режим «захист». Якщо хоча б одна із зон не закрита, пристрій тричі блимне короткими спалахами і не перейде в режим «захист».

Закрита зона - це зона, яка замикається на землю. Якщо вам не потрібні всі чотири зони, інші слід закрити. Якщо благородного ключа немає в пам'яті, світлодіод також блимне тричі.

У режимі «захист» час від часу блимає світлодіод, перевіряються всі зони на обрив і наявність зовнішнього джерела живлення. При порушенні кожної із зон на номери, збережені в пам'яті мобільного телефону, надсилаються SMS-номери та короткі дзвінки. Потім світлодіод починає швидко блимати, і пристрій чекає

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

закінчення цього режиму, торкаючись клавіші на зчитувачі, який знаходиться в пам'яті. Якщо зовнішнє живлення вимкнено в цьому режимі, на всі номери одноразово прийде SMS з текстом «Отключено внешнее питание» і пристрій продовжить працювати від акумулятора мобільного телефону, як уже було сказано. Вихід з цього режиму та перемикання в режим «очікування охорони» відбувається, коли ви підбираєте ключ, який знаходиться в пам'яті.

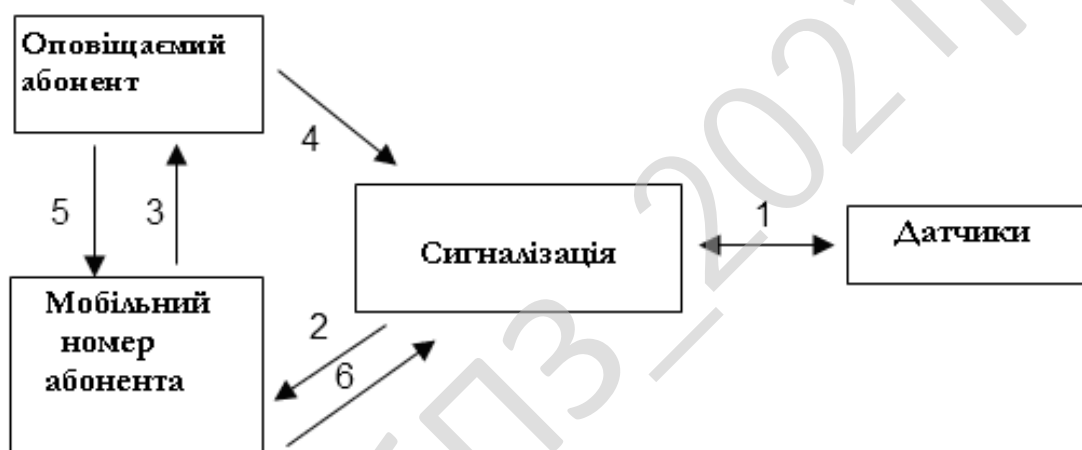


Рисунок 5.1 - Робота блоку сигналізації

Режим програмування та налаштувань телефону. У цьому режимі клавіші запрограмовані в пам'ять, і телефон налаштований на роботу. Для початку вам слід видалити всі SMS та номери телефонів, які є в пам'яті телефону та на SIM-картці, з якою телефон буде працювати. Потім вам потрібно створити на ноутбучі від одного до трьох телефонів, які в майбутньому отримуватимуть вхідні дзвінки та SMS. На цьому робота телефону завершена.

Потім необхідно увійти в режим програмування пристрою. Для цього натисніть і утримуйте кнопку But1, натисніть і відпустіть кнопку Reset. Натисніть клавішу But1. Світлодіод на зчитувачі починає блимати: блимає один раз, потім зупиняється тощо. Це вказує на вихід у перший режим програмування.

А п'яте SMS з текстом «Отключено внешнее питание» буде надіслано, якщо в режимі захисту відпаде живлення 12В. Пристрій деякий час зможе працювати від акумулятора телефону.

Четвертий режим програмування був останнім режимом програмування. Щоб вийти з будь-якого режиму програмування, натисніть кнопку Reset.

Кафедра КБПЗ – 2021 рік

					VKPM-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

6 НАУКОВА НОВИЗНА

У магістерській роботі розроблено програмне забезпечення, яке призначено для системи кібербезпеки охоронної системи.

Метою розробки є дослідження та програмна реалізація охоронної системи з впровадженням системи кібербезпеки для захисту від взлому.

Об'єктом дослідження є процес управління охоронною системою з функцією кібербезпеки.

Предметом дослідження є методи реалізації систем кібербезпеки та систем віддаленого захисту об'єкта.

Методи дослідження базуються на методах математичної статистики, методах захисту інформації, методах розробки програмного забезпечення.

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

- Удосконалено охоронну систему віддаленого об'єкта;
- Проведено огляд технологій кіберзахисту для об'єктів такого типу;
- Розроблено вітчизняний продукт системи охорони віддаленого об'єкта з впровадженням безпеки передачі даних, який має більш широкі можливості, на відміну від існуючих аналогів;
- Розроблена комп'ютерна технологія управління захистом віддаленого об'єкта.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми магістерської роботи

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація мікропроцесорної системи управління насосною станцією м. Мала Виска.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	900 (2 ост. цифри № зал·10 ²)
3. Запланований термін розробки, днів	Fp _q	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

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

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0033.00.00.ПЗ

Арк.

77

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	90000 (2 ост. цифри № зал*10 ⁴)
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

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

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

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

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

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

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

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027.$$

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

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

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

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

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

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

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

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

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 2,85 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 87 = 168 \text{ люд/день.}$$

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

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	168	Ф 7.1-7.4
Впровадження	13	Д13
Всього	209	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{nz} \cdot N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

T_{nz} – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{209 \cdot 1}{60 - 5} = 3,8 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

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

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

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

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

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

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

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

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

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

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

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

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру, Windows Server 2016 R2, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Продовження таблиці 7.4

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

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

ВКРМ-123.21.0033.00.00.ПЗ

Арк.

83

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	6225	18675
Продакт-менеджер	0,25	6000	4500
Інженер-програміст	3,8	6000	68400
Інженер-електронщик	1,2	6000	21600
Інженер-системотехнік	0,25	6000	4500
Адміністратор мережі	0,5	6000	9000
Системний програміст	0,25	6000	4500
Дизайнер WEB	0,25	6000	4500
Інженер-верстальник	0,25	6000	4500
Бухгалтер-економіст	0,5	6000	9000
Всього за період розробки	$R_{cn} = 8,25$	-	$\Phi_{роб} = 149175$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{cd} = \frac{149175}{8,25 \cdot 60} = 301 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

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

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

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

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

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

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

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800...1600 у.о./ m^2 . Враховуючи, що курс у.о. приймаємо 29 для розрахунку вартість одного метра квадратного становить 23200 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\delta} = 8 \cdot 8 \cdot 23200 = 1484800 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 148480 грн.

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

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

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

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

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

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались за прайсом Інтернет магазину Компбест за 17.11.20 – джерело <https://compbest.com.ua>.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11186
Системний блок		6488
Процесор	Intel Core i5-750 S-1156 (8M Cache, 2.66 GHz)	1636
Системна плата	MB Asrock H55M-LE s1156 (H55, s1156, DDR3 2600(OC)x, ntel(R) HD Graphics, 1xPCI-E 16x, 4xSATA2, Lan 1000 Mb/s, SB 7.1) mATX	1050
Відеокарта	VC Manli GeForce GT220 PCIe-2.0 512Mb DDR2 128 bit HDMI DVI RTL	860
Жорсткий диск	HDD 1000 Gb WD 7200 64Mb WD6400AARS SATAII	980
Оперативна пам'ять	DIMM 2048Mb DDR3 PC3-10600 Samsung 1333Mhz	468
DVD-привод	Super Multi LG SATA DVD±RW R+22x/-22x, RW+8x/-6x, DL+16x/-12x, RAM 12x, SecurDisc, black (GH22NS40RBB)	432
Корпус	ASUS TA-861 500W FSP ATX-500W (Black/Silver panel) ATX (90-PL861AF5C4-53CZ)	822
Кулер	—	—
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	240

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0033.00.00.ПЗ

Арк.

86

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000:1, 170/160, D-SUB, Wide)	3200
Принтер лазерний	CANON LBP-3010 Black	929
Принтер струминний	EPSON STYLUS PHOTO R390	1690
Сканер	Epson Perfection V200 Photo	968
Копіювальний апарат	Copier: CANON IR-1022A	5965
Пристрій безперебійного живлення	UPS APC BACK-UPS ES 525VA 230V RUSSIA (BE525-RS)	1498

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

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	11186	8948,8	98436,8
Принтер лаз.	2	929	185,8	2043,8
Принтер струм.	1	1690	169	1859

Продовження таблиці 7.7

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Сканери	1	968	96,8	1064,8
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	109966

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1484800	-	-
2. Передавальні пристрої	148480	-	-
Всього по групі	1633280	5	81664
Група 4			
3. Обчислювальна техніка	109966	-	-
Всього по групі	109966	50	54983
Нематеріальні активи			
4. Нематеріальні активи	90000	10	9000

Продовження таблиці 7.8

1	2	3	4
Група 5, 6			
5. Вимірювальні пристрої	5190	25	1297,5
6. Транспортні засоби	109200	20	21840
7. Господарський інвентар	28000	25	7000
Всього по групі	142390	-	30137,5
Разом	$K_p = 1975636$		$A_p = 175785$

Примітка: вартість автомобіля взята по даним з автосалону автотрейдинг, вкладки автобазар, джерело <http://www.auto-trading.com.ua/sale/lot20772.html>, складає 3900 USD, що враховуючи прийнятий для розрахунку курс 28 складає 109200 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

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

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 301 \cdot 209 / 900 = 70 \text{ грн.}$$

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

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

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 70 \cdot 10 \cdot 0,01 = 7 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 37\%$ від суми основної та додаткової зарплати:

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

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(70+7) = 17 \text{ грн.}$$

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

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

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 70 \cdot 15 \cdot 0,01 = 10,5 \text{ грн.}$$

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

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

де: Z_{M1} – вартість паперу, грн.;

Z_{M2} – вартість запам'ятовуючих пристроїв, грн.;

Z_{M3} – вартість фарби, картриджей, тонеру, грн.;

N_e – кількість екземплярів програм, шт.

Згідно виданих викладачем норм витрат паперу на місяць $n=2$ пачки та враховуючи, що вартість пачки паперу складає $Ц_n = 100$ грн., визначаємо вартість паперу за період розробки:

$$Z_{M1} = Ц_n \cdot N_n \cdot n. \quad (7.16)$$

$$Z_{M1} = 100 \cdot 3 \cdot 2 = 600 \text{ грн.}$$

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

$$Z_{M2} = \sum Ц_d, \quad (7.17)$$

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

$$Z_{M2} = 900 \cdot 12 = 10800 \text{ грн.}$$

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

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

$$Z_{M3} = \sum C_3, \quad (7.18)$$

де: C_3 – вартість розхідних матеріалів друкуючих пристроїв: картридж для CANON LBP-3010 Black Canon 712 – 574 грн.; картридж для EPSON STYLUS PHOTO R390 – 558 грн.; картридж для CANON IR-1022A – LJ Q2612A Cart. HP LJ 1010/1012/1015/3015/3020/3030 (2500 стр.) – 570 грн.

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

$$Z_M = (600 + 10800 + 1702) / 900 = 14,5 \text{ грн.}$$

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

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

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

$$O_n = 70 \cdot 15 \cdot 0,01 = 10,5 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 900$ прим.):

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

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

$$A_m = 175785 \cdot 3 / (900 \cdot 12) = 49 \text{ грн.}$$

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

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

$$C_n = 70 + 7 + 17 + 10,5 + 14,5 + 10,5 + 49 = 178,5 \text{ грн.}$$

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

Для даного програмного забезпечення рівень рентабельності складає 50%.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 178,5 = 89,25 \text{ грн.}$$

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

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1	2	3
1. Основна зарплата виконавців	Z_o	70
2. Додаткова зарплата виконавців	Z_d	7
3. Відрахування на соціальні потреби	C_{oc}	17
4. Загальногосподарські витрати	G_{ocn}	10,5
5. Витрати на матеріали	Z_M	14,5
6. Освоєння нових операційних систем, мов програмування	O_n	10,5
7. Амортизація основних фондів	A_m	49
8. Повна собівартість програмного забезпечення	C_n	178,5
9. Плановий прибуток	P_p	89,25
10. Ціна підприємства $C_n = C_n + P_p$	C_n	267,75
11. Податок на додану вартість $ПДВ = 0,01 \cdot H_{об} \cdot C_n$	$ПДВ$	53,55
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	321,3

Витрати на профілактичні роботи:

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

де: T_p – кількість годин обслуговування кожного комп'ютера за рік, год.;

Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 200 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 42 \cdot 1,1 \cdot 1,22 = 13527 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 200 \cdot 42 \cdot 1,1 \cdot 1,22 = 11273 \text{ грн}.$$

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

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

$$Z_{ел \text{ баз}} = 3 \cdot 0,35 \cdot 130 \cdot 1,9 = 88,73 \text{ грн}.$$

$$Z_{ел \text{ нов}} = 3 \cdot 0,35 \cdot 90 \cdot 1,9 = 61,2 \text{ грн}.$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн., за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	100	–	321	–	321
Всього відрахувань	-	–	321	–	321

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	900
2. Повна собівартість розробленої програми	Грн.	179
3. Ціна розробленої програми	Грн.	268
4. Плановий прибуток від реалізації розробленої програми	Грн.	89
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1975636
7. Загальний прибуток від реалізації програмної продукції	Грн.	80100
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	6239
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років.	6
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	321
11. Величина економічного ефекту у користувача програмної продукції	Грн.	1881
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,16

7.9 Висновки

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

8 ЗАХОДИ ЩОДО ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Протягом усієї історії людство приділяє прискіпливу увагу безпеці життя. Охорона праці є складовою частиною безпеки життя.

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

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

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98, та

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

«Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

Умови праці програміста вуючають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

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

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

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

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

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- монотонність праці;
- електромагнітні (у т.ч. високочастотні) випромінювання (коливання);

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шуми;
- статичні навантаження на кістково-м'язовий апарат;

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 - Розміри приміщення

Найменування	Значення, м
Ширина	4
Довжина	5
Висота	3

Таблиця 8.2 - Площа та обсяг приміщення, на одного працюючого

Геометрична характеристика	Одиниця виміру	Норматив не значення*	Фактич не значення
Площа, S	м ²	не менше 6.0	6,6
Обсяг, V	м ³	не менше 20.0	20

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працює 3 осіб. За даними, які наведено у табл. 8.1 та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста відповідають нормативним вимогам (Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про

затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

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

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

Таблиця 8.3 - Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Вологість, %	Швидкість повітря, м/с	Температура, °С	Вологість, %	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23,5	41-58	0,11
Тепла	23-25	50-70	0,1	24-25	50-60	0,12

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

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

В літню пору застосовується кондиціонер.

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

У приміщенні знаходяться наступні джерела шуму: принтер *Xerox Phaser 3020VI*, електродвигуни вентиляторів ЕОМ.

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

Працю працівника, який постійно працює за комп'ютером, згідно державних будівельних норм ДБН В.2.5 – 28 – 2006 р. (які замінили СНиП 11 -4-79), можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 лк. Крім того все поле зору повинне бути освітлено достатньо рівномірно - ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

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

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

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

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язково наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору розтіканню електричного струму на землю).

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

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

8.5 Розрахункова частина

Початкові данні для розрахунку штучного захисного заземлення:

Тип заземлення: повторне заземлення нульового дроту на ввводі в об'єкт.
Напруга — 220/380 В. Розташування заземлюючих електродів — по контуру.

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача методом коефіцієнта використання заземлювачів.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта — чорнозем, нижнього шару ґрунта — суглинок полутвердий. Умовна товщина верхнього шару ґрунта: $H=0,7$ м. Для захисного заземлення: застосовуються вертикальні електроди — прутки довжиною $L=2$ м. Відстань між вертикальними заземлювачами (електродами) $A=4$ м. Діаметр вертикального електрода (прутка) $D=12$ мм, Тип горизонтального заземлювача: металевий прутки. Діаметр горизонтального електрода (прутка) $D=12$ мм ($b=12$ мм.). Опір заземлювача, який нормується: $R_{3H} = 10$ Ом. Глибина закладення горизонтального контура заземлення $t= 0,7$ м.

Розрахунок захисного заземлення можна автоматизувати за допомогою програми, сирцевий код якої опублікован на стр.13-16 Метододичних вказівок до виконання розрахунків з викор. персон. ЕОМ IBM сумісного типу / Охорона праці. Ч. 1. Захисне заземлення / Кіровоград. ін-т с.-г. Машинобуд.; URL : <http://dspace.kntu.kr.ua/jspui/handle/123456789/4358>, або будь якої відповідної іншої.

Розрахунок.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,7+2/2=1,7 \text{ м.}$$

Еквівалентний питомий опір ґрунта:

$$\rho_{екв} = \psi \rho_1 \rho_2 L / [\rho_1 \psi(L-H+t) + \rho_2 \psi(H-t)] = 100 \text{ Ом}$$

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		103

$$N = R_O / (R_B * \eta_B) = 48,66 / (15,41 * 0,73) = 4 \text{ шт.}$$

де $\eta_B = 0,73$ - табличне значення коефіцієнта використання вертикального заземлювача, залежить від розташування (в ряд або по контуру) та співвідношення A/L .

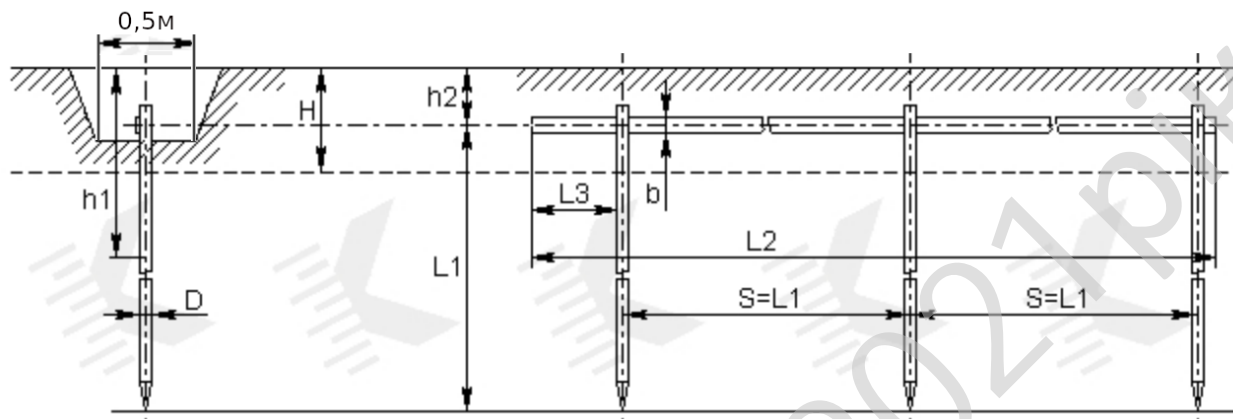


Рисунок 8.1 — Штучне повторне заземлення нульового дроту на ввіді в об'єкт.

Висновки до розділу

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

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

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0033.00.00.ПЗ

Арк.

105

9 ОСНОВНІ ВИСНОВКИ

Під час написання даної роботи були розкриті дві досить великі теми: автентифікація користувача і захист об'єктів від несанкціонованого доступу. В роботі розглянуті такі завдання як:

- проведення порівняльної оцінки різних засобів і методів захисту;
- розробка пропозицій щодо захисту інформації;
- визначення ефективності розробленої системи захисту.

Вході виконання роботи був проведений аналіз завдання, на основі якого були сформульовані вимоги до системи. На основі вимог була побудована структурна схема. На структурній схемі були підібрані відповідні пристрої для реалізації функцій, покладених на елементи системи. Особливістю даної охоронної системи є наявність великої кількості датчиків і рідкокристалічного індикатора, що оповіщає власника про можливу спробу проникнення, із зазначенням де саме. Потім, з використанням обраних пристроїв була побудована функціональна схема. Розробка завершилася складанням блок-схеми алгоритму.

Проаналізовано поняття системи охоронної сигналізації, з яких компонентів вона складається, як вони працюють між собою, які виконують функції. Розроблені під час виконання кваліфікаційної бакалаврської роботи алгоритми дозволяють успішно вирішувати завдання захисту об'єктів.

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 C++. Дана мова програмування дозволяє найбільш ефективно обробляти дані

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

призначені для системи автентифікації. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 7/8/10.

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.

Розроблені під час виконання магістерської роботи алгоритми дозволяють успішно вирішувати захист об'єктів з впровадженням системи кібербезпеки для захисту самої системи від взлому .

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

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

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 18811 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,16 років.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бородин И.Ф., Кирилин Н.И. Основы автоматики и автоматизации производственных процессов.-М.:Колос,1977.-328 с.
2. Бородин И.Ф., Недилько Н.М. Автоматизация технологических процессов.-М.:Агропромиздат,1991.-396 с.
3. Бородин И.Ф. Технические средства автоматики.-М.:Колос,1982.-303 с.
4. Горшков Б.И. Радиоэлектронные устройства: Справочник.-М.:Радио и связь,1984.-400 с.
5. Губський А.І. Цивільна оборона.-К.:Міністерство освіти, 1995.-231 с.
6. Защита объектов народного хозяйства от оружия массового поражения: Справочник/Г.П.Демиденко, Е.П.Кузьменко, П.П.Орлов и др.; Под ред. Г.П.Демиденко.-К.:Вища школа, 1989.-287 с.
7. Довідник по автоматизації с/г виробництва //За ред. І.І. Мартиненка.-К.:Урожай,1985.-212 с.
8. Кобевник В.Ф. Охрана труда.-К.:Вища школа, 1990.-286 с.
9. Мартыненко И.И. Автоматика и автоматизация производственных процессов.-М.:Агропромиздат,1985.-208 с.
10. Мартыненко И.И., Лысенко В.Ф. Проектирование систем автоматики.-2е изд., перераб. и доп.-М.:Агропромиздат,1991.-243с.
11. Мартыненко И.И., Тищенко Л.П. Курсовое и дипломное проектирование по комплексной электрификации и автоматизации.- М.:Колос, 1978.-224 с.
12. Монтаж приборов и средств автоматизации: Справочник / К.А. Алексеев, В.С. Антипин, Г.С. Борисова и др./Под ред. А.С. Клюева.- М.:Энергия,1979.-213 с.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

1. Кухарев Г.А., Биометрические системы: Методы и средства идентификации личности человека, -СПб.: Политехника, 2001. – 240 с.
2. Кухарев Г.А., Щеголева Н.Л., Системы распознавания человека по изображению лица: учебное пособие, -СПб.: издательство СПбГЭТУ «ЛЭТИ», 2006. – 176 с.
3. Веженец В.А. Обнаружение и локализация лица на изображении, 2004.
4. Michael J. Jones, James M. Rehg. Statistical Color Models with Application to Skin Detection.. In CVPR, 1999.
5. Культин В. Delphi в задачах и примерах: 2010 г. Количество страниц: 760
6. Любавин А. Самоучитель. Программирование на Delphi Win 32.: 2009г. Количество страниц:360
7. В. В. Котов, КПК для начинающих: Издательство: ЭКСМО, с. 160, 2008 г. ISBN: 978-5-699-25126-1.
8. Станислав Г. В. Работа на КПК, коммуникаторе, смартфоне под управлением Windows Mobile: с. 336, 2007 г. ISBN: 978-5-91180-498-5, 5-9706-0030-X.
9. Брюс Е. Крель Windows Mobile. Разработка приложений для КПК: Издательство: ДМК пресс, с. 352, 2009 г. ISBN: 978-5-94074-481-8.
10. Галатенко В.А. Идентификация и автентификация, управление доступом лекция из курса «Основы информационной безопасности». – Интернет Университет Информационных Технологий, 2010г.
11. Зиновьев А. Ю., Визуализация многомерных данных, Красноярск, Изд. КГТУ, 2000.
12. Muresan D. D., Parks T. W., Adaptive Principal Components and Image Denoising, in: Image Processing, 2003, Proceedings 2003 IEEE International Conference on Image Processing (ICIP), 14-17 Sept. 2003, V. 1, pp. I-101-104
13. Rao, K., Yip P. (eds.), The Transform and Data Compression Handbook,

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

CRC Press, Baton Rouge, 2001.

14. Scholz M., Fraunholz M., Selbig J., Nonlinear Principal Component Analysis: Neural Network Models and Applications, In: Gorban A. N. et al (Eds.), LNCSE 58, Springer, 2007 ISBN 978-3-540-73749-0

15. Zinovyev A., Cluster structures in genomic word frequency distributions, 1417 Sept. 2003, V. 1, pp. I-101-104

16. Дреев А.Н. Повышение вероятности доставки сообщений в телекоммуникационных системах и сетях для обеспечения информационной безопасности / А.Н. Дреев, А.А. Смирнов // «Безпека інформації» Том 21, №1 2015 р. – Київ: НАУ – 2015. – С. 22-28.

17. Дреев О.М. Метод прогнозування завантаженості серверу телекомунікаційної мережі / О.М. Дреев, О.В. Коваленко // Тези доповідей Новітні технології – для захисту повітряного простору. Дев'ята наукова конференція. 18-19 квітня 2011 р. – Х.: ХУПС. – 2012. – С. 206

18. Дреев О.М. Середньостатистичний та найімовірніший час доставки багатопакетного повідомлення в телекомунікаційній системі або мережі / О.М. Дреев, О.А. Смирнов // V Всеукраїнська науково-практична конференція "Інформатика та системні науки" ІСН – 2014, 13-15 березня 2014 року, м. Полтава – С. 92

19. ДСТУ 2481 – 94 Системи оброблення інформації інтелектуальні інформаційні технології. Терміни та визначення. – Х.: ДЕРЖСТАНДАРТ УКРАЇНИ, 1994. – 33 с.

20. ДСТУ В 3265 – 95. Зв'язок військовий. Терміни та визначення. – К.:УкрНДІССІ, 1995. – 23 с.

21. Дымарский Я.С. Управление сетями связи: принципы, протоколы, прикладные задачи / Я.С. Дымарский., Н.П. Крутякова, Г.Г. Яновский – М.: ЭкоТрендз, 2003. – 384 с.

22. Ершов В.А. Мультисервисные телекоммуникационные сети / В.А. Ершов, Н.А. Кузнецов – М.: Изд. МГТУ им. Н.Э. Баумана, 2003. – 432 с.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

23. Закон України «Про Концепцію Національної програми інформатизації» (Відомості Верховної Ради (ВВР), 1998, N 27-28, ст.181) (Із змінами, внесеними згідно із Законом N 5463-VI (5463 -17) від 16.10.2012, ВВР, 2014, N 4, ст.61). – <http://zakon1.rada.gov.ua/laws/show/74/98-%D0%B2%D1%80>.

24. Зайченко Ю.П. Компьютерные сети / Ю.П. Зайченко. – К.: Слово, 2003. – 256 с.

25. Зубко Р.А. Алгоритми стиснення зображень в системах цифрової обробки даних / Р.А. Зубко // Восточно-Европейский журнал передовых технологий. – 2013. – №1/2(61). – С. 40-44.

26. Иванов В.Г. Прогрессивные информационные технологии сжатия изображений / Иванов В. Г., Ломоносов Ю. В., Любарский М. Г. // Проблеми інформатики та комп'ютерної техніки (ПІКТ – 2014) : пр. III-ї Міжнар. наук.-практ. конф., 27–30 трав. 2014 р. – Чернівці : Родовід, 2014. – С. 172–174.

27. Ивахненко А.Г. Долгосрочное прогнозирование и управление сложными системами / А.Г. Ивахненко – Киев: «Техніка». – 1975. – 312 с.

28. Игнатенко Е.Г., Бессараб В.И. Алгоритм адаптивного мониторинга загрузки кластерных web-серверов / Е.Г. Игнатенко, В.И. Бессараб // Збірник тез. Нові технології в телекомунікаціях. VI міжнародний науково-технічний симпозіум. – Карпати, Вишків: ДУІКТ. – 2011. – С.34.

29. Карпенко С.В. Метод компактного представлення зображень в телекомунікаційних системах на основі тривимірного поліадичного кодування: автореф. дис. на здобуття наук. ступеня кандидата техн. наук: спец. 05.12.02 – телекомунікаційні системи та мережі / С.В. Карпенко. – Харків: Харківський національний університет радіоелектроніки, 2009. – 22 с.

30. Касимов Р.Р. Вдосконалення алгоритмів QoS маршрутизації в мережах з технологією IP/MPLS на основі прогнозу трафіка: автореф. дис. на здобуття наук. ступеня кандидата техн. наук: спец. 05.12.02 –телекомунікаційні системи та мережі / Р.Р. Касимов. – Київ: Київський державний університет інформаційно-комунікаційних технологій, 2011. – 24 с.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		111

31. Кириченко Л.О. Влияние методов маршрутизации на QOS в мультисервисных сетях при самоподобной нагрузке / Л.О. Кириченко, Т.А. Радивилова, Э. Кайали // Восточно-Европейский журнал передовых технологий. Информационные технологии. 1/2 (49) 2011. – С. 15-18

32. Коваленко А.А. Методи та засоби підвищення оперативності передачі даних у мультисервісних мережах: автореф. дис. на здобуття наук. ступеня кандидата техн. наук: спец. 05.13.05 «комп'ютерні системи та компоненти»/ А.А. Коваленко. – Харків: Харківський національний університет радіоелектроніки, 2008. – 20 с.

33. Комарова Л.О. Алгоритми управління потоками даних у телекомунікаційній мережі реального часу / Л.О. Комарова // Телекомунікаційні та інформаційні технології. – 2014. – №2 – С. 13-18.

34. Кормен Т. Алгоритмы: построение и анализ / Томас Кормен, Чарльз Лейзерсон, Рональд Ривест, Клиффорд Штайн. – М.: "Вильямс", 2005. – 1296 с.

35. Конахович Г.Ф. Сети передачи пакетных данных / Г.Ф. Конахович, В.М.Чуприн. – К.:МК-Пресс, 2006. – 272 с.

36. Королев А.В. Адаптивная маршрутизация в корпоративных сетях / А.В. Королев, Г.А. Кучук, А.А. Пашнев. – Х.: ХВУ, 2003. – 224 с.

37. Королев А.В. Управление сетевыми ресурсами / А.В. Королев, Г.А. Кучук, А.А. Пашнев. – Х.: ХВУ, 2004. – 272 с.

38. Красильников Н.Н. Теория передачи и восприятия изображений / Н.Н. Красильников – М.: Радио и Связь, 1986. – 248 с.

39. Кренке Д. Теория и практика построения баз данных / Д.Кренка. – 10-е изд - СПб.: Питер, 2010. – 859 с.: ил.

40. Непейвода Н.Н. Стили и методы программирования Интернет-университет информационных технологий - ИНТУИТ.ру, 2005.

41. Хоган Б. Книга веб-программиста. Секреты профессиональной разработки веб-сайтов / Б. Хоган, К. Уоррен, М. Уэбер, К. Джонсон, А. Годин. – СПб.: Питер, 2013. – 288 с.

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

42. Люк Веллинг, Лора Томсон. Розробка Web -додатків за допомогою PHP і MySQL : Вільямс, 2012р., - 880 с.
43. <http://dspace.nbuu.gov.ua/> - Наукова електронна бібліотека періодичних видань НАН України.
44. <https://habr.com> - веб-сайт, який поєднує ознаки соціальної мережі і колективного блогу, створений для публікації новин, аналітичних статей, думок, пов'язаних із інформаційними технологіями, бізнесом та Інтернетом.
45. Державні будівельні норми України: ДБН В.2.5-28:2018. - Режим доступу до ресурсу: <https://goo.su/9AkQ>
46. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. - Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>
47. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. - Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>
48. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.
49. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». - Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>
50. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. - Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99>
51. Центр післядипломної освіти та підвищення кваліфікації. - Режим доступу до ресурсу: <https://cпо.stu.cn.ua>

					ВКРМ-123.21.0033.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРМ-123.21.0033.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Нетребенко В.				<i>Дослідження та програмна реалізація системи кібербезпеки для охоронної системи</i>		
Перевірів	Босько В.В.						
Н. Контр.	Гермак В.С.				Літ.	Аркуш	Аркушів
Затв.	Смірнов О.А.				Б	1	6
					ЦНТУ КІ-20М		

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку програмного забезпечення системи кібербезпеки для охоронної системи.

2 Підстава для розробки

Підставою для розробки служить завдання на магістерську роботу, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 42-13 від 02.08.2021 року).

3 Мета та призначення розробки

Метою магістерської роботи є розробка програмного забезпечення охоронної системи з впровадженням системи кібербезпеки для захисту системи від взлому.

4 Джерела розробки

Джерелом цієї магістерської роботи є відносна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.21.0033.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

– розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- надійний захист віддаленого об'єкта;
- систему кіберзахисту від втручання в роботу системи;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРМ-123.21.0033.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище PHP та C++.

					ВКРМ-123.21.0033.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Економічні вимоги

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

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

8 Вимоги щодо охорони праці

В частині охорони праці та техніки безпеки в магістерській роботі повинен бути розглянутий аналіз умов праці програміста та розрахунок штучного захисного заземлення.

					ВКРМ-123.21.0033.00.00.ТЗ	Арк.
						5
Вим.	Арк.	№ документа	Підпис	Дата		

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 113 аркушів.

10 Етапи розробки

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

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень бакалаврської дипломної роботи.

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

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

9 Порядок контролю та приймання

11.1 Подання бакалаврської дипломної роботи на попередній захист
10.12.2021 р.

1.2 Подання магістерської роботи на захист 10.12.2021 р.

					ВКРМ-123.21.0033.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Босько В.В.

Дослідження та програмна реалізація системи кібербезпеки для охоронної системи

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 39

Літера: РП

Кропивницький – 2021 року

Файл Networks.pas - робота з мережею програмно визначаемого ЦОД на базі технологій Fujitsu

```

unit Networks;

interface

uses Windows, ShellAPI, ShlObj, ActiveX, ComObj, Dim, Classes, SysUtils,
WinSock;

type
  ComputerFound = class (Exception);
  ECannotFindNetwork = class (Exception);

  TStringObject = class (TObject)
  private
    FValue: TString;
    FTag: Integer;
    FData: Pointer;
    FRefObj: TObject;
    procedure SetValue(const Value: TString);
    procedure SetData(const Value: Pointer);
    procedure SetRefObj(const Value: TObject);
    procedure SetTag(const Value: Integer);
  public
    property Value: TString read FValue write SetValue;
    property RefObj: TObject read FRefObj write SetRefObj;
    property Tag: Integer read FTag write SetTag;
    property Data: Pointer read FData write SetData;
  end;

  TStringObjectArray = class (TDynamicArray)
  private
    function GetObject(Index: Integer): TStringObject;
    function GetData(Index: Integer): Pointer;
    function GetRefObj(Index: Integer): TObject;
    function GetTag(Index: Integer): Integer;
    function GetValue(Index: Integer): TString;
    procedure SetData(Index: Integer; const Value: Pointer);
    procedure SetRefObj(Index: Integer; const Value: TObject);
    procedure SetTag(Index: Integer; const Value: Integer);
    procedure SetValue(Index: Integer; const Value: TString);

    function FreeObject(Index: Integer; var Obj: TStringObject): Integer;

    procedure FreeItem(Index: Integer);
    procedure CreateItem(Index: Integer);

  protected
    procedure SetCount(const NewCount: Cardinal); override;
  public
    function Add: Integer; override;
    procedure Insert(Index: Integer); override;
    procedure Delete(Index: Integer); override;
    function AddItem(const Item): Integer; override;
    procedure InsertItem(Index: Integer; const Item); override;
    procedure DeleteItem(Index: Integer; out Item); override;
    property Value[Index: Integer]: TString read GetValue write SetValue;
  default;
    property RefObj[Index: Integer]: TObject read GetRefObj write SetRefObj;
    property Tag[Index: Integer]: Integer read GetTag write SetTag;
    property Data[Index: Integer]: Pointer read GetData write SetData;
    constructor Create;
    destructor Destroy; override;
  end;
end;

```

```

TStringObjectList = class (TStrings)
private
  FArray: TStringObjectArray;
  function GetData(Index: Integer): Pointer;
  function GetTag(Index: Integer): Integer;
  procedure SetData(Index: Integer; const Value: Pointer);
  procedure SetTag(Index: Integer; const Value: Integer);
protected
  function Get(Index: Integer): string; override;
  function GetCount: Integer; override;
  function GetObject(Index: Integer): TObject; override;
  procedure Put(Index: Integer; const S: string); override;
  procedure PutObject(Index: Integer; AObject: TObject); override;
public
  property Data[Index: Integer]: Pointer read GetData write SetData;
  property Tag[Index: Integer]: Integer read GetTag write SetTag;

  function Add(const S: string): Integer; override;
  procedure Clear; override;
  procedure Delete(Index: Integer); override;
  procedure Exchange(Index1, Index2: Integer); override;
  procedure Insert(Index: Integer; const S: string); override;

  constructor Create;
  destructor Destroy; override;
end;

```

{TNetworkWorkgroup - клас, який створює список всіх комп'ютерів в робочій групі. Цей клас є нащадком класу TStrings і повністю сумісний з іншим нащадками цього класу. Об'єкти цього класу записуються у властивості об'єктів класу TNetworkNeighborhood. Клас TNetworkNeighborhood містить об'єкти і властивості робочих груп. }

```
TNetworkWorkgroup = class (TStringObjectList);
```

```

{TNetworkNeighborhood - клас, який створює список всіх робочих груп в мережі}
TNetworkNeighborhood = class (TStringObjectList)
private
  function CreatePIDL(Size: Integer): PItemIDList;
  procedure DisposePIDL(ID: PItemIDList);
  function NextPIDL(IDList: PItemIDList): PItemIDList;
  function GetPIDLSize(IDList: PItemIDList): Integer;
  function CopyPIDL(IDList: PItemIDList): PItemIDList;
  procedure StripLastID(IDList: PItemIDList);
  function GetPrevPIDL(PIDL: PItemIDList): PItemIDList;
  class function GetDisplayName(ShellFolder: IShellFolder; PIDL: PItemIDList):
TString;
  function OriginFolder: IShellFolder;
  function OriginFolderNT: IShellFolder;
  class function EnumObjects(ShellFolder: IShellFolder): IEnumIDList;
  class procedure ParseFolder(Folder: IShellFolder; Items: TStringObjectList;
StorePIDLs: Boolean = False);
  class procedure ParseFolderEx(Folder: IShellFolder; Items: TStrings);

  function FreeRefObj(Index: Integer; var Obj: TStringObject): Integer;
  function GetWorkgroup(Name: TString): TNetworkWorkgroup;
public
  { Процедура Оновлення шукає всі доступні робочі групи в мережі }

  procedure Refresh;

  { містить списки всіх комп'ютерів в мережі}

  property Workgroup[Name: TString]: TNetworkWorkgroup read GetWorkgroup;

```

```

{ Функція FindComputer шукає комп' ютер зі вказаним ім' ям і повертає ім' я
робочої групи де його знайдено, або порожню строку
якщо комп' ютера з таким іменем у мережі немає }

function FindComputer(Name: TString): TString;

{ Процедура ListComputers копіює список всіх комп' ютерів в мережі
у об' екти TStrings}
procedure ListComputers(Strings: TStrings);

{ Процедура ListNetwork копіює відсортований в алфавітному порядку список
всіх робочих груп і комп' ютерів в мережі.
Робочі групи мають ` TObject(1)` у відповідному елементі властивості об'
ектів, а комп' ютери - ` nil' }

procedure ListNetwork(Strings: TStrings);

function Add(const S: string): Integer; override;
procedure Clear; override;
procedure Delete(Index: Integer); override;
procedure Insert(Index: Integer; const S: string); override;
constructor Create;
end;

{ Функція GetIPAddress отримує IP адресу комп' ютера або сервера.
Параметр NetworkName конкретизує ім' я комп' ютера або сервера.
Ця функція повертає адреси IP у форматі XXX.XXX.XXX.XXX у разі успішного
виконання, ` Error' - коли неможливо ініціалізувати, ` Unknown' - коли
параметр NetworkName посилається на неіснуючий комп' ютер або на комп' ютер без
встановленого протоколу TCP/IP }

function GetIPAddress(NetworkName: TString): TString;

{ GetIPAddresses отримує IP адреси всіх доступних комп' ютерів мережі }
procedure GetIPAddresses(Network: TNetworkNeighborhood; List: TStrings);

{ Функція EnumSharedResources перераховує загальні ресурси мережі. Параметр
ComputerName конкретизує
ім' я комп' ютера. }

function EnumSharedResources(ComputerName: TString; List: TStrings): Boolean;

implementation

uses NetConst;

{ TStringObject }

procedure TStringObject.SetData(const Value: Pointer);
begin
  FData := Value;
end;

procedure TStringObject.SetRefObj(const Value: TObject);
begin
  FRefObj := Value;
end;

procedure TStringObject.SetTag(const Value: Integer);
begin
  FTag := Value;
end;

procedure TStringObject.SetValue(const Value: TString);

```

```

begin
  FValue := Value;
end;

{ TStringObjectArray }

function TStringObjectArray.Add: Integer;
begin
  Result:=inherited Add;
  CreateItem(Result);
end;

function TStringObjectArray.AddItem(const Item): Integer;
begin
  Result:=Add;
end;

constructor TStringObjectArray.Create;
begin
  inherited Create(0, SizeOf(TStringObject));
end;

procedure TStringObjectArray.CreateItem(Index: Integer);
var
  P: ^TStringObject;
begin
  P:=GetItemPtr(Index);
  P^:=TStringObject.Create;
end;

procedure TStringObjectArray.Delete(Index: Integer);
begin
  FreeItem(Index);
  inherited;
end;

procedure TStringObjectArray.DeleteItem(Index: Integer; out Item);
begin
  Delete(Index);
end;

destructor TStringObjectArray.Destroy;
begin
  ForEach(Integer(Self), @TStringObjectArray.FreeObject);
  inherited;
end;

procedure TStringObjectArray.FreeItem(Index: Integer);
var
  P: ^TStringObject;
begin
  P:=GetItemPtr(Index);
  FreeAndNil(P^);
end;

function TStringObjectArray.FreeObject(Index: Integer;
  var Obj: TStringObject): Integer;
begin
  FreeAndNil(Obj);
  Result:=0;
end;

function TStringObjectArray.GetData(Index: Integer): Pointer;
begin
  Result:=GetObject(Index).Data;
end;

function TStringObjectArray.GetObject(Index: Integer): TStringObject;
begin

```

```

    GetItem(Index, Result);
end;

function TStringObjectArray.GetRefObj(Index: Integer): TObject;
begin
    Result:=GetObject(Index).RefObj;
end;

function TStringObjectArray.GetTag(Index: Integer): Integer;
begin
    Result:=GetObject(Index).Tag;
end;

function TStringObjectArray.GetValue(Index: Integer): TString;
begin
    Result:=GetObject(Index).Value;
end;

procedure TStringObjectArray.Insert(Index: Integer);
begin
    inherited;
    CreateItem(Index);
end;

procedure TStringObjectArray.InsertItem(Index: Integer; const Item);
begin
    Insert(Index);
end;

procedure TStringObjectArray.SetCount(const NewCount: Cardinal);
var
    i, OldCount: Integer;
begin
    OldCount:=Count;
    if NewCount > Count then begin
        inherited SetCount(NewCount);
        for i:=OldCount to NewCount - 1 do CreateItem(i);
    end else if NewCount < Count then begin
        for i:=NewCount to OldCount - 1 do FreeItem(i);
        inherited SetCount(NewCount);
    end;
end;

procedure TStringObjectArray.SetData(Index: Integer; const Value: Pointer);
begin
    GetObject(Index).Data:=Value;
end;

procedure TStringObjectArray.SetRefObj(Index: Integer;
    const Value: TObject);
begin
    GetObject(Index).RefObj:=Value;
end;

procedure TStringObjectArray.SetTag(Index: Integer; const Value: Integer);
begin
    GetObject(Index).Tag:=Value;
end;

procedure TStringObjectArray.SetValue(Index: Integer; const Value: TString);
begin
    GetObject(Index).Value:=Value;
end;

{ TStringObjectList }

function TStringObjectList.Add(const S: string): Integer;
begin
    Result:=FArray.Add;

```

```

    FArray.Value[Index]:=S;
end;

procedure TStringObjectList.Clear;
begin
    FArray.Count:=0;
end;

constructor TStringObjectList.Create;
begin
    inherited Create;
    FArray:=TStringObjectArray.Create;
end;

procedure TStringObjectList.Delete(Index: Integer);
begin
    FArray.Delete(Index);
end;

destructor TStringObjectList.Destroy;
begin
    FArray.Free;
    inherited;
end;

procedure TStringObjectList.Exchange(Index1, Index2: Integer);
begin
    FArray.Swap(Index1, Index2);
end;

function TStringObjectList.Get(Index: Integer): string;
begin
    Result:=FArray.Value[Index];
end;

function TStringObjectList.GetCount: Integer;
begin
    Result:=FArray.Count;
end;

function TStringObjectList.GetData(Index: Integer): Pointer;
begin
    Result:=FArray.Data[Index];
end;

function TStringObjectList.GetObject(Index: Integer): TObject;
begin
    Result:=FArray.RefObj[Index];
end;

function TStringObjectList.GetTag(Index: Integer): Integer;
begin
    Result:=FArray.Tag[Index];
end;

procedure TStringObjectList.Insert(Index: Integer; const S: string);
begin
    FArray.Insert(Index);
    FArray.Value[Index]:=S;
end;

procedure TStringObjectList.Put(Index: Integer; const S: string);
begin
    FArray.Value[Index]:=S;
end;

procedure TStringObjectList.PutObject(Index: Integer; AObject: TObject);
begin
    FArray.RefObj[Index]:=AObject;
end;

```

```

end;

procedure TStringObjectList.SetData(Index: Integer; const Value: Pointer);
begin
  FArray.Data[Index]:=Value;
end;

procedure TStringObjectList.SetTag(Index: Integer; const Value: Integer);
begin
  FArray.Tag[Index]:=Value;
end;

{ TNetworkNeighborhood }

function TNetworkNeighborhood.Add(const S: string): Integer;
begin
  Result:=inherited Add(S);
  Objects[Result]:=TNetworkWorkgroup.Create;
end;

procedure TNetworkNeighborhood.Clear;
begin
  FArray.ForEach(Integer(Self), @TNetworkNeighborhood.FreeRefObj);
  inherited;
end;

function TNetworkNeighborhood.CopyPIDL(IDList: PItemIDList): PItemIDList;
var
  Size: Integer;
begin
  Size := GetPIDLSize(IDList);
  Result := CreatePIDL(Size);
  if Assigned(Result) then CopyMemory(Result, IDList, Size);
end;

constructor TNetworkNeighborhood.Create;
begin
  inherited Create;
  Refresh;
end;

function TNetworkNeighborhood.CreatePIDL(Size: Integer): PItemIDList;
var
  Malloc: IMalloc;
  HR: HRESULT;
begin
  Result := nil;
  HR := SHGetMalloc(Malloc);
  if Failed(HR) then Exit;
  try
    Result := Malloc.Alloc(Size);
    if Assigned(Result) then FillChar(Result^, Size, 0);
  finally
    end;
  end;

procedure TNetworkNeighborhood.Delete(Index: Integer);
begin
  end;

procedure TNetworkNeighborhood.DisposePIDL(ID: PItemIDList);
var
  Malloc: IMalloc;
begin
  if ID = nil then Exit;
  OLECheck(SHGetMalloc(Malloc));
  Malloc.Free(ID);
end;

```

```

class function TNetworkNeighborhood.EnumObjects(
  ShellFolder: IShellFolder): IEnumIDList;
const
  Flags = SHCONTF_FOLDERS or SHCONTF_NONFOLDERS or SHCONTF_INCLUDEHIDDEN;
begin
  ShellFolder.EnumObjects(0, Flags, Result);
end;

function TNetworkNeighborhood.FindComputer(Name: TString): TString;
var
  i, j: Integer;
  List: TNetworkWorkgroup;
  S: TString;
begin
  Result:=' ';
  try
    for i:=0 to Count - 1 do begin
      List:=Objects[i] as TNetworkWorkgroup;
      for j:=0 to List.Count - 1 do begin
        S:=List[j];
        CleanUp(S);
        if EqualText(Name, S) then begin
          Result:=Strings[i];
          raise ComputerFound.Create(' ');
        end;
      end;
    end;
  except
    if not (ExceptObject is ComputerFound) then raise;
  end;
end;

function TNetworkNeighborhood.FreeRefObj(Index: Integer;
  var Obj: TStringObject): Integer;
begin
  FreeAndNil(Obj.FRefObj);
  Result:=0;
end;

class function TNetworkNeighborhood.GetDisplayName(ShellFolder: IShellFolder;
  PIDL: PItemIDList): TString;
var
  StrRet: TStrRet;
  P: PChar;
begin
  Result := ' ';
  ShellFolder.GetDisplayNameOf(PIDL, SHGDN_NORMAL, StrRet);
  case StrRet.uType of
    STRRET_CSTR: SetString(Result, StrRet.cStr, lStrLen(StrRet.cStr));
    STRRET_OFFSET: begin
      P := @PIDL.mkid.abID[StrRet.uOffset - SizeOf(PIDL.mkid.cb)];
      SetString(Result, P, PIDL.mkid.cb - StrRet.uOffset);
    end;
    STRRET_WSTR: Result := StrRet.pOleStr;
  end;
  CleanUp(Result, True);
end;

function TNetworkNeighborhood.GetPIDLSize(IDList: PItemIDList): Integer;
begin
  Result := 0;
  if Assigned(IDList) then begin
    Result := SizeOf(IDList^.mkid.cb);
    while IDList^.mkid.cb <> 0 do begin
      Result := Result + IDList^.mkid.cb;
      IDList := NextPIDL(IDList);
    end;
  end;
end;

```

```

function TNetworkNeighborhood.GetPrevPIDL(PIDL: PItemIDList): PItemIDList;
var
  Temp: PItemIDList;
begin
  Temp := CopyPIDL(PIDL);
  if Assigned(Temp) then StripLastID(Temp);
  if Temp.mkid.cb <> 0 then Result:=Temp else Result:=nil;
end;

function TNetworkNeighborhood.GetWorkgroup(Name: TString): TNetworkWorkgroup;
var
  Index: Integer;
begin
  Index:=IndexOf(Name);
  if Index<>-1 then Result:=Objects[Index] as TNetworkWorkgroup else Result:=nil;
end;

procedure TNetworkNeighborhood.Insert(Index: Integer; const S: string);
begin
end;

procedure TNetworkNeighborhood.ListComputers(Strings: TStrings);
var
  i, j: integer;
  L: TNetworkWorkgroup;
  S: TString;
begin
  Strings.BeginUpdate;
  try
    Strings.Clear;
    for i:=0 to Count - 1 do begin
      L:=Objects[i] as TNetworkWorkgroup;
      for j:=0 to L.Count - 1 do begin
        S:=L[j];
        CleanUp(S);
        Strings.Add(S);
      end;
    end;
  finally
    Strings.EndUpdate;
  end;
end;

procedure TNetworkNeighborhood.ListNetwork(Strings: TStrings);
var
  List: TStringList;
  i: Integer;
begin
  List:=TStringList.Create;
  try
    List.AddStrings(Self);
    for i:=0 to List.Count - 1 do List.Objects[i]:=TObject(1);
    for i:=0 to Count - 1 do begin
      List.AddStrings(Objects[i] as TStrings);
    end;
    for i:=Count to List.Count - 1 do List.Objects[i]:=nil;
    List.Sort;
    Strings.Assign(List);
  finally
    List.Free;
  end;
end;

function TNetworkNeighborhood.NextPIDL(IDList: PItemIDList): PItemIDList;
begin
  Result := IDList;
  Inc(PChar(Result), IDList^.mkid.cb);
end;

```

```

function TNetworkNeighborhood.OriginFolder: IShellFolder;
var
  Desktop: IShellFolder;
  S: TString;
  P: PWideChar;
  Len, Flags: LongWord;
  Machine, Workgroup, Network: PItemIDList;
begin
  S:= '\ \\' +GetComputerName;
  Len:=Length(S);
  P:=StringToOleStr(S);
  Flags:=0;
  SHGetDesktopFolder(Desktop);
  Desktop.ParseDisplayName(0, nil, P, Len, Machine, Flags);
  Workgroup:=GetPrevPIDL(Machine);
  try
    Network:=GetPrevPIDL(Workgroup);
    try
      Desktop.BindToObject(Network, nil, IShellFolder, Pointer(Result));
    finally
      DisposePIDL(Network);
    end;
  finally
    DisposePIDL(Workgroup);
  end;
end;

function TNetworkNeighborhood.OriginFolderNT: IShellFolder;
var
  Desktop: IShellFolder;
  S: TString; W: WideString; P: PWideChar;
  Len, Flags: LongWord;
  Machine, Workgroup, Network: PItemIDList;
  NetShell: IShellFolder;
  Enum: IEnumIDList;
  ID: PItemIDList;
begin
  S:= '\ \\' +GetComputerName;
  Len:=Length(S);
  W:=S; P:=PWideChar(W);
  SHGetDesktopFolder(Desktop);
  Desktop.ParseDisplayName(0, nil, P, Len, Machine, Flags);
  Workgroup:=GetPrevPIDL(Machine);
  Network:=GetPrevPIDL(Workgroup);
  Desktop.BindToObject(Network, nil, IShellFolder, NetShell);
  Enum:=EnumObjects(NetShell);
  Enum.Next(1, ID, Flags);
  NetShell.BindToObject(ID, nil, IShellFolder, Pointer(Result));
  DisposePIDL(Network);
  DisposePIDL(Workgroup);
end;

class procedure TNetworkNeighborhood.ParseFolder(Folder: IShellFolder;
  Items: TStringObjectList; StorePIDLs: Boolean);
var
  ID: PItemIDList;
  EnumList: IEnumIDList;
  NumIDs: LongWord;
  S: TString;
  Index: Integer;
begin
  Items.BeginUpdate;
  try
    Items.Clear;
    EnumList:=EnumObjects(Folder);
    if Assigned(EnumList) then while EnumList.Next(1, ID, NumIDs) = S_OK do begin
      S:=GetDisplayName(Folder, ID);
      Index:=Items.Add(S);
    end;
  end;
end;

```

```

    if StorePIDs then Items.Data[Index]:=ID;
    end;
finally
    Items.EndUpdate;
end;
end;

class procedure TNetworkNeighborhood.ParseFolderEx(Folder: IShellFolder;
    Items: TStrings);
var
    ID: PItemIDList;
    EnumList: IEnumIDList;
    NumIDs: LongWord;
    S: TString;
begin
    Items.BeginUpdate;
    try
        Items.Clear;
        EnumList:=EnumObjects(Folder);
        if Assigned(EnumList) then while EnumList.Next(1, ID, NumIDs) = S_OK do begin
            S:=GetDisplayName(Folder, ID);
            Items.Add(S);
        end;
    finally
        Items.EndUpdate;
    end;
end;

procedure TNetworkNeighborhood.Refresh;
var
    Network: IShellFolder;
    Workgroup: IShellFolder;
    i: Integer;
begin
    try
        if WinNT and (not Win2K) then Network:=OriginFolderNT else
            Network:=OriginFolder;
        ParseFolder(Network, Self, True);
        for i:=0 to Count - 1 do begin
            Network.BindToObject(PItemIDList(Data[i]), nil, IShellFolder, Workgroup);
            ParseFolder(Workgroup, Objects[i] as TStringObjectList, False);
            Workgroup:=nil;
        end;
    except
        raise ECannotFindNetwork.Create(SCannotFindNetwork);
    end;
end;

procedure TNetworkNeighborhood.StripLastID(IDList: PItemIDList);
var
    MarkerID: PItemIDList;
begin
    MarkerID := IDList;
    if Assigned(IDList) then begin
        while IDList.mkid.cb <> 0 do begin
            MarkerID := IDList;
            IDList := NextPIDL(IDList);
        end;
        MarkerID.mkid.cb := 0;
    end;
end;

procedure GetIPAddresses(Network: TNetworkNeighborhood; List: TStrings);
var
    Error: DWORD;
    HostEntry: PHostEnt;
    Data: WSADATA;
    Address: In_Adr;

```

```

i: Integer;
TmpList: TStringList;
S: TString;
begin
{ List.BeginUpdate;
try}
List.Clear;
Error:=WSAStartup(MakeWord(1, 1), Data);
if Error = 0 then begin
  TmpList:=TStringList.Create;
  try
    Network.ListComputers(TmpList);
    for i:=0 to TmpList.Count - 1 do begin
      HostEntry:=gethostbyname(PChar(TmpList[i]));
      Error:=GetLastError;
      if Error <> 0 then S:=' Unknown' else begin
        Address:=PInAddr(HostEntry^.h_addr_list^);
        S:=inet_ntoa(Address);
        end;
      List.Add(Format(' %s [%s]' , [TmpList[i], S]));
      end;
    finally
      TmpList.Free;
    end;
  end else begin
    List.Add(' Error' );
  end;
{ finally
  List.EndUpdate;
end;}
end;

function GetShellFolder(ComputerName: TString): IShellFolder;
var
S: TString;
W: WideString;
P: PWideChar;
Desktop: IShellFolder;
Len, Flags: LongWord;
Machine: PItemIDList;
begin
S:=ComputerName;
if Pos('\ \', S) <> 1 then S:='\ \\' +S;
Len:=Length(S);
W:=S;
P:=@W[1];
SHGetDesktopFolder(Desktop);
Desktop.ParseDisplayName(0, nil, P, Len, Machine, Flags);
Desktop.BindToObject(Machine, nil, IShellFolder, Pointer(Result));
end;

function EnumSharedResources(ComputerName: TString; List: TStrings): Boolean;
var
ShellFolder: IShellFolder;
begin
ShellFolder:=GetShellFolder(ComputerName);
Result:=Assigned(ShellFolder);
if Result then TNetworkNeighborhood.ParseFolderEx(ShellFolder, List);
end;

function GetIPAddress(NetworkName: TString): TString;
var
Error: DWORD;
HostEntry: PHostEnt;
Data: WSADATA;
Address: In_Addr;

```

```
begin
  Error:=WSAStartup(MakeWord(1, 1), Data);
  if Error = 0 then begin
    HostEntry:=gethostbyname(PChar(NetworkName));
    Error:=GetLastError();
    if Error = 0 then begin
      Address:=PInAddr(HostEntry^.h_addr_list)^;
      Result:=inet_ntoa(Address);
    end else begin
      Result:=' Unknown' ;
    end;
  end else begin
    Result:=' Error' ;
  end;
  WSACleanup();
end;

end.
```

Кафедра КБПЗ – 2021 рік

Основна програма

Файл Main.pas основної програми

```
unit Main;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Dim, Networks, StdCtrls, ComCtrls, ImgList, ShellAPI, ShlObj, IpHlpApi,
  IPtraff,
  ExtCtrls, About, Buttons;

type
  TForm1 = class(TForm)
    Memo1: TMemo;
    ClickMe: TButton;
    TreeView1: TTreeView;
    ImageList1: TImageList;
    ListView1: TListView;
    Memo2: TMemo;
    Button1: TButton;
    Label1: TLabel;
    Edit1: TEdit;
    Memo3: TMemo;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    MemoTR: TMemo;
    Timer1: TTimer;
    Label7: TLabel;
    Button2: TButton;
    Button3: TButton;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    MemoX: TMemo;
    Label8: TLabel;
    procedure ClickMeClick(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);

  private
    { Private declarations }
    procedure DOIpStuff;
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}
```

```

function GetShellFolder(CompName: TString): IShellFolder;
var
  S: TString;
  W: WideString;
  P: PWideChar;
  Desktop: IShellFolder;
  Len, Flags: LongWord;
  Machine: PItemIDList;
begin
  S:=CompName;
  if Pos('\ \', S) <> 1 then S:='\ \'+S;
  Len:=Length(S);
  W:=S;
  P:=@W[1];
  SHGetDesktopFolder(Desktop);
  Desktop.ParseDisplayName(0, nil, P, Len, Machine, Flags);
  Desktop.BindToObject(Machine, nil, IShellFolder, Pointer(Result));
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  Memo3.Lines.Clear;
  Screen.Cursor:=crHourGlass;
  try
    if not EnumSharedResources(Edit1.Text, Memo3.Lines)
      then raise Exception.Create('Невірне ім'я комп'ютера');
  finally
    Screen.Cursor:=crDefault;
  end;
end;

procedure TForm1.ClickMeClick(Sender: TObject);
var
  N: TNetworkNeighborhood;
  List: TStringList;
  i, j: Integer;
  ListViewItem: TListItem;
  WorkgroupNode, ComputerNode: TTreeNode;
begin
  Screen.Cursor:=crHourGlass;
  try
    // Ініціалізація об'єктів та сканування мережі
    N:=TNetworkNeighborhood.Create;
    try
      // Отримання списку всіх комп'ютерів в мережі
      Memo1.Lines.Clear;
      N.ListComputers(Memo1.Lines);

      // Отримання списку всіх робочих груп і комп'ютерів в мережі, відсортованих
      в алфавітному порядку
      List:=TStringList.Create;
      ListView1.Items.Clear;
      try
        N.ListNetwork(List);
        for i:=0 to List.Count - 1 do begin
          ListViewItem:=ListView1.Items.Add;
          ListViewItem.Caption:=List[i];
          ListViewItem.ImageIndex:=Integer(List.Objects[i]);
        end;
      finally
        List.Free;
      end;
    end;
  end;
end;

```

```

// Побудова дерева робочих груп і комп' ютерів в мережі
TreeView1.Items.Clear;
for i:=0 to N.Count - 1 do begin
  WorkgroupNode:=TreeView1.Items.Add(nil, N[i]);
  WorkgroupNode.ImageIndex:=1;
  WorkgroupNode.SelectedIndex:=1;
  for j:=0 to (N.Objects[i] as TStrings).Count - 1 do begin
    ComputerNode:=TreeView1.Items.AddChild(WorkgroupNode, (N.Objects[i] as
TStrings).Strings[j]);
    ComputerNode.ImageIndex:=0;
  end;
end;

// Отримання IP адрес комп' ютерів
GetIPAddresses(N, Memo2.Lines);

finally
  N.Free;
end;
TreeView1.FullExpand;

finally
  Screen.Cursor:=crDefault;
end;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  Edit1.Text:=GetComputerName;

  if LoadIpHlp then
  begin
    DOIpStuff;
    Timer1.Enabled := true;
  end;

end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin

  Timer1.Enabled := false;
  DoIPStuff;
  Timer1.Enabled := true;

end;

procedure TForm1.DOIpStuff;
begin
  Get_TCPTable( MemoX.Lines );
  Get_UDPTable( MemoTR.Lines );

end;

procedure TForm1.Button2Click(Sender: TObject);
begin

Form2.Show;

end;

```

```
procedure TForm1.Button3Click(Sender: TObject);  
begin  
  
Form1.Close;  
  
end;  
  
procedure TForm1.SpeedButton2Click(Sender: TObject);  
begin  
  
Form1.Close;  
  
end;  
  
procedure TForm1.SpeedButton4Click(Sender: TObject);  
begin  
  
Form2.Show;  
  
end;  
  
end.
```

Кафедра КБПЗ – 2021 рік

Файл IPtraff.pas - відслідковування та динамічний перерозподіл трафіку мережі
програмно визначаемого ЦОД на базі технологій Fujitsu

```

unit IPtraff;

interface

uses
  Windows, Messages, SysUtils, Classes, Dialogs, IpHlpApi;

const
  NULL_IP      = ' 0. 0. 0. 0' ;

type
  TWellKnownPort = record
    Prt: DWORD;
    Srv: string[20];
  end;

const
  WellKnownPorts: array[1..32] of TWellKnownPort
  = (
  //   ( Prt: 0; Srv:  ' RESRVED' ),      {Зарезервовано}
    ( Prt: 7; Srv:  ' ECHO   ' ),      {Пінгування }
    ( Prt: 9; Srv:  ' DISCARD' ),
    ( Prt: 13; Srv: ' DAYTIME' ),
    ( Prt: 17; Srv: ' QOTD   ' ),      {Показник на день}
    ( Prt: 19; Srv: ' CHARGEN' ),      {Генератор символів}
    ( Prt: 20; Srv: ' FTPDATA' ),      { Протокол File Transfer Protocol -
дані}
    ( Prt: 21; Srv: ' FTPCTRL' ),      { Протокол File Transfer Protocol -
управління}
    ( Prt: 22; Srv: ' SSH    ' ),
    ( Prt: 23; Srv: ' TELNET ' ),
    ( Prt: 25; Srv: ' SMTP   ' ),      { Протокол Simple Mail Transfer
Protocol}
    ( Prt: 37; Srv: ' TIME   ' ),      { Протокол Time Protocol }
    ( Prt: 43; Srv: ' WHOIS  ' ),      { WHO IS сервіс }
    ( Prt: 53; Srv: ' DNS    ' ),      { Domain Name Service }
    ( Prt: 67; Srv: ' BOOTPS ' ),      { BOOTP сервер }
    ( Prt: 68; Srv: ' BOOTPC ' ),      { BOOTP клієнт }
    ( Prt: 69; Srv: ' TFTP   ' ),      { Стандартний FTP }
    ( Prt: 70; Srv: ' GOPHER ' ),      { Протокол Gopher }
    ( Prt: 79; Srv: ' FINGER ' ),      { Протокол Finger }
    ( Prt: 80; Srv: ' HTTP   ' ),      { Протокол HTTP }
    ( Prt: 88; Srv: ' KERBROS' ),      { Протокол Kerberos }
    ( Prt: 109; Srv: ' POP2   ' ),      { Протокол Post Office Protocol Version
2 }
    ( Prt: 110; Srv: ' POP3   ' ),      { Протокол Post Office Protocol Version
3 }
    ( Prt: 111; Srv: ' SUN_RPC' ),      { SUN віддалений виклик функцій }
    ( Prt: 119; Srv: ' NNTP   ' ),      { Протокол Network News Transfer
Protocol }
    ( Prt: 123; Srv: ' NTP    ' ),      { Протокол Network Time protocol
}
    ( Prt: 135; Srv: ' DCOMRPC' ),      { Локальний сервіс }
    ( Prt: 137; Srv: ' NBNAME ' ),      { NETBIOS сервіс імен }
    ( Prt: 138; Srv: ' NBDGRAM' ),      { NETBIOS сервіс датаграм }
    ( Prt: 139; Srv: ' NBSESS ' ),      { NETBIOS сервіс сесій }
    ( Prt: 143; Srv: ' IMAP   ' ),      { Протокол Internet Message Access
Protocol }
    ( Prt: 161; Srv: ' SNMP   ' ),      { Протокол Simple Netw. Management
Protocol }
    ( Prt: 169; Srv: ' SEND   ' )
  );

```

```

const
ICMP_ERROR_BASE = 11000;
IcmpErr : array[1..22] of string =
(
  ' IP_BUFFER_TOO_SMALL' , ' IP_DEST_NET_UNREACHABLE' , '
IP_DEST_HOST_UNREACHABLE' ,
  ' IP_PROTOCOL_UNREACHABLE' , ' IP_DEST_PORT_UNREACHABLE' , ' IP_NO_RESOURCES'
,
  ' IP_BAD_OPTION' , ' IP_HARDWARE_ERROR' , ' IP_PACKET_TOO_BIG' , '
IP_REQUEST_TIMED_OUT' ,
  ' IP_BAD_REQUEST' , ' IP_BAD_ROUTE' , ' IP_TTL_EXPIRED_TRANSIT' ,
  ' IP_TTL_EXPIRED_REASSEM' , ' IP_PARAMETER_PROBLEM' , ' IP_SOURCE_QUENCH' ,
  ' IP_OPTION_TOO_BIG' , ' IP_BAD_DESTINATION' , ' IP_ADDRESS_DELETED' ,
  ' IP_SPEC_MTU_CHANGE' , ' IP_MTU_CHANGE' , ' IP_UNLOAD'
);

ARPEntryType : array[1..4] of string = ( ' Інший' , ' Несправний' ,
  ' Динамічний' , ' Статичний'
);
TCPConnState :
array[1..12] of string =
( ' CLOSED' , ' READ' , ' SYN_SENT' ,
  ' SYN_RCVD' , ' ESTABLISHED' , ' FIN_WAIT1' ,
  ' FIN_WAIT2' , ' WAIT' , ' CLOSING' ,
  ' LAST_ACK' , ' WAIT' , ' DELETE_TCB' );

TCPToAlgo : array[1..4] of string =
( ' Const.Timeout' , ' MIL-STD-1778' ,
  ' Van Jacobson' , ' Other' );

IPForwTypes : array[1..4] of string =
( ' other' , ' invalid' , ' local' , ' remote' );

IPForwProtos : array[1..18] of string =
( ' OTHER' , ' LOCAL' , ' NETMGMT' , ' ICMP' , ' EGP' ,
  ' GGP' , ' HELO' , ' RIP' , ' IS_IS' , ' ES_IS' ,
  ' CISCO' , ' BBN' , ' OSPF' , ' BGP' , ' BOOTP' ,
  ' AUTO_STAT' , ' STATIC' , ' NOT_DOD' );

type
// для IpHlpNetworkParams
TNetworkParams = record
  HostName: string ;
  DomainName: string ;
  CurrentDnsServer: string ;
  DnsServerTot: integer ;
  DnsServerNames: array [0..9] of string ;
  NodeType: UINT;
  ScopeID: string ;
  EnableRouting: UINT;
  EnableProxy: UINT;
  EnableDNS: UINT;
end;

TIfRows = array of TMibIfRow ; // динамічний масив колонок

// для IpHlpAdaptersInfo
TAdaptorInfo = record
  AdapterName: string ;
  Description: string ;
  MacAddress: string ;
  Index: DWORD;
  aType: UINT;
  DHCPEnabled: UINT;
  CurrIPAddress: string ;
  CurrIPMask: string ;

```

```

IPAddressTot: integer ;
IPAddressList: array of string ;
IPMaskList: array of string ;
GatewayTot: integer ;
GatewayList: array of string ;
DHCPtot: integer ;
DHCPserver: array of string ;
HaveWINS: BOOL;
PrimWINSTot: integer ;
PrimWINSserver: array of string ;
SecWINSTot: integer ;
SecWINSserver: array of string ;
LeaseObtained: LongInt ;
LeaseExpires: LongInt;
end ;

TAdaptorRows = array of TAdaptorInfo ;

```

```

function IpHlpAdaptersInfo(var AdpTot: integer;var AdpRows: TAdaptorRows):
integer ;
procedure Get_AdaptersInfo( List: TStrings );
function IpHlpNetworkParams (var NetworkParams: TNetworkParams): integer ;
procedure Get_NetworkParams( List: TStrings );
procedure Get_ARPTable( List: TStrings );
procedure Get_TCPTable( List: TStrings );
procedure Get_TCPStatistics( List: TStrings );
function IpHlpTCPStatistics (var TCPStats: TMibTCPStats): integer ;
procedure Get_UDPTable( List: TStrings );
procedure Get_UDPStatistics( List: TStrings );
function IpHlpUdpStatistics (UdpStats: TMibUDPStats): integer ;
procedure Get_IPAddrTable( List: TStrings );
procedure Get_IPForwardTable( List: TStrings );
procedure Get_IPStatistics( List: TStrings );
function IpHlpIPStatistics (var IPStats: TMibIPStats): integer ;
function Get_RTTAndHopCount( IPAddr: DWORD; MaxHops: Longint;
var RTT: longint; var HopCount: longint ): integer;
procedure Get_ICMPStats( ICMPIn, ICMPOut: TStrings );
function IpHlpIfTable(var IfTot: integer; var IfRows: TIfRows): integer ;
procedure Get_IfTable( List: TStrings );
function IpHlpIfEntry(Index: integer; var IfRow: TMibIfRow): integer ;
procedure Get_RecentDestIPs( List: TStrings );

```

```

// утілити перетворення
function MacAddr2Str( MacAddr: TMacAddress; size: integer ): string;
function IpAddr2Str( IPAddr: DWORD ): string;
function Str2IpAddr( IPStr: string ): DWORD;
function Port2Str( nwoPort: DWORD ): string;
function Port2Wrd( nwoPort: DWORD ): DWORD;
function Port2Svc( Port: DWORD ): string;
function ICMPErr2Str( ICMPErrCode: DWORD) : string;

```

```
implementation
```

```
var
RecentIPs      : TStringList;
```

```
{ перетворення токена у рядок, потім рядок знищується }
function NextToken( var s: string; Separator: char ): string;
var
Sep_Pos      : byte;
begin
Result := ` ` ;
if length( s ) > 0 then begin
Sep_Pos := pos( Separator, s );
if Sep_Pos > 0 then begin

```

```

        Result := copy( s, 1, Pred( Sep_Pos ) );
        Delete( s, 1, Sep_Pos );
    end
else begin
    Result := s;
    s := ' ';
end;
end;
end;

{ перетворення MAC-адреси до ww-xx-yy-zz рядка }
function MacAddr2Str( MacAddr: TMacAddress; size: integer ): string;
var
    i          : integer;
begin
    if Size = 0 then
        begin
            Result := '00-00-00-00-00-00' ;
            EXIT;
        end
    else Result := ' ';
        //
        for i := 1 to Size do
            Result := Result + IntToHex( MacAddr[i], 2 ) + '-';
            Delete( Result, Length( Result ), 1 );
        end;
end;

{ перетворення IP-адреси в мережний байт типу DWORD }
function IpAddr2Str( IPAddr: DWORD ): string;
var
    i          : integer;
begin
    Result := ' ';
    for i := 1 to 4 do
        begin
            Result := Result + Format( '%3d.' , [IPAddr and $FF] );
            IPAddr := IPAddr shr 8;
        end;
        Delete( Result, Length( Result ), 1 );
    end;
end;

{ перетворення крапкову десяткову IP-адресу в мережний байт типу DWORD}
function Str2IpAddr( IPStr: string ): DWORD;
var
    i          : integer;
    Num       : DWORD;
begin
    Result := 0;
    for i := 1 to 4 do
        try
            Num := ( StrToInt( NextToken( IPStr, '.' ) ) ) shl 24;
            Result := ( Result shr 8 ) or Num;
        except
            Result := 0;
        end;
    end;
end;

{ перетворення номер порту в мережний байт типу DWORD }
function Port2Wrd( nwoPort: DWORD ): DWORD;
begin
    Result := Swap( WORD( nwoPort ) );
end;

```



```

if result <> ERROR_SUCCESS then exit ;
NetworkParams.DnsServerTot := 0 ;
with FixedInfo^ do
begin
  NetworkParams.HostName := trim (HostName) ;
  NetworkParams.DomainName := trim (DomainName) ;
  NetworkParams.ScopeId := trim (ScopeID) ;
  NetworkParams.NodeType := NodeType ;
  NetworkParams.EnableRouting := EnableRouting ;
  NetworkParams.EnableProxy := EnableProxy ;
  NetworkParams.EnableDNS := EnableDNS ;
  NetworkParams.DnsServerNames [0] := DNSServerList.IPAddress ; //
  if NetworkParams.DnsServerNames [0] <> ' ' then
    NetworkParams.DnsServerTot := 1 ;
  PDnsServer := DnsServerList.Next;
  while PDnsServer <> Nil do
  begin
    NetworkParams.DnsServerNames [NetworkParams.DnsServerTot] :=
      PDnsServer^.IPAddress ; //
    inc (NetworkParams.DnsServerTot) ;
    if NetworkParams.DnsServerTot >=
      Length (NetworkParams.DnsServerNames) then exit ;
    PDnsServer := PDnsServer.Next ;
  end;
end ;
finally
  FreeMem (FixedInfo) ; //
end ;
end;

//-----

function ICMPErr2Str( ICMPErrCode: DWORD) : string;
begin
  Result := ' UnknownError : ' + IntToStr( ICMPErrCode ) ;
  dec( ICMPErrCode, ICMP_ERROR_BASE );
  if ICMPErrCode in [Low(ICMPerr)..High(ICMPerr)] then
    Result := ICMPerr[ ICMPErrCode];
end;

//-----

// включаємо байти вводу/виводу для кожного адаптеру

function IpHlpIfTable(var IfTot: integer; var IfRows: TIfRows): integer ;
var
  I,
  TableSize : integer;
  pBuf, pNext : PChar;
begin
  result := ERROR_NOT_SUPPORTED ;
  if NOT LoadIpHlp then exit ;
  SetLength (IfRows, 0) ;
  IfTot := 0 ; //
  TableSize := 0;
  // перший виклик: беремо необхідний розмір пам' яті
  result := GetIfTable (Nil, @TableSize, false) ; //
  if result <> ERROR_INSUFFICIENT_BUFFER then exit ;
  GetMem( pBuf, TableSize ) ;
  try
    FillChar (pBuf^, TableSize, #0); // очищуємо буфер, з W98 не потрібно

  // беремо показчик на таблицю
  result := GetIfTable (PTMibIfTable (pBuf), @TableSize, false) ;
  if result <> NO_ERROR then exit ;
  IfTot := PTMibIfTable (pBuf)^.dwNumEntries ;
  if IfTot = 0 then exit ;
  SetLength (IfRows, IfTot) ;

```

```

    pNext := pBuf + SizeOf(IfTot) ;
    for i := 0 to Pred (IfTot) do
    begin
        IfRows [i] := PTMibIfRow (pNext )^ ;
        inc (pNext, SizeOf (TMibIfRow)) ;
    end;
finally
    FreeMem (pBuf) ;
end ;
end;

procedure Get_IfTable( List: TStrings );
var
    IfRows      : TIfRows ;
    Error, I     : integer;
    NumEntries  : integer;
    sDescr, sIfName: string ;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    SetLength (IfRows, 0) ;
    Error := IpHlpIfTable (NumEntries, IfRows) ;
    if (Error <> 0) then
        List.Add( SysErrorMessage( GetLastError ) )
    else if NumEntries = 0 then
        List.Add( ' Даних немає.' )
    else
        begin
            for I := 0 to Pred (NumEntries) do
            begin
                with IfRows [I] do
                begin
                    if wszName [1] = #0 then
                        sIfName := ' '
                    else
                        sIfName := WideCharToString (@wszName) ; // перетворюємо
Unicode y string
                        sIfName := trim (sIfName) ;
                        sDescr := bDescr ;
                        sDescr := trim (sDescr);
                        List.Add (Format (
                            '%0.8x - %3d - %16s - %8d - %12d - %2d - %2d - %10d - %10d -
%-s - %-s' ,
                            [dwIndex, dwType, MacAddr2Str( TMacAddress( bPhysAddr ),
dwPhysAddrLen ), dwMTU, dwSpeed, dwAdminStatus,
dwOPerStatus, Int64 (dwInOctets), Int64 (dwOutOctets), //
counters are 32-bit
                            sIfName, sDescr] ) // , додаємо введення/вивід
                            );
                end;
            end ;
        end ;
        SetLength (IfRows, 0) ; // звільняємо пам' ять
    end ;

function IpHlpIfEntry(Index: integer; var IfRow: TMibIfRow): integer ;
begin
    result := ERROR_NOT_SUPPORTED ;
    if NOT LoadIpHlp then exit ;
    FillChar (IfRow, SizeOf (TMibIfRow), #0); // очищуємо буфер, з W98 не
потрібно
    IfRow.dwIndex := Index ;
    result := GetIfEntry (@IfRow) ;
end ;

//-----
{ інформація про мережні адаптери }

```

```

function IpHlpAdaptersInfo(var AdpTot: integer; var AdpRows: TAdaptorRows):
integer ;
var
  BufLen      : DWORD;
  AdapterInfo : PTIP_ADAPTER_INFO;
  PIPAddr     : PTIP_ADDR_STRING;
  PBuf        : PCHAR ;
  I           : integer ;
begin
  SetLength (AdpRows, 4) ;
  AdpTot := 0 ;
  BufLen := 0 ;
  result := GetAdaptersInfo( Nil, @BufLen );
  if (result <> ERROR_INSUFFICIENT_BUFFER) and (result = NO_ERROR) then exit ;
  GetMem( pBuf, BufLen );
  try
    FillChar (pBuf^, BufLen, #0); // очищуємо буфер
    result := GetAdaptersInfo( PTIP_ADAPTER_INFO (PBuf), @BufLen );
    if result = NO_ERROR then
      begin
        AdapterInfo := PTIP_ADAPTER_INFO (PBuf) ;
        while ( AdapterInfo <> nil ) do
          begin
            AdpRows [AdpTot].IPAddressTot := 0 ;
            SetLength (AdpRows [AdpTot].IPAddressList, 2) ;
            SetLength (AdpRows [AdpTot].IPMaskList, 2) ;
            AdpRows [AdpTot].GatewayTot := 0 ;
            SetLength (AdpRows [AdpTot].GatewayList, 2) ;
            AdpRows [AdpTot].DHCPTot := 0 ;
            SetLength (AdpRows [AdpTot].DHCPSTotal, 2) ;
            AdpRows [AdpTot].PrimWINSTot := 0 ;
            SetLength (AdpRows [AdpTot].PrimWINSServer, 2) ;
            AdpRows [AdpTot].SecWINSTot := 0 ;
            SetLength (AdpRows [AdpTot].SecWINSServer, 2) ;
            AdpRows [AdpTot].CurrIPAddress := NULL_IP;
            AdpRows [AdpTot].CurrIPMask := NULL_IP;
            AdpRows [AdpTot].AdapterName := Trim( string(
AdapterInfo^.AdapterName ) );
            AdpRows [AdpTot].Description := Trim( string(
AdapterInfo^.Description ) );
            AdpRows [AdpTot].MacAddress := MacAddr2Str( TMacAddress(
AdapterInfo^.Address ),
AdapterInfo^.AddressLength ) ;
            AdpRows [AdpTot].Index := AdapterInfo^.Index ;
            AdpRows [AdpTot].aType := AdapterInfo^.aType ;
            AdpRows [AdpTot].DHCPEnabled := AdapterInfo^.DHCPEnabled ;
            if AdapterInfo^.CurrentIPAddress <> Nil then
              begin
                AdpRows [AdpTot].CurrIPAddress :=
AdapterInfo^.CurrentIPAddress.IPAddress ;
                AdpRows [AdpTot].CurrIPMask :=
AdapterInfo^.CurrentIPAddress.IPMask ;
              end ;

            // беремо список IP адрес та масок для IPAddressList
            I := 0 ;
            PIPAddr := @AdapterInfo^.IPAddressList ;
            while (PIPAddr <> Nil) do
              begin
                AdpRows [AdpTot].IPAddressList [I] := PIPAddr.IPAddress ;
                AdpRows [AdpTot].IPMaskList [I] := PIPAddr.IPMask ;
                PIPAddr := PIPAddr.Next ;
                inc (I) ;
                if Length (AdpRows [AdpTot].IPAddressList) <= I then
                  begin
                    SetLength (AdpRows [AdpTot].IPAddressList, I * 2) ;
                    SetLength (AdpRows [AdpTot].IPMaskList, I * 2) ;
                  end ;
              end ;
            end ;
          end ;
        end ;
      end ;
    end ;
  end ;

```

```

AdpRows [AdpTot].IPAddressTot := I ;

// беремо список IP адрес для GatewayList
I := 0 ;
PIpAddr := @AdapterInfo^.GatewayList ;
while (PIpAddr <> Nil) do
begin
  AdpRows [AdpTot].GatewayList [I] := PIpAddr.IPAddress ;
  PIpAddr := PIpAddr.Next ;
  inc (I) ;
  if Length (AdpRows [AdpTot].GatewayList) <= I then
    SetLength (AdpRows [AdpTot].GatewayList, I * 2) ;
end ;
AdpRows [AdpTot].GatewayTot := I ;

// беремо список IP адрес для GatewayList
I := 0 ;
PIpAddr := @AdapterInfo^.DHCPSTotal ;
while (PIpAddr <> Nil) do
begin
  AdpRows [AdpTot].DHCPSTotal [I] := PIpAddr.IPAddress ;
  PIpAddr := PIpAddr.Next ;
  inc (I) ;
  if Length (AdpRows [AdpTot].DHCPSTotal) <= I then
    SetLength (AdpRows [AdpTot].DHCPSTotal, I * 2) ;
end ;
AdpRows [AdpTot].DHCPSTotal := I ;

// беремо список IP адрес для PrimaryWINSServer
I := 0 ;
PIpAddr := @AdapterInfo^.PrimaryWINSServer ;
while (PIpAddr <> Nil) do
begin
  AdpRows [AdpTot].PrimWINSServer [I] := PIpAddr.IPAddress ;
  PIpAddr := PIpAddr.Next ;
  inc (I) ;
  if Length (AdpRows [AdpTot].PrimWINSServer) <= I then
    SetLength (AdpRows [AdpTot].PrimWINSServer, I * 2) ;
end ;
AdpRows [AdpTot].PrimWINSTotal := I ;

// беремо список IP адрес для SecondaryWINSServer
I := 0 ;
PIpAddr := @AdapterInfo^.SecondaryWINSServer ;
while (PIpAddr <> Nil) do
begin
  AdpRows [AdpTot].SecWINSServer [I] := PIpAddr.IPAddress ;
  PIpAddr := PIpAddr.Next ;
  inc (I) ;
  if Length (AdpRows [AdpTot].SecWINSServer) <= I then
    SetLength (AdpRows [AdpTot].SecWINSServer, I * 2) ;
end ;
AdpRows [AdpTot].SecWINSTotal := I ;

AdpRows [AdpTot].LeaseObtained := AdapterInfo^.LeaseObtained ;
AdpRows [AdpTot].LeaseExpires := AdapterInfo^.LeaseExpires ;

inc (AdpTot) ;
if Length (AdpRows) <= AdpTot then
  SetLength (AdpRows, AdpTot * 2) ; // more memory
AdapterInfo := AdapterInfo^.Next ;
end ;
SetLength (AdpRows, AdpTot) ;
end ;
finally
  FreeMem( pBuf ) ;
end ;
end ;

```

```

procedure Get_AdaptersInfo( List: TStrings );
var
  AdpTot: integer;
  AdpRows: TAdaptorRows ;
  Error: DWORD ;
  I: integer ;
  //J: integer ; jpt
  //S: string ; id.
begin
  if not Assigned( List ) then EXIT;
  List.Clear;
  SetLength (AdpRows, 0) ;
  AdpTot := 0 ;
  Error := IpHlpAdaptersInfo(AdpTot, AdpRows) ;
  if (Error <> 0) then
    List.Add( SysErrorMessage( GetLastError ) )
  else if AdpTot = 0 then
    List.Add( ' Даних немає.' )
  else
    begin
      for I := 0 to Pred (AdpTot) do
        begin
          with AdpRows [I] do
            begin
              //List.Add(AdapterName + ' -' + Description ); // jpt : не корисне
              List.Add( Format( ' %8.8x - %6s - %16s - %2d - %16s - %16s - %16s' ,
                [Index, AdaptTypes[aType], MacAddress, DHCPEnabled,
                  GatewayList [0], DHCPSTServer [0], PrimWINSSServer [0]] ) );
              {if IPAddressTot <> 0 then // jpt : not useful
                begin
                  S := ' ' ;
                  for J := 0 to Pred (IPAddressTot) do
                    S := S + IPAddressList [J] + ' /' + IPMaskList [J] + '
- ' ;
                  List.Add(IntToStr (IPAddressTot) + ' IP Adresse(s): ' + S);
                end ;
                List.Add( ' ' ); }
            end ;
          end ;
        end ;
      SetLength (AdpRows, 0) ;
    end ;

//-----
{ зчитуємо кількість раундів, та час звертання до IP }
function Get_RTTAndHopCount( IPAddr: DWORD; MaxHops: Longint; var RTT: Longint;
  var HopCount: Longint ): integer;
begin
  if not GetRTTAndHopCount( IPAddr, @HopCount, MaxHops, @RTT ) then
    begin
      Result := GetLastError;
      RTT := -1; //
      HopCount := -1;
    end
  else
    Result := NO_ERROR;
  end;

//-----
{ ARP-таблиця - список відношень між віддаленими IP та віддаленими MAC-адресами.
}
procedure Get_ARPTable( List: TStrings );
var
  IPNetRow : TMibIPNetRow;
  TableSize : DWORD;
  NumEntries : DWORD;
  ErrorCode : DWORD;

```

```

i          : integer;
pBuf      : PChar;
begin
if not Assigned( List ) then EXIT;
List.Clear;
// перший виклик: беремо довжину таблиці
TableSize := 0;
ErrorCode := GetIPNetTable( Nil, @TableSize, false ); //
//
if ErrorCode = ERROR_NO_DATA then
begin
List.Add( ' ARP-кеш пустий.' );
EXIT;
end;
// беремо таблицю
GetMem( pBuf, TableSize );
NumEntries := 0 ;
try
ErrorCode := GetIpNetTable( PTMIBIPNetTable( pBuf ), @TableSize, false );
if ErrorCode = NO_ERROR then
begin
NumEntries := PTMIBIPNetTable( pBuf )^.dwNumEntries;
if NumEntries > 0 then //.
begin
inc( pBuf, SizeOf( DWORD ) ); // беремо розмір останньої таблиці
for i := 1 to NumEntries do
begin
IPNetRow := PTMIBIPNetRow( PBuf )^;
with IPNetRow do
List.Add( Format( ' %8x - %12s - %16s - %10s' ,
[dwIndex, MacAddr2Str( bPhysAddr, dwPhysAddrLen ),
IPAddr2Str( dwAddr ), ARPEntityType[dwType]
]));
inc( pBuf, SizeOf( IPNetRow ) );
end;
end
else
List.Add( ' ARP-кеш пустий.' );
end
else
List.Add( SysErrorMessage( ErrorCode ) );

// we _must_ restore pointer!
finally
dec( pBuf, SizeOf( DWORD ) + NumEntries * SizeOf( IPNetRow ) );
FreeMem( pBuf );
end ;
end;

//-----
procedure Get_TCPTable( List: TStrings );
var
TCPRow      : TMIBTCPRow;
i,
NumEntries  : integer;
TableSize   : DWORD;
ErrorCode   : DWORD;
DestIP      : string;
pBuf        : PChar;
begin
if not Assigned( List ) then EXIT;
List.Clear;
RecentIPs.Clear;
// перший виклик : беремо розмір таблиці
TableSize := 0;
NumEntries := 0 ;
ErrorCode := GetTCPTable( Nil, @TableSize, false ); //
if Errorcode <> ERROR_INSUFFICIENT_BUFFER then

```

```

EXIT;

// беремо розмір пам' яти, який потрібно та здійснюємо виклик знову
GetMem( pBuf, TableSize );
// беремо таблицю
ErrorCode := GetTCPTable( PTMIBTCPTable( pBuf ), @TableSize, false );
if ErrorCode = NO_ERROR then
begin
    NumEntries := PTMIBTCPTable( pBuf )^.dwNumEntries;
    if NumEntries > 0 then
    begin
        inc( pBuf, SizeOf( DWORD ) ); // беремо розмір останньої таблиці
        for i := 1 to NumEntries do
        begin
            TCPRow := PTMIBTCPRow( pBuf )^; // беремо наступний запис
            with TCPRow do
            begin
                if dwRemoteAddr = 0 then
                    dwRemotePort := 0;
                DestIP := IPAddr2Str( dwRemoteAddr );
                List.Add(
                    Format( ' %15s : %-7s -> %15s : %-7s -> %-16s' ,
                        [IpAddr2Str( dwLocalAddr ),
                          Port2Svc( Port2Wrd( dwLocalPort ) ),
                          DestIP,
                          Port2Svc( Port2Wrd( dwRemotePort ) ),
                          TCPConnState[dwState]
                        ] ) );
                //
                if (not ( dwRemoteAddr = 0 ))
                    and ( RecentIps.IndexOf( DestIP ) = -1 ) then
                    RecentIps.Add( DestIP );
            end;
            inc( pBuf, SizeOf( TMIBTCPRow ) );
        end;
    end;
end
else
    List.Add( SysErrorMessage( ErrorCode ) );
dec( pBuf, SizeOf( DWORD ) + NumEntries * SizeOf( TMibTCPRow ) );
FreeMem( pBuf );
end;

//-----
procedure Get_TCPStatistics( List: TStrings );
var
    TCPStats : TMibTCPStats;
    ErrorCode : DWORD;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    if NOT LoadIpHlp then exit ;
    ErrorCode := GetTCPStatistics( @TCPStats );
    if ErrorCode = NO_ERROR then
        with TCPStats do
            begin
                List.Add( ' Алгоритм повторної передачі: ' + TCPToAlgo[dwRTOAlgorithm] );
                List.Add( ' Мінімальний час виведення : ' + IntToStr( dwRTOMin )
                    + ' ms' );
                List.Add( ' Максимальний час виведення : ' + IntToStr( dwRTOMax )
                    + ' ms' );
                List.Add( ' Максимальне число підключень : ' + IntToStr( dwRTOAlgorithm )
                    );
                List.Add( ' Активні підключення : ' + IntToStr( dwActiveOpens
                    ) );
                List.Add( ' Пасивні підключення : ' + IntToStr( dwPassiveOpens
                    ) );
            end;
        end;
end;

```

```

        List.Add( ' Невдала спроба відкриття      : ' + IntToStr( dwAttemptFails )
);
        List.Add( ' Відновлений зв'язок      : ' + IntToStr( dwEstabResets ) );
        List.Add( ' Поточний зв'язок .: ' + IntToStr( dwCurrEstab ) );
        List.Add( ' Отримано сегментів      : ' + IntToStr( dwInSegs ) );
        List.Add( ' Відправлено сегментів    : ' + IntToStr( dwOutSegs )
);
        List.Add( ' Ретрансльовано сегментів  : ' + IntToStr( dwReTransSegs ) );
        List.Add( ' Помилки входження      : ' + IntToStr( dwInErrs ) );
        List.Add( ' Скидання виведення     : ' + IntToStr( dwOutRsts ) );
        List.Add( ' Сукупні зв'язки      : ' + IntToStr( dwNumConns ) );
    end
    else
        List.Add( SysErrorMessage( ErrorCode ) );
    end;

function IpHlpTCPStatistics (var TCPStats: TMibTCPStats): integer ;
begin
    result := ERROR_NOT_SUPPORTED ;
    if NOT LoadIpHlp then exit ;
    result := GetTCPStatistics( @TCPStats );
end;

//-----
procedure Get_UDPTable( List: TStrings );
var
    UDPRow      : TMIBUDPRow;
    i,
    NumEntries  : integer;
    TableSize   : DWORD;
    ErrorCode   : DWORD;
    pBuf        : PChar;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;

    // перший виклик : беремо розмір таблиці
    TableSize := 0;
    NumEntries := 0 ;
    ErrorCode := GetUDPTable( Nil, @TableSize, false );
    if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
        EXIT;

    // беремо потрібний розмір пам'яті, викликаємо знову
    GetMem( pBuf, TableSize );

    // беремо таблицю
    ErrorCode := GetUDPTable( PTMIBUDPTable( pBuf ), @TableSize, false );
    if ErrorCode = NO_ERROR then
        begin
            NumEntries := PTMIBUDPTable( pBuf )^.dwNumEntries;
            if NumEntries > 0 then
                begin
                    inc( pBuf, SizeOf( DWORD ) ); // беремо розмір останньої таблиці
                    for i := 1 to NumEntries do
                        begin
                            UDPRow := PTMIBUDPRow( pBuf )^; // беремо наступний запис
                            with UDPRow do
                                List.Add( Format( ' %15s : %-6s' ,
                                    [IpAddr2Str( dwLocalAddr ),
                                    Port2Svc( Port2Wrd( dwLocalPort ) )
                                    ] ) );
                                inc( pBuf, SizeOf( TMIBUDPRow ) );
                            end;
                        end
                    else
                        List.Add( ' Даних немає.' );
                    end
                end
            else
                end
        end
    else
        end
end
else
end

```

```

    List.Add( SysErrorMessage( ErrorCode ) );
    dec( pBuf, SizeOf( DWORD ) + NumEntries * SizeOf( TMibUDPRow ) );
    FreeMem( pBuf );
end;

//-----
procedure Get_IPAddrTable( List: TStrings );
var
    IPAddrRow      : TMibIPAddrRow;
    TableSize      : DWORD;
    ErrorCode       : DWORD;
    i               : integer;
    pBuf            : PChar;
    NumEntries      : DWORD;
begin
    if not Assigned( List ) then EXIT;
    List.Clear;
    TableSize := 0; ;
    NumEntries := 0 ;
    // перший виклик: беремо довжину таблиці
    ErrorCode := GetIpAddrTable( Nil, @TableSize, true ); //
    if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
        EXIT;

    GetMem( pBuf, TableSize );
    // беремо таблицю
    ErrorCode := GetIpAddrTable( PTMibIPAddrTable( pBuf ), @TableSize, true );
    if ErrorCode = NO_ERROR then
        begin
            NumEntries := PTMibIPAddrTable( pBuf )^.dwNumEntries;
            if NumEntries > 0 then
                begin
                    inc( pBuf, SizeOf( DWORD ) );
                    for i := 1 to NumEntries do
                        begin
                            IPAddrRow := PTMIBIPAddrRow( pBuf )^;
                            with IPAddrRow do
                                List.Add( Format( '%8.8x | %15s | %15s | %15s | %8.8d' ,
                                    [dwIndex,
                                    IPAddr2Str( dwAddr ),
                                    IPAddr2Str( dwMask ),
                                    IPAddr2Str( dwBCastAddr ),
                                    dwReasmSize
                                    ] ) );
                                inc( pBuf, SizeOf( TMIBIPAddrRow ) );
                            end;
                        end
                    else
                        List.Add( ' Даних немає.' );
                    end
                else
                    List.Add( SysErrorMessage( ErrorCode ) );
                // we must restore pointer!
                dec( pBuf, SizeOf( DWORD ) + NumEntries * SizeOf( IPAddrRow ) );
                FreeMem( pBuf );
            end;

            //-----
            { беремо дані з таблиці маршрутизатора Cisco }
            procedure Get_IPForwardTable( List: TStrings );
            var
                IPForwRow      : TMibIPForwardRow;
                TableSize      : DWORD;
                ErrorCode       : DWORD;
                i               : integer;
                pBuf            : PChar;
                NumEntries      : DWORD;
            begin

```

```

if not Assigned( List ) then EXIT;
List.Clear;
TableSize := 0;

// перший виклик: беремо довжину таблиці
NumEntries := 0 ;
ErrorCode := GetIpForwardTable( Nil, @TableSize, true);
if ErrorCode <> ERROR_INSUFFICIENT_BUFFER then
    EXIT;

// беремо таблицю
GetMem( pBuf, TableSize );
ErrorCode := GetIpForwardTable( PTMibIPForwardTable( pBuf ), @TableSize,
true);
if ErrorCode = NO_ERROR then
begin
    NumEntries := PTMibIPForwardTable( pBuf )^.dwNumEntries;
    if NumEntries > 0 then
    begin
        inc( pBuf, SizeOf( DWORD ) );
        for i := 1 to NumEntries do
        begin
            IPForwRow := PTMibIPForwardRow( pBuf )^;
            with IPForwRow do
            begin
                if (dwForwardType < 1)
                or (dwForwardType > 4) then
                    dwForwardType := 1 ; // , враховуємо погані значення
                List.Add( Format(
                    '%15s | %15s | %15s | %8.8x | %7s | %5.5d | %7s | %2.2d' ,
                    [IPAddr2Str( dwForwardDest ),
                    IPAddr2Str( dwForwardMask ),
                    IPAddr2Str( dwForwardNextHop ),
                    dwForwardIFIndex,
                    IPForwTypes[dwForwardType],
                    dwForwardNextHopAS,
                    IPForwProtos[dwForwardProto],
                    dwForwardMetric1
                    ] ) );
            end ;
            inc( pBuf, SizeOf( TMibIPForwardRow ) );
        end;
    end
else
    List.Add( ' Даних немає.' );
end
else
    List.Add( SysErrorMessage( ErrorCode ) );
dec( pBuf, SizeOf( DWORD ) + NumEntries * SizeOf( TMibIPForwardRow ) );
FreeMem( pBuf );
end;

//-----
procedure Get_IPStatistics( List: TStrings );
var
    IPStats      : TMibIPStats;
    ErrorCode    : integer;
begin
    if not Assigned( List ) then EXIT;
    if NOT LoadIpHlp then exit ;
    ErrorCode := GetIPStatistics( @IPStats );
    if ErrorCode = NO_ERROR then
    begin
        List.Clear;
        with IPStats do
        begin
            if dwForwarding = 1 then
                List.add( ' Розблоковане пересилання : ' + ' Так' )

```

```

else
    List.add( ` Розблоковане пересилання          : ` + ` Hi' );
List.add( ` Вбудований TTL                        : ` + inttostr( dwDefaultTTL ) );
List.add( ` Отримані датаграми                   : ` + inttostr( dwInReceives ) );
List.add( ` Помилка заголовку (в)                : ` + inttostr( dwInHdrErrors ) );
List.add( ` Адреса помилкова (в)                 : ` + inttostr( dwInAddrErrors ) );
List.add( ` Датаграма переслана                   : ` + inttostr( dwForwDatagrams ) );
//
List.add( ` Невідомий протокол (в)               : ` + inttostr( dwInUnknownProtos )
);
List.add( ` Датаграма відвергнута                 : ` + inttostr( dwInDiscards ) );
List.add( ` Датаграма встановлена                 : ` + inttostr( dwInDelivers ) );
List.add( ` Запит виведення                       : ` + inttostr( dwOutRequests ) );
List.add( ` Маршрутизація в маршрутизаторі Cisco відвергнута : ` +
inttostr( dwRoutingDiscards ) );
List.add( ` Немає маршрутів (з)                   : ` + inttostr( dwOutNoRoutes )
);
List.add( ` Час виведення перебору               : ` + inttostr( dwReasmTimeout )
);
List.add( ` Запити перебору                       : ` + inttostr( dwReasmReqds ) );
List.add( ` Перебори здійснено                   : ` + inttostr( dwReasmOKs ) );
List.add( ` Помилка перебору                     : ` + inttostr( dwReasmFails ) );
List.add( ` Фрагментація мережі успішна          : ` + inttostr( dwFragOKs ) );
List.add( ` Помилка фрагментації                 : ` + inttostr( dwFragFails ) );
List.add( ` Датаграми фрагментовано             : ` + inttostr( dwFragCreates ) );
List.add( ` Кількість інтерфейсів                : ` + inttostr( dwNumIf ) );
List.add( ` Кількість IP-адрес                   : ` + inttostr( dwNumAddr ) );
List.add( ` маршрут в таблиці маршрутизації Cisco : ` + inttostr(
dwNumRoutes ) );
end;
end
else
List.Add( SysErrorMessage( ErrorCode ) );
end;

function IpHlpIPStatistics (var IPStats: TMibIPStats): integer ; //
begin
result := ERROR_NOT_SUPPORTED ;
if NOT LoadIpHlp then exit ;
result := GetIPStatistics( @IPStats );
end ;

//-----
procedure Get_UdpStatistics( List: TStrings );
var
    UdpStats      : TMibUDPStats;
    ErrorCode     : integer;
begin
if not Assigned( List ) then EXIT;
ErrorCode := GetUDPStatistics( @UdpStats );
if ErrorCode = NO_ERROR then
begin
List.Clear;
with UdpStats do
begin
List.add( ` Datagrams (в)          : ` + inttostr( dwInDatagrams ) );
List.add( ` Datagrams (з)          : ` + inttostr( dwOutDatagrams ) );
List.add( ` No Ports                : ` + inttostr( dwNoPorts ) );
List.add( ` Errors (в)              : ` + inttostr( dwInErrors ) );
List.add( ` UDP Listen Ports       : ` + inttostr( dwNumAddrs ) );
end;
end
else
List.Add( SysErrorMessage( ErrorCode ) );
end;

```



```
//-----  
procedure Get_RecentDestIPs( List: TStrings );  
begin  
  if Assigned( List ) then  
    List.Assign( RecentIPs )  
end;  
  
initialization  
  
  RecentIPs := TStringList.Create;  
  
finalization  
  
  RecentIPs.Free;  
  
end.
```

Кафедра_КБПЗ_2021 рік

Файл NetConst.pas - ініціалізація констант

```

unit NetConst;

interface

resourcestring

  SShellLinkReadError = ' Помилка читання' ;
  SShellLinkWriteError = ' Помилка запису' ;
  SShellLinkLoadError = ' Не можу завантажити %s' ;
  SShellLinkSaveError = ' Не можу зберегти %s' ;
  SShellLinkCreateError = ' Не можу ініціалізувати інтерфейс' ;
  SDynArrayIndexError = ' У масиві %s немає елемента з таким індексом (%d)' ;
  SDynArrayCountError = ' Перевищена довжина масиву (%d)' ;
  SSharedMemoryError = ' Не можу створити файл' ;
  SCannotInitTimer = ' Не можу ініціалізувати таймер' ;
  SPrinterIndexError = ' Принтер не доступний (%d)' ;
  SIndicesOutOfRange = ' Недопустимий індекс матриці [%d: %d]' ;
  SRowIndexOutOfRange = ' Недопустиме значення рядка матриці (%d)' ;
  SColIndexOutOfRange = ' Недопустиме значення стовбчика матриці (%d)' ;
  SNoAdminRights = ' У Вас немає прав адміністратора' ;

  SFileError = ' Помилка %s файл %s%s' ;
  SFileReading = ' читання' ;
  SFileWriting = ' запис' ;

  SFileError002 = ' - файл не знайдено' ;
  SFileError003 = ' - шлях не знайдено' ;
  SFileError004 = ' - не можу відкрити файл' ;
  SFileError005 = ' - немає доступу' ;
  SFileError014 = ' - не достатньо пам"яті' ;
  SFileError015 = ' - не можу знайти драйвер' ;
  SFileError017 = ' - не можу перемістити файл' ;
  SFileError019 = ' - носій захищений від запису' ;
  SFileError020 = ' - не можу знайти пристрій' ;
  SFileError021 = ' - пристрій не відкривається для читання' ;
  SFileError022 = ' - пристрій не може розпізнати команду' ;
  SFileError025 = ' - вказана область не знайдена' ;
  SFileError026 = ' - пристрій недоступний' ;
  SFileError027 = ' - сектор не знайдено' ;
  SFileError029 = ' - помилка запису на пристрій' ;
  SFileError030 = ' - помилка читання з пристроєм' ;
  SFileError032 = ' - файл використовується іншою програмою' ;
  SFileError036 = ' - занадто багато відкритих файлів' ;
  SFileError038 = ' - досягнутий кінець файлу' ;
  SFileError039 = ' - диск переповнений' ;
  SFileError050 = ' - запит не підтримується' ;
  SFileError051 = ' - віддалений комп'ютер недоступний' ;
  SFileError052 = ' - у мережі знайдені ідентичні імена' ;
  SFileError053 = ' - мережний шлях не знайдено' ;
  SFileError054 = ' - мережа зайнята' ;
  SFileError055 = ' - ресурс мережі або пристрій недоступний' ;
  SFileError057 = ' - апаратна помилка в мережному адаптері' ;
  SFileError058 = ' - сервер не в змозі виконувати дану дію' ;
  SFileError059 = ' - помилка у мережі' ;
  SFileError064 = ' - недоступне мережне ім"я' ;
  SFileError065 = ' - немає доступу до мережі' ;
  SFileError066 = ' - невірно вказаний тип мережного ресурсу' ;
  SFileError067 = ' - не знайдено вказане мережне ім"я' ;
  SFileError070 = ' - відключений сервер мережі' ;
  SFileError082 = ' - не можу створити файл чи каталог' ;
  SFileError112 = ' - не достатньо вільного місця на диску' ;
  SFileError123 = ' - в імені файлу вказано недопустимий символ' ;
  SFileError161 = ' - неправильно вказано шлях' ;

```

```
SFileError183 = ` - файл не існує' ;
```

```
SCannotSetSize = ` Не можу змінити розмір файлу' ;
```

```
SUnableToCompress = ` Не можу заархівувати дані' ;
```

```
SUnableToDecompress = ` Не можу розархівувати дані' ;
```

```
SCannotFindNetwork = ` Не можу знайти мережу' ;
```

```
implementation
```

```
end.
```

Кафедра_КБПЗ_2021_рік

Файл About.pas - довідка

```
unit About;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls, jpeg, ExtCtrls;  
  
type  
  TForm2 = class(TForm)  
    Image1: TImage;  
    Memo1: TMemo;  
    Button1: TButton;  
    procedure Button1Click(Sender: TObject);  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
  
var  
  Form2: TForm2;  
  
implementation  
  
{ $R *.dfm }  
  
procedure TForm2.Button1Click(Sender: TObject);  
begin  
  Form2.Close;  
end;  
  
end.
```