

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи кібербезпеки для
відеоспостереження на базі бездротових мереж”

Виконав здобувач вищої освіти
IV курсу, групи КБ-20-3СК
ОПП «Кібербезпека»
спеціальності 125 «Кібербезпека»
_____ Чирва М.В.
« ____ » _____ 2023 р.

Керівник проекту
кандидат технічних наук
_____ Смірнова Т.В.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 125 “Кібербезпека”
Освітньо-професійна (освітньо-наукова) програма “Кібербезпека”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 17 » січня 2023 року

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

Чирві Максиму Віталійовичу

(прізвище, ім'я, по батькові)

- Тема роботи Програмне забезпечення системи кібербезпеки для відеоспостереження на базі бездротових мереж
- Керівник роботи Смірнова Тетяна Віталіївна, канд. техн. наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 13-02 від 5.01.2023 року
- Строк подання студентом роботи до захисту 23.05.2023 р.
- Мета та завдання випускної кваліфікаційної роботи: Метою роботи є розробка програмного забезпечення системи кібербезпеки для відеоспостереження на базі бездротових мереж
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - Призначення та область використання.
 - Перегляд аналогічних існуючих систем.
 - Опис і обґрунтування проектних рішень.
 - Етапи програмування системи.
 - Впровадження системи кібербезпеки в промислову експлуатацію.
 - Висновки
- Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

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

7. Дата видачі завдання « 17 » січня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2023 р.	
3.	Розробка моделі компонента	20.03.2023 р.	
4.	Розробка структур даних	25.03.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2023 р.	
6.	Програмування алгоритмів	10.04.2023 р.	
7.	Оформлення ПЗ	17.04.2023 р.	
8.	Попередній захист роботи	23.05.2023 р.	

Дата видачі завдання
« 17 » січня 2023 р.

Підпис керівника

Смірнова Т.В.
(прізвище та ініціали)

Завдання прийнято до виконання
« 17 » січня 2023 р.

Підпис здобувача

Чирва М.В.
(прізвище та ініціали)

АНОТАЦІЯ

Чирва М.В. Програмне забезпечення системи кібербезпеки для відеоспостереження на базі бездротових мереж. 125 Кібербезпека. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи кібербезпеки для відеоспостереження на базі бездротових мереж.

Метою розробки є програмне забезпечення системи кібербезпеки для відеоспостереження на базі бездротових мереж.

Результат роботи – програмна реалізація системи кібербезпеки для відеоспостереження на базі бездротових мереж.

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

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

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

Програму розроблено в середовищі Builder C++.

Ключові слова: кібербезпека, відеоспостереження

ABSTRACT

Chyrva M.V. Cyber security system software for video surveillance based on wireless networks. 125 Cyber security. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the first (bachelor) level of higher education, software is developed, which is intended for a cyber security system for video surveillance based on wireless networks.

The purpose of the development is the software of the cyber security system for video surveillance based on wireless networks.

The result of the work is a software implementation of a cyber security system for video surveillance based on wireless networks.

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

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

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

The program was developed in the Builder C++ environment.

Keywords: cyber security, video surveillance

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування.....	25
2.3 Розгорнута постановка завдання	27
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	28
3.1 Опис функціонування системи	28
3.2 Розробка структурної схеми.....	38
3.3 Розробка функціональної схеми	57
3.4 Розробка діаграми процесів.....	66
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	68
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	68
4.2 Захист розробленого програмного забезпечення.....	80
5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	83
6 ОСНОВНІ ВИСНОВКИ.....	93
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	95

ВКРБ-125.23.0039.00.00.ПЗ

Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.		Чирва М.В.			Програмне забезпечення системи кібербезпеки для відеоспостереження на базі бездротових мереж	Літ.	Аркуш	Аркушів
Перев.		Смірнова Т.В.				Б	1	101
Н.контр.		Гермак В.С.			ЦНТУ КБ-20-3СК			
Затв.		Смірнов О.А.						

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БПД	–	бездротова передача даних
ПЗ	–	програмне забезпечення
СПД	–	системи передачі даних
ACK	–	повідомлення підтвердження прийому
ARQ	–	протокол повторної передачі даних
BPSK	–	Binary phase-shift keying
FFD	–	повнофункціональний пристрій
GFSK	–	Gaussian frequency-shift keying
MAC	–	шар механізму доступу
NACK	–	повідомлення непідтвердження прийому
OSI	–	мережна модель
P2P	–	однорангові мережі
PAN	–	персональна мережа
PPS	–	Portable Protocol Stack
RFD	–	пристрій з полегшеними функціями
TDMA	–	часовий поділ
Wi-Fi	–	бездротова технологія

ВСТУП

Актуальність теми. Сучасні системи обробки візуальної інформації можуть містити в собі канали бездротової передачі даних. До таких систем, зокрема, відносяться охоронні комплекси відеоспостереження.

Більшість подібних систем побудовано на базі технології IEEE 802.11. Розвиток сучасних бездротових технологій дозволяє передавати по радіоканалу значні обсяги інформації на швидкостях, порівнянних з передачею по провідних каналах зв'язку. У той же час нескладно показати низьке використання пропускної здатності реальних бездротових каналів зв'язку. Технологічно вдосконалюється фізичний рівень: міняються способи модуляції, удосконалюються способи обробки сигналів і збільшуються бітова швидкість передачі й чутливість приймача.

Розвиток методів канального кодування трохи відстає від розвитку методів кодування фізичного рівня. Більше того, при подальшому збільшенні швидкості передачі протокол керування доступом до середовища стає вузьким місцем всієї системи, не дозволяючи одержати відчутний приріст пропускної здатності навіть при застосуванні самих високоефективних технологій фізичного рівня. Приміром, максимальний теоретичний відсоток використання пропускної здатності каналу для стандарту IEEE 802.11n з бітовою швидкістю 108 Мбіт/с становить 51 %.

Підвищити даний показник можна, застосувавши принцип системного підходу до проектування системи обробки візуальної інформації. Найбільшу складність у цьому випадку представляє проектування канального рівня. При проектуванні системи кодування канального рівня необхідно вирішити завдання оптимального проектування на вибір методу кодування і його параметрів.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи кібербезпеки для відеоспостереження на базі бездротових мереж.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

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

- Огляд існуючих систем для відеоспостереження на базі бездротових мереж.
- Дослідження системи кібербезпеки для відеоспостереження на базі бездротових мереж.
- Програмна реалізація системи кібербезпеки для відеоспостереження на базі бездротових мереж.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі для відеоспостереження на базі бездротових мереж.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для відеоспостереження на базі бездротових мереж, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Система призначена для реалізації кібербезпеки для відеоспостереження на базі бездротових мереж. Термін бездротовий (також бездротовий) використовується відносно будь-якого виду електричної або електронної дії, що виконується без використання дротового зв'язку.

Словосполучення «бездротове середовище» може ввести в оману, оскільки означає повну відсутність проводів в мережі. В більшості випадків це не зовсім так. Звичайно бездротові компоненти взаємодіють з мережею, в якій як середовище передачі використовується кабель. Така мережа із змішаними компонентами називається гібридною.

Бездротовий зв'язок – це передача інформації на відстань без використання електричних провідників або «дротів». Ця відстань може бути як малою (декілька метрів, як у телевізійному дистанційному керуванні), так і дуже великою (тисячі або навіть мільйони кілометрів для телекомунікацій). Бездротовий зв'язок як правило розглядається як галузь телекомунікацій.

Термін бездротова технологія в загальному використовується для обладнання мобільних інформаційних технологій. До їх складу входять мобільні телефони, надолонники (PDA) та бездротові мережі. Інші приклади бездротових технологій включають пристрої глобальної системи позиціонування, пристрої дистанційного відкривання гаража, бездротові комп'ютерні миші та клавіатури, супутникове телебачення і мобільні і радіо телефони.

Можливості

Ідея бездротового середовища вельми приваблива, оскільки її компоненти:

- забезпечують часове підключення до кабельної мережі;
- допомагають організувати резервне копіювання в кабельну мережу;

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

- гарантують певний рівень мобільності;
- дозволяють зняти обмеження на максимальну протяжність мережі, що накладаються мідними або навіть оптоволоконними кабелями.

1.2 Область застосування

Областю застосування є системи відеоспостереження. Системи відеоспостереження та відеоконтролю є самим затребуваним інструментом в роботі охоронної служби.

Найбільш примітивна система відеоспостереження складається з камери спостереження і записуючого відеосигнали пристрою.

Система відеоспостереження – це сукупність оптико-електронного обладнання та супутніх пристроїв, які дозволяють здійснювати запис, перегляд, відеоконтроль і візуальний аналіз зображень.

Сучасні системи відеоспостереження і відеореєстрації розвиваються в ногу з новітніми технологіями і вже зараз роблять можливими автоматичну обробку та аналіз картинок, так можуть виконувати автоматичне розпізнавання облич людей, номерів з пропусків і номерних знаків машин.

Головна мета функціонування системи відеоспостереження полягає в забезпеченні постійного візуального контролю об'єкту, що охороняється або території.

В залежності від способу передачі сигналу виділяють такі типи систем відеоспостереження:

- аналогові системи відеоспостереження;
- цифрові системи відеоспостереження;
- гібридні системи відеоспостереження.

У гібридних або комбінованих системах відеоспостереження встановлюються аналогові камери з пристроями для подальшої оцифровки відеоінформації.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Ефективна система відеоспостереження на вашому об'єкті замінює кількох робітників служби охорони та безпеки.

Більшість сучасних цифрових систем відеоспостереження складаються з ір-камер, які дозволяють переглядати процес спостереження через Інтернет.

Так для такої системи немає обмежень за способом перегляду запису – ви можете користуватися пультом охорони з стаціонарними комп'ютерами, домашніми ноутбуками, портативними комунікаторами, планшетами та мобільними смартфонами. Також стає можливим віддалений доступ через Інтернет до налаштувань і управління системи відеоспостереження.

Контролювати все, що відбувається на вашому об'єкті в режимі реального часу вам дозволить система відеоспостереження через інтернет.

Спроекувати бездротову систему відеоспостереження на базі ІР-камер дозволяють такі технології бездротової передачі сигналу:

- Wi-Fi;
- 3G;
- GSM;
- CDMA.

В даний час не існує обмежень по кількості відеокамер для таких систем, а також установка відеоспостереження можлива як на нерухомих, так і на рухомих об'єктах.

Слід зазначити, що в сучасних комплексних системах безпеки пріоритетною складовою є система відеоспостереження. Так крім візуального контролю система дає можливість програмувати дії всієї охоронної системи в різних екстрених ситуаціях на об'єкті.

Система відеоспостереження може встановлюватися на програмному та апаратному рівнях.

Клієнти використовують сучасні охоронні системи відеоспостереження для забезпечення безпеки на самих різних об'єктах: будинки, квартири, заміської дачі, офісу, заводу або великих територій автовокзалів і аеропортів і т.д.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

Також варто зазначити, що існують автономні системи відеоспостереження, які дозволяють відмовитися від трансляції сигналу на точку-приймач, а використовувати власний накопичувач відеоінформації. Така система дозволяє вирішувати проблему з відсутністю сигналу або мережі на будь-якій території або ж з непостійністю сигналу під час пересування автомобіля.

Стрімкий розвиток відеоспостереження зробив можливою появу прихованого відеоспостереження. Так система прихованого відеоспостереження складаються з мінівідеокамер. Такі мініатюрні камери найчастіше встановлюються в різних елементах інтер'єру. Розміри мікровідеокамери кілька міліметрів.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки для відеоспостереження на базі бездротових мереж, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

ORIENT W802G4

Почну, з добре знайомого мені комплекту ORIENT W802G4. Це не новинка, проте, мені здається вона дуже цікавою для тих з нас, хто підбирає недороге рішення для установки у квартирі, під'їзді або дачному будиночку. Насправді, даний комплект є лише частиною системи відеоспостереження. У комплект входять чотири бездротові камери в антивандальному всепогодному корпусі, один приймач, набір з п'яти мережних джерел живлення й два композитних кабелі. У найпростішому випадку процес складання системи відеоспостереження зводиться до двох простих кроків. Ви розставляєте камери в потрібних місцях і підключаєте приймач до будь-якого телевізора. Не чи правда, просто?! При необхідності ви можете ускладнити систему відеоспостереження й інтегрувати комплект ORIENT W802G4 в існуючу систему безпеки, підключивши її до відеореєстратора.

Давайте ближче познайомимося з особливостями камер, що входять у комплект. Як було відзначено вище, кожна камера виконана в антивандальному всепогодному корпусі, установленому на спеціальному кронштейні, що дозволяє вибрати зручний ракурс зйомки. Кронштейн може кріпитися на стіні або стелі, а також може виконувати функцію підставки для установки камери на будь-якій горизонтальній поверхні.

Кожна камера обладнана кольоровою ПЗС матрицею, розміром 1/3" і розрішенням 380 горизонтальних ТВ ліній, що відповідає дозволу 628x582 у системі PAL або 510x492 в NTSC. Використовуваний об'єктив має кут огляду 45?.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Це так званий універсальний кут, що забезпечує досить гарний огляд навіть у невеликих приміщеннях.

Якість картинки дуже пристойне. Висока деталізація й відмінний фокус, як при зйомці близько розташованих об'єктів, так і віддалених, роблять ці камери вдалим вибором для рішення широкого спектра завдань по відеоспостереженню.

На фронтальній стороні камери встановлений потужний ІЧ-прожектор, що включає 30 світлодіодів. Цього досить для нормального висвітлення сектора на відстані до 15 метрів, але при цьому не створюється засвітлення, що заважає нормальній зйомці близько розташованих об'єктів. Включення ІЧ-підсвічування відбувається в автоматичному режимі. Для цього використовується фотоелемент.

Ще однією особливістю камер можна назвати убудований мікрофон, що дозволяє вести запис відео зі звуком.

Для підключення камер можна використовувати як бездротове, так і провідне підключення. Кожна камера обладнана роз'ємом живлення, відео й аудіо виходами.

Для бездротового підключення використовується мініатюрний чотириканальний приймач. На лицьовій стороні приймача розташований індикатор, що відображає номер каналу. Управляти приймачем можна як за допомогою пульта дистанційного керування, так і за допомогою кнопки на верхній стороні корпусу. Користувач може вибрати режим відображення одного із чотирьох каналів або циклічне відображення чотирьох каналів. Режим квадратора (відображення чотирьох каналів на одному екрані) дана модель приймача не підтримує.

На тильній стороні корпусу приймача розташовано: роз'єм для підключення блока живлення, антена й дві пари композитних виходів (відео+звук). Для чого необхідні два виходи? Наприклад, для одночасного підключення приймача до відеореєстратора й до телевізора.

В ORIENT W802G4 використовується аналогова бездротова технологія передачі зображення в діапазоні 2.400-2.483Ггц. Недоліком цієї технології є

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

залежність від розташованих рядом джерел, що працюють у тім же діапазоні. Мова йде про бездротові телефони, Wi-Fi точки доступу й інших побутових пристроїв. Всі ці прилади створюють перешкоди при прийманні інформації з камер, при цьому самі камери можуть заблокувати роботу Wi-Fi у будинку. Звертаю увагу, що перешкоди можуть створювати не тільки прилади, що працюють у вашій квартирі, але й у квартирах сусідів. У цьому змісті, бездротове підключення більш актуально для замиського будинку, де можна розраховувати на куди більший радіус дії. Відповідно до специфікації, на відкритому просторі радіус дії може досягати 100 метрів.

Якщо умови у квартирі й будинку не дозволяють нормально використовувати бездротове підключення, то завжди можна скористатися провідним підключенням. У цьому випадку, камери можна прямо підключити до окремого каналу відеореєстратора й вести незалежний запис всіх чотирьох каналів, що помітно розширює можливості системи відеоспостереження.

ORIENT W203D4 Digital

На відміну від ORIENT W802G4 в ORIENT W203D4 Digital використовується цифрова технологія передачі сигналу, що забезпечує ідеальну якість картинки на відстані до 200 метрів.

Цифрова технологія бездротового підключення не єдине, що відрізняє комплект ORIENT W203D4 Digital. Це не просто набір бездротових камер, це повноцінна 4-х канална система відеоспостереження для комп'ютера. Причому, на відміну від звичних комп'ютерних систем відеоспостереження, що вимагають установку багатоканальної плати відеозахвата, у цьому випадку використовується маленький USB приймач, що дозволяє використовувати таке рішення не тільки з настільними комп'ютерами, але й з ноутбуками або неттопами. У комплект входять чотири камери, виконані в пластиковому корпусі. Камери відрізняються компактним розміром, що в сполученні з білим кольором корпуса, робить їх непомітними в інтер'єрі. Важливо помітити, що вага камери становить менш 60 грам. Це дозволяє кріпити її навіть на легких конструкціях.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Камери обладнані об'єктивом з фіксованою фокусною відстанню й CMOS сенсором, розміром 1/4". Сенсор має розрішення 380 ТВЛ, що дозволяє одержати картинку 628x582 пікселів у системі PAL або 510x492 пікселів у системі NTSC. ІЧ-підсвічування в даних камерах не передбачені. Кут огляду використовуваного об'єктива невеликий, і не дуже підходить для використання в маленьких квадратних приміщеннях, однак, завдяки конструкції камери, ви завжди можете замінити використовуваній об'єктив на іншій. Тут використовуються стандартні об'єктиви для мікрокамер M12. Наприклад, сюди можна без проблем установити об'єктив "риб'яче око" з кутом огляду 140?.

На тильній стороні є: роз'єм антени, кнопка CODE для прив'язки бездротової камери, світлодіодний індикатор, що відображає стан живлення камери й USB порт, використовуваний для підключення зовнішнього джерела живлення.

У підставці корпусу розташовані убудований мікрофон і кріплення для кронштейна. Використовуваний кронштейн також виконаний з білого пластику й дозволяє закріпити камеру на стіні або стелі, і вільно вибирати ракурс, з якого буде вестися зйомка.

Бездротовий приймач зовні нагадує велику флешку. Більша ширина й товщина корпусу приймача, можуть викликати проблему з підключенням USB пристроїв до сусідніх USB портів. Якщо це критично для вас, то можна скористатися USB перехідником або USB хабом.

На корпусі приймача розташовані: знімна поворотна антена й один світлодіодний індикатор. Якщо чесно, я сподівався побачити на корпусі роз'єм з аналоговими відеовиходами, що дозволило б використовувати дану бездротову систему в парі з відеореєстраторами, але нічого схожого тут ні, що ще раз підтверджує те, що ORIENT W203D4 Digital чисто комп'ютерна система відеоспостереження.

Для створення системи відеоспостереження досить розставити камери, підключити приймач до USB порту комп'ютера й установити програмне

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

забезпечення. Процес установки й налаштування програмного забезпечення займає лічені хвилини. Софт прекрасно працює як під Windows XP, так і під Windows 7. Однією з умов нормальної роботи є підтримка Media Player версії 9 або вище. У принципі, для роботи системи відеоспостереження не потрібно робити яких-небудь налаштувань, все працює, що називається “прямо з коробки”. Проте, у деяких випадках меню налаштування параметрів дозволять зробити систему відеоспостереження більше зручною й ефективною.

Меню налаштування включає два розділи. Розділ «System» включає набір загальних параметрів, серед яких мова інтерфейсу (росіянин не підтримується), параметри поштового сервера для оповіщення про тривожну ситуацію, а так само папки для зберігання відеозаписів і фотографій. Тут же можна настроїти параметри убудованого Web сервера й MMS сервера для віддаленого доступу до системи відеоспостереження.

Другий розділ, включаючи ряд додаткових параметрів, серед яких я виділив можливість додавання штампів часу й дати, накладення водяного знака, налаштування відеопараметрів, поворот зображення, а також налаштування дозволу і якості відеозапису, області дії детектора руху й оповіщення.

Програмне забезпечення включає базовий набір функцій, властивих сучасним системам безпеки, включаючи можливість перегляду одного або чотирьох каналів на одному екрані, виявлення руху по кожному каналі окремо, запис відео, зняття фотозображення й, що мені особливо сподобалося, доступ через Інтернет, що дозволяє віддалено контролювати систему відеоспостереження, перебуваючи в будь-якій точці миру, де є Інтернет.

Вікно програми має фіксований розмір, що з однієї сторони дуже зручно, особливо, при паралельній роботі з іншими додатками. Для додаткової зручності є можливість згортання програми в компактне вікно з відображенням картинки з однієї або чотирьох камер. У компактному режимі користувач може не тільки бачити відеозображення з різних камер, але й швидко робити скриншоти зображення.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

зображення цього каналу. Якщо обрано режим відображення чотирьох каналів, то й запис буде включати зображення всіх чотирьох каналів. Можливо, це не самий звичний спосіб подання інформації, але для системи відеоспостереження даного класу, він здається більше доступним і зрозумілим.

Якщо ви хочете мати зручний механізм для пошуку записів, то рекомендую скористатися інструментом перегляду, інтегрованим у додаток відеоспостереження. Тут всі дуже зрозуміло. Користувач задає часовий інтервал, одну або кілька камер для відображення, і натискає кнопку пошуку.

Вікно перегляду включає зручний інструмент для швидкого перегляду на швидкостях до 16x, а також дозволяє швидко переміщатися по відеофайлі. При необхідності можна перейти в папку з відеофайлом або зробити скриншот.

Одне з важливих питань, можливість підключення до системи відеоспостереження через інтернет. Для цього необхідно настроїти доступ з інтернет по порту 7788. Якщо ваш комп'ютер прямо підключений до інтернет, і при цьому ваш провайдер видає вам публічний IP адреса, то для перегляду камер досить просто ввести свій IP адреса. Якщо ж доступ в інтернет здійснюється через домашній роутер, то необхідно настроїти віртуальний сервер на порт 7788. Приємно помітити, що дане рішення відмінно працює під різними браузерами, включаючи Safari під OS X. Функціонально, користувач може тільки переглядати відео в тому форматі, що обраний в основному додатку. Це звичайно не дуже добре. Було б краще, якби користувач міг вільно вибирати канал для віддаленого перегляду, примусово запускати запис і здійснювати пошук і перегляд записів на віддаленому сервері. Але якщо врахувати, що дана система ставиться до систем початкового рівня, і її головною особливістю є простота й доступність для рядового споживача, те вже те, що в ній реалізоване, здається мені більш ніж достатнім.

Безумовно, простота реалізації бездротової системи відеоспостереження вкрай важлива, особливо коли мова йде про масового користувача, що не має спеціальних знань по налаштуванню й експлуатації традиційних систем, проте,

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

особисто мене більше цікавить питання, наскільки реально обіцяна висока якість картинки й наскільки сильно бездротове підключення впливає на Wi-Fi, установлений удома.

У моїх тестових умовах картинку можна назвати ідеальною. Ніяких перешкод і втрат зображення. Таке відчуття, що камери підключені по проведенню. Причому, висока якість картинки зберігається навіть при значному віддаленні камери від приймача. Що ж стосується впливу на Wi-Fi, то тут ніяких проблем не виявлено. Я запускав спеціальні тести пропускну здатності Wi-Fi каналу, і можу із упевненістю говорити, що навіть при включенні всіх чотирьох камер, швидкість і якість передачі файлів по Wi-Fi не міняються. У цьому змісті ORIENT W203D4 Digital ідеальна для використання в багатоквартирних будинках, причому, зовсім не обов'язково встановлювати камери й приймач в одній кімнаті. Приймач прекрасно приймає сигнал з камер, розташованих через несучу стіну, товщиною 20 см.

Деяких з потенційних покупців хвилює питання: наскільки безпечно таке бездротове підключення? Чи не стане моя квартира надбанням всіх сусідів, у яких може бути такий же приймач? Відповідь: все дуже безпечно. Сигнал передається в зашифрованому виді. Кожна камера прив'язана до певного приймача й не може бути видна на іншому приймачі.

Чи є недоліки в ORIENT W203D4 Digital? У цілому, явно виражених проблем я не виявив. Апаратна частина працює як годинник. Були думки, що софт може підвести, але й тут усе виявилось стабільно. Звичайно, розроблювачам програмного забезпечення ще є над чим попрацювати і як розширити функціонал комплекту, проте, уже те, що вміє ORIENT W203D4 Digital сьогодні, здається мені більш ніж достатнім для користувачів, що підбирають простий набір для швидкого розгортання системи відеоспостереження.

ORIENT W240D1 Digital

У комплекті тільки одна камера й один приймач, проте, це одне із самих універсальних рішень для відеоспостереження.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

На перший погляд ORIENT W240D1 Digital, саме так називається новинка, дуже схожа на цифрову відеоняньку, але з цього не виходить, що використовувати її можна тільки для спостереження за дитиною або людьми похилого віку. На мій погляд, ORIENT W240D1 Digital може стати відмінним доповненням для вже наявної системи відеоспостереження, а також може розглядатися як єдине рішення для швидкого розгортання відеоспостереження в невеликому офісі, магазині або кафе.

Чим же так цікава ORIENT W240D1 Digital? По-перше, тут використовується цифрова технологія передачі відео й аудіо сигналу, що забезпечує бездоганну якість картинки й звуку на відстані до 200 метрів. По-друге, компактний розмір, мала вага камери й приймача, повна автономність, що дозволяє харчувати камеру й приймач від убудованого акумулятора, і, нарешті, можливість підключення до чотирьох камер, а також наявність композитного аудіо-відео виходу, що дозволяє підключити пристрій до зовнішнього ресивера для реалізації функцій запису й віддаленого доступу, дозволяють розглядати ORIENT W240D1 Digital як універсальне рішення для відеоспостереження, за допомогою якого можна вирішувати широкий спектр завдань.

Знайомство з ORIENT W240D1 Digital почнемо з камери, що виконана в трохи незвичайному дизайні. З однієї сторони така незвичайна форма й сполучення кольорів нагадують дитячу іграшку, що дозволяє використовувати її в дитячій кімнаті, не побоюючись, що дитина злякається класичної камери, що стежить за нею зі стелі. З іншого боку, я не можу назвати обраний дизайн дитячим. Так що ви можете самостійно вибирати, де вам використовувати даний комплект: удома або в офісі.

Незважаючи на масивну підставку, камера відрізняється компактними розмірами, а її вага з убудованим акумулятором становить усього 130 грам. Конструкція камери дозволяє встановлювати її на будь-якій горизонтальній поверхні, а при необхідності, можна закріпити її на штативі або струбцини за допомогою кріплення, розташованого в підставці камери.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Для вибору зручного ракурсу зйомки в конструкції передбачений поворотний механізм, що дозволяє повернути камеру по горизонталі й змінити кут нахилу. Камера обладнана CMOS сенсором розміром 1/4" с розрішенням 380 ТВ ліній. Це еквівалентно дозволу 628x582 пікселів у системі PAL або 510x492 в NTSC. Об'єктив камери має фіксована фокусна відстань і кут огляду близько 50°, що забезпечує гарний огляд навіть у невеликих приміщеннях. Для зйомки в темряві на лицьовій стороні камери розташовані 9 інфрачервоних світлодіодів, що дозволяють вести зйомку в повній темряві на відстані до 10 метрів. У нижній частині корпусу розташовані два світлодіода, що відображають стан камери й зарядку убудованого акумулятора. Тут же розташований мікрофон.

У тильній частині корпусу розташований убудований динамік, роз'єм для підключення антени, роз'єм для підключення зовнішнього живлення й кнопка включення. Вона використовується для прив'язки камери до приймача.

Тепер розглянемо конструктивні особливості приймача, що має досить компактні розміри й вага близько 150 грам. Компактні розміри приймача вкрай важливі для повсякденного використання. У конструкції приймача передбачена підставка, що дозволяє розташувати приймач на столі або прикроватній тумбочці, а також є "вухко" для ремінця на зап'ястя.

На лицьовій стороні приймача розташований екран, розміром 2.4" і розрішенням 480x240 пікселів. Під екраном розташовані дві кнопки, відповідальні за включення живлення приймача й включення режиму двостороннього голосового зв'язку. Тут же є чотири світлодіодних індикатори, що відображають гучність дитячого плачу. Нижче розташована панель керування, за допомогою якої користувач може управляти основними параметрами системи відеоспостереження.

На бічній грані корпусу розташовані убудований мікрофон і композитний аудіо-відео вихід, якому можна використовувати для підключення приймача до великого телевізора або відеореєстратора для запису й організації віддаленого доступу.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

З іншої сторони розташовані роз'єм для підключення навушників (входять у комплект), зовнішнього джерела живлення, індикатор зарядки убудованого акумулятора й вимикач живлення, що дозволяє заощадити заряд акумулятора.

Розібравшись із конструктивними особливостями ORIENT W240D1 Digital, прийшов час ближче познайомитися з налаштуваннями й реальними можливостями приладу, при використанні як як відеонянька, так і як доповнення до домашнього або офісного відеореєстратора.

У принципі, ніяких особливих настроювань не потрібно. Для того щоб почати використовувати ORIENT W240D1 Digital досить зарядити убудований акумулятор приймача й камери. Камера, що входить у комплект, уже прив'язана до приймача. Проте, у деяких випадках може знадобитися налаштування деяких параметрів. Для цього натискаємо кнопку меню на панелі керування приймачем.

Системне меню включає всього чотири розділи. Розділ «Volume» включає можливість роздільної установки гучності динаміків на камері й на приймачі. У розділі «Display Opt» можна змінити яскравість і кольоровість картинки, включити режим цифрового збільшення й вибрати бажаний телевізійний стандарт. У розділі «Camera» можна виконати прив'язку трьох додаткових камер, вибрати камеру, що буде відображатися за замовчуванням або включити режим послідовного перемикання камер. І, нарешті, остання функція «Default» дозволяє повернути всі параметри в стан за замовчуванням.

У процесі використання ORIENT W240D1 Digital я звернув увагу на деякі особливості. Наприклад, приймач не має убудованого детектора руху. У перший момент це виявилось дивним, але потім стало зрозуміло, що якщо камера буде реагувати на кожний рух сплячої дитини, то батьки збожеволіють. Замість цього, тут використовується детектор звуку, що відображає за допомогою чотирьох світлодіодів гучність звуку. Наприклад, якщо дитина просто закректала, то займеться всього один світлодіод, що змусить батьків просто звернути увагу. Якщо ж дитина заплаче, то всі чотири індикатори змусять батьків вжити заходів.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Важливо відзначити підтримку сплячого режиму. У цьому режимі екран приймача перебуває у виключеному стані, що дозволяє заощаджувати заряд убудованого акумулятора, і не відволікати зайвий раз батьків. Екран включається, як тільки маля заплаче.

Якість картинки досить високо й повністю відповідає якості камер цього рівня. Є невелика проблема з кольором, але для даного класу пристроїв це зовсім неважливо. Куди важливіше, наскільки якісної картинка виходить в умовах повної темряви, і тут убудована ІЧ-підсвічування виявляється як не можна до речі. Зображення виходить досить чітким, при цьому використовувани світлодіоди не дають характерного червоного світла, що може налякати дитини.

Якість бездротового каналу таке ж високе, як і в комплекті ORIENT W203D4 Digital. Я не виявив ніяких проблем з перешкодами або втратою картинки поза залежністю від того, як далеко я віддалявся від камери. Вплив на домашній Wi-Fi не виявлено. Говорячи про можливості комплекту необхідно сказати про час автономної роботи. Однієї зарядки акумулятора, убудованого в камеру, вистачає на 3 години. Час автономної роботи приймача значно вищий й залежить від яскравості екрана й підтримки сплячого режиму.

Тепер поговоримо про можливість інтеграції ORIENT W240D1 Digital в існуючу систему відеоспостереження. Навіщо це потрібно? Найчастіше це використовують для запису поведження найнятої няньки для дитини. У цьому змісті ORIENT W240D1 Digital підходить як не можна краще. Непримітний дизайн, можливість установки в зручному місці без необхідності підключення до мережі, а також можливість вибору зручного ракурсу зйомки й запис зі звуком, роблять цей комплект ідеальним вибором для рішення поставленого завдання. Як підключити ORIENT W240D1 Digital до відеореєстратора? Дуже просто. У комплект входить композитний аудіо-відео кабель, обладнаний двома парами ("тато"+"мама") рознімачь.

Налаштування запису виробляється безпосередньо на відеореєстраторі. Звичайно ви можете вибрати режим постійного запису або запису по події. Як

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

краще зробити, вибір за вами. Я ж відзначу, що деякі моделі відеореєстраторів підтримують можливість перегляду й запису на iOS або Android пристроях. Мені здається, можливість у будь-який момент дістати свій смартфон і подивитися, чим займається ваша дитина або чи вчасно нянька погодувала його або уклала спати, здається мені дуже важливою причиною для установки такого комплекту.

Бездротова система відеоспостереження Danrou KCM-6370DRx4

Бездротова система відеоспостереження для будинку, дачі, офісу, торговельного залу або невеликого складу:

- Комплект складається з монітора-реєстратора 7” і чотирьох бездротових камер.

- Найкраща якість картинки забезпечується, завдяки новітній бездротовій цифровій технології передачі відео.

- Відсутність проводів дозволяє легко й швидко встановити спостереження без наслідків для ремонту навіть новачкові за 15 хвилин.

- Чотири кольорові вуличні камери – 420 ТВЛ, оснащені ІЧ підсвічуванням для зйомки вночі.

- Запис відео на SD карту до 16 ГБ (7 доби по русі) .

- Новітня бездротова технологія дозволяє працювати на частоті 2,4 ГГц, не створюючи перешкоди Wi-Fi, і Bluetooth мережам.

- Завдяки захищеному сигналу досягається велика відстань роботи між монітором і камерою – 300 метрів.

Danrou KCM-6370DRx4 – це бездротова система відеоспостереження, призначена для відеоспостереження в невеликих приміщеннях, офісах, приватних будинках, невеликих складах, авто-мийках, і відкритих територіях. Даний комплект складається із чотирьох бездротових відеокамер і монітора-реєстратора. Особливість системи полягає в тому, що вона швидко монтується й легко налаштовується. Працює система по наступному принципі: камера передає відео по радіоканалі на приймач убудований у монітор. Приймач приймає зображення й виводить його на екран, а при необхідності записує відео на SD карту.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Бездротова технологія передачі відео:

- Завдяки бездротовій технології, Ваші приміщення не постраждають від установки, і збережуть свій зовнішній вигляд.
- Відсутність кабелів дозволяє легко й швидко встановити систему навіть новачкові за лічені хвилини.
- Цифровий спосіб передачі даних забезпечує відмінну якість картини й звуку.
- Завдяки новітній технології бездротова цифрова передача не впливає на якість Wi-Fi, і Bluetooth мереж, а так само інших мереж у частоті 2,4 ГГц.
- Дана бездротова технологія абсолютно безпечна для здоров'я.
- Зашифрований відеосигнал захищений від несанкціонованого доступу, його неможливо перехопити.
- Сигнал на 100% захищений від усіляких перешкод, картинка завжди чітка.
- Система проінформує у випадку втрати сигналу звуком, якщо ви помістите камеру за межу досяжності.
- Швидкість передачі даних становить 2 Мб/сек.
- Система працює на частоті 2,4 ГГц, загальнодоступна й не вимагає ліцензії на використання.
- Завдяки захищеному сигналу досягається велика відстань роботи між монітором і камерою – 300 метрів при прямої видимості.

Бездротова кольорова камера:

- Ергономічний і строгий дизайн камери впише в будь-який інтер'єр або екстер'єр.
- Кольорові матриці Sony розміром 1/4" і CMOS сенсором мають розрішення 420 ТВЛ.
- Конструкцією камер передбачене зовнішнє застосування, вони легко тримають високу вологість і дуже низькі температури українських зим.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

– У камери убудований датчик руху, що забезпечує подвійну перевірку руху: програмну й інфрачервону від датчика.

– Камери оснащені мікрофоном і динаміком, що дозволяє не тільки спостерігати за приміщенням і прослуховувати його, але й відправляти голосові повідомлення на камери.

– Інфрачервоні світлодіоди дозволяють камерам знімати в темний час доби, забезпечуючи пристойну видимість на відстані до 4,5 метрів.

Монітор-реєстратор:

– ЖК-дисплей розміром 7 дюймів забезпечує максимальний розрішення 640x480 пікселів.

– Максимальна швидкість запису 25 кадрів у секунду.

– При записі можна вибирати більше низька якості 320x240 пікселів, якщо необхідно заощаджувати місце на носії інформації.

– Є присутнім можливість цифрового 2-х кратні наближення відео.

– Відео записується на SD карту ємністю до 16 Гб. Це 48 годин безперервного запису, і більше тижня запису по русі.

– Система підтримує до 4-х камер. Монітор покаже картинку відразу із всіх камер у режимі квадратора.

– Якщо камера, або датчик у неї убудований виявляють рух, монітор видає звукове попередження.

– Інтерфейс монітора й меню настроювань інтуїтивно зрозумілі й у них абсолютно неможливо заплутатися, потрібні функції легко й швидко настроюються.

– Можливі режими запису по русі, за розкладом, ручний режим і циклічний запис.

– Присутні й налаштування для комфорту, наприклад картинки заставок у режимі відсутності активності, або каскадний режим відображення камер.

– Монітор попередить Вас, у випадку, якщо рівень сигналу критично низок, або загублений.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

– Наявність A/V виходу дозволяє транслювати відео на телевизор, зовнішній відеореєстратор або професійний монітор.

– При підключенні до телевизора можна вибрати телевізійний стандарт PAL або SECAM.

– Пульт ДУ допоможе на відстані управляти монітором, якщо ви захочете прикріпити його на стіну, або просто розташувати у віддаленому місці.

Висновок

У цьому огляді я розглянув чотири комплекти бездротового відеоспостереження, кожний з яких має свої унікальні особливості, що дозволяють використовувати їх для рішення того або іншого завдання. Наприклад, не можу сказати, що комплект ORIENT W802G4 виявився дуже цікавим з погляду можливості бездротового підключення. У цьому змісті він програє цифровим системам. Але при цьому ніщо не заважає нам використовувати провідне підключення камер, при цьому з погляду ціни цей комплект поза конкуренцією.

Комплект ORIENT W203D4 Digital цікавий не тільки високою якістю картинки й більшим радіусом дії бездротового підключення. Не менш цікавим виявився мініатюрний дизайн використовуваних камер, можливість заміни об'єктива й, звичайно ж, USB приймач, що дозволяє за лічені мінуси зібрати комп'ютерну систему відеоспостереження на базі будь-якого комп'ютера або ноутбука. Дана система відеоспостереження підтримує базовий набір функцій, достатній для рішення широкого спектра завдань. Крім цього, важливо нагадати, що ORIENT W203D4 Digital підтримує можливість віддаленого доступу до системи відеоспостереження через інтернет.

І, нарешті, комплект ORIENT W240D1 Digital. На перший погляд, звичайна цифрова відеонянька. Її функціонал досить продуманий і забезпечує зручний і ефективний спосіб для спостереження за сплячою дитиною. Однак, завдяки можливості підключення до чотирьох бездротових камер і інтеграції в існуючу систему відеоспостереження, даний комплект можна використовувати для рішення більше широкого кола завдань, причому не тільки в домашнім застосуванні, але в невеликому офісі.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Оскільки потрібно розробити просту та легку у користуванні програму, яка б виконувалась під операційною системою Windows, то для її реалізації я обрав Builder C++. Існує велике число бібліотек написаних під Builder C++ , тому це одна з важливих причин вибору мови програмування. Середовище Builder C++ досить просте в користуванні, його вихідний код значно менше по об'єму в порівнянні з Delphi чи деякими іншими програмами такого типу. Досить легко організувати взаємодію між модулями програм, об'єктно-орієнтований підхід дає можливість значно скоротити код програми, а отже і час його виконання.

На заміну старого розробленого набору елементів управління у Builder C++ інтегрована бібліотека візуальних компонентів VCL, представлених на палітрі компонентів. Після переносу на форму методом перетягування (drag-and-drop) компоненти відразу становляться діючими об'єктами вашої програми. Окрім типізованих інтерфейсних елементів Windows (кнопки, смуги прокручування, редагуємі текстові області, прості та комбіновані списки, та інше) у бібліотеку включені елементи підтримки діалогових вікон, обслуговування баз даних та багато іншого. Можливо не тільки модифікувати поведінку існуючих компонентів, але і будувати нові.

Builder C++ підтримує останні розширення стандарту мови C++ та забезпечує швидку компіляцію та складання 32-розрядних програм для Windows. Результуючі програми оптимізовані з точки зору швидкості виконання програм та затрат пам'яті. Зручний відладгоджувальник (з асемблерним вікном, можливістю крокового виконання, завдання точок зупинки, трасування та інше) повністю інтегрований у систему проектування. Дизайнер форм, редактор коду, інспектор об'єктів та інші інструменти зостаються доступними під час виконання програми, саме через це вносити зміни до коду можна прямо у процесі відлагодження.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Дизайнер форм, Інспектор об'єктів і інші засоби залишаються доступними під час роботи програми, тому вносити зміни можна в процесі відлагодження.

Builder C++ поставляється в трьох варіантах: Standard (стандартний), Professional (для професіоналів розробників, орієнтованих на мережеву архітектуру) і Client/Server Suite (для розробки систем в архітектурі клієнт/сервер). Останні два варіанти доповнюють стандартний початковими текстами візуальних компонентів, різномасштабним словником даних, новими функціями мови запитів SQL для бази даних, пакетом підтримки систем Internet, службою моніторингу програм, а також рядом інших засобів.

Builder C++ підтримує зв'язок з різними базами даних 3-х видів: dBASE і Paradox; Sybase, Oracle, InterBase і Informix; Excel, Access, FoxPro і Btrieve. Механізм BDE (Borland Database Engine) додає обслуговуванню зв'язків з базами даних дивовижну простоту і прозорість. Провідник Database Explorer дозволяє зображати зв'язки і об'єкти баз даних графічно. Використовуючи компоненти баз даних, я побудував електронний записник згідно таблиці dBASE за півгодини роботи на комп'ютері. Спадкоємство готових форм і їх "підгонка" під специфічні вимоги помітно скорочують тимчасові витрати на вирішення подібних завдань.

Довідкова служба Builder C++ надавала мені допомогу в цій і багатьох інших подібних ситуаціях. Є повний опис кожного управляемого компонента, включаючи списки властивостей і методів, а також численні приклади. Виклад матеріалу в книзі був значно покращуваний і систематизований завдяки відомостям, почерпнутим мною з довідкової служби.

Завдяки засобам управління проектами, двосторонній інтеграції додатку і синхронізації між засобами візуального і текстового редагування, а також вбудованому відладнику (з асемблерним вікном прокрутки, покрокового виконання, точок останову, трасуванням і тому подібне) – Builder C++ корпорації Borland надає собою вражаюче середовище розробки, яка, мабуть, витримає конкурентну боротьбу з такими модними продуктами як Developer Studio фірми Microsoft.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки для відеоспостереження на базі бездротових мереж.

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

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

б) вибрати та обґрунтувати методику побудови системи кібербезпеки 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 кібербезпеки в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Бездротові технології

Останніми роками напрям бездротових комп'ютерних мереж та віддаленого доступу зазнав бурхливого розвитку. Це пов'язано з поширенням блокнотних комп'ютерів, систем пошукового виклику (так званих пейджерів) та появою систем класу «персональний секретар» (Personal Digital Assistant (PDA)), розширенням функціональних можливостей стільникових телефонів. Такі системи повинні забезпечити ділове планування, розрахунок часу, зберігання документів та підтримку зв'язку з віддаленими станціями. Девізом цих систем стало anytime, anywhere, тобто надання послуг зв'язку незалежно від місця та часу. Крім того, бездротові канали зв'язку актуальні там, де неможливе або дороге прокладання кабельних ліній та значні відстані. Донедавна більшість бездротових комп'ютерних мереж передавала дані зі швидкістю від 1.2 до 14.0 Кбіт/с, найчастіше тільки короткі повідомлення (передавання файлів великих розмірів чи довгі сеанси інтерактивної роботи з базою даних були недоступні). Нові технології бездротового передавання оперують зі швидкостями в декілька десятків мегабітів за секунду.

Класифікація бездротових мереж

Залежно від технологій та передавальних середовищ, які використовують, можна визначити такі класи бездротових мереж:

- мережі на радіомодемах;
- мережі на стільникових модемах;
- інфрачервоні системи;
- системи VSAT;
- системи з використанням низькоорбітальних супутників;

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

- системи з технологією SST;
- радіорелейні системи;
- системи лазерного зв'язку.

Типи бездротових мереж

Залежно від використовуваної технології бездротові мережі можна розділити на три типи:

- локальні обчислювальні мережі;
- розширені локальні обчислювальні мережі;
- мобільні мережі (переносні комп'ютери).

Основні відмінності між цими типами мереж – параметри передачі. Локальні і розширені локальні обчислювальні мережі використовують передавачі і приймачі, що належать тій організації, в якій функціонує мережа. Для переносних комп'ютерів середовищем передачі служать загальнодоступні мережі, наприклад телефонна мережа або Інтернет.

Локальні обчислювальні мережі

Типова бездротова мережа виглядає і функціонує практично так само, як кабельна, за винятком середовища передачі. Бездротовий мережевий адаптер з трансивером встановлений в кожному комп'ютері, і користувачі працюють так, ніби їх комп'ютери сполучені кабелем.

Точки доступу

Трансивер, який ще й іноді називають точкою доступу (access point), забезпечує обмін сигналами між комп'ютерами з бездротовим підключенням і кабельною мережею. У бездротових ЛОМ використовуються невеликі настінні трансивери. Вони встановлюють радіоконтакт з переносними пристроями. Наявність цих трансиверів і не дозволяє назвати таку мережу строго бездротовою.

Способи передачі даних

Бездротові локальні мережі використовують чотири способи передачі даних:

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- інфрачервоне випромінювання;
- лазер;
- радіопередачу у вузькому діапазоні (одночастотна передача);
- радіопередачу в розсіяному спектрі.

Інфрачервоне випромінювання

Всі інфрачервоні бездротові мережі використовують для передачі даних інфрачервоні промені. У подібних системах необхідно генерувати дуже сильний сигнал, оскільки інакше значний вплив робитимуть інші джерела, наприклад світло з вікна. Цей спосіб дозволяє передавати сигнали з великою швидкістю, оскільки інфрачервоне світло має широкий діапазон частот. Інфрачервоні мережі здатні нормально функціонувати на швидкості 10 Мбіт/с.

Існує чотири типи інфрачервоних мереж:

- Мережа прямої видимості. У таких мережах передача можлива лише у разі прямої видимості між передавачем і приймачем.

- Мережі на розсіяному інфрачервоному випромінюванні. При цій технології сигнали, відображаючись від стін і стелі, врешті-решт досягають приймача. Ефективна область дії обмежена приблизно 30 м (100 футами), і швидкість передачі невелика (унаслідок нерівномірності сигналу).

- Мережі на відбитому інфрачервоному випромінюванні. У цих мережах оптичні трансивери, розташовані поряд з комп'ютером, передають сигнали в певне місце, звідки вони пересилаються відповідному комп'ютеру.

- Модульовані оптичні мережі. Ці інфрачервоні бездротові мережі відповідають жорстким вимогам мультимедійного середовища і практично не поступаються в швидкості кабельним мережам. Хоча швидкість інфрачервоних мереж і зручність їх використання дуже привабливі, виникають труднощі при передачі сигналів на відстань більше 30 м (100 футів). До того ж такі мережі схильні до перешкод з боку сильних джерел світла, які є в більшості організацій.

однорангова мережа. Ви також можете підключити таку бездротову мережу до кабельної мережі на основі Windows NT Server, додавши до одного з комп'ютерів Windows NT-мережі пристрій Netwave Access Point.

Федеральна комісія з електров'язку США (FCC) визначила такі категорії PCS (Personal Communication Services) та відповідні смуги частот:

– вузькосмугові PCS (діапазон 900-901, 930-931, 940-941 МГц) для швидкісних пейджерних мереж, двонапрявленого передавання повідомлень, передавання повідомлень мовлення;

– широкосмугові PCS (120, 1850–2200 МГц);

– стільниковий зв'язок;

– цифрове передавання мовлення та даних;

– неліцензовані PCS (40 МГц, від 1890 до 1930 МГц);

– бездротові ЛМ та АТС організацій у найближчому радіусі дії;

– у межах одного будинку або групи будівель.

Неліцензовані PCS забезпечують передавання даних зі швидкістю до 10 Мбіт/с.

Мережі на радіомодемах

Для передавання даних використовують смуги частот радіо– та ультракороткохвильового діапазону. Кожен радіомодем має антену та передавач для напрямленого передавання сигналів. Найпопулярнішими технологіями бездротового передавання цього класу є:

– радіо Ethernet (IEEE 802.11);

– HIPERLAN;

– Bluetooth;

– IEEE 802.11

HiperLAN

HiperLAN (High Performance Radio Local Area Network) розроблена Європейським інститутом стандартів з телекомунікаційних технологій (European

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

Telecommunications Standards Institute). Вона є аналогом IEEE 802.11, який використовують у Європі, і буває таких різновидів:

- HiperLAN / 1 – швидкість до 20 Мбіт/с у діапазоні 5 ГГц;
- HiperLAN / 2 – швидкість до 54 Мбіт/с у діапазоні 5 ГГц.

Bluetooth

Bluetooth – це інтерфейсна бездротова технологія. Діаметр мережі 10-30 м (у перспективі – 100 м). Працює в багатопунктовому режимі, не обов'язково в зоні прямої видимості. Головне призначення – створення побутових мереж, приєднання мультимедійної периферії, пральних машин, холодильників тощо. Концепцію мережі Bluetooth розробила 1994 р. шведська фірма Ericsson. Назва технології походить від прізвища, що його дали вікінгу Геральду Блатанду, який у X ст. об'єднав розрізнені землі, створивши Данське королівство. В 1997 р. створено перші приймачі-передавачі. У 1998 р. сформовано групу SIG, у яку ввійшли Ericsson, IBM, Intel, Nokia, Toshiba. У 1999 р. випущено специфікації на обладнання. Детальніше про технологію Bluetooth. Нові технології бездротового передавання (Ultra Wideband (UWB)) пропонують швидкості передавання, які перевищують 100 Мбіт/с, та потребують мінімальних витрат енергії.

Технологія SST

У технології SST (Spread Spectrum Technology) використано розподіл сигналу за спектром частот. Це дає змогу значно підвищити перепускню здатність каналу завдяки більшій завадостійкості. Технологію SST уже тривалий період застосовували для військових потреб. Є два різновиди мереж SST:

- FH-SS. Приймач та передавач синхронно перескакують з частоти на частоту;
- DH-SS. У кожний момент часу сигнал «розмазано» по широкому діапазону частот. Технологія SST дає змогу не тільки збільшити перепускню здатність мережі, а й ліпше реалізувати захист інформації від прослуховування. Зовнішній спостерігач таку інформацію сприймає як «білий шум».

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Супутникові технології

Технологія VSAT

Технологія VSAT (Very Small Aperture Terminal) використовує для передавання даних геостаціонарні супутники, розміщені над екватором Землі на висоті 40 тис. км. Наземні станції для зв'язку зі супутником застосовують еліптичні антени діаметром 3 м. Канал VSAT:

- забезпечує швидкість передавання даних до 2 Мбіт/с;
 - дає змогу реалізувати сполучення на великі відстані з переходом державних кордонів;
 - сумірний за ціною з кабельними каналами такої ж пропускну здатності.
- Водночас цей канал відрізняється значними затримками передавання даних, зумовленими великою відстанню до супутника (затримка становить приблизно 250 мкс, тоді як для кабельних мереж – 15 мкс). Тому канал VSAT не можна використовувати у системах реального часу та оперативного зв'язку.

Оскільки вартість супутникового каналу велика, то постачальник послуг купує у власника супутника канал зв'язку великої ємності і продає частини пропускну здатності каналу. Отже, мережа з використанням ланок VSAT має зіркову структуру.

Системи низькоорбітальних супутників

Системи на базі низькоорбітальних супутників LEO (Low Earth Orbit), як і системи VSAT, для передавання використовують супутник. Супутник розміщено на висоті близько 100 км на звичайній, а не геостаціонарній орбіті. У цьому випадку зменшується затримка в передаванні даних. Крім того, вивести такий супутник на орбіту значно дешевше, ніж геостаціонарний. Водночас для підтримання постійного зв'язку треба використовувати велику кількість таких низькоорбітальних супутників. Серед наявних проєктів LEO можна виділити систему Iridium, яка використовує 66 супутників.

У першому варіанті передбачали, що в системі буде 77 супутників. Саме стільки електронів містить атом іридію. Пізніше виявилось, що достатньо 66.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Однак назву вирішили залишити (назва елемента з 66 електронами диспрозію походить від латинського *disprosius* – важкодосяжний).

Корпорація Teledesic, власниками якої є Bill Gates та Greg MacCaw, планує створити всесвітню систему передавання мультимедійної інформації на основі LEO-технології. Планують, що така мережа використовуватиме 840 супутників і надаватиме користувачам канали передавання здатності від 62 Кбіт/с до 2 Мбіт/с.

Мережі на стільникових модемах

Мережі на стільникових модемах використовують наявну інфраструктуру стільникової телефонії. Вони працюють в особливо важких умовах великих заводів, періодичного зникнення сигналу.

Серед методів доступу виділяють аналогові, які використовують для передавання аналоговий сигнал. Це класичні методи доступу у стільникових мережах FDMA (Frequency Division Multiple Access), TACS (Total Access Communication System).

Головний ресурс стільникової мережі – це призначений для неї діапазон частот. Аналогові методи доступу виділяють для кожного передавання окремий канал – смугу частот у призначеному для мережі діапазоні. У цьому випадку сусідні стільникові комірки не можуть працювати в одному й тому ж діапазоні частот (інакше передавання в сусідніх комірках заважали б одне одному). Частотний діапазон поділяють на сім частин.

Серед методів доступу, які використовують цифрове передавання, популярні різні модифікації TDMA (Time Division Multiple Access). Вони застосовують відомий принцип розподілу часу передавання на окремі часові слоти. До цієї групи методів належать AMPS (Advanced Mobile Phone Service) (частотні канали завширшки 30 кГц поділяють на три часові слоти), NAMPS (Narrowband AMPS), PDC (канали по 25 кГц, три слоти), GSM (діапазон 200 кГц, вісім слотів).

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

CDMA

Найпередовішою сьогодні є технологія CDMA (Code Division Multiple Access), що використовує цифрове передавання.

CDPD

Технологія CDPD (Cellular Digital Packet Data) реалізує як пакетне передавання (протокол TCP / IP), так і модемний інтерфейс (AT-команди). На відміну від радіомодемів, стільникові модеми використовують не спеціальні антени та приймачі-передавачі, а відповідні пристрої стільникового телефону. Під час передавання даних застосовують протоколи MNP-10 або ETC. Протокол MNP-0 динамічно оптимізує швидкість передавання даних та рівень сигналу, має розвинуті засоби працювання помилок.

ETC

Протокол ETC запропонувала 1993 р. фірма AT&T Paradyne. Він ґрунтується на стандарті V.32bis (14.4 Кбіт/с) і дає змогу підтримувати зв'язок з іншими модемами стандарту ETC та іншими протоколами. Порівняно з MNP-10 досконаліший технічно. Розвиток технологій на вищих рівнях протоколу виражений в організації доступу до Internet. Цей доступ можливий завдяки використанню WAP-технологій.

Системи на базі інфрачервоних каналів

Системи на базі інфрачервоних каналів відрізняються невеликою вартістю приймачів та передавачів (від 1.5 до 4.5 дол. США), високими швидкостями передавання. Однак інфрачервоні канали працюють тільки в умовах прямої видимості. Асоціація Infrared Data Communications розробила стандарт передавання інфрачервоним каналом зі швидкістю 115.2 Кбіт/с.

Радіорелейний зв'язок

Радіорелейні станції (PPC) використовують для передавання аналогового сигналу в телебаченні та цифрового в послідовному коді за стандартом ITU G.703 в телефонії. Канал G.703 має пропускну здатність 2 Мбіт/с. Його можна використати, наприклад, для сполучення сегментів Ethernet. Сучасні цифрові PPC

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

мають смугу пропускання 2-34 Мбіт/с. Тому часто її розділяють на декілька каналів. Максимальна відстань для зв'язку РРС – 60-80 км. Для наземних РРС використовують частотні діапазони 1, 5, 7, 15, 23, 34 ГГц. Взаємодії маршрутизатора та РРС досягають за допомогою конвертера V.35 / G.703.

IEEE 802.11

IEEE 802.11 – це родина технологій бездротового передавання в радіодіапазоні. Сьогодні найпопулярніша технологія стандарту IEEE 802.11b; вона дає змогу передавати дані зі швидкістю 11 Мбіт/с на відстань від кількох до десятків кілометрів. Вихідна швидкість залежить від рівня завод, обладнання. На базі IEEE 802.11b будують бездротові локальні мережі Wireless LAN (WLAN)).

Група стандартів IEEE 802.11 фактично визначає фізичний та каналний рівень протоколів передавання. Стандарти відрізняються реалізаціями фізичних рівнів передавання, забезпечують різні швидкості.

IEEE 802.11 – це попередня версія стандарту, відома як радіо Ethernet (Wireless Ethernet); сьогодні вже застаріла.

IEEE 802.11b забезпечує максимальну швидкість передавання 11 Мбіт/с та використовує 14 каналів у діапазоні 2.4 ГГц.

IEEE 802.11a забезпечує швидкість передавання 54 Мбіт/с. Працює в діапазоні 5 ГГц. Має 12 каналів передавання. У ній використовують два піддіапазони передавання 5.15-5.25, 5.25-5.35 ГГц.

IEEE 802.11g – забезпечує швидкість передавання 22 Мбіт/с. Працює в діапазоні 2.4 ГГц. Повністю сумісний з IEEE 802.11b, однак пропонує три нові методи кодування, які дають змогу збільшити швидкість.

Організація Wireless Ethernet Compatibility Alliance (WECA) сертифікує обладнання на відповідність IEEE 802.11b і ставить на ньому позначку Wi-Fi compatible (Wireless Fidelity).

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

3.2 Розробка структурної схеми

Опис бездротової системи відеоспостереження починається із загальних відомостей і з опису основних компонентів, з яких складається система:

– Камера – пристрій, що формує зображення. «Ока» і «вуха» системи. Камера знімає, і передає відеосигнал передавачу. Нічим не відрізняється від провідних аналогів.

– Передавач – пристрій, що передає відеосигнал по бездротовому каналі на приймач. Для бездротового відео, як правило, використовують так звані побутові частоти: 2,4 ГГц. Відповідно використання даних передавачів не вимагають ліцензій і дозволів від держорганів.

– Приймач – пристрій, що приймає відеосигнал по бездротовому каналі, і передавальне його на відеореєстратор або монітор.

– Відеореєстратор – пристрій, що записує відеосигнал, що прийшов від приймача. По суті – аналог побутового відеомагнітофона. Пише відео на жорсткий диск або флешку.

– Монітор – пристрій візуального виводу відеоінформації. На ньому ми бачимо, що, що знімає в цей момент камера.

Разом: камера знімає відео, транслює на передавач, передавач транслює відеосигнал в ефір, приймач приймає сигнал і через відеовиходи передає сигнал або на відеореєстратор.

Примітно, що фізично, система відеоспостереження далеко не завжди складається з 5 частин. Кожний виробник намагається вирішити питання компактності й ергономічності системи по-своєму. Приміром, передавач практично завжди інтегрується в камеру. Приймач із відеореєстратором або монітором поєднують в одному корпусі рідше. Але іноді навіть всі три компоненти: приймач, відеореєстратор і монітор розташовують в один корпус. Можливо, виникне питання: що краще модульні системи, або ж коли всі «в одній коробці»? Відповідь на дане питання немає. Форм фактор не визначає якість, і чи

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

будуть це дешеві системи відеоспостереження чи ні сказати складно. Все залежить від розв'язуваного завдання. Приміром, якщо купуєте камеру з передавачем, то не зможете змінити характеристики камери. Якщо ж окремо купуєте камеру й передавач, то зможете підібрати камеру з потрібними параметрами й здійснити відеоспостереження периметра більш якісно. Або ж, якщо Ви збираєтеся монтувати систему самостійно, то набагато зручніше взяти найбільш інтегрований варіант для того, щоб не возиться із з'єднанням компонентів. Тобто: монітор, об'єднаний з відеореєстратором і приймачем і камеру з убудованим передавачем.

Структурна схема розробленої системи зображена на рисунку 3.1. На ній показано структуру системи кібербезпеки для відеоспостереження на базі бездротових мереж.

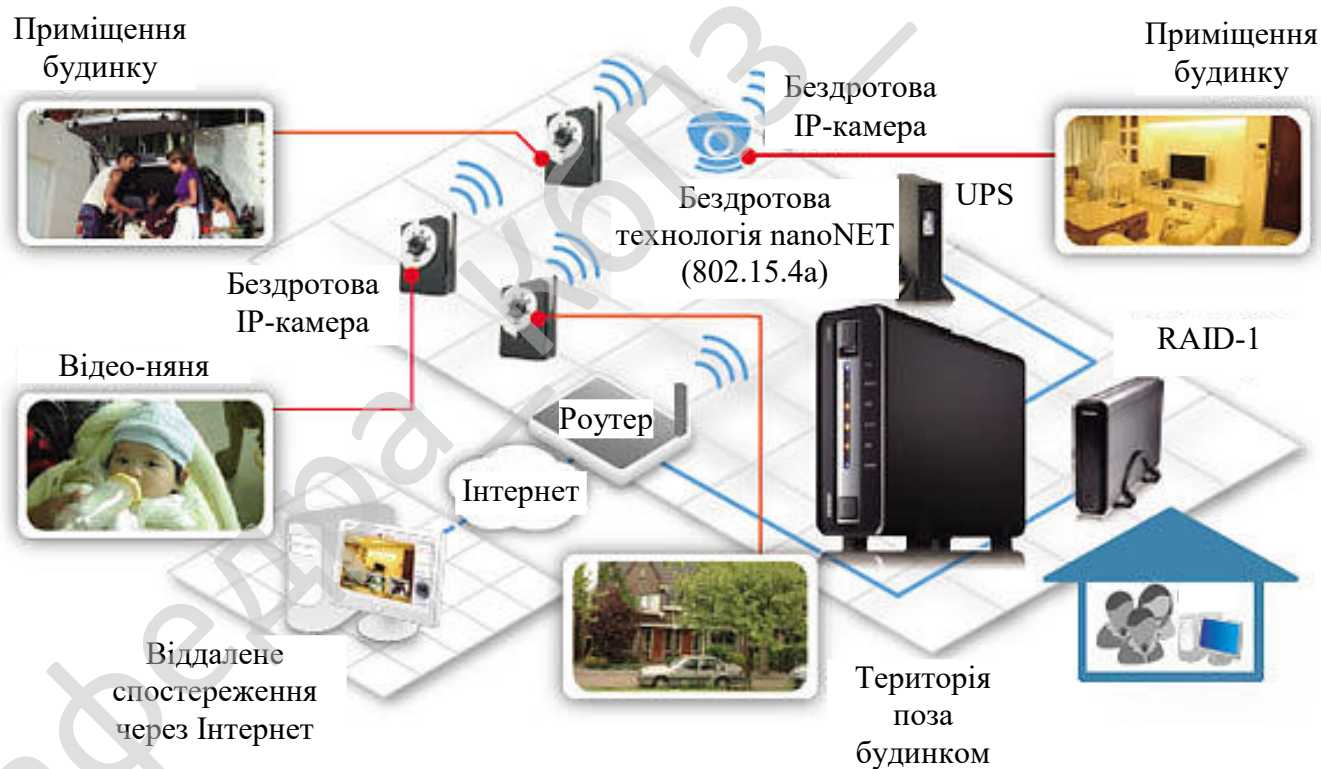


Рисунок 3.1 – Структурна схема системи

Схема складається з наступних компонент:

– Бездротові IP-камери.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

- Відеореєстратор.
- Бездротовий маршрутизатор (роутер).
- ПК куди записуються дані, при цьому на ПК реалізована технологія RAID-1, для підвищення надійності зберігання даних.
- UPS – пристрій безперебійного живлення.

Дані передаються за бездротовою технологією nanoNET (802.15.4a).

Крім того реалізований віддалений доступ через Інтернет до відеокамер.

Бездротовий маршрутизатор (роутер)

Маршрутиза́тор (англ. router) – електронний пристрій, що використовується для поєднання двох або більше мереж і керує процесом маршрутизації, тобто на підставі інформації про топологію мережі та певних правил приймає рішення про пересилання пакетів мережевого рівня (рівень 3 моделі OSI) між різними сегментами мережі.

Для звичайного користувача маршрутизатор (роутер) – це мережевий пристрій, який підключається між локальною мережею й інтернетом. Часто маршрутизатор не обмежується простим пересиланням даних між інтерфейсами, а також виконує й інші функції: захищає локальну мережу від зовнішніх загроз, обмежує доступ користувачів локальної мережі до ресурсів інтернету, роздає IP-адреси, шифрує трафік і багато іншого.

Маршрутизатори працюють на мережному рівні моделі OSI: можуть пересилати пакети з однієї мережі до іншої. Для того, щоб надіслати пакети в потрібному напрямку, маршрутизатор використовує таблицю маршрутизації, що зберігається у пам'яті. Таблиця маршрутизації може складатися засобами статичної або динамічної маршрутизації.

Крім того, маршрутизатори можуть здійснювати трансляцію адреси відправника й одержувача (англ. NAT, Network Address Translation), фільтрацію транзитного потоку даних на основі певних правил з метою обмеження доступу, шифрування/дешифрування передаваних даних тощо.

Маршрутизатори не можуть здійснювати передачу широкомовних повідомлень, таких як ARP-запит.

Маршрутизатором може виступати як спеціалізований пристрій, так і звичайний комп'ютер, що виконує функції простого маршрутизатора.

Зазвичай маршрутизатор використовує адресу одержувача, вказану в пакетах даних, і визначає за таблицею маршрутизації шлях, за яким слід передати дані. Якщо в таблиці маршрутизації для адреси немає описаного маршруту, пакет відкидається.

Існують і інші способи визначення маршруту пересилки пакетів, коли, наприклад, використовується адреса відправника, використовувані протоколи верхніх рівнів і інша інформація, що міститься в заголовках пакетів мережевого рівня. Нерідко маршрутизатори можуть здійснювати трансляцію адрес відправника і одержувача, фільтрацію транзитного потоку даних на основі певних правил з метою обмеження доступу, шифрування/дешифровка передаваних даних і т.д.

Таблиця маршрутизації

Таблиця маршрутизації містить інформацію, на основі якої маршрутизатор приймає рішення про подальшу пересилку пакетів. Таблиця складається з деякого числа записів – маршрутів, в кожній з яких міститься адреса мережі одержувача, адреса наступного вузла, якому слід передавати пакети і певна вага запису, – метрика. Метрики записів в таблиці грають роль в обчисленні найкоротших маршрутів до різних одержувачів. Залежно від моделі маршрутизатора і використовуваних протоколів маршрутизації, в таблиці може міститися деяка додаткова службова інформація. Наприклад:

Таблиця маршрутизації може складатися двома способами:

– статична маршрутизація – коли записи в таблиці вводяться і змінюються вручну. Такий спосіб вимагає втручання адміністратора щоразу, коли відбуваються зміни в топології мережі. З іншого боку, він є найстабільнішим і

таким, що вимагає мінімуму апаратних ресурсів маршрутизатора для обслуговування таблиці.

– динамічна маршрутизація – коли записи в таблиці оновлюються автоматично за допомогою одного або кількох протоколів маршрутизації – RIP, OSPF, EIGRP, IS-IS, BGP, і ін. Крім того, маршрутизатор будує таблицю оптимальних шляхів до мереж призначення на основі різних критеріїв – кількості проміжних вузлів, пропускної спроможності каналів, затримки передачі даних тощо. Критерії обчислення оптимальних маршрутів найчастіше залежать від протоколу маршрутизації, а також задаються конфігурацією маршрутизатора. Такий спосіб побудови таблиці дозволяє автоматично тримати таблицю маршрутизації в актуальному стані і обчислювати оптимальні маршрути на основі поточної топології мережі. Проте динамічна маршрутизація надає додаткове навантаження на пристрої, а висока нестабільність мережі може приводити до ситуацій, коли маршрутизатори не встигають синхронізувати свої таблиці, що приводить до суперечливих відомостей про топологію мережі в різних її частинах і втраті передаваних даних.

Часто для побудови таблиць маршрутизації використовують теорію графів.

Технологія NanoNET

Nanotron Technologies – берлінська компанія, що досліджує такі питання бездротового зв'язку з малим радіусом дії, як поліпшення показників завадостійкості, питання енергоспоживання й швидкості передачі даних у бездротових мережах малого радіуса дії, а також питання локалізації пристроїв бездротового зв'язку й розробку протоколів для мереж датчиків бездротового зв'язку. Метод, застосований самою природою – лінійно частотна модуляція (кажани, дельфіни користуються даним методом для того, щоб визначити, де вони перебувають) став основою технології за назвою NanoNet.

Де ж актуальне застосування приймачепередатчиків виробництва компанії Nanotron? Такі приймачепередатчики мають діапазон в 2,4 ГГц і

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

використовуються там, де використання мереж Wi-Fi неможливо через їхню властивість споживати багато енергії, а також там, де продуктивності ZigBee і Bluetooth катастрофічно не вистачає. Більш конкретно – це системи домашньої автоматизації, моніторингу й керування, охоронні системи.

Передача інформації через приймачепередатчики Nanotron відбувається зі швидкістю до 2 Мбіт за секунду, що є дуже великою швидкістю, особливо при досить високому рівні стійкості до перешкод. Це є основною відмінністю даних передавачів до інших пристроїв, що працюють у діапазоні 2,4 ГГц. Причому інформація може передаватися на відстань не однієї сотні метрів. У цьому сезоні також почався випуск приймачепередатчиків Nanotron nanoLOC TRX, які крім передачі даних виконують функцію визначення координат місця розташування об'єкта, що дозволяє більш ефективно використовувати системи збору даних і ідентифікації на радіочастотах.

У приймачепередатчиках Nanotron використовується метод, що дозволяє значно розширювати спектр, а також за рахунок високої потужності переданого сигналу, що розминається по всьому спектрі дії, ще більше збільшувати показники завадостійкості. Це досягається за рахунок того, що "розминання" по спектрі приводить до втрат тільки частини сигналу, що передається, що дозволяє в наслідку відновити в приймачі передані дані. Такий метод називається лінійно-частотною модуляцією.

Тепер про імпульси лінійно-частотної модуляції. Такі імпульси в приймачепередатчиках Nanotron використовуються з метою передачі двійкових даних, їхня тривалість фіксована, а несуча частота може бути як наростаючою, так і спадаючою. Ширина частотного каналу, що використовується в таких приймачепередатчиках, набагато більше ніж та, котра використовується в технологіях ZigBee і Bluetooth і становить 64 МГц. Саме цей факт дає можливість роботи на високих швидкостях і з підвищеною надійністю передачі даних, навіть якщо рівень перешкод перевищує стандартні значення.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

Якщо порівнювати з бездротовою передачею даних за технологією Wi-Fi, у якій ширина частотного каналу також має значні показники, то технологи Nanotron також мають свої переваги, у цьому випадку в плані швидкості, дальності передачі, а також кількості споживаної енергії. Справа в тому, що обробка імпульсів лінійно-частотної модуляції на первинному рівні відбувається аналоговим способом, що дозволяє використання приймачепередатчиків у пристроях, які працюють від батарей.

Сигнал лінійно-частотної модуляції форматується (при передачі) і обробляється (при прийманні) при використанні дисперсійної лінії затримки, що виконана на базі фільтра ПАВ. Якщо рівень помилок фіксований, то високі швидкості прийому-передачі даних досягаються за рахунок високого рівня ширини спектра сигналу, що дорівнює 64 МГц. Але є один недолік. Така ширина не дозволяє використовувати в одному приміщенні більше двох мереж.

Приймачепередатчики NanoNet TRX відрізняються високою швидкістю передачі даних (2 Мб у секунду), потужністю діапазоном 1 мВт – 6,3 мВт, більшим радіусом дії, що на відкритому просторі може рівнятися аж до 900 метрів, а також убудованим контролером типу MAC, що одночасно підтримує кілька різних методів доступу до спектра передачі.

При використанні приймачепередатчиків nanoNET TRX і створенні на їхній базі мережних додатків рекомендується використовувати один із двох варіантів ПЗ – "Driver software" або "Portable Protocol Stack (PPS)". Те, який варіант у підсумку буде обраний, залежить від того, наскільки складне бездротове з'єднання маєтись на увазі. Пропоноване програмне забезпечення являє собою вихідні коди, написані мовою C. Перший пакет здатний забезпечити працездатність функцій по прийому-передачі інформації й управляти режимами функціонування приймачепередатчика. Другий пакет призначений для більше складних мереж, і дозволяє набудовувати конфігурацію протоколу залежно від вимог самого додатка.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

Використання програмного забезпечення PPS і приймачепередатчиків nanoNET TRX дозволяють реалізовувати різні типи мереж, які можуть підтримувати доступ до спектра передачі як прямого, так і випадкового плану. Випадковий доступ може організовуватися методом CSMA / CA (запобігання колізій) або кількарізним доступом з визначенням несучої як апаратними засобами, так і за допомогою ПЗ PPS. При цьому мережа може складатися з однієї або декількох підмереж. У випадку декількох радіус дії мережі може збільшуватися. Реалізація прямого доступу можлива за схемою TDMA (часовий поділ) або за схемою " майстер-ведений".

Наприкінці березня цього року був затверджений новий стандарт для фізичного рівня БПД – IEEE 802.15.4a. Він розроблений для систем з високим рівнем перешкод на базі технології CSS розробленою компанією Nanotron і затверджений інститутом інженерів електротехніки й електроніки IEEE.

IEEE 802.15.4

IEEE 802.15.4 – стандарт, що визначає фізичний шар і керування доступом до середовища для бездротових персональних мереж з низьким рівнем швидкості. Стандарт підтримується робочою групою IEEE 802.15. Є базовою основою для протоколів ZigBee, WirelessHART, і MiWi, кожний з яких, у свою чергу, пропонує рішення для побудови мереж за допомогою будівлі верхніх шарів, які не регламентуються стандартом. Як альтернатива він може бути використаний спільно зі стандартом 6LoWPAN і стандартними протоколами Інтернету для побудови убудованого бездротового Інтернету.

Ціль стандарту IEEE 802.15 – запропонувати нижні шари основи мережі для мереж типу бездротових персональних мереж, орієнтованих на низьку вартість, низьку швидкість повсюдного зв'язку між пристроями (по контрасту з багатьма більше кінцево-орієнтованих на користувача мережами, як наприклад Wi-Fi). Акцент робиться на дуже низькій вартості зв'язку з найближчими пристроями, зовсім без (або з невеликий) базовою структурою, з метою експлуатації на досі небувалому низькому рівні енергії.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Основна межа прийому – 10-метрова область зв'язку зі швидкістю передачі 250 Кбіт/с. Компроміси можливі на користь більш, що вбудовуються радикально пристроїв, із ще більш низькою потребою в енергії, шляхом визначення не одного, а декількох фізичних рівнів. Спочатку були визначені низькі швидкості передачі в 20 і 40 Кбіт/с, швидкість в 100 Кбіт/с була додана в поточному перевипуску.

Ще більш низькі швидкості передачі можуть бути розглянуті з результирующим ефектом зниження енергоспоживання. Як уже згадувалося, головною відмінною рисою стандарту 802.15.4 серед бездротових персональних мереж важливим є низька вартість виробництва й витрат по експлуатації, простота технології.

У ряді найважливіших функцій перебувають забезпечення роботи в режимі реального часу за допомогою збереження часових слотів, запобігання одночасного доступу й комплексна підтримка захисту мереж. Пристрої також включають функції керування витратою енергії, такі як якість з'єднань і детектування енергії. Сумісні зі стандартом 802.15.4 пристрої можуть використовувати одну із трьох можливих частотних смуг для роботи.

Архітектура протоколу

Пристрої розроблені з метою взаємодіяти один з одним за допомогою понятійної простої бездротової мережі. Визначення шарів мережі засновано на мережній моделі OSI, хоча тільки нижні шари визначені в стандарті, взаємодія з верхніми шарами передбачається, з можливим використанням підрівня керування логічним зв'язком, допускаючи MAC крізь підрівень збіжності. Реалізовані пристрої можуть покладатися на зовнішні пристрої або бути просто убудовані як самостійно функціонуючі пристрої.

Фізичний шар

Фізичний шар, в остаточному підсумку, надає послуги передачі даних, також як і інтерфейс організації керування фізичним шаром і забезпечує базу дані інформації відповідної персональної мережі. У такий спосіб фізичний шар

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

управляє трансиверною радіостанцією й виконує вибір каналів і енергії й сигнальні функції керування. Він діє в одній із трьох можливих неліцензуємих радіочастотних смугах.

– 868.0-868.6 МГц: Європа, дозволяється один канал зв'язку (2003, 2006).

– 902-928 МГц: Північна Америка понад десять каналів (2003), розширене до тридцяти (2006).

– 2400-2483.5 МГц: використовується в усьому світі понад шістнадцять каналів (2003, 2006).

Первісна версія 2003 стандарту визначає два фізичних шари, заснованих на широкополосній модуляції із прямим розширенням спектра, один працює на смузі 868 / 915 МГц зі швидкістю передачі в 20 і 40 Кбіт/с, а іншої на смузі 2450 МГц зі швидкістю 250 Кбіт/с.

Перевипуск 2006 підвищує максимальні швидкості передачі даних на частотах 868 / 915 МГц, також надаючи їм швидкості в 100 і 250 Кбіт/с. Крім того, він іде далі, визначаючи чотири фізичних рівні залежно від методу модуляції. Три з них зберігають підхід широкополосній модуляції, у діапазоні 868 / 915 МГц використовується як двійкова так і квадратурна фазова маніпуляція (остання виглядає більше оптимальної) у діапазоні 2450 МГц, за допомогою останнього. Як альтернатива, оптимальний шар на частоті 868 / 915 МГц визначається використовуючи комбінацію двійкового кодування й амплітудної маніпуляції (таким чином, на основі паралельного, а не послідовного розширення спектра). Можливо динамічне перемикавання між підтримуваними шарами 868 / 915 МГц.

Крім цих трьох діапазонів IEEE 802.15.4c дослідницька група IEEE 802.15.4c бере до уваги недавно відкриті діапазони 314-316 МГц, 430-434 МГц, і 779-787 МГц у Китаї, у те час як цільова група IEEE 802.15.4d визначає виправлення до існуючого стандарту 802.15.4-2006 щоб підтримувати новий діапазон 950-956 МГц у Японії. Перші виправлення до стандарту, внесені цими групами були випущені у квітні 2009.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

У серпні 2007 IEEE 802.15.4a розширила чотири фізичних шари доступних у ранній версії 2006 до шести, включаючи один фізичний шар, що використовує послідовну радіо-технологію для високошвидкісної передачі даних Ultra-wideband (UWB) і іншу, що використовує частотне розширення спектра (CSS). Фізичний шар UWB виділений частотами в трьох діапазонах: нижче 1 ГГц, між 3 і 5 ГГц, і між 6 і 10 ГГц. На фізичний шар CSS виділений спектр у смузі 2450 МГц діапазону ISM.

У квітні 2009 стандарти IEEE 802.15.4c і IEEE 802.15.4d розширили доступні фізичні шари додавши кілька шарів, один з додаткових для частоти 780 МГц використовуючи квадратурну фазову маніпуляцію (Quadrature phase-shift keying, QPSK) або фазову маніпуляцію більше високих порядків (M-PSK), іншу для частоти 950 МГц, використовуючи гауссівську частотну маніпуляцію (Gaussian frequency-shift keying, GFSK) або двійкову фазову маніпуляцію (Binary phase-shift keying, BPSK).

Шар MAC

Шар механізму доступу (Media Access Control, MAC) здійснює передачу фрагментів даних структури MAC за допомогою використання фізичного каналу. Крім інформаційних послуг він пропонує керування інтерфейсом і сам по собі управляє розміщенням маячків на каналах. Він також контролює перевірку фрагментів структури, гарантує множинний доступ з поділом за часом і управляє зв'язками вузлів. Нарешті він пропонує точки-пастки для послуг безпеки.

Вищі шари

Стандарт не визначає інші більше високі шари й сумісність проміжних шарів. Існують специфікації, такі як ZigBee, побудовані на даному стандарті для того, щоб пропонувати інтегральні рішення. Стеки [операційної системи] TinyOS також використовують деякі види апаратного забезпечення IEEE 802.15.4.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

Модель мережі

Типи вузлів

Стандарт визначає два типи вузлів мережі Перший – повнофункціональний пристрій (FFD – Full-Function Device). Він може слугувати як координатор персональних мереж, так само може функціонувати як загальний вузол. Він реалізує загальну модель зв'язку, що дозволяє переговорюватися з іншими пристроями, також може передавати далі повідомлення, у цьому випадку він називається координатором (координатор PAN, коли він відповідає за всю мережу).

Інший – пристрій з полегшеними функціями (RFD – Reduced-Function Device). Визначення означає надзвичайно прості пристрої з дуже скромним ресурсом і вимогами до мережі, у зв'язку із цим вони можуть тільки зв'язуватися з повнофункціональними пристроями й ніколи не можуть діяти як координаторів.

Топології

Мережі можуть бути одноранговими (P2P, peer-to-peer, point-to-point), або мати топологію «зірка». Однак, будь-яка мережа повинна мати принаймні один FFD, що буде працювати як координатор мережі. Таким чином, мережі формуються із груп пристроїв, розділених відповідною дистанцією. Кожний пристрій має 64-бітний ідентифікатор, у деяких випадках може використовуватися 16-бітний ідентифікатор усередині обмеженої області. Таким чином, усередині кожної персональної мережі (англ. PAN, personal area network), для з'єднання будуть використовуватися короткі ідентифікатори.

Однорангові мережі (P2P) можуть формувати довільні структури з'єднань і їхніх розширень обмежені тільки дистанцією між кожною парою вузлів. Вони покликані слугувати як основу для бездротових мереж, що самоорганізуються, здатних до самоврядування й організації. Так як стандарт не визначає мережний рівень, маршрутизація не підтримується прямо, але такий додатковий рівень може здійснити підтримку мереж з ретрансляторами.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Також можуть бути додані додаткові топологічні обмеження: наприклад, дерево кластерів – структура, у якій RFD може бути зв'язаним тільки з одним FFD одноразово, таким чином, RFD є винятково листами дерева, а більшість вузлів є FFD. Також можлива ситуація ніздрюватої топології мережі, чії вузли є мережами кластерних дерев з локальним координатором для кожного кластера, крім глобального координатора.

Ще підтримується більше структурована топологія «зірка», де координатор мережі обов'язково повинен бути центральним вузлом. Така мережа може виникнути, коли FFD вирішує створити свою власну персональну мережу (PAN) і оголосити себе її координатором, після чого вибирається унікальний ідентифікатор для PAN. Після цього інші пристрою можуть приєднатися до мережі, що повністю незалежна від інших мереж з топологією «зірка».

Архітектура передачі даних

Фрагменти даних – основа для передачі даних, що здійснюється по чотирьох основних типах: (дані, підтвердження, маячок і фрагменти команд механізму доступу), що забезпечує розумний баланс між простотою й міцністю. На додаток може використовуватися суперфрагментна структура обумовлена координатором, у цьому випадку два маячка діють як її межі й забезпечують синхронізацію інших пристроїв, також як і інформацію про конфігурацію. Суперфрагмент складається із шістнадцяти слотів однакової довжини, які можуть бути надалі розділені на активну й неактивну частини, у ході [виконання] яким координатор може входити в енергозберігаючий режим, у якому не потрібен контроль мережі.

Твердження меж суперфрагментів виробляється системою CSMA / CA. Кожна передача повинна закінчуватися перед появою наступного маячка. Як згадувалося вище додатка нужденні в чітко певній широті діапазону можуть використовувати сім областей з однієї або більше беззмистовних гарантованих областей множинного доступу, що йдуть у кінці суперфрагмента. Звичайно суперфрагменти використовуються при роботі пристроїв з низьким схованим

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

[енерго]станом, чії зв'язки повинні зберігатися, навіть протягом довгого періоду неактивності.

Передачі даних до координатора вимагають фази маячкової синхронізації, за допомогою передачі режиму CSMA / CA, якщо це можливо (за допомогою множинного доступу, якщо використовуються суперфрагменти), [сигнал] підтвердження необов'язковий. Передача даних від координатора звичайно супроводжує запити до пристроїв, якщо маячки використовуються, то використовуються сигнали-запити, координатор підтверджує запит і позика посилає інформаційні пакети, які підтверджуються пристроєм. Те ж відбувається, якщо суперфрагменти не використовуються, тільки в цьому випадку немає маячків, щоб зберігати шляхи передачі інформації. Однорангові мережі можуть також використовувати режим CSMA / CA або механізми синхронізації, у цьому випадку зв'язок між двома пристроями можлива, у той час як в «структурованих» режимах один із пристроїв повинне бути координатором мережі. У загальному всі наступні процедури супроводжуються звичайним запитом-підтвердженням / індикацією-класифікацією відповіді.

Надійність і безпека

Фізичний носій можна одержати через протокол CSMA / CA. Мережі, що не використовують маячковий механізм використовують варіант, заснований на прослуховуванні носія, який підвергся впливу алгоритму зниження швидкості передачі, підтвердження не підкоряються цьому порядку. Загальна передача даних використовує вільні слоти, де використовуються маячки, процес не супроводжується підтвердженнями.

Повідомлення про підтвердження можуть носити необов'язковий характер при деяких обставинах, якщо зроблено припущення про успіх. У кожному разі якщо пристрій не може обробити фрагмент у цей момент, він просто не підтверджує його одержання: ретрансляція заснована на перерві може виконатися кілька разів, супроводжуючи після цього рішення або припинити або продовжити спроби.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Так як передбачене встаткування для цих пристроїв вимагає максимального збільшення життя батарей, для протоколів вибираються методи цьому сприятливі, здійснюючі періодичні перевірки для повідомлень, що очікують, частота яких залежить від застосування.

Що стосується захисту зв'язків, підрівень MAC пропонує можливості, які можуть бути використані у верхніх шарах для досягнення бажаного рівня безпеки. Процеси у вищих шарах можуть визначати ключі для виконання симетричної криптографії для захисту навантаження й обмеження її для груп пристроїв або просто для однорангового зв'язку, ці групи пристроїв можуть бути описані в списках контролю доступу.

Крім того MAC обчислює давнину перевірки між послідовними прийомів для запобігання можливого виходу старих кадрів, або даних (які більше не вважається дійсними) не виходять на більше високі шари. На додаток до цього захищеного режиму захисту є інший незахищений режим MAC, що дозволяє списки контролю доступу тільки як засіб для рішення про прийняття фрагментів відповідно до їхнього передбачуваного джерела.

Методи повторної передачі (ARQ)

Для підвищення завадостійкості системи бездротового відеоспостереження в бакалаврській роботі пропонується використовувати протокол ARQ – протокол повторної передачі даних.

Багато протоколів каналного рівня підтримують надійну передачу даних, виконуючи повторні передачі невдалих передач. Невдалі передачі повідомляються за допомогою повідомлень зворотного зв'язку, таких як повідомлення підтвердження прийому (ACK) і непідтвердження прийому (NACK) відповідно до протоколів автоматичного запиту повторної передачі (ARQ). Механізми ARQ, зокрема, важливі для бездротового середовища передачі, але також застосовуються до провідних ліній зв'язку. Приклади механізмів ARQ, що працюють по бездротових каналах, містять у собі:

– протоколи керування радіоканалом (RLC) для системи пакетного радіозв'язку загального користування (GPRS) і широкополосного множинного доступу з кодовим поділом каналу (WCDMA);

– протокол гібридного ARQ (HARQ) у високошвидкісному керуванні доступом до середовища (MAC-hs) для високошвидкісного пакетного доступу по спадній лінії зв'язку (HSDPA).

Проблема з такими протоколами в тому, що вони не можуть надати швидкий і надійний зворотний зв'язок і ефективно використання радіоресурсів.

Деякі протоколи попереднього рівня техніки використовують просту й швидку концепцію ACK / NACK, що вказує, чи був кадр даних успішно прийнятий. Такі протоколи не надають порядкових номерів у зворотному зв'язку, а замість цього передавач і приймач неявно встановлюють зворотний зв'язок для окремої передачі, експлуатуючи фіксовану часову залежність. Це часто називається синхронним зворотним зв'язком. Перевагою такого підходу є те, що короткі сигнали можуть посилати часто, тоді як витрата ресурсу передачі є відносно низьким. Ефективність кодування, що досягається, однак, обмежена або неможлива, якщо кожний ACK або NACK є одиночним бітом. Таким чином, існує ризик невірної тлумачення такого одиночного біта в приймачі. Загасаючі провали додатково збільшують імовірність помилки, і досягнення дуже низького коефіцієнта помилок може споживати багато ресурсів, щоб покрити найгірші провали. Таким чином, така передача сигналу також є дорогою, якщо потрібні дуже низькі коефіцієнти помилок, тому що це може бути досягнуто тільки за допомогою збільшення потужності передачі або за допомогою повтору інформації. Відновлення або повторна передача кожного повідомлення зворотного зв'язку, однак, неможлива, тому що необхідно неї синхронізувати за часом з передачею відповідних даних.

Інший клас протоколів використовує блоки зворотного зв'язку, або керування, (іноді іменовані повідомленнями про стан). Такі механізми найчастіше застосовуються для заснованих на вікнах ARQ-протоколів. Блоки

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

зворотного зв'язку можуть явно містити в собі порядкові номери й контрольну суму, а отже, може підтримуватися надійність повідомлень зворотного зв'язку. Неправильно прийнятий зворотний зв'язок не використовується, а відкидається на стороні відправника даних. Повторні передачі або передачі відновлень зворотного зв'язку використовуються, щоб гарантувати те, що зворотний зв'язок коректно прийнятий. Повинне бути відзначене, що такі блоки зворотного зв'язку не вимагають якого-небудь вирівнювання за часом з відповідними блоками даних через порядкову нумерацію блоків даних і посилання на них у блоках зворотного зв'язку. Ці типи механізмів зворотного зв'язку мають перевага в тому, що є дуже надійними; однак вони типово набагато повільніше в порівнянні із синхронними механізмами АСК / NACK – зворотного зв'язку.

Отже, в області техніки необхідні інтегровані протоколи повторної передачі, які досягають ефективності традиційних АСК / NACK-протоколів при одночасній реалізації надійності явних повідомлень зворотного зв'язку. Переважно, такі інтегровані протоколи повторної передачі можуть бути здійснені в одній категорії протоколів і засновані на тих самих блоках даних протоколу, стані протоколу й логіку.

Можливий ряд варіантів механізму ARQ. Кожний правильно прийнятий кадр може бути підтверджений окремим спеціальним кадром, або підтвердження може бути вставлене в керуюче поле інформаційних кадрів, що переносять дані у зворотному напрямку. В останньому випадку також повинні застосовуватися спеціальні кадри підтвердження, оскільки інформаційного кадру в потрібний момент може не виявитися.

Існує два види підтвердження про прийом: позитивне (АСК) і негативне (NACK або NAK). Але в кожному разі щоб уникнути перевантажень повинні застосовуватися перерви. Передавальна сторона, що не одержала відповіді (АСК або NACK) протягом заданого проміжку часу після передачі, повторює відповідний кадр. Щоб організувати процедуру перерв, кадри повинні зберігатися

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

в накопичувачі передавальної сторони до одержання підтвердження правильності передачі.

Існує три основні способи обробки відповідей на позитивні й негативні підтвердження:

– Старт / стопний, або передача із зупинкою й очікуванням (SAW – Stop And Wait), часто називаний блоковим методом передачі.

– З поверненням на N кадрів (GBN – Go Back N), також називаний потоковим методом передачі.

– Метод вибіркового (селективного) повтору (SR – Selective Repeat).

Коротко розглянемо принцип роботи перерахованих процедур.

SAW

Відповідно до цієї процедури без підтвердження може бути переданий тільки один кадр. Після передачі чергового кадру передавальна сторона чекає підтвердження. Якщо надходить негативне підтвердження або відбудеться перевищення часу тайм-ауту, кадр передається повторно. Кадр скидається (стирається) з накопичувача передавача лише після одержання позитивного підтвердження.

Дану процедуру зручно використовувати при напівдуплексному зв'язку, коли передача сторін чергується. Однак вона неефективна у випадку організації повнодуплексного зв'язку, особливо, якщо час поширення сигналу по каналі значно більше часу передачі кадру, що типово для супутникових і ряду інших каналів.

Якщо час поширення дуже малий (при невеликій довжині каналу або через низьку швидкість передачі), процедура SAW не приведе до серйозного зниження продуктивності всієї системи.

GBN

У цьому випадку кадри передаються безупинно без очікування підтвердження прийому певної кількості кадрів. При одержанні негативного

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

підтвердження або після закінчення встановленого часу очікування непідтверджених і всі наступні кадри передаються повторно.

У практичних версіях процедур GBN, наприклад у складі протоколу V.42, не всі кадри вимагають підтвердження. Позитивне, підтвердження може служити підтвердженням правильної передачі не тільки даного кадру, але й всіх попередніх йому.

Процедуру GBN часто називають ARQ типу REJ (REJECT), також як службові кадри, що переносять підтвердження NACK від приймача до передавача.

SR

Відповідно до процедури SR повторна передача даних здійснюється тільки для кадру, на який надійшло негативне підтвердження або минув час тайм-ауту підтвердження. Дана процедура, у порівнянні із процедурами SAW і GBN, істотно збільшує пропускну здатність СПД. Але для передачі й прийому кадрів не один по одному їхніх номерів на прийомній стороні повинен перебувати буферний накопичувач із довільним доступом. Зі збільшенням затримки поширення сигналу в каналі зв'язку необхідно збільшувати буферну пам'ять. Очевидно, реалізація процедури SR є більше складною й дорогою. Із цієї причини вона довго не могла знайти широкого комерційного застосування. Навіть у найбільш зробленому на сьогоднішній день протоколі V.42 процедура селективного повтору не є обов'язковою.

Спосіб SR часто називають ARQ типу SREJ (Selective REJECT), також як однойменні службові кадри, що переносять підтвердження про селективний неприйом від приймача до передавача.

Ефективність СПД зі схемою ARQ типу SR в ідеальному випадку залежить тільки від імовірності безпомилкового прийому кадрів, тобто від якості каналу зв'язку.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

3.3 Розробка функціональної схеми

Програмно-апаратний комплекс об'єкту кібербезпеки для відеоспостереження на базі бездротових мереж є комплектом устаткування, що розміщується на території об'єкту кібербезпеки для відеоспостереження на базі бездротових мереж та забезпечує первинний збір і обробку відеосигналу, його подальше зберігання та передачу в захищеному вигляді з використанням мережі передавання даних до центральних серверних комплексів.

Конфігурація програмно-апаратного комплексу об'єкту кібербезпеки для відеоспостереження на базі бездротових мереж включає:

– дві відеокамери з роздільною здатністю не менш як 640 x 480, з можливістю передачі звукового сигналу та швидкістю відеопотоку не менш як 25 кадрів на секунду, можливістю кольорової відеозйомки, настінного/стельового кріплення, а також обов'язковою фіксацією дати та часу зйомки;

– пристрій для запису та передавання відеосигналу, який здатен забезпечити запис із роздільною здатністю не менш як 640 x 480 із швидкістю відеопотоку не менш як 25 кадрів на секунду, тривалістю не менше 120 годин з кожної камери та передавання відеосигналу до центру обробки даних з роздільною здатністю та швидкістю залежно від якості каналу передавання даних, але не менш як 320 x 240 із швидкістю відеопотоку не менш як 15 кадрів на секунду;

– система безперебійного електропостачання обладнання програмно-апаратного комплексу об'єкту кібербезпеки для відеоспостереження на базі бездротових мереж протягом не менш як однієї години;

– необхідне комутаційне обладнання та з'єднувальні кабелі (довжина кабелів між камерами та пристроєм для запису і передаванням відеосигналу не повинна перевищувати 15 метрів);

– програмне забезпечення, необхідне для функціонування програмно-апаратного комплексу.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Функціональна схема розробленої системи зображена на рисунку 3.2.

З рисунку видно, що розроблена система складається з наступних частин:

- блок приймання та запису відеосигналу;
- блок обслуговування користувачів та трансляції відеосигналу (веб-сайт).

На етапі розроблення і впровадження системи відеоспостереження функціональні блоки можуть бути об'єднані в довільні логічні функціональні модулі. При такому об'єднанні кожен з модулів повинен бути функціонально незалежним від інших модулів. При цьому допускається створення територіально розподіленої системи функціональних модулів для забезпечення безвідмовності роботи системи відеоспостереження в цілому.

Для підвищення рівня безвідмовності роботи системи відеоспостереження кожен функціональний мережевий вузол центрального серверного комплексу повинен дублюватися. Застосування мережевих технологій і протоколів безвідмовного включення устаткування повинне гарантувати коректну роботу центрального серверного комплексу в разі виходу з ладу одного з двох мережевих пристроїв, з яких складається відповідний функціональний вузол.

Для захисту від мережевих загроз і атак (зокрема типу DoS та DDoS) обладнання центрального серверного комплексу повинне мати відповідне мережеве обладнання. Програмне забезпечення зазначеного обладнання повинне забезпечувати обробку мережевого трафіку на швидкості підключення до мережі передавання даних і мати у своєму складі достатню кількість мережевих інтерфейсів для підключення.

Блок приймання та запису відеоінформації повинен відповідати таким функціональним вимогам:

- безперервне приймання відео- та аудіопотоків з програмно-апаратних комплексів;
- зберігання отриманих даних в архіві відеозаписів протягом не менш як 120 годин з кожної камери відеоспостереження;
- авторизація джерела відеопотоку перед отриманням даних;

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

- заповнення цілісності відеопотоку шляхом дозавантаження відсутніх фрагментів з архіву програмно-апаратного комплексу;
- наявність можливості перегляду відеозаписів архіву, в тому числі з використанням пошуку.

Блок обслуговування користувачів та трансляції відеосигналу (веб-сайт) повинен відповідати таким функціональним вимогам:

- трансляція відеопотоків для інтернет-користувачів;
- розподіл навантаження за кластером серверів;
- використання кешування для оптимізації завантаження магістральних каналів;
- доступ до веб-порталу з використанням браузерів (Internet 7.0 і вище, Firefox 7.0 і вище, Opera 10.0 і вище, Chrome 10.0.648 і вище) та мобільних пристроїв (Android, iOS);
- попередня реєстрація з можливістю вибору камер відеоспостереження для здійснення перегляду;
- збереження вибраних камер відеоспостереження у списках відтворення;
- швидке перемикання користувача на одну з попередньо обраних камер відеоспостереження;
- пріоритетний доступ до всіх камер відеоспостереження для групи спеціальних користувачів;
- об'єднання на головній сторінці сайту декількох навігаційних систем;
- пошук за всіма географічними назвами;
- перегляд відеоматеріалів у режимі прямої трансляції з будь-якої попередньо обраної камери відеоспостереження;
- функція додавання камер в обране;
- використання тесту САРТСНА для забезпечення додаткового рівня захисту від великого потоку запитів;

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

– використання системи обмеження доступу за набором параметрів (комплексна перевірка, що передбачає обмеження кількості підключень з однієї IP-адреси за визначений період).

Функціональні вимоги до мережі передавання даних

Конфігурація каналів зв'язку мережі передавання даних включає:

– канал зв'язку об'єкту кібербезпеки для відеоспостереження на базі бездротових мереж для передавання відеосигналу до центрального серверного комплексу з пропускнуою здатністю за напрямом передавання даних до 512 Кбіт на секунду, інтерфейс підключення на об'єкті кібербезпеки для відеоспостереження на базі бездротових мереж Ethernet 10/100 Мбіт на секунду (RJ45);

– вхідний канал зв'язку центру обробки даних для отримання відеосигналу з програмно-апаратного комплексу з пропускнуою здатністю за напрямом отримання даних не менш як 15 Гбіт на секунду;

– вихідні канали зв'язку центру обробки даних для трансляції відеосигналу в Інтернеті із сумарною пропускнуою здатністю не менш як 80 Гбіт на секунду;

– канал зв'язку між центром обробки даних та центром керування системи відеоспостереження із пропускнуою здатністю не менш як 15 Гбіт на секунду.

Канали зв'язку повинні відповідати таким функціональним вимогам:

– організація тунельних каналів зв'язку між об'єктами кібербезпеки для відеоспостереження на базі бездротових мереж, центральним серверним комплексом і ситуаційним центром системи відеоспостереження з використанням технологій IP/MPLS, L2VPN, L3VPN та інших;

– забезпечення цілодобового обслуговування передачі даних каналами зв'язку на всіх вузлах мережі передавання даних;

– наявність централізованої диспетчерської служби, що працює в цілодобовому режимі, для обслуговування каналів передачі даних;

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

– мережа передавання даних у частині центрального серверного комплексу повинна мати повнопов'язану топологію архітектури побудови, яка забезпечує обмін інформацією між об'єктами кібербезпеки для відеоспостереження на базі бездротових мереж та вузлами центрального серверного комплексу і ситуаційним центром системи відеоспостереження без послуги, що додатково замовляється з високим рівнем надійності;

– зона відповідальності оператора зв'язку – від інтерфейсу мережевого устаткування в центральному серверному комплексі і ситуаційному центрі системи відеоспостереження до інтерфейсу кінцевого устаткування.

Вимоги до ситуаційного центру системи відеоспостереження

Ситуаційний центр системи відеоспостереження повинен забезпечувати виконання завдань з управління процесами функціонування всіх складових частин системи відеоспостереження і нагляд за їх характеристиками, здійснення контролю за відеопотоками, що передаються з веб-камер, параметрів функціонування програмно-апаратних комплексів і складових частин центрального серверного комплексу, а також стану підключення обладнання.

У разі виявлення некоректного функціонування системи відеоспостереження ситуаційний центр повинен визначити джерело несправності.

Функціональними вимогами до ситуаційного центру системи відеоспостереження є:

- приймання діагностичної інформації про складові частини системи відеоспостереження;
- відображення діагностичної інформації на всіх пристроях системи відеоспостереження;
- здійснення нагляду за змінами заданого набору параметрів;
- забезпечення зворотного зв'язку із службою технічної підтримки для розв'язання проблем.

Моніторинг центрального серверного комплексу

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

Моніторинг центрального серверного комплексу здійснюється шляхом внутрішнього контролю за працездатністю серверного обладнання.

Серверний комплекс повинен мати такі параметри:

- стан використання пам'яті;
- доступність у мережі;
- кількість зайнятого місця за розділами жорстких дисків;
- кількість підключених каналів (камер).

Моніторинг програмно-апаратного комплексу

Під час проведення моніторингу програмно-апаратного комплексу встановлюється наявність таких відомостей:

1. У період здійснення відеотрансляцій для кожної об'єкту кібербезпеки для відеоспостереження на базі бездротових мереж:

- номер об'єкту кібербезпеки для відеоспостереження на базі бездротових мереж;
- місце її розташування;
- мережева адреса об'єкту кібербезпеки для відеоспостереження на базі бездротових мереж;
- номінальна швидкість доступу до мережі передачі даних;
- фактична швидкість доступу до мережі передачі даних;
- наявність доступу програмно-апаратного комплексу до мережі передачі даних;
- тривалість періоду доступу програмно-апаратного комплексу до мережі передачі даних або його відсутності;
- час початку зйомки і збирання записів;
- час закінчення зйомки і збирання записів;
- сумарна тривалість зібраних записів (час);
- сумарний об'єм зібраних записів (Мбайт).

2. У період здійснення відеотрансляцій для повідомлень про проблеми, що виникають під час перегляду відеотрансляцій:

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

- кількість повідомлень про проблеми перегляду відеотрансляцій;
- кількість відеокамер, стосовно яких надійшли повідомлення про проблеми, що виникають під час перегляду відеотрансляцій;
- кількість відеокамер, стосовно яких повідомлення про проблеми перегляду відеотрансляцій були підтверджені.

3. У період здійснення відеотрансляцій для єдиного порталу:

- доступність/недоступність єдиного порталу в Інтернеті (період проведення перевірки – не більше 10 хвилин);
- тривалість періоду доступності єдиного порталу в Інтернеті або її відсутності.

4. У період зберігання зібраних записів відеотрансляцій:

- сумарна тривалість записів відеотрансляцій, зібраних у процесі роботи системи відеоспостереження (час);
- сумарний об'єм записів відеотрансляцій, зібраних у процесі роботи системи відеоспостереження (Мбайт);
- сумарна тривалість записів відеотрансляцій, що зберігаються на відповідний час;
- сумарний об'єм записів відеотрансляцій на даний час (Мбайт).

Вимоги до забезпечення цілісності інформаційних ресурсів та захисту інформації від несанкціонованого доступу

Для унеможливлення несанкціонованого доступу до системи відеоспостереження, використання інформації не за призначенням та порушення її цілісності створюється система захисту інформації від несанкціонованого доступу.

Захист інформації в системі відеоспостереження забезпечується відповідно до Правил забезпечення захисту інформації в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах, затверджених постановою Кабінету Міністрів України від 29 березня 2006 р. № 373.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

Захист інформації здійснюється за такими принципами:

– безперервність процесу захисту шляхом забезпечення захисту інформації від несанкціонованого доступу протягом усього періоду функціонування системи відеоспостереження;

– керованість процесу захисту шляхом забезпечення оперативних режимів контролю за станом виконання правил доступу до системи відеоспостереження та прийняття відповідних рішень у разі порушення таких правил;

– комплексне використання засобів та методів захисту інформації від несанкціонованого доступу шляхом спільного використання технічних, програмних, організаційних засобів захисту, а також відповідних правил доступу до інформації.

Система захисту інформації від несанкціонованого доступу повинна виконувати такі функції:

– ідентифікація користувачів мережі, захист параметрів ідентифікації користувачів і параметрів;

– розмежування доступу до ресурсів і процесів у системі відеоспостереження;

– забезпечення цілісності інформаційних ресурсів;

– захист носіїв інформації від несанкціонованого доступу;

– забезпечення функцій адміністрування в системі захисту інформації від несанкціонованого доступу;

– захист від шкідливих комп'ютерних програм.

Необхідно визначити комплекс адміністративних заходів для дотримання відповідних вимог до рівня надійності приміщень, в яких розташовуються елементи системи відеоспостереження, та регламентації дій персоналу.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

3.4 Розробка діаграми процесів

Діаграма взаємодії процесів розробленої системи зображена на рисунку 3.3. Процеси взаємодіють наступним чином. Спершу запускається процес виведення головного вікна програми. Цей процес взаємодіє з наступними процесами:

- Процес ведення та перегляду журналу подій.
- Процес відслідковування руху.
- Процес встановлення фільтру.
- Процес вибору пристроїв відеоспостереження.
- Процес встановлення загальних параметрів.

Процес відслідковування руху взаємодіє з процесом підняття тривоги, у випадку несанкціонованого руху на охороняє мій території, або об'єкті.

Процес встановлення фільтру взаємодіє з наступними процесами:

- Процес встановлення умови підняття тривоги.
- Процес встановлення області контролю.
- Процес встановлення області чутливості реагування системи на рухи.

Процес вибору пристроїв відеоспостереження взаємодіє з наступними процесами:

- Процес перегляду раніше записаного відео.
- Процес активізування пристрою відеоспостереження.

Останній процес взаємодіє з наступними процесами:

- Процес відображення області контролю.
- Процес відображення поточного відео.
- Процес запису відео.

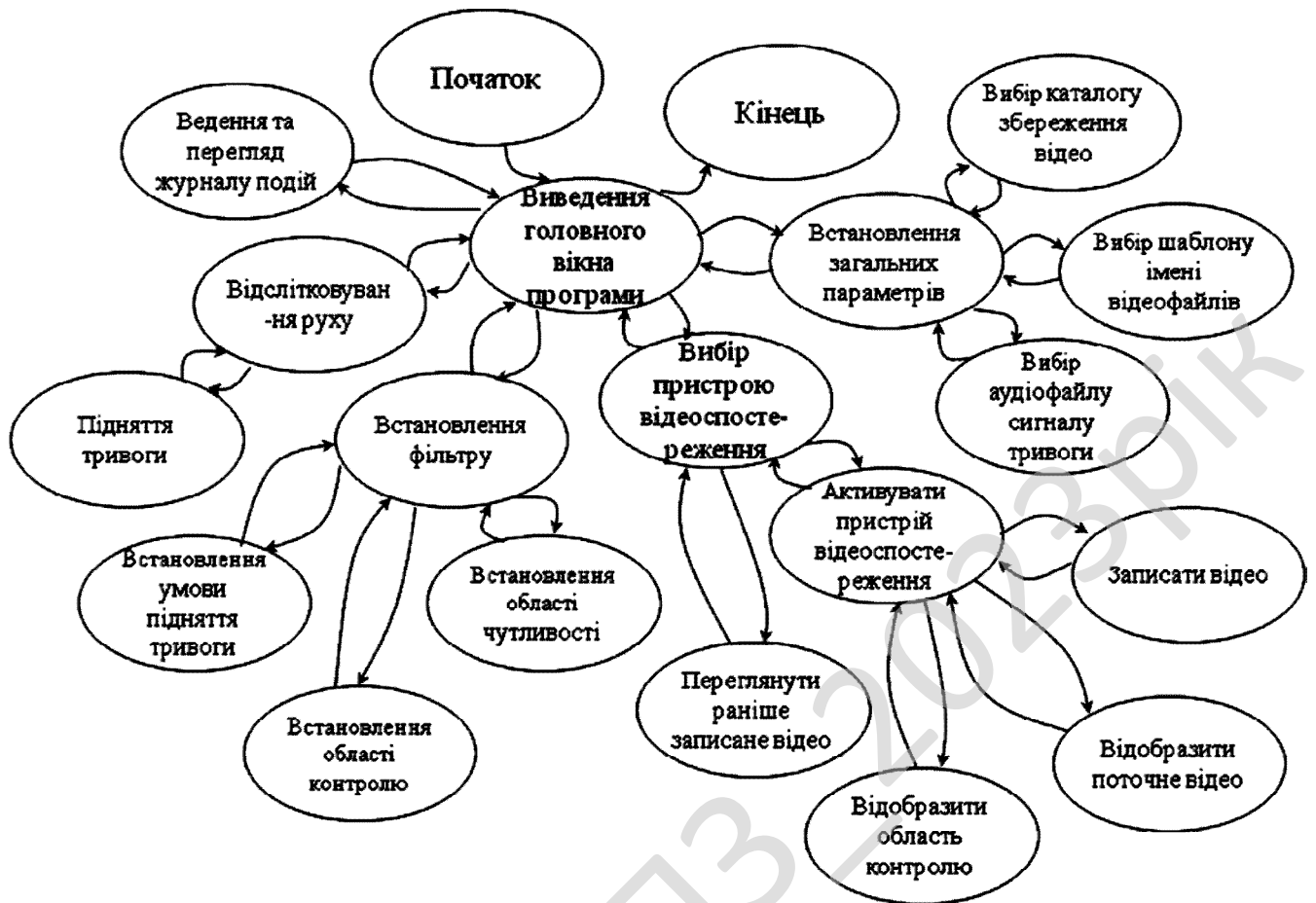


Рисунок 3.3 – Діаграма взаємодії процесів

Процес встановлення загальних параметрів взаємодіє з наступними процесами:

- Процес вибору каталогу для збереження відео.
- Процес вибору шаблону імені відеофайлів.
- Процес вибору аудіофайлу сигналу тривоги.

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

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо алгоритм роботи основної програми, блок-схема якої, зображена на рисунку 4.1.

З рисунку видно, що після запуску програми спочатку відбувається виведення основного вікна програми. Потім здійснюється вибір пристрою відеоспостереження. Якщо пристрій вибрано, пропонується переглянути раніше записане відео з вибраного пристрою.

Після цього пропонується виконати активацію пристрою спостереження. Якщо активація не виконана, можливо провести операції по налаштуванню системи.

При активації пристрою спостереження відбувається виконання наступних операцій:

- Виведення відеозображення з пристрою на екран.
- За потреби відображається область контролю, відзначена у налаштуваннях фільтрації та відбувається виведення на екран цієї області.
- За потреби відбувається запис відео з пристрою на жорсткий диск.
- Якщо в процесі роботи система помічає підозрілий рух, видається сигнал тривоги.

Також програма дозволяє виконувати наступні дії:

- Встановити параметри фільтрації відео.
- Змінити загальні параметри програми.
- Переглянути журнал подій.
- Записати дію та час її виникнення у журнал подій.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

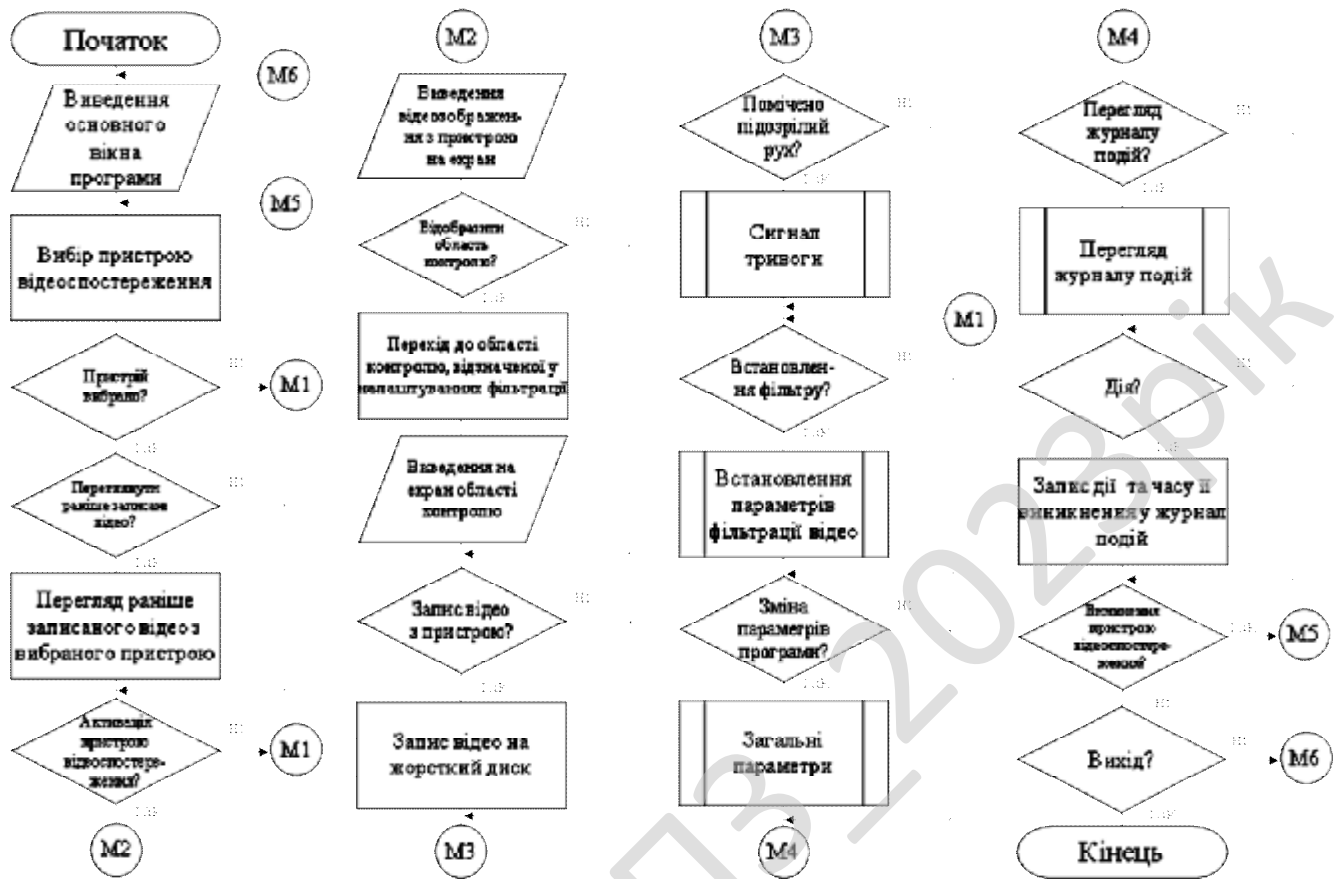


Рисунок 4.1 – Блок-схема основної програми

Наведемо частину коду підпрограми вибору пристрою для відеоспостереження:

```

TSetDeviceForm *SetDeviceForm;
//-----
--
__fastcall TSetDeviceForm::TSetDeviceForm(TComponent* Owner)
: TForm(Owner)
{
}
//-----
--
void __fastcall TSetDeviceForm::FormShow(TObject *Sender)
{
    //зберегти поточні налаштування
    Sets.SaveToFile("prev.cfg");
}

```

```

UpdateState();
// перерахування пристроїв
char szDeviceName[80];
char szDeviceVersion[80];
DeviceSelect->Items->Clear();
for (int wIndex = 0; wIndex < 10; wIndex++)
{
    if (capGetDriverDescription (wIndex, szDeviceName,
        sizeof (szDeviceName), szDeviceVersion,
        sizeof (szDeviceVersion)))
    {
        // Додати ім'я до списку встановлених драйверів захвату
        // та потім дозволити користувачеві вибрати драйвер для
використання
        DeviceSelect->Items-
>Add(AnsiString(szDeviceName)+" (" +AnsiString(szDeviceVersion)+" )");
        if(Sets.DevIndex == wIndex) DeviceSelect->Text =
AnsiString(szDeviceName)+" (" +AnsiString(szDeviceVersion)+" )";
    }
}
}
//-----
--
void __fastcall TSetDeviceForm::DeviceSelectSelect(TObject *Sender)
{
    //update device index
    Sets.DevIndex = DeviceSelect->ItemIndex;
    UpdateState();
}
//-----
--
void __fastcall TSetDeviceForm::OkBitBtnClick(TObject *Sender)
{
    DeleteFile("prev.cfg");
    Close();
}
//-----
--
void __fastcall TSetDeviceForm::CancelBitBtnClick(TObject *Sender)
{
    Sets.LoadFromFile("prev.cfg");
    DeleteFile("prev.cfg");
}

```

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

```

    Close();
}
//-----
--
void __fastcall TSetDeviceForm::VideoSrcBitBtnClick(TObject *Sender)
{
    capDlgVideoSource (Sets.hCaptureW);
}
//-----
--
void __fastcall TSetDeviceForm::FormatBitBtnClick(TObject *Sender)
{
    capDlgVideoFormat (Sets.hCaptureW);
    capGetStatus (Sets.hCaptureW, &Sets.CapStatus, sizeof (CAPSTATUS));

    Sets.VideoFormatSize = capGetVideoFormatSize (Sets.hCaptureW);
    if (Sets.VideoFormat) {
        delete[] Sets.VideoFormat;
        Sets.VideoFormat = NULL;
    }
    if (Sets.VideoFormatSize > 0) {
        Sets.VideoFormat = (LPBITMAPINFO) new BYTE [Sets.VideoFormatSize];
        capGetVideoFormat (Sets.hCaptureW, Sets.VideoFormat,
Sets.VideoFormatSize);
    }
}
//-----
--
void __fastcall TSetDeviceForm::DisplayBitBtnClick(TObject *Sender)
{
    capDlgVideoDisplay (Sets.hCaptureW);
}
//-----
--
void TSetDeviceForm::UpdateState ()
{
    //під'єднати новий пристрій
    capDriverDisconnect (Sets.hCaptureW);
    bool fOK = capDriverConnect (Sets.hCaptureW, Sets.DevIndex);
    if (fOK) {
        //отримати заголовки драйверу

```

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

```

        capDriverGetCaps (Sets.hCaptureW, &Sets.CapDrvCaps, sizeof
(CAPDRIVERCAPS));
        //отримати заголовки вікна
        capGetStatus (Sets.hCaptureW, &Sets.CapStatus, sizeof (CAPSTATUS));
        //отримати доступні діалоги для драйверу
        // діалогове вікно для джерела відеосигналу
        VideoSrcBitBtn->Visible = false;
        if (Sets.CapDrvCaps.fHasDlgVideoSource)
            VideoSrcBitBtn->Visible = true;
        // діалогове вікно для формату відео
        FormatBitBtn->Visible = false;
        if (Sets.CapDrvCaps.fHasDlgVideoFormat)
            FormatBitBtn->Visible = true;
        // діалогове вікно для відеозображення
        DisplayBitBtn->Visible = false;
        if (Sets.CapDrvCaps.fHasDlgVideoDisplay)
            DisplayBitBtn->Visible = true;
    }
}

```

Окремо розглянемо роботу підпрограми встановлення параметрів фільтрації відео. Її блок-схема наведена на рисунку 4.2.

Після виведення в основній програмі вікна параметрів фільтрації можна провести наступні налаштування:

- Вибрати область контролю після виведення на екран зображення з камери.
- Вибрати область чутливості.
- Вибрати умови збереження зображення та подання сигналу тривоги.

Крім області чутливості також вибираються значення чутливості у процентах та квадрати чутливості.

Для вибору умов збереження зображення та подання сигналу тривоги визначаються такі параметри, як період захвату, поріг руху, час руху, період збереження зображення.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

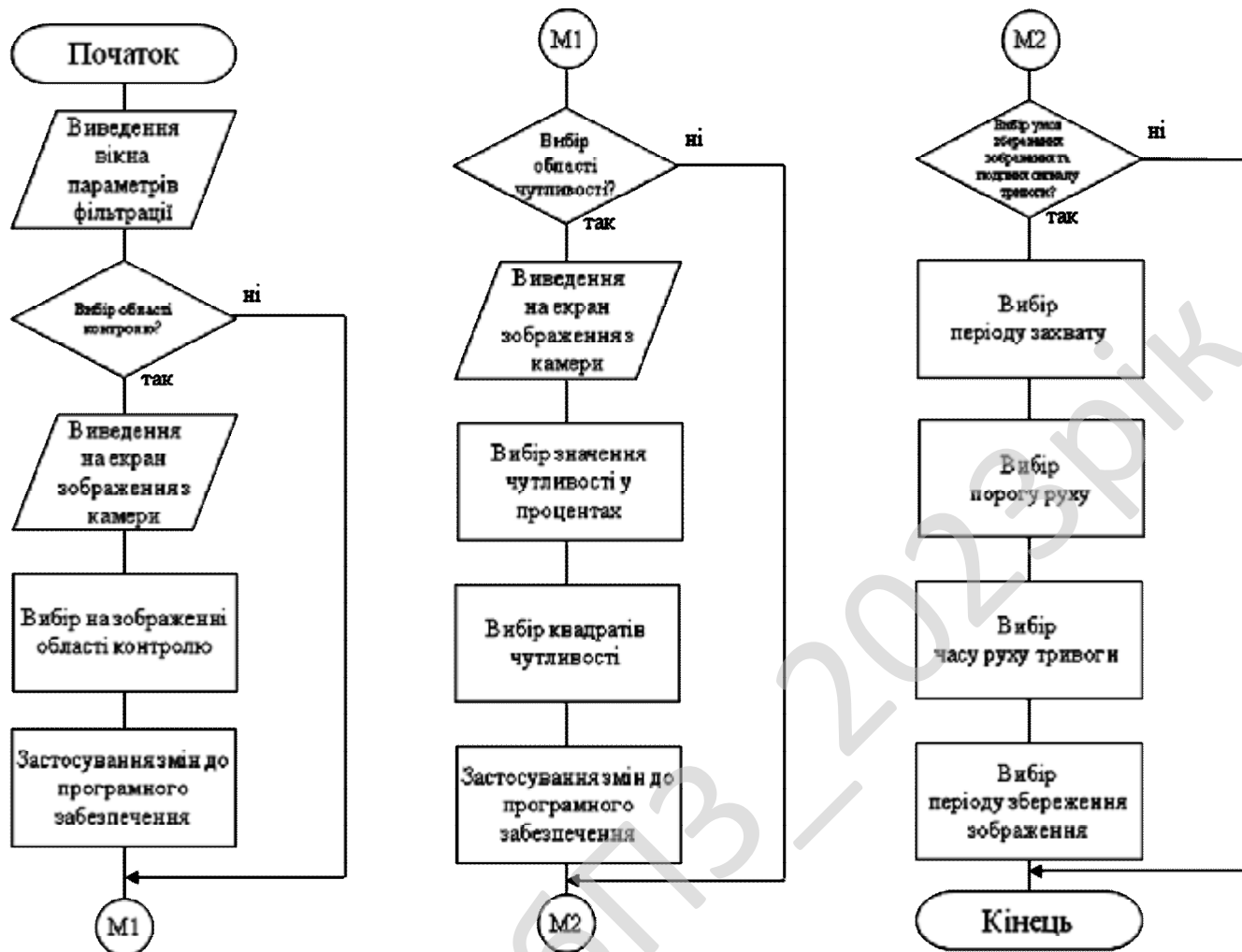


Рисунок 4.2 – Блок-схема підпрограми встановлення параметрів фільтрації відео

Наведемо частину коду підпрограми для фільтрації відеозображення:

```

TSetFilterForm *SetFilterForm;
int SRX1, SRX2, SRY1, SRY2;
int CurSens = -1;
//-----
--
__fastcall TSetFilterForm::TSetFilterForm(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
--

```

```

void __fastcall TSetFilterForm::OkBitBtnClick(TObject *Sender)
{
    DeleteFile("prev.cfg");
    Close();
}
//-----
--
void __fastcall TSetFilterForm::CancelBitBtnClick(TObject *Sender)
{
    Sets.LoadFromFile("prev.cfg");
    DeleteFile("prev.cfg");
    Close();
}
//-----
--
void __fastcall TSetFilterForm::SRPBPaint(TObject *Sender)
{
    DrawSR();
}
//-----
--
void __fastcall TSetFilterForm::SRPBMouseMove(TObject *Sender,
        TShiftState Shift, int X, int Y)
{
    if(DrawRectButton->Down){
        SRPB->Cursor = crCross;
        if(Shift.Contains(ssLeft)){//натиснута ліва клавіша миші
            SRX2 = X;
            SRY2 = Y;
            DrawSR();
        }
    }
    else{
        SRPB->Cursor = crDefault;
    }
}
//-----
--
void __fastcall TSetFilterForm::SRPBMouseDown(TObject *Sender,
        TMouseButton Button, TShiftState Shift, int X, int Y)
{
    if(Button == mbLeft && DrawRectButton->Down){

```

```

        SRX1 = SRX2 = X;
        SRY1 = SRY2 = Y;
    }
}
//-----
--
void TSetFilterForm::DrawSR()
{
    TPicture *tPic = new TPicture;
    tPic->Bitmap->Width = SRPB->Width;
    tPic->Bitmap->Height = SRPB->Height;
    tPic->Bitmap->Canvas->Brush->Color = (TColor)RGB(100,100,100);
    tPic->Bitmap->Canvas->FillRect(tPic->Bitmap->Canvas->ClipRect);
    tPic->Bitmap->Canvas->Brush->Color = clWhite;
    tPic->Bitmap->Canvas->FillRect(Rect(SRX1, SRY1, SRX2, SRY2));
    tPic->Bitmap->Canvas->CopyMode = cmSrcAnd;
    tPic->Bitmap->Canvas->StretchDraw(tPic->Bitmap->Canvas->ClipRect,
Sets.FramePic->Bitmap);
    SRPB->Canvas->Draw(0, 0, tPic->Bitmap);
    delete tPic;
}
void __fastcall TSetFilterForm::FormShow(TObject *Sender)
{
    //зберегти поточні налаштування
    Sets.SaveToFile("prev.cfg");
    SRPB->Width = Sets.pW;
    SRPB->Height = Sets.pH;
    SensPB->Width = Sets.pW;
    SensPB->Height = Sets.pH;
    if(Sets.ClipRect.bottom > 0){
        SRX1 = Sets.ClipRect.left;
        SRX2 = Sets.ClipRect.right;
        SRY1 = Sets.ClipRect.top;
        SRY2 = Sets.ClipRect.bottom;
    }else{
        SRX1 = SRY1 = 0;
        SRX2 = Sets.pW;
        SRY2 = Sets.pH;
    }
    ActivePointsEdit->Text = IntToStr(Sets.MaxActivePoints);
    AlarmTimeEdit->Text = IntToStr(Sets.AlarmMotionTime);
    CycleTimeEdit->Text = IntToStr(Sets.CycleTime);
}

```

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

```

ForceSaveEdit->Text = IntToStr(Sets.ForceSaveTime);
}
//-----
--
void __fastcall TSetFilterForm::RectSaveButtonClick(TObject *Sender)
{
    int t;
    if(SRX1 > SRX2){
        t = SRX1; SRX1 = SRX2; SRX2 = t;
    }
    if(SRY1 > SRY2){
        t = SRY1; SRY1 = SRY2; SRY2 = t;
    }
    Sets.ClipRect.left = SRX1;
    Sets.ClipRect.right = SRX2;
    Sets.ClipRect.top = SRY1;
    Sets.ClipRect.bottom = SRY2;
    if(Sets.bConnected) capGrabFrame(Sets.hCaptureW);
}
//-----
--
void __fastcall TSetFilterForm::SensPBPaint(TObject *Sender)
{
    DrawSensRegions();
}
//-----
--
void TSetFilterForm::DrawSensRegions()
{
    int i;
    SensRegion *sR;
    TPicture *tPic = new TPicture;
    tPic->Bitmap->Width = SensPB->Width;
    tPic->Bitmap->Height = SensPB->Height;
    for(i = 0; i < Sets.SensList->Count; i++){
        sR = (SensRegion*)Sets.SensList->Items[i];
        int cl = sR->sens * 255.0 / 100.0;
        tPic->Bitmap->Canvas->Brush->Color = (TColor)RGB(cl, cl, cl);
        tPic->Bitmap->Canvas->FillRect(*sR);
    }
    tPic->Bitmap->Canvas->CopyMode = cmSrcAnd;
    tPic->Bitmap->Canvas->Draw(0, 0, Sets.GrayPic->Bitmap);
}

```

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76


```

//-----
--
void __fastcall TSetFilterForm::SensPBMouseMove(TObject *Sender,
        TShiftState Shift, int X, int Y)
{
    if(CurSens != -1){
        SensPB->Cursor = crHandPoint;
    }else{
        SensPB->Cursor = crDefault;
    }
}
//-----
--
void __fastcall TSetFilterForm::SensClearButtonClick(TObject *Sender)
{
    int i;
    SensRegion *sR;
    for(i = 0; i < Sets.SensList->Count; i++){
        sR = (SensRegion*)Sets.SensList->Items[i];
        sR->sens = 90;
    }
    DrawSensRegions();
}
//-----
--
void __fastcall TSetFilterForm::AlarmApplyBitBtnClick(TObject *Sender)
{
    try{
        Sets.MaxActivePoints = StrToInt(ActivePointsEdit->Text.Trim());
        Sets.AlarmMotionTime = StrToInt(AlarmTimeEdit->Text.Trim());
        Sets.CycleTime = StrToInt(CycleTimeEdit->Text.Trim());
        Sets.ForceSaveTime = StrToInt(ForceSaveEdit->Text.Trim());
    }catch(...){
        ShowMessage("Введене число не входить в допустимий діапазон
значень!");
    }
}

```

Для передачі відеоданих по бездротовій мережі використовується підпрограма, блок-схема якої наведена на рисунку 4.3. Розглянемо детальніше її роботу.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Після підключення до бездротової мережі за допомогою технології nanoNET (802.15.4a) та підключення бази даних користувачів відбувається перевірка цілісності бази даних і доступу до ресурсів. Якщо перевірка на цілісність не пройдена виводиться повідомлення про помилку, звільнення виділених ресурсів, відключення від бездротової мережі та завершення роботи підпрограми.

У випадку успішного проходження перевірки цілісності підпрограма переходить в режим очікування запиту від програмного забезпечення. При надходженні запиту операції по його обробці відбуваються в наступній послідовності:

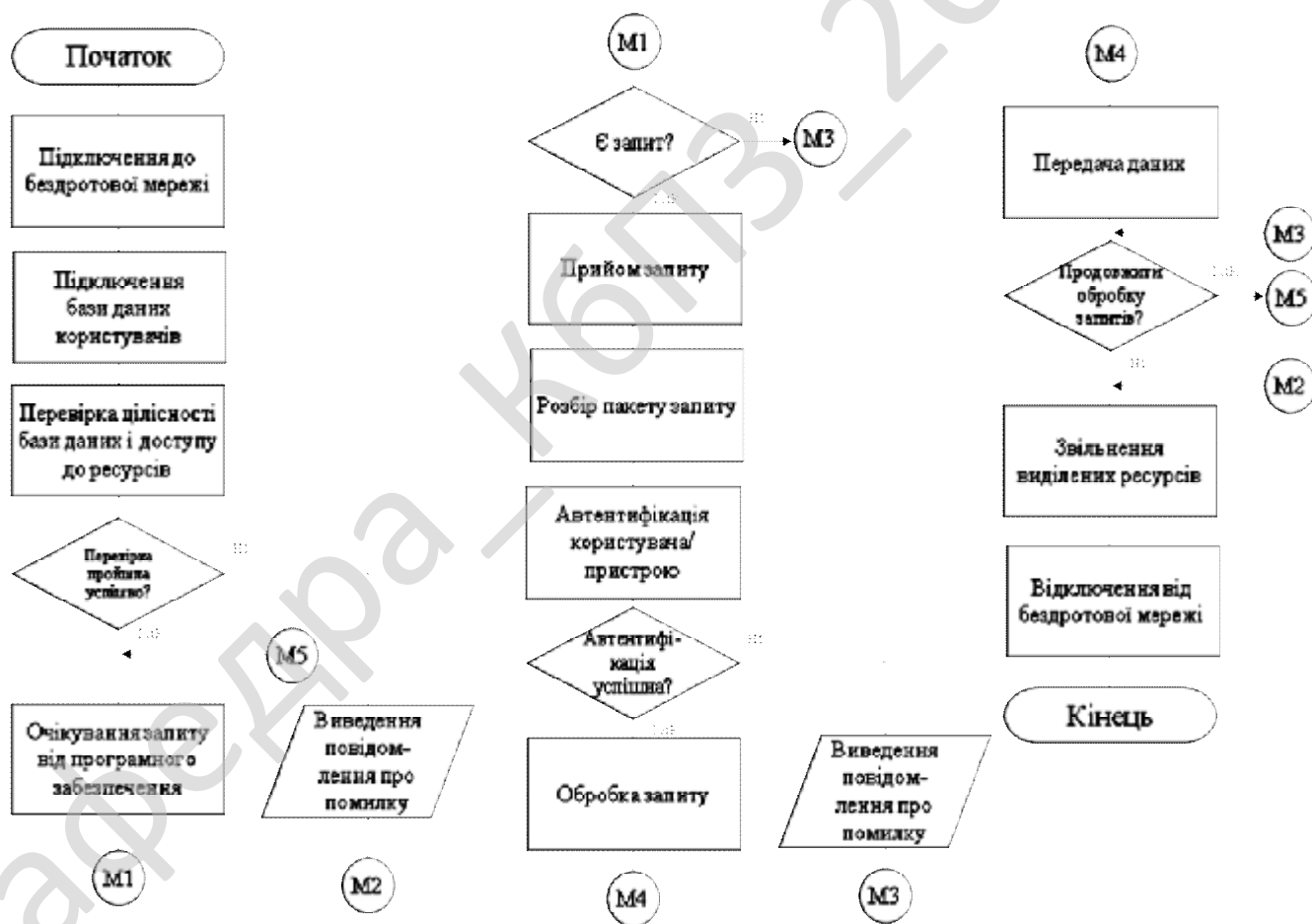


Рисунок 4.3 – Блок-схема підпрограми передачі відеоданих по бездротовій мережі

- Прийом запиту.
- Розбір пакету запиту.
- Автентифікація користувача/пристрою.
- При успішній автентифікації відбувається обробка запиту та передача даних. (В протилежному випадку видається повідомлення про помилку).

При необхідності обробка запитів продовжується, після чого відбувається звільнення виділених ресурсів та відключення від бездротової мережі.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SHACAL-2, який шифрує дані 256-бітними блоками з використанням 512-бітного ключа. Допускається використання ключів менших розмірів (не менш 128 біт), які доповнюються бітовими нулями до 512 біт.

Шифруємий блок даних ділиться на 8 фрагментів по 32 біта (які позначені буквами $A...H$). Алгоритм виконує 64 раунду перетворень, у кожному з яких дані фрагменти обробляються в такий спосіб:

$$T = H_i + S_1(E_i) + Ch(E_i, F_i, G_i) + M_i + K_i,$$

$$H_{i+1} = G_i,$$

$$G_{i+1} = F_i,$$

$$F_{i+1} = E_i,$$

$$E_{i+1} = D_i + T,$$

$$D_{i+1} = C_i,$$

$$C_{i+1} = B_i,$$

$$B_{i+1} = A_i,$$

$$A_{i+1} = T + S_0(A_i) + Maj(A_i, B_i, C_i).$$

де T – тимчасова змінна.

Використовувані функції визначені в такий спосіб:

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

$$S_0(x) = (x \ggg 2) \oplus (x \ggg 13) \oplus (x \ggg 22),$$

$$S_1(x) = (x \ggg 6) \oplus (x \ggg 11) \oplus (x \ggg 25),$$

$$Ch(x, y, z) = (x \& y) \oplus (x' \& z),$$

$$Maj(x, y, z) = (x \& y) \oplus (x \& z) \oplus (y \& z),$$

де \ggg – операція побітового циклічного зрушення вправо.

Константи, що модифікують, M_i ($i = 0 \dots 63$) наведено нижче (одна за одною від M_0 до M_{63}):

428A2F98	71374491	B5C0FBCF	E9B5DBA5
3956C25B	59F111F1	923F82A4	AB1C5ED5
D807AA98	12835B01	243185BE	550C7DC3
72BE5D74	80DEB1FE	9BDC06A7	C19BF174
E49B69C1	EFBE4786	0FC19DC6	240CA1CC
2DE92C6F	4A7484AA	5CB0A9DC	76F988DA
983E5152	A831C66D	B00327C8	BF597FC7
C6E00BF3	D5A79147	06CA6351	14292967
27B70A85	2E1B2138	4D2C6DFC	53380D13
650A7354	766A0ABB	81C2C92E	92722C85
A2BFE8A1	A81A664B	C24B8B70	C76C51A3
D192E819	D6990624	F40E3585	106AA070
19A4C116	1E376C08	2748774C	34B0BCB5
391C0CB3	4ED8AA4A	5B9CCA4F	682E6FF3
748F82EE	78A5636F	84C87814	8CC70208
90BEFFFA	A4506CEB	BEF9A3F7	C67178F2

Фрагменти розширеного ключа $K_0 \dots K_{63}$ обчислюються в процесі процедури розширення ключа, що виконується в такий спосіб:

Етап 1. 512-бітний вихідний ключ шифрування ділиться на 16 фрагментів по 32 біта $K_0 \dots K_{15}$...

Етап 2. Інші фрагменти розширеного ключа $K_{16} \dots K_{63}$ обчислюються з перших 16 фрагментів у такий спосіб:

$$K_i = O_1(K_{i-2}) + K_{i-7} + O_0(K_{i-15}) + K_{i-16},$$

де функції O_0 і O_1 визначені так:

$$O_0(x) = (x \gg \gg 7) \oplus (x \gg \gg 18) \oplus (x \gg 3),$$

$$O_1(x) = (x \gg \gg 17) \oplus (x \gg \gg 19) \oplus (x \gg 10),$$

де \gg – операція побітового зрушення (не циклічного) вправо.

Раунди розшифрування алгоритму виконуються у зворотній послідовності:

$$T = A_{i+1} + S_0'(B_{i+1}) + Maj'(B_{i+1}, C_{i+1}, D_{i+1}) + 2,$$

$$H_i = T + S_1'(F_{i+1}) + Ch'(F_{i+1}, G_{i+1}, H_{i+1}) + M_i' + K_i' + 4,$$

$$G_i = H_{i+1},$$

$$F_i = G_{i+1},$$

$$E_i = F_{i+1},$$

$$D_i = E_{i+1} + T' + 1,$$

$$C_i = D_{i+1},$$

$$B_i = C_{i+1},$$

$$A_i = B_{i+1}.$$

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене програмне забезпечення призначене для ведення кібербезпеки для відеоспостереження на базі бездротових мереж.

Дане програмне забезпечення володіє наступними функціональними можливостями:

- Виконує захват зображень із будь-якого пристрою відеозахвату, підтримуваного Windows XP/Vista/7.
- Виділяє вказану контрольну область із оригінального зображення.
- Робить автоматичний баланс рівнів контрольної області, що забезпечує незалежність роботи алгоритму виявлення від освітленості й рівня посилення відеосигналу.
- Обробляє контрольну область з налаштовуваною чутливістю до руху за 64 підобластями.
- Візуально представляє кожний етап обробки.
- Дозволяє встановити граничні умови для захвату зображення й виклику тривоги.
- При виконанні умов тривоги запускає зазначену програму, командний файл, документ або програє звуковий сигнал.
- При перевищенні порогу виявлення руху зберігає зображення у форматі JPEG.
- Примусово зберігає зображення через зазначені проміжки часу або при натисканні на кнопку **Зняти!**
- Має винятково простий і функціональний інтерфейс без непотрібних для такого роду програм графічних надмірностей.
- Абсолютно повне й досить тонке настроювання всіх параметрів користувачем.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

– Не вимагає для роботи ніяких додаткових файлів і бібліотек, крім стандартних бібліотек Windows, не вносить ніякі свої записи до реєстру або файлів налаштувань ОС.

Інтерфейс головного вікна програми зображений на рисунку 5.1



Рисунок 5.1 – Головне вікно програми, закладка «Початковий кадр», що відображає кадр, безпосередньо отриманий із пристрою відеозахвату

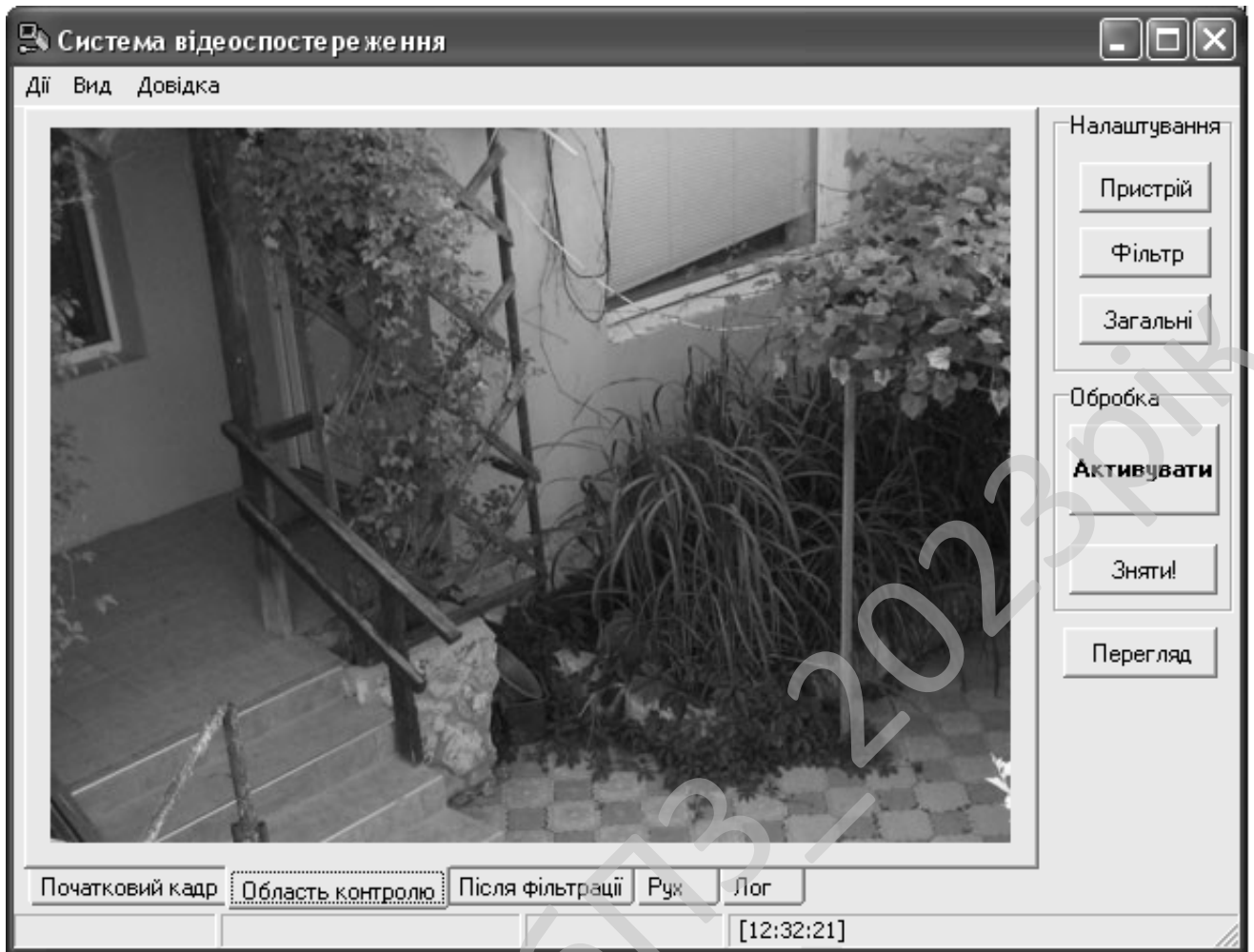


Рисунок 5.2 – Головне вікно програми, закладка **Область контролю**, що відображає зображення контрольованої області після виконання процедури автобалансування



Рисунок 5.3 – Головне вікно програми, Закладка **Після фільтрації**, що відображає повністю оброблене зображення відповідно до установок користувацького фільтра руху

Закладка **Рух** відображає області, де в поточний момент спостерігається рух. Закладка **Лог** відображає інформацію про статус, помилки драйвера відеозахвату й виконаних програмою дій.

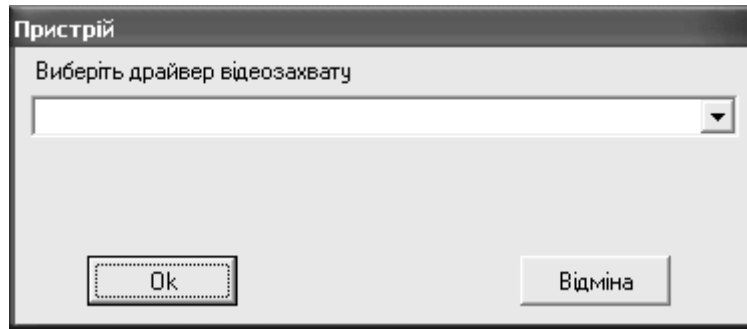


Рисунок 5.4 – Діалогове вікно **Пристрій**, в якому обирається пристрій відеоспостереження

Натиснувши на кнопку **Фільтр**, відкриваємо вікно з найважливішими для даної програми налаштуваннями.



Рисунок 5.5 – Вікно **Фільтр**, закладка **Область контролю**

Для виділення області контролю слід тиснути кнопку **Задати область** і «намалювати» мишкою прямокутник на зображенні. Все, що всередині прямокутника потрапляє в область контролю. Щоб зміни вступили в дію – треба натиснути кнопку **Застосувати**.

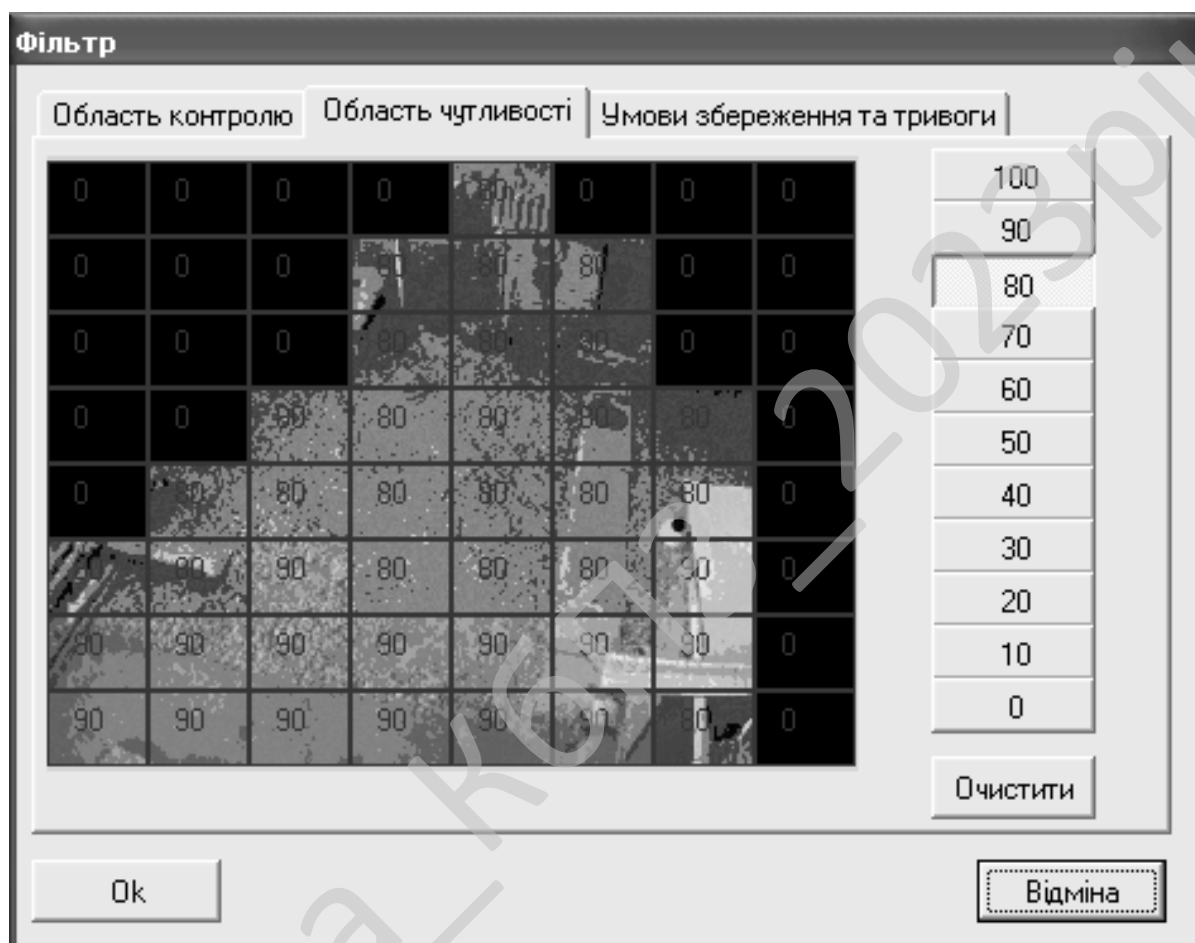


Рисунок 5.6 – Вікно **Фільтр**, закладка **Область чутливості**

На цій закладці встановлюється чутливість до руху окремих областей контрольованого зображення. Для завдання чутливості слід натиснути на кнопку праворуч із потрібним значенням чутливості, а потім клацнути лівою кнопкою миші усередині червоних прямокутників на зображенні. Непотрібні для контролю області необхідно позначити чутливістю 0 (крім усього іншого, це прискорює роботу фільтра).

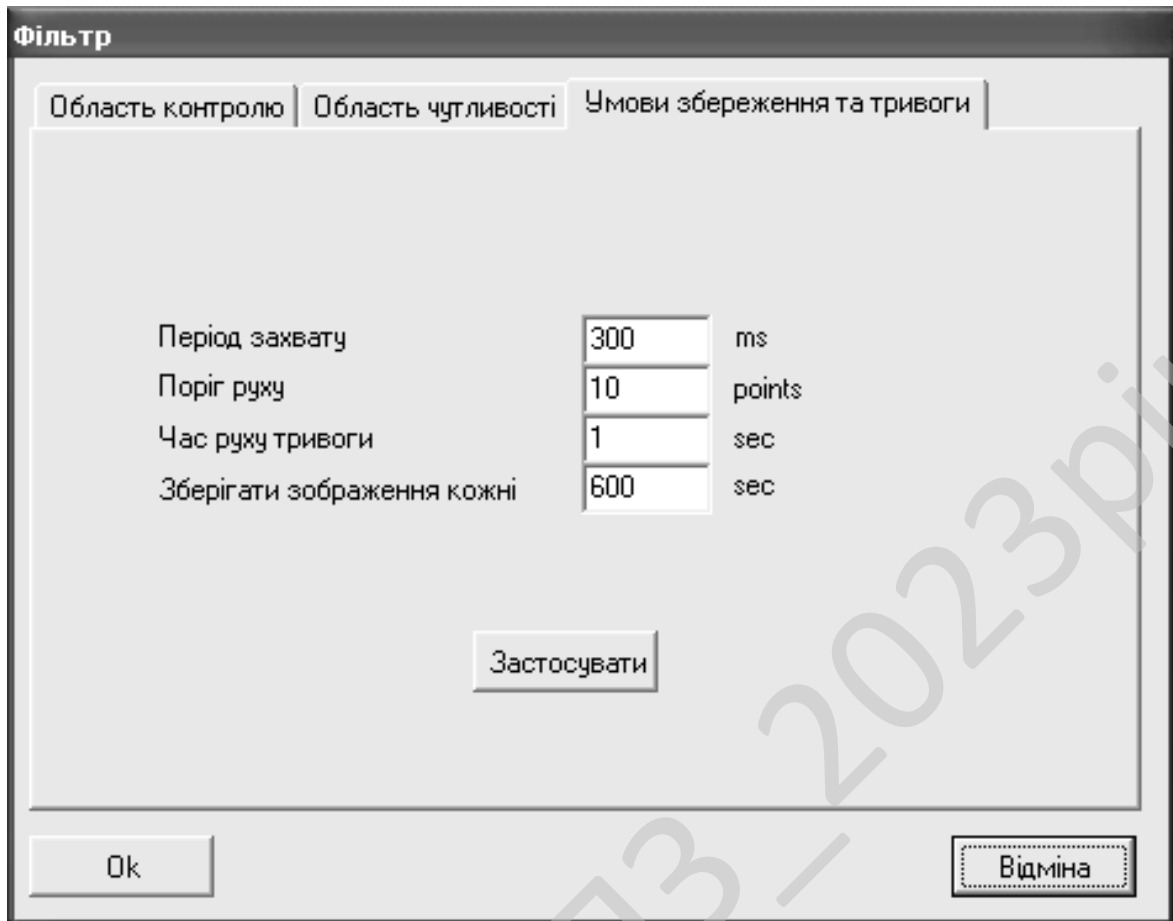


Рисунок 5.6 – Вікно **Фільтр**, закладка **Умови збереження та тривога**

Дана закладка містить налаштування порогів. Значення параметрів наступні:

Період захвату – час між захватами зображення з відеокамери. Встановлюється в мілісекундах. Оптимальне значення – 300. Якщо, наприклад, встановити 2, то програма не зможе виконати таких умов, і буде обробляти зображення з максимально можливою для вашого комп'ютера швидкістю. Впливає на мінімальну швидкість руху об'єкта в кадрі, що виявляється програмою. Чим менше час циклу, тим вище мінімальна швидкість руху для досягнення порога виявлення.

Поріг руху – мінімальна кількість точок, у яких програма засікла рух і повинна виконати збереження картинки на диск. Від цього параметра залежить мінімальний розмір об'єкта, що рухається, що буде помічений (наприклад, щоб

виявити курку досить 20 точок, а пропустити курку, але помітити людину – приблизно 100 точок). Конкретні значення залежать від масштабів зображення.

Час руху тривоги – мінімальний час, протягом якого безупинно засікається рух вище порога, заданого попереднім параметром, через який треба виконувати процедуру тривоги. Вказується в секундах, ефективно фільтрує випадкові захвати через рух гілок дерев при сильних поривах вітру або повільних пересувань домашніх тварин, наприклад, собаки.

Зберігати зображення кожні... – час у секундах, через який зображення буде зберігатися на диску в «примусовому» порядку.

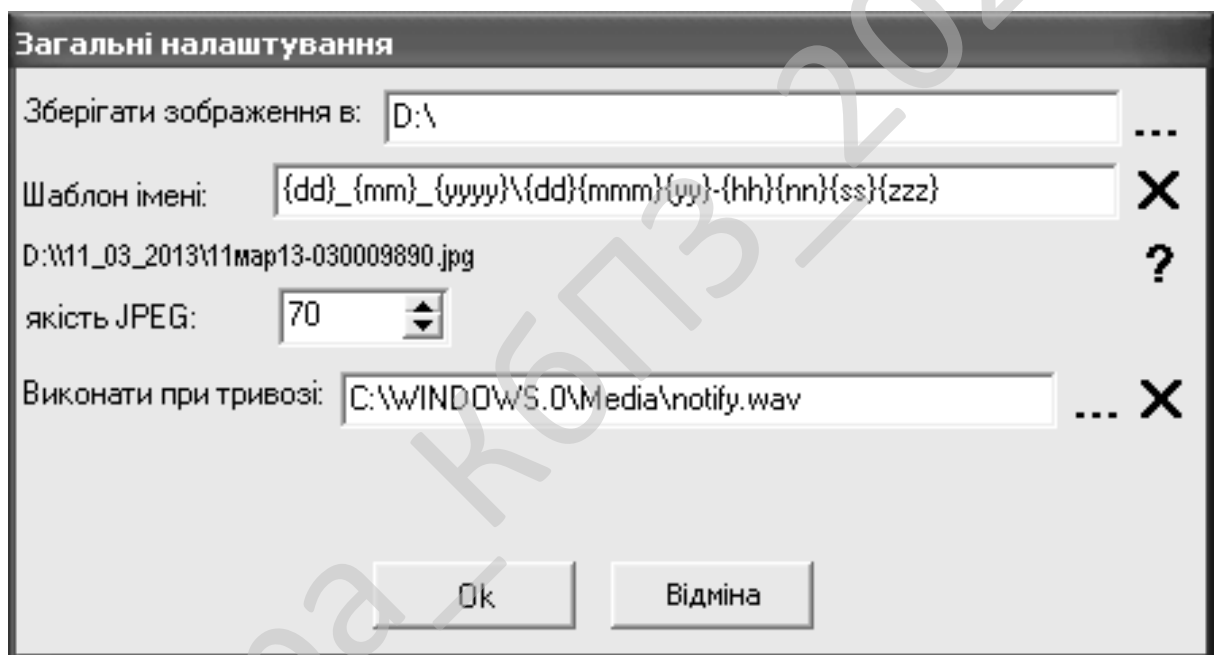


Рисунок 5.7 – Вікно **Загальні налаштування**

Дане вікно з'являється при натисканні на кнопку **Загальні**. У ньому потрібно вказати папку, у яку буде здійснюватися збереження зображень, якість збережених зображень (від 5 до 100, 100 – максимальна якість, але й максимальний розмір файлів).

Кнопки з трикрапками відкривають діалоги вибору папки для збереження картинок і програми обробки ситуації тривоги.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

Виконати при тривозі... – обрання дії, яку програма буде виконувати при настанні умов тривоги. Якщо вибрати аудіофайл – програма програє записану у ньому мелодію, у всіх інших випадках спробує виконати вибраний файл.

На рисунку 5.8 зображене вікно «Про програму...» з короткою довідкою про розроблене програмне забезпечення.

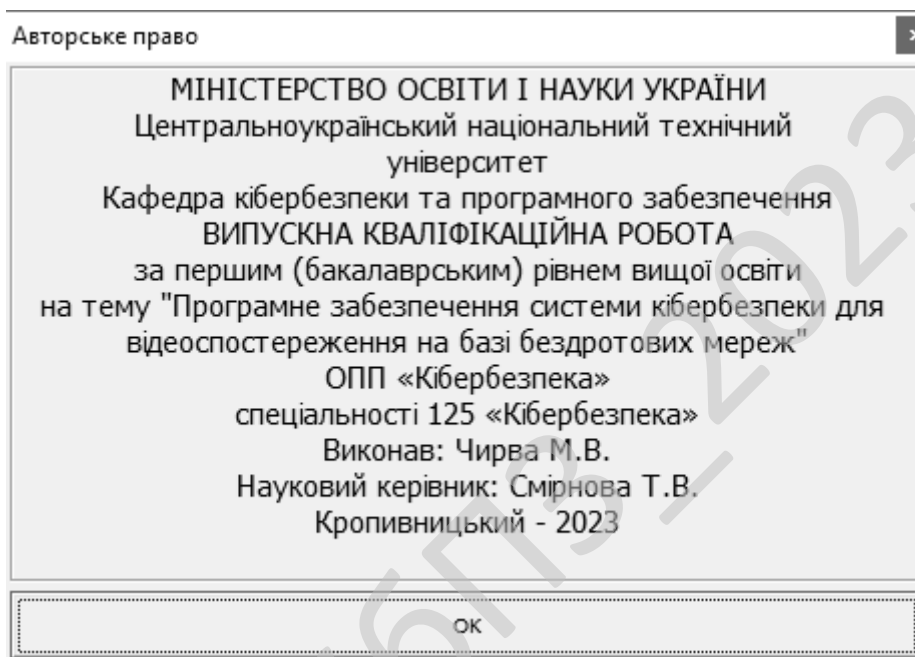


Рисунок 5.8 – Вікно «Про програму...»

Для початку роботи з програмою необхідно:

1. Переконайтеся в тому, що у вас є правильно встановлений пристрій та встановлені драйвера до нього для захвату зображень у системі. Такими пристроями є: вебкамери, ТВ-тюнери, відеокарти з відеовходом, цифрові фотоапарати в режимі вебкамери.
2. Підключити до відеовходу пристрою захвату джерело сигналу (аналогову відеокамеру). Вебкамер і цифрових фотоапаратів це не стосується.
3. Запустити програму. Задати налаштування пристрою, фільтра, збереження зображень і дій при тривозі.

4. Натиснути на кнопку **Активувати**. Програма почне захват й обробку зображень.

5. Якщо виникне помилка, переглянути журнал системних подій, що перебуває на закладці **Лог**. Спробувати змінити налаштування. Налаштування застосовуються «на льоту», тобто перезапуску програми не вимагають. Програма сама виконує всі необхідні дії для переініціалізації. Налаштування зберігаються автоматично, при виході із програми (це не стосується специфічних налаштувань пристроїв відеозахвату).

6. Переглянути безпосередній результат роботи програми можна, натиснувши кнопку **Перегляд** – відкриється вікно провідника для папки, у яку зберігаються зображення.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи кібербезпеки для відеоспостереження на базі бездротових мереж.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем для відеоспостереження на базі бездротових мереж.

– Досліджена система для відеоспостереження на базі бездротових мереж.

– На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки для відеоспостереження на базі бездротових мереж.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання для відеоспостереження на базі бездротових мереж.

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 Builder C++. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи кібербезпеки для відеоспостереження на базі бездротових мереж. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід,

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

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 кібербезпеки й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи кібербезпеки Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм SHAKAL-2.

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

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		94

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гриньов Д.В. Методи стиснення зображень в системах цифрової обробки даних / Д.В. Гриньов, З.З. Закіров // Системи обробки інформації. – Х.: ХУПС, 2010. – Вип. 2(83). – С. 66-70.

2. Гук М.Ю. Аппаратные средства локальных сетей. Энциклопедия / Михаил Юрьевич Гук – СПб.: Издательство “Питер”, 2000. – 576 с.

3. Донченко В.С. Теорія ймовірностей та математична статистика: навчальний посібник / В.С. Донченко, М.В.-С. Сидоров, М.М. Шарапов – К.: ВЦ «Академія», 2009. – 288 с.

4. Дреев А.Н. Использование неравномерного распределения единичных битов для дополнительного сжатия SPIHT кода / А.Н. Дреев, А.А. Смирнов // Информационные системы в управлении, образовании, промышленности: монография. Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2014. – С. 498.

5. Дреев О.М. Дослідження впливу шляху розгортки на ступінь ентропійного стиснення цифрового зображення / О.М. Дреев, О.В. Слюсар // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. Випуск 21. – Кіровоград: КНТУ. – 2008 – С. 115-118.

6. Дреев О.М. Метод розвантаження телекомунікаційного сервера за рахунок кешування зображень / О.М. Дреев // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. Випуск 25. Ч. I. – Кіровоград: КНТУ. – 2012 – С. 419-424.

7. Дреев О.М. Метод прогнозування завантаженості серверу телекомунікаційної мережі / О.М. Дреев, О.А. Смірнов, Є.В. Мелешко,

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

О.В. Коваленко // Системи обробки інформації. Випуск 3(101) Том 2. – Х.: ХУПС. – 2012. – С. 181-188.

8. Дреєв О.М. Оцінка якості стиснення зображень на основі дискретного перетворення Хартлі / О.В. Коваленко, О.П. Доренський, О.М. Дреєв // Системи озброєння і військова техніка. Науковий журнал 2(34) – Х.: ХУПС – 2013. С. 99-102.

9. Дреєв О.М. Дослідження впливу ступеня стиснення зображень на оперативність їх доставки у телекомунікаційній системі / О.А. Смирнов, О.М. Дреєв, О.П. Доренський // Збірник наукових праць "Системи обробки інформації". – Випуск 8(115). – Х.: ХУПС – 2013. – С. 234-239.

10. Дреєв А.Н. Сравнение битовых плотностей при использовании различных методов кодирования информации / А.Н. Дреєв, А.А. Смирнов // Системи обробки інформації, 2014, випуск 2 (118), том 2 – Харків: ХУПС – 2014. С 64-66.

11. Дреєв О.М. Моделювання впливу інтенсивності трафіку на оперативність доставляння інформації / О.М. Дреєв // Науково-виробничий журнал "Зв'язок". – Київ: ДУТ, 2014. – № 2 (108) С. 24-29.

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

13. Дреєв О.М. Узагальнення вейвлету Хаара / О.М. Дреєв, Г.М. Дреєва // Збірник тез доповідей Комбінаторні конфігурації та їх застосування, 15-16 жовтня 2010 р. – Кіровоград – С. 58

14. Дреєв О.М. Узагальнення вейвлету Хаара / О.М. Дреєв // Матеріали науково-практичної конференції, присвяченої 80-річчю фізико-математичного факультету КДПУ ім. В. Винниченка 26 листопада 2010 р. – Кіровоград – С. 12

15. Дреєв О.М. Метод прогнозування завантаженості серверу телекомунікаційної мережі / О.М. Дреєв, О.В. Коваленко // Тези доповідей Новітні

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

технології – для захисту повітряного простору. Дев'ята наукова конференція. 18-19 квітня 2011 р. – Х.: ХУПС. – 2012. – С. 206

16. Дреєв О.М. Метод довгострокового прогнозування навантаження серверу телекомунікаційної мережі / О.М. Дреєв, Г.М. Дреєва // Комбінаторні конфігурації та їх застосування. Кіровоград. 13-14 квітня 2012 р. – Кіровоград: “Ексклюзів-систем”. – 2012. – С. 50

17. Дреєв О.М. Вдосконалення стиснення зображень SPIHT методу шляхом додаткового кодування та відкладеної передачі уточнення вейвлет коефіцієнтів / О.М. Дреєв // Дискретна математика та її застосування у економіко-математичному моделюванні та інформаційних технологіях. 11-13 жовтня 2012 р. – Запоріжжя: ЗНУ – 2012. – С. 22-23.

18. Дреєв О.М. Методи підвищення якості обслуговування у телекомунікаційних системах та мережах / О.М. Дреєв, Г.М. Дреєва, О.А. Смірнов // Збірник тез доповідей. Академія внутрішніх військ МВС України “Застосування інформаційних технологій у підготовці та діяльності сил охорони правопорядку” 20-21 березня 2013р. – Харків: АВВ. – 2013. С. – 18-19

19. Дреєв А.Н. SPIHT кодирование с отложенной передачей значимых битов / А.Н. Дреєв // Тези доповідей. Новітні технології – для захисту повітряного простору. Дев'ята наукова конференція 17 квітня 2013 р. – Х.: ХУПС. – 2013. – С. 206

20. Дреєв А.Н. Повышение оперативности доставки данных повышенной востребованности в телекоммуникационных системах и сетях / А.Н. Дреєв, А.А. Смирнов, Е.В. Мелешко // Проблеми і перспективи розвитку ІТ-індустрії 25-26 квітня 2013 р. Системи обробки інформації. – Випуск 3 (110). Том 2. – Харків: ХУПС. – 2013. С. – 199.

21. Дреєв О.М. Середньостатистичний та найімовірніший час доставки багатопакетного повідомлення в телекомунікаційній системі або мережі / О.М. Дреєв, О.А. Смірнов // V Всеукраїнська науково-практична

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

конференція "Інформатика та системні науки" ІСН – 2014, 13-15 березня 2014 року, м. Полтава – С. 92

22. Дреєв О.М. Визначення оптимального розміру блоку при бітовому арифметичному кодуванні / О.М. Дреєв, Г.М. Дреєва // Збірник тез доповідей Комбінаторні конфігурації та їх застосування, 11-12 квітня 2014 р. – Кіровоград – С. 44

23. Дреєв А.Н. Экстраполяция квазипериодических процессов с аддитивными помехами / А.Н. Дреєв, А.А. Смирнов // П'ята Міжнародна науково-практична конференція "Інформаційні технології та моделювання в економіці" 15-16 травня 2014 р. – Черкаси – С. 59

24. Дреєв А.Н. Статистическая модель передачи многопакетного сообщения в телекоммуникационной системе или сети / А.Н. Дреєв, А.А. Смирнов // «Компьютерное моделирование в наукоемких технологиях (КМНТ-2014)» Харьков, 28-31 мая 2014 года – С. 137-140

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

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

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

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

29. 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>.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

30. Зайченко Ю.П. Компьютерные сети / Ю.П. Зайченко. – К.: Слово, 2003. – 256 с.
31. Зубко Р.А. Алгоритмы сжатия изображений в системах цифровой обработки данных / Р.А. Зубко // Восточно-Европейский журнал передовых технологий. – 2013. – №1/2(61). – С. 40-44.
32. Иванов В.Г. Прогрессивные информационные технологии сжатия изображений / Иванов В. Г., Ломоносов Ю. В., Любарский М. Г. // Проблемы информатики та комп'ютерної техніки (ПІКТ – 2014) : пр. III-ї Міжнар. наук.-практ. конф., 27–30 трав. 2014 р. – Чернівці : Родовід, 2014. – С. 172–174.
33. Ивахненко А.Г. Долгосрочное прогнозирование и управление сложными системами / А.Г. Ивахненко – Киев: «Техніка». – 1975. – 312 с.
34. 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 симпозиум. – Карпати, Вишків: ДУІКТ. – 2011. – С.34.
35. Карпенко С.В. Метод компактного представления изображений в телекоммуникационных системах на основе 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.
36. 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.
37. 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

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

38. Кветний Р.Н. Методи та засоби передавання інформації у проблемно-орієнтованих розподілених комп'ютерних системах: монографія / Р.Н. Кветний, А.Я. Кулик – Вінниця: ВНТУ, 2010. – 362 с.

39. 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.

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

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

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

43. Кононенко В.О. Колебательные системы с ограниченным возбуждением / В.О. Кононенко – М.: Наука, 1964. – 232 с.

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

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

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

47. Кузнецов О.О. Методи обробки сигналів даних та зображень. Навчальний посібник / О.О. Кузнецов, Г.А. Кучук, С.Г. Семенов – Х.: НТУ «ХП», 2011. – 301 с.

48. Кучерявый Е.А. Управление трафиком и качество обслуживания в сети Интернет / Е.А. Кучерявый – М.: Наука и Техника, 2004. – 336 с.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

49. Кучук Г.А. Метод дослідження фрактального мережевого трафіка / Г.А. Кучук // Системи обробки інформації. – Х.: ХВУ, 2005. – Вип. 5 (45). – С. 74-84.

50. Кучук Г.А. Метод агрегування фрактального трафіка / Г.А. Кучук, А.А. Можаяев, О.В. Воробьов // Радіоелектронні та комп'ютерні системи. – 2006. – №6(18). – С. 181-188.

51. Лемешко А.В. Поточковые модели многоадресной и широковещательной маршрутизации в телекоммуникационных сетях [Электронный ресурс] / А.В. Лемешко, К.М. Арус // Проблеми телекомунікацій. – 2013. – No 1 (10), с. 38 - 45. – Режим доступа: http://pt.journal.kh.ua/2013/1/1/131_lemeshko_multicast.pdf.

					ВКРБ-125.23.0039.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

Додаток А
(обов'язковий)

Технічне завдання

Зміст

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

					ВКРБ-125.23.0039.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Чирва М.В.				<i>Програмне забезпечення системи кібербезпеки для відеоспостереження на базі бездротових мереж</i>	Літ.	Аркуш	Аркушів
Перевірів	Смірнова Т.В.					Б	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КБ-20-3СК			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи кібербезпеки для відеоспостереження на базі бездротових мереж.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 13-02 від 5.01.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи кібербезпеки для відеоспостереження на базі бездротових мереж.

4 Джерела розробки

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

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

					ВКРБ-125.23.0039.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

5.2 Показники призначення

Система повинна забезпечувати:

- системи кібербезпеки для відеоспостереження на базі бездротових мереж;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

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

5.4 Вимоги до архітектури

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

5.5 Вимоги до надійності

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

					ВКРБ-125.23.0039.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

5.7 Вимоги до складу та параметрів технічних засобів

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

5.8 Вимоги до інформаційної і програмної сумісності

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

5.8.1 Обладнання

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

5.8.2 Мова програмування

Середовище Builder C++.

					ВКРБ-125.23.0039.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

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

7 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 101 аркуші.

8 Етапи розробки

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

					ВКРБ-125.23.0039.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

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

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

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

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

11.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 6.06.2023 р.

					ВКРБ-125.23.0039.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
першим (бакалаврським) рівнем вищої освіти
_____ Смірнова Т.В.

*Програмне забезпечення системи кібербезпеки для відеоспостереження на
базі бездротових мереж*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 30

Літера: РП

Кропивницький – 2023 року

```
// MainUnit.cpp - основна програма
```

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "MainUnit.h"
#include "SetDeviceUnit.h"
#include "SetFilterUnit.h"
#include "SetCommonUnit.h"
#include "ReConst.hpp"
#include "reinit.hpp"
//-----
#pragma package(smart_init)
#pragma link "trayicon"
#pragma resource "*.dfm"
TMainForm *MainForm;
DevSettings Sets;
bool bWork = false;
AnsiString ConfigName;
//-----
__fastcall TMainForm::TMainForm(TComponent* Owner)
    : TForm(Owner)
{
    ConfigName = ExtractFilePath(Application->ExeName) + "va.cfg";
}
//-----

void __fastcall TMainForm::ActivateButtonClick(TObject *Sender)
{
    bWork = !bWork;
    int worktime = GetTickCount(), frametime = 0;
    int lastsavetime = 0;

    if(bWork && !Sets.Connect()) return;

    Graphics::TBitmap *tbmp = new Graphics::TBitmap;

    if(bWork) ActivateButton->Caption = "Стоп";
    else ActivateButton->Caption = "Активувати";

    while (bWork){
        capGrabFrame(Sets.hCaptureW);
        switch (MainPageControl->TabIndex){
            case 0: // вхідний кадр
                PrevPB->Canvas->StretchDraw(PrevPB->ClientRect, Sets.FramePic->Bitmap);
                break;
            case 1: // контрольний кадр
                Sets.PaintArray(Sets.GrayData, tbmp);
                FramePB->Canvas->StretchDraw(FramePB->ClientRect, tbmp);
                break;
            case 2: // відфільтрований кадр
                Sets.PaintArray(Sets.FiltData, tbmp);
                FiltPB->Canvas->StretchDraw(FiltPB->ClientRect, tbmp);
                break;
            case 3: // кадр із рухом
                Sets.PaintArray(Sets.DifData, tbmp);
                MotionPB->Canvas->StretchDraw(MotionPB->ClientRect, tbmp);
                break;
        }

        if(worktime - lastsavetime > Sets.ForceSaveTime * 1000){
            lastsavetime = worktime;
        }
    }
}
```

```

        Sets.SaveFrame();

    }

    StatusBar->Panels->Items[0]->Text = "FPS: " + IntToStr(1000/(frametime ==
0?1:frametime)) + " (" + IntToStr(frametime) + "ms)";
    StatusBar->Panels->Items[1]->Text = "Кадрів оброблено: " +
IntToStr(Sets.FramesCaptured);
    StatusBar->Panels->Items[2]->Text = "Рухомих точок: " +
IntToStr(Sets.ActivePoints);

    frametime = GetTickCount() - worktime;
    while(bWork && frametime < Sets.CycleTime){
        frametime = GetTickCount() - worktime;
        Application->ProcessMessages();
        Sleep(5);
    }
    worktime = GetTickCount();
    Application->ProcessMessages();
}
delete tbmp;
Sets.Disconnect();
}
//-----
void __fastcall TMainForm::DeviceItemClick(TObject *Sender)
{
    if(!Sets.bConnected) Sets.Connect();
    SetDeviceForm->ShowModal();
    Sets.Connect();
}
//-----

void __fastcall TMainForm::FormShow(TObject *Sender)
{
    Sets.hParentW = CaptPanel->Handle;
    Sets.LogMemo = LogMemo;
    Sets.LoadFromFile(ConfigName);

    AnsiString asLcid;
    int iLcid = 0;
    asLcid = LoadStr(_SLcid.id);
    if(asLcid != "") iLcid = asLcid.ToInt();
    if((iLcid != Sets.LocalId) && (Sets.LocalId == ENGLISH || Sets.LocalId ==
RUSSIAN)){
        if(LoadNewResourceModule(Sets.LocalId) != 0)
        {
            ReinitializeForms();
        }
    }
    asLcid = LoadStr(_SLcid.id);
    if(asLcid != "") iLcid = asLcid.ToInt();
    EnglishItem->Checked = iLcid == EnglishItem->Tag;
    RussianItem->Checked = iLcid == RussianItem->Tag;

    if(!Sets.bConnected && Sets.Connect()) capGrabFrame(Sets.hCaptureW);
}
//-----

void DevSettings::SaveToFile(AnsiString FN){
    int FH = FileCreate(FN);
    if(FH != -1){
        FileWrite(FH, &DevIndex, sizeof(DevIndex));
        FileWrite(FH, &CaptureParms, sizeof(CAPTUREPARMS));
        FileWrite(FH, &CapStatus, sizeof(CAPSTATUS));
        FileWrite(FH, &VideoFormatSize, sizeof(VideoFormatSize));
        FileWrite(FH, VideoFormat, VideoFormatSize);
        FileWrite(FH, &ClipRect, sizeof(ClipRect));
        FileWrite(FH, &((int)SensList->Count), sizeof(int));
    }
}

```

```

for(int i=0; i < SensList->Count; i++){
    FileWrite(FH, SensList->Items[i], sizeof(SensRegion));
}
FileWrite(FH, &CycleTime, sizeof(CycleTime));
FileWrite(FH, &MaxActivePoints, sizeof(MaxActivePoints));
FileWrite(FH, &AlarmMotionTime, sizeof(AlarmMotionTime));
FileWrite(FH, &ForceSaveTime, sizeof(ForceSaveTime));
FileWrite(FH, &JPEGCompression, sizeof(JPEGCompression));
char cstr[255];
strcpy(cstr, ImagesFolder.c_str());
FileWrite(FH, cstr, sizeof(cstr));
strcpy(cstr, AlarmExecuteFN.c_str());
FileWrite(FH, cstr, sizeof(cstr));
FileWrite(FH, &LocalId, sizeof(LocalId));
strcpy(cstr, FNTemplate.c_str());
FileWrite(FH, cstr, sizeof(cstr));
FileClose(FH);
}
}

void DevSettings::LoadFromFile(AnsiString FN){
    int FH = FileOpen(FN, fmOpenRead);
    if(FH != -1){
        FileRead(FH, &DevIndex, sizeof(DevIndex));
        FileRead(FH, &CaptureParms, sizeof(CAPTUREPARMS));
        FileRead(FH, &CapStatus, sizeof(CAPSTATUS));
        FileRead(FH, &VideoFormatSize, sizeof(VideoFormatSize));
        if(VideoFormat){
            delete[] VideoFormat;
            VideoFormat = NULL;
        }
        if(VideoFormatSize > 0){
            VideoFormat = (LPBITMAPINFO) new BYTE[VideoFormatSize];
            FileRead(FH, VideoFormat, VideoFormatSize);
        }
        FileRead(FH, &ClipRect, sizeof(ClipRect));
        int senscnt;
        FileRead(FH, &senscnt, sizeof(senscnt));
        for(int i=0; i < senscnt && i < SensList->Count; i++){
            FileRead(FH, SensList->Items[i], sizeof(SensRegion));
        }
        FileRead(FH, &CycleTime, sizeof(CycleTime));
        FileRead(FH, &MaxActivePoints, sizeof(MaxActivePoints));
        FileRead(FH, &AlarmMotionTime, sizeof(AlarmMotionTime));
        FileRead(FH, &ForceSaveTime, sizeof(ForceSaveTime));
        FileRead(FH, &JPEGCompression, sizeof(JPEGCompression));
        char cstr[255];
        FileRead(FH, cstr, sizeof(cstr));
        ImagesFolder = cstr;
        FileRead(FH, cstr, sizeof(cstr));
        AlarmExecuteFN = cstr;
        FileRead(FH, &LocalId, sizeof(LocalId));
        FileRead(FH, cstr, sizeof(cstr));
        FNTemplate = cstr;
        if(FNTemplate == "") FNTemplate = "{dd}_{mm}_{yyyy}\\{dd}{mmm}{yy}-
{hh}{nn}{ss}{zzz}";
        FileClose(FH);
    }
}

// StatusCallbackProc: стан функції зворотнього виклику
// hWnd: дескриптор вікна захвату
// nID: код стану для поточного стану
// lpStatusText: рядок повідомлення для поточного стану
//
LRESULT PASCAL DevSettings::StatusCallbackProc(HWND hWnd, int nID,

```

```

LPSTR lpStatusText)
{
    if (nID == 0) { // Видалити старі повідомлення про стан
        return (LRESULT) TRUE;
    }
    // Показати ID стану та текстове повідомлення про стан
    Sets.AddToLog("Status("+IntToStr(nID)+"): "+lpStatusText);
    return (LRESULT) TRUE;
}

// errorCallbackProc: помилка функції зворотнього виклику
// hWnd: дескриптор вікна захвату
// nErrID: код помилки для неочікуваної помилки
// lpErrorText: повідомлення про помилку для неочікуваної помилки
//
LRESULT PASCAL DevSettings::ErrorCallbackProc(HWND hWnd, int nErrID,
    LPSTR lpErrorText)
{
    if (nErrID == 0) // Початок нової головної функції
        return (LRESULT) TRUE; // Очистити старі помилки

    // Показати ідентифікатор помилки та текст
    Sets.AddToLog("Помилка("+IntToStr(nErrID)+"): "+lpErrorText);
    MessageBox(hWnd, lpErrorText, AnsiString("Помилка
    #"+IntToStr(nErrID)).c_str(), MB_OK | MB_ICONEXCLAMATION);
    return (LRESULT) TRUE;
}

// FrameCallbackProc: функція зворотнього виклику кадру
// hWnd: дескриптор вікна захвату
// lpVHdr: вказівник на структуру, яка містить
// інформацію про кадр
//
LRESULT PASCAL DevSettings::FrameCallbackProc(HWND hWnd, LPVIDEOHDR lpVHdr)
{
    Sets.UpdateFrame(lpVHdr);
    return (LRESULT) TRUE ;
}

bool DevSettings::Connect(){
    int nID = 0;
    Disconnect();
    if(!hCaptureW) hCaptureW = capCreateCaptureWindow("Система
    відеоспостереження",
        WS_CHILD, 0, 0, 100, 100, hParentW, nID);
    bool fOK = false;
    if(hCaptureW && DevIndex != -1){
        //установка функції зворотнього виклику
        fOK = capSetCallbackOnError(hCaptureW, &DevSettings::ErrorCallbackProc);
        if(fOK) fOK = capSetCallbackOnStatus(hCaptureW,
        &DevSettings::StatusCallbackProc);
        if(fOK) fOK = capSetCallbackOnFrame(hCaptureW,
        &DevSettings::FrameCallbackProc);
        //під'єднатися до драйвера
        if(fOK) fOK = capDriverConnect(hCaptureW, DevIndex);
        //установка параметрів захвату
        if(fOK && CaptureParms.dwRequestMicroSecPerFrame > 0){
            fOK = capCaptureSetSetup(hCaptureW, &CaptureParms, sizeof
            (CAPTUREPARMS));
            if(!fOK) fOK = capCaptureGetSetup(hCaptureW, &CaptureParms,
            sizeof(CAPTUREPARMS));
        }else if(fOK) fOK = capCaptureGetSetup(hCaptureW, &CaptureParms,
            sizeof(CAPTUREPARMS));
        //установка формату відео
    }
}

```

```

if(fOK && VideoFormat){
    fOK = capSetVideoFormat(hCaptureW, VideoFormat, VideoFormatSize);
}
if((!fOK && VideoFormat) || (fOK && !VideoFormat)){
    VideoFormatSize = capGetVideoFormatSize(hCaptureW);
    if(VideoFormat){
        delete[] VideoFormat;
        VideoFormat = NULL;
    }
    if(VideoFormatSize > 0){
        VideoFormat = (LPBITMAPINFO) new BYTE[VideoFormatSize];
        fOK = capGetVideoFormat(hCaptureW, VideoFormat, VideoFormatSize);
    }
}
if(fOK){
    //установка розміру вікна
    capGetStatus(hCaptureW, &CapStatus, sizeof (CAPSTATUS));
    // SetWindowPos(hCaptureW, NULL, 0, 0, CapStatus.uiImageWidth,
    //             CapStatus.uiImageHeight, SWP_NOZORDER | SWP_NOMOVE);
    FramePic->Bitmap->Width = CapStatus.uiImageWidth;
    FramePic->Bitmap->Height = CapStatus.uiImageHeight;
    FramePic->Bitmap->HandleType = bmDIB;
    FramePic->Bitmap->PixelFormat = pf24bit;
    FramePic->Bitmap->IgnorePalette = true;

    GrayPic->Bitmap->Width = pW;
    GrayPic->Bitmap->Height = pH;
    GrayPic->Bitmap->HandleType = bmDIB;
    GrayPic->Bitmap->PixelFormat = pf24bit;
    GrayPic->Bitmap->IgnorePalette = true;

    if(GrayData) delete[] GrayData;
    GrayData = new BYTE[pW * pH];
    memset(GrayData, 0, pW*pH);

    //if(FiltData) delete[] FiltData;
    //FiltData = new BYTE[pW * pH];
}
}
if(!fOK){
    AddToLog("Помилка: неможливо під'єднатися!");
    bConnected = false;
}else{
    bConnected = true;
    AddToLog("Стан: Драйвер захвату під'єднано");
}
CurBufFrame = 0;
FramesCaptured = 0;
return fOK;
}

bool DevSettings::Disconnect(){
if(!bConnected && !hCaptureW) return true;
if(hCaptureW){
    // вивільнити функції зворотнього виклику
    capSetCallbackOnError(hCaptureW, NULL);
    capSetCallbackOnStatus(hCaptureW, NULL);
    capSetCallbackOnVideoStream(hCaptureW, NULL);
    capSetCallbackOnFrame(hCaptureW, NULL);
    //від'єднатися від пристрою
    capDriverDisconnect(hCaptureW);
    DestroyWindow(hCaptureW);
    hCaptureW = NULL;
}
bConnected = false;
AddToLog("Стан: Драйвер захвату від'єднано");
}

```

```

    return true;
}

void DevSettings::UpdateFrame(LPVIDEOHDR lpVHdr)
{
    HDC hdc;
    HDRAWDIB hdd;
    hdd = DrawDibOpen();

    //виводити вихідний кадр в FramePic
    hdc = FramePic->Bitmap->Canvas->Handle ;
    FramePic->Bitmap->Canvas->TryLock();
    DrawDibBegin(hdd, hdc,
        VideoFormat->bmiHeader.biWidth, //dxDest
        VideoFormat->bmiHeader.biHeight, //dyDest
        &VideoFormat->bmiHeader, //lpbi
        VideoFormat->bmiHeader.biWidth, //dxSrc
        VideoFormat->bmiHeader.biHeight, //dySrc
        NULL);
    DrawDibDraw(hdd, hdc,
        0, //xDst
        0, //yDst
        VideoFormat->bmiHeader.biWidth, //dxDest
        VideoFormat->bmiHeader.biHeight, //dyDest
        &VideoFormat->bmiHeader, //lpbi
        lpVHdr->lpData, //lpBits
        0, //xSrc
        0, //ySrc
        VideoFormat->bmiHeader.biWidth, //dxSrc
        VideoFormat->bmiHeader.biHeight, //dySrc
        NULL);
    DrawDibEnd(hdd);
    FramePic->Bitmap->Canvas->Unlock();

    //виводити вихідний кадр в GrayPic (для обробки)
    DWORD xSrc = 0, ySrc = 0,
        dxSrc = VideoFormat->bmiHeader.biWidth,
        dySrc = VideoFormat->bmiHeader.biHeight;
    if(ClipRect.bottom > 0){
        //використати відсікання
        float MX = dxSrc / (float)pW,
            MY = dySrc / (float)pH;
        xSrc = ClipRect.left * MX;
        ySrc = ClipRect.top * MY;
        dxSrc = (ClipRect.right - ClipRect.left) * MX;
        if(dxSrc > (DWORD)VideoFormat->bmiHeader.biWidth - xSrc) dxSrc =
VideoFormat->bmiHeader.biWidth - xSrc;
        dySrc = (ClipRect.bottom - ClipRect.top) * MY;
        if(dySrc > (DWORD)VideoFormat->bmiHeader.biHeight - ySrc) dySrc =
VideoFormat->bmiHeader.biHeight - ySrc;
    }
    hdc = GrayPic->Bitmap->Canvas->Handle ;
    GrayPic->Bitmap->Canvas->TryLock();
    DrawDibBegin(hdd, hdc,
        pW, //dxDest
        pH, //dyDest
        &VideoFormat->bmiHeader, //lpbi
        dxSrc,
        dySrc,
        NULL);
    //DrawDibRealize(hdd, hdc, fBackground);
    DrawDibDraw(hdd, hdc,
        0, //xDst
        0, //yDst
        pW, //dxDest
        pH, //dyDest
        &VideoFormat->bmiHeader, //lpbi
        lpVHdr->lpData, //lpBits

```

```

        xSrc,
        ySrc,
        dxSrc,
        dySrc,
        NULL);
DrawDibEnd(hdd);
DrawDibClose(hdd);
GrayPic->Bitmap->Canvas->Unlock();

// створити зображення у відтинках сірого і автоматично збалансувати його
BITMAP bm;
GetObject(GrayPic->Bitmap->Handle, sizeof(BITMAP), &bm);
register DWORD offs;
DWORD maxoffs = bm.bmHeight * bm.bmWidthBytes;
register int Y;
BYTE minY = 255, maxY = 0;
for(offs = 0; offs < maxoffs; offs+= 3){
    Y =
((int)((BYTE*)bm.bmBits)[offs]+(int)((BYTE*)bm.bmBits)[offs+1]+(int)((BYTE*)bm.b
mBits)[offs+2]) / 3;
    if(minY > Y) minY = Y;
    if(maxY < Y) maxY = Y;
    GrayData[offs/3] = Y;
}

maxoffs = maxoffs/3;
if(maxY - minY == 0){
    AddToLog(AnsiString("Помилка захвату зображення ") + __LINE__ + " " + __FILE__);
    return;
}
static float M = 255/(maxY - minY);
static float avminY = minY, avmaxY = maxY;
avmaxY = 0.95*avmaxY + 0.05*maxY;
if(avmaxY > 255) avmaxY = 255;

avminY = 0.95*avminY + 0.05*minY;
if(avminY > 250) avminY = 250;

if(avmaxY <= avminY) avmaxY = avminY + 1;
M = 255/(avmaxY - avminY);
for(offs = 0; offs < maxoffs; offs++){
    Y = ((int)GrayData[offs] - avminY) * M;
    if(Y < 0) GrayData[offs] = 0;
    else if(Y > 255) GrayData[offs] = 255;
    else GrayData[offs] = Y;
}

CurBufFrame++;
CurBufFrame = CurBufFrame % (FrameBufCnt-1);

FiltData = (BYTE*)FrameBuffer->Items[CurBufFrame];
FilterFrame(GrayData, FiltData);
FramesCaptured ++;

if(FramesCaptured > 2){
    CompareFrames(FiltData,
        (BYTE*)FrameBuffer->Items[CalcBufIndex(CurBufFrame-1)],
        DifData );
}
}

void __fastcall TMainForm::FormClose(TObject *Sender, TCloseAction &Action)
{
    bWork = false;
    Sets.SaveToFile(ConfigName);
}

```

```

    Sets.Disconnect();
}
//-----

void __fastcall TMainForm::FilterItemClick(TObject *Sender)
{
    if(!Sets.bConnected) Sets.Connect();
    if(Sets.bConnected) capGrabFrame(Sets.hCaptureW);
    SetFilterForm->ShowModal();
}
//-----

void DevSettings::FilterFrame(BYTE* Src, BYTE* Dst)
{
    register int idx, idy, x, y, i, offs, toffs, cnt;
    int ytop, ybot, dx, dy, srw, srh;
    float SY;
    DWORD sum;
    SensRegion *sr = (SensRegion*)SensList->Items[0];
    srw = sr->right - sr->left;
    srh = sr->bottom - sr->top;

    for(i=0; i<SensList->Count; i++){
        sr = (SensRegion*)SensList->Items[i];
        ybot = pH-sr->top; ytop = pH-sr->bottom;
        dx = (100 - sr->sens) * (srw)/400;
        dy = (100 - sr->sens) * (srh)/300;
        SY = sr->sens/100.0*0.4 + 0.6;
        //dx=0; dy=5;
        for(y = ytop; y < ybot; y++){
            for(x = sr->left; x < sr->right; x++){
                offs = pW*y + x;
                if(sr->sens == 0){
                    Dst[offs] = 0;
                }else if(sr->sens == 100){
                    Dst[offs] = Src[offs];
                }
                else{
                    sum = 0;
                    cnt = 0;
                    for(idy = y-dy; idy <= y+dy; idy++){
                        for(idx = x-dx; idx <= x+dx; idx++){
                            toffs = pW*((idy < ytop)? idy+srh : (idy >= ybot)? idy-srh
: idy) +
                                ((idx < sr->left)? idx+srw : ((idx >= sr->right)?
idx-srw : idx));
                            // toffs = pW*((idy < 0)? idy+srh : (idy >= pH)? idy-srh :
idy) +
                            //
                                ((idx < 0)? idx+srw : ((idx >= pW)? idx-srw :
idx));
                            sum+= Src[toffs];
                            cnt++;
                        }
                    }
                    Dst[offs] = sum / cnt * SY;
                }
            }
        }
    }
}

void DevSettings::PaintArray(BYTE * arr, Graphics::TBitmap * bmp)
{
    BITMAP bm;
    bmp->Width = pW;
    bmp->Height = pH;
    bmp->PixelFormat = pf24bit;
}

```

```

GetObject (bmp->Handle, sizeof (BITMAP), &bm);
DWORD maxoffs = bm.bmHeight * bm.bmWidthBytes,
        offs;
BYTE Y;
if (!arr) return;
for (offs = 0; offs < maxoffs; offs += 3) {
    Y = arr[offs/3];
    ((BYTE*)bm.bmBits)[offs] = Y;
    ((BYTE*)bm.bmBits)[offs+1] = Y;
    ((BYTE*)bm.bmBits)[offs+2] = Y;
}
}

void DevSettings::CompareFrames (BYTE * f1, BYTE * f2, BYTE * dest)
{
    register int offs, maxoffs = pW * pH, Y;
    static int StartMotionTime;
    ActivePoints = 0;
    for (offs = 0; offs < maxoffs; offs++) {
        Y = abs (f2[offs] - f1[offs]) * 5;
        if (Y >= 255) {
            Y = 255;
            ActivePoints ++;
        }
        dest[offs] = Y;
    }

    if (ActivePoints >= MaxActivePoints) {
        if (MotionTime == 0) StartMotionTime = GetTickCount ();
        MotionTime = GetTickCount () - StartMotionTime + 1;
        SaveFrame ();
        if (MotionTime > AlarmMotionTime * 1000) {
            DoAlarm ();
            MotionTime = 0;
        }
    } else {
        MotionTime = 0;
    }
}

int inline DevSettings::CalcBufIndex (int arg)
{
    return ((arg >= FrameBufCnt-1)? arg%(FrameBufCnt-1) : ((arg < 0)? arg +
FrameBufCnt - 1 : arg));
}

void __fastcall TMainForm::PrevPBPaint (TObject *Sender)
{
    Graphics::TBitmap *tbitmap = new Graphics::TBitmap;

    switch (MainPageControl->TabIndex) {
        case 0: //вхідний кадр
            PrevPB->Canvas->StretchDraw (PrevPB->ClientRect, Sets.FramePic->Bitmap);
            break;
        case 1: //контрольний кадр
            Sets.PaintArray (Sets.GrayData, tbitmap);
            FramePB->Canvas->StretchDraw (FramePB->ClientRect, tbitmap);
            break;
        case 2: //відфільтрований кадр
            Sets.PaintArray (Sets.FiltData, tbitmap);
            FiltPB->Canvas->StretchDraw (FiltPB->ClientRect, tbitmap);
            break;
        case 3: //кадр із рухом
    }
}

```

```

        Sets.PaintArray(Sets.DifData, tbmp);
        MotionPB->Canvas->StretchDraw(MotionPB->ClientRect, tbmp);
        break;
    }

    delete tbmp;
}
//-----

void __fastcall TMainForm::SaveItemClick(TObject *Sender)
{
    Sets.SaveToFile(ConfigName);
}
//-----

void __fastcall TMainForm::LogMemoChange(TObject *Sender)
{
    if(LogMemo->Lines->Count > 0)
        StatusBar->Panels->Items[3]->Text = LogMemo->Lines->Strings[LogMemo-
>Lines->Count-1];
}
//-----

void __fastcall TMainForm::Show1Click(TObject *Sender)
{
    Application->Restore();
}
//-----

void __fastcall TMainForm::Hide1Click(TObject *Sender)
{
    Application->Minimize();
}
//-----

void DevSettings::DoAlarm()
{
    AddToLog("ТРИБОГА!");

    if(AlarmExecuteFN != "" && FileExists(AlarmExecuteFN)){
        if(ExtractFileExt(AlarmExecuteFN).LowerCase() == ".wav"){
            PlaySound(AlarmExecuteFN.c_str(), NULL, SND_ASYNC|SND_FILENAME);
        }else{
            ShellExecute(
                hParentW, // дескриптор батьківського вікна
                "open", // вказівник на рядок, який визначає
операцию для виконання
                AlarmExecuteFN.c_str(), // вказівник на рядок із іменем
файлу або папки
                NULL, // вказівник на рядок, який визначає параметри
виконуваних файлів
                NULL, // вказівник на рядок, який визначає папку по
замовчуванню
                SW_SHOWDEFAULT // чи показується файл, коли його
відкрили
            );
        }
    }
}

void DevSettings::SaveFrame()
{
    AnsiString FN = EvalImgPath();
}

```

```

    if(!DirectoryExists(ExtractFilePath(FN)))
ForceDirectories(ExtractFilePath(FN));

    FramePic->Bitmap->Canvas->TextOut(0, 0, DateTimeToStr(Now()));

    TJPEGImage *jpp = new TJPEGImage();
    try
    {
        jpp->Assign(FramePic->Bitmap);
        jpp->CompressionQuality = JPEGCompression;
        jpp->Compress();
        jpp->SaveToFile(FN);
    }
    catch(...){
        delete jpp;
        return;
    }
    delete jpp;
    AddToLog("Кадр збережено");
}

AnsiString DevSettings::EvalImgPath(){
    static unsigned int sn = 0;
    sn++;
    TDateTime dt = Now();
    AnsiString tS, res = FNTemplate;
    AnsiString dd, ddd, mm, mmm, yy, yyyy, hh, nn, ss, zzz;
    DateTimeToString(dd, "dd", dt);
    DateTimeToString(ddd, "ddd", dt);
    DateTimeToString(mm, "mm", dt);
    DateTimeToString(mmm, "mmm", dt);
    DateTimeToString(yy, "yy", dt);
    DateTimeToString(yyyy, "yyyy", dt);
    DateTimeToString(hh, "hh", dt);
    DateTimeToString(nn, "nn", dt);
    DateTimeToString(ss, "ss", dt);
    DateTimeToString(zzz, "zzz", dt);
    DateTimeToString(tS, "dd_mmm_yyyy", dt); // {dd}_{mmm}_{yyyy}\\{dd}{mmm}{yy}-
{hh}{nn}{ss}{zzz}.jpg
    AnsiString FN = ImagesFolder + "\\\" + tS + "\\\";
    //if(!DirectoryExists(FN)) CreateDir(FN);
    DateTimeToString(tS, "ddmmmyy-hhnnsszzz", Now());
    FN = FN + tS + ".jpg";
    TReplaceFlags rf; rf << rfReplaceAll;
    res = StringReplace(res, "{dd}", dd, rf);
    res = StringReplace(res, "{ddd}", ddd, rf);
    res = StringReplace(res, "{mm}", mm, rf);
    res = StringReplace(res, "{mmm}", mmm, rf);
    res = StringReplace(res, "{yy}", yy, rf);
    res = StringReplace(res, "{yyyy}", yyyy, rf);
    res = StringReplace(res, "{hh}", hh, rf);
    res = StringReplace(res, "{nn}", nn, rf);
    res = StringReplace(res, "{ss}", ss, rf);
    res = StringReplace(res, "{zzz}", zzz, rf);
    res = StringReplace(res, "{sn}", IntToStr(sn), rf);
    return ImagesFolder + "\\\" + res + ".jpg";
}

void __fastcall TMainForm::CommonItemClick(TObject *Sender)
{
    SetCommonForm->ShowModal();
}
//-----

void __fastcall TMainForm::OpenImgFolderBitBtnClick(TObject *Sender)
{
    ShellExecute(
        Handle, // дескриптор батьківського вікна

```

```

        "open",          // вказівник на рядок, який визначає операцію для
виконання
        "explorer.exe", // вказівник на рядок із іменем файлу або
папки
        Sets.ImagesFolder.c_str(), // вказівник на рядок, який
визначає параметри виконуваних файлів
        NULL, // вказівник на рядок, який визначає папку по
замовчуванню
        SW_SHOWDEFAULT // чи показується файл, коли його
відкрили
    );
}
//-----

void __fastcall TMainForm::ShootNowBitBtnClick(TObject *Sender)
{
    if(!Sets.bConnected) Sets.Connect();
    if(Sets.bConnected){
        capGrabFrame(Sets.hCaptureW);
        Sets.SaveFrame();
        PrevPBPaint(Sender);
    }
}
//-----

void __fastcall TMainForm::FormCreate(TObject *Sender)
{
    EnglishItem->Tag = ENGLISH;
    RussianItem->Tag = RUSSIAN;

    AnsiString asLcid;
    int iLcid = 0;
    asLcid = LoadStr(_SLcid.id);
    if(asLcid != "") iLcid = asLcid.ToInt();

    EnglishItem->Checked = iLcid == EnglishItem->Tag;
    RussianItem->Checked = iLcid == RussianItem->Tag;
}
//-----

void __fastcall TMainForm::EnglishItemClick(TObject *Sender)
{
    if(LoadNewResourceModule(dynamic_cast<TComponent*>(Sender)->Tag) != 0)
    {
        bWork = false;
        Sets.Disconnect();
        Application->ProcessMessages();
        Sets.LocalId = dynamic_cast<TComponent*>(Sender)->Tag;
        Sets.SaveToFile(ConfigName);
        ReinitializeForms();
        EnglishItem->Checked = EnglishItem == Sender;
        RussianItem->Checked = RussianItem == Sender;
    }
}
//-----

```

// MainUnit.h - заголовочний файл основної програми

```
//-----
#ifndef MainUnitH
#define MainUnitH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Menus.hpp>
#include "vfw.h"
#include <ExtCtrls.hpp>
#include <ComCtrls.hpp>
#include <Graphics.hpp>
#include <Buttons.hpp>
#include <jpeg.hpp>
#include <ActnList.hpp>
#include <ExtActns.hpp>
//-----
struct SensRegion:public RECT{
    int sens;
};

class DevSettings{
public:
    HWND hParentW,
        hCaptureW;
    TMemo *LogMemo;
    TPicture *FramePic;
    TPicture *GrayPic;
    TList *SensList;
    TList *FrameBuffer;
    BYTE *GrayData,
        *FiltData, *DifData;
    int pW, pH;
    int SensRectNx, SensRectNy;
    int FrameBufCnt;
    DWORD FramesCaptured;
    int ActivePoints;
    int CurBufFrame;
    int DevIndex;
    bool bConnected;
    int CycleTime;
    int MaxActivePoints;
    int AlarmMotionTime;
    int ForceSaveTime;
    int MotionTime;
    AnsiString AlarmExecuteFN;
    AnsiString ImagesFolder;
    AnsiString FNTemplate;
    int JPEGCompression;
    int LocalId;

    CAPDRIVERCAPS CapDrvCaps;
    CAPSTATUS CapStatus;
    CAPTUREPARMS CaptureParms;
    LPBITMAPINFO VideoFormat;
    DWORD VideoFormatSize;
    RECT ClipRect;

    DevSettings(){
        JPEGCompression = 70;
        ImagesFolder = ".";
        FNTemplate = "{dd}_ {mm}_ {yyyy} \\ {dd} {mmm} {yy} - {hh} {nn} {ss} {zzz}";
        MotionTime = 0; // мс
        CycleTime = 300; // мс
    }
};
```

```

MaxActivePoints = 10;
AlarmMotionTime = 1; // в секундах
ForceSaveTime = 600; // секунди
DevIndex = -1;
VideoFormatSize = 0;
pW = 320; pH = 240;
SensRectNx = SensRectNy = 8;
ClipRect.bottom = 0;
FrameBufCnt = 10;
ActivePoints = 0;
bConnected = false;
hParentW = hCaptureW = VideoFormat = NULL;
LogMemo = NULL;
GrayData = FiltData = NULL;
FramePic = new TPicture;
GrayPic = new TPicture;
SensList = new TList;
FrameBuffer = new TList;
int nx, ny, dx, dy;
SensRegion *sR;
dx = pW / SensRectNx;
dy = pH / SensRectNy;
for(nx = 0; nx < SensRectNx; nx++){
    for(ny = 0; ny < SensRectNy; ny++){
        sR = new SensRegion;
        sR->left = nx * dx;
        sR->right = sR->left + dx;
        sR->top = ny*dy;
        sR->bottom = sR->top + dy;
        sR->sens = 100;
        SensList->Add(sR);
    }
}
BYTE *tbuf;
for(nx = 0; nx < FrameBufCnt; nx++){
    tbuf = new BYTE[pW*pH];
    memset(tbuf, 0, pW*pH);
    FrameBuffer->Add(tbuf);
}
DifData = (BYTE*)FrameBuffer->Items[FrameBufCnt - 1];
}
~DevSettings() {
    if(VideoFormat) delete[] VideoFormat;
    if(GrayData) delete[] GrayData;
    //if(FiltData) delete[] FiltData;
    delete FramePic;
    delete GrayPic;
    for(int i = 0; i < SensList->Count; i++){
        delete (SensRegion*)(SensList->Items[i]);
    }
    delete SensList;

    for(int i = 0; i < FrameBuffer->Count; i++){
        delete (BYTE*)(FrameBuffer->Items[i]);
    }
    delete FrameBuffer;
}
void SaveToFile(AnsiString FN);
void LoadFromFile(AnsiString FN);
bool Connect();
bool Disconnect();
//callback functions
static LRESULT PASCAL StatusCallbackProc(HWND hWnd, int nID, LPSTR
lpStatusText);
static LRESULT PASCAL ErrorCallbackProc(HWND hWnd, int nErrID, LPSTR
lpErrorText);
static LRESULT PASCAL FrameCallbackProc(HWND hWnd, LPVIDEOHDR lpVHdr);
void AddToLog(AnsiString S){
    if(LogMemo) LogMemo->Lines->Add([" "+TimeToStr(Time())+"] "+S);
}

```

```

}
void UpdateFrame(LPVIDEOHDR lpVHdr);
void FilterFrame(BYTE* Src, BYTE* Dst);
void PaintArray(BYTE * arr, Graphics::TBitmap * bmp);
void CompareFrames(BYTE * f1, BYTE * f2, BYTE * dest);
void DoAlarm();
void SaveFrame();
AnsiString EvalImgPath();

private:
    int inline CalcBufIndex(int arg);

} extern Sets;

class TMainForm : public TForm
{
__published:        // IDE-managed Components
    TMainMenu *MainMenu1;
    TMenuItem *SetItem;
    TMenuItem *DeviceItem;
    TPageControl *MainPageControl;
    TTabSheet *TabSheet1;
    TTabSheet *TabSheet2;
    TPanel *CaptPanel;
    TMemo *LogMemo;
    TTabSheet *TabSheet3;
    TPaintBox *FramePB;
    TMenuItem *FilterItem;
    TPopupMenu *TrayPopupMenu;
    TMenuItem *Show1;
    TTabSheet *TabSheet4;
    TPaintBox *FiltPB;
    TTabSheet *TabSheet5;
    TPaintBox *MotionPB;
    TPaintBox *PrevPB;
    TMenuItem *N1;
    TMenuItem *SaveItem;
    TMenuItem *CommonItem;
    TGroupBox *GroupBox1;
    TBitBtn *SetDeviceBitBtn;
    TBitBtn *SetFilterBitBtn;
    TBitBtn *SetCommonBitBtn;
    TStatusBar *StatusBar;
    TMenuItem *Hide1;
    TBitBtn *OpenImgFolderBitBtn;
    TActionList *ActionList1;
    TBrowseURL *BrowseURL1;
    TMenuItem *WebSite1;
    TGroupBox *GroupBox2;
    TBitBtn *ShootNowBitBtn;
    TButton *ActivateButton;
    TMenuItem *Language1;
    TMenuItem *EnglishItem;
    TMenuItem *RussianItem;
    void __fastcall ActivateButtonClick(TObject *Sender);
    void __fastcall DeviceItemClick(TObject *Sender);
    void __fastcall FormShow(TObject *Sender);
    void __fastcall FormClose(TObject *Sender, TCloseAction &Action);
    void __fastcall FilterItemClick(TObject *Sender);
    void __fastcall PrevPBPaint(TObject *Sender);
    void __fastcall SaveItemClick(TObject *Sender);
    void __fastcall LogMemoChange(TObject *Sender);
    void __fastcall Show1Click(TObject *Sender);
    void __fastcall Hide1Click(TObject *Sender);
    void __fastcall CommonItemClick(TObject *Sender);
    void __fastcall OpenImgFolderBitBtnClick(TObject *Sender);
    void __fastcall ShootNowBitBtnClick(TObject *Sender);

```

```
void __fastcall FormCreate(TObject *Sender);
void __fastcall EnglishItemClick(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TMainForm(TComponent* Owner);
};
//-----
extern PACKAGE TMainForm *MainForm;
//-----
#endif
```

Кафедра _КБПЗ_ 2023рік

// SetFilterUnit.cpp - модуль фільтрації відеозображення

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "SetFilterUnit.h"
#include "MainUnit.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TSetFilterForm *SetFilterForm;
int SRX1, SRX2, SRY1, SRY2;
int CurSens = -1;

//-----
__fastcall TSetFilterForm::TSetFilterForm(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TSetFilterForm::OkBitBtnClick(TObject *Sender)
{
    DeleteFile("prev.cfg");
    Close();
}
//-----
void __fastcall TSetFilterForm::CancelBitBtnClick(TObject *Sender)
{
    Sets.LoadFromFile("prev.cfg");
    DeleteFile("prev.cfg");
    Close();
}
//-----
void __fastcall TSetFilterForm::SRPBPaint(TObject *Sender)
{
    DrawSR();
}
//-----
void __fastcall TSetFilterForm::SRPBMouseMove(TObject *Sender,
    TShiftState Shift, int X, int Y)
{
    if(DrawRectButton->Down){
        SRPB->Cursor = crCross;
        if(Shift.Contains(ssLeft)){//натиснута ліва клавіша миші
            SRX2 = X;
            SRY2 = Y;
            DrawSR();
        }
    }
    else{
        SRPB->Cursor = crDefault;
    }
}
//-----

void __fastcall TSetFilterForm::SRPBMouseDown(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    if(Button == mbLeft && DrawRectButton->Down){
        SRX1 = SRX2 = X;
        SRY1 = SRY2 = Y;
    }
}
//-----
```

```

void TSetFilterForm::DrawSR()
{
    TPicture *tPic = new TPicture;
    tPic->Bitmap->Width = SRPB->Width;
    tPic->Bitmap->Height = SRPB->Height;
    tPic->Bitmap->Canvas->Brush->Color = (TColor)RGB(100,100,100);
    tPic->Bitmap->Canvas->FillRect(tPic->Bitmap->Canvas->ClipRect);
    tPic->Bitmap->Canvas->Brush->Color = clWhite;
    tPic->Bitmap->Canvas->FillRect(Rect(SRX1, SRY1, SRX2, SRY2));

    tPic->Bitmap->Canvas->CopyMode = cmSrcAnd;
    tPic->Bitmap->Canvas->StretchDraw(tPic->Bitmap->Canvas->ClipRect,
Sets.FramePic->Bitmap);
    SRPB->Canvas->Draw(0, 0, tPic->Bitmap);

    delete tPic;
}
void __fastcall TSetFilterForm::FormShow(TObject *Sender)
{
    //зберегти поточні налаштування
    Sets.SaveToFile("prev.cfg");

    SRPB->Width = Sets.pW;
    SRPB->Height = Sets.pH;
    SensPB->Width = Sets.pW;
    SensPB->Height = Sets.pH;
    if(Sets.ClipRect.bottom > 0){
        SRX1 = Sets.ClipRect.left;
        SRX2 = Sets.ClipRect.right;
        SRY1 = Sets.ClipRect.top;
        SRY2 = Sets.ClipRect.bottom;
    }else{
        SRX1 = SRY1 = 0;
        SRX2 = Sets.pW;
        SRY2 = Sets.pH;
    }

    ActivePointsEdit->Text = IntToStr(Sets.MaxActivePoints);
    AlarmTimeEdit->Text = IntToStr(Sets.AlarmMotionTime);
    CycleTimeEdit->Text = IntToStr(Sets.CycleTime);
    ForceSaveEdit->Text = IntToStr(Sets.ForceSaveTime);

}
//-----

void __fastcall TSetFilterForm::RectSaveButtonClick(TObject *Sender)
{
    int t;
    if(SRX1 > SRX2){
        t = SRX1; SRX1 = SRX2; SRX2 = t;
    }

    if(SRY1 > SRY2){
        t = SRY1; SRY1 = SRY2; SRY2 = t;
    }

    Sets.ClipRect.left = SRX1;
    Sets.ClipRect.right = SRX2;
    Sets.ClipRect.top = SRY1;
    Sets.ClipRect.bottom = SRY2;
    if(Sets.bConnected) capGrabFrame(Sets.hCaptureW);
}
//-----

void __fastcall TSetFilterForm::SensPBPaint(TObject *Sender)
{

```

```

    DrawSensRegions();
}
//-----

void TSetFilterForm::DrawSensRegions()
{
    int i;
    SensRegion *sR;
    TPicture *tPic = new TPicture;
    tPic->Bitmap->Width = SensPB->Width;
    tPic->Bitmap->Height = SensPB->Height;

    for(i = 0; i < Sets.SensList->Count; i++){
        sR = (SensRegion*)Sets.SensList->Items[i];
        int cl = sR->sens * 255.0 / 100.0;
        tPic->Bitmap->Canvas->Brush->Color = (TColor)RGB(cl, cl, cl);
        tPic->Bitmap->Canvas->FillRect(*sR);
    }

    tPic->Bitmap->Canvas->CopyMode = cmSrcAnd;
    tPic->Bitmap->Canvas->Draw(0, 0, Sets.GrayPic->Bitmap);

    tPic->Bitmap->Canvas->Font->Color = clRed;
    tPic->Bitmap->Canvas->Pen->Width = 1;

    for(i = 0; i < Sets.SensList->Count; i++){
        sR = (SensRegion*)Sets.SensList->Items[i];
        tPic->Bitmap->Canvas->Brush->Style = bsSolid;
        tPic->Bitmap->Canvas->Brush->Color = clRed;
        tPic->Bitmap->Canvas->FrameRect(*sR);
        tPic->Bitmap->Canvas->Brush->Style = bsClear;
        tPic->Bitmap->Canvas->TextOut((sR->left+sR->right)*0.5-10, (sR->top+sR->bottom)*0.5-10, IntToStr(sR->sens));
    }

    SensPB->Canvas->Draw(0, 0, tPic->Bitmap);

    delete tPic;
    // SensPB->Canvas->Draw(0, 0, Sets.GrayPic->Bitmap);
}

void __fastcall TSetFilterForm::SensSpeedButtonClick(TObject *Sender)
{
    if(!((TSpeedButton*)Sender)->Down){
        CurSens = -1;
    }else{
        CurSens = StrToInt(((TSpeedButton*)Sender)->Caption);
    }
}
//-----

void __fastcall TSetFilterForm::SensPBMouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    int i;
    SensRegion *sR;
    if(CurSens != -1 && Button == mbLeft){
        for(i = 0; i < Sets.SensList->Count; i++){
            sR = (SensRegion*)Sets.SensList->Items[i];
            if(sR->left < X && sR->right > X && sR->top < Y && sR->bottom > Y){
                sR->sens = CurSens;
            }
        }
        DrawSensRegions();
    }
}

```

```

}
//-----

void __fastcall TSetFilterForm::SensPBMouseMove(TObject *Sender,
    TShiftState Shift, int X, int Y)
{
    if(CurSens != -1){
        SensPB->Cursor = crHandPoint;
    }else{
        SensPB->Cursor = crDefault;
    }
}
//-----

void __fastcall TSetFilterForm::SensClearButtonClick(TObject *Sender)
{
    int i;
    SensRegion *sR;
    for(i = 0; i < Sets.SensList->Count; i++){
        sR = (SensRegion*)Sets.SensList->Items[i];
        sR->sens = 90;
    }
    DrawSensRegions();
}
//-----

void __fastcall TSetFilterForm::AlarmApplyBitBtnClick(TObject *Sender)
{
    try{
        Sets.MaxActivePoints = StrToInt(ActivePointsEdit->Text.Trim());
        Sets.AlarmMotionTime = StrToInt(AlarmTimeEdit->Text.Trim());
        Sets.CycleTime = StrToInt(CycleTimeEdit->Text.Trim());
        Sets.ForceSaveTime = StrToInt(ForceSaveEdit->Text.Trim());
    }catch(...){
        ShowMessage("Введено число не входить в допустимий діапазон значень!");
    }
}
//-----

```

Кафедра — КОДЗ — 2023 рік

// SetFilterUnit.h - заголовочний файл модуля фільтрації відеозображення

```
//-----
#ifndef SetFilterUnitH
#define SetFilterUnitH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Buttons.hpp>
#include <ComCtrls.hpp>
#include <ExtCtrls.hpp>
//-----
class TSetFilterForm : public TForm
{
    __published:        // IDE-компоненти
        TBitBtn *OkBitBtn;
        TBitBtn *CancelBitBtn;
        TPageControl *PageControl1;
        TTabSheet *TabSheet1;
        TPaintBox *SRPB;
        TTabSheet *TabSheet2;
        TBitBtn *RectSaveButton;
        TSpeedButton *DrawRectButton;
        TPaintBox *SensPB;
        TBitBtn *SensClearButton;
        TSpeedButton *SensSpeedButton;
        TSpeedButton *SpeedButton2;
        TSpeedButton *SpeedButton3;
        TSpeedButton *SpeedButton4;
        TSpeedButton *SpeedButton5;
        TSpeedButton *SpeedButton6;
        TSpeedButton *SpeedButton7;
        TSpeedButton *SpeedButton8;
        TSpeedButton *SpeedButton9;
        TSpeedButton *SpeedButton10;
        TSpeedButton *SpeedButton11;
        TTabSheet *TabSheet3;
        TLabel *Label1;
        TLabel *Label2;
        TLabel *Label3;
        TLabel *Label4;
        TLabel *Label5;
        TLabel *Label6;
        TLabel *Label7;
        TEdit *CycleTimeEdit;
        TEdit *ActivePointsEdit;
        TEdit *AlarmTimeEdit;
        TEdit *ForceSaveEdit;
        TLabel *Label8;
        TBitBtn *AlarmApplyBitBtn;
        void __fastcall OkBitBtnClick(TObject *Sender);
        void __fastcall CancelBitBtnClick(TObject *Sender);
        void __fastcall SRPBPaint(TObject *Sender);
        void __fastcall SRPBMouseMove(TObject *Sender, TShiftState Shift,
            int X, int Y);
        void __fastcall SRPBMouseDown(TObject *Sender, TMouseButton Button,
            TShiftState Shift, int X, int Y);
        void __fastcall FormShow(TObject *Sender);
        void __fastcall RectSaveButtonClick(TObject *Sender);
        void __fastcall SensPBPaint(TObject *Sender);
        void __fastcall SensSpeedButtonClick(TObject *Sender);
        void __fastcall SensPBMouseUp(TObject *Sender, TMouseButton Button,
            TShiftState Shift, int X, int Y);
        void __fastcall SensPBMouseMove(TObject *Sender, TShiftState Shift,
            int X, int Y);
        void __fastcall SensClearButtonClick(TObject *Sender);

```

```
void __fastcall AlarmApplyBitBtnClick(TObject *Sender);
private: // користувацькі об'яви
public: // користувацькі об'яви
    __fastcall TSetFilterForm(TComponent* Owner);
    void DrawSR();
    void DrawSensRegions();
};
//-----
extern PACKAGE TSetFilterForm *SetFilterForm;
//-----
#endif
```

Кафедра _КБПЗ_ 2023рік

// SetDeviceUnit.cpp - модуль вибору пристрою для ведення відеоспостереження

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "SetDeviceUnit.h"
#include "MainUnit.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TSetDeviceForm *SetDeviceForm;
//-----
__fastcall TSetDeviceForm::TSetDeviceForm(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TSetDeviceForm::FormShow(TObject *Sender)
{
    //зберегти поточні налаштування
    Sets.SaveToFile("prev.cfg");

    UpdateState();

    // перерахування пристроїв
    char szDeviceName[80];
    char szDeviceVersion[80];

    DeviceSelect->Items->Clear();
    for (int wIndex = 0; wIndex < 10; wIndex++)
    {
        if (capGetDriverDescription(wIndex, szDeviceName,
            sizeof(szDeviceName), szDeviceVersion,
            sizeof(szDeviceVersion)))
        {
            // Додати ім'я до списку встановлених драйверів захвату
            // та потім дозволити користувачеві вибрати драйвер для використання
            DeviceSelect->Items-
>Add(AnsiString(szDeviceName)+" ("+AnsiString(szDeviceVersion)+")");
            if(Sets.DevIndex == wIndex) DeviceSelect->Text =
AnsiString(szDeviceName)+" ("+AnsiString(szDeviceVersion)+")";
        }
    }
}
//-----
void __fastcall TSetDeviceForm::DeviceSelectSelect(TObject *Sender)
{
    //update device index
    Sets.DevIndex = DeviceSelect->ItemIndex;
    UpdateState();
}
//-----
void __fastcall TSetDeviceForm::OkBitBtnClick(TObject *Sender)
{
    DeleteFile("prev.cfg");
    Close();
}
//-----
void __fastcall TSetDeviceForm::CancelBitBtnClick(TObject *Sender)
{
    Sets.LoadFromFile("prev.cfg");
    DeleteFile("prev.cfg");
    Close();
}
//-----
```

```

void __fastcall TSetDeviceForm::VideoSrcBitBtnClick(TObject *Sender)
{
    capDlgVideoSource (Sets.hCaptureW);
}
//-----
void __fastcall TSetDeviceForm::FormatBitBtnClick(TObject *Sender)
{
    capDlgVideoFormat (Sets.hCaptureW);
    capGetStatus (Sets.hCaptureW, &Sets.CapStatus, sizeof (CAPSTATUS));

    Sets.VideoFormatSize = capGetVideoFormatSize (Sets.hCaptureW);
    if (Sets.VideoFormat) {
        delete[] Sets.VideoFormat;
        Sets.VideoFormat = NULL;
    }
    if (Sets.VideoFormatSize > 0) {
        Sets.VideoFormat = (LPBITMAPINFO) new BYTE [Sets.VideoFormatSize];
        capGetVideoFormat (Sets.hCaptureW, Sets.VideoFormat, Sets.VideoFormatSize);
    }
}
//-----
void __fastcall TSetDeviceForm::DisplayBitBtnClick(TObject *Sender)
{
    capDlgVideoDisplay (Sets.hCaptureW);
}
//-----

void TSetDeviceForm::UpdateState ()
{
    //під'єднати новий пристрій
    capDriverDisconnect (Sets.hCaptureW);
    bool fOK = capDriverConnect (Sets.hCaptureW, Sets.DevIndex);

    if (fOK) {
        //отримати заголовки драйверу
        capDriverGetCaps (Sets.hCaptureW, &Sets.CapDrvCaps, sizeof
(CAPDRIVERCAPS));
        //отримати заголовки вікна
        capGetStatus (Sets.hCaptureW, &Sets.CapStatus, sizeof (CAPSTATUS));
        //отримати доступні діалоги для драйверу
        // діалогове вікно для джерела відеосигналу
        VideoSrcBitBtn->Visible = false;
        if (Sets.CapDrvCaps.fHasDlgVideoSource)
            VideoSrcBitBtn->Visible = true;
        // діалогове вікно для формату відео
        FormatBitBtn->Visible = false;
        if (Sets.CapDrvCaps.fHasDlgVideoFormat)
            FormatBitBtn->Visible = true;
        // діалогове вікно для відеозображення
        DisplayBitBtn->Visible = false;
        if (Sets.CapDrvCaps.fHasDlgVideoDisplay)
            DisplayBitBtn->Visible = true;
    }
}

```

// SetDeviceUnit.h - заголовочний файл модуля вибору пристрою для ведення відеоспостереження

```
//-----
#ifndef SetDeviceUnitH
#define SetDeviceUnitH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Buttons.hpp>
//-----
class TSetDeviceForm : public TForm
{
__published:      // IDE-managed Components
    TComboBox *DeviceSelect;
    TBitBtn *OkBitBtn;
    TBitBtn *CancelBitBtn;
    TBitBtn *VideoSrcBitBtn;
    TBitBtn *FormatBitBtn;
    TBitBtn *DisplayBitBtn;
    TLabel *Label1;
    void __fastcall FormShow(TObject *Sender);
    void __fastcall DeviceSelectSelect(TObject *Sender);
    void __fastcall OkBitBtnClick(TObject *Sender);
    void __fastcall CancelBitBtnClick(TObject *Sender);
    void __fastcall VideoSrcBitBtnClick(TObject *Sender);
    void __fastcall FormatBitBtnClick(TObject *Sender);
    void __fastcall DisplayBitBtnClick(TObject *Sender);
private:          // користувацькі об'яви
public:           // користувацькі об'яви
    __fastcall TSetDeviceForm(TComponent* Owner);
    void UpdateState();
};
//-----
extern PACKAGE TSetDeviceForm *SetDeviceForm;
//-----
#endif
```

// SetCommonUnit.cpp - модуль вибору параметрів відеоспостереження

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "SetCommonUnit.h"
#include "MainUnit.h"

//-----
#pragma package(smart_init)
#pragma link "cdirout1"
#pragma link "CSPIN"
#pragma resource "*.dfm"
TSetCommonForm *SetCommonForm;
//-----
__fastcall TSetCommonForm::TSetCommonForm(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TSetCommonForm::OkBitBtnClick(TObject *Sender)
{
    Sets.ImagesFolder = ImageDirEdit->Text;
    Sets.JPEGCompression = JPEGCompressionEdit->Value;
    Sets.AlarmExecuteFN = AlarmOpenEdit->Text;
    DeleteFile("prev.cfg");
    Close();
}
//-----
void __fastcall TSetCommonForm::CancelBitBtnClick(TObject *Sender)
{
    Sets.LoadFromFile("prev.cfg");
    DeleteFile("prev.cfg");
    Close();
}
//-----
void __fastcall TSetCommonForm::FormShow(TObject *Sender)
{
    //зберегти поточні налаштування
    Sets.SaveToFile("prev.cfg");

    DLB->Directory = Sets.ImagesFolder;
    ImageDirEdit->Text = Sets.ImagesFolder;
    FNTemplateEdit->Text = Sets.FNTemplate;
    FNExampleLabel->Caption = Sets.EvalImgPath();
    JPEGCompressionEdit->Value = Sets.JPEGCompression;
    AlarmOpenEdit->Text = Sets.AlarmExecuteFN;
    AlarmOpenDialog->FileName = Sets.AlarmExecuteFN;
}
//-----
void __fastcall TSetCommonForm::ImagePathSpeedButtonClick(TObject *Sender)
{
    DLPanel->Show();
}
//-----
void __fastcall TSetCommonForm::DLOkButtonClick(TObject *Sender)
{
    ImageDirEdit->Text = DLB->Directory;
    DLPanel->Hide();
    Sets.ImagesFolder = ImageDirEdit->Text;
    Sets.FNTemplate = FNTemplateEdit->Text;
    FNExampleLabel->Caption = Sets.EvalImgPath();
}
//-----
void __fastcall TSetCommonForm::DLCancelButtonClick(TObject *Sender)
{
}
```

```

    DLPanel->Hide();
}
//-----
void __fastcall TSetCommonForm::AlarmExecuteSpeedButtonClick(
    TObject *Sender)
{
    if(AlarmOpenDialog->Execute()){
        AlarmOpenEdit->Text = AlarmOpenDialog->FileName;
    }
}
//-----
void __fastcall TSetCommonForm::OnAlarmNOExecuteSpeedButtonClick(
    TObject *Sender)
{
    AlarmOpenEdit->Clear();
}
//-----
void __fastcall TSetCommonForm::FNTemplateEditKeyUp(TObject *Sender,
    WORD &Key, TShiftState Shift)
{
    Sets.FNTemplate = FNTemplateEdit->Text;
    FNExampleLabel->Caption = Sets.EvalImgPath();
}
//-----

void __fastcall TSetCommonForm::HideFNHelpButtonClick(TObject *Sender)
{
    FNHelpBox->Visible = false;
}
//-----

void __fastcall TSetCommonForm::SpeedButton2Click(TObject *Sender)
{
    FNHelpBox->Visible = true;
}
//-----

void __fastcall TSetCommonForm::ClearFNTemplateButtonClick(TObject *Sender)
{
    FNTemplateEdit->Text = "{dd}_{mm}_{yyyy}\\{dd}{mmm}{yy}-{hh}{nn}{ss}{zzz}";
    Sets.FNTemplate = FNTemplateEdit->Text;
    FNExampleLabel->Caption = Sets.EvalImgPath();
}
//-----

```

// SetCommonUnit.h - заголовочний файл модуля вибору параметрів
відеоспостереження

```
//-----
#ifndef SetCommonUnitH
#define SetCommonUnitH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Buttons.hpp>
#include "cdiroutl.h"
#include <Grids.hpp>
#include <Outline.hpp>
#include <FileCtrl.hpp>
#include <ExtCtrls.hpp>
#include "CSPIN.h"
#include <Dialogs.hpp>

//-----

class TSetCommonForm : public TForm
{
__published: // IDE-managed Components
    TBitBtn *OkBitBtn;
    TBitBtn *CancelBitBtn;
    TEdit *ImageDirEdit;
    TSpeedButton *ImagePathSpeedButton;
    TPanel *DLPanel;
    TDirectoryListBox *DLB;
    TDriveComboBox *DCB;
    TButton *DLOkButton;
    TButton *DLCancelButton;
    TLabel *Label1;
    TCSpinEdit *JPEGCompressionEdit;
    TLabel *Label2;
    TLabel *Label3;
    TEdit *AlarmOpenEdit;
    TSpeedButton *AlarmExecuteSpeedButton;
    TOpenDialog *AlarmOpenDialog;
    TSpeedButton *OnAlarmNOExecuteSpeedButton;
    TGroupBox *GroupBox1;
    TMemo *AboutMemo;
    TLabel *Label4;
    TEdit *FNTemplateEdit;
    TLabel *FNExampleLabel;
    TSpeedButton *ClearFNTemplateButton;
    TSpeedButton *SpeedButton2;
    TGroupBox *FNHelpBox;
    TMemo *Memo1;
    TButton *HideFNHelpButton;
    void __fastcall OkBitBtnClick(TObject *Sender);
    void __fastcall CancelBitBtnClick(TObject *Sender);
    void __fastcall FormShow(TObject *Sender);
    void __fastcall ImagePathSpeedButtonClick(TObject *Sender);
    void __fastcall DLOkButtonClick(TObject *Sender);
    void __fastcall DLCancelButtonClick(TObject *Sender);
    void __fastcall AlarmExecuteSpeedButtonClick(TObject *Sender);
    void __fastcall OnAlarmNOExecuteSpeedButtonClick(TObject *Sender);
    void __fastcall FNTemplateEditKeyUp(TObject *Sender, WORD &Key,
        TShiftState Shift);
    void __fastcall HideFNHelpButtonClick(TObject *Sender);
    void __fastcall SpeedButton2Click(TObject *Sender);
    void __fastcall ClearFNTemplateButtonClick(TObject *Sender);
};
```

```
private:    // користувацькі об'яви
public:    // користувацькі об'яви
    __fastcall TSetCommonForm(TComponent* Owner);
};
//-----
extern PACKAGE TSetCommonForm *SetCommonForm;
//-----
#endif
```

Кафедра _КБПЗ_ 2023рік