

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 захищеної  
бездротової підсистеми керування об’єктами”**

Виконав здобувач вищої освіти  
IV курсу, групи КБ-19  
ОПП «Кібербезпека»  
спеціальності 125 «Кібербезпека»  
\_\_\_\_\_ Діденко О.О.  
« \_\_\_\_ » \_\_\_\_\_ 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**

*Діденку Олександр Олександровичу*

(прізвище, ім'я, по батькові)

1. Тема роботи	<i>Програмне забезпечення системи кібербезпеки захищеної бездротової підсистеми керування об'єктами</i>
2. Керівник роботи	<i>Смірнова Тетяна Віталіївна, канд. техн. наук</i> (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 12-02 від 5.01.2023 року	
3. Строк подання студентом роботи до захисту	<i>23.05.2023 р.</i>
4. Мета та завдання випускної кваліфікаційної роботи:	<i>Метою роботи є розробка програмного забезпечення системи кібербезпеки захищеної бездротової підсистеми керування об'єктами</i>
5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)	<i>1. Призначення та область використання. 2. Перегляд аналогічних існуючих систем. 3. Опис і обґрунтування проектних рішень. 4. Етапи програмування системи. 5. Впровадження системи кібербезпеки в промислову експлуатацію. 6. Висновки</i>
6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)	
<i>Структурна схема системи кібербезпеки</i>	<i>1 аркуш</i>
<i>Функціональна схема системи кібербезпеки</i>	<i>1 аркуш</i>
<i>Діаграма процесів</i>	<i>1 аркуш</i>
<i>Блок-схема алгоритму роботи додатку</i>	<i>2 аркуша</i>

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

Діденко О.О.  
(прізвище та ініціали)

## АНОТАЦІЯ

**Діденко О.О. Програмне забезпечення системи кібербезпеки захищеної бездротової підсистеми керування об'єктами. 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 захищеної бездротової підсистеми керування об'єктами.

Метою розробки є програмне забезпечення системи кібербезпеки захищеної бездротової підсистеми керування об'єктами.

Результат роботи – програмна реалізація системи кібербезпеки захищеної бездротової підсистеми керування об'єктами.

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

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

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

Програму розроблено в середовищі Delphi 10.4 Sydney.

**Ключові слова:** кібербезпека, захищена бездротова підсистема керування об'єктами

## ABSTRACT

**Didenko O.O. Secure wireless facility management subsystem cybersecurity system software. 125 Cyber security. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.**

In this final qualification work for the first (bachelor) level of higher education, software is developed, which is intended for the cyber security system of the protected wireless subsystem of object management.

The purpose of the development is the software of the cyber security system of the protected wireless subsystem of object management.

The result of the work is the software implementation of the cyber security system of the protected wireless subsystem of object management.

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

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

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

The program was developed in the Delphi 10.4 Sydney environment.

**Keywords:** cybersecurity, secure wireless facility management subsystem

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	8
1.1 Призначення системи.....	8
1.2 Область застосування.....	12
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	23
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	23
2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування.....	32
2.3 Розгорнута постановка завдання .....	38
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	40
3.1 Опис функціонування системи .....	40
3.2 Розробка структурної схеми.....	62
3.3 Розробка функціональної схеми .....	67
3.4 Розробка діаграми процесів.....	71
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	73
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	73
4.2 Захист розробленого програмного забезпечення.....	81
5 ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	84
6 ОСНОВНІ ВИСНОВКИ.....	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	90

**ВКРБ-125.23.0008.00.00.ПЗ**

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Ціденко О.О.			<i>Програмне забезпечення системи кібербезпеки захищеної бездротової підсистеми керування об'єктами</i>	Літ.	Аркуш	Аркушів
Перев.		Смірнова Т.В.				Б	1	100
Н.контр.		Гермак В.С.			ЦНТУ КБ-19			
Затв.		Смірнов О.А.						

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АСКОЕ	–	автоматизовані системи комерційного обліку електроенергії
АСУВ	–	автоматизовані системи управління виробництвом
АСУ ТП	–	автоматизовані системи управління технологічними процесами
ІС	–	інформаційна система
ОС	–	операційна система
ЗЦПС	–	захищена цифрова промислова система
DSSS	–	метод розширення спектра
LAN	–	локальна мережа
RFID	–	радіочастотна ідентифікація
SCADA	–	диспетчерський контроль і накопичення даних
SSID	–	ідентифікатор мережі
Wi-Fi	–	бездротова технологія
WLAN	–	бездротова локальна мережа

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

## ВСТУП

**Актуальність теми.** На сучасний час відома велика кількість теоретичних досліджень, що розглядають різні питання проектування систем управління АСУТП і складними технічними системами. Можна виділити роботи із проектування бездротових мереж із централізованим і децентралізованим управлінням, при високому навантаженні й в умовах перешкод, локальних мереж із багатопроцесорними серверами й т.д.; з питань аналітичної оцінки продуктивності й надійності керуючих обчислювальних комплексів реального часу; із проблем автоматизованого проектування спеціалізованих структур для випробувань складних об'єктів.

Проведені дотепер дослідження створили міцну науково-методологічну базу для проектування структур управління промисловими й технологічними об'єктами. Однак впровадження нових, усе більше досконалих технологій, ставить і ряд нових завдань, що вимагають свого рішення.

В останні кілька років у розвитку промислових мереж чітко виявилися наступні тенденції: відбулася відмова від традиційних схем побудови з вираженим ядром, у якому зосереджені основні обчислювальні потужності, а до виконавчого устаткування тягнеться безліч кабельних з'єднань. Такі рішення не задовольняють сучасним вимогам по масштабованості мережі, надійності й відказостійкості, забезпеченню безпеки, а сам головне – не гарантують виконання вимог за швидкодією для трафіку реального часу – передачі голосу, відео- і команд управління, обсяг якого постійно зростає; відбувся перехід до розподіленої обробки інформації з активним використанням промислових локальних мереж, перепрограмувальних у процесі роботи мікроконтролерів, мікропроцесорів і інтелектуальних датчиків; для управління виробничими й технологічними процесами активно використовується мережа Internet, у тому числі й у режимі реального часу; для побудови центральної частини захищених цифрових промислових систем (ЗЦПС) великих підприємств і складних

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

технічних комплексів використовуються найсучасніші високошвидкісні каналні технології, такі як технологія синхронної цифрової ієрархії SDH/SONET, технологія ущільненого хвильового мультиплексування DWDM, технології 10 Gigabit Ethernet і Metro Ethernet. Для підвищення швидкості передачі маршрутизатори мережі Internet працюють по новій високошвидкісній технології MPLS; на середньому й нижчому рівнях ЗЦПС стали активно використовуватися бездротові технології, такі як Wi-Fi, WiMAX, GSM, 3G/4G/5G і т.п.; впровадження в ЗЦПС вищевказаних нових мережних технологій обумовило появу в мережі нових видів інформаційних потоків:

- інформації від відеокамер, призначених для спостереження за ходом випробувань складних виробів, за ходом технологічного процесу, переміщенням роботів і т.п.;

- передачі екстрених голосових повідомлень; передачі в реальному часі шифрованої мови й відео;

- відеокліпів, файлів великої довжини від цифрових фотоапаратів, програм оперативного перезавантаження флеш-пам'яті керуючого устаткування й т.д.;

- додаткове навантаження на мережу створюють розподілені по мережі компоненти інформаційного захисту, деякі з яких також генерують потоки реального часу, наприклад при передачі екстрених повідомлень про ознаки виявлення атак зловмисників; все більші масштаби приймає впровадження комплексних рішень великих фірм – системних інтеграторів, наприклад проект SCADA TRACE MODE 6. Системи такого роду характеризуються дуже великою функціональністю, інтегрують роботу всіх служб великого підприємства, а в методологічному плані вписуються в концепцію CALS-технології.

Впровадження цілого ряду технологічних новацій породило й цілий ряд проблем. Чисто механічне з'єднання різних мережних технологій може привести й до небажаних наслідків. Без більших витрат на впровадження нових технологій однаково не обійтися, але який при цьому буде досягнутий ефект, не завжди ясно.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Найбільш важкою проблемою є забезпечення в ЗЦПС суперечливих вимог до параметрів якості передачі інформаційних потоків з різною структурою. Передача даних характеризується великим ступенем пульсації трафіку, що неприпустимо для мови, аудіо- і відео- інформації, для яких установлені тверді обмеження на загальну затримку й варіацію величини інтервалів між пакетами. У той же час при передачі дані перекручування й втрати пакетів неприпустимі, а при передачі мови, аудіо- і відео- інформації невеликий відсоток втрат пакетів допускається. Інформаційні потоки не є постійними в часі. Тому комутація й маршрутизація таких потоків можлива тільки при виконанні попереднього дослідження їхньої структури в межах деякого інтервалу часу. Така процедура, обумовлена як профілювання трафіку, є необхідною для управління ресурсами мережі з метою досягнення встановлених показників якості її роботи.

Для досягнення потрібних показників якості виробляється оптимізація структури комп'ютерної мережі. Дане завдання є надзвичайно складним, оскільки на показники якості мережі впливають такі основні параметри протоколів передачі, як час доступу до загального середовища передачі (для неперевавантаженої мережі), величина тайм-ауту непідтверджених пакетів, установлене значення максимальної довжини кадру в проміжній мережі MTU, частка службової інформації в пакеті, час життя пакета TTL і т.д. Якщо ці величини брати як варіюємі змінні, то завдання оптимізації мережі стає чисто комбінаторною з експонентним часом рішення. Спроба вирішити її шляхом натурного моделювання, приречена на невдачу, тому що зміна тільки одного з параметрів вимагає перевавантаження комп'ютера. Параметрами цільової функції є основні показники якості для найбільш важливих додатків, наприклад, середній час затримки при передачі пакетів і її варіація, середнє значення й варіація інтервалів між пакетами й т.д.

Найпростіший, і самий дорогий, шлях рішення цих проблем – це пряме збільшення продуктивності каналів зв'язку й комутаційного устаткування. Однак в умовах твердої конкуренції на ринку мало хто з мережних інтеграторів іде цим

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

шляхом. До останнього часу використовувалися переважно дві стратегії управління потоками в мережі DiffServ і IntServ. Перша з них забезпечує: профілювання трафіку (поділ потоків трафіку на класи) по алгоритму "дірявого ведра", його синхронізацію по алгоритмах типу "ведра токенів", що зменшує величину черг у комутаторах і маршрутизаторах, а також різні способи пріоритетної обробки в проміжних комутаційних пристроях. Показники якості мережі забезпечуються "по можливості", але не гарантуються. Однак достоїнством стратегії DiffServ є те, що вона простіше реалізується на практиці. Стратегія IntServ передбачає гарантоване забезпечення смуги пропускання багатьох користувачів, зокрема гарантованої середньої швидкості передачі, передачі пульсацій трафіку протягом погодженого, щодо невеликого інтервалу часу. При цьому виконуються й інші показники якості, такі, як вимоги до затримок окремих пакетів, вірогідності передачі, часу включення резервного устаткування й т.д. Реалізувати таку стратегію на практиці дуже складно.

Останнім часом рішення проблем забезпечення показників якості в мережах реального часу намітилося в рамках високошвидкісної технології комутації міток MPLS. Ця технологія повинна сполучити достоїнства мережі Internet з розвиненими можливостями її додатків, таких, як Web-, електронна пошта й широкі можливості масштабування, і достоїнства мереж з віртуальними каналами, таких як frame relay і ATM з їхньою високою швидкістю передачі й захищеністю. Мережі такого роду повинні забезпечувати резервування смуги пропускання, інжиніринг трафіку, що забезпечує паралельну й збалансовану передачу трафіку через маршрутизатори MPLS по багатьом шляхам, і, в остаточному підсумку – рішення завдання оптимізації мережі, з урахуванням варіювання параметрів протоколів в окремих ланках мережі. Рішення цієї проблеми покладає на розташовані поза мережею автоматизовані комплекси, що включають у свій состав засобу імітаційного моделювання.

Навіть короткий аналіз особливостей передаваного в ЗЦПС трафіку й попереднього розгляду показників якості говорить про те, що в цих мережах

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

виникають такі ж проблеми оптимізації, що й у мережах MPLS і Metro Ethernet. Більше того, окремі сегменти, як MPLS, так і Metro Ethernet можуть використовуватися в центральній частині ЗЦПС.

Можна констатувати той факт, що технологічний ривок в області створення нового покоління ЗЦПС відбувся, але адекватні йому засоби моделювання й оптимізації ЗЦПС ще не створені. Тому завдання розробки спеціалізованого математичного забезпечення, методів і інструментальних засобів моделювання й оптимізації цифрових промислових мереж, поставлене в даній бакалаврській роботі, є досить актуальною.

В останні роки бездротові мережі передачі даних зайняли міцні позиції в повсякденному житті. Сфера їхнього застосування простирається від забезпечення взаємодії між побутовими приладами (наприклад, між телефоном і телефонною гарнітурою, комп'ютером і монітором і т.д.) до побудови мереж передачі мультимедійної інформації. При створенні бездротових мереж передачі даних все більше поширення одержують пристрої на базі технологій IEEE 802.11 (Wi-Fi). Активний ріст числа бездротових мереж висуває в ряд першочергових завдань розробку методів оптимізації їхньої роботи, розробку нових алгоритмів функціонування таких мереж, а також оцінку їхньої продуктивності. Проблемам розробки математичних моделей мереж передачі даних присвячена значна кількість робіт. Особливості бездротових мереж при оцінці їхньої продуктивності досить повно відбиті в ряді робіт, однак недоліками цих робіт є, слабка увага, що приділяється механізмам адаптивного централізованого управління, хоча саме ці механізми націлені на рішення основної проблеми промислових бездротових мереж – проблеми „прихованих станцій". Таким чином, завдання аналізу, розробки й оптимізації механізмів адаптивного динамічного управління є однією з найважливіших для розвитку бездротових широкосмугових промислових мереж передачі даних. Крім цього, потрібно розробити комплекс аналітичних і імітаційних моделей механізмів централізованого управління для одержання

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

адекватних оцінок показників продуктивності й оптимізації пристроїв широкопasmової бездротової промислової мережі.

**Мета й завдання дослідження.** Метою роботи є програмне забезпечення системи кібербезпеки захищеної бездротової підсистеми керування об'єктами.

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

- Огляд існуючих систем захищеної бездротової підсистеми керування об'єктами.
- Дослідження системи кібербезпеки захищеної бездротової підсистеми керування об'єктами.
- Програмна реалізація системи кібербезпеки захищеної бездротової підсистеми керування об'єктами.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі захищеної бездротової підсистеми керування об'єктами.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки захищеної бездротової підсистеми керування об'єктами, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>8</b>

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Система призначена для захищеного управління об'єктами у АСУ ТП за допомогою технології Wi-Fi. Wi-Fi – торговельна марка Wi-Fi Alliance для бездротових мереж на базі стандарту IEEE 802.11. Термін ніяк не розшифровується. Будь-яке устаткування, що відповідає стандарту IEEE 802.11, може бути протестовано в Wi-Fi Alliance і одержати відповідний сертифікат.

Звичайно схема Wi-Fi мережі містить не менш однієї точки доступу й не менш одного клієнта. Також можливе підключення двох клієнтів у режимі точка-точка (Ad-hoc), коли точка доступу не використовується, а клієнти з'єднуються за допомогою мережних адаптерів «прямо». Точка доступу передає свій ідентифікатор мережі (SSID) за допомогою спеціальних сигнальних пакетів на швидкості 0,1 Мбіт/с кожні 100 мс. Тому 0,1 Мбіт/с – найменша швидкість передачі даних для Wi-Fi. Знаючи SSID мережі, клієнт може з'ясувати, чи можливо підключення до даної точки доступу. При влученні в зону дії двох точок доступу з ідентичними SSID приймач може вибирати між ними на підставі даних про рівень сигналу. Стандарт Wi-Fi дає клієнтові повну волю при виборі критеріїв для з'єднання. Більш докладно принцип роботи описаний в офіційному тексті стандарту. Однак, стандарт не описує всі аспекти побудови бездротових локальних мереж Wi-Fi. Тому кожний виробник устаткування вирішує це завдання по-своєму, застосовуючи ті підходи, які він вважає найкращими з тієї або іншої точки зору. Тому виникає необхідність класифікації способів побудови бездротових локальних мереж.

За способом об'єднання точок доступу в єдину систему можна виділити:

– Автономні точки доступу (називаються також самостійні, децентралізовані, розумні).

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

– Точки доступу, що працюють під управлінням контролера (називаються також «легковагові», централізовані).

– Безконтролерні, але не автономні (керовані без контролера).

За способом організації й управління радіоканалами можна виділити бездротові локальні мережі:

– Зі статичними налаштуваннями радіоканалів.

– З динамічними (адаптивними) налаштуваннями радіоканалів.

– З «шаруватою» або багат шаровою структурою радіоканалів.

Переваги Wi-Fi:

– Дозволяє розгорнути мережу без прокладки кабелю, що може зменшити вартість розгортання й/або розширення мережі. Місця, де не можна прокласти кабель, наприклад, поза приміщеннями й у будинках, що мають історичну цінність, можуть обслуговуватися бездротовими мережами.

– Дозволяє мати доступ до мережі мобільним пристроям.

– Wi-Fi пристрої широко поширені на ринку. Гарантується сумісність устаткування завдяки обов'язковій сертифікації устаткування з логотипом Wi-Fi.

– Випромінювання від Wi-Fi пристроїв у момент передачі даних на два порядки (в 100 разів) менше, ніж у стільникового телефону.

Недоліки Wi-Fi:

– У діапазоні 2.4 GHz працює безліч пристроїв, таких як пристрої, що підтримують Bluetooth, і ін, й даже мікрохвильові печі, що погіршує електромагнітну сумісність.

– Реальна швидкість передачі даних в Wi-Fi мережі завжди нижче максимальної швидкості, що заявляється виробниками Wi-Fi устаткування.

Реальна швидкість залежить від багатьох факторів: наявності між пристроями фізичних перешкод (меблі, стіни), наявності перешкод від інших бездротових пристроїв або електронної апаратури, розташування пристроїв друг щодо друга й т.п.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

– Кількість одночасно-спостережуваних Wi-Fi мереж в одній точці не може бути більше кількості використовуваних каналів, тобто 13 каналів/мереж в Україні (дане обмеження все частіше відчувається в багатоквартирних будинках).

– Частотний діапазон і експлуатаційні обмеження в різних країнах неоднакові. У багатьох європейських країнах дозволені два додаткових канали, які заборонені в США; У Японії є ще один канал у верхній частині діапазону, а інші країни, наприклад Іспанія, забороняють використання низькочастотних каналів. Більше того, деякі країни, наприклад Росія, Білорусь і Італія, вимагають реєстрації всіх мереж Wi-Fi, що працюють поза приміщеннями, або вимагають реєстрації Wi-Fi-оператора.

– Як було згадано вище – в Україні точки бездротового доступу, а також адаптери Wi-Fi з ЕІМ, що перевищує 100 мВт (20 дБм), підлягають обов'язковій реєстрації.

– Стандарт шифрування WEP може бути відносно легко зламаний навіть при правильній конфігурації (через слабку стійкість алгоритму). Нові пристрої підтримують протокол шифрування даних WPA і WPA2. Прийняття стандарту IEEE 802.11i (WPA2) зробило доступною більше безпечну схему, що доступна в новому обладнанні. Обидві схеми вимагають більше стійкий пароль, ніж ті, які звичайно призначаються користувачами. Багато організацій використовують додаткове шифрування (наприклад VPN) для захисту від вторгнення. На даний момент основним методом злому WPA2 є підбор пароля, тому рекомендується використовувати складні цифро-буквені паролі для того, щоб максимально ускладнити завдання підбора пароля.

– У режимі ad-hoc стандарт пропонує лише реалізувати швидкість 11 Мбіт/сек (802.11b). Шифрування WPA(2) недоступно, тільки WEP.

### **Бездротові технології в промисловості**

Для використання в промисловості технології Wi-Fi пропонуються поки обмеженим числом постачальників. Так Siemens Automation & Drives пропонує Wi-Fi-рішення для своїх контролерів SIMATIC у відповідності зі стандартом

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

IEEE 802.11g у вільному ISM-діапазоні 2,4 ГГц і забезпечуючий максимальну швидкість передачі 54 Мбіт/с. Дані технології застосовуються в основному для управління об'єктами, що рухаються, і в складській логістиці, а також у тих випадках, коли з якої-небудь причини неможливо прокласти провідні мережі Ethernet.

На Україні використання Wi-Fi без дозволу Українського державного центра радіочастот можливо лише у випадку використання точки доступу зі стандартною всеспрямованою антеною (<6 дБ, потужність сигналу  $\leq 100$  мВт на 2.4 ГГц і  $\leq 200$  мВт на 5 ГГц) для внутрішніх (використання усередині приміщення) потреб організації (Рішення Національної комісії з регулювання зв'язку України № 914 від 2007.09.06) У випадку сигналу більшої потужності або надання послуг доступу в Інтернет, або до яких-небудь ресурсів, необхідно реєструвати передавач і одержати ліцензію УДЦР.

## 1.2 Область застосування

Визначимо область застосування системи, яка розробляється. Областю застосування системи є автоматизована інформаційна система на підприємстві.

Проведемо огляд автоматизованих інформаційних систем на підприємстві. Пропонується використовувати наступну класифікацію систем і підсистем ІС. Залежно від рівня обслуговування виробничих процесів на підприємстві сама ІС або його складова частина (підсистеми) можуть бути віднесені до різних класів:

– Клас А: системи (підсистеми) управління технологічними об'єктами й/або процесами.

– Клас В: системи (підсистеми) підготовки й обліку виробничої діяльності підприємства.

– Клас С: системи (підсистеми) планування й аналізу виробничої діяльності підприємства.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Системи (підсистеми) класу А – системи (підсистеми) контролю й управління технологічними об'єктами й/або процесами. Ці системи, як правило, характеризуються наступними властивостями:

- досить високим рівнем автоматизації виконуваних функцій;
- наявністю явно вираженої функції контролю за поточним станом об'єкта управління;
- наявністю контуру зворотного зв'язка;
- об'єктами контролю й управління такої системи виступають:
  - а) технологічне устаткування;
  - б) датчики;
  - в) виконавчі пристрої й механізми.
- малим часовим інтервалом обробки даних (тобто інтервалом часу між одержанням даних про поточний стан об'єкта управління й видачею керуючого впливу на нього);
- слабкою (несуттєвою) часовою залежністю (кореляцією) між станами, об'єктів управління й системи (підсистеми) управління, що динамічно змінюються.

Як класичні приклади систем класу А можна вважати:

- SCADA – Supervisory Control And Data Acquisition (диспетчерський контроль і накопичення даних).
- DCS – Distributed Control Systems (розподілені системи управління).
- Batch Control – системи послідовного управління.
- АСУ ТП – Автоматизовані Системи Управління Технологічними Процесами.

Системи класу В – це системи (підсистеми) підготовки й обліку виробничої діяльності підприємства. Системи класу В призначені для виконання класу завдань, що вимагають особистої участі людини для прийняття оперативних (тактичних) рішень, що роблять вплив на обмежене коло видів діяльності або невеликий період роботи підприємства. У деякому змісті до таких

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

систем прийнято відносити ті, які перебувають на рівні технологічного процесу, але з технологією прямо не зв'язані.

У перелік основних функцій систем (підсистем) даного класу можна включити:

- виконання облікових завдань, що виникають у діяльності підприємства;
- збір, попередню підготовку даних, що надходять в ІС із систем класу А, і їхню передачу в системи класу С;
- підготовку даних і завдань для автоматичного виконання завдань системами класу А.

Класичними прикладами систем класу В можна вважати:

- MES – Manufacturing Execution Systems (системи управління виробництвом).
- MRP – Material Requirements Planning (системи планування потреб у матеріалах).
- MRP II – Manufacturing Resource Planning (системи планування ресурсів виробництва).
- CRP – C Resource Planning (система планування виробничих потужностей).
- CAD – Computing Aided Design (автоматизовані системи проектування – САПР).
- CAM – Computing Aided Manufacturing (автоматизовані системи підтримки виробництва).
- CAE – Computing Aided Engineering (автоматизовані системи інженерного проектування – САПР).
- PDM – Product Data Management (автоматизовані системи управління даними).
- CRM – Customer Relationship Management (системи управління взаєминами із клієнтами).

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

– ITSM – IT Service Management (автоматизована система сервісу інформаційних технологій)

– Усілякі облікові системи й т.п.

Одна із причин виникнення подібних систем – необхідність виділити окремі завдання управління на рівні технологічного підрозділу підприємства.

Системи класу С – це системи (підсистеми) планування й аналізу виробничої діяльності підприємства. Системи класу С призначені для виконання класу завдань, що вимагають особистої участі людини для прийняття стратегічних рішень, що роблять вплив на діяльність підприємства в цілому.

У коло завдань розв'язуваних системами (підсистемами) даного класу можна включити:

– аналіз діяльності підприємства на основі даних і інформації, що надходить із систем класу В;

– планування діяльності підприємства;

– регулювання глобальних параметрів роботи підприємства;

– планування й розподіл ресурсів підприємства;

– підготовку виробничих завдань і контроль їхнього виконання.

– наявність взаємодії з керуючим суб'єктом (персоналом), при виконанні вартих перед ними завдань;

– інтерактивність обробки інформації;

– підвищеною тривалістю обробки даних, що коливається від декількох мінут до кілька годин або доби;

– тривалим періодом ухвалення керуючого рішення;

– наявністю істотних часової й параметричної залежностей (кореляцій) між оброблюваними даними;

– система впливає на діяльність підприємства в цілому;

– система впливає на значний період роботи підприємства (від напівроку до декількох років);

– наявністю безпосереднього сполучення із системами класу В.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Класичними назвами системи класу С можна вважати:

- EAM – Enterprise Asset Management (управління фондами й активами підприємства).
- ERP – Enterprise Resource Planning (Планування Ресурсів Підприємства).
- IRP – Intelligent Resource Planning (системами інтелектуального планування).
- АСУВ – автоматизовані системи управління виробництвом.

### **Системи класу А**

#### **АСКОЕ – автоматизовані системи комерційного обліку електроенергії**

Одним з найважливіших компонентів ринку електроенергії є його інструментальне забезпечення, що являє собою сукупність систем, приладів, пристроїв, каналів зв'язку, алгоритмів і т.д. для контролю й управління параметрами енергоспоживання. Базою формування й розвитку інструментального забезпечення є автоматизовані системи контролю й обліку споживання електроенергії.

Розвиток ринку електроенергії на основі економічного методу управління зажадало створення повномасштабних ієрархічних систем: автоматизованих систем виміру електроенергії (АСВЕ), обліку споживання й збуту електроенергії (АСОСЗЕ), диспетчерського управління (АСДУ), контролю й обліку енергоспоживання (АСКОЕ).

#### **АСУТП – автоматизовані системи управління технологічними процесами**

Основою для впровадження промислової автоматизації служать так звані системи SCADA. Термін SCADA – це скорочення англійського терміна Supervisory Control And Data Acquisition – диспетчерське управління й збір даних. Дотепер більшість SCADA-пакетів застосовувалося, як правило, для створення інтерфейсу оператора й реєстрації даних виробничого процесу. У рідких випадках до цього додавалися можливості по автоматичному управлінню й генерації звітів. Основними причинами, що стримують комплексне впровадження

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

SCADA-систем на промислових об'єктах, були недостатня надійність таких систем, низька продуктивність, труднощі в нарощуванні й інтеграції SCADA з корпоративними системами управління й складності адаптації до реальних виробничих умов і алгоритмів.

З появою SCADA-пакетів нового покоління фірми, що займаються промисловою автоматизацією, відділи АСУТП на підприємствах і компанії – системні інтегратори одержали можливість використовувати на практиці величезний потенціал, пропонований такими системами. Відсутність у нових SCADA-системах недоліків, що стримували їхнє впровадження в минулому, дозволяє на їхній основі будувати інтегровані системи управління як для дуже більших, так і компактних систем АСУТП у будь-якій області промисловості.

### **Управління апаратурою, візуалізація, моніторинг**

SCADA пакети, на відміну від більшості програм, безпосередньо пов'язані із процесами, що відбуваються на підприємстві, оскільки через систему серверів вводу-виводу підключаються до різної апаратури, що управляє й контролює виробничий процес:

– Сигнали тривоги. Поряд з алгоритмами обробки в сучасній SCADA – системі передбачена можливість автоматичної генерації сигналів тривоги відповідно до заданих інженерів або технологами критеріями. Виникаючі сигнали тривоги можуть відображатися на екрані, записуватися в журнал і бути доступні одночасно для декількох робочих місць оператора. Підтвердження цих сигналів, або квітування здійснюється операторами, що мають відповідні права доступу до системи. З кожною тривоگوю можна зв'язати певна дія, що буде виконуватися з появою цього тривоги (наприклад, запуск звукового файлу).

– Графічні засоби SCADA – системи Citect RAD. Графіки будуються на базі досить простого набору графічних об'єктів, а саме: прямокутників, еліпсів, точкових зображень, відрізків, кривих, ламаних, тексту, символів і труб. У кожного об'єкта є деякий, загальний для всіх набір властивостей. Всі вони можуть бути безпосередньо пов'язані з параметрами контрольованого устаткування, які будуть визначати поведінку графічних об'єктів.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

– Генерація звітів. Звіт SCADA – системи Citect – це документ, що відбиває деякі виробничі показники й надаваємий періодично, за запитом або при виникненні якої-небудь події, наприклад, при зміні стану який-небудь змінної, у момент запуску SCADA – системи Citect або в зазначений час дня. Звіти можуть генеруватися в будь-якому зручному для користувача форматі. У нього може входити форматований текст, оперативна й інформація, що накопичується, і навіть результати математичних обчислень. Крім того, звіти можуть містити й деякі команди: заміна виробничих параметрів, завантаження інструкцій, виконання діагностики, зміна составів сумішей і т.д. Звіти можуть виводитися на екран, роздруковуватися, а також зберігатися на диску для наступної роздруківки або перегляду. Звіт можна обробляти засобами будь-якого текстового редактора. Їх можна автоматично зберігати в SQL-базах і інших ODBC-сумісних базах даних.

– Захист від несанкціонованого доступу. Практично у всіх системах певний набір дій повинен виконуватися тільки вповноваженим персоналом. Використовуваний людино-машинний інтерфейс повинен забезпечувати певний рівень захисту щоб уникнути випадкового або навмисного виконання заборонених операцій. Захист від несанкціонованого доступу інтегрований в усі інтерфейсні елементи SCADA – системи Citect, гарантуючи повну безпеку виконавчої системи.

– Резервування. У промислових системах автоматизації й інших відповідальних додатків відмови устаткування приводять до з виробництва й іноді до виникнення потенційно небезпечних ситуацій. Усувати відмови в системі без втрати її функціональних можливостей і продуктивності дозволяє реалізація функцій резервування.

### **Системи класу В**

MES – (Manufacturing Enterprise Solutions) інформаційна й комунікаційна система виробничого середовища підприємства. За визначенням APICS (American Production and Inventory Control Society) MES – це інформаційна й

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

комунікаційна система виробничого середовища підприємства. Більше розгорнутим є визначення, прийняте в некомерційній асоціації MESA (Manufacturing Enterprise Solutions Association), що поєднує виробників і консультантів-впроваджувачів MES-систем:

MES – це автоматизована система управління виробничою діяльністю підприємства, що у режимі реального часу:

- планує;
- оптимізує;
- контролює;
- документує виробничі процеси від початку формування замовлення до випуску готової продукції.

Системи MES визначаються як сукупність програмних функцій, що відрізняються від функцій систем планування ресурсів підприємства (ERP), автоматизованого проектування й програмування (CAD/CAM) і автоматизованих систем управління технологічним процесом (АСУТП).

Асоціація MESA визначила 11 основних функцій MES:

1. Контроль стану й розподіл ресурсів (RAS). Ця функціональність MES-систем забезпечує управління ресурсами виробництва (машинами, інструментальними засобами, методиками робіт, матеріалами, устаткуванням) і іншими об'єктами, наприклад, документами про порядок виконання кожної виробничої операції. У рамках цієї функції описується детальна історія ресурсів і гарантується правильність настроювання устаткування у виробничому процесі, а також відслідковується стан устаткування в режимі реального часу.

2. Оперативне/Детальне планування (ODS). Ця функція забезпечує оперативне й детальне планування роботи, засноване на пріоритетах, атрибутах, характеристиках і властивостях конкретного виду продукції, а також детально й оптимально обчислює завантаження устаткування при роботі конкретної зміни.

3. Диспетчеризація виробництва (DPU). Забезпечує поточний моніторинг і диспетчеризацію процесу виробництва, відслідковуючи виконання операцій,

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

зайнятість устаткування й людей, виконання замовлень, обсягів, партій і контролює в реальному часі виконання робіт відповідно до плану. У режимі реального часу відслідковуються всі зміни, що відбуваються, і вносяться коректування в план цеху.

4. Управління документами (DOC). Контролює зміст і проходження документів, які повинні супроводжувати випускається изделие, що, включаючи інструкції й нормативи робіт, способи виконання, креслення, процедури стандартних операцій, програми обробки деталей, запису партій продукції, повідомлення про технічні зміни, передачу інформації від зміни до зміни, а також забезпечує можливість вести планову й звітну цехову документацію. Передбачається архівування інформації.

5. Збір і зберігання даних (DCA). Ця функція забезпечує інформаційну взаємодію різних виробничих підсистем для одержання, накопичення й передачі технологічних і керуючих даних, що циркулюють у виробничому середовищі підприємства. Дані про хід виробництва можуть уводитися як вручну персоналом, так і автоматично із заданою періодичністю з АСУТП або безпосередньо з виробничих ліній.

6. Управління персоналом (LM). Надає інформацію про персонал із заданою періодичністю, включаючи звіти про час і присутність на робочому місці, спостереження за відповідністю сертифікації, а також можливість урахувати й контролювати основні, додаткового й обов'язки, що сполучаються, персоналу, такі як виконання підготовчих операцій, розширення зони роботи.

7. Управління якістю продукції (QM). Надає дані вимірів про якість продукції, у тому числі й у режимі реального часу, зібрані з виробничого рівня, забезпечуючи належний контроль якості й звертаючи пильну увагу на критичних точках. Може запропонувати дії по виправленню ситуації в даній точці на основі аналізу кореляційних залежностей і статистичних даних причинно-наслідкових зв'язків контрольованих подій.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

8. Управління виробничими процесами (PM). Відслідковує заданий виробничий процес, а також автоматично вносить коректування або пропонує відповідне рішення операторові для виправлення або підвищення якості поточних робіт.

9. Управління виробничими фондами (техобслуговування) (MM). Підтримка процесу технічного обслуговування, планового й оперативного ремонту виробничого й технологічного устаткування й інструментів протягом усього виробничого процесу.

10. Відстеження історії продукту (PTG). Надає інформацію про те, де й у якому порядку велася робота з даною продукцією. Інформація про стан може містити в собі: звіт про персонал, що працює із цим видом продукції, компоненти продукції, матеріали від постачальника, партію, серійний номер, що течуть умови виробництва, невідповідності встановленим нормам, індивідуальний технологічний паспорт виробу.

11. Аналіз продуктивності (PA). Надає звіти про реальні результати виробничих операцій, а також порівнює з попередніми й очікуваними результатами. Представлені звіти можуть містити в собі такі виміри, як використання ресурсів, наявність ресурсів, час циклу виробничого ресурсу, відповідність плану, стандартам і інші.

Система управління виробництвом – це сполучна ланка між орієнтованими на господарські операції ERP-системами, системами планування ланцюжка поставок і діяльністю в реальному масштабі часу на рівні виробництва.

Деякі функції MES деякою мірою перекриваються з іншими системами, які, у свою чергу, перекриваються між собою. Ступінь їхнього перекриття залежить від конкретного завдання, виду галузі й способу реалізації системи:

- MES – Система управління виробництвом.
- SSM – Система управління збутом і обслуговуванням.
- SCM – Система управління ланцюжком поставок.
- ERP – Система планування ресурсів виробництва.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

– P/PE – Система проектування виробничих процесів і продукції.

– Industrial (АСУТП) – PLC, розподілені системи управління, засобу автоматизації. Так, і в MES, і в системі управління мережею поставок є календарне планування; функції управління трудовими ресурсами є в MES, у системі управління збутом і обслуговуванням і в підсистемі управління кадрами системи планування виробництва; документообіг контролюється й MES, і системою проектування процесів і продукції; управління технологічними процесами здійснюється як MES, так і пристроями автоматизації.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи кібербезпеки захищеної бездротової підсистеми керування об'єктами, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

## 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

Проведемо огляд сучасного програмного забезпечення захищених систем керування об'єктами.

Програмне забезпечення для автоматизованих систем керування підрозділяються в такий спосіб:

- SCADA-системи.
- Оперативно-інформаційні комплекси.
- Система планування й диспетчеризації виробництва.
- Бази даних реального часу.
- Платформи для промислової автоматизації.
- Засоби аналізу технологічних даних і системи генерації звітів.
- ПЗ для систем оперативного керування виробництвом (MES).
- Засоби розробки.
- Рішення для галузей.

#### **Wonderware InTouch Compact Edition**

Програмне забезпечення Wonderware InTouch HMI, сумісне із пристроями на базі Windows CE.

Wonderware InTouch Compact Edition надає широкі можливості графічної візуалізації й доступний разом із пристроями на базі Windows CE. Для активації попередньо встановлюваного HMI необхідно вибрати розмір ліцензій, які пропонуються окремо або в пакетах. Wonderware InTouch Compact Edition має можливість сполучення з безліччю драйверів для пристроїв уведення-виводу

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

одночасно. Число підключень визначається кількістю тегів; мале число тегів допускає один драйвер, велике – до трьох драйверів.

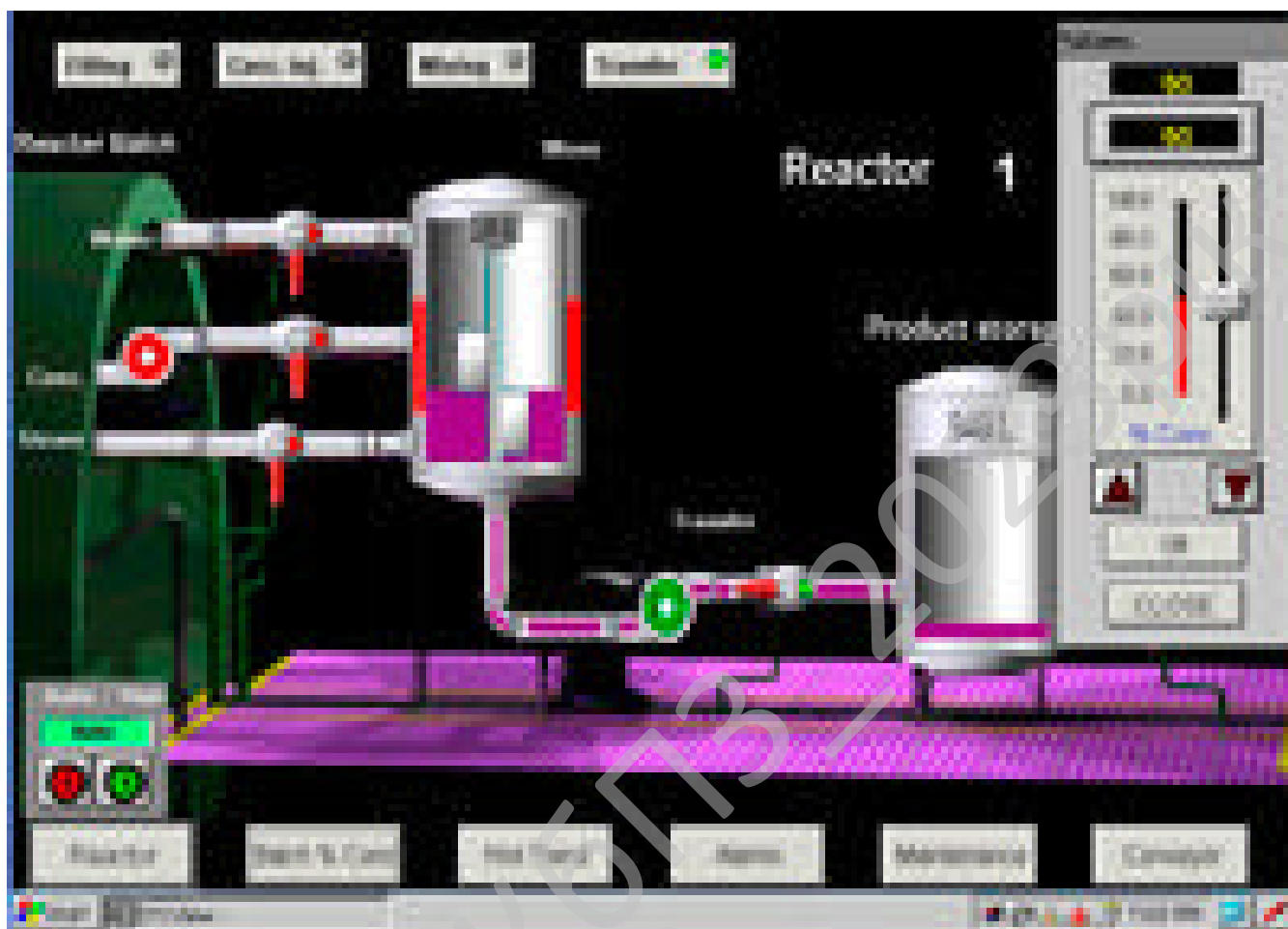


Рисунок 2.1 – Інтерфейс користувача Wonderware InTouch Compact Edition

Основні характеристики:

– Централізована розробка й конфігурування додатків, завантаження на операторські станції.

– Широкий набір інструментарію створення сценаріїв для розширення й налаштування додатків у відповідності зі специфікацією системи.

– Інтеграція панелей і операторських інтерфейсів з архітектурою Wonderware Archestr.

– Убудовані тренди реального часу й історичні тренди, моніторинг алармів і реєстрація.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

Ключові переваги:

- Підвищення періоду продуктивної експлуатації для додатків НМІ шляхом використання твердотільних накопичувачів з підтримкою CE.
- Скорочення витрат з використанням систем, які не вимагають або вимагають у мінімальному ступені технічне обслуговування на місці, а також мінімального навчання користувачів.
- Економія часу й грошей за допомогою протестованого й передвстановленого програмного забезпечення на пристроях, що працюють під керуванням Windows CE.
- Простота у використанні програмного забезпечення НМІ на всіх рівнях виробництва.
- Скорочення витрат і часу на технічне обслуговування шляхом використання єдиного середовища розробки для систем на базі операційної системи Windows CE і Windows XP/Vista.

### **СК 200X**

Оперативно-інформаційний комплекс, призначений для прийому, обробки, передачі й зберігання телеметричної інформації.

Оперативно-інформаційний комплекс (ОІК) СК-2003 призначений для прийому, обробки, передачі й зберігання телеметричної інформації про режим роботи енергетичної системи (рівень ОДУ, РДУ, АО-енерго), що надходить у реальному часі, і надання оперативно-диспетчерському персоналу доступу до неї.

З 2004 року комплекс впроваджується сертифікованими фахівцями компанії "Ртсофт" на підприємствах електроенергетики.

ОІК СК-2003 обмінюється даними із системою збору інформації двома способами – через сервери телемеханіки й по каналах міжрівневого (міжмашиного) обміну. В обох випадках використовуються відповідні протоколи обміну.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Функції:

1. Прийом телеметричної інформації.
2. Міжмашинний обмін.
3. Обмін макетами ЦБІ.
4. Обробка телеметричної інформації.
5. Архівування інформації.
6. Оцінка стану.
7. Повідомлення про події.
8. Універсальний дорозрахунок.
9. Система відображення (вивід і занесення інформації):
  - Панель керування.
  - Студія НСІ.
  - Енергетичний календар.
  - Таблиці.
  - Схеми.
  - Система векторної графіки TOPAZ.
  - Web-інтерфейс.
  - Звуковий супровід.
  - Списки.
  - Графіки.
10. Диспетчерські позначки.
11. Добова відомість.
- 12 Звітні документи.
13. Контроль перетікань у небезпечних перерізах.
14. Контроль рівнів напруги.
15. Контроль виконання диспетчерського графіка.
16. Контроль роботи протиаварійної автоматики.
17. Підсистема планування режиму.
- 18 Режимний тренажер.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

## Wonderware Performance Software

Програмне забезпечення для відстеження простоїв устаткування й контролю ефективності виробництва (MES-модуль).

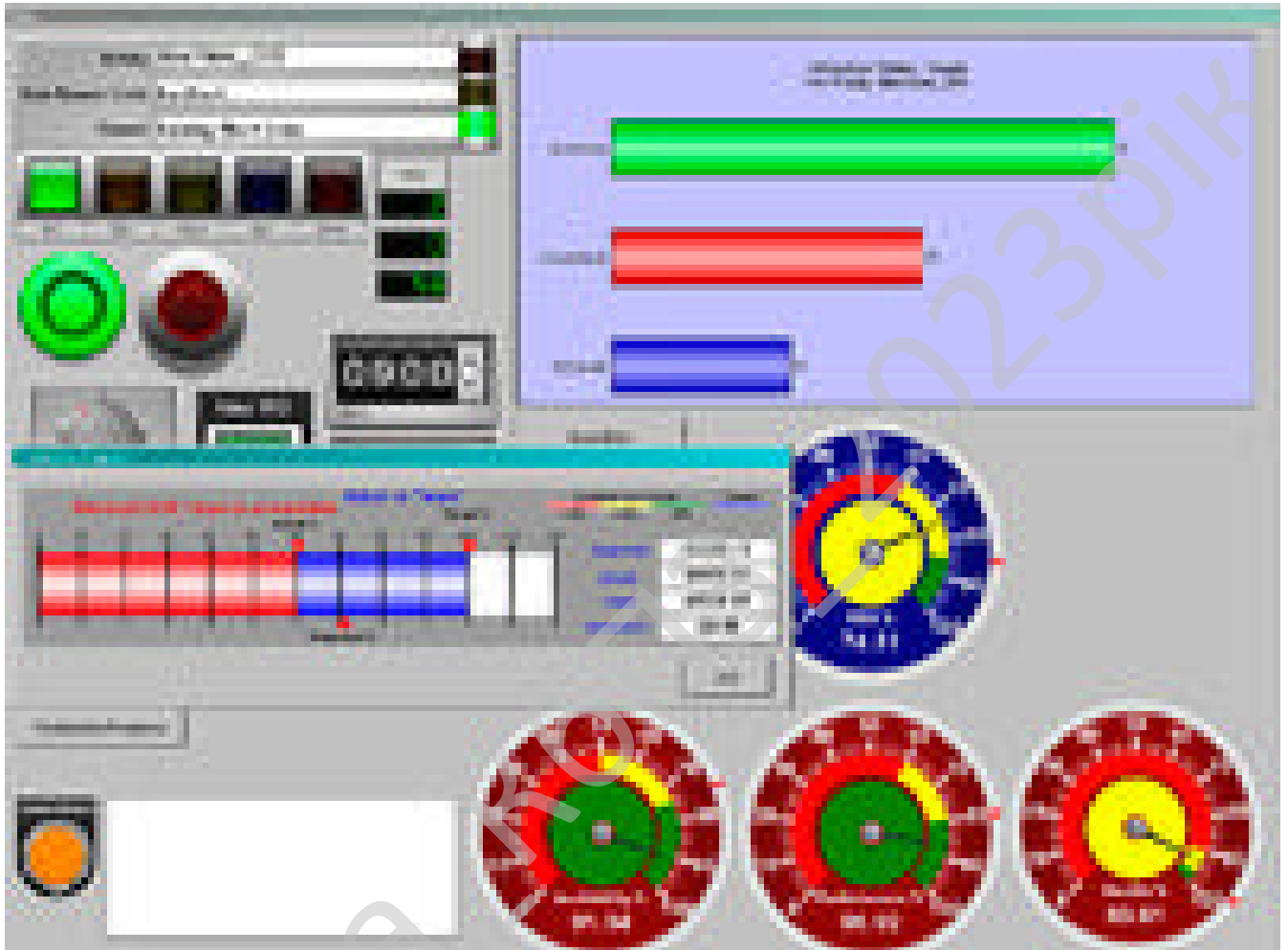


Рисунок 2.2 – Інтерфейс користувача Wonderware Performance Software

Додаток Wonderware Performance Software призначений для контролю ефективності виробництва (раніше відомо як Factelligence) і надає програмне рішення рівня підприємства для збору, відстеження й передачі інформації реального часу про продуктивність і ефективність роботи устаткування.

Wonderware Performance Software надає критичну інформацію про простої й ефективність устаткування операторам і керівникам, які можуть почати негайні дії для підвищення ефективності виробництва.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Wonderware Performance Software легко конфігурується й повністю використовує можливості системної платформи Wonderware, що забезпечує численні переваги, такі як інтеграція, розробка й генерування звітів за рахунок технології Archestr, що лежить у її основі. Даний підхід знижує витрати по впровадженню й підтримці, а також полегшує прискорене поширення додатка в масштабах усього підприємства.

Основні характеристики:

- Відстеження простоїв устаткування.
- Overall Equipment Effectiveness (OEE) моніторинг.
- Конфігурація на базі шаблонів для швидкої розробки/розгортання.
- Спеціалізований дисплей оператора на базі .NET.
- Можливості генерації звітів на базі web за допомогою служби Microsoft Reporting Services.

- Широкі можливості нарощування й взаємодії додатків.

Ключові переваги:

- Wonderware OEE – засобу для підвищення продуктивності й загальної ефективності експлуатації устаткування й генерації звітів.
- Поточний контроль експлуатаційних показників устаткування з метою максимізації використання основних фондів.
- Поширення передових методик і досягнень удосконалювання виробництва завдяки універсальним і масштабованим рішенням, спрямованим на вдосконалення виробничих технологій.

### **Wonderware Development Studio**

Комплексний набір інструментів для швидкої розробки й розгортання додатків Wonderware.

Рішення Wonderware Development Studio надає комплексний набір інструментів для швидкої розробки й розгортання додатків Wonderware.

Ядром Development Studio служить інтегроване компонентне середовище розробки (IDE), у рамках якої інженерний персонал може займатися розробкою,

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

створенням, тестуванням і розгортанням будь-яких промислових додатків автоматизації.

Не залишаючи середовища розробки IDE, користувачі можуть створювати ефективні й складні графічні подання й включати їх у додатки HMI, SCADA, MES або системи виробничого інтелекту. Як тільки розробка готова до доставки, одне клацання – це все, що необхідно для розгортання додатків, об'єктів і відповідної логіки.

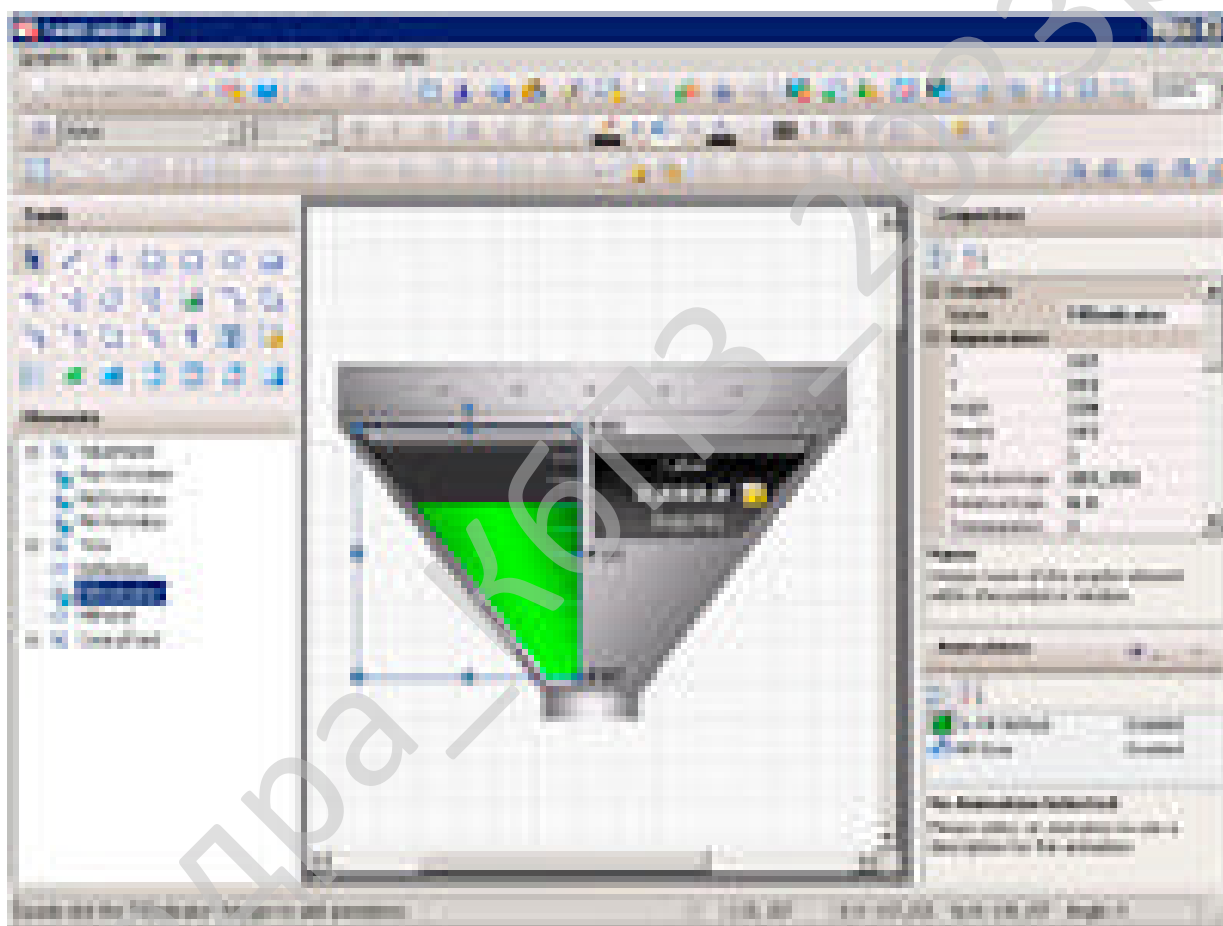


Рисунок 2.3 – Інтерфейс користувача Wonderware Development Studio

Основні характеристики:

- Шаблонно-орієнтована розробка об'єктів і графічних елементів.
- Можливості розширення при розробці на базі Microsoft .NET і використанні Wonderware API.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- Повна й набудовувати^ся бібліотека, що, графічних елементів.
- Багаторівневий захист і журнали реєстрації подій.
- Керування пристроями уведення-виводу, архівом, алармами, подіями, графікою, сценаріями й іншими за допомогою одного інструмента.

Ключові переваги:

- Єдине середовище розробки централізованого керування й вилученого розгортання додатків HMI, SCADA, MES і EMI.
- Простота використання.
- Централізоване багатокористувальницьке середовище для розробки.
- Можливість повторного використання технічних наробітків, створених у попередніх проектах.
- Швидке поширення змін по мережних вузлах.
- Підвищення продуктивності.

У зв'язку з тим, що у даній бакалаврській роботі розглядається бездротова система керування об'єктами, проведемо огляд підтримки Wi-Fi у різних операційних системах.

ОС сімейства BSD (FreeBSD, NetBSD, OpenBSD) можуть працювати з більшістю адаптерів, починаючи з 1998 року. Драйвери для чипів Atheros, Prism, Harris/Intersil і Aironet (від відповідних виробників Wi-Fi пристроїв) звичайно входять в ОС BSD починаючи з версії 3. В OpenBSD 3.7, було включено більше драйверів для бездротових чипів, включаючи RealTek RTL8180L, Ralink RT25x0, Atmel AT76C50x, і Intel 2100 і 2200BG/2225BG/2915ABG. Завдяки цьому частково вдалося вирішити проблему недостатчі відкритих драйверів бездротових чипів для OpenBSD. Можливо деякі драйвери, реалізовані для інших BSD-систем, можуть бути перенесені, якщо вони ще не були створені. NDISwrapper також доступний для FreeBSD.

Mac OS. Адаптери виробництва Apple підтримувалися із системи Mac OS 9, випущеної в 1999 році. З 2006 року всі настільні комп'ютери й ноутбуки Apple Inc. (а також, що з'явилися пізніше телефони, iPhone, плеєри iPod Touch і

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

планшетні комп'ютери iPad) штатно оснащуються адаптерами Wi-Fi, мережа Wi-Fi у цей час є основним рішенням Apple для передачі даних, і повністю підтримується Mac OS X. Можливий режим роботи адаптера комп'ютера як точки доступу, що дозволяє при необхідності зв'язувати комп'ютери Macintosh у бездротові мережі у відсутності інфраструктури. Darwin і Mac OS X, незважаючи на частковий збіг з BSD, мають свою власну, унікальну реалізацію Wi-Fi.

Linux: Починаючи з версії 2.6, підтримка деяких Wi-Fi пристроїв з'явилася безпосередньо в ядрі Linux. Підтримка для чипів Orinoco, Prism, Aironet, Atmel, Ralink включена в основну галузь ядра, чипи ADMtek і Realtek RTL8180L підтримуються як закритими драйверами виробників, так і відкритим, написаним співтовариством. Intel Calexico підтримуються відкритими драйверами, доступними на SourceForge.net. Atheros підтримується через відкриті проекти. Підтримка інших бездротових пристроїв доступна при використанні відкритого драйвера NDISwrapper, що дозволяє Linux-системам, що працюють на комп'ютерах з архітектурою Intel x86, «обертати» драйвера виробника для Microsoft Windows для прямого використання. Відома принаймні одна комерційна реалізація цієї ідеї. FSF створило список адаптерів, що рекомендуються, більше докладну інформацію можна знайти на сайті Linux wireless.

Існує досить велика кількість Linux-based прошивань для бездротових роутерів, розповсюджуваних під ліцензією GNU GPL. До них відносяться так зване «прошивки від Олега», FreeWRT, OpenWRT, X-WRT, DD-WRT і т.д. Як правило, вони підтримують набагато більше функцій, ніж оригінальні прошивки. Необхідні сервіси легко додаються шляхом установки відповідних пакетів. Список підтримуваного устаткування постійно росте.

В ОС сімейства Microsoft Windows підтримка Wi-Fi забезпечується, залежно від версії, або за допомогою драйверів, якість яких залежить від постачальника, або засобами самої Windows:

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

– Ранні версії Windows, такі як Windows 2000 і молодше, не містять убудованих засобів для налаштування й управління, і отут ситуація залежить від постачальника устаткування.

– Microsoft Windows XP підтримує налаштування бездротових пристроїв. І хоча первісна версія включала досить слабку підтримку, вона значно покращилася з виходом Service Pack 2, а з виходом Service Pack 3 була додана підтримка WPA2.

– Microsoft Windows Vista містить поліпшену в порівнянні з Windows XP підтримку Wi-Fi.

– Microsoft Windows 7/8/10/11 підтримує всі сучасні, на момент її виходу, бездротові пристрої й протоколи шифрування. Крім іншого, в Windows 7/8/10/11 створена можливість створювати віртуальні адаптери Wi-Fi, що теоретично дозволило б підключатися не до однієї Wi-Fi-мережі, а до декількох відразу. На практиці в Windows 7/8/10/11 підтримується створення тільки одного віртуального адаптера, за умови написання спеціальних драйверів. Це може бути корисно при використанні комп'ютера в локальній Wi-Fi-мережі й, одночасно, в Wi-Fi-мережі підключеної до Інтернет.

## **2.2 Обґрунтування вибору засобів для побудови системи кібербезпеки та мови програмування**

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент належить й розроблюється Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

## Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

### Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

– Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

– Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

– Тип даних Delphi «record» тепер підтримуватиме довільні ініціалізацію, фіналізацію й операції копіювання.

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCl, libSIMDpp і Nematode.

– Відладник Win 64 (на LLDB) і збирач для C++.

– Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

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

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

- Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

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

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

RAD Studio 10.4 Короткий огляд:

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

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

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

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

### **Істотне поліпшення Delphi Code Insight**

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

### **Delphi Custom Managed Records**

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільнюються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

### **Єдине керування пам'яттю**

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++

використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

### **Розширена підтримка бібліотек C++**

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

### **Win 64-відладник і збирач для C++**

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

### **Підвищення якості й швидкодії інструментів**

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Cmake.
- Велика кількість виправлень для підвищення стабільності і якості.
- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.
- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

## **Змінені стилі VCL для High DPI**

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

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

## **Нові High DPI стилі й стилізація окремих VCL компонент**

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

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізовані компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

## **Поліпшена кроссплатформеність**

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

## Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

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

## Універсальний інсталятор для установки Online і Offline

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

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

## 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи кібербезпеки захищеної бездротової підсистеми керування об'єктами.

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

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

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) сформуванати висновки про виконаний обсяг робіт та одержані результати.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

#### Промисловий Wi-Fi

Промисловий Wi-Fi дозволяє підключити комп'ютер, КПК, ноутбук або інший пристрій, постачений бездротовою мережною картою до локальної мережі Ethernet. Функціонально бездротова мережа складаються з тих же компонентів, що й традиційні локальні провідні Ethernet-мережі. Протоколи передачі даних у цих мережах сумісні із традиційними протоколами Ethernet.

У бездротовій мережі комп'ютери можуть з'єднуватися один з одним безпосередньо, також як і в провідній мережі, а можуть і використовувати для з'єднання комутуючий пристрій.

Залежно від реалізованого в устаткуванні протоколу устаткування може функціонувати або на частоті 2,4 ГГц або на частоті 5 ГГц. Незважаючи на обмежену пропускну здатність у порівнянні із сучасними локальними мережами бездротові мережі, у частотності Wi-Fi, відповідають вимогам ринку надання послуг передачі даних, потребуючих, широкосмугового доступу.

Стандарт бездротових мереж, на якому базується устаткування Wi-Fi IEEE 802.11, описує загальний протокол управління доступом до передавального середовища й кілька фізичних рівнів бездротових локальних мереж.

Промисловий Wi-Fi – це сучасна бездротова технологія передачі даних по радіо каналу базі стандарту IEEE 802.11. Зараз всі продукти, що базуються на стандарті 802.11 такі, як 802.11a, 802.11b або які-небудь інші, відносяться до продуктів Wi-Fi. У цей час сімейство протоколів 802.11, розроблених IEEE і описуючих взаємодію між бездротовим клієнтом і базовою станцією або між двома бездротовими клієнтами, включає наступні протоколи: 802.11, 802.11a, 802.11b, 802.11g, 802.11h, 802.11i, 802.11j, 802.11n.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

Мережа WLAN (Wireless Local Area Network – бездротова локальна мережа) – вид локальної обчислювальної мережі (LAN), що використовує для зв'язку й передачі даних між вузлами високочастотні радіохвилі, а не кабельні з'єднання. Це гнучка система передачі даних, що застосовується як розширення – або альтернатива – кабельної локальної мережі усередині одного офісу, будинку або в межах певної території. Дана технологія дозволяє заощаджувати Ваші засоби за рахунок відсутності необхідності прокладати метри кабелю, а простота установки не забирає час на складні ремонтно-технічні роботи. Розширення й реконфігурація мережі для WLAN не є складним завданням: користувальницькі пристрої можна інтегрувати в мережу, установивши на них бездротові мережні адаптери.

#### Промисловий Wi-Fi:

– Дозволяє розгорнути мережу без прокладки кабелю, що може зменшити вартість розгортання й/або розширення мережі. Місця, де не можна прокласти кабель, наприклад, поза приміщеннями й у будинках, що мають історичну цінність, можуть обслуговуватися бездротовими мережами.

– Дозволяє мати доступ до мережі мобільним пристроям.

– Wi-Fi пристрої широко поширені на ринку. Гарантується сумісність устаткування завдяки обов'язковій сертифікації устаткування з логотипом Wi-Fi.

– Випромінювання від Wi-Fi пристроїв у момент передачі даних на два порядки (в 100 разів) менше, ніж у стільникового телефону.

RFID (Radio Frequency IDentification, радіочастотна ідентифікація) – метод автоматичної ідентифікації об'єктів, у якому за допомогою радіосигналів зчитуються або записуються дані, що зберігаються в так званих транспондерах, або RFID-мітках. Будь-яка RFID-система складається із пристрою, що зчитує (зчитувач або рідер) і транспондера (він же RFID-мітка, іноді також застосовується термін RFID-тег). Більшість RFID-міток складається із двох частин. Перша – інтегральна схема (IC) для зберігання й обробки інформації, модулювання й демодулювання радіочастотного (RF) сигналу й деяких інших функцій. Друга – антена для прийому й передачі сигналу.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

## Класифікація RFID-міток

Існує кілька способів систематизації RFID-міток і систем.

По робочій частоті:

– Мітки діапазону LF (125-134 кГц). Пасивні системи даного діапазону мають низькі ціни, і у зв'язку з фізичними характеристиками, використовуються для підшкірних міток. Однак, у зв'язку з довжиною хвилі, існують проблеми зі зчитуванням на великі відстані, а також проблеми, пов'язані з появою колізій при зчитуванні.

– Мітки діапазону HF (13,56 МГц). Системи 13МГц дешеві, не мають екологічних і ліцензійних проблем, добре стандартизовані, мають широку лінійку рішень. Застосовуються в платіжних системах, логістиці, ідентифікації особи. Для частоти 13,56 МГц розроблений стандарт ISO 14443 (види А/В). Як і для діапазону LF, у системах, побудованих в HF-діапазоні, існують проблеми зі зчитуванням на великі відстані, зчитування в умовах високої вологості, наявності металу, а також проблеми, пов'язані з появою колізій при зчитуванні.

– Мітки діапазону UHF (860-960 МГц). Мітки даного діапазону мають найбільшу дальність реєстрації, у багатьох стандартах даного діапазону присутні антиколізійні механізми. В UHF RFID-системах у порівнянні з LF і HF нижче вартість міток, при цьому вище вартість іншого устаткування.

По джерелу живлення:

– Пасивні. Пасивні RFID-мітки не мають убудованого джерела енергії. Електричний струм, індукований в антені електромагнітним сигналом від зчитувача, забезпечує достатню потужність для функціонування кремнієвого CMOS-чипа, розміщеного в мітці, і передачі відповідного сигналу.

– Активні. Активні RFID-мітки мають власне джерело живлення й не залежать від енергії зчитувача, внаслідок чого вони читаються на далекій відстані, мають більші розміри й можуть бути оснащені додатковою електронікою. Однак, такі мітки найбільш дорогі, а в батареї обмежений час роботи.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

За типом використовуваної пам'яті:

– RO (Read Only) – дані записуються тільки один раз, відразу при виготовленні. Такі мітки придатні тільки для ідентифікації. Ніяку нову інформацію в них записати не можна, і їх практично неможливо підробити.

– WORM (Write Once Read Many) – крім унікального ідентифікатора такі мітки містять блок однократно записуваної пам'яті, що надалі можна багаторазово читати.

– RW (Read and Write) – такі мітки містять ідентифікатор і блок пам'яті для читання/запису інформації. Дані в них можуть бути перезаписані багаторазово.

### **Рідери (Зчитувачі)**

Пристрою, які читають інформацію з міток і записують у них дані. Ці пристрої можуть бути постійно підключеними до облікової системи, або працювати автономно:

– Стационарні. Стационарні зчитувачі кріпляться нерухомо на стінах, дверях, що рухаються, складських пристроях (штабеляторах, навантажувачах). Вони можуть бути виконані у вигляді замка, вмонтовані в стіл або закріплені поруч із конвеєром на шляху проходження виробів. У порівнянні з переносними, зчитувачі такого типу звичайно мають більшу зону читання й потужністю й здатні одночасно обробляти дані з декількох десятків міток. Стационарні зчитувачі підключаються до ПЛК, інтегруються в DCS або підключаються до ПК. Завдання таких зчитувачів – поетапно фіксувати переміщення маркірованих об'єктів у реальному часі, або ідентифікувати положення мічених предметів у просторі

– Мобільні. Мають порівняно меншу дальність дії й найчастіше не мають постійного зв'язку із програмою контролю й обліку. Мобільні зчитувачі мають внутрішню пам'ять, у яку записуються дані із прочитаних міток (потім цю інформацію можна завантажити в комп'ютер) і, як і стационарні зчитувачі, здатні записувати дані в мітку (наприклад, інформацію про зроблений контроль).

Залежно від частотного діапазону мітки, дистанція стійкого зчитування й запису даних у них буде різною.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

## Параметри Wi-Fi-устаткування, дозволеного для використання в Україні

В основу технології лягла методика передачі даних по радіоканалі на частоті 2,4 ГГц із використанням кодування сигналу робочими частотами й спеціальними додатками. Технологія Wi-Fi використовується для організації високошвидкісних бездротових локальних мереж, що працюють у міжнародному неліцензуюемому діапазоні частот (ISM) 2,4 ГГц і 5 ГГц. Области застосування цієї технології пов'язані з мережами для виходу в Інтернет, бездротовою передачею аудио- і відеоінформації, промисловою телеметрією, транспортними локальними бездротовими мережами.

У цей час використовуються наступні стандарти Wi-Fi [2, 3]:

- 802.11 – 1 Мбіт/с і 2 Мбіт/с, 2,4 ГГц;
- 802.11a – 54 Мбіт/с, 5 ГГц;
- 802.11b – 5,5 і 11 Мбіт/с, 2,4 ГГц;
- 802.11g – 54 Мбіт/с, 2,4 ГГц;
- 802.11n – 600 Мбіт/с, 2, 4-2,5 ГГц або 5 ГГц.

Основною перевагою Wi-Fi перед іншими технологіями (Bluetooth, ZigBee) є висока швидкість передачі (до 600 Мбіт/с). Тому ця технологія настільки бурхливо розвивається в таких областях побутової електроніки, як бездротовий доступ в Інтернет, бездротове телебачення, бездротові DVD-програвачі. Широко застосовується Wi-Fi у різних бездротових телеметричних системах на транспорті. Практично всі бездротові відеокамери й реєстратори швидкості, установлені на автомагістралях, використовують Wi-Fi. Також ця технологія використовується для організації локальних мереж між будинками й промисловими об'єктами. Варто підкреслити, що діапазон Wi-Fi 5 ГГц є найбільш кращим для організації промислових локальних мереж при наявності перешкод високого рівня. Завдяки твердій прив'язці до конкретної області, усередині якої поширюється інформація, Wi-Fi є ідеальною технологією для платного виходу в Інтернет у кафе, ресторанах, готелях.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

## Стандарт 802.11

В 1997 р. була прийнята перша специфікація Wi-Fi – 802.11. У стандарті 802.11 регламентується робота устаткування на центральній частоті 2,4 ГГц із максимальною швидкістю до 2 Мбіт/с. У базовому варіанті стандарту 802.11 використовується метод розширення спектра Frequency Hopping Spread Spectrum (FHSS). Опційно може застосовуватися також метод Direct Sequence Spread Spectrum (DSSS).

Для модуляції сигналу використовується технологія Gaussian Frequency Shift Keying. Як правило, коли задіяний метод FHSS, смуга ділиться на 79 каналів по 1 МГц (хоча зустрічається устаткування й з іншим способом розбивки частотного діапазону). Відправник і одержувач погоджують схему перемикання каналів, і дані посилають послідовно по різних каналах з використанням обраної схеми.

Варто особливо підкреслити, що в стандартах 802.11xxx регламентується архітектура мережі й самих пристроїв, описуються основні сім рівнів моделі й протоколи їхньої взаємодії. Стандарт задає базову частоту, а також методи модуляції й розширення спектра на фізичному рівні. Так, наприклад, у стандарті 802.11 задані центральна частота 2,4 ГГц і метод модуляції FHSS PHY. Крім того, первісний варіант стандарту 802.11 описував передачу даних в інфрачервоному діапазоні. Смуга частот і підчастоти для пристроїв стандарту 802.11 виділяються й регламентуються в кожній конкретній країні уповноваженим на те урядовим органом. Також місцевим законодавством регламентуються правила експлуатації самих пристроїв, їхня потужність, розбивка частотного діапазону, потужності передавача й інші характерні риси. У нашій країні таким органом є Міністерство зв'язку й масових комунікацій України. В останньому нормативному документі цього міністерства прописано, що в Україні дозволена експлуатація всіх варіантів стандартів 802.11 (a, b, g, n) на всіх базових частотах [5].

Відмінні риси різних локальних мереж відбиті в поділі каналного рівня (Data Link Layer) на два підрівня: «рівні логічної передачі даних Logical Link

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Control, LLC» і «рівень управління доступом до середовища Media Access Control, MAC» [6]. Рівень MAC забезпечує коректне спільне використання загального середовища. Після одержання доступу до середовища ним може користуватися більше високий рівень LLC, що реалізує функції інтерфейса із прилягаючим до нього мережним рівнем. Протоколи рівнів MAC і LLC взаємно незалежні. Тому кожний протокол рівня MAC може застосовуватися з будь-яким протоколом рівня LLC, і навпаки.

У стандарті 802.11 MAC схожий на рівень, реалізований в 802.3 для Ethernet-мереж. Принципова відмінність полягає в тому, що в 802.11 застосовується напівдуплексний режим приймачепередачі, що не дозволяє виявити колізію під час сеансу зв'язку. Для узгодження MAC-рівнів у стандарті 802.11 використовується спеціальний протокол Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), або Distributed Coordination Function (DCF). У цьому випадку CSMA/CA не допускає колізій, контролюючи підтвердження того, що пакет (ACK) отриманий неушкодженим.

Крім того, MAC-рівень 802.11 підтримує два режими споживання енергії – «режим тривалої роботи» і «зберігаючий». У сплячому режимі устаткування періодично включається через певні проміжки часу для прийому «маячкових» сигналів, які постійно посилає точка доступу. Ці сигнали містять у собі також адресу станції, що повинна прийняти дані. З інших особливостей MAC 802.11 слід зазначити функцію динамічного підключення й перепідключення. Клієнт 802.11 у зоні дії однієї або декількох точок доступу може вибрати ту, сигнал від якої краще. Якщо така точка виявляється, станція автоматично перенастроюється на її частоту.

Для підтримки потокового відео в MAC 802.11 реалізована функція Point Coordination Function (PCF). У режимі PCF тільки точка доступу управляє передачею даних по конкретному каналі. У цьому випадку вона опитує всі станції, і на кожному з них виділяється фіксований проміжок часу. Жодна з інших станцій не може передавати в цей період. Кожна точка доступу має свій

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46



шифрування. В 2004 році було опубліковане виправлення до стандарту 802.11 з новими алгоритмами безпеки WPA і WPA2. Технологія WEP була визнана застарілою. Нові методи забезпечення безпеки WPA і WPA2 (Wi-Fi Protected Access) сумісні між безліччю бездротових пристроїв як на апаратному, так і на програмному рівнях.

Незважаючи на те, що метод FHSS дозволяє застосовувати просту схему приймачепередатчика, він обмежує максимальну швидкість до 2 Мб/с.

### **Стандарт 802.11b**

Обмеження швидкості в стандарті 802.11 привело до того, що пристрої й локальні мережі цього типу практично перестали використовуватися. На зміну 802.11 в 1999 р. прийшов більше швидкий стандарт 802.11b (802.11 High rate), що працює на тій же центральній частоті 2,4 ГГц із максимальною швидкістю до 22 Мбіт/с. У специфікації 802.11b використовується метод розширення спектра Direct Sequence Spread Spectrum (DSSS) – розширення спектра радіосигналу за допомогою застосування прямої послідовності. Основна архітектура, ідеологія, структура й характерні риси рівнів нового стандарту 802.11b аналогічні первісному варіанту Wi-Fi – 802.11, змінився тільки фізичний рівень, що характеризує більше високі швидкості доступу й передачі даних.

Для модуляції й підтримки різних режимів швидкості передачі даних є різні способи. Швидкість 1 Мбіт/с підтримується за рахунок методу DBPSK (Differential Binary Phase Shift Keying). Для забезпечення швидкості 2 Мбіт/с використовується метод DQPSK (Differential Quadrature Phase Shift Keying). Схема модуляції ССК (Complementary Code Keying) допускає швидкості передачі 5,5 і 11 Мбіт/с. Використання ССК-кодів дозволяє кодувати 8 біт на один символ. Символьна швидкість 1,385 мегасимволів у секунду ( $11/8 = 1,385$ ) відповідає швидкості 11 Мбіт/с. При цьому кодується 8 біт на символ. При швидкості передачі 5,5 біт/с в одному символі кодується тільки 4 бітка.

У протоколі також передбачена корекція помилок методом FEC. У розширеному варіанті стандарту 802.11b+ швидкість передачі даних може

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

досягати 22 Мбіт/с. Оскільки метод частотних стрибків FHSS, використовуваний в 802.11, не може підтримувати високі швидкості, він виключений з 802.11b. Тому устаткування для 802.11b сумісно з DSSS-системами 802.11, але не буде працювати із системами FHSS 802.11.

У стандарті 802.11b передбачений режим роботи в умовах сильних перешкод і слабого сигналу. Із цією метою використовується динамічне зрушення швидкості, що дозволяє автоматично змінювати швидкість передачі даних залежно від рівня сигналу й перешкод. Так, наприклад, у тому випадку, коли підвищується рівень перешкод, автоматично знижується швидкість передачі даних до 5,5, 2 або 1 Мбіт/с. При зменшенні перешкод пристрій повертається до нормального режиму роботи на більших швидкостях.

У стандарті 802.11b контроль доступу реалізований як на MAC-рівні, так і за допомогою шифрування даних через WEP. Коли включений WEP, він захищає тільки пакет даних, але не захищає заголовки фізичного рівня, так що інші станції в мережі можуть переглядати дані, необхідні для управління мережею. Необхідно підкреслити, що за останні роки в шифрі RC4 минулого знайдені численні вади. Тому всі частіше стали використовуватися модернізовані протоколи шифрування. Наприклад, стандарт TKIP (Temporal Key Integrity Protocol) використовує той же шифр RC4, що й WEP, але з ініціалізованим вектором довжиною 48 біт. Для перевірки цілісності повідомлень доданий протокол MIC (Message Integrity Check). При його використанні станція блокується, якщо протягом мінути буде послано більше двох не минулу перевірку запитів. У протоколі AES-CCMP розподіл ключів і перевірка цілісності виконана в одному компоненті CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol). Для шифрування використовується шифр AES.

З розвитком технологій LAN в усьому світі різко зросла кількість різних бездротових пристроїв, і виникла проблема перешкод і перевантаженості діапазону 2,4 ГГц. Це пов'язане з тим, що такі пристрої, як мікрохвильові печі, бездротові телефони, рації, Bluetooth-устаткування й інші аналогічні прилади

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

помітно впливають один на одного. Зокрема, це позначається й на якості роботи устаткування Wi-Fi.

Як було відзначено вище, у стандарті 802.11 максимальна швидкість передачі визначається як сума по каналах. Тому теоретична швидкість не однозначно відповідає реальній швидкості передачі даних. У випадках, коли різні пристрої 802.11 використовують ті самі канали або працюють у зоні потужних радіоперешкод, можуть виникати істотні зниження швидкості. Наприклад, бездротова станція, що встановила з'єднання на швидкості 11 Мбіт/с, реально буде працювати зі швидкістю не більше 1 Мбіт/с, якщо вона перебуває в зоні дії потужної мікрохвильової печі.

### **Стандарт 802.11a**

Щоб якось розвантажити діапазон 2,4 ГГц, був розроблений стандарт 802.11a для частот 5 ГГц. У цьому діапазоні не так багато джерел перешкод, як у діапазоні 2,4 ГГц, і середній рівень сукупних шумів значно нижче. У стандарті 802.11a використовуються дві базові центральні частоти в районі 5 ГГц і максимальна швидкість передачі даних становить до 54 Мбіт/с. У цьому стандарті як спосіб доступу до середовища застосований множинний метод з контролем несучої й запобіганням колізій. Як основний метод розширення спектра прийняте Orthogonal Frequency Division Multiplexing (OFDM) – мультиплексування с ортогональним частотним поділом сигналів [8]. Для стандарту 802.11a в Україні виділені дві частотні смуги.

Відповідно до документа [5] на території України для стандарту 802.11a частотні смуги підрозділяються на п'ять робочих піддіапазонів. Діапазони 5, 150-5,250 і 5, 250-5,350 ГГц призначені для роботи устаткування з потужністю передавача до 100 мВт (20 дБм). Діапазони 5, 650-5,725; 5, 725-5,825 і 5, 825-6,425 ГГц відведені для устаткування з потужністю передавача до 1000 мВт (30 дБм).

У стандарті 802.11a в якості основного використовується метод, розроблений фірмою Intersil і отримавший назву Orthogonal Frequency Division Multiplexing (OFDM) – мультиплексування с ортогональним частотним поділом сигналів.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Весь діапазон частот розбивається на що піднесуть, які хоча частково й перекриваються, але перебувають в ортогональному положенні відносно один одного. Ортогональність несучих сигналів забезпечується в тому випадку, коли за час тривалості одного символу несучий сигнал буде робити ціле число коливань. Для реалізації методу в передавальних пристроях використовується зворотне швидке перетворення Фур'є (IFFT), що переводить попередньо мультиплексований на одному з каналів сигнал з часового подання в частотне. Таким чином, там, де одна несуча має максимум амплітуди, сусідня несуча має нульове значення. Інформація в даному методі передається у вигляді так званих OFDM-символів.

Перед символом постійно заноситься префікс. Для захисту від виникнення міжсимвольних колізій у технології OFDM вводиться поняття охоронного інтервалу (Guard Interval, GI), протягом якого буде йти циклічне повторення OFDM. Префікс додається до переданого символу в передавачі й віддаляється при прийманні символу в приймачі. Охоронний інтервал знижує швидкість передачі даних.

У стандарті 802.11a діапазон розбивається із частотним розносом каналів 20 МГц. При цьому в кожному з каналів є 52 піднесучі частоти. З них 48 використовуються для передачі даних, а інші чотири – для кодів корекції помилок. Рознос частот, що піднесуть, становить 312,5 кГц. Ширина сигнальної смуги – 16,66 МГц. Швидкості згортчного кодування: 1/2, 9/16, 2/3, 3/4. У протоколі IEEE 802.11a максимальна швидкість згортчного кодування становить 3/4, коли до кожним трьох вхідним бітам додається ще один. На різних рівнях використовуються різні схеми модуляції. На самому нижньому застосовується бінарна фазова модуляція (Binary Phase Shift Keying, BPSK). Вона забезпечує пропускну здатність підканалу 125 кбіт/с. Тому для основного каналу пропускну здатність становить 6 Мбіт/с (48 помножити на 125). На наступному рівні використовується квадратурна фазова модуляція (QPSK), що дозволяє подвоїти пропускну здатність до 12 Мбіт/с.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

У випадку, коли на фізичному рівні задіяна 16-рівнева квадратурна амплітудна модуляція (16QAM), що кодує 4 біт на один Герц несучої частоти, пропускна здатність каналу складе 24 Мбіт/с. При використанні 64-рівневої квадратурної амплітудної модуляції (64QAM), що кодує 8 або 10 біт на один Герц несучої частоти, забезпечується максимальна для цього стандарту швидкість – 54 Мбіт/с.

Таким чином, у стандарті 802.11a підтримуються швидкості передачі даних: 6, 12, 24, 36, 48 і 54 Мбіт/с. Однак сам стандарт допускає також реалізацію й більше високих швидкостей. Так, наприклад, фірма Atheros випускає устаткування 802.11a з одночасним використанням двох несучих частот, за рахунок чого максимальна пропускна здатність може досягати значення 108 Мбіт/с.

Необхідно звернути увагу, що діапазон 5 ГГц примикає до частот, які частково використовуються наземними станціями спостереження за супутниками зв'язку. Для того щоб неліцензуєме Wi-Fi-обладнання не заважало роботі інших відомчих систем, Європейським інститутом по стандартизації в області телекомунікацій (European Telecommunications Standards Institute, ETSI) були розроблені два додаткових протоколи: DFS (Dynamic Frequency Selection) і TPC (Transmit Power Control). З їхньою допомогою бездротові пристрої Wi-Fi можуть автоматично міняти частотні канали або знижувати випромінювану потужність у випадках виникнення колізій на несучих частотах.

### **Стандарт 802.11g**

Наступним кроком на шляху розвитку пристроїв Wi-Fi був стандарт 802.11g, прийнятий в 2003 р. Практично 802.11g – це вдосконалений варіант 802.11b. Він призначений для пристроїв, що працюють на частотах 2,4 ГГц із максимальною швидкістю 54 Мбіт/с. Цей стандарт замислювався як універсальний. Тому в ньому допускаються методи розширення спектра, що використовуються в попередніх версіях, а саме DSSS, OFDM, PBCC [9].

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

Виділена для 802.11g смуга частот в Україні становить 2400-2483,5 МГц. Стандарт 802.11g повністю сполучимий з 802.11b. Основна відмінність полягає в припустимих методах доступу до середовища й способах модуляції. У стандарті 802.11g використовуються розглянуті вище технології DSSS, PBCC, які взяті з 802.11b. Метод OFDM прийнятий зі стандарту 802.11a. Методи модуляції DBPSK, DBPSK, CCK, CCK, PBCC також узяті з 802.11a, b.

Не вдаючись особливо в подробиці, можна сказати, що стандарт 802.11g аналогічний стандарту 802.11b по частоті 2,4 ГГц і схожий на стандарт 802.11a по максимальній швидкості передачі 54 Мбіт/с [10].

### **Стандарт 802.11n**

Останнім із прийнятих для технології Wi-Fi став стандарт 802.11n, у якому розроблювачі почали спробу об'єднати все краще, що було реалізовано в попередніх версіях. Стандарт 802.11n розроблений для устаткування, що функціонує на центральних частотах 2,4 і 5 ГГц із максимально можливою швидкістю аж до 600 Мбіт/с [11]. Цей стандарт був затверджений IEEE у вересні 2009-го, а в Україні схвалений і дозволений до використання у всіх діапазонах тільки наприкінці 2010 р. [5]. Стандарт заснований на технології OFDM-MIMO. В IEEE 802.11n максимальна швидкість передачі даних у кілька разів більше, ніж у попереднім. Це досягається завдяки подвоєнню ширини каналу з 20 до 40 МГц, а також за рахунок реалізації технології MIMO з безліччю антен.

В ідеальному випадку подвоєння ширини смуги означає прямо пропорційне збільшення швидкості передачі даних на фізичному рівні (PHY). На практиці все виявляється набагато складніше. В основу технології MIMO (Multiple Input Multiple Output) покладена ідея застосування роздільно декількох передавальних і прийомних антен [12]. Переданий потік даних розбивається на незалежні послідовності бітів, які пересилаються одночасно, з використанням різних антен. При цьому антени передають дані незалежно друг від друга й у тому самому частотному діапазоні. Іншими словами, у технології MIMO реалізовано декілька просторово рознесених підканалів, по яких дані

передаються одночасно в тому самому частотному діапазоні. У найпростішому прикладі це виглядає як передавач із двома антенами й приймач із двома антенами, у яких по кожному каналу одночасно й незалежно передаються й приймаються потоки даних.

Технологія MIMO не впливає на метод кодування даних і може використовуватися з різними способами модуляції. У стандарті 802.11n як метод розширення спектра використовується Orthogonal Frequency Division Multiplexing (OFDM), що добре зарекомендував себе в стандарті 802.11a. Технології MIMO містять у собі складні векторні й матричні алгоритми обробки в системах з безліччю антен (multi-antenna).

Метод кодування OFDM по своїй структурі в цей час є оптимальним для підтримки технології MIMO. В MIMO використовується методика попереднього кодування й наступного декодування (Precoding) з формуванням просторової діаграми спрямованості (beamforming), що являє собою якість векторне розширення стандартної плоскої діаграми спрямованості. При формуванні просторової діаграми спрямованості використовується безліч антен для передачі сигналів. Такий підхід дозволяє значно поліпшити охопит і ємність системи, а також зменшити ймовірність порушення зв'язку. Щоб забезпечити просторове рознесення й оптимальний запас часу на завмирання, у методі MIMO використовуються коди «простір-час» (Space-Time Code, STC).

Методика MIMO містить у собі так зване «просторове мультиплексування» (Spatial Multiplexing, SM), що підвищує швидкості передачі й збільшує пропускну здатність у порівнянні з окремою одиночною антеною. При просторовому мультиплексуванні безліч потоків передаються по безлічі антен. Наприклад, якщо приймач і передавач мають по дві антени і є можливість виділити із усього різноманіття електромагнітного випромінювання необхідні хвилі, то можна збільшити пікову швидкість даних удвічі.

Процес передачі даних іде незалежно. Це значить, що в напрямку «нагору» (UL) кожний користувач має тільки одну передавальну антену. Два

незалежних користувачі можуть одночасно передавати в тому же самому слоті, аналогічно тому случаю, коли два потоки просторово мультиплексовані від двох антен одного користувача. Такий процес називається «спільне просторове мультиплексування нагору». Коли повідомлення відправляється від базової станції до мобільного, то говорять про напрямок «долілиць».

У процесі передачі послідовність символів, що надходить на кодер, перетворюється символьним перетворювачем у просторову форму відповідно до програми, закладеної в адаптивному перетворювачі (наприклад, відбиття інформації підканалів у просторовий код відповідно до заданої матриці).

У методі MIMO необхідно постійно запитувати інформацію з ідентифікації каналу, його стану й конкретним параметрам. Залежно від поточного стану каналу сигнали передаються по різним підканалам. Спеціальні сигнали використовуються для перетворення параметрів самих підканалів, таких, наприклад, як діаграма спрямованості елементів адаптивної антени, корекція помилок, швидкість передачі й ін. Для корекції помилок використовується коефіцієнт помилок пакетів (Packet Error Rate, PER). Коли канал перебуває в поганому стані, збільшується значення цього коефіцієнта й, як наслідок, автоматично зона покриття обмежується до величини, де може бути витримане розрахункове значення PER. Варто мати на увазі, що SM і STC забезпечують великий охопити незалежно від стану каналу, але не підвищують пікову швидкість даних.

При декодуванні в прийомному пристрої отримані сигнали обробляються за певним законом відповідно до заданої матриці, наприклад за допомогою алгоритму зворотного перетворення Фур'є. Таким чином, у приймачі просторово розподілені сигнали поєднуються, і відбувається відновлення переданих даних.

Для стандарту 802.11n в Україні виділена одна смуга із центральною частотою 2,4 ГГц і дві смуги в районі 5 ГГц:

- 2400-2483,5 МГц;
- 5150-5350 МГц;
- 5650-6425 МГц.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Кількість піднесучих у каналі визначено рівним 56 при ширині каналу 20 МГц і 114 – при ширині каналу 40 МГц. Частотний рознос каналів дозволений як для 20, так і для 40 МГц. У стандарті 802.11n відповідно до нормативів України допускається використання до чотирьох каналів передачі даних. Мається на увазі, що не менш двох каналів можуть бути в Wi-Fi-точки доступу й не менш одного каналу повинне бути в бездротової абонентської станції. Устаткування Wi-Fi у стандарті 802.11n може працювати в трьох режимах:

- режим попередніх версій (Legacy), у якому забезпечується підтримка всіх попередніх версій стандарту 802.11a, b, g (немає підтримки 802.11n);
- змішаний режим (Mixed), у якому забезпечується підтримка всіх попередніх версій стандарту 802.11a, b, g і часткова підтримка 802.11n;
- високошвидкісний режим (High Throughput, HT), у якому забезпечується тільки повна підтримка 802.11n і не підтримуються повністю всі попередні версії.

Варто підкреслити, що тільки в режимі High Throughput можна повною мірою користуватися перевагами підвищеної швидкості й збільшеною дальністю передачі даних, досягнутими в стандарті 802.11n. У режимі з високою пропускною здатністю (High Throughput) при ширині каналу в 20 МГц застосовуються 56 частотних підканалів, з яких 52 задіюються для передачі даних, а чотири є службовими. При використанні 40-мгц каналу й режиму з високою пропускною здатністю застосовуються 114 частотних підканалів, з яких 108 – інформаційні, а шість – керуючі.

Ще один параметр, що впливає на швидкість передачі, – це тривалість охоронного інтервалу GI, уведений в стандарті 802.11a. У стандарті 802.11 тривалість охоронного інтервалу може приймати два значення: 800 і 400 нс. Швидкості передачі даних визначаються комбінацією розглянутих вище параметрів. Усього таких комбінацій у стандарті 802.11n може бути 76.

#### **Додаткові стандарти IEEE 802.11**

Крім розглянутих вище основних стандартів 802.11a, b, g, n, існує ряд допоміжних, що описують сервісні функції різних Wi-Fi-виробів:

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>56</b>

– 802.11d. Призначений для адаптації різних Wi-Fi-пристроїв до специфічних умов країни. Як уже згадувалося вище, конкретні діапазони частот для кожного окремо взятої держави визначаються усередині самої країни й можуть розрізнятися залежно від географічного положення. Стандарт IEEE 802.11d дозволяє регулювати смуги частот у пристроях різних виробників за допомогою спеціальних опцій, уведених у протоколи управління доступом до середовища передачі.

– 802.11e. Описує класи якості QoS для додатків, що забезпечують передачу аудіо- і відеофайлів. Зміни, уведені на рівні MAC-протоколів 802.11e, регламентують якість одночасної передачі звуку й зображення для бездротових аудіо- і відеосистем.

– 802.11f. Уніфікує параметри Wi-Fi-точок доступу різних виробників. Стандарт дозволяє користувачеві працювати з різними мережами при переміщенні між зонами дії окремих мереж.

– 802.11h. Як відзначалося вище, у більшості європейських країн наземні станції спостереження за метеорологічними супутниками й супутниками зв'язку, а також радары військового призначення працюють у діапазонах, близьких до 5 МГц. Для запобігання конфліктних ситуацій стандарт 802.11h вводить обов'язковий для використання в Європі механізм автоматичного скидання потужності на частотах 5 ГГц для побутових пристроїв Wi-Fi при влученні їх у зону дії виробів 802.11 спеціального й військового призначення. Цей стандарт є необхідною вимогою ETSI, пропонованим до устаткування, допущеному для експлуатації на території країн Європейського Союзу. Так, наприклад, всі Wi-Fi-устаткування, що випускається французькою фірмою ACKSYS, проходить обов'язкову європейську сертифікацію на відповідність стандарту 802.11h.

– 802.11i. У перших варіантах стандартів 802.11 для забезпечення безпеки мереж Wi-Fi використовувався алгоритм WEP. Передбачалося, що цей метод може забезпечити конфіденційність і захист переданих даних авторизованих користувачів бездротової мережі від прослуховування. Однак, як з'ясувалося, цей

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

захист можна зламати всього за кілька мінут. Тому в стандарті 802.11i були розроблені нові методи захисту мереж Wi-Fi, реалізовані як на фізичному, так і програмному рівнях. У цей час для організації системи безпеки в мережах 802.11 рекомендується використовувати алгоритми Wi-Fi Protected Access (WPA). Вони також забезпечують сумісність між бездротовими пристроями різних стандартів і різних модифікацій. Протоколи WPA використовують удосконалену схему шифрування RC4 і метод обов'язкової автентифікації з використанням EAP. Стійкість і безпека сучасних мереж Wi-Fi визначається протоколами перевірки конфіденційності й шифрування даних (RSNA, TKIP, CCMP, AES).

– 802.11k. Цей стандарт був розроблений, щоб поліпшити розподіл трафіку між абонентами усередині мережі. У бездротовій локальній мережі абонентський пристрій звичайно з'єднується з тією точкою доступу, що забезпечує найбільш сильний сигнал. Це може привести до перевантаження мережі, якщо до однієї точки доступу будуть прагнути підключитися відразу багато абонентів. Для контролю подібних ситуацій у стандарті 802.11k запропонований механізм, що обмежує кількість абонентів, що підключаються до однієї точки доступу, і підєднуючий нових абонентів до іншої точки, незважаючи на більше слабкий сигнал від неї. У цьому випадку повна пропускна здатність мережі збільшується завдяки більше ефективному використанню ресурсів.

– 802.11m. У рамках IEEE 802.11 існує робоча група TASK GROUP, що займається виправленням помилок і відповідями на запити й зауваження, які будь-яка людина може відправити в IEEE. Ці виправлення й виправлення підсумуються в окремому документі із загальною назвою 802.11m. Перший випуск 802.11m був в 2007 р. Наступний випуск виправлень, доповнень і виправлень до всіх редакцій 802.11 планується на 2013 р.

– 802.11p. Регулює взаємодію Wi-Fi-устаткування, що рухається зі швидкістю до 200 км/год повз нерухливі точки доступу, віддалені на відстань до 1 км. Він входить до складу стандарту Wireless Access in Vehicular Environ (WAVE) і є свого роду інтерфейсом для зв'язку з IEEE 1609. Стандарти WAVE

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

визначають архітектуру й додатковий набір службових функцій і інтерфейсів, які забезпечують безпечний механізм радіозв'язку між транспортними засобами, що рухаються. Ці стандарти розроблені для таких додатків, як, наприклад, організація дорожнього руху, контроль безпеки руху, автоматизований збір платежів, навігація й маршрутизація транспортних засобів і ін.

– 802.11r. Регламентує швидкий автоматичний роумінг Wi-Fi-пристроїв при переході із зони дії однієї точки доступу до зони охопту іншої. Цей стандарт орієнтований в основному на інтернет-телефонію й на мобільні телефони з підтримкою Wi-Fi. До появи цього стандарту при русі абонент часто втрачав зв'язок з однією точкою доступу, був змушений шукати іншу й заново виконувати процедуру підключення. Пристроєм з підтримкою 802.11r можуть зареєструватися заздалегідь із сусідніми точками доступу й виконувати процес перепідключення в автоматичному режимі. У такий спосіб значно зменшується мертвий час, коли абонент не доступний у мережах Wi-Fi.

– 802.11s. Розроблений для топології багатовузлових або коміркових мереж (Wireless Mesh Network), де будь-який пристрій може служити як маршрутизатором, так і точкою доступу. Якщо найближча точка доступу перевантажена, дані перенаправляються до найближчого незавантаженого вузла. При цьому пакет даних передається від одного вузла до іншого, поки не досягне кінцевого місця призначення. У даному стандарті уведені нові протоколи на рівнях MAC і PHY, які підтримують ширококомовну й багатоадресну передачу, а також одноадресну поставку по самоконфігуруючій системі точок доступу Wi-Fi. З цією метою в стандарті уведений чотирьохадресний формат кадру. Проект одержав внутрішню назву SEE-MESH і в цей час перебуває в стадії розробки (в основному роботи із цього проекту веде німецька компанія Riedel Communications).

– 802.11t. Цей документ являє собою набір методик, рекомендованих IEEE для тестування мереж 802.11: способи вимірів і обробки результатів, вимоги, пропоновані до іспитового устаткування.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

– 802.11u. Призначений для регулювання взаємодії мереж Wi-Fi із зовнішніми мережами. Стандарт повинен визначати протоколи доступу, протоколи пріоритету й заборони на роботу із зовнішніми мережами. У цей час стандарт перебуває на етапах оцінки й твердження проекту.

– 802.11v. У стандарті повинні бути розроблені виправлення, спрямовані на вдосконалювання систем управління мережами IEEE 802.11. Модернізація на MAC- і PHY-рівнях повинна дозволити централізувати й упорядкувати конфігурацію клієнтських пристроїв, з'єднаних з мережею. Перебуває в стадії розробки.

– 802.11y. Додатковий стандарт зв'язку для діапазону частот 3, 65-3,70 ГГц. Призначений для пристроїв останнього покоління, що працюють із зовнішніми антенами на швидкостях до 54 Мбіт/с на відстані до 5 км на відкритому просторі. Стандарт повністю не завершений.

– 802.11w. Розроблений з метою поліпшення захисту й безпеки рівня управління доступом до середовища передачі даних (MAC). Протоколи стандарту структурують систему контролю цілісності даних, дійсності їхнього джерела, заборони несанкціонованого відтворення й копіювання, конфіденційності даних і інших засобів захисту. У стандарті введений захист фрейму управління, а додаткові заходи безпеки дозволяють нейтралізувати зовнішні атаки, такі, як, наприклад, DoS. Крім того, ці міри забезпечать безпека для найбільш уразливої мережної інформації, що буде передаватися по мережах з підтримкою IEEE 802.11r, k, y. У цей час стандарт ще не завершений.

На закінчення необхідно відзначити, що технологія Wi-Fi є одним з напрямків бездротового зв'язку, що найбільше бурхливо розвиваються. У цей час устаткування для Wi-Fi випускають багато компаній. При цьому найчастіше провідні фірми – виробники Wi-Fi-чипсетів виходять за рамки прийнятих стандартів IEEE і випускають на ринок власні розробки, не схвалені Wi-Fi Alliance. Як приклад можна привести технологію Super G, розроблену фірмою Atheros для збільшення ефективної пропускної здатності. В основу технології покладений так званий метод «зв'язування каналів»: два радіоканали зв'язуються

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

таким чином, щоб вони здавалися одним каналом як для передавача, так і для приймача. Теоретично це дозволяє збільшити швидкість передачі даних у стандарті 802.11g у два рази й довести неї до 108 Мбіт/с.

Крім того, теоретично повинен збільшуватися радіус дії мережі. Однак за іншим даними ефект зв'язування каналів сильно залежить від відстані й зменшується з його збільшенням [15].

Як інші приклади «несанкціонованого» виходу за рамки стандартів IEEE можна привести технології 25 High Speed Mode від Broadcom, розроблене Airgo Networks «MIMO-розширення» і Nitro, пропоновану Conexant. Навіть така солідна фірма, як Texas Instruments, і та вийшла за рамки стандартів IEEE, запропонувавши технологію 802.11b+.

Багато учасників Wi-Fi-альянсу затверджують, що устаткування з підтримкою Super G і інших неузгоджених технологій заважає нормальній роботі в частотному діапазоні 2,4 ГГц. Однак, як справедливо відзначається в [15], існує безліч виробів, наприклад підсилювачів потужності й активних антен, які можуть заважати сусіднім бездротовим мережам і не мають ніяких механізмів регламентації в зоні дії іншого Wi-Fi-устаткування.

З появою в 2009 р. стандарту 802.11n, що вбрали в себе все найкраще з попередніх версій 802.11, розжарення суперечки про те, який стандарт краще, повинен був би ослабнути. Безумовно, стандарт 802.11n зараз найшвидший. Але оскільки у світі виробляється й ще якийсь час буде вироблятися устаткування, що підтримує стандарти 802.11a, b, g і Super G, те питання, «що вибрати з 802.11», залишається відкритим. Щоб знайти відповідь на нього, потрібно чітко розуміти, для яких цілей призначається конкретна Wi-Fi-мережа.

Наприклад, для передачі більших обсягів інформації на невеликі відстані швидкість є визначальним чинником. Однак боротьба за швидкість передачі не завжди є оправданою. Насправді завдання повинна формулюватися як підтримка оптимальної швидкості на необхідній відстані. Не можна забувати й про перевантаженість конкретного обсягу бездротовим устаткуванням. Відомо, що Wi-Fi-пристрою починають конфліктувати, коли працюють у безпосередній

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

близькості друг до друга. У закритих приміщеннях також існує проблема відбиття від стін і масивного устаткування. Варто також подумати й про вибір частоти. У частотному діапазоні 2,4 ГГц дальність дії більше. Однак перевантаженість цього діапазону й наявність перешкод набагато більше, ніж у діапазоні 5 МГц. Найкращим варіантом може бути вибір двох приватних діапазонів і змінна робота в одному з них залежно від стану середовища передачі.

### 3.2 Розробка структурної схеми

Бездротове керування об'єктами, на сучасному етапі, як правило, застосовується в автоматизованих системах управління технологічними процесами (АСУ ТП).

Автоматизація промисловості сьогодні розуміється як комплексний підхід для рішення наступних завдань виробництва:

- автоматизація технологічних процесів;
- інформаційне забезпечення виробничих процесів (автоматизовані системи диспетчерського контролю) АСДК;
- автоматизація окремих ділянок і підсистем;

Автоматизована система управління технологічними процесами (АСУ ТП) – це комплекс програмних і апаратних засобів, що функціонує під управлінням людини-оператора.

Автоматизована система управління складається із частин, представлених автоматичними підсистемами, зв'язаних між собою в єдине ціле й виконуюча контроль (вимір) і регулювання параметрів станів технологічних процесів відповідно до необхідних правил і алгоритмами.

У даній бакалаврській роботі розглянуті створення й впровадження високотехнологічних систем АСУ ТП на підприємствах і виробництвах, починаючи з автоматизації керування окремими процесами, аж до створення систем комплексної автоматизації.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

## АСУ ТП

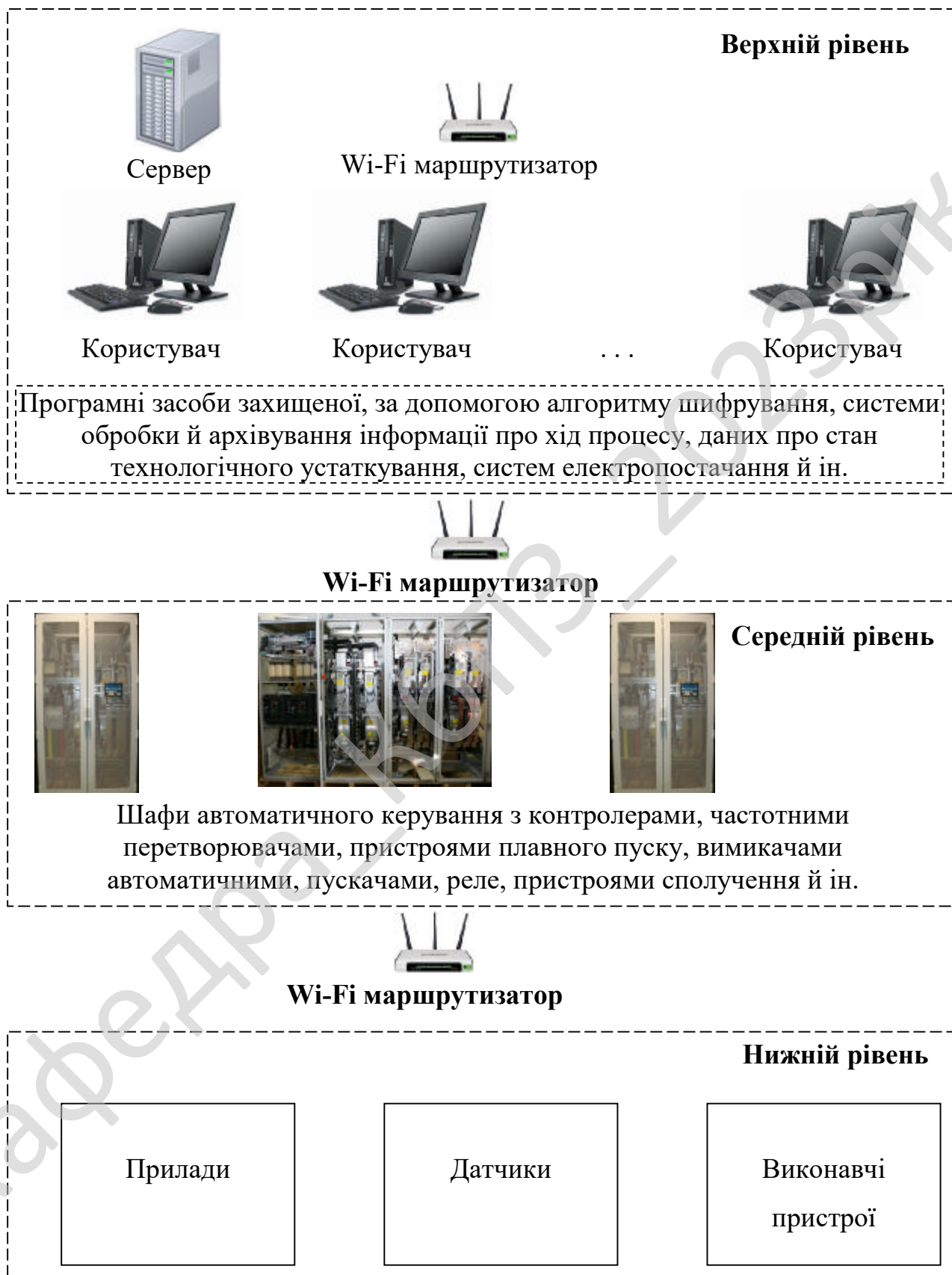


Рисунок 3.1 – Структурна схема системи

Структурна схема розробленої системи зображена на рисунку 3.1. На ній показано структуру автоматизованої системи управління технологічним процесом з використанням бездротової системи керування об'єктами.

Як правило, АСУ ТП має єдину систему операторського керування технологічним процесом, що складається:

- на верхньому рівні – із серверів, одного або декількох АРМ, програмних засобів обробки й архівування інформації про хід процесу, даних про стан технологічного устаткування, систем електропостачання й ін.;

- на середньому рівні – шаф автоматичного керування з контролерами, частотними перетворювачами, пристроями плавного пуску, вимикачами автоматичними, пускачами, реле, пристроями сполучення й ін.;

- на нижньому рівні – приладів, датчиків, виконавчих пристроїв.

Розглянута, під час виконання бакалаврської роботи, АСУ ТП забезпечує:

- цілодобовий режим роботи;
- регулювання технологічного процесу, автоматична підтримка технологічних параметрів;
- контроль виконання команд керування з робочої станції;
- підтримка зв'язку з диспетчерським пунктом об'єкта керування;
- вивід заданих і поточних параметрів на робочу станцію верхнього рівня.
- одержання й обробку параметрів технологічних процесів;
- формування візуальних і звукових передаварійних і аварійних сигналів попередження, вивід на екран панелі оператора інформації про аварійні ситуації;
- контроль дій операторів.

Для інформаційного зв'язку всіх підсистем використовуються промислові мережі.

АСУ ТП може бути реалізована з використанням наступних інформаційних каналів зв'язку між контролером і комп'ютером диспетчера:

- провідний канал (Profibus, Ethernet і ін.);

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

- бездротовий канал стандарту Wi-Fi;
- бездротовий канал, заснований на сервісі передачі даних через мережу GPRS; надаваний операторами стільникового зв'язка;
- бездротовий канал з використанням радіомодемів.

Кінцевою метою автоматизації будь-якого об'єкта є розробка й впровадження АСУ ТП, що дозволяє підтримувати заданий технологічний режим.

Впровадження АСУ ТП дозволяє:

- здійснювати дистанційне керування об'єктом;
- здійснювати дистанційна зміна налаштувань;
- контролювати з'єднання елементів системи;
- оперативно реагувати на збої й несправності системи;
- проводити аналіз роботи об'єкта;
- здійснювати прогнозування необхідних витрат;
- передавати необхідну інформацію в АСУП (підприємства);
- при необхідності – сповіщати користувача за допомогою SMS-

Повідомлень про виникнення аварійної ситуації;

- проводити архівування параметрів і повідомлень про хід технологічного процесу;
- контролювати дії операторів.

Автоматизовані системи диспетчерського контролю (АСДК), як один з видів АСУ ТП, забезпечують контроль над об'єктами, де не передбачається втручання людини.

Часто такі завдання коштують при автоматизації великої кількості вилучених друг від друга й від диспетчерського пункту об'єктів, з'єднаних у єдину систему або при контролі систем автоматичного керування, де процеси йдуть по жорстко певному алгоритмі.

Процес створення конкретних АСУ ТП починається зі складання технічного завдання, де визначаються:

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

- призначення й мети створення системи;
- характеристики об'єкта автоматизації;
- вимоги до системи;
- склад і зміст робіт зі створення АСУ ТП;
- вимоги до документування й ін.

При створенні системи визначається структурна схема АСУ ТП, проводиться вибір технічних засобів: сервера(ів), АРМа (ів), контролера (ів), а також програмного забезпечення (SCADA) WinCC, Intouch, iFix і ін.

Для рішення завдання зв'язку вилучених об'єктів виробляється вибір каналів зв'язку й відповідної апаратури:

- проводований зв'язок (Ethernet, Profibus, Modbus);
- бездротовий канал стандарту Wi-Fi (IEEE 802.11) (невеликі відстані, радіус дії (45 м – 500 м);
- радіоканал (відстані 3-5 км у зоні прямої видимості без використання ретрансляційних пристроїв);
- бездротовий канал GPRS (при знаходженні об'єктів у зоні обслуговування операторів).

Різні стандарти сімейства IEEE 802 строго регламентують два нижніх рівні моделі OSI – фізичний і канальний, які характеризують особливості конкретних локальних мереж. Верхні рівні збігаються по своїй структурі як для бездротових, так і для провідних локальних мереж. Як і всі стандарти цього сімейства, Wi-Fi 802.11 працює на нижніх двох рівнях моделі ISO/OSI, фізичному й канальному [6].

Тому мережні додатки й мережні протоколи, які працюють у мережі Ethernet (стандарт 802.3), такі, наприклад, як TCP/IP, можуть аналогічним образом використовуватися й в Wi-Fi-мережах 802.11. Іншими словами, якщо є якийсь Ethernet-роутер з декількома входами, то для мережі байдуже, чи буде до нього підключене провідний пристрій стандарту 802.3 або бездротовий Wi-Fi-

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

пристрій стандарту 802.11: всі периферійні пристрої будуть бачити один одного й правильно взаємодіяти.

### 3.3 Розробка функціональної схеми

АСУ ТП призначена для реалізації функцій оперативного контролю, обліку, аналізу й керування технологічними процесами. Вона забезпечує автоматизований і автоматичний режими роботи устаткування, поліпшує інформаційне забезпечення оперативного й керівного персоналу.

АСУ ТП функціонально реалізована у вигляді дворівневої розподіленої системи з наступним групуванням виконуваних функцій:

- відображення й архівування інформації – верхній рівень;
- збір/обробка інформації й керування технологічним процесом, система протиаварійного захисту – нижній рівень.

Функціональна схема розробленої системи зображена на рисунку 3.2.

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

Верхній рівень:

- Функціональний блок відображення на екрані робочої станції інформації про хід технологічного процесу, значеннях параметрів і стані устаткування.
- Функціональний блок надання по кожному контуру регулювання повної інформації, що включають значення параметра, завдання й керуючого впливу, шкалу приладу, одиниці виміру, аварійні й передаварійні границі.
- Функціональний блок виведення інформації про виконання команди керування електроустаткуванням: насосами й компресорами, електрозасувками й відсекателями.
- Функціональний блок виведення інформації про параметри й керуючі впливи у вигляді трендів і графіків з можливістю друку й зміни масштабу.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

## АСУ ТП

### Верхній рівень

Блок відображення на екрані робочої станції інформації про хід технологічного процесу, значеннях параметрів і стані устаткування

Блок виведення інформації про параметри й керуючі впливи у вигляді трендів і графіків

Блок виведення інформації про виконання команди керування електроустаткуванням

Блок надання по кожному контуру регулювання повної інформації

Блок архівації аналогових і дискретних технологічних параметрів, завдань і величин керуючих впливів

Блок формування змінних рапортів оператора й балансових звітів з виведенням на друк

### Нижній рівень

Блок первинного збору й контролю технологічних параметрів

Блок зміни режиму роботи й завдання контурам регулювання

Блок автоматичного регулювання технологічних параметрів

Блок реєстрації й оповіщення про відхилення технологічних параметрів

Блок протиаварійного захисту устаткування

Блок реєстрації повідомлень, формованих програмами керування й системою протиаварійного захисту

Блок фіксації дій оператора при роботі із системою

Блок фіксації й оповіщення про порушення зв'язку з контролерами й обриві вимірювального каналу

Блок діагностування системи на всіх рівнях

Блок зміни налаштувань контурів регулювання

Блок аналітичного контролю показників виробництва

Блок ведення таблиць лабораторних аналізів технологічних потоків

Блок дистанційного керування аналоговими й позиційними виконавчими механізмами: клапанами, засувками, відсекателями, насосами й компресорами



### Wi-Fi маршрутизатор

### Об'єкти управління

Рисунок 3.2 – Функціональна схема системи

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>68</b>

- Функціональний блок архівації аналогових і дискретних технологічних параметрів, завдань і величин керуючих впливів.
- Функціональний блок формування змінних рапортів оператора й балансових звітів з виведенням на друк.

Нижній рівень:

- Функціональний блок первинного збору й контролю технологічних параметрів.
- Функціональний блок зміни режиму роботи й завдання контурам регулювання.
- Функціональний блок автоматичного регулювання технологічних параметрів відповідно до регламентних вимог.
- Функціональний блок дистанційного керування аналоговими й позиційними виконавчими механізмами: клапанами, засувками, відсекателями, насосами й компресорами.
- Функціональний блок реєстрації й оповіщення про відхилення технологічних параметрів за передаварійні й аварійні границі.
- Функціональний блок протиаварійного захисту устаткування.
- Функціональний блок фіксації дій оператора при роботі із системою.
- Функціональний блок реєстрації повідомлень, формованих програмами керування й системою противоаварійного захисту, з можливістю їхнього виведення на друк.
- Функціональний блок фіксації й оповіщення про порушення зв'язку з контролерами й обриві вимірювального каналу із вказівкою несправного датчика.
- Функціональний блок діагностування системи на всіх рівнях.
- Функціональний блок зміни настроювань контурів регулювання.
- Функціональний блок аналітичного контролю показників виробництва.
- Функціональний блок ведення таблиць лабораторних аналізів технологічних потоків.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

На вибір технічних коштів вплинули наступні особливості об'єкта керування:

- територіальна розподіленість об'єкта керування;
- необхідність використання різних алгоритмів керування високого рівня складності;
- велика кількість температурних датчиків у реакторах;
- високий рівень промислових перешкод;
- використання існуючих приладів із пневматичними сигналами.

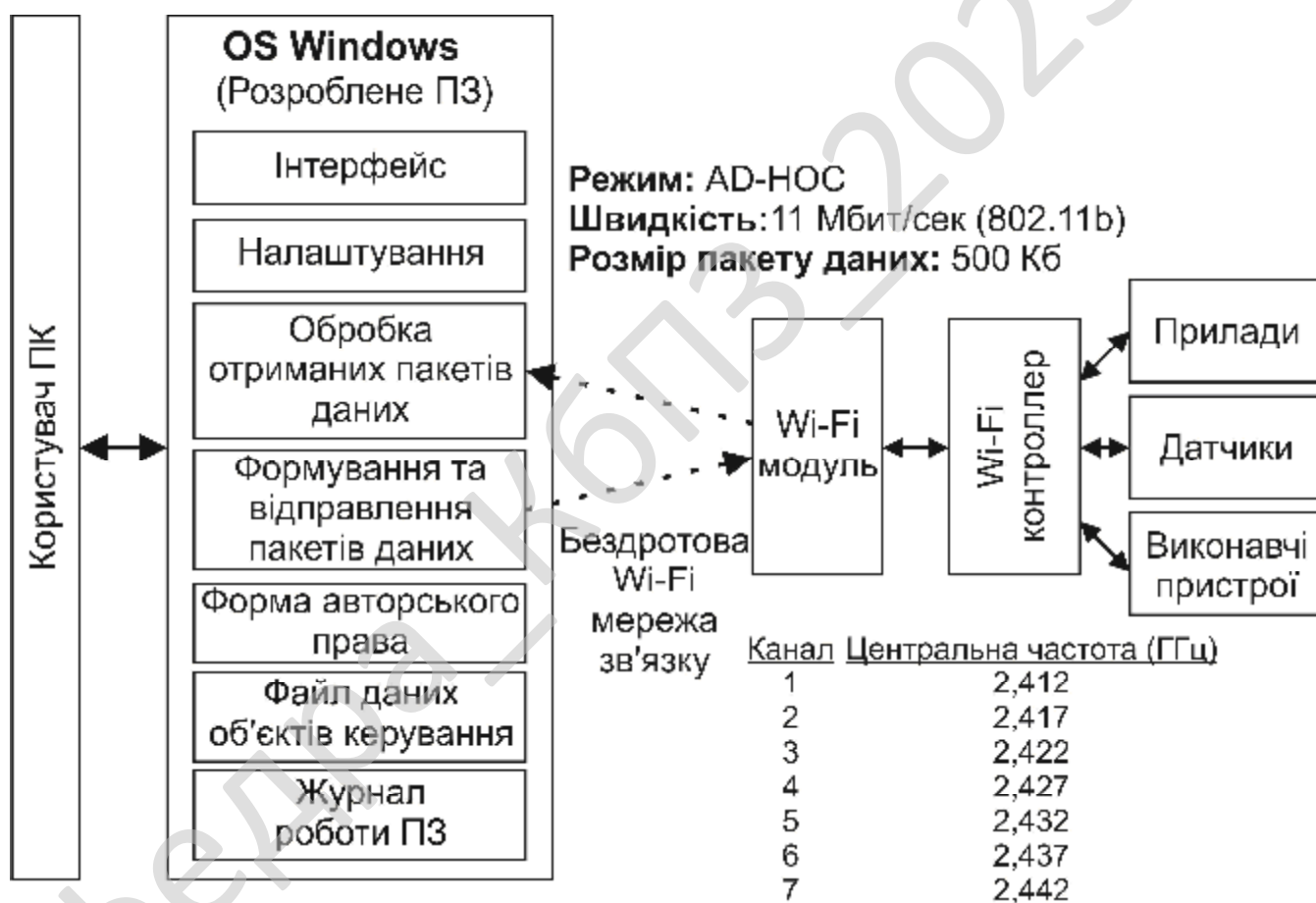


Рисунок 3.3 – Функціональна схема взаємодії з об'єктом керування

На рисунку 3.3 зображена функціональна схема взаємодії з об'єктом керування. З нього видно, що схема взаємодії складається з наступних функціональних елементів:

- Користувач ПК.
- Розроблене ПЗ.
- Wi-Fi модуль.
- Об'єкт керування.

Бездротова мережа Wi-Fi реалізується у режимі Ad-hoc. У режимі "Ad-hoc" кожний пристрій або станція можуть зв'язуватися безпосередньо один з одним, без використання точки доступу (AP). Режим "Ad-hoc" називають також "режим рівний-з-рівним" (peer-to-peer) або Independent Basic Service Set (IBSS – незалежний базовий набір служб).

Розроблене ПЗ складається з наступних функціональних модулів:

- Інтерфейс.
- Налаштування.
- Обробка отриманих пакетів даних.
- Формування та відправлення пакетів даних.
- Форма авторського права.
- Файл даних об'єктів керування.
- Журнал роботи ПЗ.

### 3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.4. З нього видно, що у системі взаємодіють наступні процеси.

Першим процесом, який запускається у системі, є процес початку роботи, який взаємодіє з процесом основного блоку програмного забезпечення.

Процес основного блоку програмного забезпечення взаємодіє з процесом модулю захисту програмного забезпечення.

Процес модулю захисту програмного забезпечення взаємодіє з головним процесом розробленого програмного забезпечення.

Головний процесом розробленого програмного забезпечення взаємодіє з

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

наступними процесами:

- Процес налаштування ПЗ.
- Процес авторської форми ПЗ.
- Процес інтерфейсу розробленого ПЗ, обробки запитів.

Останній процес взаємодіє з процесом перевірки Wi-Fi обладнання.

Процесом перевірки Wi-Fi обладнання взаємодіє з наступними процесами:

- Процес отримання вхідного пакету даних з об'єкта керування.
- Процес формування пакету даних та контрольної суми, який у свою

чергу, взаємодіє з процесом відправлення пакету даних до об'єкта керування.

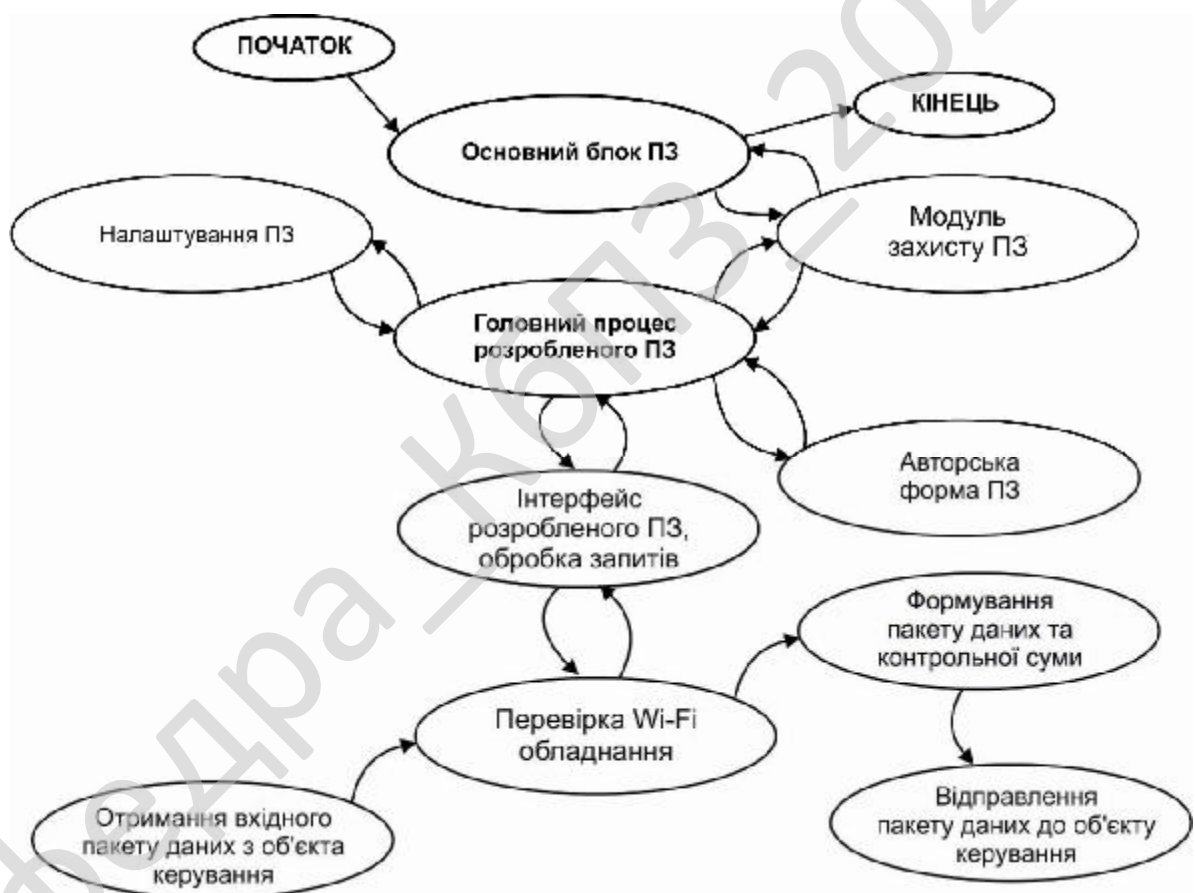


Рисунок 3.4 – Діаграма взаємодії процесів

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. Його блок-схема зображена на рисунку 4.1.

З рисунку видно, що після запуску програми спочатку відбувається початкова ініціалізація розробленого, у ході виконання бакалаврської роботи, програмного забезпечення. Потім здійснюється виклик підпрограми перевірки Wi-Fi обладнання.

Наступним кроком є визначення коду повернення.

Якщо код повернення повертає помилку, тоді відбувається виведення опису та коду помилки.

У іншому випадку, відбувається виконання наступних дій:

- Виводиться головне вікно програми.
- Створюється бездротова точка доступу.
- Відбувається читання файлу об'єктів керування.
- Відбувається пошук бездротових мереж.

Якщо є запит WM\_CLOSE, тобто закінчення роботи бездротової мережі, тоді відбуваються наступні дії:

- Приховуються вікна програмного забезпечення.
- Звільняються виділені ресурси програмного забезпечення.

У іншому випадку, відбувається перевірка на знаходження бездротової мережі.

Якщо вона знайдена, тоді відбувається перевірка на те, чи співпадає назва мережі з об'єктом керування.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Якщо співпадає, тоді виконуються наступні дії:

- Запускається підпрограма керування об'єктами.
- Записуються дії у журнал роботи програмного забезпечення.

Якщо є запит WM\_CLOSE, тобто закінчення роботи бездротової мережі, тоді відбуваються наступні дії:

- Приховуються вікна програмного забезпечення.
- Звільняються виділені ресурси програмного забезпечення.

На цьому програма закінчує свою роботу.

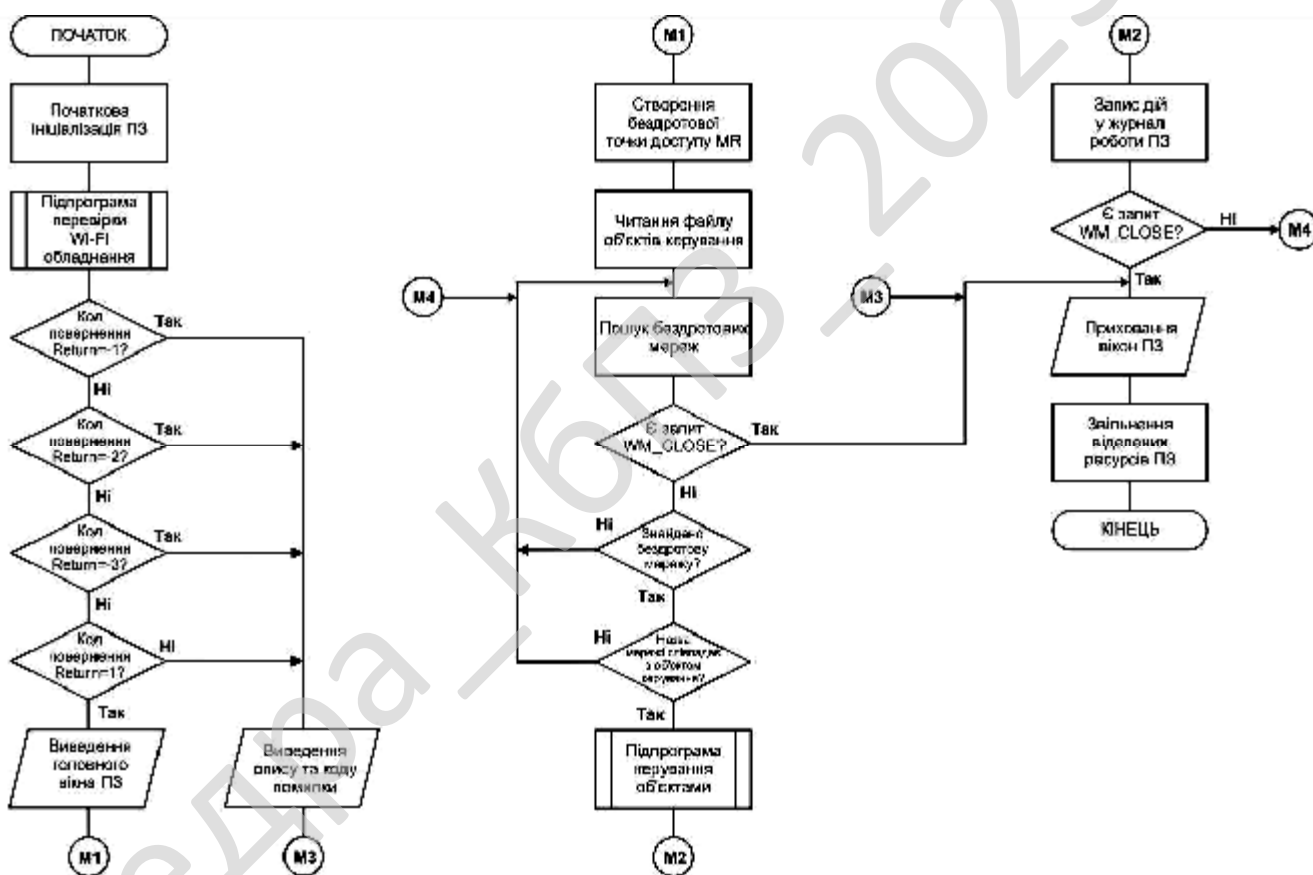


Рисунок 4.1 – Блок-схема програми

На рисунку 4.2 зображена блок-схема роботи підпрограми перевірки Wi-Fi обладнання. Вона працює наступним чином.

Спершу відбувається читання файлу налаштувань Wi-Fi обладнання.

Якщо не отримані дані файлу налаштувань, тоді відбувається встановлення сигналу завершення роботи ПЗ WM\_CLOSE.

У іншому випадку відбувається читання ключа реєстру Wi-Fi обладнання.

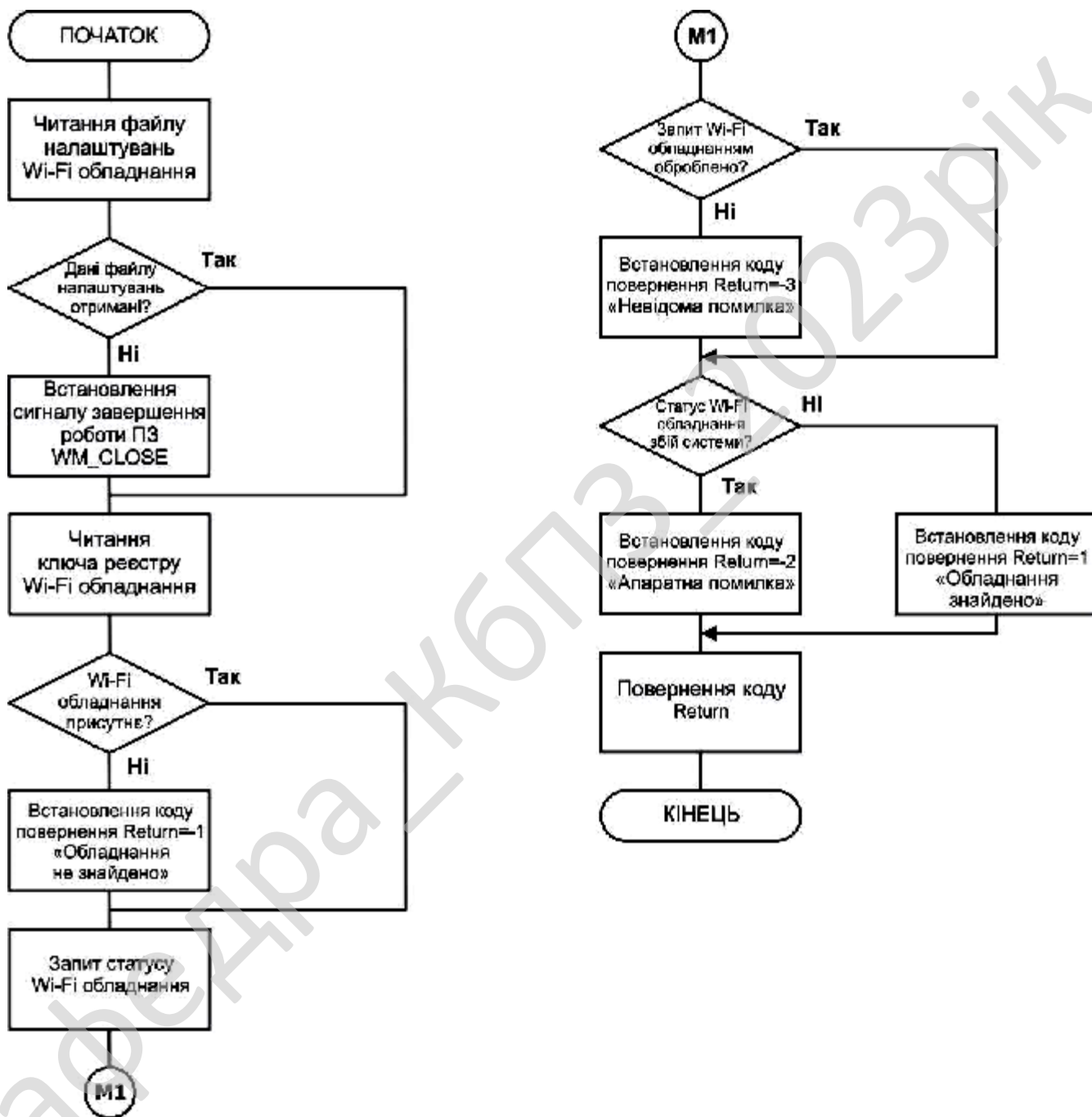


Рисунок 4.2 – Блок-схема роботи підпрограми перевірки Wi-Fi обладнання

Якщо Wi-Fi обладнання присутнє, тоді відбувається запит статусу Wi-Fi обладнання.

У іншому випадку відбувається встановлення коду повернення «Обладнання не знайдено».

Якщо запит Wi-Fi обладнанням не оброблений, тоді повертається код помилки «Невідома помилка».

Якщо статусом Wi-Fi обладнання є збій системи, тоді встановлюється код повернення «Апаратна помилка», у іншому випадку, встановлюється код «Обладнання знайдено».

Після виконання усіх вищеперерахованих дій відбувається повернення коду перевірки Wi-Fi обладнання.

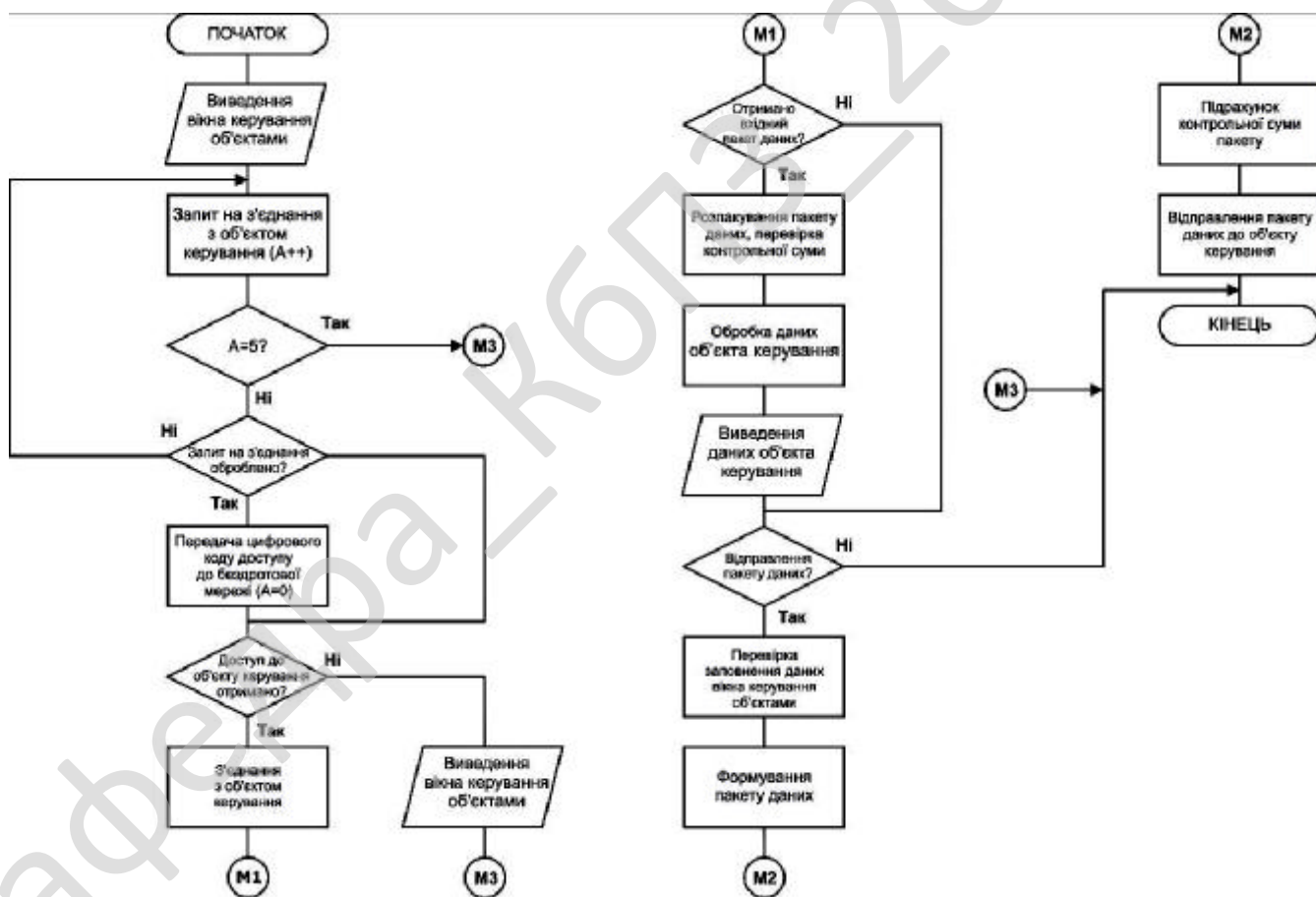


Рисунок 4.3 – Блок-схема роботи підпрограми керування об'єктами

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРБ-125.23.0008.00.00.ПЗ

Арк.

76

На рисунку 4.3 зображена блок-схема роботи підпрограми керування об'єктами. Вона працює наступним чином.

Спершу відбувається виведення вікна керування об'єктами.

Після цього подається запит на з'єднання з об'єктом керування.

Після виконання п'яти посилок, якщо з'єднання не відбулося, тоді підпрограма завершує свою роботу.

У іншому випадку відбувається перевірка на обробку запита на з'єднання.

Якщо запит на з'єднання оброблено, тоді відбувається передача цифрового коду доступу до бездротової мережі.

Наступним кроком, є обробка доступу до об'єкта керування.

Якщо доступ до об'єкту керування отримано, тоді відбувається з'єднання з об'єктом керування.

У іншому випадку відбувається виведення вікна керування об'єктами й підпрограма закінчує свою роботу.

Якщо отримано вхідний пакет даних, тоді відбувається виконання наступних дій:

- Розпакування пакету даних, перевірка контрольної суми.
- Обробка даних об'єктів керування.
- Виведення даних об'єктів керування.

Якщо відбувається відправлення пакету даних, тоді виконуються наступні дії:

- Перевіряється заповнення даних вікна керування об'єктами.
- Формується пакет даних.
- Підраховується контрольна сума пакету.
- Відбувається відправлення пакету даних до об'єкту керування.

Після виконання усіх вищеперахованих дій, підпрограма завершує свою роботу.

Опишемо роботу з бездротовою мережею, побудованою на використанні Wi-Fi, на низькому рівні в Delphi. Працювати будемо з winsock 2-ї версії.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

Для початку потрібно скопіювати файли цієї бібліотеки (якщо у вас їх немає) у папку Delphi\Lib, далі створюємо новий проект, і в розділ uses додаємо модуль winsock2. Напишемо сервер і клієнт і змусимо їх взаємодіяти один з одним.

### З'єднання

Почнемо із сервера. Для початку ініціалізуємо бібліотеку, щоб це зробити викличемо функцію WSASStartup. Ця функція має два параметри версія Winsock і структура типу TWSADATA яка прийме інформацію про Winsock. У першому параметрі вкажемо MAKEWORD(2,0) (для першої версії winsock буде MAKEWORD(1,0)), для другого параметра в var опишемо змінну inf, типу TWSADATA:

```
WSASStartup(MAKEWORD(2,0), inf);
```

Далі виконаємо функцію socket із трьома параметрами. Перший параметр це тип адресації, укажемо PF\_INET для синхронної роботи або AF\_INET для асинхронної. У другому параметрі виберемо протокол з яким збираємося працювати UDP(SOCK\_DGRAM) або TCP(SOCK\_STREAM), для нашого приклада вибираємо друге. У третьому параметрі потрібно вибрати конкретний протокол, ставимо 0 (IPPROTO\_IP). Для коректної роботи далі нам буде потрібна змінна зі значенням яке повертає функції socket, тому в var опишемо змінну типу integer, назвемо її наприклад – socks, і привласнимо їй значення нашої функції:

```
socks:=socket(AF_INET,SOCK_STREAM,0);
```

Тепер опишемо змінну структури TSocketAddr:

```
sockaddr:TSocketAddr;
```

Для роботи нам знадобляться тільки три поля із цієї структури, це sin\_family – сімейство адресації, sin\_addr – адреса сервера й sin\_port – порт для підключення.

```
sockaddr.sin_family := AF_INET;  
sockaddr.sin_port := htons(21);  
sockaddr.sin_addr.S_addr := htonl(INADDR_ANY);
```

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Прив'яжемо структуру до сокету, використовую функцію bind, знову ж із трьома параметрами – змінна зі значенням яке повертає функції socket, змінна із заповненими полями типу TSocketAddr і розмір цієї змінної:

```
bind(socks, @sockaddr, sizeof(sockaddr));
```

Залишилося почати прослуховувати порт, для це є функція listen, із двома параметрами – значення функції socket і кіл-в одночасних обробок:

```
listen(socks, 3);
```

Для клієнта все теж саме, але з деякими змінами: В var додамо змінну – Server:TInaddr; Їй потрібно привласнити значення з адресою сервера :

```
ServerName.s_addr:=inet_addr(PChar('127.0.0.1?));
```

Далі заміняємо – sockaddr.sin\_addr.S\_addr := htonl(INADDR\_ANY) на – sockaddr.sin\_addr := ServerName; Ну й замість bind і list додамо функцію connect, із трьома параметрами, такими ж як в bind:

```
connect(socks, @sockaddr, sizeof(sockaddr));
```

З підключенням розібралися, подивимося як це виглядає в зборі:

**Сервер:**

```
var inf: TWSADATA;  
socks: integer;  
sockaddr: TSocketAddr;  
begin  
  WSStartup(MAKEWORD(2,0), inf);  
  socks:=socket(AF_INET, SOCK_STREAM, 0);  
  sockaddr.sin_family := AF_INET;  
  sockaddr.sin_port := htons(210);  
  sockaddr.sin_addr.S_addr := htonl(INADDR_ANY);  
  bind(socks, @sockaddr, 16);  
  listen(socks, 3);  
end;
```

**Клієнт:**

```
var Server:TInaddr;  
inf: TWSADATA;  
socks: integer;  
sockaddr: TSocketAddr;  
begin  
  Server.s_addr:=inet_addr(PChar('127.0.0.1'));  
  WSStartup(MAKEWORD(2,0), inf);
```

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

```

socks:=socket(AF_INET,SOCK_STREAM,0);
sockaddr.sin_family := AF_INET;
sockaddr.sin_port := htons(210);
sockaddr.sin_addr := Server;
connect(socks,@sockaddr,sizeof(sockaddr));
end;

```

### Обмін даними

Для того щоб сервер прийняв підключення, використовується функція `accept`. Параметри в неї наступні – наша всюдисуща змінна зі значенням функції `socket`, змінна структури `TSocketAddr`, і розмір цієї змінної.

```

i:=sizeof(sockaddr);
accept(socks,sockaddr,i);

```

Для передачі даних використовується функція `send`. З 4 параметрами – перший параметр для клієнта це змінна зі значенням яке повертає функція `socket`, для сервера – змінна зі значенням від функції `accept`, інші значення однакові – текст повідомлення, розмір і прапор(прапори ми торкати не будемо):

```

send(socks,s,Length(s),0);

```

Щоб одержати дані, використовуємо функцію `recv`, з такими ж параметрами як в `send`:

```

recv(socks,s,Length(s),0);

```

Розглянемо приклад, коли клієнт посилає дані, а сервер їх приймає: Додаємо змінну, котра буде містити наші дані, потім привласнимо їй яке-небудь значення, якщо з'єднання відбулося успішно, те посилаємо.

```

var s: array[0..100] of Char;
begin
  s:='Test';
  if connect(socks,@sockaddr,sizeof(sockaddr))<>-1 then
    begin
      send(socks,s,Length(s),0);
    end;
  end;
end;

```

Що робить сервер? Для сервера потрібно зробити нескінченний цикл, де як тільки відбувається підключення, перевіряється значення функції, і якщо всі ок, одержуємо повідомлення, дивимося якщо одержали вдало, виводимо повідомлення з отриманими даними.

```

while true do
  begin
    sockc:=accept(socks,sockaddr,i);
    if sockc<>-1 then
      begin
        z:=recv(sockc,s,100,0);
        if z<>-1 then
          ShowMessage(s);
        end;
      end;
    end;
  end;
end;

```

## 4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SHACAL-1, який заснований на перетвореннях алгоритму SHA-1. SHACAL-1 шифрує 160-бітний блок даних з використанням 512-бітного ключа шифрування. Допускається використання більше коротких ключів шифрування (не коротше 128 біт), які перед виконанням розширення ключа (процедура розширення ключа також успадкована від SHA-1 і буде описана нижче) повинні бути доповнені нульовими бітами для досягнення 512-бітного розміру.

В алгоритмі SHACAL-1 передбачено 80 раундів шифрування. Шифруєме повідомлення представляється у вигляді п'яти 32-бітних субблоків  $A$ ,  $B$ ,  $C$ ,  $D$  і  $E$ , над якими в кожному раунді виконуються наступні дії:

$$A_{i+1} = K_i + (A_i \lll 5) + f_i(B_i, C_i, D_i) + E_i + M_i,$$

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

$$C_{i+1} = B_i \lll 30,$$

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

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

де  $i$  – номер раунду ( $i = 0 \dots 79$ ),

$K_i$  – фрагмент розширеного ключа для  $i$ -го раунду,

$f_i$  – функція для  $i$ -го раунду (див. нижче),

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

$\lll$  – операція побітового циклічного зрушення вліво,

$M_i$  – константи, що модифікують, певні в такий спосіб:

Раунди	Значення константи
0...19	5A827999
20...39	6ED9EBA1
40...59	8F1BBCDC
60...79	CA62C1D6

Використовувані в раундах функції  $f_i$  визначені так:

Раунди	Функція
0...19	$f(x,y,z)=(x\&y) (x'\&z)$
20...39,60...79	$f(x,y,z)=x \square \oplus y \square \oplus z$
40...59	$f(x,y,z)=(x \square \oplus y) (x \oplus z) (y \oplus z)$

У таблиці символами  $\&$ ,  $|$  і  $\oplus$  позначені, відповідно, побітові логічні операції «і», «або» й «або, що виключає» (XOR);  $x'$  позначає побітовий комплемент до  $x$ .

Шифртекстом є конкатенація вмісту змінних  $A_{80}$ ,  $B_{80}$ ,  $C_{80}$ ,  $D_{80}$  і  $E_{80}$ .

Процедура розширення ключа в алгоритмі SHACAL-1 також досить проста, вона виконується у два етапи:

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

Етап 2. Інші фрагменти розширеного ключа  $K_{16} \dots K_{79}$  обчислюються з перших 16 фрагментів у такий спосіб:

$$K_i = (K_{i-3} \oplus K_{i-8} \oplus K_{i-14} \oplus K_{i-16}) \lll 1.$$

Раунди розшифрування виконуються у зворотній послідовності; кожний з них має на увазі виконання наступних операцій:

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

$$B_i = C_{i+1} \lll 2,$$

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

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

$$E_i = K'_i + (B_{i+1} \lll 5)' + f'_i(C_{i+1} \lll 2, D_{i+1}, E_{i+1}) + A_{i+1} + M'_i + 4.$$

Тут запис  $f(x)$  позначає побітовий комплемент результату виконання операції  $f(x)$ .

Кафедра \_ КБПЗ \_ 2023 рік

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ КІБЕРБЕЗПЕКИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Програма має простий та інтуїтивно зрозумілий інтерфейс, який зображений на рисунку 5.1.

З рисунку видно, що інтерфейс користувача складається з наступних блоків:

- Блок меню.
- Блок інформації про Wi-Fi підключення.
- Вікно відображення знайдених мереж.
- Вікно відображення профілів мереж.

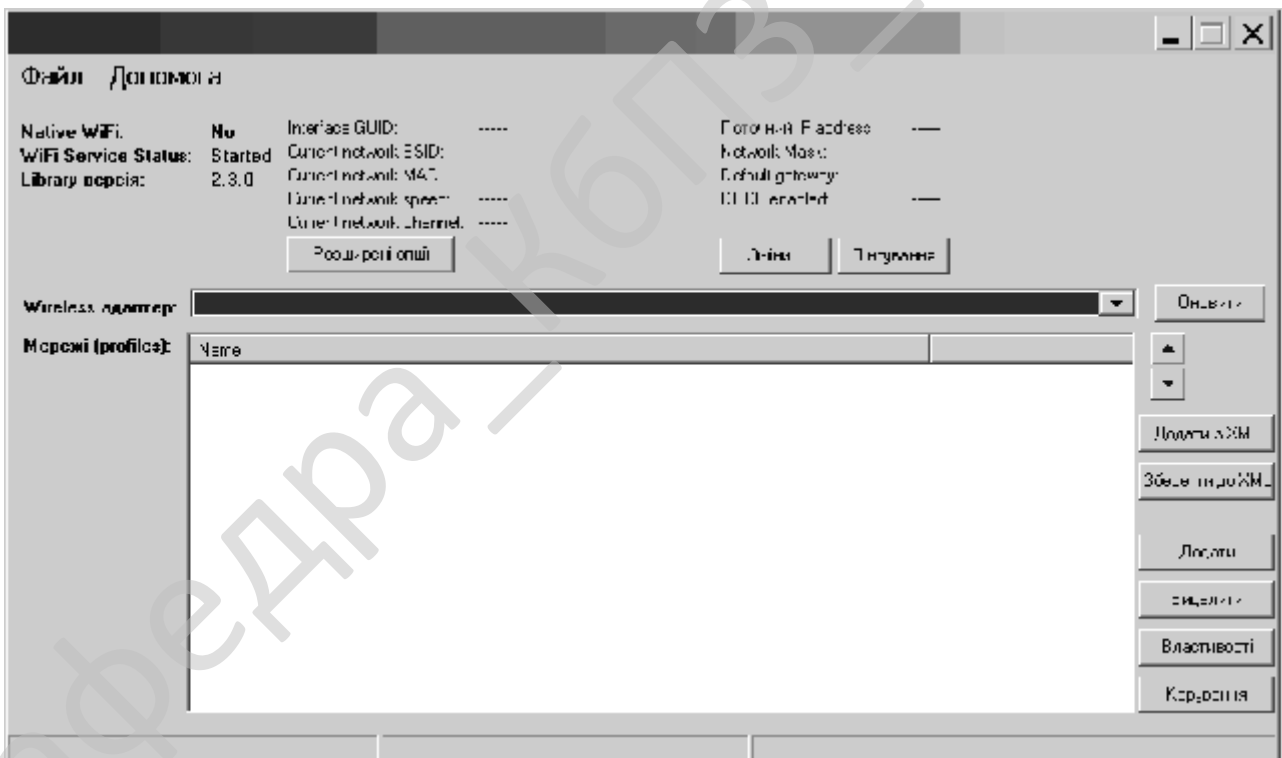


Рисунок 5.1 – Головне вікно ПЗ, початок роботи ПЗ

Блок меню складається з наступних елементів:

- Файл.
- Допомога.

На рисунку 5.2 зображено головне вікно ПЗ, підключення до бездротової мережі керування об'єктом.

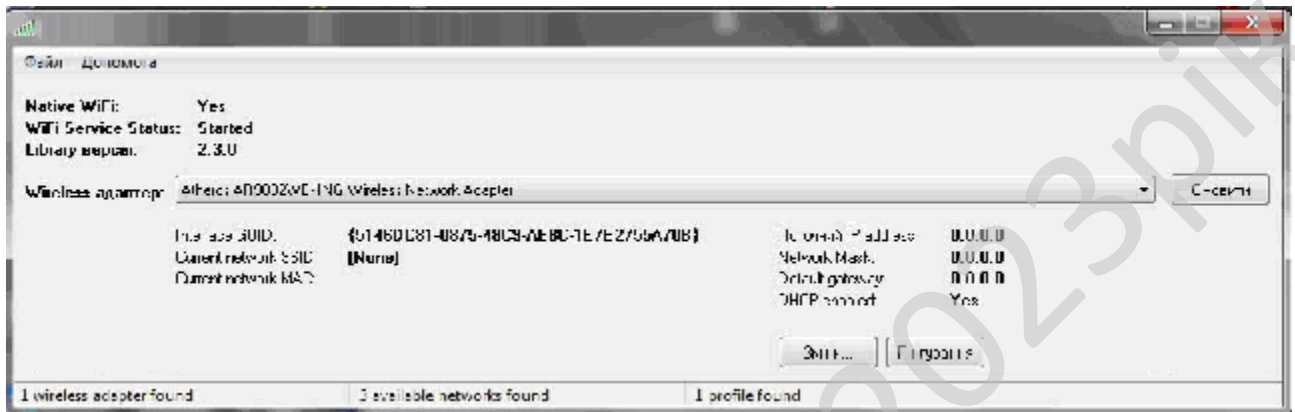


Рисунок 5.2 – Головне вікно ПЗ, підключення до захищеної бездротової мережі керування об'єктом

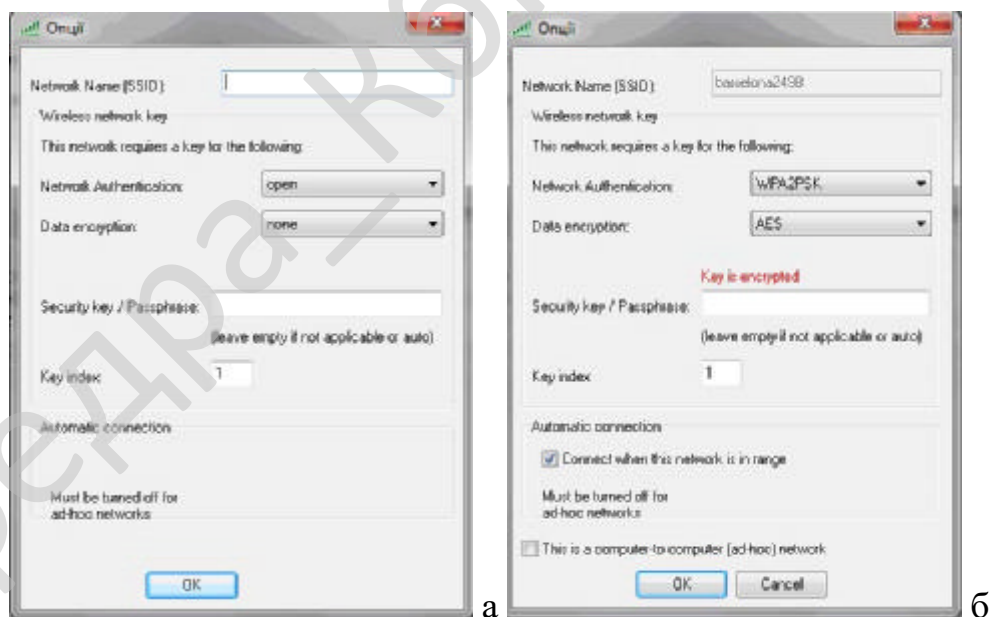


Рисунок 5.3 – Налаштування захищеної бездротової мережі: а – виклик вікна опцій, б – заповнення персональних даних системи керування об'єктом

На рисунку 5.3 зображені вікна налаштування бездротової мережі: а – виклик вікна опцій, б – заповнення персональних даних системи керування об'єктом. На рисунку 5.4 зображено вікно підпрограми PING за адресою.

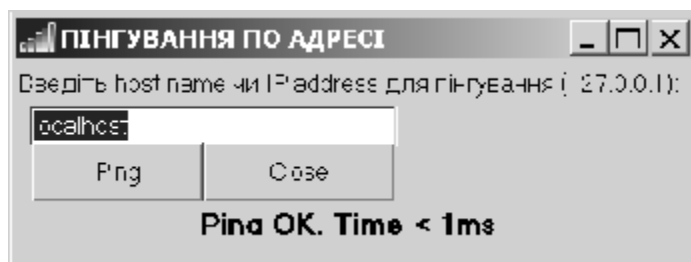


Рисунок 5.4 – Вікно підпрограми PING за адресою

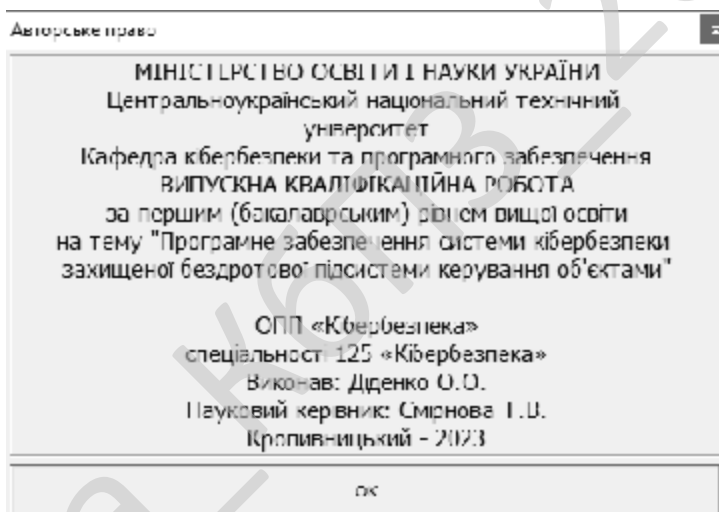


Рисунок 5.5 – Вікно авторського права розробленого ПЗ

На рисунку 5.5 зображено вікно авторського права розробленого програмного забезпечення, з якого можливо отримати наступну інформацію:

- Тема бакалаврської роботи.
- Виконавець бакалаврської роботи.
- Місце розробки бакалаврської роботи.

## 6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи кібербезпеки захищеної бездротової підсистеми керування об'єктами.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем захищеної бездротової підсистеми керування об'єктами.

– Досліджена система захищеної бездротової підсистеми керування об'єктами.

– На основі отриманих результатів досліджень створена програмна реалізація системи кібербезпеки захищеної бездротової підсистеми керування об'єктами.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання захищеної бездротової підсистеми керування об'єктами.

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 Delphi 10.4 Sydney. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи кібербезпеки захищеної бездротової підсистеми керування об'єктами. Це

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи 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 SHACAL-1.

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.

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>88</b>

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Смірнова Т.В., Дреєв О.М., «Інформаційна технологія оптимізації технологічного процесу відновлення та зміцнювання поверхонь валів зі сталі як хмарний сервіс» у *Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 92-107.*

2. Смірнова Т.В., Поліщук Л.І., «Дослідження хмарних технологій як сервісів для системи інженерних розрахунків» у *Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛІУС», 2020.С. С. 106-121.*

3. Smirnova, T., Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., «Biometric authentication using convolutional neural networks». *Lecture Notes in Networks and Systems*, 2021, vol. 152, pp. 85–98. (Scopus). Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85090914783&origin=resultslist>

4. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184. (Scopus). Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85118101973&origin=AuthorNamesList&txGid=9fba77a9424db54ff3b099e4400c22bb>

5. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156, 2022, Pages 390-399.* (Scopus). Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85133613188&origin=resultslist&sort=plf->

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

6. Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Smirnov O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland)* Volume 22, Issue 16, 6223, 2022. (Scopus).

Режим доступу: [https://www.scopus.com/record/display.uri?eid=2-s2.0-85137126823&origin=SingleRecordEmailAlert&dgcid=raven\\_sc\\_author\\_ru\\_ru\\_email&txGid=ee77c55f56deb4fdf2c6ce0e37730eb9](https://www.scopus.com/record/display.uri?eid=2-s2.0-85137126823&origin=SingleRecordEmailAlert&dgcid=raven_sc_author_ru_ru_email&txGid=ee77c55f56deb4fdf2c6ce0e37730eb9)

7. Смірнова Т.В., Дреєв О.М., Смірнов О.А., «Експертна система оптимізації процесу відновлення та зміцнення поверхонь деталей типу «вал» електродуговим напиленням», *Системи управління, навігації та зв'язку*, № 2 (54). с. 149-154, 2019. Режим доступу: [http://nbuv.gov.ua/UJRN/suntz\\_2019\\_2\\_32](http://nbuv.gov.ua/UJRN/suntz_2019_2_32)

8. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М., «Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей», *Центральноукраїнський науковий вісник. Технічні науки*. № 1(32). с. 184-194, 2019. Режим доступу: [http://nbuv.gov.ua/j-pdf/znpkntu\\_2019\\_1\\_23.pdf](http://nbuv.gov.ua/j-pdf/znpkntu_2019_1_23.pdf)

9. Смірнова Т.В., Дреєв О.М., Смірнов О.А., Солових Є.К., «Методи оптимізації технологічних процесів відновлення сталевих покриттів», *Shipbuilding & marine infrastructure / Суднобудування і морська інфраструктура* № 1 (11). с. 48-57, 2019. Режим доступу: <http://smi.nuos.mk.ua/archive/2019/1/6.pdf>

10. Smirnova T., Ageev M., Lopata L., Dudan A., Lopata A., «Of combined electric arc coatings», *Problems of Tribology*, Vol. 24 № 3/93. P. 51-61, 2019. Режим доступу: <http://tribology.khnu.km.ua/index.php/ProbTrib/article/view/727/1207>

11. Смирнова Т.В., Смирнов А.А., Дреєв А.Н., Дудан А.В., «Оптимизация технологического процесса восстановления и упрочнения поверхностей с заданными характеристиками в виде облачного сервиса», *Вестник Полоцкого государственного университета. Серия В, Промышленность. Прикладные науки*.

Республика Беларусь - 2020. - № 3. - С. 50-61. Режим доступа: <http://elib.psu.by:8080/handle/123456789/24988>

12. Смірнова Т.В., Поліщук Л.І., Смірнов О.А., Буравченко К.О., Макевнін А.О., «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020. Режим доступа: <https://www.csecurity.kubg.edu.ua/index.php/journal/article/view/132/132> (Категорія Б)

13. Смірнова Т.В., «Формалізація та реалізація структури технологічного процесу електродугового напилення для оптимізаційної експертної системи», *Технічні науки та технології*. № 1(19). С. 104-113. 2020. Режим доступа: <http://tst.stu.cn.ua/article/view/204062> (Категорія Б)

14. Smirnova T., Smirnov I., Lopata A., Lopata L., «Improvement of functional properties of gas-thermal coatings by electro-contact treatment», *Problems of Tribology*, Vol. 25, № 1/95, P. 41-48. 2020. Режим доступа: <http://tribology.khnu.km.ua/index.php/ProbTrib/article/view/742/1222>

15. Смірнова Т.В. «Формування евристичних правил, бази знань та формалізація структури й правил технологічного процесу для оптимізаційної хмарної інформаційної системи», *Системи управління, навігації та зв'язку*, № 2 (60). с. 101-104, 2020. Режим доступа: <http://journals.nupp.edu.ua/sunz/article/view/1839/1515> (Категорія Б)

16. Смірнова, Т.В., Смірнов, С.А., Минайленко, Р.М., Доренський, О.П., Сисоєнко С.В., «Хмарна автоматизована система інтелектуальної підтримки прийняття рішень для технологічних процесів». *Вісник Черкаського державного технологічного університету. Технічні науки*. №4, 2020, С. 84-92. Режим доступа: <https://er.chdtu.edu.ua/handle/ChSTU/1888> (Фахове видання. Категорія «Б»)

17. Смирнова Т.В., Дудан А.В., Смирнов А.А., «Формализация структуры технологического процесса электродугового напиления». *Сборник научных трудов «Актуальные вопросы машиноведения»*. Объединенный институт

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

машиностроения Национальной Академии Наук Беларуси. №9. С. 308-312, 2020.

Режим доступу: <http://oim.by/ru/izdaniya/sbornik.html>

18. Смірнова Т.В., Пархоменко Д.О., Голубець Р.О., Щербань А.В., Багдасарян Е.К., «Формалізація проблеми підтримки технологічних процесів у хмарних сервісах». *Системи озброєння і військова техніка*. 2021. № 3(67). С. 105-112. <https://doi.org/10.30748/soivt.2021.67.14> (Фахове видання. Категорія «Б»)

19. Смірнова Т.В., Столяренко М.П., Янков М.О., Грудік В.В., Моторін Ю.Ю., «Модель реалізації структури технологічного процесу у хмарному сервісі». *Збірник наукових праць Харківського національного університету Повітряних Сил*. 2021. № 4(70). С. 132-142. <https://doi.org/10.30748/zhups.2021.70.19>. (Фахове видання. Категорія «Б»)

20. Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., Смірнов О.А., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95 Режим доступу: <http://ais.khpi.edu.ua/article/view/247293> (Фахове видання. Категорія «Б»)

21. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Бурмак Ю.А., Оспанова Д.М., «Удосконалений модуль криптографічного захисту інформації в сучасних інформаційно-комунікаційних системах та мережах». *Кібербезпека: освіта, наука, техніка*. 2021. № 2(14). С. 176-185. Режим доступу: <https://csecurity.kubg.edu.ua/index.php/journal/article/view/329> (Фахове видання. Категорія «Б»)

22. Смірнова Т.В., Бурмак Ю.А., Улічев О.С., Усік П.С., Доренський О.П., «Стійка функція шифрування удосконаленого модуля криптографічного захисту інформації в інформаційно-комунікаційних системах» *Кібербезпека: освіта, наука, техніка*. 2021. № 1(13). С. 183-201. Режим доступу: <https://csecurity.kubg.edu.ua/index.php/journal/article/view/346> (Фахове видання. Категорія «Б»)

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

23. Смірнова Т.В., Якименко Н.М., Улічев О.С., Коноплицька-Слободенюк О.К., Смірнов С.А., «Дослідження лінійних перетворень запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Кібербезпека: освіта, наука, техніка*. 2022. № 3(15). С. 85-92. Режим доступу: <https://csecurity.kubg.edu.ua/index.php/journal/article/view/337> (Фахове видання. Категорія «Б»)

24. Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., Смірнов О.А., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022. № 3(69). С. 93-98. Режим доступу: <http://journals.nupp.edu.ua/sunz/article/view/2615> (Фахове видання. Категорія «Б»)

25. Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., Смірнов О.А. «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89. Режим доступу: <http://journals.nupp.edu.ua/sunz/article/view/2449/1918> (Фахове видання. Категорія «Б»)

26. Смірнова Т.В., Якименко Н.М., Смірнов О.А., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, 2022. № 2 (307). С. 46-52. Режим доступу: <http://journals.khnu.km.ua/vestnik/?cat=65> (Фахове видання. Категорія «Б»)

27. Смірнова Т.В., Моторін Ю.Ю., Буравченко К.О., Бочуля Т.В., Коваленко О.В. «Вибір оптимальної технології побудови хмарної інформаційно-

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

комунікаційної системи автоматизації виробничих процесів». *Вимірювальна та обчислювальна техніка в технологічних процесах*, № 1 (2022). С. 15-26. 2022.

Режим доступу: <http://vottp.khmnu.edu.ua/index.php/vottp/article/view/30/36>

(Фахове видання. Категорія «Б»)

28. Смірнова Т.В., Янков М.О., Грудік В.В., Горбов В.О., Коваленко А.С. «Планування радіопокриття та моделювання поширення радіосигналів мобільних мереж 5G для автоматизації виробничих процесів». *Електронне моделювання*, № 3, т. 44. С. 113-122. 2022. Режим доступу: <https://www.emodel.org.ua/uk/archive-ukr/2022/44-3-u/c-113-122> (Фахове видання. Категорія «Б»)

29. Смірнова Т.В., Буравченко К.О., Щербань А.В., Багдасарян Е.К., Коваленко А.С. «Проектування та оптимізація структурованих кабельних систем для автоматизації виробничих процесів підприємства» *Сучасні інформаційні системи*. 2022. Т. 6, № 1. С. 129-133. Режим доступу: <http://ais.khpi.edu.ua/article/view/254256/251522> (Фахове видання. Категорія «Б»)

30. Смірнова Т.В. «Метод забезпечення надійності підключення вузлів до інформаційно-комунікаційної системи підприємства на базі 5G» *Сучасні інформаційні системи*. 2022. Т. 6, № 2. С. 82-87. Режим доступу: <http://ais.khpi.edu.ua/article/view/260767/257189> (Фахове видання. Категорія «Б»)

31. Смірнова Т.В., Верховець О.С., Буравченко К.О., Смірнов С.А., Гермак В.С. «Алгоритмічне забезпечення для планування інформаційно-комунікаційної мережі підприємства на базі технологій 5G». *Збірник наукових праць Національного університету кораблебудування імені адмірала Макарова*. № 1 (488). 2022. С. 81-88. Режим доступу: <http://znp.nuos.mk.ua/archives/2022/1/11.pdf> (Фахове видання. Категорія «Б»).

32. Смірнова Т.В., Гнатюк С.О., Юдін О.Ю., Сидоренко В.М., Жаксигулова Д.Д., «Експериментальне дослідження моделі розрахунку кількісного критерію оцінювання захищеності інформаційно-телекомунікаційних систем критичної інфраструктури держави» *Кібербезпека: освіта, наука, техніка*. № 4(16). 2022. С. 6-18. Режим доступу:

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

видання. Категорія «Б»)

33. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., «Метод розрахунку критичності галузевих інформаційно-телекомунікаційних систем». *Наукоємні технології* № 2(54), 2022. С. 94-104. Режим доступу: <https://jrnl.nau.edu.ua/index.php/SBT/article/view/16757> (Фахове видання. Категорія «Б»)

34. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37. Режим доступу: <https://jrnl.nau.edu.ua/index.php/PIU/article/view/16844> (Фахове видання. Категорія «Б»)

35. Смірнова Т.В., Охріменко Т.О., Бредніков А.В., Макаренко О.І., «Обґрунтування необхідності впровадження інформаційних систем управління підприємствами нафтопереробної промисловості». *Інформаційно-аналітичні системи обробки даних*, Том 23, № 4, 2021. С. 17-27. Режим доступу: <http://drsp.ipri.kiev.ua/article/view/265648> (Фахове видання. Категорія «Б»)

36. Смірнова Т.В., Смірнов О.А., Дреєв О.М., С.А. Смірнов, «Використання хмарних експертних систем в сфері інформаційного забезпечення обробки поверхні деталей», *Комп'ютерна інженерія і кібербезпека: досягнення та інновації*, м. Кропивницький. 27-29 листопада, 2018, с. 111-113

37. Смірнова Т.В., Дреєв О.М., Смірнов О.А. «Захист даних у інформаційній технології відновлення поверхонь деталей у вигляді хмарної платформи як послуги», *Проблеми кібербезпеки інформаційно-телекомунікаційних систем*, м. Київ, 11-12 квітня, 2019, с. 221-224.

38. Смірнова Т.В., Дреєв О.М., Солових Є.К., «Хмарний сервіс інформаційної технології оптимізації технологічного процесу відновлення та зміцнювання поверхонь зі сталі», *Інформаційна безпека та інформаційні*

технології, *Information Security and Information Technologies*, м. Харків, 24-25 квітня, 2019, с. 36

39. Смірнова Т.В., Дреєв О.М., Смірнов О.А., «Побудова хмарної експертної системи оптимізації технологічного процесу електродугового напилення сталевих покриттів», *Міжнародний форум з інформаційних систем і технологій INFOS-2019*, м. Харків, 24-27 квітня, 2019, с. 95-98.

40. Смирнова Т.В., Смирнов А.А., Соловых Е.К. «Разработка математической модели и программного обеспечения для оптимизации режима электроконтактной обработки газотермических покрытий», *Комплексне забезпечення якості технологічних процесів та систем*, том 2, м. Чернігів, 14 - 16 травня, 2019, с. 78-79.

41. Смірнова Т.В., Дреєв О.М., Солових Є.К., Смірнов О.А., «Експертна система інформаційної технології оптимізації абстрактного технологічного процесу як хмарного сервісу», *Інформаційні технології: Наука, техніка, технологія, освіта, здоров'я. MicroCAD-2019*, м. Харків, 15-17 травня, 2019, с. 195.

42. Смірнова Т.В., Дреєв О.М., Смірнов О.А., «Формування абстрактних експертних систем на основі досліджень відомих експертних систем», *XXI Міжнародний науково-практичний семінар «Комбінаторні конфігурації та їх застосування»*, м. Кропивницький, 17-18 травня, 2019, с. 143-147.

43. Смірнова Т.В., Смірнов О.А., «Захист даних у інформаційній технології відновлення поверхонь деталей у вигляді хмарної платформи як послуги», *Інформація, комунікація, суспільство – 2019*, см.т. Чинадієво, 16-18 травня, 2019, с. 25-26

44. Смирнова Т.В., Дреєв А.Н., Смирнов А.А., «Информационная технология оптимизации технологического процесса восстановления и упрочнения поверхностей в виде облачного сервиса», *Modern Information, Measurement And Control Systems: Problems And Perspectives (MIMCS 2019)*, Baku, Azerbaijan, 01-02 July, 2019, p. 282.

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

45. Смірнова Т.В., Дреєв О.М., Смірнов О.А., «Огляд відомих експертних систем оптимізації технологічних процесів», *Стратегія якості в промисловості і освіті*, м. Варна, Болгарія, 3-6 червня, 2019, с.442-444

46. Смірнова Т.В., Дреєв О.М., Смірнов О.А., «Формалізація та узагальнення інформаційної моделі технологічних операцій зміцнення та відновлення сталевих поверхонь», *Інтелектуальні системи та інформаційні технології*, м.Одеса, 19-24 серпня, 2019, с. 211-213.

47. Смірнова Т.В., Дреєв О.М., Смірнов О.А., «Формування інформаційної моделі технологічного процесу». *V Всеукраїнська науково-практична Інтернет-конференція «Електронні та мехатронні системи: теорія, інновації, практика»*, м. Полтава, 8 листопада, 2019, с. 87-91.

48. Smirnova T.V., Chernovol M.S., Ageev M. «Study of the spraying process and the influence of its factors on the properties of electric arc spraying coatings». *Materials of the 20th international scientific and technical seminar «Modern questions of production and repair in industry and in transport»*, Tbilisi, Georgia, 23-28 March 2020, с. 201-205.

49. Смірнова Т.В., Солових Є.К., Смірнов О.А., «Дослідження стандартів забезпечення кібербезпеки хмарних технологій як сервісів», *X міжнародна науково-технічна конференція «ITSec: Безпека інформаційних технологій»*, 19-24 березня 2020 р. – К.: НАУ, 2020. с. 28-29. Режим доступу: [http://bit.nau.edu.ua/wp-content/uploads/2020/05/ITSec-2020\\_Zbirnyk.pdf](http://bit.nau.edu.ua/wp-content/uploads/2020/05/ITSec-2020_Zbirnyk.pdf)

50. Смірнова Т.В., Поліщук Л.І., Смірнов О.А., «Аналіз хмарних технологій як сервісів», *XIII всеукраїнська науково-практична конференція «Комп'ютерні інтелектуальні системи та мережі (KICM-2020)»*. м. Кривий Ріг. 24-26 березня, 2020, С. 210-212.

51. Смірнова Т.В., Поліщук Л.І., Дреєв О.М., «Застосування сервісу SAЕaaS як системи інженерних розрахунків з використанням хмарних технологій», *II Міжнародна науково-практична конференція «Інформаційна*

					ВКРБ-125.23.0008.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

*безпека та інформаційні технології, Information Security and Information Technologies», м. Кропивницький, 2-3 квітня, 2020, с. 52.*

52. Смірнова Т.В., Дреєв О.М., Смірнов О.А., Солових Є.К., «Інформаційна структура технологічного процесу електродугового напилення». *Всеукраїнська науково-технічна конференція «Стан, досягнення та перспективи інформаційних систем і технологій», м. Одеса, 21-22 квітня, 2020, с. 184-186.*

53. Смірнова Т.В., Дреєв О.М., Смірнов О.А., Солових Є.К., «Реалізація інформаційної структури технологічного процесу електродугового напилення», *XXII Міжнародний науково-практичний семінар «Комбінаторні конфігурації та їх застосування», м. Кропивницький, 15-16 травня, 2020, с. 158-162.*

54. Смірнова Т.В., Щербань А.В., Моторін Ю.Ю., Смірнов О.А., «Перспективні напрямки забезпечення кібербезпеки сервісу CAEaaS», *VI Всеукраїнська науково-практична конференція «Перспективні напрями захисту інформації», м. Одеса, 2-6 вересня 2020, с. 58-59*

55. Смірнова Т.В., Дреєв О.М., Смірнов О.А. «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». *IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30. Режим доступу: <http://www.kntu.kr.ua/doc/science/zahody/vikl/2021/1-tez.pdf>*

56. Смірнова Т.В., Смірнов О.А., Буравченко К.О. «Системи підтримки технологічних процесів з використанням хмарних технологій». *IV Міжнародна науково-практична конференція «Інтеграція інформаційних систем і інтелектуальних технологій в умовах трансформації інформаційного суспільства», м. Полтава, 21-22 жовтня 2021 р., – Полтава: ПДАУ. – 2021. – С. 56-60. Режим доступу: <https://doi.org/10.32782/978-966-289-562-9>*

57. Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Стійка функція шифрування удосконаленого модуля криптографічного захисту інформації». *I міжнародна науково-практична конференція «Цифрова трансформація*

*суспільства (Digital Society – 2022)», м. Кропивницький. 21-22 квітня 2022 р. – Кропивницький: ЦНТУ. – 2022. – С. 52-55. Режим доступу: [http://dspace.kntu.kr.ua/jspui/bitstream/123456789/11933/1/%d0%a1onference\\_abstract\\_book\\_Digital\\_Society\\_2022.pdf](http://dspace.kntu.kr.ua/jspui/bitstream/123456789/11933/1/%d0%a1onference_abstract_book_Digital_Society_2022.pdf)*

58. Смірнова Т.В., Смірнов С.А., Смірнов О.А. «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування». *Дванадцята міжнародна науково-практична конференція «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління», м. Баку – Харків – Жиліна. 27-28 квітня 2022 року. – Харків: ХІІ. – 2022. – С. 130. Режим доступу: [https://nure.ua/wp-content/uploads/conf-2022-akov/telecom\\_2022\\_volume\\_1.pdf](https://nure.ua/wp-content/uploads/conf-2022-akov/telecom_2022_volume_1.pdf)*

59. Смірнова Т.В., Смірнов С.А., Смірнов О.А. «Оптимальна технологія побудови хмарної інформаційно-комунікаційної системи автоматизації виробничих процесів». *Міжнародна науково-практична конференція «Комбінаторні конфігурації та їх застосування», м. Кропивницький - Запоріжжя. 13-14 травня 2022р. – Кропивницький: ЦНТУ. – 2022. – С. 30. Режим доступу:*

60. Смірнова Т.В., Смірнов С.А., Смірнов О.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування». *V міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 19-20 травня 2022р. – Кропивницький: ЦНТУ. – 2022. – С. 26-27. Режим доступу: [http://dspace.kntu.kr.ua/jspui/bitstream/123456789/11994/3/%d0%97%d0%b1%d1%96%d1%80%d0%bd%d0%b8%d0%ba\\_%d1%82%d0%b5%d0%b7%202022.pdf](http://dspace.kntu.kr.ua/jspui/bitstream/123456789/11994/3/%d0%97%d0%b1%d1%96%d1%80%d0%bd%d0%b8%d0%ba_%d1%82%d0%b5%d0%b7%202022.pdf)*

61. Смірнова Т.В., Одарченко Р.С., Смірнов О.А. «Проектування приватних мереж 5G для потреб виробничих підприємств». *VI Міжнародна науково-практична конференція «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2022), м. Черкаси, 23-25 червня 2022 р. – Черкаси: ЧДТУ, 2022. – С.65-67.*

					<b>ВКРБ-125.23.0008.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

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

					<b>ВКРБ-125.23.0008.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Діденко О.О.				Програмне забезпечення системи кібербезпеки захищеної бездротової підсистеми керування об'єктами	Літ.	Аркуш	Аркушів
Перевірів	Смірнова Т.В.					Б	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КБ-19			
Затв.	Смірнов О.А.							

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи кібербезпеки захищеної бездротової підсистеми керування об'єктами.

## 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. № 12-02 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 захищеної бездротової підсистеми керування об'єктами.

## 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-125.23.0008.00.00.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 захищеної бездротової підсистеми керування об'єктами;
- 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.

					<b>ВКРБ-125.23.0008.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

## 5.7 Вимоги до складу та параметрів технічних засобів

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

## 5.8 Вимоги до інформаційної і програмної сумісності

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

### 5.8.1 Обладнання

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

### 5.8.2 Мова програмування

Середовище Delphi 10.4 Sydney.

					ВКРБ-125.23.0008.00.00.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 – 100 аркушів.

## 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).

					<b>ВКРБ-125.23.0008.00.00.ТЗ</b>	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 12.06.2023 р.

					ВКРБ-125.23.0008.00.00.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 захищеної бездротової  
підсистеми керування об'єктами*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 49

Літера: РП

Кропивницький – 2023 року

**Основна програма****Файл Wifi.dpr основної програми**

```
program Wifi.dpr;
// Тема програми - Програмне забезпечення системи кібербезпеки захищеної
// бездротової підсистеми керування об'єктами
// Автор - Діденко Олександр Олександрович
// гр. КБ-19, Кропивницький 2023

Uses // Бібліотеки які підключаються
  Forms,
  Unit1 in 'Unit1.pas' {fmMain},
  WiFiMan in 'WiFiMan.pas',
  Unit2 in 'Unit2.pas' {fmChangeIP},
  Unit3 in 'Unit3.pas' {fmProfProp},
  Unit4 in 'Unit4.pas' {fmAdapterOpt},
  Unit5 in 'Unit5.pas' {fmPing},
  Unit6 in 'Unit6.pas' {Form6};

{$R *.res} // Ресурси проекту магістерського ПЗ

Begin // Початок коду основного потоку розробленої магістерської програми
  Application.Initialize; // початок ініціалізації
  Application.CreateForm(TfmMain, fmMain); // створення форми 1
  Application.CreateForm(TfmChangeIP, fmChangeIP); // створення форми 2
  Application.CreateForm(TfmProfProp, fmProfProp); // створення форми 3
  Application.CreateForm(TfmAdapterOpt, fmAdapterOpt); // створення форми 4
  Application.CreateForm(TfmPing, fmPing); // створення форми 5
  Application.CreateForm(TForm6, Form6); // створення форми 6
  Application.Run; // Запуск на виконання ПЗ
end.
```

## Файл Unit1.pas

```

unit Unit1; //Назва модулю

interface // об'ява даних

uses // Бібліотеки які підключаються
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, StdCtrls, ComCtrls, Buttons, XPMAN, ShellAPI, WiFiMan;

Type //Об'ява власних типів даних
  TfmMain = class(TForm) // об'ява класу
    // Початок опису компонентів класу TfmMain
    mMenu: TMainMenu; // Основне меню
    nnnFile: TMenuItem; // значення меню;
    nnnHelp: TMenuItem; // значення меню;
    Label1: TLabel; // Мітка 1
    cbAdapters: TComboBox; // Кнопки вибору
    Label2: TLabel; // Мітка 2
    lbGuid: TLabel; // Мітка
    Label4: TLabel; // Мітка 4
    lvAvNets: TListView;
    Label5: TLabel; // Мітка 5
    lbCurrent: TLabel; // Мітка
    bkConnect: TButton; // кнопка
    bkDisconnect: TButton; // кнопка
    Label7: TLabel; // Мітка 7
    lvProfiles: TListView;
    bkRemove: TButton; // кнопка
    stBar: TStatusBar; // Статусно строка вікна
    bkManagement: TButton; // кнопка
    // Продовження опису компонентів класу TfmMain
    sbUp: TSpeedButton; // Розширена кнопка
    sbDown: TSpeedButton; // Розширена кнопка;
    XPMANifest: TXPMANifest;
    bkRefresh: TButton; // кнопка
    bkScan: TButton; // кнопка
    nnEnableLog: TMenuItem; // значення меню
    nnDisableLog: TMenuItem; // значення меню;
    N1: TMenuItem; // значення меню;
    nnExit: TMenuItem; // значення меню;
    N2: TMenuItem; // значення меню;
    nnRefresh: TMenuItem; // значення меню;
    nnHelp: TMenuItem; // значення меню;
    pmAvNet: TPopupMenu; // Впливаюче меню
    nnConnect: TMenuItem; // значення меню;
    nnDisconnect: TMenuItem; // значення меню;
    Label3: TLabel; // Мітка 3
    lbCurSpeed: TLabel; // Мітка
    Label8: TLabel; // Мітка 8
    lbChannel: TLabel; // Мітка
    Label6: TLabel; // Мітка 6
    lbNative: TLabel; // Мітка
    Label10: TLabel; // Мітка 10
    lbSvcStatus: TLabel; // Мітка
    Label9: TLabel; // Мітка 9
    lbVer: TLabel; // Мітка
    Label11: TLabel; // Мітка 11
    lbIP: TLabel; // Мітка
    lbMask: TLabel; // Мітка
    Label14: TLabel; // Мітка 14
    Label15: TLabel; // Мітка 15
    lbGate: TLabel; // Мітка
    Label17: TLabel; // Мітка 17
    lbDHCP: TLabel; // Мітка
    bkChangeIPInfo: TButton; // кнопка
    bkAdd: TButton; // кнопка
    bkChange: TButton; // кнопка
    bkAddXml: TButton; // кнопка

```

```

bkSaveXML: TButton; // кнопка
opXML: TOpenDialog;
sdXML: TSaveDialog;
bkAdapterOptions: TButton; // кнопка
Button1: TButton; // кнопка
Label12: TLabel; // Мітка 12
lbCurMac: TLabel; // Мітка
procedure bkRefreshClick(Sender: TObject);
procedure cbAdaptersClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure nnExitClick(Sender: TObject);
procedure nnRefreshClick(Sender: TObject);
procedure nnEnableLogClick(Sender: TObject);
procedure nnDisableLogClick(Sender: TObject);
procedure nnHelpClick(Sender: TObject);
procedure bkScanClick(Sender: TObject);
procedure bkConnectClick(Sender: TObject);
procedure bkDisconnectClick(Sender: TObject);
procedure bkManagementClick(Sender: TObject);
procedure sbUpClick(Sender: TObject);
procedure sbDownClick(Sender: TObject);
procedure bkRemoveClick(Sender: TObject);
procedure pmAvNetPopup(Sender: TObject);
procedure nnConnectClick(Sender: TObject);
procedure nnDisconnectClick(Sender: TObject);
procedure lvAvNetsSelectItem(Sender: TObject; Item: TListItem;
    Selected: Boolean);
procedure lvProfilesSelectItem(Sender: TObject; Item: TListItem;
    Selected: Boolean);
procedure bkChangeIPInfoClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure bkAddClick(Sender: TObject);
procedure bkChangeClick(Sender: TObject);
procedure bkAddXmlClick(Sender: TObject);
procedure bkSaveXMLClick(Sender: TObject);
procedure bkAdapterOptionsClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
// Кінець опису компонентів класу TfmMain
private
    {Визначення приватних типів}
public
    {Визначення публічних типів (зони бачення)}
    procedure ShowError(api:string; res:integer);
    procedure Wait(adapter:integer);
    procedure ShowAvNets(scan:integer);
    procedure ShowProfiles;
    function GetAuthStr(mode:integer):string;
    function GetCipherStr(mode:integer):string;
end;

var // змінні
    fmMain: TfmMain;

implementation // реалізація

uses // Бібліотеки які підключаються
    Unit2, Unit3, Unit4, Unit5, Unit6;

{$R *.dfm} // Ресурси модуля магістерського ПЗ

function HEX(b:byte):string;
begin
    result:=inttohex(b, 2);
end;

procedure TfmMain.FormCreate(Sender: TObject);
begin
    EnableLog('WIFIMAN-LOG.txt', 0);
end;

```

```

procedure TfmMain.FormShow(Sender: TObject);
var // змінні
    res, n:integer;
    buf:array[0..1024] of char;
begin
    bkRefresh.Click;
    EnumerateAdapters();
    EnumerateProfiles(0);
    n:=GetTmpProfileFromAdapter(0, 0);
    res:=GetTmpProfileOption(n, 'name', buf, sizeof(buf));
    res:=GetTmpProfileOption(n, 'SSIDConfig/SSID/name', buf, sizeof(buf));
    buf := 'bbbbbb';
    res:=n;
    res:=SetTmpProfileOption(n, 'SSIDConfig/SSID/name', buf);
    res:=SetTmpProfileOption(n, 'name', buf);

    if res > ERROR_OFFSET then exit;

    SaveTmpProfile(n, 'c:\d.txt');

    res:=GetTmpProfileOption(n, 'name', buf, sizeof(buf));

    SetTmpProfileToAdapter(n, 0);

    Showmessage(buf);

{**}
end;

procedure TfmMain.bkRefreshClick(Sender: TObject);
var // змінні
    res, i, n, ver, status:integer;
    buf:array[0..256] of char;
    s:string;
begin
    if cbAdapters.ItemIndex<0 then s:=''
    else s:=cbAdapters.Items[cbAdapters.ItemIndex];

    stBar.Panels[0].Text:='';
    stBar.Panels[1].Text:='';
    stBar.Panels[2].Text:='';
    cbAdapters.Clear;
    lvAvNets.Clear;
    lvProfiles.Clear;
    lbGuid.Caption:='-----';
    lbCurrent.Caption:='-----';
    lbCurSpeed.Caption:='-----';

    //плани роботи wifi
    if IsNativeWiFi()=0 then lbNative.Caption:='Hi'
    else lbNative.Caption:='Так';

    bkAdapterOptions.Visible:=IsNativeWiFi() = 0;

    //версія використаної бібліотеки роботи з wifi
    ver:=GetWIFIManagerVersion();
    lbVer.Caption:=inttostr(ver div 100)+'.'+inttostr((ver mod 100) div
10)+'.'+inttostr(ver mod 10);

    //перевірка статусу WLAN з'єднання
    status:=GetWIFIServiceStatus();
    if status=0 then
        begin
            //Повідомлення
            Showmessage('Windows WiFi сервіс не запущено, рестарт');
            SetWIFIServiceStatus(1); //старт сервісу
            end;
            status:=GetWIFIServiceStatus();

```

```

if status=0 then lbSvcStatus.Caption:='Нема початку'
else if status=1 then lbSvcStatus.Caption:='Початок'
else lbSvcStatus.Caption:=inttostr(status);

////////////////////////////////////
//список адаптерів
////////////////////////////////////

res:=EnumerateAdapters();
if res > ERROR_OFFSET then
begin
ShowError('EnumerateAdapters', res);
exit;
end;

if res=1 then stBar.Panels[0].Text:='1 адаптер WI-FI знайдено'
else stBar.Panels[0].Text:=' '+inttostr(res)+' адаптерів WI-FI знайдено ';

for i:=0 to res-1 do
begin
n:=GetAdapterName(i, buf, sizeof(buf));
if n > ERROR_OFFSET then
begin
ShowError('Отримання ім'я адаптера', n);
exit;
end;
cbAdapters.Items.Add(buf);
end;

if s<>' ' then
begin
n:=cbAdapters.Items.IndexOf(s);
if n<>-1 then
begin
cbAdapters.ItemIndex:=n;
cbAdaptersClick(Self);
end;
end;
if (cbAdapters.ItemIndex<0) and (cbAdapters.Items.Count>0) then
begin
cbAdapters.ItemIndex:=0;
cbAdaptersClick(Self);
end;
end;

procedure TfmMain.ShowError(api:string; res: integer);
begin
Showmessage(api+' Error #'+inttohex(res, 8));
end;

procedure TfmMain.cbAdaptersClick(Sender: TObject);
var // змінні
buf:array[0..256] of char;
ad, res, dhcp:integer;
ip, mask, gate:array[0..256] of char;
d1, d2, d3, d4, d5, d6:byte;
begin
ad:=cbAdapters.ItemIndex;
if ad<0 then exit;

Wait(ad);

res:=GetAdapterGUID(ad, buf, sizeof(buf));
if res > ERROR_OFFSET then
begin
lbGuid.Caption:='-----';
ShowError('Отримання значення адаптера GUID', res);
exit;
end;

```

```

lbGuid.Caption:=buf;

res:=GetCurrentNetworkName(ad, buf, sizeof(buf));
if res > ERROR_OFFSET then
begin
  lbCurrent.Caption:='-----';
  ShowError('Отримання поточного ім'я мережі', res);
  exit;
end;
if buf='' then
begin
  lbCurrent.Caption:=' [HEMA]';
  Label3.Visible:=false;
  lbCurSpeed.Visible:=false;
  lbCurMac.Visible:=false;
end
else
begin
  lbCurrent.Caption:=buf;
  Label3.Visible:=true;
  lbCurSpeed.Visible:=true;

  res:=GetCurrentNetworkMac(ad, d1, d2, d3, d4, d5, d6);
  if res>ERROR_OFFSET then lbCurMac.Caption:=''
  else lbCurMac.Caption:=HEX(d1)+HEX(d2)+HEX(d3)+HEX(d4)+HEX(d5)+HEX(d6);
  lbCurMac.Visible:=true;
end;

res:=GetCurrentNetworkSpeed(ad);
if res > ERROR_OFFSET then
begin
  lbCurSpeed.Caption:='-----';
  ShowError('Отримання поточної швидкості мережі', res);
  exit;
end;
lbCurSpeed.Caption:=inttostr(res)+' Bits/s';

res:=GetCurrentNetworkChannel(ad);
if res > ERROR_OFFSET then
begin
  Label8.Visible:=false;
  lbChannel.Visible:=false;
end
else
begin
  lbChannel.Caption:=inttostr(res);
  Label8.Visible:=true;
  lbChannel.Visible:=true;
end;

//отримати дані IP
res:=GetAdapterIPInfo(ad, ip, mask, gate, dhcp, 256);
if res > ERROR_OFFSET then
begin
  lbIP.Caption:='-----';
  lbMask.Caption:='-----';
  lbGate.Caption:='-----';
  lbDHCP.Caption:='-----';
  ShowError('GetAdapterIPInfo', res);
end
else
begin
  lbIP.Caption:=ip;
  lbMask.Caption:=mask;
  lbGate.Caption:=gate;
  if dhcp<>0 then lbDHCP.Caption:='Так'
  else lbDHCP.Caption:='HI';
end;

```

```

    ShowAvNets(0);
    ShowProfiles;
end;

procedure TfmMain.Wait(adapter:integer);
begin
    CheckAdapterBusyStatus(adapter, 7000);
end;

procedure TfmMain.nnExitClick(Sender: TObject);
begin
    Close;
end;

procedure TfmMain.nnRefreshClick(Sender: TObject);
begin
    bkRefresh.Click;
end;

procedure TfmMain.nnEnableLogClick(Sender: TObject);
begin
    //Дозвіл вести лог файл
    EnableLog('WIFIMAN-LOG.txt', 0); // Лог файл
end;

procedure TfmMain.nnDisableLogClick(Sender: TObject);
begin
    //функція повертає ERROR_DEMOSVERSION_NOTSUPPORTED
    version
    DisableLog();
end;

function ExecuteFile(Wnd:hwnd; const FileName, Params, DefaultDir:string;
ShowCmd:Integer; Explore:boolean=false):THandle;
var // змінні
    zFileName, zParams, zDir:array[0..256] of Char;
    com:string;
begin
    result:=0;
    if not Explore then com:='Відкрити' else com:='Пошук';
    try
        Result:=ShellExecute(0, pchar(com),
            StrPCopy(zFileName, FileName), StrPCopy(zParams, Params),
            StrPCopy(zDir, DefaultDir), ShowCmd);
    except
        end;
end;

procedure TfmMain.nnHelpClick(Sender: TObject);
begin
    Form6.show;
    fmMain.hide;
end;

procedure TfmMain.ShowAvNets(scan: integer);
var // змінні
    ad, res, n, i:integer;
    buf:array[0..256] of char;
    it:tlistitem;
    s:string;
    d1, d2, d3, d4, d5, d6:byte;
begin
    lvAvNets.Clear;
    lvAvNets.SelectItem(Self, nil, false);
    stBar.Panels[1].Text:='';
    Repaint;
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;

```

```

lvAvNets.Color:=clBtnFace;
Repaint;
Wait(ad);

res:=EnumerateAvailableNetworks(ad, scan);
if res>ERROR_OFFSET then
begin
lvAvNets.Color:=clWindow;
ShowError('EnumerateAvailableNetworks', res);
exit;
end;

if res=1 then stBar.Panels[1].Text:=' 1 адаптер WI-FI знайдено '
else stBar.Panels[1].Text:=' '+inttostr(res)+' адаптерів WI-FI знайдено.';

for i:=0 to res-1 do
begin
n:=GetAvailableNetworkName(ad, i, buf, sizeof(buf));
if n > ERROR_OFFSET then
begin
lvAvNets.Color:=clWindow;
ShowError('GetAvailableNetworkName', n);
exit;
end;
it:=lvAvNets.Items.Add;
it.Caption:=buf;
////
n:=GetAvailableNetworkMAC(ad, i, d1, d2, d3, d4, d5, d6);
if n > ERROR_OFFSET then
begin
lvAvNets.Color:=clWindow;
ShowError('GetAvailableNetworkMAC', n);
exit;
end;
it.SubItems.Add(HEX(d1)+HEX(d2)+HEX(d3)+HEX(d4)+HEX(d5)+HEX(d6));

n:=GetAvailableNetworkType(ad, i);
if n > ERROR_OFFSET then
begin
lvAvNets.Color:=clWindow;
ShowError('GetAvailableNetworkType', n);
exit;
end;
case n of
dot11_BSS_type_infrastructure: it.SubItems.Add('Infrastructure');
dot11_BSS_type_independent: it.SubItems.Add('ad hoc');
dot11_BSS_type_any: it.SubItems.Add('');
else it.SubItems.Add('Невідомо ('+inttostr(n)+'')');
end;
////
n:=GetAvailableNetworkSignalQuality(ad, i);
if n > ERROR_OFFSET then
begin
lvAvNets.Color:=clWindow;
ShowError('GetAvailableNetworkSignalQuality', n);
exit;
end;
it.SubItems.Add(inttostr(n)+'%');
////
n:=IsAvailableNetworkSecure(ad, i);
if n > ERROR_OFFSET then
begin
lvAvNets.Color:=clWindow;
ShowError('IsAvailableNetworkSecure', n);
exit;
end;
if n=0 then it.SubItems.Add('Hi')
else it.SubItems.Add('Tak');
////

```

```

n:=GetAvailableNetworkAuthMode(ad, i);
if n > ERROR_OFFSET then
begin
  lvAvNets.Color:=clWindow;
  ShowError('GetAvailableNetworkAuthMode', n);
  exit;
end;
s:=GetAuthStr(n);
it.SubItems.Add(s);
////
n:=GetAvailableNetworkCipherMode(ad, i);
if n > ERROR_OFFSET then
begin
  lvAvNets.Color:=clWindow;
  ShowError('GetAvailableNetworkCipherMode', n);
  exit;
end;
s:=GetCipherStr(n);
it.SubItems.Add(s);
end;
lvAvNets.Color:=clWindow;
end;

procedure TfmMain.ShowProfiles;
var // зміни
  ad, res, i, n:integer;
  buf:array[0..256] of char;
  it:tlistitem;
begin
  lvProfiles.Clear;
  lvProfiles.SelectItem(Self, nil, false);
  stBar.Panels[2].Text:='';
  Repaint;
  ad:=cbAdapters.ItemIndex;
  if ad<0 then exit;

  lvProfiles.Color:=clBtnFace;
  Repaint;
  Wait(ad);

  res:=EnumerateProfiles(ad);
  if res>ERROR_OFFSET then
  begin
    lvProfiles.Color:=clWindow;
    ShowError('EnumerateAvailableNetworks', res);
    exit;
  end;

  if res=1 then stBar.Panels[2].Text:=' 1 профайл знайдено'
  else stBar.Panels[2].Text:=' '+inttostr(res)+' профайлів знадено';

  for i:=0 to res-1 do
  begin
    n:=GetProfileName(ad, i, buf, sizeof(buf));
    if n > ERROR_OFFSET then
    begin
      lvProfiles.Color:=clWindow;
      ShowError('GetProfileName', n);
      exit;
    end;
    it:=lvProfiles.Items.Add;
    it.Caption:=buf;
  end;
  lvProfiles.Color:=clWindow;
end;

procedure TfmMain.bkScanClick(Sender: TObject);
begin
  bkScan.Enabled:=false;

```

```

    ShowAvNets(1);
    bkScan.Enabled:=true;
end;

function TfmMain.GetAuthStr(mode: integer): string;
begin
    case mode of
        Ndis802_11AuthModeOpen: result:='Відкрити';
        Ndis802_11AuthModeAutoSwitch: result:='Автопереключення';
        Ndis802_11AuthModeWPA: result:='WPA';
        Ndis802_11AuthModeWPAPSK: result:='WPAPSK';
        Ndis802_11AuthModeWPANone: result:='WPANone';
        Ndis802_11AuthModeWPA2: result:='WPA2';
        Ndis802_11AuthModeWPA2PSK: result:='WPA2PSK';
        else result:='Невідомо ('+inttostr(mode)+' )';
    end;
end;

function TfmMain.GetCipherStr(mode: integer): string;
begin
    case mode of
        DOT11_CIPHER_ALGO_NONE: result:='Немає';
        DOT11_CIPHER_ALGO_WEP40: result:='WEP 40 біт';
        DOT11_CIPHER_ALGO_TKIP: result:='TKIP';
        DOT11_CIPHER_ALGO_CCMP: result:='AES';
        DOT11_CIPHER_ALGO_WEP104: result:='WEP 104 біт';
        DOT11_CIPHER_ALGO_WPA_USE_GROUP: result:='WPA група';
        DOT11_CIPHER_ALGO_WEP: result:='WEP';
        else result:='Невідомо ('+inttostr(mode)+' )';
    end;
end;

procedure TfmMain.bkConnectClick(Sender: TObject);
var // змінні
    ad, n, res, i, m:integer;
    s:string;
begin
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;

    n:=lvAvNets.ItemIndex;
    if n<0 then
        begin
            ShowMessage('Будь ласка введіть ім'я бездротової мережі!');
            exit;
        end;

    s:=lvAvNets.Items[n].Caption;
    m:=-1;
    for i:=0 to lvProfiles.Items.Count-1 do
        if lvProfiles.Items[i].Caption=s then
            begin
                m:=i;
                break;
            end;
    if m=-1 then
        begin
            if MessageDlg('Неможли провести з'єднання через відсутності заповненого про
            файлу. Створити профайл?', mtConfirmation, [mbYes, mbNo], 0)<>mrYes then exit;
                bkAdd.Click;
                exit;
            end;
        bkConnect.Enabled:=false;
        Wait(ad);

        res:=ConnectToNetwork(ad, pchar(s));
        if res>ERROR_OFFSET then
            begin
                bkConnect.Enabled:=true;
            end;
        end;
end;

```

```

        ShowError('Підключення до мережі', res);
        exit;
    end;

    Wait(ad);
    bkConnect.Enabled:=true;
    bkRefresh.Click;
end;

procedure TfmMain.bkDisconnectClick(Sender: TObject);
var // змінні
    ad, res:integer;
begin
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;

    bkDisconnect.Enabled:=false;
    Wait(ad);
    res:=DisconnectFromNetwork(ad);

    if res>ERROR_OFFSET then
        begin
            bkDisconnect.Enabled:=true;
            ShowError('Відключення від мережі', res);
            exit;
        end;

    Wait(ad);
    bkDisconnect.Enabled:=true;
    bkRefresh.Click;
end;

procedure TfmMain.bkManagementClick(Sender: TObject);
var // змінні
    ad, res:integer;
begin
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;

    res:=OpenProfilesUI(ad);
    if res>ERROR_OFFSET then
        begin
            bkDisconnect.Enabled:=true;
            ShowError('OpenProfilesUI', res);
            exit;
        end;
end;

procedure TfmMain.sbUpClick(Sender: TObject);
var // змінні
    ad, n, res:integer;
begin
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;
    n:=lvProfiles.ItemIndex;
    if n<0 then
        begin
            ShowMessage('Будь ласка вкажіть профайл!');
            exit;
        end;
    if n=0 then exit; //вихід

    sbUp.Enabled:=false;
    Wait(ad);

    res:=MoveProfile(ad, n, n-1);
    if res>ERROR_OFFSET then
        begin
            sbUp.Enabled:=true;

```

```

        ShowError('MoveProfile', res);
        exit;
    end;

    Wait(ad);
    sbUp.Enabled:=true;
    bkRefresh.Click;
end;

procedure TfmMain.sbDownClick(Sender: TObject);
var // змінні
    ad, n, res:integer;
begin
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;
    n:=lvProfiles.ItemIndex;
    if n<0 then
        begin
            ShowMessage(' Будь ласка вкажіть профайл!');
            exit;
        end;
    if n=lvProfiles.Items.Count-1 then exit; //верхня частина

    sbDown.Enabled:=false;
    Wait(ad);

    res:=MoveProfile(ad, n, n+1);
    if res>ERROR_OFFSET then
        begin
            sbDown.Enabled:=true;
            ShowError('MoveProfile', res);
            exit;
        end;

    Wait(ad);
    sbDown.Enabled:=true;
    bkRefresh.Click;
end;

procedure TfmMain.bkRemoveClick(Sender: TObject);
var // змінні
    ad, n, res:integer;
begin
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;
    n:=lvProfiles.ItemIndex;
    if n<0 then
        begin
            ShowMessage(' Будь ласка вкажіть профайл!');
            exit;
        end;

    bkRemove.Enabled:=false;
    Wait(ad);

    res:=DeleteProfile(ad, n);
    if res>ERROR_OFFSET then
        begin
            bkRemove.Enabled:=true;
            ShowError('Профайл видалено', res);
            exit;
        end;

    Wait(ad);
    bkRemove.Enabled:=true;
    bkRefresh.Click;
end;

```

```

procedure TfmMain.pmAvNetPopup(Sender: TObject);
var // зміни
  n:integer;
begin
  n:=lvAvNets.ItemIndex;
  if n<0 then
    begin
      nnConnect.Enabled:=false;
      nnDisconnect.Enabled:=false;
      exit;
    end;
  if lvAvNets.Items[n].Caption=lbCurrent.Caption then
    begin
      nnConnect.Enabled:=false;
      nnDisconnect.Enabled:=true;
    end
  else
    begin
      nnConnect.Enabled:=true;
      nnDisconnect.Enabled:=false;
    end;
end;

procedure TfmMain.nnConnectClick(Sender: TObject);
begin
  bkConnect.Click;
end;

procedure TfmMain.nnDisconnectClick(Sender: TObject);
begin
  bkDisconnect.Click;
end;

procedure TfmMain.lvAvNetsSelectItem(Sender: TObject; Item: TListItem; Selected:
Boolean);
var // зміни
  n:integer;
begin
  n:=lvAvNets.ItemIndex;
  if n<0 then
    begin
      bkConnect.Enabled:=false;
      bkDisconnect.Enabled:=false;
      exit;
    end;
  if lvAvNets.Items[n].Caption=lbCurrent.Caption then
    begin
      bkConnect.Enabled:=false;
      bkDisconnect.Enabled:=true;
    end
  else
    begin
      bkConnect.Enabled:=true;
      bkDisconnect.Enabled:=false;
    end;
end;

procedure TfmMain.lvProfilesSelectItem(Sender: TObject; Item: TListItem;
Selected: Boolean);
var // зміни
  n:integer;
begin
  n:=lvProfiles.ItemIndex;
  sbUp.Enabled:=n>0;
  sbDown.Enabled:=(n<>-1) and (n<lvProfiles.Items.Count-1);
  bkRemove.Enabled:=n<>-1;
  bkChange.Enabled:=n<>-1;
  bkSaveXML.Enabled:=n<>-1;
end;

```

```

procedure TfmMain.bkChangeIPInfoClick(Sender: TObject);
var // змінні
    ad:integer;
begin
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;

    fmChangeIP.AdInd:=ad;
    fmChangeIP.ShowModal;
    bkRefresh.Click;
end;

procedure TfmMain.bkAddClick(Sender: TObject);
var // змінні
    ad:integer;
begin
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;

    fmProfProp.AdapterInd:=ad;
    fmProfProp.ProfInd:=-1;
    fmProfProp.ShowModal;
end;

procedure TfmMain.bkChangeClick(Sender: TObject);
var // змінні
    ad:integer;
begin
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;

    fmProfProp.AdapterInd:=ad;
    fmProfProp.ProfInd:=lvProfiles.ItemIndex;
    fmProfProp.ShowModal;
end;

procedure TfmMain.bkAddXmlClick(Sender: TObject);
var // змінні
    ProfHandle, res, ad:integer;
begin
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;

    if not opXML.Execute then exit;
    ProfHandle:=LoadTmpProfile(pchar(opXML.FileName));
    if ProfHandle>ERROR_OFFSET then
        begin
            ShowError('LoadTmpProfile', ProfHandle);
            exit;
        end;
    res:=SetTmpProfileToAdapter(ProfHandle, ad);
    if res>ERROR_OFFSET then Showmessage('Неможливо використати цей профал, помилка #'+inttohex(res, 8)+''. Будь ласка перегляньте налаштування профайлу');
    FreeTmpProfile(ProfHandle);
    bkRefresh.Click;
end;

procedure TfmMain.bkSaveXMLClick(Sender: TObject);
var // змінні
    ProfHandle, res, ad, n:integer;
begin
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;
    n:=lvProfiles.ItemIndex;
    if n<0 then
        begin
            ShowMessage('Будь ласка вкажіть профайл!');
        end;
    end;
end;

```

```
        exit;
    end;

    if not sdXML.Execute then exit;

    ProfHandle:=GetTmpProfileFromAdapter(ad, n);
    if ProfHandle>ERROR_OFFSET then
        begin
            ShowError('GetTmpProfileFromAdapter', ProfHandle);
            exit;
        end;
    res:=SaveTmpProfile(ProfHandle, pchar(sdXML.FileName));
    if res>ERROR_OFFSET then ShowError('SaveTmpProfile', res);
    FreeTmpProfile(ProfHandle);
end;

procedure TfmMain.bkAdapterOptionsClick(Sender: TObject);
var // змінні
    ad:integer;
begin
    // Параметри адаптера доступна тільки для WinXP,
    // Vista цих параметрів немає
    ad:=cbAdapters.ItemIndex;
    if ad<0 then exit;
    fmAdapterOpt.AdInd:=ad;
    fmAdapterOpt.ShowModal;
end;

procedure TfmMain.Button1Click(Sender: TObject);
begin
    fmPing.ShowModal;
end;

end.
```

Кафедра \_ КБПЗ \_ 2023 рік

## Файл Unit2.pas

```

unit Unit2; //Назва модулю

interface // об'ява даних

uses // Бібліотеки які підключаються
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, WiFiMan;

type //Об'ява власних типів даних
  TfmChangeIP = class(TForm) // об'ява класу
    // Початок опису компонентів класу TfmChangeIP
    bkOK: TButton; // кнопка
    bkCancel: TButton; // кнопка
    Label1: TLabel; // Мітка
    edIP: TEdit; // Поле введення даних;
    Label2: TLabel; // Мітка 2
    edMask: TEdit; // Поле введення даних;
    Label3: TLabel; // Мітка 3
    edGate: TEdit; // Поле введення даних;
    cbDHCP: TCheckBox;
    procedure bkCancelClick(Sender: TObject);
    procedure bkOKClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure FormKeyPress(Sender: TObject; var Key: Char);
    // Кінець опису компонентів класу TfmChangeIP
  private
    {Визначення приватних типів}
  public
    {Визначення публічних типів (зони бачення)}
    AdInd:integer; //поточне число адаптера
  end;

var // змінні
  fmChangeIP: TfmChangeIP;

implementation // реалізація

uses // Бібліотеки які підключаються
  Unit1;

{$R *.dfm} // Ресурси модуля магістерського ПЗ

procedure TfmChangeIP.bkCancelClick(Sender: TObject);
begin
  close;
end;

procedure TfmChangeIP.bkOKClick(Sender: TObject);
var // змінні
  res:integer;
begin
  res:=SetAdapterIPInfo(adInd, pchar(edIP.Text), pchar(edMask.Text),
  pchar(edGate.Text), ord(cbDHCP.Checked));
  if res > ERROR_OFFSET then
    fmMain.ShowError('SetAdapterIPInfo', res);
  close;
end;

procedure TfmChangeIP.FormShow(Sender: TObject);
var // змінні
  dhcp, res:integer;
  ip, mask, gate:array[0..256] of char;
begin
  //отримання даних по IP адресі
  res:=GetAdapterIPInfo(adInd, ip, mask, gate, dhcp, 256);

```

```
if res > ERROR_OFFSET then
begin
  fmMain.ShowError('GetAdapterIPInfo', res);
end
else
begin
  edIP.Text:=ip;
  edMask.Text:=mask;
  edGate.Text:=gate;
  cbDHCP.Checked:=dhcp<>0;
end;
end;

procedure TfmChangeIP.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if key=#27 then close;
end;

end.
```

Кафедра \_ КБПЗ \_ 2023рік

## Файл Unit3.pas

```

unit Unit3; //Назва модулю

interface // об'ява даних

uses // Бібліотеки які підключаються
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, WiFiMan;

type //Об'ява власних типів даних
  TfmProfProp = class(TForm) // об'ява класу
    // Початок опису компонентів класу TfmProfProp
    bkOK: TButton; // кнопка
    bkCancel: TButton; // кнопка
    Label1: TLabel; // Мітка 1
    edSSID: TEdit; // Поле введення даних;
    GroupBox1: TGroupBox;
    Label2: TLabel; // Мітка 2
    Label3: TLabel; // Мітка 3
    Label4: TLabel; // Мітка 4
    Label5: TLabel; // Мітка 5
    Label7: TLabel; // Мітка 7
    cbAuth: TComboBox; // Кнопки вибору
    cbEnc: TComboBox; // Кнопки вибору
    edKey: TEdit; // Поле введення даних
    edInd: TEdit; // Поле введення даних;
    GroupBox2: TGroupBox;
    Label8: TLabel; // Мітка 8
    cbAutoConnect: TCheckBox;
    cbHOC: TCheckBox;
    Label9: TLabel; // Мітка 9
    lbProt: TLabel; // Мітка
    procedure bkCancelClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure bkOKClick(Sender: TObject);
    procedure FormKeyPress(Sender: TObject; var Key: Char);
    procedure edKeyChange(Sender: TObject);
    procedure cbAuthClick(Sender: TObject);
    // Кінець опису компонентів класу TfmProfProp
  private
    {Визначення приватних типів}
    KeyProtected:boolean;
    KeyChanged:boolean;
  public
    {Визначення публічних типів (зони бачення)}
    AdapterInd:integer;
    ProfInd:integer;
    ProfHandle:integer;
  end;

var // змінні
  fmProfProp: TfmProfProp;

implementation // реалізація

uses // Бібліотеки які підключаються
  Unit1;

{$R *.dfm} // Ресурси модуля магістерського ПЗ

procedure TfmProfProp.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if key=#27 then close;
end;

procedure TfmProfProp.bkCancelClick(Sender: TObject);
begin

```

```

    close;
end;

procedure TfmProfProp.edKeyChange(Sender: TObject);
begin
    KeyChanged:=true;
end;

procedure TfmProfProp.FormShow(Sender: TObject);
var // змінні
    res:integer;
    buf:array[0..1024] of char;
    s:string;
begin
    KeyChanged:=false;
    if ProfInd=-1 then //новий профайл wifi мережі
        begin
            cbHOC.Checked:=false;
            edSSID.Text:='';
            edssid.Enabled:=true;
            edssid.Color:=clwindow;
            cbAuth.ItemIndex:=0;
            cbEnc.ItemIndex:=0;
            edKey.Text:='';
            edInd.Text:='1';
            cbAutoConnect.Checked:=false;
            KeyProtected:=false;
            lbProt.Visible:=KeyProtected;
            edKey.Enabled:=not KeyProtected;
        end
    else
        begin
            // створення тимчасового профілю з профілю реального адаптеру
            ProfHandle:=GetTmpProfileFromAdapter(AdapterInd, ProfInd);
            if ProfHandle>ERROR_OFFSET then
                begin
                    fmMain.ShowError('GetTmpProfileFromAdapter', ProfHandle);
                    exit;
                end;
            //ssid
            res:=GetTmpProfileOption(ProfHandle, '.ssid', buf, sizeof(buf));

            if res>ERROR_OFFSET then
                begin
                    fmMain.ShowError('GetTmpProfileOption - ssid', res);
                    exit;
                end;
            edSSID.Text:=buf;
            edssid.Enabled:=false;
            edssid.Color:=clbtnface;
            //початок Автентифікації
            GetTmpProfileOption(ProfHandle, '.Authentication', buf, sizeof(buf));
            s:=buf;
            if comparetext(buf, 'open')=0 then cbAuth.ItemIndex:=0;
            if comparetext(buf, 'shared')=0 then cbAuth.ItemIndex:=1;
            if comparetext(buf, 'WPA')=0 then cbAuth.ItemIndex:=2;
            if comparetext(buf, 'WPAPSK')=0 then cbAuth.ItemIndex:=3;
            if comparetext(buf, 'WPA2')=0 then cbAuth.ItemIndex:=4;
            if comparetext(buf, 'WPA2PSK')=0 then cbAuth.ItemIndex:=5;
            //цінець Автентифікації
            GetTmpProfileOption(ProfHandle, '.Encryption', buf, sizeof(buf));
            s:=buf;
            if comparetext(buf, 'none')=0 then cbEnc.ItemIndex:=0;
            if comparetext(buf, 'WEP')=0 then cbEnc.ItemIndex:=1;
            if comparetext(buf, 'TKIP')=0 then cbEnc.ItemIndex:=2;
            if comparetext(buf, 'AES')=0 then cbEnc.ItemIndex:=3;
            //флаг та ключ захисту
            res:=GetTmpProfileOption(ProfHandle, '.IsKeyProtected', buf, sizeof(buf));
            if res>ERROR_OFFSET then KeyProtected:=false

```

```

else KeyProtected:=(comparetext(buf, 'true')=0) or (cbEnc.ItemIndex>1);
lbProt.Visible:=KeyProtected;
//ключ
res:=GetTmpProfileOption(ProfHandle, '.KeyMaterial', buf, sizeof(buf));
if (res>ERROR_OFFSET) or KeyProtected then edKey.Text:=''
//створення пустого ключа
else edKey.Text:=buf;
//індекс ключа
res:=GetTmpProfileOption(ProfHandle, '.KeyIndex', buf, sizeof(buf));
if res>ERROR_OFFSET then edInd.Text:='1'
else edInd.Text:=inttostr(strtointdef(buf, 0)+1);
//автоматичне підключення
res:=GetTmpProfileOption(ProfHandle, '.ConnectionMode', buf, sizeof(buf));
if res>ERROR_OFFSET then cbAutoConnect.Checked:=true
else cbAutoConnect.Checked:=comparetext(buf, 'auto')=0;
//тип мережі
res:=GetTmpProfileOption(ProfHandle, '.ConnectionType', buf, sizeof(buf));
if res>ERROR_OFFSET then
begin
fmMain.ShowError('GetTmpProfileOption - ConnectionType', res);
exit;
end;
cbHOC.Checked:=comparetext(buf, 'IBSS')=0;

FreeTmpProfile(ProfHandle); //профіль
end;
KeyChanged:=false;
cbAuthClick(nil);
end;

procedure TfmProfProp.bkOKClick(Sender: TObject);
var // змінні
n, res:integer;
begin
if edssid.Text='' then
begin
Showmessage('Будь ласка введіть ssid!');
exit;
end;

if ProfInd=-1 then //новий профіль
begin
ProfHandle:=CreateTmpProfile(0); //створення порожнього профайлу
end
else //модифікація профайлу
begin
ProfHandle:=GetTmpProfileFromAdapter(AdapterInd, ProfInd);
end;

if ProfInd=-1 then
begin
SetTmpProfileOption(ProfHandle, '.name', pchar(edssid.Text));
SetTmpProfileOption(ProfHandle, '.ssid', pchar(edssid.Text));
end;

if cbHOC.Checked then SetTmpProfileOption(ProfHandle, '.ConnectionType',
'IBSS')
else SetTmpProfileOption(ProfHandle, '.ConnectionType', 'ESS');

if cbAutoConnect.Checked then SetTmpProfileOption(ProfHandle,
'.ConnectionMode', 'auto')
else SetTmpProfileOption(ProfHandle, '.ConnectionMode', 'manual');

SetTmpProfileOption(ProfHandle, '.Authentication',
pchar(cbAuth.Items[cbAuth.ItemIndex]));

SetTmpProfileOption(ProfHandle, '.Encryption',
pchar(cbEnc.Items[cbEnc.ItemIndex]));

```

```

    if (KeyChanged or (ProfInd = -1)) and ((edKey.Text<>'') and not KeyProtected)
then
    begin
        if cbEnc.ItemIndex<2 then
            SetTmpProfileOption(ProfHandle, '.KeyType', 'networkKey')
        else //WPA
            SetTmpProfileOption(ProfHandle, '.KeyType', 'passPhrase');

        SetTmpProfileOption(ProfHandle, '.IsKeyProtected', 'false');

        SetTmpProfileOption(ProfHandle, '.KeyMaterial', pchar(edKey.Text));

        n:=strtointdef(edInd.Text, 1)-1;
        SetTmpProfileOption(ProfHandle, '.KeyIndex', pchar(inttostr(n)));
    end;

    SaveTmpProfile(ProfHandle, 'c:\e.txt'); //дублювання у файл

    res:=SetTmpProfileToAdapter(ProfHandle, AdapterInd);
    FreeTmpProfile(ProfHandle); //порожній профайл
    if res>ERROR_OFFSET then
        begin
            Showmessage(' Неможливо використати цей профал, помилка #' +inttohex(res, 8)+''.
            Будь ласка перегляньте налаштування профайлу. ');
            exit;
        end;

        close;
        fmmain.bkRefresh.Click;
end;

procedure TfmProfProp.cbAuthClick(Sender: TObject);
begin
    //
end;

end.

```

## Файл Unit4.pas

```

unit Unit4; //Назва модулю

interface // об'ява даних

uses // Бібліотеки які підключаються
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, WiFiMan;

type //Об'ява власних типів даних
  TfmAdapterOpt = class(TForm) // об'ява класу
    // Початок опису компонентів класу TfmAdapterOpt
    Panel1: TPanel;
    Label4: TLabel; // Мітка 4
    rb13: TRadioButton;
    rb12: TRadioButton;
    rb11: TRadioButton;
    Panel2: TPanel; // Панель
    Label5: TLabel; // Мітка 5
    rb21: TRadioButton;
    rb22: TRadioButton;
    Panel3: TPanel;
    Label6: TLabel; // Мітка 6
    rb32: TRadioButton;
    rb31: TRadioButton;
    bkOK: TButton; // кнопка
    bkCancel: TButton; // кнопка
    procedure bkCancelClick(Sender: TObject);
    procedure FormKeyPress(Sender: TObject; var Key: Char);
    procedure bkOKClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    // Кінець компонентів класу TfmAdapterOpt
  private
    {Визначення приватних типів}
  public
    {Визначення публічних типів (зони бачення)}
    AdInd:integer;
  end;

var // змінні
  fmAdapterOpt: TfmAdapterOpt;

implementation // реалізація

uses // Бібліотеки які підключаються
  Unit1;

{$R *.dfm} // Ресурси модуля магістерського ПЗ

procedure TfmAdapterOpt.bkCancelClick(Sender: TObject);
begin
  close;
end;

procedure TfmAdapterOpt.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if key=#27 then close;
end;

procedure TfmAdapterOpt.FormShow(Sender: TObject);
var // змінні
  allopt, n, w, c:integer;
begin
  //відразу встановлення всіх значень
  allopt:=GetAdapterOption(AdInd, ADAPTER_NETTYPE or ADAPTER_USEWINDOWS or
  ADAPTER_CONNECTFLAG);
  //окремі варіанти

```

```

n:=allopt and ADAPTER_NETTYPE;
w:=allopt and ADAPTER_USEWINDOWS;
c:=allopt and ADAPTER_CONNECTFLAG;
case n of // вибір
  NETTYPE_ANY: rb11.Checked:=true;
  NETTYPE_AONLY: rb12.Checked:=true;
  NETTYPE_ADHOCONLY: rb13.Checked:=true;
end;

if (w and USEWINDOWS_FLAG)<>0 then rb21.Checked:=true
  else rb22.Checked:=true;

if (c and CONNECTFLAG_FLAG)<>0 then rb31.Checked:=true
  else rb32.Checked:=true;

end;

procedure TfmAdapterOpt.bkOKClick(Sender: TObject);
var // змінні
  res, allopt:integer;
begin
  allopt:=0;

  if rb11.Checked then allopt:=allopt or NETTYPE_ANY
  else if rb12.Checked then allopt:=allopt or NETTYPE_AONLY
  else allopt:=allopt or NETTYPE_ADHOCONLY;

  if rb21.Checked then allopt:=allopt or USEWINDOWS_FLAG;
  if rb31.Checked then allopt:=allopt or CONNECTFLAG_FLAG;

  res:=SetAdapterOption(AdInd, ADAPTER_NETTYPE or ADAPTER_USEWINDOWS or
ADAPTER_CONNECTFLAG, allopt);
  if res>ERROR_OFFSET then
    begin
      fmmain.ShowError('SetAdapterOption', res);
      exit;
    end;
  fmmain.bkRefresh.Click;
  close; // Вихід
end;

end.

```

## Файл Unit5.pas

```

unit Unit5; //Назва модулю

interface // об'ява даних

uses // Бібліотеки які підключаються
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, WiFiMan;

type //Об'ява власних типів даних
  TfmPing = class(TForm) // об'ява класу
    // Початок опису компонентів класу TfmPing
    Label1: TLabel; // Мітка 1
    edPing: TEdit; // Поле введення даних;
    bkPing: TButton; // кнопка
    lbRes: TLabel; // Мітка
    bkClose: TButton; // кнопка
    procedure bkCloseClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure edPingKeyPress(Sender: TObject; var Key: Char);
    procedure FormKeyPress(Sender: TObject; var Key: Char);
    procedure bkPingClick(Sender: TObject);
    // Кінець опису компонентів класу TfmPing
  private
    {Визначення приватних типів}
  public
    {Визначення публічних типів (зони бачення)}
  end;

var // змінні
  fmPing: TfmPing;

implementation // реалізація

{$R *.dfm} // Ресурси модуля магістерського ПЗ

procedure TfmPing.bkCloseClick(Sender: TObject);
begin
  close;
end;

procedure TfmPing.FormShow(Sender: TObject);
begin
  lbRes.Caption:='';
  edPing.SetFocus;
end;

procedure TfmPing.edPingKeyPress(Sender: TObject; var Key: Char);
begin
  if key=#13 then bkPing.Click;
end;

procedure TfmPing.FormKeyPress(Sender: TObject; var Key: Char);
begin
  if key=#27 then close;
end;

procedure TfmPing.bkPingClick(Sender: TObject);
var // змінні
  s:string;
  res:integer;
begin
  lbRes.Caption:='';
  lbRes.Repaint;
  edPing.SetFocus;
  s:=trim(edPing.Text);
  if s='' then

```

```
begin
  Showmessage('Вкажіть хост!');
  exit;
end;
res:=Ping(pchar(s), 3500); //3.5 сек запиту
if res>ERROR_OFFSET then
begin
  if res=ERROR_PINGTIMEOUT then lbRes.Caption:='Пингування не вдалося:
Timeout'
  else if res=ERROR_PINGFAIL then lbRes.Caption:='Ping не пройшов'
  else if res=ERROR_HOSTRESOLVE then lbRes.Caption:='Немає можливості
отримати IP хоста'
  else lbRes.Caption:='Помилка #' + inttohex(res, 8);
  exit;
end;
if res=0 then lbRes.Caption:='Ping пройшов, Час < 1мс'
else lbRes.Caption:='Ping OK, Time = '+inttostr(res)+'мс';
end;
end.
```

Кафедра \_ КБПЗ \_ 2023 рік

## Файл Unit6.pas

```
unit Unit6; //Назва модулю

interface // об'ява даних

uses // Бібліотеки які підключаються
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;

type //Об'ява власних типів даних
  TForm6 = class(TForm) // об'ява класу
  // Початок опису компонентів класу TForm6
    GroupBox1: TGroupBox;
    Button1: TButton; // кнопка
    Label1: TLabel; // Мітка 1
    Label2: TLabel; // Мітка 2
    Label3: TLabel; // Мітка 3
    procedure Button1Click(Sender: TObject);
  // Кінець опису компонентів класу TForm6
  private
    {Визначення приватних типів}
  public
    {Визначення публічних типів (зони бачення)}
  end;

var // змінні
  Form6: TForm6;

Implementation // реалізація

uses // Бібліотеки які підключаються
  Unit1, Unit2, Unit3, Unit4, Unit5;

{$R *.dfm} // Ресурси модуля магістерського ПЗ

procedure TForm6.Button1Click(Sender: TObject);
begin
  fmMain.show; // показати
  Form6.hide; // приховати
end;

end.
```

## Файл WiFiMan.pas

```

unit WiFiMan;
interface // об'ява даних
uses // Бібліотеки які підключаються
Windows, Classes;

// константи

// Коди помилок
Const // константи
  ERROR_OFFSET = $70000000;
  ERROR_INIT = $70000001;
  ERROR_APIENUM = $70000002;
  ERROR_INVALIDADAPTER = $70000003;
  ERROR_SMALLBUFFER = $70000004;
  ERROR_APIQUERY = $70000005;
  ERROR_ADAPTERBUSY = $70000006;
  ERROR_INVALIDNETWORK = $70000007;
  ERROR_INVALIDPROFILE = $70000008;
  ERROR_APISET = $70000009;
  ERROR_PROFILENOTFOUND = $7000000A;
  ERROR_INVALIDPARAMETER = $7000000B;
  ERROR_DATACHANGED = $7000000C;
  ERROR_FAIL = $7000000D; //невідомо помилка
  ERROR_ADAPTERNOTFOUND = $7000000E;
  ERROR_SCAN = $7000000F;
  ERROR_NOCURRENTNETWORK = $70000010;
  ERROR_NOTSUPPORTED = $70000011;
  ERROR_NOSERVICE = $70000012;
  ERROR_ACCESSDENIED = $70000013;
  ERROR_SERVICEDISABLED = $70000014;
  ERROR_PINGTIMEOUT = $70000015;
  ERROR_PINGFAIL = $70000016;
  ERROR_HOSTRESOLVE = $70000017;

  ERROR_TOOMANYPROFILES = $70000020;
  ERROR_CREATEXMLDOCUMENT = $70000021;
  ERROR_INVALIDTEMPLATE = $70000022;
  ERROR_LOADXML = $70000023;
  ERROR_SAVEXML = $70000024;
  ERROR_OPTIONNOTFOUND = $70000025;
  ERROR_INVALIDOPTIONVALUE = $70000026;
  ERROR_BADPROFILE = $70000027;

  ERROR_DEMVERSION_NOTSUPPORTED = $700000F0;
  ERROR_DEMVERSION_EXPIRED = $700000F1;

//константи
UNKNOWNVALUEOFFSET = $100; //зсув від невідомої помилки

//дані для створення нового профіля wifi
TEMPLATE_EMPTY = 0;
TEMPLATE_UNSECURE_OPEN = 1;
TEMPLATE_WEP_OPEN = 2;
TEMPLATE_UNSECURE_SHARED = 3;
TEMPLATE_WEP_SHARED = 4;

//_NDIS_802_11_AUTHENTICATION_MODE, for GetAvailableNetworkAuthMode
Ndis802_11AuthModeOpen = 0;
Ndis802_11AuthModeShared = 1;
Ndis802_11AuthModeAutoSwitch = 2;
Ndis802_11AuthModeWPA = 3;
Ndis802_11AuthModeWPAPSK = 4;
Ndis802_11AuthModeWPA_None = 5;
Ndis802_11AuthModeWPA2 = 6;
Ndis802_11AuthModeWPA2PSK = 7;

```

```

// DOT11_CIPHER_ALGORITHM, for GetAvailableNetworkCipherMode
DOT11_CIPHER_ALGO_NONE = $00;
DOT11_CIPHER_ALGO_WEP40 = $01;
DOT11_CIPHER_ALGO_TKIP = $02;
DOT11_CIPHER_ALGO_CCMP = $04;
DOT11_CIPHER_ALGO_WEP104 = $05;
DOT11_CIPHER_ALGO_WPA_USE_GROUP = $100;
DOT11_CIPHER_ALGO_WEP = $101;

//DOT11_BSS_TYPE, for GetAvailableNetworkType
dot11_BSS_type_infrastructure = 1;
dot11_BSS_type_independent = 2;
dot11_BSS_type_any = 3;

//Встановлення чи скид даних адаптера
ADAPTER_NETTYPE=$00000003; //тип доступу до мережі
ADAPTER_USEWINDOWS=$00008000;
ADAPTER_CONNECTFLAG = $00004000; //автоматичне підключення до
// бездротової мережі ( якщо до цього вже були підключення)

NETTYPE_ANY=$00000002; //будь яка мережа
NETTYPE_APONLY = $00000001; //тільки тип AP
NETTYPE_ADHOCONLY=$00000000; //тільки ПК до ПК мережа (ad-hoc)
USEWINDOWS_FLAG=$00008000; //використання налаштувань адаптера
// з Windows
CONNECTFLAG_FLAG = $00004000 ;//тип з'єднання

Const // константи
WiFiLib = 'WiFiMan.dll'; // використана бібліотека

// WiFi функції, які отримуються з бібліотеки WiFiMan.dll

function EnumerateAdapters:dword; stdcall; external WiFiLib;
function GetAdapterGUID(AdapterInd:integer; Guid:pchar; MaxCount:integer):dword;
stdcall; external WiFiLib;
function GetAdapterName(AdapterInd:integer; Name:pchar; MaxCount:integer):dword;
stdcall; external WiFiLib;
function CheckAdapterBusyStatus(AdapterInd:integer; Timeout:integer):dword;
stdcall; external WiFiLib;
function GetAdapterOption(AdapterInd:integer; OptionCode:integer):dword;
stdcall; external WiFiLib;
function SetAdapterOption(AdapterInd:integer; OptionCode:integer;
Value:integer):dword; stdcall; external WiFiLib;
function FreeAllResources():dword; stdcall; external WiFiLib;
function IsNativeWIFI():dword; stdcall; external WiFiLib;
function GetWIFIManagerVersion():dword; stdcall; external WiFiLib;
function GetWIFIServiceStatus():dword; stdcall; external WiFiLib;
function SetWIFIServiceStatus(StartService:integer):dword; stdcall;
external WiFiLib;
function GetAdapterIPInfo(AdapterInd:integer; IP:pchar; Mask:pchar;
Gateway:pchar; var DHCPEnabled:integer; MaxCount:integer):dword; stdcall;
external WiFiLib;
function SetAdapterIPInfo(AdapterInd:integer; IP:pchar; Mask:pchar;
Gateway:pchar; DHCPEnabled:integer):dword; stdcall; external WiFiLib;
function Ping(Host:pchar; Timeout:integer):dword; stdcall;
external WiFiLib;
function EnumerateAvailableNetworks(AdapterInd:integer; Search:integer):dword;
stdcall; external WiFiLib;
function GetAvailableNetworkName(AdapterInd:integer; NetworkInd:integer;
Name:pchar; MaxCount:integer):dword; stdcall; external WiFiLib;
function GetAvailableNetworkSignalQuality(AdapterInd:integer;
NetworkInd:integer):dword; stdcall; external WiFiLib;
function GetAvailableNetworkType(AdapterInd:integer; NetworkInd:integer):dword;
stdcall; external WiFiLib;
function IsAvailableNetworkSecure(AdapterInd:integer; NetworkInd:integer):dword;
stdcall; external WiFiLib;
function GetAvailableNetworkAuthMode(AdapterInd:integer;
NetworkInd:integer):dword; stdcall; external WiFiLib;

```

```

function GetAvailableNetworkCipherMode(AdapterInd:integer;
NetworkInd:integer):dword; stdcall; external WiFiLib;
function GetAvailableNetworkMac(AdapterInd:integer; NetworkInd:integer;
                                var d1, d2, d3, d4, d5, d6:byte)
                                :dword; stdcall; external WiFiLib;
function EnumerateProfiles(AdapterInd:integer):dword; stdcall; external
                                WiFiLib;
function GetProfileName(AdapterInd:integer; ProfileInd:integer; Name:pchar;
MaxCount:integer):dword; stdcall; external WiFiLib;

function GetCurrentNetworkName(AdapterInd:integer; Name:pchar;
MaxCount:integer):dword; stdcall; external WiFiLib;
function GetCurrentNetworkSpeed(AdapterInd:integer):dword; stdcall;
                                external WiFiLib;
function GetCurrentNetworkChannel(AdapterInd:integer):dword; stdcall;
                                external WiFiLib;
function GetCurrentNetworkMac(AdapterInd:integer; var d1, d2, d3, d4, d5,
d6:byte):dword; stdcall; external WiFiLib;

function ConnectToNetwork(AdapterInd:integer; Name:pchar):dword; stdcall;
                                external WiFiLib;
function DisconnectFromNetwork(AdapterInd:integer):dword; stdcall;
                                external WiFiLib;

function MoveProfile(AdapterInd:integer; ProfileInd:integer;
NewIndex:integer):dword; stdcall; external WiFiLib;
function DeleteProfile(AdapterInd:integer; ProfileInd:integer):dword;
                                stdcall; external WiFiLib;
function OpenProfilesUI(AdapterInd:integer):dword; stdcall; external
                                WiFiLib;
function EnableLog(Path:pchar; RemoveOld:dword):dword; stdcall; external
                                WiFiLib;
function DisableLog():dword; stdcall; external WiFiLib;

function CreateTmpProfile(Template:integer):dword; stdcall; external
                                WiFiLib;
function LoadTmpProfile(FileName:pchar):dword; stdcall; external WiFiLib;
function SaveTmpProfile(ProfileHandle:integer; FileName:pchar):dword;
                                stdcall; external WiFiLib;
function CloneTmpProfile(ProfileHandle:integer):dword; stdcall;
                                external WiFiLib;
function GetTmpProfileOption(ProfileHandle:integer; OptionName:pchar;
Value:pchar; MaxCount:integer):dword; stdcall; external WiFiLib;
function SetTmpProfileOption(ProfileHandle:integer; OptionName:pchar;
Value:pchar):dword; stdcall; external WiFiLib;
function FreeTmpProfile(ProfileHandle:integer):dword; stdcall;
                                external WiFiLib;
function GetTmpProfileFromAdapter(AdapterInd:integer; ProfileInd:integer):dword;
stdcall; external WiFiLib;
function SetTmpProfileToAdapter(ProfileHandle:integer;
AdapterInd:integer):dword; stdcall; external WiFiLib;
end.

```

## Файл налаштувань компілятора Wifi.dof

```
[FileVersion]
Version=7.0
[Compiler]

A=8
B=0
C=1
D=1
E=0
F=0
G=1
H=1
I=1
J=0
K=0
L=1
M=0
N=1
O=1
P=1
Q=0
R=0
S=0
T=0
U=0
V=1
W=0
X=1
Y=1
Z=1
ShowHints=1
ShowWarnings=1

UnitAliases=WinTypes=Windows;WinProcs=Windows;DbiTypes=BDE;DbiProcs=BDE;DbiErrs=
BDE;

NamespacePrefix=
SymbolDeprecated=1
SymbolLibrary=1
SymbolPlatform=1
UnitLibrary=1
UnitPlatform=1
UnitDeprecated=1
HResultCompat=1
HidingMember=1
HiddenVirtual=1
Garbage=1
BoundsError=1
ZeroNilCompat=1
StringConstTruncated=1
ForLoopVarVarPar=1
TypedConstVarPar=1
AsgToTypedConst=1
CaseLabelRange=1
ForVariable=1
ConstructingAbstract=1
ComparisonFalse=1
ComparisonTrue=1
ComparingSignedUnsigned=1
CombiningSignedUnsigned=1
UnsupportedConstruct=1
FileOpen=1
FileOpenUnitSrc=1
BadGlobalSymbol=1
DuplicateConstructorDestructor=1
InvalidDirective=1
```

```
PackageNoLink=1
PackageThreadVar=1
ImplicitImport=1
HPPEMITIgnored=1
NoRetVal=1
UseBeforeDef=1
ForLoopVarUndef=1
UnitNameMismatch=1
NoCFGFileFound=1
MessageDirective=1
ImplicitVariants=1
UnicodeToLocale=1
LocaleToUnicode=1
ImagebaseMultiple=1
SuspiciousTypecast=1
PrivatePropAccessor=1
UnsafeType=0
UnsafeCode=0
UnsafeCast=0
[Linker]
MapFile=0
OutputObjs=0
ConsoleApp=1
DebugInfo=0
RemoteSymbols=0
MinStackSize=16384
MaxStackSize=1048576
ImageBase=4194304
ExeDescription=
[Directories]
OutputDir=
UnitOutputDir=
PackageDLLOutputDir=
PackageDCPOutputDir=
SearchPath=
Packages=vcl;rtl;vclx;indy;inet;xmlrtl;vclie;inetdbbde;inetdbxpress;dbRTL;dsnapsnap;
dsnapsnap;vclldb;soaprtl;VclSmp;dbexpress;dbxcds;inetdb;bdertl;vcldbx;webdsnapsnap;web
snap;adortl;ibxpress;teeui;teedb;tee;dss;visualclx;visualdbclx;vclactnband;vclsh
lctrls;IntrawebDB_50_70;Intraweb_50_70;Rave50CLX;Rave50VCL;dclOfficeXP;SynEdit_D
7_PE;DSPack_D7;RxCtl7;RxDB7;RxBDE7;Netrix;MyControls;tb2k_d7;ztv70rt
Conditionals=
DebugSourceDirs=
UsePackages=0
[Parameters]
RunParams=
HostApplication=
Launcher=
UseLauncher=0
DebugCWD=
[Version Info]
IncludeVerInfo=0
AutoIncBuild=0
MajorVer=1
MinorVer=0
Release=0
Build=0
Debug=0
PreRelease=0
Special=0
Private=0
DLL=0
Locale=1049
CodePage=1251
[Version Info Keys]
CompanyName=
FileDescription=
FileVersion=1.0.0.0
InternalName=
LegalCopyright=Sergeeva
```

LegalTrademarks=Sergeeva  
OriginalFilename=  
ProductName=MR  
ProductVersion=1.0.0.0  
Comments=MR 2012

Кафедра \_ КБПЗ \_ 2023рік

## Файл налаштувань nduWlanAPI.pas

```

unit nduWlanAPI;

interface // об'ява даних

uses // Бібліотеки які підключаються
nduCType, nduL2cmn, nduWlanTypes, nduWinDot11, nduWinNT, Windows, nduEapTypes;

Const // константи
NDU_WLAN_API_VERSION=1;
NDU_WLAN_MAX_NAME_LENGTH=NDU_L2_PROFILE_MAX_NAME_LENGTH;

//Флаги профайлу
NDU_WLAN_PROFILE_GROUP_POLICY = $00000001;
NDU_WLAN_PROFILE_USER=$00000002;

WLAN_SET_EAPHOST_DATA_ALL_USERS = $00000001;

WLAN_MAX_PHY_TYPE_NUMBER = 8;

type //Об'ява власних типів даних
Pndu_WLAN_PROFILE_INFO = ^Tndu_WLAN_PROFILE_INFO;
Tndu_WLAN_PROFILE_INFO = record
  strProfileName: array[0..NDU_WLAN_MAX_NAME_LENGTH - 1] of wchar;
  dwFlags: DWORD;
end;

Pndu_DOT11_NETWORK = ^Tndu_DOT11_NETWORK;
Tndu_DOT11_NETWORK = record
  dot11Ssid: Tndu_DOT11_SSID;
  dot11BssType: Tndu_DOT11_BSS_TYPE;
end;

Const // константи
NDU_DOT11_PSD_IE_MAX_DATA_SIZE= 220; // 255 - 6 - 2 - ФОРМАТ ID
NDU_DOT11_PSD_IE_MAX_ENTRY_NUMBER= 10; // мах 10 входжень

type //Об'ява власних типів даних
Pndu_WLAN_RAW_DATA = ^Tndu_WLAN_RAW_DATA;
Tndu_WLAN_RAW_DATA = record
  dwDataSize: DWORD;
  DataBlob: array[0..0] of Byte;
end;

Pndu_WLAN_RAW_DATA_LIST = ^Tndu_WLAN_RAW_DATA_LIST;
PPndu_WLAN_RAW_DATA_LIST = ^Pndu_WLAN_RAW_DATA_LIST;
Tndu_WLAN_RAW_DATA_LIST = record
  dwTotalSize: DWORD;
  dwNumberOfItems: DWORD;
  case Integer of
    0: (dwDataOffset: DWORD);
    1: (dwDataSize: DWORD);
  end;
{$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання
// перелікових типів
Pndu_WLAN_CONNECTION_MODE = ^Tndu_WLAN_CONNECTION_MODE;
Tndu_WLAN_CONNECTION_MODE = (
  wlan_connection_mode_profile = 0,
  wlan_connection_mode_temporary_profile,
  wlan_connection_mode_discovery_secure,
  wlan_connection_mode_discovery_unsecure,
  wlan_connection_mode_auto,
  wlan_connection_mode_invalid);

Tndu_WLAN_REASON_CODE = DWORD;
Pndu_WLAN_REASON_CODE = ^Tndu_WLAN_REASON_CODE;

```

```

Const // константи
    NDU_WLAN_REASON_CODE_SUCCESS= NDU_L2_REASON_CODE_SUCCESS;
    NDU_WLAN_REASON_CODE_UNKNOWN= NDU_L2_REASON_CODE_UNKNOWN;

    NDU_WLAN_REASON_CODE_RANGE_SIZE= NDU_L2_REASON_CODE_GROUP_SIZE;
    NDU_WLAN_REASON_CODE_BASE= NDU_L2_REASON_CODE_DOT11_AC_BASE;

    NDU_WLAN_REASON_CODE_AC_BASE= NDU_L2_REASON_CODE_DOT11_AC_BASE;

    NDU_WLAN_REASON_CODE_AC_CONNECT_BASE=
    (NDU_WLAN_REASON_CODE_AC_BASE + NDU_WLAN_REASON_CODE_RANGE_SIZE div 2);

    NDU_WLAN_REASON_CODE_AC_END= (NDU_WLAN_REASON_CODE_AC_BASE +
    NDU_WLAN_REASON_CODE_RANGE_SIZE - 1);

    NDU_WLAN_REASON_CODE_PROFILE_BASE= NDU_L2_REASON_CODE_PROFILE_BASE;

    NDU_WLAN_REASON_CODE_PROFILE_CONNECT_BASE=(NDU_WLAN_REASON_CODE_PROFILE
    + NDU_WLAN_REASON_CODE_RANGE_SIZE div 2);

    NDU_WLAN_REASON_CODE_PROFILE_END= (NDU_WLAN_REASON_CODE_PROFILE_BASE+
    NDU_WLAN_REASON_CODE_RANGE_SIZE-1);

    // діапазон MSM
    NDU_WLAN_REASON_CODE_MSM_BASE = NDU_L2_REASON_CODE_DOT11_MSM_BASE;

    NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE
    =
    (NDU_WLAN_REASON_CODE_MSM_BASE + NDU_WLAN_REASON_CODE_RANGE_SIZE div 2);

    NDU_WLAN_REASON_CODE_MSM_END
    =
    (NDU_WLAN_REASON_CODE_MSM_BASE + NDU_WLAN_REASON_CODE_RANGE_SIZE - 1);

    // діапазон MSMSEC
    NDU_WLAN_REASON_CODE_MSMSEC_BASE= NDU_L2_REASON_CODE_DOT11_SECURITY_BASE;

    NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE=
    (NDU_WLAN_REASON_CODE_MSMSEC_BASE+NDU_WLAN_REASON_CODE_RANGE_SIZE div 2);

    NDU_WLAN_REASON_CODE_MSMSEC_END=(NDU_WLAN_REASON_CODE_MSMSEC_BASE
    + NDU_WLAN_REASON_CODE_RANGE_SIZE - 1);

    // AC мережні коди
    NDU_WLAN_REASON_CODE_NETWORK_NOT_COMPATIBLE =
    (NDU_WLAN_REASON_CODE_AC_BASE + 1);
    NDU_WLAN_REASON_CODE_PROFILE_NOT_COMPATIBLE =
    (NDU_WLAN_REASON_CODE_AC_BASE + 2);

    // діапазон кодів з'єднань AC
    NDU_WLAN_REASON_CODE_NO_AUTO_CONNECTION =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 1);
    NDU_WLAN_REASON_CODE_NOT_VISIBLE =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 2);
    NDU_WLAN_REASON_CODE_GP_DENIED =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 3);
    NDU_WLAN_REASON_CODE_USER_DENIED =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 4);
    NDU_WLAN_REASON_CODE_BSS_TYPE_NOT_ALLOWED =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 5);
    NDU_WLAN_REASON_CODE_IN_FAILED_LIST =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 6);
    NDU_WLAN_REASON_CODE_IN_BLOCKED_LIST =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 7);
    NDU_WLAN_REASON_CODE_SSID_LIST_TOO_LONG =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 8);
    NDU_WLAN_REASON_CODE_CONNECT_CALL_FAIL =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 9);

```

```

NDU_WLAN_REASON_CODE_SCAN_CALL_FAIL =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 10);
NDU_WLAN_REASON_CODE_NETWORK_NOT_AVAILABLE =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 11);
NDU_WLAN_REASON_CODE_PROFILE_CHANGED_OR_DELETED =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 12);
NDU_WLAN_REASON_CODE_KEY_MISMATCH =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 13);
NDU_WLAN_REASON_CODE_USER_NOT_RESPOND =
    (NDU_WLAN_REASON_CODE_AC_CONNECT_BASE + 14);

// Профайлофі коди помилок
NDU_WLAN_REASON_CODE_INVALID_PROFILE_SCHEMA =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 1);
NDU_WLAN_REASON_CODE_PROFILE_MISSING =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 2);
NDU_WLAN_REASON_CODE_INVALID_PROFILE_NAME =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 3);
NDU_WLAN_REASON_CODE_INVALID_PROFILE_TYPE =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 4);
NDU_WLAN_REASON_CODE_INVALID_PHY_TYPE =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 5);
NDU_WLAN_REASON_CODE_MSM_SECURITY_MISSING =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 6);
NDU_WLAN_REASON_CODE_IHV_SECURITY_NOT_SUPPORTED =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 7);
NDU_WLAN_REASON_CODE_IHV_OUI_MISMATCH =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 8);
NDU_WLAN_REASON_CODE_IHV_OUI_MISSING =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 9);
NDU_WLAN_REASON_CODE_IHV_SETTINGS_MISSING =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 10);
NDU_WLAN_REASON_CODE_CONFLICT_SECURITY =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 11);
NDU_WLAN_REASON_CODE_SECURITY_MISSING =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 12);
NDU_WLAN_REASON_CODE_INVALID_BSS_TYPE =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 13);
NDU_WLAN_REASON_CODE_INVALID_ADHOC_CONNECTION_MODE =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 14);
NDU_WLAN_REASON_CODE_NON_BROADCAST_SET_FOR_ADHOC =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 15);
NDU_WLAN_REASON_CODE_AUTO_SWITCH_SET_FOR_ADHOC =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 16);
NDU_WLAN_REASON_CODE_AUTO_SWITCH_SET_FOR_MANUAL_CONNECTION =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 17);
NDU_WLAN_REASON_CODE_IHV_SECURITY_ONEX_MISSING =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 18);
NDU_WLAN_REASON_CODE_PROFILE_SSID_INVALID =
    (NDU_WLAN_REASON_CODE_PROFILE_BASE + 19);

// MSM мережні коди помилок
NDU_WLAN_REASON_CODE_UNSUPPORTED_SECURITY_SET_BY_OS =
    (NDU_WLAN_REASON_CODE_MSM_BASE + 1);
NDU_WLAN_REASON_CODE_UNSUPPORTED_SECURITY_SET =
    (NDU_WLAN_REASON_CODE_MSM_BASE + 2);
NDU_WLAN_REASON_CODE_BSS_TYPE_UNMATCH =
    (NDU_WLAN_REASON_CODE_MSM_BASE + 3);
NDU_WLAN_REASON_CODE_PHY_TYPE_UNMATCH =
    (NDU_WLAN_REASON_CODE_MSM_BASE + 4);
NDU_WLAN_REASON_CODE_DATA_RATE_UNMATCH =
    (NDU_WLAN_REASON_CODE_MSM_BASE + 5);
NDU_WLAN_REASON_CODE_USER_CANCELLED =
    (NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 1);
NDU_WLAN_REASON_CODE_ASSOCIATION_FAILURE =
    (NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 2);
NDU_WLAN_REASON_CODE_ASSOCIATION_TIMEOUT =
    (NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 3);
NDU_WLAN_REASON_CODE_PRE_SECURITY_FAILURE =

```

```

(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 4);
NDU_WLAN_REASON_CODE_START_SECURITY_FAILURE =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 5);
NDU_WLAN_REASON_CODE_SECURITY_FAILURE =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 6);
NDU_WLAN_REASON_CODE_SECURITY_TIMEOUT =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 7);
NDU_WLAN_REASON_CODE_ROAMING_FAILURE =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 8);
NDU_WLAN_REASON_CODE_ROAMING_SECURITY_FAILURE =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 9);
NDU_WLAN_REASON_CODE_ADHOC_SECURITY_FAILURE =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 10);
NDU_WLAN_REASON_CODE_DRIVER_DISCONNECTED =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 11);
NDU_WLAN_REASON_CODE_DRIVER_OPERATION_FAILURE =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 12);
NDU_WLAN_REASON_CODE_IHV_NOT_AVAILABLE =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 13);
NDU_WLAN_REASON_CODE_IHV_NOT_RESPONDING =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 14);
NDU_WLAN_REASON_CODE_DISCONNECT_TIMEOUT =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 15);
NDU_WLAN_REASON_CODE_INTERNAL_FAILURE =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 16);
NDU_WLAN_REASON_CODE_UI_REQUEST_TIMEOUT =
(NDU_WLAN_REASON_CODE_MSM_CONNECT_BASE + 17);

// MSMSEC коди помилок
NDU_WLAN_REASON_CODE_MSMSEC_MIN = NDU_WLAN_REASON_CODE_MSMSEC_BASE;
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_INVALID_KEY_INDEX =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 1);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_PSK_PRESENT =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 2);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_KEY_LENGTH =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 3);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_PSK_LENGTH =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 4);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_NO_AUTH_CIPHER_SPECIFIED =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 5);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_TOO_MANY_AUTH_CIPHER_SPECIFIED =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 6);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_DUPLICATE_AUTH_CIPHER =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 7);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_RAWDATA_INVALID =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 8);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_INVALID_AUTH_CIPHER =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 9);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_ONEX_DISABLED =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 10);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_ONEX_ENABLED =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 11);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_INVALID_PMKCACHE_MODE =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 12);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_INVALID_PMKCACHE_SIZE =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 13);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_INVALID_PMKCACHE_TTL =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 14);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_INVALID_PREAUTH_MODE =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 15);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_INVALID_PREAUTH_THROTTLE =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 16);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_PREAUTH_ONLY_ENABLED =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 17);
NDU_WLAN_REASON_CODE_MSMSEC_CAPABILITY_NETWORK =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 18);
NDU_WLAN_REASON_CODE_MSMSEC_CAPABILITY_NIC =
(NDU_WLAN_REASON_CODE_MSMSEC_BASE + 19);
NDU_WLAN_REASON_CODE_MSMSEC_CAPABILITY_PROFILE =

```

```

    (NDU_WLAN_REASON_CODE_MSMSEC_BASE + 20);
NDU_WLAN_REASON_CODE_MSMSEC_CAPABILITY_DISCOVERY =
    (NDU_WLAN_REASON_CODE_MSMSEC_BASE + 21);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_PASSPHRASE_CHAR =
    (NDU_WLAN_REASON_CODE_MSMSEC_BASE + 22);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_KEYMATERIAL_CHAR =
    (NDU_WLAN_REASON_CODE_MSMSEC_BASE + 23);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_WRONG_KEYTYPE =
    (NDU_WLAN_REASON_CODE_MSMSEC_BASE + 24);
NDU_WLAN_REASON_CODE_MSMSEC_MIXED_CELL =
    (NDU_WLAN_REASON_CODE_MSMSEC_BASE + 25);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_AUTH_TIMERS_INVALID =
    (NDU_WLAN_REASON_CODE_MSMSEC_BASE + 26);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_INVALID_GKEY_INTV =
    (NDU_WLAN_REASON_CODE_MSMSEC_BASE + 27);
NDU_WLAN_REASON_CODE_MSMSEC_TRANSITION_NETWORK =
    (NDU_WLAN_REASON_CODE_MSMSEC_BASE + 28);
NDU_WLAN_REASON_CODE_MSMSEC_PROFILE_KEY_UNMAPPED_CHAR =
    (NDU_WLAN_REASON_CODE_MSMSEC_BASE + 29);
NDU_WLAN_REASON_CODE_MSMSEC_UI_REQUEST_FAILURE =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 1);
NDU_WLAN_REASON_CODE_MSMSEC_AUTH_START_TIMEOUT =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 2);
NDU_WLAN_REASON_CODE_MSMSEC_AUTH_SUCCESS_TIMEOUT =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 3);
NDU_WLAN_REASON_CODE_MSMSEC_KEY_START_TIMEOUT =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 4);
NDU_WLAN_REASON_CODE_MSMSEC_KEY_SUCCESS_TIMEOUT =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 5);
NDU_WLAN_REASON_CODE_MSMSEC_M3_MISSING_KEY_DATA =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 6);
NDU_WLAN_REASON_CODE_MSMSEC_M3_MISSING_IE =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 7);
NDU_WLAN_REASON_CODE_MSMSEC_M3_MISSING_GRP_KEY =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 8);
NDU_WLAN_REASON_CODE_MSMSEC_PR_IE_MATCHING =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 9);
NDU_WLAN_REASON_CODE_MSMSEC_SEC_IE_MATCHING =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 10);
NDU_WLAN_REASON_CODE_MSMSEC_NO_PAIRWISE_KEY =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 11);
NDU_WLAN_REASON_CODE_MSMSEC_G1_MISSING_KEY_DATA =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 12);
NDU_WLAN_REASON_CODE_MSMSEC_G1_MISSING_GRP_KEY =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 13);
NDU_WLAN_REASON_CODE_MSMSEC_PEER_INDICATED_INSECURE =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 14);
NDU_WLAN_REASON_CODE_MSMSEC_NO_AUTHENTICATOR =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 15);
NDU_WLAN_REASON_CODE_MSMSEC_NIC_FAILURE =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 16);
NDU_WLAN_REASON_CODE_MSMSEC_CANCELLED =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 17);
NDU_WLAN_REASON_CODE_MSMSEC_KEY_FORMAT =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 18);
NDU_WLAN_REASON_CODE_MSMSEC_DOWNGRADE_DETECTED =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 19);
NDU_WLAN_REASON_CODE_MSMSEC_PSK_MISMATCH_SUSPECTED =
    (NDU_WLAN_REASON_CODE_MSMSEC_CONNECT_BASE + 20);

NDU_WLAN_REASON_CODE_MSMSEC_MAX = NDU_WLAN_REASON_CODE_MSMSEC_END;

type //Об'ява власних типів даних
    Tndu_WLAN_SIGNAL_QUALITY = ulong;
    Pndu_WLAN_SIGNAL_QUALITY = ^Tndu_WLAN_SIGNAL_QUALITY;

Const // константи
    NDU_WLAN_AVAILABLE_NETWORK_CONNECTED = $00000001;
    NDU_WLAN_AVAILABLE_NETWORK_HAS_PROFILE = $00000002;

```

```

NDU_WLAN_AVAILABLE_NETWORK_INCLUDE_ALL_ADHOC_PROFILES    = $00000001;
NDU_WLAN_AVAILABLE_NETWORK_INCLUDE_ALL_MANUAL_HIDDEN_PROFILES = $00000002;

type //Об'ява власних типів даних
    Pndu_WLAN_RATE_SET = ^Tndu_WLAN_RATE_SET;
    Tndu_WLAN_RATE_SET = record
        uRateSetLength: ulong;
        usRateSet: array[0..NDU_DOT11_RATE_SET_MAX_LENGTH - 1] of ushort;
    end;

    Pndu_WLAN_AVAILABLE_NETWORK = ^Tndu_WLAN_AVAILABLE_NETWORK;

    Tndu_WLAN_AVAILABLE_NETWORK = record
        strProfileName: array[0..NDU_WLAN_MAX_NAME_LENGTH - 1] of wchar;
        dot11Ssid: Tndu_DOT11_SSID;
        dot11BssType: Tndu_DOT11_BSS_TYPE;
        uNumberOfBssids: ulong;
        bNetworkConnectable: Bool;
        wlanNotConnectableReason: Tndu_WLAN_REASON_CODE;
        uNumberOfPhyTypes: ulong;
        dot11PhyTypes: array[0..WLAN_MAX_PHY_TYPE_NUMBER - 1] of Tndu_DOT11_PHY_TYPE;
        bMorePhyTypes: Bool;
        wlanSignalQuality: Tndu_WLAN_SIGNAL_QUALITY;
        bSecurityEnabled: Bool;
        dot11DefaultAuthAlgorithm: Tndu_DOT11_AUTH_ALGORITHM;
        dot11DefaultCipherAlgorithm: Tndu_DOT11_CIPHER_ALGORITHM;
        dwFlags: DWORD;
        dwReserved: DWORD;
    end;

    Pndu_WLAN_BSS_ENTRY = ^Tndu_WLAN_BSS_ENTRY;
    Tndu_WLAN_BSS_ENTRY = record
        dot11Ssid: Tndu_DOT11_SSID;
        uPhyId: ulong;
        dot11Bssid: Tndu_DOT11_MAC_ADDRESS;
        dot11BssType: Tndu_DOT11_BSS_TYPE;
        dot11BssPhyType: Tndu_DOT11_PHY_TYPE;
        lRssi: long;
        uLinkQuality: ulong;
        bInRegDomain: Boolean;
        usBeaconPeriod: ushort;
        ullTimestamp: ulonglong;
        ullHostTimestamp: ulonglong;
        usCapabilityInformation: ushort;
        ulChCenterFrequency: ulong;
        wlanRateSet: Tndu_WLAN_RATE_SET;
        ulIeOffset: ulong;
        ulIeSize: ulong;
    end;

    Pndu_WLAN_BSS_LIST = ^Tndu_WLAN_BSS_LIST;
    PPndu_WLAN_BSS_LIST = ^Pndu_WLAN_BSS_LIST;
    Tndu_WLAN_BSS_LIST = record
        dwTotalSize: DWORD;
        dwNumberOfItems: DWORD;
        wlanBssEntries: array[0..0] of Tndu_WLAN_BSS_ENTRY;
    end;

    {$MINENUMSIZE 4}
    // Встановлює мінімальну пам'ять, використовувану для зберігання
    // перелікових типів

    Pndu_WLAN_INTERFACE_STATE = ^Tndu_WLAN_INTERFACE_STATE;
    Tndu_WLAN_INTERFACE_STATE = (
        wlan_interface_state_not_ready = 0,
        wlan_interface_state_connected,
        wlan_interface_state_ad_hoc_network_formed,
        wlan_interface_state_disconnecting,

```

```

wlan_interface_state_disconnected,
wlan_interface_state_associating,
wlan_interface_state_discovering,
wlan_interface_state_authenticating);

Pndu_WLAN_INTERFACE_INFO = ^Tndu_WLAN_INTERFACE_INFO;
Tndu_WLAN_INTERFACE_INFO = record
    InterfaceGuid: TGUID;
    strInterfaceDescription: array[0..NDU_WLAN_MAX_NAME_LENGTH - 1] of wchar;
    isState: Tndu_WLAN_INTERFACE_STATE;
end;

Pndu_WLAN_ASSOCIATION_ATTRIBUTES = ^Tndu_WLAN_ASSOCIATION_ATTRIBUTES;
Tndu_WLAN_ASSOCIATION_ATTRIBUTES = record
    dot11Ssid: Tndu_DOT11_SSID;
    dot11BssType: Tndu_DOT11_BSS_TYPE;
    dot11Bssid: Tndu_DOT11_MAC_ADDRESS;
    dot11PhyType: Tndu_DOT11_PHY_TYPE;
    uDot11PhyIndex: ulong;
    wlanSignalQuality: Tndu_WLAN_SIGNAL_QUALITY;
    ulRxRate: ulong;
    ulTxRate: ulong;
end;

Pndu_WLAN_SECURITY_ATTRIBUTES = ^Tndu_WLAN_SECURITY_ATTRIBUTES;
Tndu_WLAN_SECURITY_ATTRIBUTES = record
    bSecurityEnabled: Bool;
    dot11AuthAlgorithm: Tndu_DOT11_AUTH_ALGORITHM;
    dot11CipherAlgorithm: Tndu_DOT11_CIPHER_ALGORITHM;
end;

Pndu_WLAN_CONNECTION_ATTRIBUTES = ^Tndu_WLAN_CONNECTION_ATTRIBUTES;
Tndu_WLAN_CONNECTION_ATTRIBUTES = record
    isState: Tndu_WLAN_INTERFACE_STATE;
    wlanConnectionMode: Tndu_WLAN_CONNECTION_MODE;
    strProfileName: array[0..NDU_WLAN_MAX_NAME_LENGTH - 1] of wchar;
    wlanAssociationAttributes: Tndu_WLAN_ASSOCIATION_ATTRIBUTES;
    wlanSecurityAttributes: Tndu_WLAN_SECURITY_ATTRIBUTES;
end;

{$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання
// перелікових типів
Pndu_DOT11_RADIO_STATE = ^Tndu_DOT11_RADIO_STATE;
Tndu_DOT11_RADIO_STATE = (
    dot11_radio_state_unknown = 0,
    dot11_radio_state_on,
    dot11_radio_state_off);

Const // константи
    // максимальна кількість підключень
    NDU_WLAN_MAX_PHY_INDEX = 63;

type //Об'ява власних типів даних
    Pndu_WLAN_PHY_RADIO_STATE = ^Tndu_WLAN_PHY_RADIO_STATE;
    Tndu_WLAN_PHY_RADIO_STATE = record
        dwPhyIndex: DWORD;
        dot11SoftwareRadioState: Tndu_DOT11_RADIO_STATE;
        dot11HardwareRadioState: Tndu_DOT11_RADIO_STATE;
    end;

Pndu_WLAN_RADIO_STATE = ^Tndu_WLAN_RADIO_STATE;
Tndu_WLAN_RADIO_STATE = record
    dwNumberOfPhys: DWORD;
    PhyRadioState: array[0..NDU_WLAN_MAX_PHY_INDEX - 1] of
Tndu_WLAN_PHY_RADIO_STATE;
end;
{$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання

```

```

// перелікових типів
Pndu_WLAN_INTERFACE_TYPE = ^Tndu_WLAN_INTERFACE_TYPE;
Tndu_WLAN_INTERFACE_TYPE = (
    wlan_interface_type_emulated_802_11 = 0,
    wlan_interface_type_native_802_11,
    wlan_interface_type_invalid);

Pndu_WLAN_INTERFACE_CAPABILITY = ^Tndu_WLAN_INTERFACE_CAPABILITY;
PPndu_WLAN_INTERFACE_CAPABILITY = ^Pndu_WLAN_INTERFACE_CAPABILITY;
Tndu_WLAN_INTERFACE_CAPABILITY = record
    interfaceType: Tndu_WLAN_INTERFACE_TYPE;
    bDot11DSupported: Bool;
    dwMaxDesiredSsidListSize: DWORD;
    dwMaxDesiredBssidListSize: DWORD;
    dwNumberOfSupportedPhys: DWORD;
    dot11PhyTypes: array[0..NDU_WLAN_MAX_PHY_INDEX - 1] of Tndu_DOT11_PHY_TYPE;
end;

Pndu_WLAN_AUTH_CIPHER_PAIR_LIST = ^Tndu_WLAN_AUTH_CIPHER_PAIR_LIST;
Tndu_WLAN_AUTH_CIPHER_PAIR_LIST = record
    pAuthCipherPairList: array[0..0] of Tndu_DOT11_AUTH_CIPHER_PAIR;
end;

Pndu_WLAN_COUNTRY_OR_REGION_STRING_LIST =
^Tndu_WLAN_COUNTRY_OR_REGION_STRING_LIST;
Tndu_WLAN_COUNTRY_OR_REGION_STRING_LIST = record
    pCountryOrRegionStringList: array[0..0] of
Tndu_DOT11_COUNTRY_OR_REGION_STRING;
end;

Pndu_WLAN_PROFILE_INFO_LIST = ^Tndu_WLAN_PROFILE_INFO_LIST;
PPndu_WLAN_PROFILE_INFO_LIST = ^Pndu_WLAN_PROFILE_INFO_LIST;
Tndu_WLAN_PROFILE_INFO_LIST = record
    dwNumberOfItems: DWORD;
    dwIndex: DWORD;
    ProfileInfo: array[0..0] of Tndu_WLAN_PROFILE_INFO;
end;

Pndu_WLAN_AVAILABLE_NETWORK_LIST = ^Tndu_WLAN_AVAILABLE_NETWORK_LIST;
PPndu_WLAN_AVAILABLE_NETWORK_LIST = ^Pndu_WLAN_AVAILABLE_NETWORK_LIST;
Tndu_WLAN_AVAILABLE_NETWORK_LIST = record
    dwNumberOfItems: DWORD;
    dwIndex: DWORD;
    Network: array[0..0] of Tndu_WLAN_AVAILABLE_NETWORK;
end;

Pndu_WLAN_INTERFACE_INFO_LIST = ^Tndu_WLAN_INTERFACE_INFO_LIST;
PPndu_WLAN_INTERFACE_INFO_LIST = ^Pndu_WLAN_INTERFACE_INFO_LIST;
Tndu_WLAN_INTERFACE_INFO_LIST = record
    dwNumberOfItems: DWORD;
    dwIndex: DWORD;
    InterfaceInfo: array[0..0] of Tndu_WLAN_INTERFACE_INFO;
end;

Pndu_DOT11_NETWORK_LIST = ^Tndu_DOT11_NETWORK_LIST;
PPndu_DOT11_NETWORK_LIST = ^Pndu_DOT11_NETWORK_LIST;
Tndu_DOT11_NETWORK_LIST = record
    dwNumberOfItems: DWORD;
    dwIndex: DWORD;
    Network: array[0..0] of Tndu_DOT11_NETWORK;
end;

{$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання
// перелікових типів
Pndu_WLAN_POWER_SETTING = ^Tndu_WLAN_POWER_SETTING;
Tndu_WLAN_POWER_SETTING = (
    wlan_power_setting_no_saving = 0,
    wlan_power_setting_low_saving,

```

```

wlan_power_setting_medium_saving,
wlan_power_setting_maximum_saving,
wlan_power_setting_invalid);

Const // константи
NDU_WLAN_CONNECTION_HIDDEN_NETWORK= $00000001;
NDU_WLAN_CONNECTION_ADHOC_JOIN_ONLY = $00000002;

type //Об'ява власних типів даних
    Pndu_WLAN_CONNECTION_PARAMETERS = ^Tndu_WLAN_CONNECTION_PARAMETERS;
    Tndu_WLAN_CONNECTION_PARAMETERS = record
        wlanConnectionMode: Tndu_WLAN_CONNECTION_MODE;
        strProfile: LPCTSTR;
        pDot11Ssid: Pndu_DOT11_SSID;
        pDesiredBssidList: Pndu_DOT11_BSSID_LIST;
        dot11BssType: Tndu_DOT11_BSS_TYPE;
        dwFlags: DWORD;
    end;

Pndu_WLAN_MSM_NOTIFICATION_DATA = ^Tndu_WLAN_MSM_NOTIFICATION_DATA;
Tndu_WLAN_MSM_NOTIFICATION_DATA = record
    wlanConnectionMode: Tndu_WLAN_CONNECTION_MODE;
    strProfileName: array[0..NDU_WLAN_MAX_NAME_LENGTH - 1] of wchar;
    dot11Ssid: Tndu_DOT11_SSID;
    dot11BssType: Tndu_DOT11_BSS_TYPE;
    dot11MacAddr: Tndu_DOT11_MAC_ADDRESS;
    bSecurityEnabled: Bool;
    bFirstPeer: Bool;
    bLastPeer: Bool;
    wlanReasonCode: Tndu_WLAN_REASON_CODE;
end;

Pndu_WLAN_CONNECTION_NOTIFICATION_DATA =
^Tndu_WLAN_CONNECTION_NOTIFICATION_DATA;
Tndu_WLAN_CONNECTION_NOTIFICATION_DATA = record
    wlanConnectionMode: Tndu_WLAN_CONNECTION_MODE;
    strProfileName: array[0..NDU_WLAN_MAX_NAME_LENGTH - 1] of wchar;
    dot11Ssid: Tndu_DOT11_SSID;
    dot11BssType: Tndu_DOT11_BSS_TYPE;
    bSecurityEnabled: Bool;
    wlanReasonCode: Tndu_WLAN_REASON_CODE;
    strProfileXml: array[0..0] of wchar;
end;

Const // константи

NDU_WLAN_NOTIFICATION_SOURCE_NONE = NDU_L2_NOTIFICATION_SOURCE_NONE;
NDU_WLAN_NOTIFICATION_SOURCE_ALL = NDU_L2_NOTIFICATION_SOURCE_ALL;

NDU_WLAN_NOTIFICATION_SOURCE_ACM= NDU_L2_NOTIFICATION_SOURCE_WLAN_ACM;
NDU_WLAN_NOTIFICATION_SOURCE_MSM =NDU_L2_NOTIFICATION_SOURCE_WLAN_MSM;
NDU_WLAN_NOTIFICATION_SOURCE_SECURITY=
NDU_L2_NOTIFICATION_SOURCE_WLAN_SECURITY;
NDU_WLAN_NOTIFICATION_SOURCE_IHV= NDU_L2_NOTIFICATION_SOURCE_WLAN_IHV;

type //Об'ява власних типів даних
    {$MINENUMSIZE 4}
    // Встановлює мінімальну пам'ять, використовувану для зберігання
    // перелікових типів
    Pndu_WLAN_NOTIFICATION_ACM = ^Tndu_WLAN_NOTIFICATION_ACM;
    Tndu_WLAN_NOTIFICATION_ACM = (
        wlan_notification_acm_start = NDU_L2_NOTIFICATION_CODE_PUBLIC_BEGIN,
        wlan_notification_acm_autoconf_enabled,
        wlan_notification_acm_autoconf_disabled,
        wlan_notification_acm_background_scan_enabled,
        wlan_notification_acm_background_scan_disabled,
        wlan_notification_acm_bss_type_change,
        wlan_notification_acm_power_setting_change,
        wlan_notification_acm_scan_complete,

```

```

wlan_notification_acm_scan_fail,
wlan_notification_acm_connection_start,
wlan_notification_acm_connection_complete,
wlan_notification_acm_connection_attempt_fail,
wlan_notification_acm_filter_list_change,
wlan_notification_acm_interface_arrival,
wlan_notification_acm_interface_removal,
wlan_notification_acm_profile_change,
wlan_notification_acm_profile_name_change,
wlan_notification_acm_profiles_exhausted,
wlan_notification_acm_network_not_available,
wlan_notification_acm_network_available,
wlan_notification_acm_disconnecting,
wlan_notification_acm_disconnected,
wlan_notification_acm_end);
{$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання
// перелікових типів
Pndu_WLAN_NOTIFICATION_MSM = ^Tndu_WLAN_NOTIFICATION_MSM;
Tndu_WLAN_NOTIFICATION_MSM = (
    wlan_notification_msm_start = NDU_L2_NOTIFICATION_CODE_PUBLIC_BEGIN,
    wlan_notification_msm_associating,
    wlan_notification_msm_associated,
    wlan_notification_msm_authenticating,
    wlan_notification_msm_connected,
    wlan_notification_msm_roaming_start,
    wlan_notification_msm_roaming_end,
    wlan_notification_msm_radio_state_change,
    wlan_notification_msm_signal_quality_change,
    wlan_notification_msm_disassociating,
    wlan_notification_msm_disconnected,
    wlan_notification_msm_peer_join,
    wlan_notification_msm_peer_leave,
    wlan_notification_msm_end);

{$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання
// перелікових типів
Pndu_WLAN_NOTIFICATION_SECURITY = ^Tndu_WLAN_NOTIFICATION_SECURITY;
Tndu_WLAN_NOTIFICATION_SECURITY = (
    wlan_notification_security_start = NDU_L2_NOTIFICATION_CODE_PUBLIC_BEGIN,
    wlan_notification_security_end);

Tndu_WLAN_NOTIFICATION_DATA = Tndu_L2_NOTIFICATION_DATA;
Pndu_WLAN_NOTIFICATION_DATA = ^Tndu_WLAN_NOTIFICATION_DATA;

Tndu_WLAN_NOTIFICATION_CALLBACK = PVOID;
Pndu_WLAN_NOTIFICATION_CALLBACK = PVOID;

{$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання
// перелікових типів
Pndu_WLAN_OPCODE_VALUE_TYPE = ^Tndu_WLAN_OPCODE_VALUE_TYPE;
Tndu_WLAN_OPCODE_VALUE_TYPE = (
    wlan_opcode_value_type_query_only = 0,
    wlan_opcode_value_type_set_by_group_policy,
    wlan_opcode_value_type_set_by_user,
    wlan_opcode_value_type_invalid);

{$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання
// перелікових типів
Pndu_WLAN_INTF_OPCODE = ^Tndu_WLAN_INTF_OPCODE;
Tndu_WLAN_INTF_OPCODE = (
    wlan_intf_opcode_autoconf_start = $000000000,
    wlan_intf_opcode_autoconf_enabled,
    wlan_intf_opcode_background_scan_enabled,
    wlan_intf_opcode_media_streaming_mode,
    wlan_intf_opcode_radio_state,

```

```

wlan_intf_opcode_bss_type,
wlan_intf_opcode_interface_state,
wlan_intf_opcode_current_connection,
wlan_intf_opcode_channel_number,
wlan_intf_opcode_supported_infrastructure_auth_cipher_pairs,
wlan_intf_opcode_supported_adhoc_auth_cipher_pairs,
wlan_intf_opcode_supported_country_or_region_string_list,
wlan_intf_opcode_autoconf_end = $0fffffff,
wlan_intf_opcode_msm_start = $10000100,
wlan_intf_opcode_statistics,
wlan_intf_opcode_rssi,
wlan_intf_opcode_msm_end = $1fffffff,
wlan_intf_opcode_security_start = $20010000,
wlan_intf_opcode_security_end = $2fffffff,
wlan_intf_opcode_ihv_start = $30000000,
wlan_intf_opcode_ihv_end = $3fffffff);

{$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання
// перелікових типів
Pndu_WLAN_AUTOCONF_OPCODE = ^Tndu_WLAN_AUTOCONF_OPCODE;
Tndu_WLAN_AUTOCONF_OPCODE = (
    wlan_autoconf_opcode_start = 0,
    wlan_autoconf_opcode_show_denied_networks,
    wlan_autoconf_opcode_power_setting,
    wlan_autoconf_opcode_connect_with_all_user_profile_only,
    wlan_autoconf_opcode_end);

{$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання
// перелікових типів
Pndu_WLAN_IHV_CONTROL_TYPE = ^Tndu_WLAN_IHV_CONTROL_TYPE;
Tndu_WLAN_IHV_CONTROL_TYPE = (
    wlan_ihv_control_type_service,
    wlan_ihv_control_type_driver);

{$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання
// перелікових типів
Pndu_WLAN_FILTER_LIST_TYPE = ^Tndu_WLAN_FILTER_LIST_TYPE;
Tndu_WLAN_FILTER_LIST_TYPE = (
    wlan_filter_list_type_gp_permit,
    wlan_filter_list_type_gp_deny,
    wlan_filter_list_type_user_permit,
    wlan_filter_list_type_user_deny);

Pndu_WLAN_PHY_FRAME_STATISTICS = ^Tndu_WLAN_PHY_FRAME_STATISTICS;
Tndu_WLAN_PHY_FRAME_STATISTICS = record
    ullTransmittedFrameCount: ulonglong;
    ullMulticastTransmittedFrameCount: ulonglong;
    ullFailedCount: ulonglong;
    ullRetryCount: ulonglong;
    ullMultipleRetryCount: ulonglong;
    ullMaxTXLifetimeExceededCount: ulonglong;
    ullTransmittedFragmentCount: ulonglong;
    ullRTSSuccessCount: ulonglong;
    ullRTSFailureCount: ulonglong;
    ullACKFailureCount: ulonglong;
    ullReceivedFrameCount: ulonglong;
    ullMulticastReceivedFrameCount: ulonglong;
    ullPromiscuousReceivedFrameCount: ulonglong;
    ullMaxRXLifetimeExceededCount: ulonglong;
    ullFrameDuplicateCount: ulonglong;
    ullReceivedFragmentCount: ulonglong;
    ullPromiscuousReceivedFragmentCount: ulonglong;
    ullFCSErrorCount: ulonglong;
end;

Pndu_WLAN_MAC_FRAME_STATISTICS = ^Tndu_WLAN_MAC_FRAME_STATISTICS;

```

```

Tndu_WLAN_MAC_FRAME_STATISTICS = record
    ullTransmittedFrameCount: ulonglong;
    ullReceivedFrameCount: ulonglong;
    ullWEPExcludedCount: ulonglong;
    ullTKIPLocalMICFailures: ulonglong;
    ullTKIPReplays: ulonglong;
    ullTKIPICVErrorCount: ulonglong;
    ullCCMPReplays: ulonglong;
    ullCCMPDecryptErrors: ulonglong;
    ullWEPUndecryptableCount: ulonglong;
    ullWEPICVErrorCount: ulonglong;
    ullDecryptSuccessCount: ulonglong;
    ullDecryptFailureCount: ulonglong;
end;

Pndu_WLAN_STATISTICS = ^Tndu_WLAN_STATISTICS;
Tndu_WLAN_STATISTICS = record
    ullFourWayHandshakeFailures: ulonglong;
    ullTKIPCounterMeasuresInvoked: ulonglong;
    ullReserved: ulonglong;
    MacUcastCounters: Tndu_WLAN_MAC_FRAME_STATISTICS;
    MacMcastCounters: Tndu_WLAN_MAC_FRAME_STATISTICS;
    dwNumberOfPhys: DWORD;
    PhyCounters: array[0..0] of Tndu_WLAN_PHY_FRAME_STATISTICS;
end;

Const // константи
    NDU_WLAN_READ_ACCESS = (NDU_STANDARD_RIGHTS_READ or NDU_FILE_READ_DATA);
    NDU_WLAN_EXECUTE_ACCESS = (NDU_WLAN_READ_ACCESS or
        NDU_STANDARD_RIGHTS_EXECUTE or NDU_FILE_EXECUTE);
    NDU_WLAN_WRITE_ACCESS = (NDU_WLAN_READ_ACCESS or NDU_WLAN_EXECUTE_ACCESS or
        NDU_STANDARD_RIGHTS_WRITE or NDU_FILE_WRITE_DATA or NDU_DELETE or
        NDU_WRITE_DAC);

type //Об'ява власних типів даних
    {$MINENUMSIZE 4}
// Встановлює мінімальну пам'ять, використовувану для зберігання
// перелікових типів
    Pndu_WLAN_SECURABLE_OBJECT = ^Tndu_WLAN_SECURABLE_OBJECT;
Tndu_WLAN_SECURABLE_OBJECT = (
    wlan_secure_permit_list = 0,
    wlan_secure_deny_list,
    wlan_secure_ac_enabled,
    wlan_secure_bc_scan_enabled,
    wlan_secure_bss_type,
    wlan_secure_show_denied,
    wlan_secure_interface_properties,
    wlan_secure_ihv_control,
    wlan_secure_all_user_profiles_order,
    wlan_secure_sso,
    wlan_secure_add_new_all_user_profiles,
    wlan_secure_add_new_per_user_profiles,
    wlan_secure_manual_connect_single_user,
    wlan_secure_manual_connect_multi_user,
    wlan_secure_media_streaming_mode_enabled,
    NDU_WLAN_SECURABLE_OBJECT_COUNT);

Const // константи
    wlan_api_dll = 'wlanapi.dll';

    function WlanOpenHandle(dwClientVersion: DWORD; pReserved: PVOID;
        pdwNegotiatedVersion: PWord; phClientHandle: PHandle): DWORD; stdcall;

function WlanCloseHandle(hClientHandle: Handle;
    pReserved: PVOID): DWORD; stdcall;

    function WlanEnumInterfaces(hClientHandle: Handle;
        pReserved: PVOID; ppInterfaceList: PPndu_WLAN_INTERFACE_INFO_LIST
    ): DWORD; stdcall;

```

```

function WlanSetAutoConfigParameter(hClientHandle: Handle;
    OpCode: Tndu_WLAN_AUTOCONF_OPCODE; dwDataSize: DWORD;
    const pData: PVOID; pReserved: PVOID): DWORD; stdcall;

function WlanQueryAutoConfigParameter(hClientHandle: Handle;
    OpCode: Tndu_WLAN_AUTOCONF_OPCODE; pReserved: PVOID;
    pdwDataSize: PDWORD; ppData: PPVOID;
    pWlanOpcodeValueType: Pndu_WLAN_OPCODE_VALUE_TYPE): DWORD; stdcall;

function WlanGetInterfaceCapability(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; pReserved: PVOID;
    ppCapability: PPndu_WLAN_INTERFACE_CAPABILITY): DWORD; stdcall;

function WlanSetInterface(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; OpCode: Tndu_WLAN_INTF_OPCODE;
    dwDataSize: DWORD; const pData: PVOID; pReserved: PVOID): DWORD; stdcall;

function WlanQueryInterface(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; OpCode: Tndu_WLAN_INTF_OPCODE;
    pReserved: PVOID; pdwDataSize: PDWORD; ppData: PPVOID;
    pWlanOpcodeValueType: Pndu_WLAN_OPCODE_VALUE_TYPE): DWORD; stdcall;

function WlanIhvControl(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; aType: Tndu_WLAN_IHV_CONTROL_TYPE;
    dwInBufferSize: DWORD; pInBuffer: pvoid; dwOutBufferSize: DWORD;
    pOutBuffer: PVOID): DWORD; stdcall;

function WlanScan(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; const pDot11Ssid: Pndu_DOT11_SSID;
    const pIeData: Pndu_WLAN_RAW_DATA; pReserved: PVOID): DWORD; stdcall;

    function WlanGetAvailableNetworkList(hClientHandle: Handle;
        const pInterfaceGuid: PGUID; dwFlags: DWORD; pReserved: PVOID;
        var pAvailableNetworkList: Pndu_WLAN_AVAILABLE_NETWORK_LIST):DWORD;
        stdcall;

function WlanGetNetworkBssList(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; const pDot11Ssid: Pndu_DOT11_SSID;
    dot11BssType: Tndu_DOT11_BSS_TYPE; bSecurityEnabled: BOOL;
    pReserved: PVOID; ppWlanBssList: PPndu_WLAN_BSS_LIST):DWORD;
    stdcall;

function WlanConnect(hClientHandle: Handle;const pInterfaceGuid: PGUID;
    const pConnectionParameters: Pndu_WLAN_CONNECTION_PARAMETERS;
    pReserved: PVOID): DWORD; stdcall;

function WlanDisconnect(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; pReserved: PVOID): DWORD; stdcall;

function WlanRegisterNotification(hClientHandle: Handle;
    dwNotifSource: DWORD; bIgnoreDuplicate: Bool;
    funcCallback: Tndu_WLAN_NOTIFICATION_CALLBACK;
    pCallbackContext: PVOID; pReserved: PVOID;
    pdwPrevNotifSource: PDWORD): DWORD; stdcall;

function WlanGetProfile(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; strProfileName: LPCWSTR;
    pReserved: PVOID; pstrProfileXml: LPWSTR; pdwFlags: PDWORD;
    pdwGrantedAccess: PDWORD): DWORD; stdcall;

function WlanSetProfileEapUserData(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; strProfileName: LPCWSTR;
    eapType: Tndu_EAP_METHOD_TYPE; dwFlags: DWORD;
    dwEapUserDataSize: DWORD; const pbEapUserData: LPByte;
    pReserved: PVOID): DWORD; stdcall;

function WlanSetProfileEapXMLUserData(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; strProfileName: LPCWSTR;
    eapType: Tndu_EAP_METHOD_TYPE; dwFlags: DWORD;

```

```

strEapXMLUserData: LPCWSTR; pReserved: PVOID): DWORD; stdcall;

function WlanSetProfile(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; dwFlags: DWORD; strProfileXml: LPCWSTR;
    strAllUserProfileSecurity: LPCWSTR;
    bOverwrite: Bool; pReserved: PVOID;
    pdwReasonCode: PDWORD): DWORD; stdcall;

function WlanDeleteProfile(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; strProfileName: LPCWSTR;
    pReserved: PVOID): DWORD; stdcall;

function WlanRenameProfile(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; strOldProfileName: LPCWSTR;
    strNewProfileName: LPCWSTR; pReserved: PVOID): DWORD; stdcall;

function WlanGetProfileList(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; pReserved: PVOID;
    ppProfileList: PPndu_WLAN_PROFILE_INFO_LIST): DWORD; stdcall;

function WlanSetProfileList(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; dwItems: DWORD;
    strProfileNames: LPCWSTR; pReserved: PVOID): DWORD; stdcall;

function WlanSetProfilePosition(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; strProfileName: LPCWSTR;
    dwPosition: DWORD; pReserved: PVOID): DWORD; stdcall;

function WlanSetProfileCustomUserData(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; strProfileName: LPCWSTR;
    dwDataSize: DWORD; const pData: LPByte;
    pReserved: PVOID): DWORD; stdcall;

function WlanGetProfileCustomUserData(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; strProfileName: LPCWSTR;
    pReserved: PVOID; pdwDataSize: PDWORD; ppData: PPByte): DWORD; stdcall;

function WlanSetFilterList(hClientHandle: Handle;
    wlanFilterListType: Tndu_WLAN_FILTER_LIST_TYPE;
    const pNetworkList: Pndu_DOT11_NETWORK_LIST;
    pReserved: PVOID): DWORD; stdcall;

function WlanGetFilterList(hClientHandle: Handle;
    wlanFilterListType: Tndu_WLAN_FILTER_LIST_TYPE;
    pReserved: PVOID; ppNetworkList: PPndu_DOT11_NETWORK_LIST): DWORD; stdcall;

function WlanSetPsdIEDataList(hClientHandle: Handle; strFormat: LPCWSTR;
    const pPsdIEDataList: Pndu_WLAN_RAW_DATA_LIST;
    pReserved: pvoid): DWORD; stdcall;

function WlanSaveTemporaryProfile(hClientHandle: Handle;
    const pInterfaceGuid: PGUID; strProfileName: LPCWSTR;
    strAllUserProfileSecurity: LPCWSTR; dwFlags: DWORD;
    bOverwrite: Bool; pReserved: PVOID): DWORD; stdcall;

function WlanExtractPsdIEDataList(hClientHandle: Handle;
    dwIeDataSize: DWORD; const pRawIeData: PByte;
    strFormat: LPCWSTR; pReserved: PVOID;
    ppPsdIEDataList: PPndu_WLAN_RAW_DATA_LIST): DWORD; stdcall;

    function WlanReasonCodeToString(dwReasonCode: DWORD;
        dwBufferSize: DWORD; pStringBuffer: PWChar;
        pReserved: PVOID): DWORD; stdcall;

function WlanAllocateMemory(dwMemorySize: DWORD): pvoid; stdcall;

function WlanFreeMemory(pMemory: PVOID): PVOID; stdcall;

```

```

function WlanSetSecuritySettings(hClientHandle: Handle;
    SecurableObject: Tndu_WLAN_SECURABLE_OBJECT;
    strModifiedSDDL: LPCWSTR): DWORD; stdcall;

function WlanGetSecuritySettings(hClientHandle: Handle;
    SecurableObject: Tndu_WLAN_SECURABLE_OBJECT;
    pstrCurrentSDDL: PLPWSTR; pdwGrantedAccess: PWORD): DWORD; stdcall;

Const // константи
    NDU_WLAN_UI_API_VERSION = 1;
    NDU_WLAN_UI_API_INITIAL_VERSION = 1;

type //Об'ява власних типів даних
    Pndu_WL_DISPLAY_PAGES = ^Tndu_WL_DISPLAY_PAGES;
    Tndu_WL_DISPLAY_PAGES = (
        WLConnectionPage,
        WLSecurityPage);

function WlanUIEditProfile(dwClientVersion: DWORD;
    wstrProfileName: LPCWSTR; pInterfaceGuid: PGUID;
    hWnd: HWND; wlStartPage: Tndu_WL_DISPLAY_PAGES;
    pReserved: PVOID; pWlanReasonCode: Pndu_WLAN_REASON_CODE): DWORD; stdcall;

implementation // реалізація
//Перелік функцій які отримуються з бібліотеки wlanapi.dll
// Функції взаємодії з бездротовою мережею
function WlanOpenHandle;external wlan_api_dll name 'WlanOpenHandle';
function WlanCloseHandle;external wlan_api_dll name 'WlanCloseHandle';
function WlanEnumInterfaces;external wlan_api_dll name 'WlanEnumInterfaces';
function WlanSetAutoConfigParameter;external wlan_api_dll
    name 'WlanSetAutoConfigParameter';
function WlanQueryAutoConfigParameter;external wlan_api_dll
    name 'WlanQueryAutoConfigParameter';
function WlanGetInterfaceCapability;external wlan_api_dll name
    'WlanGetInterfaceCapability';
function WlanSetInterface;external wlan_api_dll name 'WlanSetInterface';
function WlanQueryInterface;external wlan_api_dll name 'WlanQueryInterface';
function WlanIhvControl; external wlan_api_dll name 'WlanIhvControl';
function WlanScan;external wlan_api_dll name 'WlanScan';
function WlanGetAvailableNetworkList;external wlan_api_dll name
    'WlanGetAvailableNetworkList';
function WlanGetNetworkBssList;external wlan_api_dll name
    'WlanGetNetworkBssList';
function WlanConnect;external wlan_api_dll name 'WlanConnect';
function WlanDisconnect;external wlan_api_dll name 'WlanDisconnect';
function WlanRegisterNotification;external wlan_api_dll name
    'WlanRegisterNotification';
function WlanGetProfile;external wlan_api_dll name 'WlanGetProfile';
function WlanSetProfileEapUserData;external wlan_api_dll name
    'WlanSetProfileEapUserData';
function WlanSetProfileEapXMLUserData;external wlan_api_dll name
    'WlanSetProfileEapXMLUserData';
function WlanSetProfile;external wlan_api_dll name 'WlanSetProfile';
function WlanDeleteProfile;external wlan_api_dll name 'WlanDeleteProfile';
function WlanRenameProfile;external wlan_api_dll name 'WlanRenameProfile';
function WlanGetProfileList;external wlan_api_dll name 'WlanGetProfileList';
function WlanSetProfileList;external wlan_api_dll name 'WlanSetProfileList';
function WlanSetProfilePosition;external wlan_api_dll name
    'WlanSetProfilePosition';
function WlanSetProfileCustomUserData;external wlan_api_dll name
    'WlanSetProfileCustomUserData';
function WlanGetProfileCustomUserData;external wlan_api_dll name
    'WlanGetProfileCustomUserData';
function WlanSetFilterList;external wlan_api_dll name 'WlanSetFilterList';
function WlanGetFilterList;external wlan_api_dll name 'WlanGetFilterList';
function WlanSetPsdiEDataList;external wlan_api_dll name
    'WlanSetPsdiEDataList';

```

```
function WlanSaveTemporaryProfile;external wlan_api_dll name
    'WlanSaveTemporaryProfile';
function WlanExtractPsdiEDataList;external wlan_api_dll name
    'WlanExtractPsdiEDataList';
function WlanReasonCodeToString;external wlan_api_dll name
    'WlanReasonCodeToString';
function WlanAllocateMemory;external wlan_api_dll name 'WlanAllocateMemory';
function WlanFreeMemory;external wlan_api_dll name 'WlanFreeMemory';
function WlanSetSecuritySettings;external wlan_api_dll name
    'WlanSetSecuritySettings';
function WlanGetSecuritySettings;external wlan_api_dll name
    'WlanGetSecuritySettings';
function WlanUIEditProfile;external wlan_api_dll name 'WlanUIEditProfile';
end.
```

Кафедра \_ КБПЗ \_ 2023 рік