

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  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за першим (бакалаврським) рівнем вищої освіти**  
на тему  
**“Програмне забезпечення системи інтелектуальної підтримки  
життєвого циклу з використанням ІоТ”**

Виконав здобувач вищої освіти  
IV курсу, групи КІ-21-ЗСК  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Малінніков В.І.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Коваленко А.С.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

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

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
д.т.н., проф.  
Олексій СМІРНОВ  
« 17 » січня 2024 року

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

*Маліннікову Владиславу Ігоровичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи інтелектуальної підтримки життєвого циклу з використанням IoT*

2. Керівник роботи *Коваленко Анна Степанівна, канд. техн. наук, доцент*

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

затверджені наказом вищого навчального закладу № 132-02 від 01.04.2024 року

3. Строк подання студентом роботи до захисту *23.05.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи інтелектуальної підтримки життєвого циклу з використанням IoT*

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

*1. Призначення та область використання.*

*2. Перегляд аналогічних існуючих систем.*

*3. Опис і обґрунтування проектних рішень.*

*4. Етапи програмування системи.*

*5. Впровадження системи в промислову експлуатацію.*

*6. Висновки*

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

*Структурна схема системи* *1 аркуш*

*Функціональна схема системи* *1 аркуш*

*Діаграма процесів* *1 аркуш*

*Блок-схема алгоритму роботи додатку* *2 аркуша*

7. Дата видачі завдання « 17 » січня 2024 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	10.03.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2024 р.	
3.	Розробка моделі компонента	20.03.2024 р.	
4.	Розробка структур даних	25.03.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2024 р.	
6.	Програмування алгоритмів	10.04.2024 р.	
7.	Оформлення ПЗ	17.04.2024 р.	
8.	Попередній захист роботи	23.05.2024 р.	

Дата видачі завдання  
« 17 » січня 2024 р.

Підпис керівника

Коваленко А.С.  
(прізвище та ініціали)

Завдання прийнято до виконання  
« 17 » січня 2024 р.

Підпис здобувача

Малінніков В.І.  
(прізвище та ініціали)

## АНОТАЦІЯ

**Малінніков В.І. Програмне забезпечення системи інтелектуальної підтримки життєвого циклу з використанням IoT. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.**

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтелектуальної підтримки життєвого циклу з використанням IoT.

Метою розробки є програмне забезпечення системи інтелектуальної підтримки життєвого циклу з використанням IoT.

Результат роботи – програмна реалізація системи інтелектуальної підтримки життєвого циклу з використанням IoT.

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 з ОС Windows 10/11.

Програму розроблено в середовищі Builder C++.

**Ключові слова:** комп'ютерна інженерія, IoT

## ABSTRACT

**Malinnikov V.I. Software of intelligent life cycle support system using IoT. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this final qualification work for the first (bachelor) level of higher education, software is developed, which is intended for the system of intellectual support of the life cycle using IoT.

The goal of the development is the software of the intelligent life cycle support system using IoT.

The result of the work is the software implementation of the intelligent life cycle support system using IoT.

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 a PC with Windows 10/11 OS.

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

**Keywords:** computer engineering, IoT

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	5
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	7
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	7
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	13
2.3 Розгорнута постановка завдання .....	15
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	17
3.1 Опис функціонування системи .....	17
3.2 Розробка структурної схеми.....	24
3.3 Розробка функціональної схеми .....	33
3.4 Розробка діаграми процесів.....	38
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	40
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	40
4.2 Захист розробленого програмного забезпечення.....	59
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	60
6 ОСНОВНІ ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	66

					ВКРБ-123.24.0053.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи інтелектуальної підтримки життєвого циклу з використанням IoT	Літ.	Аркуш	Аркушів
Розроб.	Малініков В.І.					Б	1	72
Перев.	Коваленко А.С.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-21-3СК		
Затв.	Смірнов О.А.							

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АСУ	–	автоматизована система управління
АЦП	–	аналогово-цифровий перетворювач
КПК	–	кишеньковий персональний комп'ютер
ПЗ	–	програмне забезпечення
ПЗО	–	пристрій зв'язку з об'єктом
ПК	–	персональний комп'ютер
СУБД	–	системи управління базами даних
ТП	–	технологічний процес
ЦОД	–	центр обробки даних
ІоТ	–	інтернет речей
НМІ	–	людино-машинний інтерфейс
SCADA	–	диспетчерське управління й збір даних
WAP		Wireless Application Protocol

КБПЗ – 2024

					ВКРБ-123.24.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

## ВСТУП

**Актуальність теми.** Інтернет речей – мабуть, сама обговорювана тема в галузі ІКТ. Уже ніхто не сумнівається в тому, що число й різноманіття підключених пристроїв будуть стрімко рости. Так, за прогнозом IDC, число пристроїв IoT збільшиться з 9,1 млрд в 2024 році до 28,1 млрд в 2025-м.

Розвитку Інтернету речей (Internet of Things, IoT) сприяє ряд обставин. Вартість електроніки стрімко падає, тому незабаром уже практично не залишиться пристроїв, які б не були наділені «інтелектом». При цьому потужності багатьох існуючих центрів обробки даних (ЦОД) завантажені менш ніж наполовину, що дає можливість розмістити додаткові ІТ-ресурси, необхідні для зберігання й обробки інформації, що збирається речами.

Правда, перешкод теж вистачає. Незважаючи на велике число різноманітних мережних технологій, у тому числі бездротових, вони поки не оптимізовані під Інтернет речей. Нові сервіси IoT зажадають зниження мережної затримки й підвищення надійності зв'язку, а також більшого часу автономної роботи пристроїв. У ряді критично важливих сервісів, наприклад для своєчасної автоматичної реакції автомобіля на дорожні умови, що змінюються, затримка не повинна перевищувати 1 мс. Такий рівень недосяжний навіть для систем 4G LTE: у них цей показник становить близько 10 мс. Тільки нові рішення 5G, які будуть доступні не раніше 2025 року, зможуть гарантувати мілісекундні затримки при мобільному зв'язку.

**Мета й завдання дослідження.** Метою роботи є програмне забезпечення системи інтелектуальної підтримки життєвого циклу з використанням IoT.

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

– Огляд існуючих систем інтелектуальної підтримки життєвого циклу з використанням IoT.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

– Дослідження системи інтелектуальної підтримки життєвого циклу з використанням IoT.

– Програмна реалізація системи інтелектуальної підтримки життєвого циклу з використанням IoT.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтелектуальної підтримки життєвого циклу з використанням IoT.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи інтелектуальної підтримки життєвого циклу з використанням IoT, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ\_2024

					ВКРБ-123.24.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Величезні ресурси, які виділяють провідні виробники на розробки в області IoT, дозволяють сподіватися, що всі технічні проблеми будуть вирішені. А тому експерти активно обговорюють, хто і як зможе заробити на Інтернеті речей.

Технології IoT будуть активно застосовуватися для організації післяпродажного сервісу шляхом створення автоматизованих систем керування виробами й системами. Він думає, що часи, коли вендори тільки роблять і продають свою продукцію, проходять. Сьогодні навіть консервативні замовники вимагають висновки сервісних контрактів на довгий час.

Якщо речі зможуть обробляти інформацію про власний стан, то їхні виробники одержать можливість автоматично управляти «рівнем сервісу» у контрактах повного життєвого циклу. Якщо ж речі додатково зможуть збирати інформацію про навколишнє середовище, то з'являється можливість створювати нові сервісні функції речей. Все це створює технологічну основу для переходу від бізнес-моделі продажу зробленого в статкуванні до моделі продажу послуг. Наприклад, коли компанія – виробник кондиціонерів буде продавати не сам агрегат, а те, для чого він потрібний, – холодне повітря.

## 1.2 Область застосування

Інтернет речей (IoT) – це найцікавіший мир, у якому зосереджені самі різні види діяльності – від навчання, досліджень і стандартизації до економічного планування. Загальноприйнятого визначення терміна "інтернет речей" не існує, однак він може розглядатися як можливість речей і людей дистанційно

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

взаємодіяти через інтернет у будь-якому місці й у будь-який час завдяки конвергенції різних технологій. У більшості випадків зв'язок "річ-річ" відбувається в сегменті зв'язку" компанія-компанія" (B2B), а зв'язок "річ-людин" – у сегменті зв'язку " компанія-клієнт" (B2C). Основне визначення МСЕ, опубліковане 4 липня 2012 року, забезпечує практичне розуміння й основу для подальшого аналізу й дослідження інтернету речей. Важливо відзначити, що відповідно до концепції МСЕ інтернет речей являє собою концепцію, а не одну певну технологію, і обумовлює "технічні й соціальні наслідки".

### **Створення економічної моделі**

На нашій планеті набагато більше речей, ніж людей, експерти називають самі різні цифри, оцінюючи число об'єктів, які можуть бути частиною інтернету речей. Неважливо, яке їхня точна кількість, їх дуже багато! Наприклад, по оцінках Internet Business Solutions Group, стратегічного відділу компанії Cisco, число підключених до інтернету пристроїв до 2024 року досягне 25 мільярдів, а до 2025 року – 50 мільярдів. До таких речей ставляться пристрої рухливого зв'язку, парковочні лічильники, термостати, кардіомонітори, автопокришки, дороги, автомобілі, полки супермаркетів і навіть велика рогата худоба.

Результати дослідження, проведеного раніше компанією Ericsson, також показали, що до 2025 року до інтернету буде підключено 50 мільярдів пристроїв, що зменшує масштаб і розмах знайомого нам світу інтернету й рухливого зв'язку. У ході дослідження концепції "інтернет усього", проведеного компанією Cisco, була розроблена економічна модель формування до 2022 року оцінюваного в 14,4 трильйони доларів США ринку, що припускає об'єднання в мережу практично всього.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи інтелектуальної підтримки життєвого циклу з використанням IoT, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

## 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

**Інтернет Речей (Internet of Things або IoT)** дозволяє підключати різні прилади повсякденного попиту до мережі Інтернет, збирати з них інформацію, віддалено управляти цими приладами через програмні додатки широкого кола споживчих пристроїв таких, як ПК, смартфони й смарт-ТБ. IoT забезпечує підвищення ефективності життєдіяльності людей, посилення їхньої безпеки, поліпшення навколишнього середовища або успішне досягнення особистих цілей, наприклад, таких як підтримка здорового способу життя.

Ринок споживчих пристроїв для Інтернету речей і M2M-комунікацій стрімко розвивається, впливаючи на багато індустрій і напрямки, такі як споживча техніка й електроніка, розважальні сервіси, системи ЖКГ і безпека, охорона здоров'я, спорт і фітнес.

#### **Споживчі пристрої для розумного будинку**

#### **Споживчі пристрої для ведення обліку споживання електрики, води й газу**

Смарт-лічильники або інтелектуальні крапки обліку дозволяють населенню контролювати й управляти витратами електрики, води й газу. Варто відзначити, що даний сегмент має високий потенціал розвитку в національному масштабі, основний внесок у який привносять державні програми, націлені на підвищення енергоефективності в країні.

Проте, на сьогоднішній день ринок споживчих пристроїв віддаленого онлайн моніторингу енергії, газу й води в Україні розвинений слабо. Поки

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

використовувані рішення більше покликані підвищити ефективність і знизити витрати постачальних компаній, ніж знизити витрати самих домогосподарств.

### **Рішення Smart Meters від British Gas**

Компанія **British Gas** уже пропонує рішення по моніторингу споживання ресурсів (smart meters) для своїх клієнтів. Більше того, компанія відзначає, що до 2025 року кожному жителеві буде запропонований варіант його установки. Смарт-лічильник дозволяє одержувати достовірну й точну інформацію про споживання електроенергії або газу щопівгодини, щодня, раз на місяць або раз у рік відповідно до бажання споживача. Через споживчий пристрій – смарт-монітор – користувач може контролювати витрати на газ і електрику, а також порівнювати споживання/витрати за різні періоди.

Споживчий пристрій оснащений бездротовою технологією передачі даних, за допомогою якої показання лічильника відправляються в компанію, а користувач може одержувати повідомлення про виставлений рахунок і про враховану витрату на енергію.

Даний підхід дозволяє уникнути некоректно виставлені рахунки від обслуговуючої компанії, а також дозволяє контролювати скільки електроенергії житель споживає, порівнювати свої метрики з показниками сусідніх будинків і квартир, а також одержувати персоналізовані підказки по більш ефективному споживанню на наданому моніторі.

### **Домашня техніка для Розумного будинку**

Самими популярними пристроями для розумного будинку, у яких зацікавлені великі гравці галузі, є наступні:

- Системи безпеки будинку із системою відеоспостереження, охоронно-пожежною сигналізацією, контролем і управлінням доступу.
- Інтелектуальні дверні замки, які запрограмовані на автоматичне відкриття при наближенні хазяїна будинку
- Термостати з датчиками руху, які дозволяють уникнути зайвих витрат на обігрівання порожньої кімнати.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

– Бездротові системи керування «Розумний будинок». Система автоматичного керування дозволяє контролювати відразу кілька споживчих пристроїв у режимі реального часу зі смартфона або планшетного ПК. Користувач може встановити автоматично керовану акустичну систему, разом із системою керування висвітленням, кліматичною системою, а також інтегрованою системою безпеки й контролювати всі складові з екрана свого мобільного пристрою.

– Розумні пристрійі, що служать також елементом декору. На ринок випускаються системи автоматизації й диспетчеризації, які можуть краще гармоніювати з декором, чим традиційні пристрої. Наприклад, розумні термостати, виконані в гарному дизайні, які естетично доповняють внутрішнє наповнення кімнати, а також інтелектуальні дверні замки, які стали виглядати більше сучасними й стильними.

Виробники споживчої техніки також випускають на ринок інтелектуальні версії вже популярних побутових приладів, таких як пральні машини, годинники, холодильники, духовки, пилососи, камери спостереження, кондиціонери, кофемашини, які полегшують і вносять розмаїтість у повсякденне життя користувача.

В Україні росте попит на пристрої для «Розумного будинку», однак основна аудиторія покупців і користувачів обмежена «дорожнечою» даних рішень. Багато користувачів не готові купувати інтелектуальні пристрої споживчої техніки й розумного будинку замість традиційних у зв'язку із ціновим фактором, однак у сегменті елітного міського й заміського житла дані рішення вже на сьогоднішній день користуються попитом.

Варто відзначити, що якщо для масової аудиторії споживчого ринку є критичного стримуючого фактора, то для сегмента корпоративної нерухомості дані рішення стають усе більше актуальними й мають великий потенціал розвитку.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

## **Рішення NestThermostat – розумний термостат**

Розумний термостат від компанії Nest дозволяє заощаджувати споживання енергії, а регулює комфортну для користувача температуру повітря в будинку. З 2011 року даний пристрій заощадив більше 4 млн кВтч енергії в мільйонах будинках по усьому світі. Відповідно до незалежних досліджень, на які посилається компанія Nest, їхнє рішення дозволяє заощадити в середньому 10-12% витрат на опалення й 15% на кондиціонування.

Коли користувач входить у приміщення, розумний термостат загоряється, показує встановлену температуру, яку можна поміняти зі смартфона або планшетного ПК, а також час. Пристрій саме регулює температуру за рівнем світла в приміщенні.

## **Рішення SamsungSmartFridge – Розумний холодильник**

Компанія Samsung пропонує розумний холодильник SamsungAppFridge, що повідомляє користувача про список продуктів для покупки, допомагає шукати рецепти, завантажувати фотографії, контролювати температуру, переглядати новини, слухати музику Pandora і навіть перевіряти Twitter.

## **Рішення LGSmart ThinQ™ 6.3 cu. – Розумна духовка**

Компанія LG розробила розумну духовку, що оснащена бездротовою технологією wi-fi і дозволяє регулювати час готування блюд, а також дивитися рецепти готування.

## **Рішення смарт-мультиварка REDMOND SKYCOKER M40S**

Одним з рішень Смарт-мультиварка REDMOND SKYCOKER M40S оснащена опцією дистанційного керування з мобільного пристрою через додаток ReadyforSky4S.

За допомогою даного рішення можна розраховувати калорійність блюд, запускати програми, створювати рецепти й зберігати в пам'яті, міняти температуру й час готування блюд.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

## **Споживчі пристрої для ConnectedCars**

Великою популярністю на ринку інтернету речей на транспорті користуються смарт-мітки для установки їх в автомобіль. Дані рішення дозволяють контролювати місцезнаходження автомобіля, швидкість руху, витрата палива, а також управляти пристроєм дистанційно.

В Україні основним драйвером поширення смарт-міток є страхові компанії, які зацікавлені в перегляді вартості полісів, залежно від реальних показників водіння клієнта. В Україні дані рішення на споживчому ринку поки не одержали належного поширення, однак внаслідок росту необхідності в підвищеній безпеці в економічно нестабільній країні, ситуація може змінитися.

### **Рішення Automatic – Смарт-адаптер для автомобіля**

Рішення від компанії Automatic дозволяє через мобільний додаток відслідковувати переборені кілометри, середня швидкість руху, витрата, а також місцезнаходження транспортного засобу.

Адаптер міститься в стандартний порт для діагностики транспортного засобу, тим самим пристрій підключається до бортового комп'ютера й може зчитувати дані про водіння, які за допомогою бездротової технології Bluetooth відправляються на екран смартфона через мобільний додаток.

### **Медичні високотехнологічні пристрої, які носяться**

Медичні високотехнологічні пристрої, які носяться, що, у перспективі до 2025 року витиснуть на сьогоднішній день саму значну категорію ринку – спортивні й фітнес пристрій за рахунок розширення асортиментів, розвитку електронної медицини й телемедицини, виходу нових гравців на ринок, а також зміни відносини до здоров'я з боку споживачів.

По даним J'son & Partners Consulting, ринок високотехнологічних пристроїв, що носяться, в Україні росте, зберігаючи позитивну динаміку, однак темпи росту даного ринку знижуються в найближчій перспективі, у зв'язку з ослабленням економіки України.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Для українського ринку високотехнологічних пристроїв, що носяться, характерна присутність великої кількості виробників, особливо китайських вендорів, а також українських розроблювачів, які зосередили особливу увагу на даному споживчому сегменті ринку.

Як і у світі, в Україні росте частка медичних смарт-пристроїв, а також смарт-годинників і камер, що носяться. Українські розроблювачі поряд зі світовими представниками націлені на інноваційний підхід у створенні нових споживчих пристроїв для ринку, щоб обійти конкуренцію або зайняти свою нішу на ринку.

### **Рішення AliveCorMobileECG – Мобільний вимірювальний пристрій для самостійного проведення ЕКГ**

Компанія AliveCor представила медичне рішення для контролю за серцевим ритмом і своєчасним виявленням серйозних серцевих проблем через ЕКГ.

Пацієнт повинен прикласти пристрій, який носить, The AliveCorMobileECG магнітною стороною до смартфона або планшетного ПК і помістити пальці на спеціально виділені металеві електродні пластини. Протягом 30 секунд пристрій вимірює серцевий ритм і миттєво транслює результати на мобільний додаток AliveECGApp.

На основі отриманих даних система інформує чи користувача виявлені аритмія або інші проблеми в даній області здоров'я. Рішення від AliveECG дозволяє стежити за симптомами й активністю пацієнта, які дозволяють йому й докторові виявити діагноз, визначивши, що впливає на збої в серцевому ритмі на картограмі ЕКГ.

### **Рішення ValedoBackTherapyKit – Пристрій, який носить, для лікування болю в спині**

Медичний смарт-пристрій від Valedo призначений для людей, що випробовують біль у нижній частині спини, які хочуть самостійно, а головне правильно займатися фізичними вправами будинку. Компанія запропонувала

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

рішення, що сполучає в собі розважальний ігровий елемент, а також пророблену лікарями фізичну терапію.

Користувачеві необхідно завантажити додаток Valedo App, приєднати до тіла два сенсори: один до грудної клітки, другий – до нижньої частини спини й почати тренування. Установлені сенсори через мережу Bluetooth допомагають виконувати вправи правильно, контролюючи процес тренування. Можна встановлювати до 4 різних профілів для використання рішення Valedo Back Therapy Kit.

Починаючи тренування, людина поринає в гру, де йому необхідно зібрати кристали, щоб перемогти й пройти на наступний рівень.

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

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

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

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

багато іншого. Можливо не тільки модифікувати поведінку існуючих компонентів, але і будувати нові.

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

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

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

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

Довідкова служба Builder C++ надавала мені допомогу в цій і багатьох інших подібних ситуаціях. Є повний опис кожного управляемого компонента,

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

включаючи списки властивостей і методів, а також численні приклади. Виклад матеріалу в книзі був значно покращуваний і систематизований завдяки відомостям, почерпнутим мною з довідкової служби.

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

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтелектуальної підтримки життєвого циклу з використанням IoT.

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 контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

в роботі технологічного обладнання;

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

КБПЗ - 2024

					ВКРБ-123.24.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Інтернет речей включає повсюдний обчислювальний процес, радіочастотну ідентифікацію (RFID), кіберфізичні системи, бездротові сенсорні мережі й міжмашинна взаємодія (M2M). Існують і інші кластери, які не розглядаються в справжній статті, наприклад орієнтовані на повсюдну комп'ютеризацію, автономні обчислення, взаємодію людина-машина, що оточує інтелект і, у цілому, на "розумні" об'єкти, системи й технології, але які теж по суті пов'язані з інтернетом речей.

В області повсюдної комп'ютеризації працює величезне число лабораторій і дослідницьких груп в усьому світі. На основі цієї концепції були розроблені програми, що охоплюють цілі країни, наприклад "u-Japan" (продовження програми "e-Japan") і "u-Korea".

#### Радіочастотна ідентифікація

Приблизно в 1998 році Санжей Сарма й Девід Брок з Массачусетського технологічного інституту (MIT) висунули ідею про те, щоб постачити всі об'єкти недорогими мітками RFID і з'єднати їх з інтернетом. Рішення про включення інтернету в архітектуру, очевидне сьогодні, у той час було стрибком у невідомість.

В 1999 році Рада по єдиному коду, Міжнародна асоціація по європейському номері товару (EAN International), компанії Procter & Gamble і Gillette ухвалили рішення щодо створенні Центра автоматичної ідентифікації (Auto-ID Center) на базі Массачусетського технологічного інституту. До складу команди дослідників (крім Сарми й Брока) увійшли Деніел Енджелс, Кай-Йен Сіу й Кевін Ештон, що став автором терміна "інтернет речей". Одне із завдань Центра полягала в розробці технології автоматичної ідентифікації, електронного коду

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

продукту (EPC) для заміни штрихкоду універсального товарного коду (UPC). Наприкінці жовтня 2003 року функції Центра автоматичної ідентифікації замінили Лабораторії автоматичної ідентифікації (Auto-ID Labs) і EPCglobal. Auto-ID Labs – це мережа семи університетів, розташованих на чотирьох континентах. Мережа EPCglobal стала спільним проектом організацій GS1 (раніше EAN International) і GS1 US (колишня Рада по єдиному коді). EPCglobal займається розробкою стандартів і керуванням мережею EPC.

Інтернет речей включає поряд з RFID широке коло інтерфейсів, однак багато з людей усе ще вважають, що по своїй суті інтернет речей базується на технології RFID і орієнтований на керування роздрібними мережами й ланцюжками поставок. У листопаді 2012 року Сарма оголосив про впровадження в Массачусетському технологічному інституті проекту "Хмара речей", розширивши сферу досліджень на базі RFID для включення хмарних обчислень і більших даних.

### **Кіберфізичні системи**

Концепції кіберфізичних систем і інтернету речей безсумнівно переплітаються. Приблизно в 2006 році Хелен Джилл із Національного наукового фонду Сполучених Штатів (NSF) висунула ідею про те, що "кіберфізичні системи – це фізичні, біологічні й спроектовані системи, операції яких інтегруються, контролюються й/або управляються комп'ютерним ядром. Компоненти поєднуються в мережу на всіх рівнях". Відповідно до її концепції "комп'ютеризація "глибоко убудована" в усі фізичні компоненти, можливо навіть у матеріали. Комп'ютерне ядро – це убудована система, що, як правило, вимагає реакції в режимі реального часу й найчастіше є розподіленою".

Ця нова концепція, повною мірою застосовувана на центральних комп'ютерах федеральних агентств США, включаючи Національний інститут по стандартах і технологіям (NIST), спричинила незабаром за собою відповідні дослідницькі проекти й ініціативи в сфері освіти. Наприклад, у червні 2009 року в Атланті (штат Джорджія) при фінансовій підтримці NSF була організована

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

перша літня школа кіберфізичних систем. Як і концепція повсюдної комп'ютеризації, що була використана Японією й Республікою Кореєю як технологічне керівництво, кіберфізичні системи стали опорною крапкою переходу до "розумного" виробництва в Німеччині. На думку агентства економічного розвитку Germany Trade & Invest (GTAI) "розумна" індустрія або "індустрія 4.0" – це технологічна еволюція від убудованих систем до кіберфізичних систем. GTAI заявляє про те, що індустрія 4.0 стане четвертою індустріальною революцією на шляху до інтернету речей, даних і послуг.

### **Бездротові сенсорні мережі**

Бездротові сенсорні мережі є однією з основних складових інтернету речей. Ця область має стійкі наукові, технологічні й промислові основи, а також безпосередньо пов'язана з інтернетом речей.

Як наочний приклад можна привести проект "Бездротові сенсорні мережі з відкритим вихідним кодом" (OpenWSN), що з 2010 року здійснюється Каліфорнійським університетом у Берклі з метою реалізації інтернету речей. OpenWSN служить як єдина база реалізацій з відкритим вихідним кодом пакетів протоколів на базі стандартів інтернету речей з використанням різних платформ програмного й апаратного забезпечення.

В усьому світі ведуться дослідження поширення сенсорних мереж на досить малі й молекулярні рівні. Такі дослідження, наприклад революційний проект Кріса Пістера з Каліфорнійського університету в Берклі, присвячений "розумного пилу" (скупчення незліченного числа малюсіньких мікроелектромеханічних систем), і робота Яна Акілдиза з Технологічного інституту Джорджії, присвячена інтернету наноречей, відкривають дорогу до майбутнього образу інтернету речей.

### **Міжмашинна взаємодія**

Міжмашинна взаємодія (M2M), можливо, є найпершим втіленням інтернету речей. Відповідно до визначення міжнародного партнерства найбільших організацій по розробці стандартів, названого one2M, рішення

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

M2M – це "сполучення пристроїв, програм і послуг, що функціонують із мінімальною участю або без участі людини".

Прототипами M2M можуть по праву вважатися передові технології передачі даних, такі як базові системи телеметрії й галузевих систем керування. Телеметричні послуги, надавані Mobitex – низькошвидкісною бездротовою мережею передачі коротких повідомлень із пакетною комутацією, розробленої на початку 1980-х років шведською компанією Televerket (попередником Telia Sonera), а пізніше в партнерстві з Ericsson, це одні з перших технологій, які орієнтовані безпосередньо на потребі ринку, що зароджується, M2M.

З роками технологія M2M одержала розвиток у напрямку дистанційного моніторингу й керування. Останнім часом завдяки технології M2M почали створюватися платформи, що дозволяють інтегрувати архітектуру побудови мереж рухливий і/або фіксованої, провідний і/або бездротового зв'язку (наприклад, персональних бездротових мереж) і послуги стільникового й супутникового зв'язку (у тому числі послуги глобальної супутникової системи визначення місця розташування). За своїм характером, M2M забезпечує взаємодія "річ-річ" і орієнтована на ринок зв'язку "компанія-компанія". Це важлива рушійна сила реалізації інтернету речей.

Протягом останніх п'яти років в усьому світі росте інтерес до інтернету речей. З'явилися наукові журнали, присвячені безпосередньо даній темі, у світі проводяться приваблюючу увагу міжнародні форуми, конгреси й саміти, присвячені інтернету речей, а у світових ЗМІ регулярно обговорюються наслідки появи інтернету речей і його вплив на соціальні перетворення.

Прикладом важливої ролі інтернету речей як рушійної сили перетворень і розвитку є включення інтернету речей у п'ятирічний план Китаю в якості національного пріоритетного стратегічного напрямку. У плані інтернет речей визначається як ключовий напрямок нового покоління інновацій і розвитку інформаційних технологій Китаю. Сьогодні в багатьох університетах Китаю студенти можуть одержати ступінь бакалавра по проектуванню й розробкам

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

в області інтернету речей. Будівництво "розумних" міст – ще одна сфера, у якій Китай планує використовувати додатки інтернету речей для підвищення взаємозв'язку й ефективності інфраструктури й послуг.

Іншим прикладом може служити Європейський союз. У червні 2010 року Європейський парламент прийняв резолюцію, присвячену інтернету речей, у яку увійшли багато пов'язані з ним питання й плановані дії, у тому числі "думка про те, що розвиток інтернету речей і пов'язаних з ним додатків вплине на повсякденне життя європейців і їхній спосіб життя в наступні роки, що приведе до значних змін в економічній і соціальній сфері".

### **Короткий огляд діяльності в сфері стандартизації**

На сьогоднішній день робота в області стандартизації є каталізатором обговорення концепції інтернету речей у світі. МСЕ відіграє ключову роль у стандартизації в даній сфері. МСЕ здійснює це через свою Групу по спільній координаційній діяльності в області інтернету речей (JCA-IoT), Глобальну ініціативу по стандартах інтернету речей (IoT-GSI) і Оперативну групу за рівнем обслуговування при M2M. JCA-IoT сприяє багатобічному співробітництву з іншими організаціями по розробці стандартів і забезпечує відсутність дублювання роботи. IoT-GSI є головною структурою для розробки стандартів в області інтернету речей у всесвітньому масштабі. Крім того, у вересні 2010 року в Пекіні (Китай) з ініціативи Глобального співробітництва по стандартах була створена цільова група по стандартизації в області M2M (GSC MSTF) для сприяння узгодженню стандартів.

Інші великі міжнародні організації по розробці стандартів також реалізують проекти стандартизації інтернету речей. До них ставляться Міжнародна організація по стандартизації (ISO), Міжнародна електротехнічна комісія (МЕК) і Цільова група по інженерних проблемах інтернету (IETF), що тісно співробітничує з Консорціумом World Wide Web (W3C), ISO й МЕК.

Результатом діяльності IETF є створення пакета протоколів, які можуть забезпечити реалізацію функціонально сумісного інтернету речей. Відповідні

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

протоколи IETF включають протокол IPv6 для персональних бездротових мереж малої потужності, протокол IPv6 для мереж з малою потужністю й втратами, а також протокол обмеженого застосування.

І звичайно, фундаментальні дослідження веде велику кількість комітетів по стандартах Інституту інженерів по електроніці й електротехніці (IEEE), без яких була б неможлива ефективна реалізація інтернету речей.

Крім того, у рамках організацій по розробці стандартів у Канаді, Китаї, країнах Європи, Індії, Японії, Республіці Корея й США реалізуються важливі національні проекти в даній сфері. Частина виконуваної ними роботи в області стандартизації M2M була передана партнерству one2M, утвореному в липні 2012 року.

В обговоренні стандартів, пов'язаних з інтернетом речей, також беруть участь аналогічні організації, діяльність яких сконцентрована на вертикальних і регіональних ринках. Серед них можна відзначити роботу Continua Health Alliance, Американської асоціації телемедицини, European Research Cluster on the Internet of Things, European Internet of Things Architecture, European Internet of Things Initiative, а також інших груп по проведенню досліджень і розробці стандартів у сфері інтелектуальних транспортних систем, "розумних" електромереж, "розумного" виробництва, забезпечення ланцюжка поставок і т.д.

Рух прихильників програмного забезпечення з відкритим вихідним кодом також сприяє створенню значимих стандартів і протоколів стосовно до інтернету речей. Це відбувається завдяки роботі таких груп і організацій, як, наприклад, Співтовариство Contiki, Робоча група Eclipse по M2M, Організація по розвитку стандартів структурованої інформації (OASIS), Альянс TinyOS, недавно створеної організації Інтернет речей з відкритим вихідним кодом у Каліфорнії (Сполучені Штати) і Рішення з відкритим вихідним кодом для інтернету речей у хмарі (Європа), а також апаратних обчислювальних платформ із відкритим вихідним кодом (таких, як Arduino).

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Що стосується технологій передачі повідомлень і побудови мереж, заснованих на стандартах, таких як технологія передачі телеметричних повідомлень по черзі (MQTT), удосконалений протокол передачі повідомлень по черзі (AMQP), служба розподілу даних (DDS) і мережні специфікації "білого простору", те викликає інтерес їхня здатність виступати як стандарти для M2M і інтернет речей.

Однак ми розповіли далеко не про всю діяльність, згадаєте всі створені останнім часом торговельні групи з питань інтернету речей і M2M, а також постійну просвітительську роботу Ради по інтернету речей. Також необхідно відзначити реалізацію корпоративних ініціатив, у рамках яких досліджуються перспективні напрямки, такі як "центральна нервова система планети Земля", "цифрове життя", "промисловий інтернет", "інтернет усього", "інтернет речей і послуг", "інтернет речей і датчиків і приводів", "інтелектуальна планета" і "соціальна мережа речей".

У цьому розділі ми не торкалися найважливіших аспектів інтернету речей, такі як безпека, конфіденційність і довіра. Ці питання вивчаються, наприклад, Європейською комісією в рамках Цифрового порядку денного для Європи, Федеральною торговельною комісією США й, більш узагальнено, МСЕ (у рамках дорожньої карти по стандартах безпеки ІКТ). Більші дані й хмарні обчислення також пов'язані з інтернетом речей, однак у даній статті ці зв'язки не розглядаються.

Проте існування безлічі що переплітаються екосистем зі схожими цілями й проблемами є доказом важливості соціальних змін, пов'язаних з інтернетом речей. Він несе в собі масштабні перетворення, які необхідно глибоко розуміти, планувати, органічно й ефективно інтегрувати в соціально-економічну систему в благо людства.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

## 3.2 Розробка структурної схеми

### Інтернет речей і розумні будинки

Для споживачів «розумний дім» – це, мабуть, місце, де вони, швидше за все, стикаються з речами, що підтримують Інтернет, і це одна сфера, де великі технологічні компанії (зокрема Amazon, Google і Apple) жорстко конкурують.

Найбільш очевидними з них є розумні динаміки, такі як Echo від Amazon, але є також розумні розетки, лампочки, камери, термостати та смарт-холодильник, який часто висміюють. Але крім того, що ви демонструєте свій ентузіазм щодо нових блискучих гаджетів, у програм для розумного дому є й серйозніша сторона. Вони можуть допомогти людям похилого віку довше залишатися незалежними та залишатися у власних домівках, спростивши спілкування сім'ї та опікунів із ними та спостерігаючи за тим, як вони живуть. Краще розуміння того, як працюють наші домівки, і можливість налаштувати ці налаштування може допомогти заощадити енергію, наприклад, скоротивши витрати на опалення.

### Інтернет речей і аналітика великих даних

IoT генерує величезну кількість даних: від датчиків, прикріплених до деталей машин, або датчиків навколишнього середовища, або слів, які ми кричимо в наші розумні колонки.

Ці дані надходять у багатьох різних формах – голосові запити, відео, показання температури чи інших датчиків, усі з яких можна отримати для аналізу. Як зазначає аналітик IDC, категорія метаданих IoT є зростаючим джерелом даних, якими потрібно керувати та використовувати. «Метадані є основним кандидатом для введення в бази даних NoSQL, такі як MongoDB, щоб структурувати неструктурований вміст або вводити в когнітивні системи, щоб привнести нові рівні розуміння, інтелекту та порядку в зовнішньо випадкові середовища», – йдеться в повідомленні.

Зокрема, IoT забезпечить передачу великих обсягів даних у реальному

					ВКРБ-123.24.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

часі. За підрахунками Cisco, міжмашинні з'єднання, які підтримують додатки IoT, становитимуть більше половини із загальних 27,1 мільярдів пристроїв і з'єднань, і до 2025 року становитимуть 5% глобального IP-трафіку.

### **Як Інтернет речей і 5G підключаються та обмінюються даними?**

Пристрої IoT використовують різноманітні методи підключення та обміну даними, хоча більшість з них використовуватиме певну форму бездротового підключення: будинки та офіси використовуватимуть стандартний Wi-Fi, Zigbee або Bluetooth Low Energy (або навіть Ethernet, якщо вони не дуже мобільні); інші пристрої використовуватимуть LTE (існуючі технології включають вузькосмуговий IoT і LTE-M, в основному призначені для невеликих пристроїв, які надсилають обмежені обсяги даних) або навіть супутникові з'єднання для зв'язку. Однак величезна кількість різних варіантів вже змусила декого стверджувати, що комунікаційні стандарти IoT мають бути такими ж прийнятними та сумісними, як сьогодні Wi-Fi.

Однією з сфер зростання в найближчі кілька років, безсумнівно, стане використання мереж 5G для підтримки проектів IoT. 5G пропонує можливість розмістити до одного мільйона пристроїв 5G на квадратному кілометрі, що означає, що можна буде використовувати величезну кількість датчиків на дуже маленькій території, що робить більш можливим масштабне промислове розгортання IoT. Великобританія щойно розпочала випробування 5G та IoT на двох «розумних фабриках». Однак може пройти деякий час, перш ніж розгортання 5G стане широким розповсюдженням: Ericsson прогнозує, що до 2025 року буде близько п'яти мільярдів пристроїв IoT, підключених до стільникових мереж, але лише близько чверті з них буде широкосмуговим IoT, причому 4G підключатиме більшість ті.

За даними Gartner, камери зовнішнього спостереження стануть найбільшим ринком пристроїв 5G IoT у найближчій перспективі, на них припадатиме більшість (70%) пристроїв 5G IoT цього року, а до кінця 2024 року їх частка впаде приблизно до 30%. в якій точці їх наздоженуть сполучені машини.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Аналітична фірма прогнозує, що цього року буде використовуватися 3,5 мільйона пристроїв 5G IoT, а до 2025 року – майже 50 мільйонів. За прогнозами компанії, у довгостроковій перспективі автомобільна промисловість буде найбільшим сектором для випадків використання 5G IoT.

Одна ймовірна тенденція полягає в тому, що з розвитком IoT може статися, що менше даних буде надсилатися для обробки в хмарі. Щоб знизити витрати, більше обробки можна здійснювати на пристрої, а лише корисні дані надсилати назад у хмару – стратегія, відома як «граничні обчислення». Для цього знадобляться нові технології, такі як захищені від несанкціонованого доступу периферійні сервери, які можуть збирати й аналізувати дані далеко від хмари чи корпоративного центру обробки даних.

### **Промисловий Інтернет речей**

Промисловий Інтернет речей (IIoT) стосується розширення та використання Інтернету речей (IoT) у промислових секторах і додатках. Завдяки сильному фокусу на зв'язку між машинами (M2M), великим даним і машинному навчанню, IIoT дозволяє галузям і підприємствам підвищити ефективність і надійність своїх операцій. IIoT охоплює промислові програми, включаючи робототехніку, медичні пристрої та програмно-визначені виробничі процеси.

Інтернет речей виходить за рамки звичайних споживчих пристроїв і мережевого зв'язку фізичних пристроїв, які зазвичай пов'язані з Інтернетом речей. Що робить його відмінним, так це перетин інформаційних технологій (IT) і операційних технологій (OT). OT відноситься до об'єднання в мережу операційних процесів і промислових систем управління (ICS), включаючи людино-машинні інтерфейси (HMI), системи диспетчерського керування та збору даних (SCADA), розподілені системи управління (DCS) і програмовані логічні контролери (PLC). Конвергенція IT і OT забезпечує галузям більшу системну інтеграцію з точки зору автоматизації та оптимізації, а також кращу видимість ланцюжка поставок і логістики. Моніторинг і контроль фізичної інфраструктури в промислових операціях, таких як сільське господарство, охорона здоров'я,

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

виробництво, транспорт і комунальні послуги, полегшуються завдяки використанню інтелектуальних датчиків і виконавчих механізмів, а також віддаленого доступу та контролю. У контексті четвертої промислової революції, яка отримала назву «Індустрія 4.0», ІоТ є невід'ємною частиною трансформації кіберфізичних систем і виробничих процесів за допомогою великих даних і аналітики. Дані в режимі реального часу з датчиків та інших джерел інформації допомагають промисловим пристроям та інфраструктурі «приймати рішення», виробляти ідеї та конкретні дії. Крім того, машини можуть виконувати й автоматизувати завдання, з якими попередні промислові революції не могли впоратися. У ширшому контексті ІоТ має вирішальне значення для використання випадків, пов'язаних із пов'язаними екосистемами чи середовищами, наприклад, як міста стають розумними містами, а фабрики – розумними фабриками.

Послідовний збір і передача даних між розумними пристроями та машинами надає галузям і підприємствам багато можливостей для зростання. Дані дозволяють галузям і підприємствам виявляти помилки або неефективність, наприклад, у ланцюжку постачання, і негайно виправляти їх, таким чином сприяючи повсякденній ефективності операцій і фінансів. Належна інтеграція ІоТ може також оптимізувати використання активів, передбачити точки збою та навіть автономно запускати процеси обслуговування.

Використовуючи підключені та інтелектуальні пристрої, компанії можуть збирати та аналізувати більші обсяги даних із більшою швидкістю. Це не тільки підвищить масштабованість і продуктивність, але також може подолати розрив між виробничими поверхами та загальними офісами. Інтеграція ІоТ може дати промисловим підприємствам більш точне уявлення про те, як просуваються їхні операції, і допомогти їм приймати обґрунтовані бізнес-рішення.

### **Безпека та виклики, які виникають під час впровадження ІоТ**

Прийняття ІоТ може революціонізувати роботу галузей, але існує проблема, пов'язана з розробкою стратегій для посилення зусиль з цифрової трансформації, зберігаючи при цьому безпеку на тлі розширеного зв'язку.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Можна очікувати, що галузі та підприємства, які працюють з оперативними технологіями, будуть добре обізнані в таких аспектах, як безпека працівників і якість продукції. Однак, враховуючи те, що ОТ інтегрується в Інтернет, організації бачать впровадження більш інтелектуальних і автоматизованих машин у роботі, що, у свою чергу, запрошує низку нових викликів, які вимагають розуміння внутрішньої роботи ПоТ.

При реалізації ПоТ необхідно зосередитися на трьох сферах: доступності, масштабованості та безпеці. Доступність і масштабованість вже можуть бути другою натурою промислових операцій, оскільки вони вже можуть бути встановлені або працювати в бізнесі протягом досить тривалого часу. Проте багато хто може спіткнутися під час інтеграції ПоТ у свої операції з безпекою. По-перше, багато компаній досі використовують застарілі системи та процеси. Багато з них функціонують десятиліттями і тому залишаються незмінними, що ускладнює впровадження нових технологій.

Крім того, поширення інтелектуальних пристроїв породило вразливість системи безпеки та занепокоєння щодо підзвітності безпеки. Прихильники Інтернету речей несуть фактичну відповідальність за забезпечення налаштування та використання своїх підключених пристроїв, але виробники пристроїв зобов'язані захищати своїх споживачів, коли вони випускають свої продукти. Виробники повинні мати можливість гарантувати безпеку користувачів і вживати профілактичних заходів або виправлення у разі виникнення проблем із безпекою.

Навіть більше, потреба в кібербезпеці висувається на перший план, оскільки з роками з'являються все більш серйозні інциденти безпеки. Хакери, які отримують доступ до підключених систем, означають не лише наражати бізнес на серйозний злом, але також означають потенційне припинення роботи. До певної міри галузі та підприємства, які впроваджують ПоТ, повинні планувати та працювати як технологічні компанії, щоб безпечно керувати як фізичними, так і цифровими компонентами.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Користувачі також стикаються з проблемою правильної інтеграції промислових операцій з ІТ, де як з'єднання, так і інформація повинні бути захищені. Дані користувачів слід обробляти відповідно до чинних норм конфіденційності, таких як Загальний регламент захисту даних (GDPR) Європейського Союзу (ЄС) . Хоча зібрані дані відіграють важливу роль у створенні розуміння для пристроїв та інфраструктури, вкрай важливо, щоб особиста інформація була відокремлена від загальних даних журналу. Таку інформацію, як ідентифікаційна інформація (PII), слід зберігати в зашифрованій базі даних. Зберігання незашифрованої інформації разом з іншою відповідною діяльністю в хмарі може означати, що підприємства ризикують стати жертвою.

Однією з головних проблем, пов'язаних з ІоТ, є технологічна фрагментація, і ПоТ, відповідно, не є винятком із співіснування різних стандартів, протоколів і архітектур. Різне використання в системах ПоТ, наприклад, стандартів і протоколів, таких як Message Queuing Telemetry Transport (MQTT) і Constrained Application Protocol (CoAP), може перешкоджати сумісності систем ПоТ.

### **Ризики для систем ПоТ**

Багато проблем безпеки, пов'язаних з ПоТ, виникають через відсутність базових заходів безпеки. Прогалини в безпеці, як-от відкриті порти, неадекватні методи автентифікації та застарілі програми, сприяють виникненню ризиків. Поєднайте це з прямим підключенням мережі до Інтернету, і ви отримаєте більше потенційних ризиків.

Компанії, можливо, вже знайомі з можливим впливом на бізнес відмова ІТ-систем через кіберзлочинність або зараження зловмисним програмним забезпеченням. Однак конвергенція ІТ та ОТ створює новий значний фактор ризику: реальні загрози, які можуть вплинути навіть на цивільних.

Незахищені системи ПоТ можуть призвести до збоїв у роботі та грошових втрат, серед інших значних наслідків. Більше підключених середовищ означає більше ризиків для безпеки, як-от:

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- Уразливості програмного забезпечення, які можна використати для атаки на системи.
- Загальнодоступні пристрої та системи, підключені до Інтернету.
- Зловмисна діяльність, як-от хакерство, цілеспрямовані атаки та витік даних.
- Маніпуляції з системою, які можуть спричинити порушення роботи (наприклад, відкликання продукції) або саботаж процесів (наприклад, зупинка виробничої лінії).
- Збій у роботі системи, який може призвести до пошкодження пристроїв і фізичних об'єктів або травмування операторів або людей поблизу.
- Системи ОТ, утримувані для вимагання, як скомпрометовані через ІТ-середовище.

Сумнозвісним прикладом скомпрометації системи ОТ через ІТ-середовище є кібератака на енергомережу в Україні в грудні 2015 року, коли зловмисник зміг заразити ІТ-інфраструктуру, вимкнувши критичні системи та порушивши електропостачання в тисячах домогосподарств.

У той час як підвищення продуктивності операцій є важливим для систем ПоТ, безпека повинна розглядатися як важлива. Підключення ОТ до Інтернету може зробити бізнес більш життєздатним за допомогою багатьох датчиків і підключених пристроїв у роботі та даних у реальному часі, які вони генерують. Але якщо не інвестувати в кібербезпеку, це може звести нанівець переваги. Саме тут повинні з'явитися підходи до безпеки за проектом і вбудованої безпеки.

Наявність центру безпеки (SOC) має вирішальне значення для проактивного моніторингу та захисту від широкого спектру загроз, які впливають на підключене середовище. Цей централізований підрозділ дозволяє галузям і підприємствам контролювати значну кількість сповіщень, з якими вони можуть зіткнутися, і забезпечувати швидке реагування. SOCс особливо корисні для об'єктів, які потребують кращої видимості та постійного аналізу стану безпеки. Метою команд SOC є виявлення інцидентів із безпекою чи будь-якої аномальної

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

активності та можливість негайного вирішення проблем до того, як може статися будь-який компромет. Цей підхід усуває проблеми, які можуть виникнути із застарілими системами, низькою видимістю системи та повільним часом відгуку. Завдяки SOC сповіщенням буде визначено пріоритет, а кореляція загроз буде більш оптимізованою, щоб підприємства могли керувати як ІТ, так і ОТ.

Однак зміни в ландшафті загроз, а також промислових інфраструктур вимагають від організацій адаптувати свій захист до нових і невідомих загроз, з якими вони можуть зіткнутися. Прихильники ІоТ могли б зробити акцент на наявності спеціальної команди для вирішення питань безпеки в середовищі ОТ, враховуючи, що це спеціалізована сфера. Наймання експертів із безпеки, які можуть розуміти різні типи загроз і вживати швидких заходів для пом'якшення наслідків атак, має бути найголовнішим для галузей і підприємств, якщо вони хочуть процвітати в умовах конвергенції ІТ/ ОТ .

Наявність повного набору засобів захисту, спеціально вбудованих у різні рівні реалізацій Інтернету речей, дозволить галузям і підприємствам безпечно виконувати свої операції. Ці рівні безпеки включають пристрій, мережу та хмару.

Рівень пристроїв зазвичай складається з пристроїв і додатків ІоТ, які надходять від різних виробників і постачальників послуг. Прихильники Інтернету речей повинні знати, як їхні виробники та постачальники послуг передають і зберігають дані. У разі виникнення проблеми з безпекою виробники та постачальники послуг також повинні мати можливість активно повідомляти підприємства про те, про що потрібно подбати.

У зоні мережі є шлюз, який збирає дані з пристроїв. Це та частина, де організації повинні мати системи запобігання вторгнень наступного покоління (IPS), щоб вони могли контролювати та виявляти потенційні атаки. Шлюз також є місцем, де зазвичай є центр керування, який видає команди різним пристроям. Центр керування — це найважливіше місце, де організації повинні запровадити посилення безпеки, щоб забезпечити захист від зараження зловмисним програмним забезпеченням або отримання контролю над ним зловмисників.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

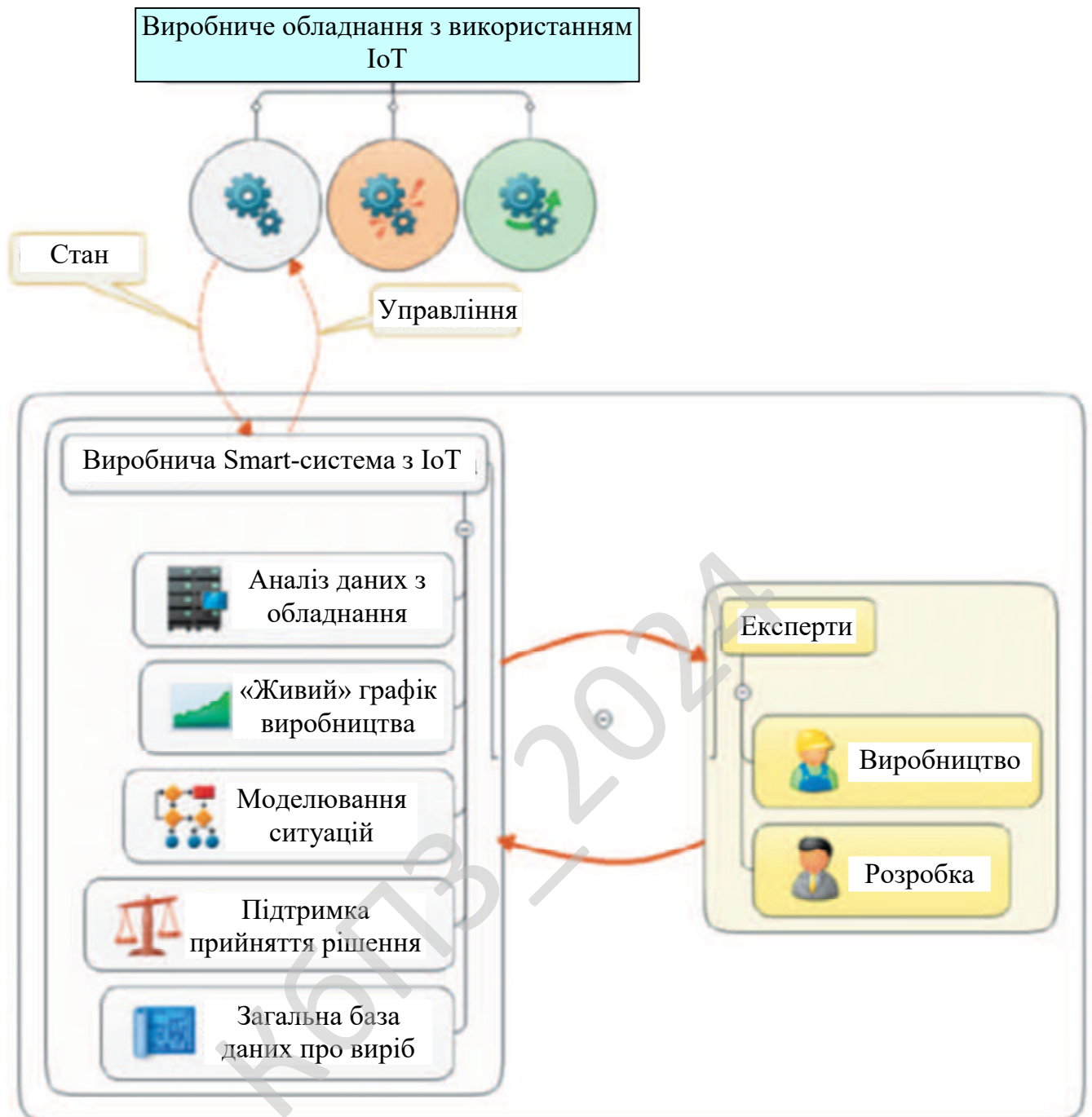


Рисунок 3.1 – Структурна схема системи

Нарешті, хмара – це місце, де постачальники повинні мати впровадження безпеки, які запускають захист на основі сервера, щоб зменшити ризик використання хакерами серверів і збережених даних. Це ще раз підтверджує стурбованість тим, що організації підлягають відповідним покаранням щодо захисту даних.

Таким чином, захист систем ІоТ потребує підключеного захисту від загроз і наскрізного захисту від шлюзу до кінцевої точки, які можуть забезпечити:

- Регулярний моніторинг і виявлення зараження шкідливими програмами.
- Краща видимість загроз і раннє виявлення аномалій.
- Проактивне запобігання загрозам і атакам між ІТ і ОТ.
- Безпечна передача даних.
- IPS наступного покоління для запобігання атакам з використанням

вразливостей.

- Захист серверів і програм у центрі обробки даних і хмарі.

У роботі реалізована система керування проектами й виробництвом виробів. По суті, це означає створення «розумного» виробництва. Ключовою частиною повинна стати база знань, де будуть накопичуватися знання про організацію й порядок проведення робіт, номенклатурі виробів, підрозділах і співробітниках, результатах випробувань і т.д.

Концепція інтелектуального керування в реальному часі буде поширена на всі інші процеси життєвого циклу, включаючи експлуатацію. Тут ключову роль буде грати кіберфізична модель виробу (КФМ) – наприклад, повна віртуальна модель. Подібне дзеркалювання стану допоможе приймати максимально ефективні управлінські рішення, наприклад, моделюючи різні події.

### 3.3 Розробка функціональної схеми

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

Для ефективної й правильної розробки інноваційної стратегії компанії необхідна комплексна характеристика життєвого циклу виробленої продукції.

Життєвий цикл виробу складається з ряду стадій, на яких ідея трансформуються в нову техніку, здатну задовольнити вимоги споживачів.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Початковою стадією життєвого циклу є науково-дослідні роботи, які проводяться по єдиному технічному завданню.

Науково-дослідна робота складається з наступних етапів:

- розробка технічного завдання;
- вибір напрямків досліджень;
- теоретичні й експериментальні дослідження;
- узагальнення й оцінка результатів науково-дослідних робіт.

Технічне завдання визначає мету, зміст, порядок виконання робіт і спосіб реалізації результатів науково-дослідних робіт і є обов'язковим документом для початку робіт.

Закінчена науково-дослідна робота обговорюється на науково-технічній раді або відповідній секції, на якому розглядається відповідність виконаних робіт прийнятому технічному завданню, обґрунтованість висновків і рекомендацій і виноситься рішення про продовження роботи на наступних стадіях життєвого циклу.

Слід зазначити, що вирішальний вплив на створення нововведення робить рівень наукового забезпечення. Саме на етапі наукових досліджень заставляється потенціал нововведення, що матеріалізується через проектно-конструкторські розробки й виробництво.

Другою стадією життєвого циклу є дослідно-конструкторські роботи. На цій стадії розробляється конструкторська документація: технічна пропозиція, ескізний проект, технічний проект, робоча конструкторська документація. Роботи проводяться також для створення технологічного встаткування, потрібного для виготовлення досвідчених зразків і партій виробів.

Розробка виробу завершується після усунення недоробок по зауваженнях приймальної комісії й твердження акту приймання досвідченого зразка, партії. До складу приймальної комісії можуть входити представники компанії-розроблювача, компанію-виробника й компанію-споживача.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Наступними стадіями життєвого циклу є підготовка виробництва й вихід на потужність, тобто постановка продукції на виробництво. Ці стадії включають заходу щодо організації виробництва нового виробу або освоєного іншими компаніями.

Вихід на потужність відбувається після завершення робіт по підготовці виробництва:

- пуск і перевірка технологічного встаткування;
- запуск у виробництво настановної серії;
- проведення кваліфікаційних випробувань виробів настановної серії;
- доробка й коректування технологічної й іншої документації.

Настановна серія або перша промислова партія виробів виготовляється для перевірки здатності даного виробництва забезпечити промисловий випуск продукції відповідно до вимог науково-технічної документації й споживачів.

Зразки настановної партії, що пройшли приймально-здавальні й кваліфікаційні випробування, можуть бути представлені на ринку нововведень (проведення рекламної кампанії, демонстрація на виставках, торгових центрах і т.п.).

Всі розглянуті вище стадії життєвого циклу зуться передвиробничими. Тут формується виріб і його якість, а також заставляється технічний рівень виробу і його прогресивність.

Компанія повинна контролювати строки передвиробничих стадій, щоб уникнути їхнього розтягування в часі, інакше до стадії виробництва можуть дійти застарілі розробки. Ця проблема досить актуальна для багатьох наукових розробок, здійснюваних в Україні.

Наступною стадією життєвого циклу є виробництво створеного виробу у відповідності зі сформованим портфелем замовлень.

Завершальною стадією життєвого циклу є експлуатація (для виробів тривалого користування) або споживання (для сировини, палива й т.п.)

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

замовником або споживачем, що використовує дану продукцію по призначенню або як комплектуючі вироби при виробництві іншої продукції.

Взаємини між споживачем і виробником продукції визначається договором на поставку.

Важливо забезпечити систематичне відновлення продукції за рахунок випуску нових виробів і зняття з виробництва застарілих.

Тривалість життєвого циклу в кожний конкретний період науково-технічного прогресу визначається фізичним і моральним строком старіння техніки незалежно від строків виконання й організації робіт зі стадій життєвого циклу й усередині них по етапах.

Важливим завданням для виробництва, є перехід на економіку «реального часу», що дасть можливість в оперативному режимі відслідковувати основні параметри всіх проектів: що, де і як. Збір даних у реальному часі дозволить не допустити виникнення аварії й навіть позаштатної ситуації завдяки проактивному керуванню системою або проектом.

У роботі реалізована інтелектуальна система керування реальним часом підтримки життєвого циклу виробів: від збору вимог і розробки, через виробництво й експлуатацію, аж до утилізації. Це мережецентрична «система систем», у якій різні підсистеми зможуть узгоджено взаємодіяти за принципом «кожний з кожним» (P2P) через загальну інформаційну шину. Це забезпечить ту саму взаємодію в реальному часі, що неможливо реалізувати в традиційних ієрархічних системах з більшим числом рівнів керування.

					ВКРБ-123.24.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36





Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

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.

КБПЗ\_2024

					ВКРБ-123.24.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

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

### 4.1 Блок-схеми та опис алгоритмів функціонування системи

Під час роботи над бакалаврською дипломною роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>40</b>



програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

UML необхідний:

– Керівникам проектів, які керують розподілом завдань і контролем за проектом.

– Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.

– Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.

– Програмістам які реалізують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Розглянемо API розробленої системи яке являє собою дерево об'єктів системи, кожний об'єкт якого може представляти власний перелік властивостей і функцій. Властивості й функції об'єктів можуть використовуватися користувачем у процедурах.

Крапкою входу для доступу до об'єктів системи є зарезервоване слово "SYS" кореневого об'єкта. Наприклад, для доступу до функції вихідного транспорту потрібно записати: `SYS.Transport.Serial.out_ModBus.messIO(mess);`.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

API об'єктів, надаваних модулями, описується у власній документації модуля.

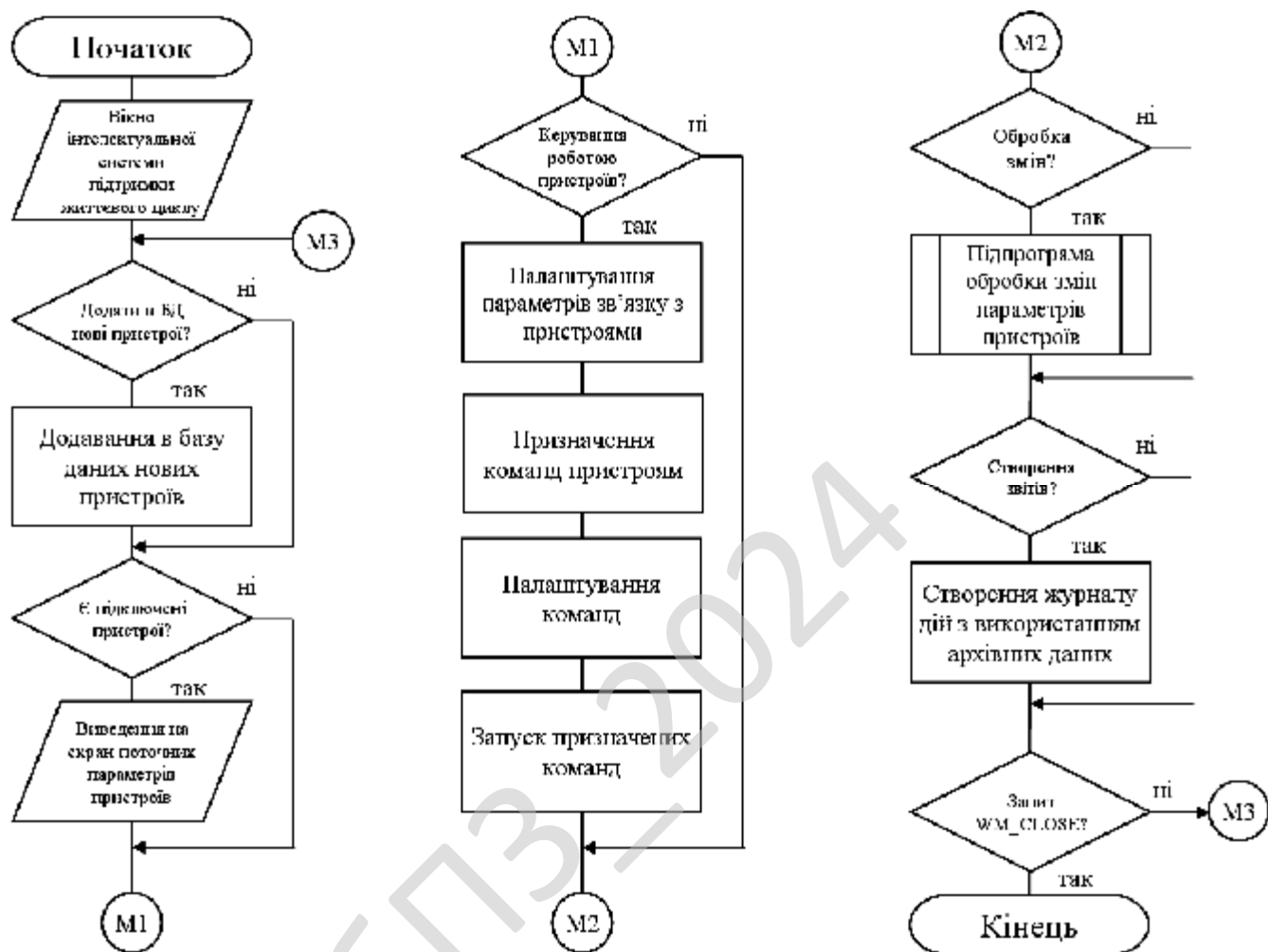


Рисунок 4.1 – Блок-схема основної програми

Абстрактний об'єкт являє собою асоціативний контейнер властивостей і функцій. Властивості можуть містити як дані чотирьох базових типів, так і інші об'єкти. Доступ до властивостей об'єкта звичайно здійснюється за допомогою запису імен властивостей через крапку до об'єкта `<obj.prop>`, а також за допомогою висновку ім'я властивості у квадратні дужки `<obj["prop"]>`.

Очевидно, що перший механізм статичний, а другий дозволяє вказувати ім'я властивості через змінну. Базове визначення об'єкта не містить функцій. Операції копіювання об'єкта насправді роблять посилання на вихідний об'єкт. При

видаленні об'єкта здійснюється зменшення лічильника посилань, а при досягненні лічильника посилань нуля об'єкт віддаляється фізично.

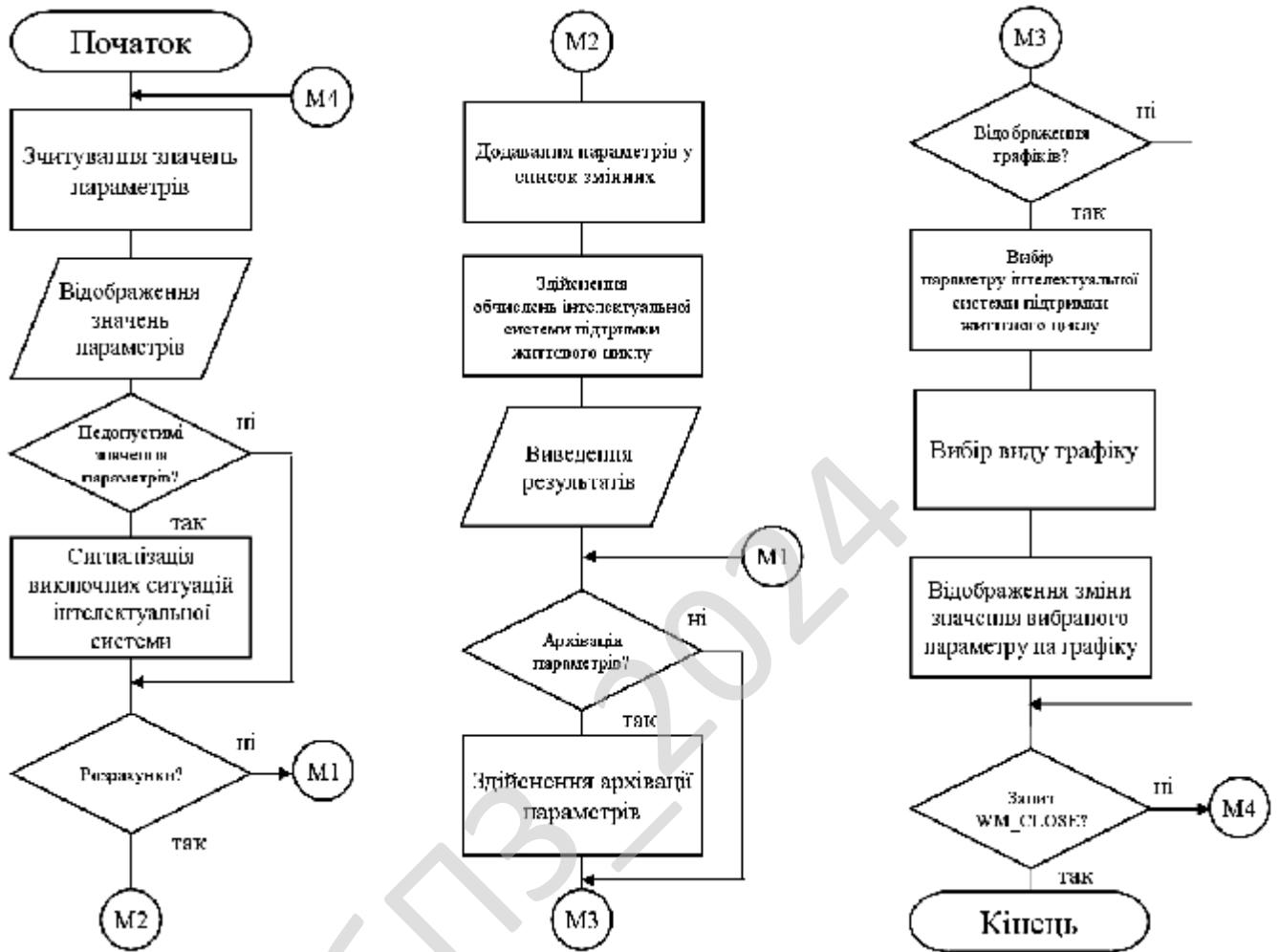


Рисунок 4.2 – Блок-схема роботи підпрограми

Різні компоненти можуть до визначати базовий об'єкт особливими властивостями й функціями. Стандартним розширенням об'єкта є масив "Array".

Об'єкт Array. Особливістю масиву є те, що він працює із властивостями, як з індексами, і повне їхнє іменування безглуздо, а значить доступний механізм обігу тільки висновком індексу у квадратні дужки <arr[1]>. Масив зберігає властивості у власному контейнері одномірного масиву. Цифрові властивості масиву використовуються для доступу безпосередньо до масиву, а символічні працюють як властивості об'єкта.





- string text() – Текст вузла, уміст XML-тегу.
- string attr( string id) – Значення атрибута вузла <id>.
- XMLNodeObj setName( string vl) – Установка ім'я вузла в <vl>. Повертає поточний вузол.
- XMLNodeObj setText( string vl) – Установка тексту вузла в <vl>. Повертає поточний вузол.
- XMLNodeObj setAttr( string id, string vl) – Установка атрибута <id> у значення <vl>. Повертає поточний вузол.
- int childSize() – Кількість вкладених вузлів.
- XMLNodeObj childAdd( EITp no = XMLNodeObj ); XMLNodeObj childAdd(string no) – Додавання об'єкта <no> як вкладеного. <no> може бути як безпосередньо об'єктом-результатом функції SYS.XMLNode(), так і рядком з ім'ям нового тегу. Вертається вкладений вузол.
- XMLNodeObj childIns( int id, EITp no = XMLNodeObj ); XMLNodeObj childIns(int id, string no) – Вставка об'єкта <no> як вкладеного в позицію <id>. <no> може бути як безпосередньо об'єктом-результатом функції SYS.XMLNode(), так і рядком з ім'ям нового тегу. Вертається вкладений вузол.
- XMLNodeObj childDel( int id) – Видалення вкладеного вузла в позиції <id>. Повертає поточний вузол.
- XMLNodeObj childGet( int id) – Одержання вкладеного вузла в позиції <id>.
- XMLNodeObj parent() – Одержати батьківський вузол.
- string load( string str, bool file = false, bool full = false, string cp = " UTF-8") – Завантаження XML з рядка <str> або з файлу зі шляхом в <str> якщо <file> "true", з кодуванням <cp>. <full> – для повного завантаження, із блоками тексту й коментарями в спеціальних вузлах.
- string save( int opt = 0, string path = "", string cp = " UTF-8") – Збереження дерева XML у рядок або у файл <path> з параметрами форматування <opt> і кодуванням <cp>. Повертає текст XML або код помилки.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>48</b>



string cat, string mess ); int messEmerg( string cat, string mess ); – формування системного повідомлення <mess> з категорією <cat> і відповідним рівнем.

– XMLNodeObj XMLNode( string name = "" ); – створення об'єкта вузла XML с ім'ям <name>.

– string cntrReq( XMLNodeObj req, string stat = "" ); – запит інтерфейсу керування до системи за допомогою XML. Звичайний запит записується у вигляді <get path="/OPath/ %2felem"/>. При вказівці станції здійснюється запит до зовнішньої станції.

```
req = SYS.XMLNode("get").setAttr("path", "/%2fgen%2fid");
SYS.cntrReq(req);
idSt = req.text();
```

– string sleep(int tm, int ntm = 0); – приспати потік виконання на <tm> секунд і <ntm> наносекунд. Функція додана тільки для повноти й украй не рекомендована до використання, особливо в процедурах користувальницького інтерфейсу оскільки це приведе до блокування інтерфейсу.

– int time( int usec ); – повертає абсолютний час у секундах від епохи 1.1.1970 і мікросекундах, якщо <usec> зазначений.

– int localtime( int fullsec, int sec, int min, int hour, int mday, int month, int year, int wday, int yday, int isdst); – повертає повну дату в секундах (sec), хвилинах (min), годинниках (hour), днях місяця (mday), місяці (month), році (year), днях тижня (wday), днях у році (yday) і ознака літнього часу (isdst), виходячи з абсолютного часу в секундах <fullsec> від епохи 1.1.1970.

– string strftime( int sec, string form = "% Y-Y-%m-%d %H:%M:%S" ); – Перетворить абсолютний час <sec> у рядок потрібного формату <form>. Запис формату відповідає POSIX-функції strftime.

– int strptime( string str, string form = "% Y-Y-%m-%d %H:%M:%S" ); – Повертає час у секундах від епохи 1.1.1970, виходячи зі строкового запису часу <str>, відповідно до зазначеного шаблону <form>. Наприклад, шаблону "% Y-Y-%m-%d %H:%M:%S" відповідає час " 2006-08-08 11:21:55". Опис формату шаблону можна одержати з документації на POSIX-функцію "strptime".





– string fieldSeek(int row, XMLNodeObj fld); – Запит поля <row> таблиці. Якщо поле отримане то повертається "1" інакше "0". У випадку помилки повертається "0:Error".

– string fieldGet(XMLNodeObj fld); – Запит значень поля. У випадку помилки повертається "0:Error".

```
req = SYS.XMLNode("field");  
req.childAdd("user").setAttr("type","str").setAttr("key","1").setText("root");  
req.childAdd("id").setAttr("type","str").setAttr("key","1").setText("/Lang2CodeBase");  
req.childAdd("val").setAttr("type","str");  
SYS.BD.MySQL.GenDB.SYS.fieldGet(req);  
SYS.messDebug("TEST DB","Value: "+req.childGet(2).text());
```

– string fieldSet(XMLNodeObj fld); – Установка поля. У випадку помилки повертається "0:Error".

– string fieldDel(XMLNodeObj fld); – Видалення поля. У випадку помилки повертається "0:Error".

Підсистема "Збір даних" (SYS.DAQ). Функції об'єкта підсистеми (SYS.DAQ):

– bool funcCall(string progLang, TVarObj args, string prog); – виклик тексту функції <prog> з аргументами в об'єкті <args> для мови програмування <progLang>. Повертає "true" при коректному виклику.

```
var args = new Object(); args.y = 0; args.x = 123;  
SYS.DAQ.funcCall("BuilderCLikeCalc.BuilderCScript",args,"y=2*x");  
SYS.messDebug("TEST Calc","TEST Calc result: "+args.y);
```

Функції об'єкта контролера (SYS.DAQ["Modul"]["Controller"]):

– ElTr cfg(string nm) – одержання значення конфігураційного поля <nm> об'єкта.

– ElTr cfgSet(string nm, ElTr val) – установка конфігураційного поля <nm> об'єкта в значення <val>.

– string name() – ім'я контролера.

– string descr() – опис контролера.

– string status() – статус контролера.

– bool alarmSet(string mess, int lev = -5, string prm = "") – установка/зняття порушення <mess> з рівнем <lev> (негативний для установки інакше зняття), для параметра <prm>.

– bool enable(bool newSt = EVAL) – одержання стану "Включена" або зміна його призначенням атрибута <newSt>.

– bool start(bool newSt = EVAL) – одержання стану "Запущена" або зміна його призначенням атрибута <newSt>.

Функції об'єкта параметра контролера (SYS.DAQ["Modul"]["Controller"] ["Parameter"]):

– ElTp cfg(string nm) – одержання значення конфігураційного поля <nm> об'єкта.

– ElTp cfgSet(string nm, ElTp val) – установка конфігураційного поля <nm> об'єкта в значення <val>.

Функції об'єкта атрибута параметра контролера (SYS.DAQ["Modul"] ["Controller"]["Parameter"]["Attribute"]):

– ElTp get( int tm = 0, int utm = 0, bool sys = false ) – запит значення атрибута на час <tm:utm> і ознакою системного доступу <sys>.

– bool set( ElTp val, int tm = 0, int utm = 0, bool sys = false ) – запис значення <val> в атрибут з міткою часу <tm:utm> і ознакою системного доступу <sys>.

– TCntrNodeObj arch() – одержання об'єкта архіву, пов'язаного із цим атрибутом. У випадку відсутності зв'язаного архіву повертається "false".

– string descr() – опис атрибута.

– int time(int utm) – час останнього значення в секундах і мікросекундах в <utm>.

– int len() – довжина поля.

– int dec() – дозвіл для дійсного числа.

– int flg() – прапори поля.

– string def() – значення за замовчуванням.

– string values() – список припустимих значень або діапазон.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54





– string messIO( string mess, real timeOut = 0 ); – відправлення повідомлення <mess> через транспорт із таймаутом очікування <timeOut> (у секундах). У випадку нульового таймаута цей час береться з налаштувань вихідного транспорту.

```
rez=SYS.Transport.Serial.out_ttyUSB0.messIO(SYS.strFromCharCode(0x4B,0x0 0,
0x37,0x40),0.2);
while(true)
{
trez = SYS.Transport.Serial.out_ttyUSB0.messIO("");
if( !trez.length ) break;
rez+=trez;
}
```

– int messIO( XMLNodeObj req, string prt ); – відправлення запиту <req> до протоколу <prt> для здійснення сеансу зв'язку через транспорт за допомогою протоколу.

```
req = SYS.XMLNode("TCP");
req.setAttr("id","test").setAttr("reqTm",1).setAttr("node",1).setAttr("req
Try",2).setText(SYS.strFromCharCode(0x03,0x0 0,0x0 0,0x0 0,0x05));
SYS.Transport.Sockets.out_testModBus.messIO(req,"ModBus");
test = Special.FLibSYS.strDec4Bin(req.text());
```

Підсистема "Користувальницькі інтерфейси" (SYS.UI). Модуль UI.VCAEngine. Об'єкт "Сеанс" ( this.ownerSess() ).

– string user( ) – поточний користувач сеансу.

– string almSndPlay( ) – шлях віджета для якого на даний момент відтворюється повідомлення про порушення.

– int almQuittance( int quitmpl, string wpath = "" ) – квітирування порушень <wpath> із шаблоном <quit\_tmpl>. Якщо <wpath> порожній рядок то виробляється глобальне квітирування.

Об'єкт "Віджет" (this).

– TCntrNodeObj ownerSess( ) – одержати об'єкт сеансу даного віджета.

– TCntrNodeObj ownerPage( ) – одержати об'єкт батьківської сторінки даного віджета.

– TCntrNodeObj ownerWdg(bool base = false) – одержати батьківський віджет даного віджета. При вказівці <base> буде повернуті й об'єкти сторінок.

– TCntrNodeObj wdgAdd(string wid, string wname, string parent) – додавання віджета <wid> з ім'ям <wname> на основі бібліотечного віджета <parent>.

```
//Додати новий віджет на основі віджета текстового примітива  
nw = this.wdgAdd("nw", "Новий віджет", "/wlb originals/wdg Text");  
nw.attrSet("geom",50).attrSet("geom", 50)7~
```

– bool wdgDel(string wid) – видалення віджета <wid>.

– bool attrPresent(string attr) – перевірка атрибута віджета <attr> на факт присутності.

– ElTp attr(string attr) – одержання значення атрибута віджета <attr>. Для відсутніх атрибутів вертається порожній рядок.

– TCntrNodeObj attrSet(string attr, ElTp vl) – установка атрибута віджета <attr> у значення <vl>. Вертається поточний об'єкт для конкатенації функцій установки.

– string link(string attr, bool prm = false) – одержання посилання для атрибута віджета <attr>. При установці <prm> запитується посилання для групи атрибутів (параметр), представлених зазначеним атрибутом.

– string linkSet(string attr, string vl, bool prm) – установка посилання для атрибута віджета <attr>. При установці <prm> здійснюється установка посилання для групи атрибутів (параметр), представлених зазначеним атрибутом.

```
//Установити посилання восьмого тренда параметром  
this.linkSet("el8.name", "prm:/LogicLev/experiment/Pi", true);
```

Об'єкт "Віджет", примітива "Документ" (this). string getArhDoc(integer nDoc) – одержати текст документа архіву на глибині <nDoc> ( 0-0-{aSize-1}).

Підсистема "Спеціальні" (SYS.Special). Модуль Special.FLibSYS. Об'єкт "Бібліотека функцій" (SYS.Special.FLibSYS)

ElTp {funcID}(ElTp prml, ... ) – виклик функції бібліотеки {funcID}.  
Повертає результат викликуваної функції.

					ВКРБ-123.24.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

## 4.2 Захист розробленого програмного забезпечення

Розроблене програмне забезпечення захистимо за допомогою наступного алгоритму захисту інформації DSA.

Відправник і одержувач електронного документа використовують при обчисленні великі цілі числа:  $G$  і  $P$  – прості числа,  $L$  біт кожне ( $512 < L < 1024$ );  $q$  – просте число довжиною 160 біт (дільник числа  $(P-1)$ ). Числа  $G$ ,  $P$ ,  $q$  є відкритими й можуть бути загальними для всіх користувачів мережі.

Відправник вибирає випадкове ціле число  $X$ ,  $1 < X < q$ . Число  $X$  є секретним ключем відправника для формування електронного цифрового підпису. Потім відправник обчислює значення  $Y = G^X \bmod P$ . Число  $Y$  є відкритим ключем для перевірки підпису відправника. Число  $Y$  передається всім одержувачам документів. Цей алгоритм також передбачає використання однобічної функції гешування  $h(-)$ . У стандарті DSS визначений алгоритм безпечного гешування SHA. Для того щоб підписати документ  $M$ , відправник гешує його в ціле геш-значення  $m$ :  $m = h(M)$ ,  $1 < m < q$ , потім генерує випадкове ціле число  $K$ ,  $1 < K < q$ , і обчислює число  $r$ :  $r = (G^K \bmod P) \bmod q$ . Потім відправник обчислює за допомогою секретного ключа  $X$  ціле число  $s$ :

$$s = \frac{m + r * X}{K} \bmod q .$$

Пара чисел  $r$  і  $s$  утворює цифровий підпис  $S = (r, s)$  під документом  $M$ . Таким чином, підписане повідомлення являє собою трійку чисел  $[M, r, s]$ . Одержувач підписаного повідомлення  $[M, r, s]$  перевіряє виконання умов  $0 < r < q$ ,  $0 < s < q$  і відкидає підпис, якщо хоча б одна із цих умов не виконана. Потім одержувач обчислює значення  $w = 1/s \bmod q$ , геш-значення  $m = h(M)$  і числа  $u_1 = (m * w) \bmod q$ ,  $u_2 = (r * w) \bmod q$ . Далі одержувач за допомогою відкритого ключа  $Y$  обчислює значення  $v = ((G^{u_1} * Y^{u_2}) \bmod P) \bmod q$  і перевіряє виконання умови  $v = r$ . Якщо умова  $v = r$  виконується, тоді підпис  $S = (r, s)$  під документом  $M$  визнається одержувачем справжнім.

					ВКРБ-123.24.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59



Для перегляду короткої довідки про програму слід натиснути на основному вікні кнопку авторського права, після чого на екрані з'явиться вікно показане на рисунку 5.2.

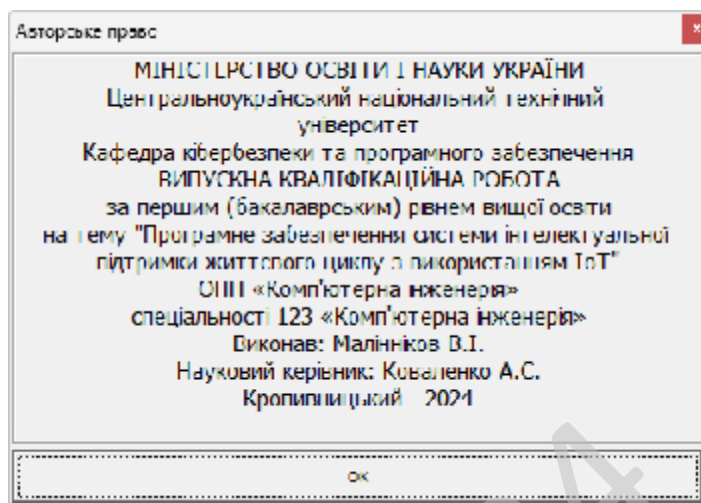


Рисунок 5.2 – Вікно розробника ПЗ

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування форматом чорної скриньки. Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме  $10^{10}$ . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чию поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

- Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).
- Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс;

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

– Сформулювати такі очікувані результати, які з високою імовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

- Некоректних чи відсутніх функцій;
- Помилки інтерфейсу;
- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних;
- Помилки характеристик (необхідна ємність пам'яті і т.д.);
- Помилки ініціалізації та завершення.

Обрано умови розповсюдження – Freeware.

Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

## 6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи інтелектуальної підтримки життєвого циклу з використанням IoT.

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

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем інтелектуальної підтримки життєвого циклу з використанням IoT.

– Досліджена система інтелектуальної підтримки життєвого циклу з використанням IoT.

– На основі отриманих результатів досліджень створена програмна реалізація системи інтелектуальної підтримки життєвого циклу з використанням IoT.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтелектуальної підтримки життєвого циклу з використанням IoT.

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

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

Програма реалізована на мові високого рівня Builder C++. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи інтелектуальної підтримки життєвого циклу з використанням IoT. Це

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм DSA.

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.

КБПЗ-2024

					ВКРБ-123.24.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

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

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

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

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

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

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

7. Donald Norris “Programming with STM32. Getting started with the Nucleo Board and C/C++” 416 p. 2018.

8. Neil Kolban “Kolban’s book on ESP32”. Texas, USA. 951 p.

9. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

10. Kuznetsov, O., Kuznetsova, Y., Smirnov, O., Kostenko, O., Zvieriev, V. «Evaluating Hashing Algorithms in the Age of ASIC Resistance». *CEUR Workshop Proceedings*, 2023, 3628, pp. 93-105.

11. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

12. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

13. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.

14. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

15. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

16. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

17. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable*

*Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

18. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

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

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

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

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

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

24. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of*

*Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

25. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

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

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

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

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

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

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

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

32. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів ІЕС60880 та ІЕС62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

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

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

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

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

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

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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

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

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

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

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

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

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

45. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan

					<b>ВКРБ-123.24.0053.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

46. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

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

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

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

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

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

					ВКРБ-123.24.0053.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

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-123.24.0053.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Малініков В.І.				Програмне забезпечення системи інтелектуальної підтримки життєвого циклу з використанням IoT	Літ.	Аркуш	Аркушів
Перевірів	Коваленко А.С.					Б	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-21-ЗСК			
Затв.	Смірнов О.А.							

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи інтелектуальної підтримки життєвого циклу з використанням IoT.

## 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. № 132-02 від 01.04.2024 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи інтелектуальної підтримки життєвого циклу з використанням IoT.

## 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-123.24.0053.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

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

## 5.2 Показники призначення

Система повинна забезпечувати:

- системи інтелектуальної підтримки життєвого циклу з використанням IoT;
- 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-123.24.0053.00.00.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, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

## 5.8 Вимоги до інформаційної і програмної сумісності

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

### 5.8.1 Обладнання

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

### 5.8.2 Мова програмування

Середовище Builder C++.

					<b>ВКРБ-123.24.0053.00.00.ТЗ</b>	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 – 72 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-123.24.0053.00.00.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.

## 9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2024 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 5.06.2024 р.

					ВКРБ-123.24.0053.00.00.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

\_\_\_\_\_ Коваленко А.С.

*Програмне забезпечення системи інтелектуальної підтримки життєвого  
циклу з використанням IoT*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 68

Літера: РП

Кропивницький – 2024 року

## main.cpp - головна програма

```

//Інтелектуальна система підтримки життєвого циклу з використанням IoT файл
системи: main.cpp

#include <getopt.h>
#include "terror.h"
#include "tmess.h"
#include "tsys.h"

using namespace IoT;

int main(int argc, char *argv[], char *envp[] )
{
    int rez = 0;

    //Перевірка початку роботи режиму основного процесу
    int next_opt;
    optind=opterr=0;
    struct option long_opt[] = { {"Режим основного процесу" ,0,NULL,'d'}, {NULL
,0,NULL,0 } };
    while((next_opt=getopt_long(argc,argv,"",long_opt,NULL)) != -1)
        if( next_opt == 'd' )
        {
            printf("Початок роботи режиму основного процесу!\n");
            int pid = fork();
            if( pid == -1 )
            {
                printf("Помилка: неможливо створити новий процес!\n");
                return -1;
            }
            if( pid != 0 )        return 0;

            //Готується оточення режиму основного процесу
            setsid();
            break;
        }

    try
    {
        SYS = new TSYS(argc,argv,envp);

        SYS->load();
        if( (rez=SYS->stopSignal()) > 0 ) return rez;
        rez = SYS->start();

        delete SYS;
    }catch(TError err) { mess_err(err.cat.c_str(),"%s",err.mess.c_str()); }

    printf("Інтелектуальна система підтримки життєвого циклу з використанням IoT
система коректно працює з даними %d.\n",rez);

    return rez;
}

```

**tsys.cpp - встановлення та запуск системи**

```
//Інтелектуальна система підтримки життєвого циклу з використанням IoT системний
файл: tsys.cpp

#include <features.h>
#include <ieee754.h>
#include <syscall.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/utsname.h>
#include <errno.h>
#include <string.h>
#include <fcntl.h>
#include <unistd.h>
#include <getopt.h>
#include <stdio.h>
#include <signal.h>
#include <stdarg.h>
#include <stdlib.h>
#include <langinfo.h>
#include <zlib.h>

#include "terror.h"
#include "tmess.h"
#include "tsys.h"

using namespace IoT;

//Поточна зміна доступу
TMess *IoT::Mess;
TSYS *IoT::SYS;
bool TSYS::finalKill = false;
pthread_key_t TSYS::sTaskKey;

TSYS::TSYS( int argi, char ** argb, char **env ) : argc(argi), argv((const char
**)argb), envp((const char **)env),
    mUser("root"), mConfFile("/etc/IoT.xml"), mId("EmptySt"), mName(_("Порожня
Станція")), mIcoDir("./icons/"), mModDir("./"),
    mWorkDB("<cfg>"), mSaveAtExit(false), mSavePeriod(0), rootModifCnt(0),
mStopSignal(-1), mMultCPU(false)
{
    finalKill = false;
    SYS = this;          //Ініціалізуємо значення глобальних змінних доступу
    mSubst = grpAdd("sub_",true);
    nodeEn();
    pthread_key_create(&sTaskKey, NULL);

    Mess = new TMess();

    if(getenv("USER")) mUser = getenv("USER");

    //> Ініціалізуємо системний годинник
    clkCalc();

#ifdef __GLIBC_PREREQ(2,4)
    //> Multi CPU дозволяють перевірку
    cpu_set_t cpuset;
    CPU_ZERO(&cpuset);
    CPU_SET(1,&cpuset);
    mMultCPU = !pthread_setaffinity_np(pthread_self(), sizeof(cpu_set_t),
&cpuset);
#endif

    //> Встановлюємо сигнальні програми обробки
    signal(SIGINT,sighandler);
```

```

    signal(SIGTERM, sighandler);
    //signal(SIGCHLD, sighandler);
    signal(SIGALRM, sighandler);
    signal(SIGPIPE, sighandler);
    //signal(SIGFPE, sighandler);
    //signal(SIGSEGV, sighandler);
    signal(SIGABRT, sighandler);
}

TSYS::~~TSYS( )
{
    finalKill = true;

    //Видаляємо всі вузли в команді управління
    del("ModSched");
    del("UI");
    del("Special");
    del("Archive");
    del("DAQ");
    del("Protocol");
    del("Transport");
    del("Security");
    del("BD");

    delete Mess;
    pthread_key_delete(sTaskKey);
}

string TSYS::host( )
{
    utsname ubuf; uname(&ubuf);
    return ubuf.nodename;
}

string TSYS::workDir( )
{
    char buf[STR_BUF_LEN];
    return getcwd(buf, sizeof(buf));
}

void TSYS::setWorkDir( const string &wdir )
{
    if(workDir() == wdir) return;
    if(chdir(wdir.c_str()) != 0)
        mess_warning(nodePath().c_str(), _("Змініть робочу директорію'%s' Помилка:
%s. Можливо поточний каталог вже встановлено правильно'%s'."),
        wdir.c_str(), strerror(errno), workDir().c_str());
    modif( );
}

XMLNode *TSYS::cfgNode( const string &path, bool create )
{
    string s_el, ndNm;

    XMLNode *t_node = &rootN;
    if(t_node->name() != "Інтелектуальна система підтримки життєвого циклу з
використанням IoT")
    {
        if(!create) return NULL;
        t_node->setName("Інтелектуальна система підтримки життєвого циклу з
використанням IoT");
    }

    for(int l_off = 0, nLev = 0; true; nLev++)
    {
        s_el = TSYS::pathLev(path, 0, true, &l_off);
        if(s_el.empty()) return t_node;
        bool ok = false;
        for(unsigned i_f = 0; !ok && i_f < t_node->childSize(); i_f++)

```

```

        if(t_node->childGet(i_f)->attr("id") == s_el)
        {
            t_node = t_node->childGet(i_f);
            ok = true;
        }
    if(!ok)
    {
        if(!create)    return NULL;
        ndNm = "prm";
        switch(nLev)
        {
            case 0: ndNm = "station";    break;
            case 1: if(s_el.compare(0,4,"sub_") == 0) ndNm = "node";    break;
            case 2: if(s_el.compare(0,4,"mod_") == 0) ndNm = "node";    break;
        }
        if(ndNm == "prm") t_node = t_node->childIns(0,ndNm)-
>setAttr("id",s_el);
        else t_node = t_node->childAdd(ndNm)->setAttr("id",s_el);
    }
    }
    return t_node;
}

string TSYS::int2str( int val, TSYS::IntView view )
{
    char buf[STR_BUF_LEN];
    if(view == TSYS::Dec)    snprintf(buf,sizeof(buf),"%d",val);
    else if(view == TSYS::Oct)    snprintf(buf,sizeof(buf),"%o",val);
    else if(view == TSYS::Hex)    snprintf(buf,sizeof(buf),"%x",val);

    return buf;
}

string TSYS::uint2str( unsigned val, IntView view )
{
    char buf[STR_BUF_LEN];
    if(view == TSYS::Dec)    snprintf(buf,sizeof(buf),"%u",val);
    else if(view == TSYS::Oct)    snprintf(buf,sizeof(buf),"%o",val);
    else if(view == TSYS::Hex)    snprintf(buf,sizeof(buf),"%x",val);

    return buf;
}

string TSYS::ll2str( int64_t val, IntView view )
{
    char buf[STR_BUF_LEN];
    if(view == TSYS::Dec)    snprintf(buf,sizeof(buf),"%lld",val);
    else if(view == TSYS::Oct)    snprintf(buf,sizeof(buf),"%llo",val);
    else if(view == TSYS::Hex)    snprintf(buf,sizeof(buf),"%llx",val);

    return buf;
}

string TSYS::real2str( double val, int prec, char tp )
{
    char buf[STR_BUF_LEN];
    if(tp == 'g')    snprintf(buf,sizeof(buf),"%.*g",prec,val);
    else if(tp == 'e')    snprintf(buf,sizeof(buf),"%.*e",prec,val);
    else    snprintf(buf,sizeof(buf),"%.*f",prec,val);

    return buf;
}

string TSYS::time2str( time_t itm, const string &format )
{
    struct tm tm_tm;
    localtime_r(&itm,&tm_tm);
    char buf[100];

```

```

    int ret = strftime(buf, sizeof(buf), format.empty()?"%d-%m-%Y
%H:%M:%S":format.c_str(), &tm_tm);
    return (ret > 0) ? string(buf,ret) : string("");
}

string TSYS::time2str( double utm )
{
    if(utm < 1e-6) return "0";
    int lev = 0;
    int days = (int)floor(utm/(24*60*60*1e6));
    int часы = (int)floor(utm/(60*60*1e6))%24;
    int mins = (int)floor(utm/(60*1e6))%60;
    double usec = utm - 1e6*(days*24*60*60 + часы*60*60 + mins*60);

    string rez;
    if(days)          { rez += TSYS::int2str(days)+"_("day"); lev =
vmax(lev,6); }
    if(часы)          { rez += (rez.size()?"
":"")+TSYS::int2str(часы)+"_("годин"); lev = vmax(lev,5); }
    if(mins && lev < 6) { rez += (rez.size()?"
":"")+TSYS::int2str(mins)+"_("хвилини"); lev = vmax(lev,4); }
    if((1e-6*usec) > 0.5 && lev < 5)      { rez += (rez.size()?"
":"")+TSYS::real2str(1e-6*usec,3)+"_("секунд"); lev = vmax(lev,3); }
    else if((1e-3*usec) > 0.5 && !lev)    { rez += (rez.size()?"
":"")+TSYS::real2str(1e-3*usec,4)+"_("мікросекунд"); lev = vmax(lev,2); }
    else if(usec > 0.5 && !lev)          { rez += (rez.size()?"
":"")+TSYS::real2str(usec,4)+"_("us"); lev = vmax(lev,1); }
    else if(!lev) rez += (rez.size()?" ":"")+TSYS::real2str(1e3*usec,4)+"_("ns");
    return rez;
}

string TSYS::cpct2str( double cnt )
{
    if(cnt > 0.2*pow(2,80)) return
TSYS::real2str(cnt/pow(2,80),3,'g')+"_("YiB");
    if(cnt > 0.2*pow(2,70)) return
TSYS::real2str(cnt/pow(2,70),3,'g')+"_("ZiB");
    if(cnt > 0.2*pow(2,60)) return
TSYS::real2str(cnt/pow(2,60),3,'g')+"_("EiB");
    if(cnt > 0.2*pow(2,50)) return
TSYS::real2str(cnt/pow(2,50),3,'g')+"_("PiB");
    if(cnt > 0.2*pow(2,40)) return
TSYS::real2str(cnt/pow(2,40),3,'g')+"_("TiB");
    if(cnt > 0.2*pow(2,30)) return
TSYS::real2str(cnt/pow(2,30),3,'g')+"_("GiB");
    if(cnt > 0.2*pow(2,20)) return
TSYS::real2str(cnt/pow(2,20),3,'g')+"_("MiB");
    if(cnt > 0.2*pow(2,10)) return
TSYS::real2str(cnt/pow(2,10),3,'g')+"_("KiB");
    return TSYS::real2str(cnt,3,'g')+"_("B");
}

string TSYS::addr2str( void *addr )
{
    char buf[sizeof(void*)*2+3];
    snprintf(buf,sizeof(buf),"%p",addr);

    return buf;
}

void *TSYS::str2addr( const string &str )
{
    return (void *)strtoul(str.c_str(),NULL,16);
}

string TSYS::strNoSpace( const string &val )
{
    int beg = -1, end = -1;

```



```

bool TSYS::cfgFileLoad( )
{
    bool cmd_help = false;

    //===== Редагування параметрів=====
    int next_opt;
    const char *short_opt="h";
    struct option long_opt[] =
    {
        {"help"      ,0,NULL,'h'},
        {"Config"   ,1,NULL,'f'},
        {"Station"  ,1,NULL,'s'},
        {NULL       ,0,NULL,0 }
    };

    optind=opterr=0;
    do
    {
        next_opt=getopt_long(argc, (char * const *)argv, short_opt, long_opt, NULL);
        switch(next_opt)
        {
            case 'h':
                fprintf(stdout, "%s", optDescr().c_str());
                Mess->setMessLevel(7);
                cmd_help = true;
                break;
            case 'f': mConfFile = optarg; break;
            case 's': mId = optarg; break;
            case -1 : break;
        }
    } while(next_opt != -1);

    //Завантажуємо конфігураційний файл
    int hd = open(mConfFile.c_str(), O_RDONLY);
    if(hd < 0) mess_err(nodePath().c_str(),_("Конфігураційний файл '%s' Помилка:
%s"),mConfFile.c_str(),strerror(errno));
    else
    {
        string s_buf;
        int cf_sz = lseek(hd,0,SEEK_END);
        if(cf_sz > 0)
        {
            lseek(hd,0,SEEK_SET);
            char *buf = (char *)malloc(cf_sz+1);
            read(hd,buf,cf_sz);
            buf[cf_sz] = 0;
            s_buf = buf;
            free(buf);
        }
        close(hd);

        try
        {
            ResAlloc res(nodeRes(),true);
            rootN.load(s_buf,true);
            if(rootN.name() == "Інтелектуальна система підтримки життєвого циклу з
використанням IoT")
            {
                XMLNode *stat_n = NULL;
                for(int i_st = rootN.childSize()-1; i_st >= 0; i_st--)
                    if(rootN.childGet(i_st)->name() == "station")
                    {
                        stat_n = rootN.childGet(i_st);
                        if(stat_n->attr("id") == mId) break;
                    }
                if(stat_n && stat_n->attr("id") != mId)
                {

```

```

        mess_warning(nodePath().c_str(),_("Робоча станція '%s' не
представлена у конфігураційному файлі Використайте '%s' конфігурацію робочої
станції!"),
        mId.c_str(), stat_n->attr("id").c_str());
        mId      = stat_n->attr("id");
    }
    if(!stat_n) rootN.clear();
} else rootN.clear();
if(!rootN.childSize()) mess_err(nodePath().c_str(),_("Помилка
конфігурації '%s'!"),mConfFile.c_str());
rootModifCnt = 0;
}
catch(TError err) { mess_err(nodePath().c_str(),_("Завантажуємо
конфігураційний файл Помилка: %s"),err.mess.c_str() ); }
}

return cmd_help;
}

void TSYS::cfgFileSave( )
{
    ResAlloc res(nodeRes(),true);
    if(!rootModifCnt) return;
    int hd = open(mConfFile.c_str(), O_CREAT|O_TRUNC|O_WRONLY, 0664);
    if(hd < 0) mess_err(nodePath().c_str(),_("Конфігураційний файл '%s' Помилка:
%s"),mConfFile.c_str(),strerror(errno));

    string rezFile = rootN.save(XMLNode::XMLHeader);
    int rez = write(hd, rezFile.data(), rezFile.size());
    if(rez != (int)rezFile.size()) mess_err(nodePath().c_str(),_("Помилка запису
конфігурації. %s"),mConfFile.c_str(),((rez<0)?strerror(errno):""));
    rootModifCnt = 0;
    rootFlTm = time(NULL);
}

void TSYS::cfgPrmLoad( )
{
    //Системні параметри
    mName =
TBDS::genDBGet (nodePath()+"StName",name(),"root",TBDS::UseTranslate);
mWorkDB = TBDS::genDBGet (nodePath()+"WorkDB",workDB(),"root",TBDS::OnlyCfg);
setWorkDir (TBDS::genDBGet (nodePath()+"Workdir").c_str());
setIcoDir (TBDS::genDBGet (nodePath()+"IcoDir",icoDir()));
setModDir (TBDS::genDBGet (nodePath()+"ModDir",modDir()));
setSaveAtExit (atoi (TBDS::genDBGet (nodePath()+"SaveAtExit","0").c_str()));
setSavePeriod (atoi (TBDS::genDBGet (nodePath()+"SavePeriod","0").c_str()));
}

void TSYS::load_()
{
    static bool first_load = true;

    bool cmd_help = cfgFileLoad();
    mess_info (nodePath().c_str(),_("Load!"));
    cfgPrmLoad();
    Mess->load(); //Завантажуємо повідомлення

    if( first_load )
    {
        //> Створюємо підсистему
        add( new TBDS() );
        add( new TSecurity() );
        add( new TTransportS() );
        add( new TProtocolS() );
        add( new TDAQS() );
        add( new TArchiveS() );
        add( new TSpecialS() );
        add( new TUIS() );
        add( new TModSchedul() );
    }
}

```

```

//> Завантажуємо модулі
modSchedul().at().load();
if( !modSchedul().at().loadLibS() )
{
    mess_err(nodePath().c_str(),_("Жоден модуль не завантажений. Ваша
конфігурація перервана!"));
    stop();
}

//> Завантажуємо базу даних першої підсистеми
db().at().load();
if( !cmd_help ) modSchedul().at().modifG(); // Для перевантаження
спроби від бази даних

//> Друге завантаження для завантаження від родової БД
Mess->load();
cfgPrmLoad();
}

//> Пряме завантаження підсистем та модулів
vector<string> lst;
list(lst);
for( unsigned i_a=0; i_a < lst.size(); i_a++ )
    try { at(lst[i_a]).at().load(); }
    catch(TError err)
    {
        mess_err(err.cat.c_str(),"%s",err.mess.c_str());
        mess_err(nodePath().c_str(),_("Помилка завантаження підсистеми
'%s'."),lst[i_a].c_str());
    }

if( cmd_help ) stop();
first_load = false;
}

void TSYS::save_ ( )
{
    char buf[STR_BUF_LEN];

    mess_info(nodePath().c_str(),_("Save!"));

    //> Системні параметри
    getcwd(buf, sizeof(buf));
    TBDS::genDBSet (nodePath()+"StName", mName, "root", TBDS::UseTranslate);
    TBDS::genDBSet (nodePath()+"Workdir", buf);
    TBDS::genDBSet (nodePath()+"IcoDir", icoDir());
    TBDS::genDBSet (nodePath()+"ModDir", modDir());
    TBDS::genDBSet (nodePath()+"SaveAtExit", TSYS::int2str(saveAtExit()));
    TBDS::genDBSet (nodePath()+"SavePeriod", TSYS::int2str(savePeriod()));

    Mess->save(); //Завантажуємо повідомлення
}

int TSYS::start ( )
{
    vector<string> lst;
    list(lst);

    mess_info(nodePath().c_str(),_("Start!"));
    for(unsigned i_a=0; i_a < lst.size(); i_a++)
        try { at(lst[i_a]).at().subStart(); }
        catch(TError err)
        {
            mess_err(err.cat.c_str(),"%s",err.mess.c_str());
            mess_err(nodePath().c_str(),_("Помилка запуску
підсистеми'%s'."),lst[i_a].c_str());
        }
}

```

```

cfgFileScan( true );

mess_info(nodePath().c_str(),_("Завершення запуску!"));

unsigned int i_cnt = 1;
mStopSignal = 0;
while(!mStopSignal)
{
    //> CPU підрахунок частоти
    if(!(i_cnt%(10*1000/STD_WAIT_DELAY))    clkCalc( );

    //> Кофігураційний файл змін періодичних перевірок
    if(!(i_cnt%(10*1000/STD_WAIT_DELAY))    cfgFileScan( );

    //> Періодична загальна перевірка бібліотек
    if(modSchedul( ).at().chkPer( ) && !(i_cnt%(modSchedul(
).at().chkPer( )*1000/STD_WAIT_DELAY))
        modSchedul( ).at().libLoad(modDir(),true);

    //> Періодичний запис змін до БД
    if(savePeriod( ) && !(i_cnt%(savePeriod( )*1000/STD_WAIT_DELAY)) save();

    //> Кофігураційний файл зберігає необхідні зміни
    if(!(i_cnt%(10*1000/STD_WAIT_DELAY))    cfgFileSave();

    //> Викликаємо підсистему кожні 10 сек.
    if(!(i_cnt%(10*1000/STD_WAIT_DELAY))
        for(unsigned i_a=0; i_a < lst.size(); i_a++)
            try { at(lst[i_a]).at().perSYSCall(i_cnt/(1000/STD_WAIT_DELAY)); }
            catch(TError err) { mess_err(err.cat.c_str(),"%s",err.mess.c_str()); }
}

    usleep(STD_WAIT_DELAY*1000);
    i_cnt++;
}

mess_info(nodePath().c_str(),_("Stop!"));
if(saveAtExit( ) || savePeriod( )    save();
cfgFileSave();
for(int i_a=lst.size()-1; i_a >= 0; i_a--)
    try { at(lst[i_a]).at().subStop(); }
    catch(TError err)
    {
        mess_err(err.cat.c_str(),"%s",err.mess.c_str());
        mess_err(nodePath().c_str(),_("Помилка остановки
підсистеми'%s'."),lst[i_a].c_str());
    }

    return mStopSignal;
}

void TSYS::stop( )
{
    mStopSignal = SIGUSR1;
}

bool TSYS::chkSelDB( const string& wDB, bool isStrong )
{
    if(selDB().empty( ) && !isStrong) return true;
    if(SYS->selDB( ) == TBDS::realDBName(wDB)) return true;
    return false;
}

void TSYS::sighandler( int signal )
{
    switch(signal)
    {
        case SIGINT:
            SYS->mStopSignal=signal;
    }
}

```

```

        break;
    case SIGTERM:
        mess_warning(SYS->nodePath().c_str(),_("Отриманий сигнал переривання
роботи. Сервер зупиняється!"));
        SYS->mStopSignal=signal;
        break;
    case SIGFPE:
        mess_warning(SYS->nodePath().c_str(),_("Виключення плаваючої крапки
спіймане!"));
        exit(1);
        break;
    case SIGCHLD:
    {
        int status;
        pid_t pid = wait(&status);
        if(!WIFEXITED(status) && pid > 0)
            mess_info(SYS->nodePath().c_str(),_("Вивільнено процес-
потомок%d!"),pid);
        break;
    }
    case SIGPIPE:
        //mess_warning(SYS->nodePath().c_str(),_("Сигнал переривання
PIPE!"));
        break;
    case SIGSEGV:
        mess_emerg(SYS->nodePath().c_str(),_("Сигнал помилки від сегменту!"));
        break;
    case SIGABRT:
        mess_emerg(SYS->nodePath().c_str(),_("Інтелектуальна система підтримки
життєвого циклу з використанням IoT перервала роботу!"));
        break;
    case SIGALRM:      break;
    default:
        mess_warning(SYS->nodePath().c_str(),_("Невизначений
сигнал%d!"),signal);
    }
}

void TSYS::cfgFileScan( bool first )
{
    struct stat f_stat;

    if(stat(cfgFile().c_str(),&f_stat) != 0) return;
    bool up = false;
    if(rootCfgFl != cfgFile() || rootFlTm != f_stat.st_mtime) up = true;
    rootCfgFl = cfgFile();
    rootFlTm = f_stat.st_mtime;

    if(up && !first)
    {
        modifG();
        setSelDB("<cfg>");
        load();
        setSelDB("");
    }
}

int64_t TSYS::curTime( )
{
    timeval cur_tm;
    gettimeofday(&cur_tm,NULL);
    return (int64_t)cur_tm.tv_sec*1000000 + cur_tm.tv_usec;
}

bool TSYS::eventWait( bool &m_mess_r_stat, bool exempl, const string &loc,
time_t tm )
{
    time_t t_tm, s_tm;

```

```

t_tm = s_tm = time(NULL);
while( m_mess_r_stat != exempl )
{
    time_t c_tm = time(NULL);
    //Контролюємо перерву
    if( tm && ( c_tm > s_tm+tm ) )
    {
        mess_crit(loc.c_str(),_("Timeouted !!!"));
        return true;
    }
    //Створюємо повідомлення
    if( c_tm > t_tm+1 ) //1sec
    {
        t_tm = c_tm;
        mess_info(loc.c_str(),_("Чекаємо подію..."));
    }
    usleep(STD_WAIT_DELAY*1000);
}
return false;
}

string TSYS::strSepParse( const string &path, int level, char sep, int *off )
{
    int an_dir = off ? *off : 0;
    int t_lev = 0;
    size_t t_dir;

    if(an_dir >= (int)path.size()) return "";
    while(true)
    {
        t_dir = path.find(sep,an_dir);
        if( t_dir == string::npos )
        {
            if( off ) *off = path.size();
            return (t_lev == level) ? path.substr(an_dir) : "";
        }
        else if( t_lev == level )
        {
            if( off ) *off = t_dir+1;
            return path.substr(an_dir,t_dir-an_dir);
        }
        an_dir = t_dir+1;
        t_lev++;
    }
    return "";
}

string TSYS::strParse( const string &path, int level, const string &sep, int
*off, bool mergeSepSymb )
{
    int an_dir = off ? *off : 0;
    int t_lev = 0;
    size_t t_dir;

    if(an_dir >= (int)path.size() || sep.empty()) return "";
    while(true)
    {
        t_dir = path.find(sep,an_dir);
        if( t_dir == string::npos )
        {
            if( off ) *off = path.size();
            return (t_lev == level) ? path.substr(an_dir) : "";
        }
        else if( t_lev == level )
        {
            if( off ) *off = t_dir+sep.size();
            return path.substr(an_dir,t_dir-an_dir);
        }
        if( mergeSepSymb && sep.size() == 1 )

```

```

        for(an_dir = t_dir; an_dir < (int)path.size() && path[an_dir] ==
sep[0]; ) an_dir++;
        else an_dir = t_dir+sep.size();
        t_lev++;
    }
    return "";
}

string TSYS::strLine( const string &str, int level, int *off )
{
    int an_dir = off ? *off : 0;
    int t_lev = 0, edLnSmbSz = 1;
    size_t t_dir;

    if(an_dir >= (int)str.size()) return "";
    while(true)
    {
        for(t_dir = an_dir; t_dir < str.size(); t_dir++)
            if(str[t_dir] == '\x0D' || str[t_dir] == '\x0A')
                { edLnSmbSz = (str[t_dir] == '\x0D' && ((t_dir+1) < str.size()) &&
str[t_dir+1] == '\x0A') ? 2 : 1; break; }
        if(t_dir >= str.size())
        {
            if(off) *off = str.size();
            return (t_lev==level) ? str.substr(an_dir) : "";
        }
        else if(t_lev == level)
        {
            if(off) *off = t_dir+edLnSmbSz;
            return str.substr(an_dir,t_dir-an_dir);
        }
        an_dir = t_dir+edLnSmbSz;
        t_lev++;
    }
    return "";
}

string TSYS::pathLev( const string &path, int level, bool encode, int *off )
{
    int an_dir = off ? *off : 0;
    int t_lev = 0;
    size_t t_dir;

    //> Перший роздільний прохід
    while(an_dir < (int)path.size() && path[an_dir]=='/') an_dir++;
    if(an_dir >= (int)path.size()) return "";
    //> Шлях рівня процесу
    while(true)
    {
        t_dir = path.find("/",an_dir);
        if( t_dir == string::npos )
        {
            if( off ) *off = path.size();
            return (t_lev == level) ? ( encode ?
TSYS::strDecode(path.substr(an_dir),TSYS::PathEl) : path.substr(an_dir) ) : "";
        }
        else if( t_lev == level )
        {
            if( off ) *off = t_dir;
            return encode ? TSYS::strDecode(path.substr(an_dir,t_dir-
an_dir),TSYS::PathEl) : path.substr(an_dir,t_dir-an_dir);
        }
        an_dir = t_dir;
        t_lev++;
        while(an_dir < (int)path.size() && path[an_dir]=='/') an_dir++;
    }
}

string TSYS::path2sepstr( const string &path, char sep )

```

```

{
    string rez, curv;
    int off = 0;
    while( !(curv=TSYS::pathLev(path,0,false,&off)).empty() )
        rez+=curv+sep;
    if(!rez.empty())    rez.resize(rez.size()-1);

    return rez;
}

string TSYS::sepstr2path( const string &str, char sep )
{
    string rez, curv;
    int off = 0;
    while( !(curv=TSYS::strSepParse(str,0,sep,&off)).empty() )
        rez+="/" + curv;

    return rez;
}

string TSYS::strEncode( const string &in, TSYS::Code tp, const string & symb )
{
    int i_sz;
    string sout;

    switch(tp)
    {
    case TSYS::PathEl:
        sout = in;
        for( i_sz = 0; i_sz < (int)sout.size(); i_sz++ )
            switch( sout[i_sz] )
            {
                case '/': sout.replace(i_sz,1,"%2f"); i_sz+=2; break;
                case '%': sout.replace(i_sz,1,"%25"); i_sz+=2; break;
            }
        break;
    case TSYS::HttpURL:
        sout = in;
        for( i_sz = 0; i_sz < (int)sout.size(); i_sz++ )
            switch( sout[i_sz] )
            {
                case '%': sout.replace(i_sz,1,"%25"); i_sz+=2; break;
                case '!': sout.replace(i_sz,1,"%20"); i_sz+=2; break;
                case '\t': sout.replace(i_sz,1,"%09"); i_sz+=2; break;
                default:
                    if( sout[i_sz]&0x80 )
                    {
                        char buf[4];
                        snprintf(buf,sizeof(buf),"%%02X", (unsigned
char) sout[i_sz]);
                        sout.replace(i_sz,1,buf);
                        i_sz+=2;
                        break;
                    }
            }
        break;
    case TSYS::Html:
        sout.reserve(in.size()+10);
        for( i_sz = 0; i_sz < (int)in.size(); i_sz++ )
            switch( in[i_sz] )
            {
                case '>':    sout+="&gt;";    break;
                case '<':    sout+="&lt;";    break;
                case '"':    sout+="&quot;";  break;
                case '&':    sout+="&amp;";  break;
                case '\':    sout+="&apos;";  break;
                default:     sout+=in[i_sz];
            }
        break;
    }
}

```

```

case TSYS::JavaSc:
    sout.reserve(in.size()+10);
    for( i_sz = 0; i_sz < (int)in.size(); i_sz++ )
        switch( in[i_sz] )
        {
            case '\n':    sout+="\n";        break;
            default:      sout+=in[i_sz];
        }
    break;
case TSYS::SQL:
    sout.reserve(in.size()+10);
    for( i_sz = 0; i_sz < (int)in.size(); i_sz++ )
        switch( in[i_sz] )
        {
            case '\':    sout+="\\";        break;
            case '\"':   sout+="\\";        break;
            case '`':    sout+="\`";        break;
            case '\\':   sout+="\\";        break;
            default:     sout+=in[i_sz];
        }
    break;
case TSYS::Custom:
    sout.reserve(in.size()+10);
    for( i_sz = 0; i_sz < (int)in.size(); i_sz++ )
    {
        unsigned i_smb;
        for(i_smb = 0; i_smb < symb.size(); i_smb++)
            if(in[i_sz] == symb[i_smb])
            {
                char buf[4];
                sprintf(buf, "%02X", (unsigned char)in[i_sz]);
                sout += buf;
                break;
            }
        if(i_smb >= symb.size()) sout += in[i_sz];
    }
    break;
case TSYS::base64:
    {
        sout.reserve(in.size()+in.size()/4+in.size()/57+10);
        const char *base64alph =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/";
        for( i_sz = 0; i_sz < (int)in.size(); i_sz+=3 )
        {
            if(i_sz && !(i_sz%57)) sout.push_back('\n');
            sout.push_back(base64alph[(unsigned char)in[i_sz]>>2]);
            if((i_sz+1) >= (int)in.size())
            {
                sout.push_back(base64alph[((unsigned char)in[i_sz]&0x03)<<4]);
                sout += "==";
            }
            else
            {
                sout.push_back(base64alph[(((unsigned
char)in[i_sz]&0x03)<<4|((unsigned char)in[i_sz+1]>>4))]);
                if((i_sz+2) >= (int)in.size())
                {
                    sout.push_back(base64alph[(((unsigned
char)in[i_sz+1]&0x0F)<<2])]);
                    sout.push_back('=');
                }
                else
                {
                    sout.push_back(base64alph[(((unsigned
char)in[i_sz+1]&0x0F)<<2|((unsigned char)in[i_sz+2]>>6))]);
                    sout.push_back(base64alph[(unsigned char)in[i_sz+2]&0x3F]);
                }
            }
        }
    }
}

```

```

        break;
    }
    case TSYS::FormatPrint:
        sout = in;
        for(i_sz = 0; i_sz < (int)sout.size(); i_sz++)
            if(sout[i_sz] == '%') { sout.replace(i_sz,1,"%%"); i_sz++; }
        break;
    case TSYS::oscdID:
        sout.reserve(in.size());
        for(i_sz = 0; i_sz < (int)in.size(); i_sz++)
            switch(in[i_sz])
            {
                case ' ': case '/': case '\\': case '&': case '(':
                case ')': case '[': case ']': case '!': case '~':
                case `': case '@': case '%': case '^': case '-':
                case '+': case '=': case '*': case '{': case '}':
                case ':': case ';': case '"': case '\': case '<':
                case '>': case '?': case '.': case ',':
                    sout+="_"; break;
                default:      sout+=in[i_sz];
            }
        break;
    case TSYS::Bin:
    {
        string svl, evl;
        sout.reserve(in.size());
        for(int off = 0; (svl=TSYS::strSepParse(in,0,'\n',&off)).size(); )
            for(int offE = 0; (evl=TSYS::strSepParse(svl,0,' ',&offE)).size(); )
                sout+=(char)strtol(evl.c_str(),NULL,16);
        break;
    }
    case TSYS::Reverse:
        for(i_sz = in.size()-1; i_sz >= 0; i_sz--) sout += in[i_sz];
        break;
    case TSYS::ShieldSimb:
        sout.reserve(in.size());
        for(i_sz = 0; i_sz < (int)in.size(); i_sz++)
            if(in[i_sz] == '\\') && i_sz < ((int)in.size()-1)
            {
                switch(in[i_sz+1])
                {
                    case 'a':  sout += '\a';    break;
                    case 'b':  sout += '\b';    break;
                    case 'f':  sout += '\f';    break;
                    case 'n':  sout += '\n';    break;
                    case 'r':  sout += '\r';    break;
                    case 't':  sout += '\t';    break;
                    case 'v':  sout += '\v';    break;
                    case 'x': case 'X':
                        if((i_sz+3) < (int)in.size() && isxdigit(in[i_sz+2]) &&
isxdigit(in[i_sz+3]))
                            { sout +=
(char)strtol(in.substr(i_sz+2,2).c_str(),NULL,16); i_sz += 2; }
                        else sout += in[i_sz+1];
                            break;
                    default:
                        if((i_sz+3) < (int)in.size() && in[i_sz+1] >= '0' &&
in[i_sz+1] <= '7' &&
                                                                    in[i_sz+2] >= '0' &&
in[i_sz+2] <= '7' &&
                                                                    in[i_sz+3] >= '0' &&
in[i_sz+3] <= '7')
                            { sout +=
(char)strtol(in.substr(i_sz+1,3).c_str(),NULL,8); i_sz += 2; }
                        else sout += in[i_sz+1];
                            }
                }
                i_sz++;
            }else sout += in[i_sz];
        break;

```

```

    }
    return sout;
}

unsigned char TSYS::getBase64Code(unsigned char asymb)
{
    switch(asymb)
    {
        case 'A' ... 'Z': return asymb-(unsigned char)'A';
        case 'a' ... 'z': return 26+asymb-(unsigned char)'a';
        case '0' ... '9': return 52+asymb-(unsigned char)'0';
        case '+':         return 62;
        case '/':         return 63;
    }
    return 0;
}

string TSYS::strDecode( const string &in, TSYS::Code tp )
{
    unsigned i_sz;
    string sout;

    switch(tp)
    {
        case TSYS::PathEl: case TSYS::HttpURL: case TSYS::Custom:
            sout.reserve(in.size());
            for(i_sz = 0; i_sz < in.size(); i_sz++)
                switch(in[i_sz])
                {
                    case '%':
                        if(i_sz+2 < in.size())
                        {
                            sout += (char)strtol(in.substr(i_sz+1,2).c_str(),NULL,16);
                            i_sz += 2;
                        }else sout += in[i_sz];
                        break;
                    default: sout += in[i_sz];
                }
            break;
        case TSYS::base64:
            sout.reserve(in.size());
            for( i_sz = 0; i_sz < in.size(); )
            {
                if(in[i_sz] == '\n')    i_sz+=sizeof('\n');
                if((i_sz+3) < in.size())
                    if( in[i_sz+1] != '=' )
                    {
                        char w_code1 = TSYS::getBase64Code(in[i_sz+1]);

                        sout.push_back((TSYS::getBase64Code(in[i_sz])<<2) | (w_code1>>4));
                        if( in[i_sz+2] != '=' )
                        {
                            char w_code2 = TSYS::getBase64Code(in[i_sz+2]);
                            sout.push_back((w_code1<<4) | (w_code2>>2));
                            if( in[i_sz+3] != '=' )
                                sout.push_back((w_code2<<6) | TSYS::getBase64Code(in[i_sz+3]));
                        }
                    }
                i_sz+=4;
            }
            break;
        case TSYS::Bin:
            sout.reserve(in.size());
            for( i_sz = 0; i_sz < in.size(); i_sz++ )
                sout += TSYS::strMess(((i_sz+1)%16)?"%0.2x ":"%0.2x\n", (unsigned
char)in[i_sz]);
            break;
        default: sout = in;    break;
    }
}

```

```

    }

    return sout;
}

string TSYS::strCompr( const string &in, int lev )
{
    z_stream strm;

    if( in.empty() )    return "";

    strm.zalloc = Z_NULL;
    strm.zfree  = Z_NULL;
    strm.opaque = Z_NULL;

    if( deflateInit(&strm,lev) != Z_OK ) return "";

    uLongf comprLen = deflateBound(&strm,in.size());
    char out[comprLen];

    strm.next_in = (Bytef*)in.data();
    strm.avail_in = (uInt)in.size();
    strm.next_out = (Bytef*)out;
    strm.avail_out = comprLen;

    if( deflate(&strm, Z_FINISH) != Z_STREAM_END )
    {
        deflateEnd(&strm);
        return "";
    }

    comprLen = strm.total_out;

    deflateEnd(&strm);

    return string(out,comprLen);
}

string TSYS::strUncompr( const string &in )
{
    int ret;
    z_stream strm;
    unsigned char out[STR_BUF_LEN];
    string rez;

    if( in.empty() )    return "";

    strm.zalloc = Z_NULL;
    strm.zfree  = Z_NULL;
    strm.opaque = Z_NULL;

    if( inflateInit(&strm) != Z_OK )    return "";

    strm.avail_in = in.size();
    strm.next_in = (Bytef*)in.data();
    do
    {
        strm.avail_out = sizeof(out);
        strm.next_out = out;
        ret=inflate(&strm,Z_NO_FLUSH);
        if( ret == Z_STREAM_ERROR || ret == Z_NEED_DICT || ret == Z_DATA_ERROR ||
ret == Z_MEM_ERROR )
            break;
        rez.append((char*)out,sizeof(out)-strm.avail_out);
    } while( strm.avail_out == 0 );

    inflateEnd(&strm);

    if( ret != Z_STREAM_END ) return "";
}

```

```

    return rez;
}

float TSYS::floatLE(float in)
{
#ifdef __BYTE_ORDER == __BIG_ENDIAN
    ieee754_double ieee754_be;
    union ieee754_le
    {
        float f;
        struct
        {
            unsigned int mantissa:23;
            unsigned int exponent:8;
            unsigned int negative:1;
        } ieee;
    } ieee754_le;

    ieee754_be.f = in;
    ieee754_le.ieee.mantissa = ieee754_be.ieee.mantissa;
    ieee754_le.ieee.exponent = ieee754_be.ieee.exponent;
    ieee754_le.ieee.negative = ieee754_be.ieee.negative;

    return ieee754_le.f;
#endif

    return in;
}

float TSYS::floatLErev(float in)
{
#ifdef __BYTE_ORDER == __BIG_ENDIAN
    ieee754_double ieee754_be;
    union ieee754_le
    {
        float f;
        struct
        {
            unsigned int mantissa:23;
            unsigned int exponent:8;
            unsigned int negative:1;
        } ieee;
    } ieee754_le;

    ieee754_le.f = in;
    ieee754_be.ieee.mantissa = ieee754_le.ieee.mantissa;
    ieee754_be.ieee.exponent = ieee754_le.ieee.exponent;
    ieee754_be.ieee.negative = ieee754_le.ieee.negative;

    return ieee754_be.f;
#endif

    return in;
}

double TSYS::doubleLE(double in)
{
#ifdef __BYTE_ORDER == __BIG_ENDIAN || __FLOAT_WORD_ORDER == __BIG_ENDIAN
    ieee754_double ieee754_be;
    union ieee754_le
    {
        double d;
        struct
        {
            unsigned int mantissa1:32;
            unsigned int mantissa0:20;
            unsigned int exponent:11;
            unsigned int negative:1;

```

```

    } ieee;
} ieee754_le;

ieee754_be.d = in;
ieee754_le.ieee.mantissa0 = ieee754_be.ieee.mantissa0;
ieee754_le.ieee.mantissa1 = ieee754_be.ieee.mantissa1;
ieee754_le.ieee.exponent = ieee754_be.ieee.exponent;
ieee754_le.ieee.negative = ieee754_be.ieee.negative;

return ieee754_le.d;
#endif

return in;
}

double TSYS::doubleLErev(double in)
{
#ifdef __BYTE_ORDER == __BIG_ENDIAN || __FLOAT_WORD_ORDER == __BIG_ENDIAN
    ieee754_double ieee754_be;
    union ieee754_le
    {
        double d;
        struct
        {
            unsigned int mantissa1:32;
            unsigned int mantissa0:20;
            unsigned int exponent:11;
            unsigned int negative:1;
        } ieee;
    } ieee754_le;

    ieee754_le.d = in;
    ieee754_be.ieee.mantissa0 = ieee754_le.ieee.mantissa0;
    ieee754_be.ieee.mantissa1 = ieee754_le.ieee.mantissa1;
    ieee754_be.ieee.exponent = ieee754_le.ieee.exponent;
    ieee754_be.ieee.negative = ieee754_le.ieee.negative;

    return ieee754_be.d;
#endif

return in;
}

long TSYS::HZ()
{
return sysconf(_SC_CLK_TCK);
}

bool TSYS::cntrEmpty()
{
ResAlloc res( nodeRes(), false );
return mCntrs.empty();
}

double TSYS::cntrGet( const string &id )
{
ResAlloc res( nodeRes(), false );
map<string,double>::iterator icnt = mCntrs.find(id);
if( icnt == mCntrs.end() ) return 0;
return icnt->second;
}

void TSYS::cntrSet( const string &id, double vl )
{
ResAlloc res( nodeRes(), true );
mCntrs[id] = vl;
}

```

```

void TSYS::taskCreate( const string &path, int priority, void
>(*start_routine)(void *), void *arg, int wtm, pthread_attr_t *pAttr, bool
*startSt )
{
    int detachStat = 0;
    pthread_t procPthr;
    pthread_attr_t locPAttr, *pthr_attr;
    map<string,STask>::iterator ti;

    ResAlloc res(taskRes, true);
    for(time_t c_tm = time(NULL); mTasks.find(path) != mTasks.end(); )
    {
        if(time(NULL) >= (c_tm+wtm)) throw TError(nodePath().c_str(),_("Завдання
'%s' вже присутнє!"),path.c_str());
        res.release();
        usleep(10000);
        res.request(true);
    }
    STask &htsk = mTasks[path];
    htsk.path = path;
    htsk.task = start_routine;
    htsk.taskArg = arg;
    htsk.flgs = 0;
    res.release();

    if(pAttr) pthr_attr = pAttr;
    else
    {
        pthr_attr = &locPAttr;
        pthread_attr_init(pthr_attr);
    }
    pthread_attr_setinheritsched(pthr_attr, PTHREAD_EXPLICIT_SCHED);
    struct sched_param prior;
    prior.sched_priority = 0;

    int policy = SCHED_OTHER;
#ifdef __GLIBC_PREREQ(2,4)
    if(priority < 0)    policy = SCHED_BATCH;
#endif
    if(priority > 0 /*&& SYS->user() == "root"*/)    policy = SCHED_RR;
    pthread_attr_setschedpolicy(pthr_attr, policy);
    prior.sched_priority =
vmax(sched_get_priority_min(policy),vmin(sched_get_priority_max(policy),priority
));
    pthread_attr_setschedparam(pthr_attr,&prior);

    try
    {
        pthread_attr_getdetachstate(pthr_attr,&detachStat);
        if(detachStat == PTHREAD_CREATE_DETACHED) htsk.flgs |= STask::Detached;
        int rez = pthread_create(&procPthr, pthr_attr, taskWrap, &htsk);
        if(rez == EPERM)
        {
            mess_warning(nodePath().c_str(),_("No permission for create real-time
policy. Default thread is created!"));
            policy = SCHED_OTHER;
            pthread_attr_setschedpolicy(pthr_attr, policy);
            prior.sched_priority = 0;
            pthread_attr_setschedparam(pthr_attr,&prior);
            rez = pthread_create(&procPthr, pthr_attr, taskWrap, &htsk);
        }
        if(!pAttr) pthread_attr_destroy(pthr_attr);

        if(rez) throw TError(nodePath().c_str(), _("Завдання створило
помилку%d."), rez);

        //> Чекаємо закінчення ініціалізації структури потоку для не відривних
завдань
        while(!(htsk.flgs&STask::Detached) && !htsk.thr) pthread_yield();

```

```

        //> Чекаємо запуск статусу
        for(time_t c_tm = time(NULL); !(htsk.flgs&STask::Detached) && startSt &&
!(*startSt); )
        {
            if(time(NULL) >= (c_tm+wtm)) throw
TError(nodePath().c_str(),_("Завдання '%s' запуск відкладений!"),path.c_str());
            usleep(STD_WAIT_DELAY *1000);
        }
    }
    catch(TError)
    {
        res.request(true);
        mTasks.erase(path);
        res.release();
        throw;
    }
}

void TSYS::taskDestroy( const string &path, bool *endrunCntr, int wtm, bool
noSignal )
{
    ResAlloc res(taskRes, false);
    map<string,STask>::iterator it = mTasks.find(path);
    if(it == mTasks.end()) return;
    pthread_t thr = it->second.thr;
    res.release();

    if(endrunCntr) *endrunCntr = true;
    if(!noSignal) pthread_kill(thr, SIGALRM);

    //> Чекаємо завершення завдання та повторюємо відправлення SIGALRM
    time_t t_tm, s_tm;
    t_tm = s_tm = time(NULL);
    while(!(it->second.flgs&STask::FinishTask))
    {
        if(!noSignal) pthread_kill(thr, SIGALRM);
        time_t c_tm = time(NULL);
        //Контролюємо перерву
        if(wtm && (c_tm > (s_tm+wtm)))
        {
            mess_crit((nodePath()+path+": stop").c_str(),_("Timeouted !!!"));
            throw TError(nodePath().c_str(),_(«завдання '%s' is not
stopped!"),path.c_str());
        }
        //Створюємо повідомлення
        if(c_tm > t_tm+1) //1sec
        {
            t_tm = c_tm;
            mess_info((nodePath()+path+": stop").c_str(),_("Чекаємо подію..."));
        }
        usleep(STD_WAIT_DELAY*1000);
    }

    if(!(it->second.flgs&STask::Detached)) pthread_join(thr, NULL);

    res.request(true);
    mTasks.erase(it);
}

void *TSYS::taskWrap( void *stas )
{
    //> Беремо тимчасову структуру завдання
    STask *tsk = (STask *)stas;
    pthread_setspecific(TSYS::sTaskKey, tsk);

    //> Запам'ятовуємо параметри виклику
    void *(*wTask) (void *) = tsk->task;
    void *wTaskArg = tsk->taskArg;
}

```

```

//> Отримуємо поточні політику і пріоритет
int policy;
struct sched_param param;
pthread_getschedparam(pthread_self(), &policy, &param);
tsk->policy = policy;
tsk->prior = param.sched_priority;

#if __GLIBC_PREREQ(2,4)
//> Отримуємо і завантажуюмо CPU установлення
if(SYS->multCPU() && !(tsk->flgs & STask::Detached))
{
    tsk->cpuSet = TBDS::genDBGet(SYS->nodePath()+"CpuSet:"+tsk->path);
    cpu_set_t cpuset;
    CPU_ZERO(&cpuset);
    string sval;
    bool cpuSetOK = false;
    for(int off = 0; (sval=TSYS::strParse(tsk->cpuSet,0,":",&off)).size();
cpuSetOK = true)
        CPU_SET(atoi(sval.c_str()),&cpuset);
    if(cpuSetOK) pthread_setaffinity_np(pthread_self(), sizeof(cpu_set_t),
&cpuset);
}
else if(SYS->multCPU() && (tsk->flgs & STask::Detached)) tsk->cpuSet = "NA";
#endif

//> Закінчуємо установки та ініціалізуємо індикатор закінчення
tsk->tid = syscall(SYS_gettid);
tsk->thr = pthread_self();

//> Викликаємо робочі завдання
void *rez = NULL;
try { rez = wTask(wTaskArg); }
catch(TError err)
{
    mess_err(err.cat.c_str(),err.mess.c_str());
    mess_err(SYS->nodePath().c_str(),_("Завдання %u несподівано закінчено
виключенням."),tsk->thr);
}

//> Відмічаємо закінчення завдання
tsk->flgs |= STask::FinishTask;

//> Переміщуємо об'єкти завдання окремо
if(tsk->flgs & STask::Detached) SYS->taskDestroy(tsk->path, NULL);

return rez;
}

void TSYS::taskSleep( int64_t per, time_t cron )
{
    struct timespec sp_tm;
    STask *stsk = (STask*)pthread_getspecific(sTaskKey);

    if(!cron)
    {
        if(!per) per = 1000000000;
        clock_gettime(CLOCK_REALTIME,&sp_tm);
        int64_t end_tm = (int64_t)sp_tm.tv_sec*1000000000+sp_tm.tv_nsec;
        int64_t pnt_tm = (end_tm/per + 1)*per;
        do
        {
            sp_tm.tv_sec = pnt_tm/1000000000; sp_tm.tv_nsec = pnt_tm%1000000000;
            if(clock_nanosleep(CLOCK_REALTIME,TIMER_ABSTIME,&sp_tm,NULL))
                return;
            clock_gettime(CLOCK_REALTIME,&sp_tm);
        }while(((int64_t)sp_tm.tv_sec*1000000000+sp_tm.tv_nsec) < pnt_tm);

        if(stsk)

```

```

    {
        stsk->tm_beg = stsk->tm_per;
        stsk->tm_end = end_tm;
        stsk->tm_per = (int64_t)sp_tm.tv_sec*1000000000+sp_tm.tv_nsec;
    }
}
else
{
    time_t end_tm = time(NULL);
    while(time(NULL) < cron && usleep(1000000) == 0) ;
    if(stsk)
    {
        stsk->tm_beg = stsk->tm_per;
        stsk->tm_end = 1000000000ll*end_tm;
        stsk->tm_per = 1000000000ll*time(NULL);
    }
}
}

time_t TSYS::cron( const string &vl, time_t base )
{
    string cronEl, tEl;
    int vbegin, vend, vstep, vm;

    time_t ctm = base?base:time(NULL);
    struct tm ttm;
    localtime_r(&ctm,&ttm);
    ttm.tm_sec = 0;

reload:
    bool isReload = false;

    //> Хвилини check
    cronEl = TSYS::strSepParse(vl,0,' ');
    vm = 200;
    for(int eoff = 0; (tEl=TSYS::strSepParse(cronEl,0,',',&eoff)).size(); )
    {
        vbegin = vend = -1; vstep = 0;
        sscanf(tEl.c_str(),"%d-%d/%d",&vbegin,&vend,&vstep);
        if(vbegin < 0) { sscanf(tEl.c_str(),"*/%d",&vstep); vbegin=0; vend=59; }
        if(vend < 0) vm = vmin(vm,vbegin+((ttm.tm_min>=vbegin)?60:0));
        else if((vbegin=vmax(0,vbegin)) < (vend=vmin(59,vend)))
        {
            if(ttm.tm_min < vbegin) vm = vmin(vm,vbegin);
            else if((vstep>1 && ttm.tm_min >= (vbegin+((vend-vbegin)/vstep)*vstep)) ||
(vstep <= 0 && ttm.tm_min >= vend))
                vm = vmin(vm,vbegin+60);
            else if(vstep>1 ) vm = vmin(vm, vbegin + vstep*(((ttm.tm_min+1)-
vbegin)/vstep + (((ttm.tm_min+1)-vbegin)%vstep)?1:0));
            else vm = vmin(vm, ttm.tm_min+1);
        }
        if(vm == ttm.tm_min+1) break;
    }
    ttm.tm_min = vm;
    mktime(&ttm);

    //> Перевіряємо час
    cronEl = TSYS::strSepParse(vl,1,' ');
    vm = 200;
    for(int eoff = 0; (tEl=TSYS::strSepParse(cronEl,0,',',&eoff)).size(); )
    {
        vbegin = vend = -1; vstep = 0;
        sscanf(tEl.c_str(),"%d-%d/%d",&vbegin,&vend,&vstep);
        if(vbegin < 0) { sscanf(tEl.c_str(),"*/%d",&vstep); vbegin=0; vend=23; }
        if(vend < 0) vm = vmin(vm,vbegin+((ttm.tm_hour>vbegin)?24:0));
        else if((vbegin=vmax(0,vbegin)) < (vend=vmin(23,vend)))
        {
            if(ttm.tm_hour < vbegin) vm = vmin(vm,vbegin);

```

```

        else if((vstep>1 && ttm.tm_hour > (vbeg+((vend-vbeg)/vstep)*vstep)) ||
(vstep <= 0 && ttm.tm_hour > vend))
            vm = vmin(vm,vbeg+24);
        else if(vstep>1 ) vm = vmin(vm, vbeg + vstep*((ttm.tm_hour-vbeg)/vstep
+ (((ttm.tm_hour-vbeg)%vstep)?1:0)));
        else vm = vmin(vm, ttm.tm_hour);
    }
    if(vm == ttm.tm_hour) break;
}
isReload = (vm != 200 && ttm.tm_hour!=vm);
ttm.tm_hour = vm;
mtime(&ttm);
if(isReload) { ttm.tm_min = -1; goto reload; }

//> Перевіряємо день
cronEl = TSYS::strSepParse(vl,2,' ');
string cronElw = TSYS::strSepParse(vl,4,' ');
vm = 200;
if(cronEl != "")
    for(int eoff = 0; (tEl=TSYS::strSepParse(cronEl,0,',',&eoff)).size(); )
    {
        vbeg = vend = -1; vstep = 0;
        sscanf(tEl.c_str(),"%d-%d/%d",&vbeg,&vend,&vstep);
        if(vbeg < 0) { sscanf(tEl.c_str(),"*/%d",&vstep); vbeg=1; vend=31; }
        if(vend < 0) vm = vmin(vm,vbeg+((ttm.tm_mday>vbeg)?31:0));
        else if((vbeg=vmax(1,vbeg)) < (vend=vmin(31,vend)))
        {
            if(ttm.tm_mday < vbeg) vm = vmin(vm,vbeg);
            else if((vstep>1 && ttm.tm_mday > (vbeg+((vend-vbeg)/vstep)*vstep))
|| (vstep <= 0 && ttm.tm_mday > vend))
                vm = vmin(vm,vbeg+31);
            else if(vstep>1 ) vm = vmin(vm, vbeg + vstep*((ttm.tm_mday-
vbeg)/vstep + (((ttm.tm_mday-vbeg)%vstep)?1:0)));
            else vm = vmin(vm, ttm.tm_mday);
        }
        if(vm == ttm.tm_mday) break;
    }
if(cronEl == "" || (cronElw != "" && !cronElw.empty()))
    for(int eoff = 0; (tEl=TSYS::strSepParse(cronElw,0,',',&eoff)).size(); )
    {
        vbeg = vend = -1; vstep = 0;
        sscanf(tEl.c_str(),"%d-%d/%d",&vbeg,&vend,&vstep);
        if(vbeg < 0) { sscanf(tEl.c_str(),"*/%d",&vstep); vbeg=0; vend=6; }
        if(vend < 0) vm = vmin(vm, ttm.tm_mday - ttm.tm_wday +
vbeg+((ttm.tm_wday>vbeg)?7:0));
        else if((vbeg=vmax(0,vbeg)) < (vend=vmin(6,vend)))
        {
            if(ttm.tm_wday < vbeg) vm = vmin(vm, ttm.tm_mday - ttm.tm_wday +
vbeg);
            else if((vstep>1 && ttm.tm_wday > (vbeg+((vend-vbeg)/vstep)*vstep))
|| (vstep <= 0 && ttm.tm_wday > vend))
                vm = vmin(vm, ttm.tm_mday - ttm.tm_wday + vbeg+7);
            else if(vstep>1) vm = vmin(vm, ttm.tm_mday - ttm.tm_wday + vbeg +
vstep*((ttm.tm_wday-vbeg)/vstep + (((ttm.tm_wday-vbeg)%vstep)?1:0)));
            else vm = vmin(vm, ttm.tm_mday);
        }
        if(vm == ttm.tm_mday) break;
    }
isReload = (vm!=200 && ttm.tm_mday!=vm);
if(vm <= 31) ttm.tm_mday = vm;
else { ttm.tm_mday = vm-31; ttm.tm_mon++; }
mtime(&ttm);
if(isReload) { ttm.tm_min = -1; ttm.tm_hour = 0; goto reload; }

//> Перевіряємо місяць
cronEl = TSYS::strSepParse(vl,3,' ');
vm = 200;
for(int eoff = 0; (tEl=TSYS::strSepParse(cronEl,0,',',&eoff)).size(); )
{

```

```

vbeg = vend = -1; vstep = 0;
sscanf(tEl.c_str(), "%d-%d/%d", &vbeg, &vend, &vstep);
if(vbeg < 0) { sscanf(tEl.c_str(), "*/%d", &vstep); vbeg=1; vend=12; }
if(vend < 0) vm = vmin(vm, vbeg+(((ttm.tm_mon+1)>vbeg)?12:0));
else if((vbeg=vmax(1, vbeg)) < (vend=vmin(12, vend)))
{
    if((ttm.tm_mon+1) < vbeg) vm = vmin(vm, vbeg);
    else if((vstep>1 && (ttm.tm_mon+1) > (vbeg+((vend-vbeg)/vstep)*vstep))
|| (vstep <= 0 && (ttm.tm_mon+1) > vend)
        vm = vmin(vm, vbeg+12);
    else if(vstep>1) vm = vmin( vm, vbeg + vstep*(((ttm.tm_mon+1)-
vbeg)/vstep + (((ttm.tm_mon+1)-vbeg)%vstep)?1:0));
    else vm = vmin(vm, ttm.tm_mon+1);
}
    if(vm == (ttm.tm_mon+1)) break;
}
isReload = (vm!=200 && ttm.tm_mon!=(vm-1));
ttm.tm_mon = vm-1;
mktime(&ttm);
if(isReload) { ttm.tm_min = -1; ttm.tm_hour = 0; ttm.tm_mday = 1; goto
reload; }

return mktime(&ttm);
}

```

```

TVariant TSYS::objFuncCall( const string &iid, vector<TVariant> &prms, const
string &user )
{
    // int message(string cat, int level, string mess) - форматуємо системне
повідомлення <mess> з категорією <cat>, рівнем <level>
    // cat - повідомлення категорії
    // level - повідомлення рівня
    // mess - повідомлення тексту
    if(iid == "message" && prms.size() >= 3) { message(
prms[0].getS().c_str(), (TMess::Type)prms[1].getI(), "%s",
prms[2].getS().c_str() ); return 0; }
    // int messDebug(string cat, string mess) - форматуємо системне повідомлення
<mess> з категорією <cat> і відповідний рівень
    // cat - повідомлення категорії
    // mess - повідомлення тексту
    if(iid == "messDebug" && prms.size() >= 2) { mess_debug(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messInfo" && prms.size() >= 2) { mess_info(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messNote" && prms.size() >= 2) { mess_note(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messWarning" && prms.size() >= 2){ mess_warning(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messErr" && prms.size() >= 2) { mess_err(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messCrit" && prms.size() >= 2) { mess_crit(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messAlert" && prms.size() >= 2) { mess_alert(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    if(iid == "messEmerg" && prms.size() >= 2) { mess_emerg(
prms[0].getS().c_str(), "%s", prms[1].getS().c_str() ); return 0; }
    // string system(string cmd, bool noPipe = false) - викликаємо консольні
команди <cmd> для повернення у ОС результатів з каналів
    // cmd - текст команди
    // noPipe - результат блокують для другорядного виклику
    if(iid == "system" && prms.size() >= 1)
    {
        if(prms.size() >= 2 && prms[1].getB()) return
system(prms[0].getS().c_str());
        FILE *fp = popen(prms[0].getS().c_str(), "r");
        if(!fp) return string("");

        char buf[STR_BUF_LEN];
        string rez;

```

```

for(int r_cnt = 0; (r_cnt=fread(buf,1,sizeof(buf),fp)); )
    rez.append(buf,r_cnt);

pclose(fp);
return rez;
}
// string fileRead( string file ) - Повертаємо <file> контент у рядку.
if(iid == "fileRead" && prms.size() >= 1)
{
    char buf[STR_BUF_LEN];
    string rez;
    int hd = open(prms[0].getS().c_str(),O_RDONLY);
    if(hd != -1)
    {
        for(int len = 0; (len=read(hd,buf,sizeof(buf))) > 0; )
rez.append(buf,len);
        close(hd);
    }
    return rez;
}
// int fileWrite( string file, string str, bool append = false ) - Записуємо
<str> до <file>, перемішуємо представлення, або <append>.
// Return wrote bytes count.
if(iid == "fileWrite" && prms.size() >= 2)
{
    int wcnt = 0, wflags = O_WRONLY|O_CREAT|O_TRUNC;
    string val = prms[1].getS();
    if(prms.size() >= 3 && prms[2].getB()) wflags = O_WRONLY|O_CREAT|O_APPEND;
    int hd = open(prms[0].getS().c_str(), wflags, 0664);
    if(hd != -1)
    {
        wcnt = write(hd,val.data(),val.size());
        close(hd);
    }
    return wcnt;
}
// XMLNodeObj XMLNode(string name = "") - створюємо XML об'єкт вузлів з
іменем <name>
// name - XML ім'я вузлу
if(iid == "XMLNode") return new XMLNodeObj((prms.size())>=1 ? prms[0].getS()
: "");
// string cntrReq(XMLNodeObj req, string stat = "") - запит інтерфейсу
управління до системи через XML
// req - запити XML вузлів
// stat - перемішуємо Інтелектуальна система підтримки життєвого циклу з
використанням IoT-робочу станцію для запиту
if(iid == "cntrReq" && prms.size() >= 1)
{
    XMLNode req;
    if(!dynamic_cast<XMLNodeObj*>(prms[0].getO())) return string(_("1:Запит не
об'єктний!"));
    ((XMLNodeObj*)prms[0].getO())->toXMLNode(req);
    string path = req.attr("path");
    if(prms.size() < 2 || prms[1].getS().empty())
    {
        req.setAttr("user",user);
        cntrCmd(&req);
    }
    else
    {
        req.setAttr("path","/"+prms[1].getS()+path);
        transport().at().cntrIfCmd(req,"cntrReq");
        req.setAttr("path",path);
    }
    ((XMLNodeObj*)prms[0].getO())->fromXMLNode(req);
    return string("0");
}
// string sleep(int tm, int ntm = 0) - викликаємо завдання переходу до
сплячого режиму через <tm> секунд та <ntm> наносекунд.

```

```

// tm - чекаємо цей час у секундах
// ntm - чекаємо цей час у наносекундах
if(iid == "sleep" && prms.size() >= 1)
{
    struct timespec sp_tm;
    sp_tm.tv_sec = prms[0].getI();
    sp_tm.tv_nsec = (prms.size() >= 2) ? prms[1].getI() : 0;
    int rez = clock_nanosleep(CLOCK_REALTIME, 0, &sp_tm, NULL);
    return rez;
}
// int time(int usec) - повертаємо абсолютний час у секундах від 1/1/1970 та
в мікросекундах, якщо <usec> задано
// usec - мікросекунди of time
if(iid == "time")
{
    if(prms.empty()) return (int)time(NULL);
    int64_t tm = curTime();
    prms[0].setI(tm%1000000); prms[0].setModify();
    return (int)(tm/1000000);
}
// int localtime(int fullsec, int sec, int min, int hour, int mday, int
month, int year, int wday, int yday, int isdst)
// - повертаємо повну дату, базуємо на абсолютному часі у секундах
<fullsec> від 1.1.1970
// fullsec - час джерела у секундах від 1.1.1970
// sec - секунди
// min - хвилини
// hour - часи
// mday - дні місяця
// month - місяці
// year - роки
// wday - дні неділі
// yday - дні року
// isdst - відмітка про літній час
if(iid == "localtime" && prms.size() >= 2)
{
    time_t tm_t = prms[0].getI();
    struct tm tm_tm;
    localtime_r(&tm_t, &tm_tm);

    prms[1].setI(tm_tm.tm_sec); prms[1].setModify();
    if(prms.size() >= 3) { prms[2].setI(tm_tm.tm_min); prms[2].setModify();
}
    if(prms.size() >= 4) { prms[3].setI(tm_tm.tm_hour);
prms[3].setModify(); }
    if(prms.size() >= 5) { prms[4].setI(tm_tm.tm_mday);
prms[4].setModify(); }
    if(prms.size() >= 6) { prms[5].setI(tm_tm.tm_mon); prms[5].setModify();
}
    if(prms.size() >= 7) { prms[6].setI(1900+tm_tm.tm_year);
prms[6].setModify(); }
    if(prms.size() >= 8) { prms[7].setI(tm_tm.tm_wday);
prms[7].setModify(); }
    if(prms.size() >= 9) { prms[8].setI(tm_tm.tm_yday);
prms[8].setModify(); }
    if(prms.size() >= 10) { prms[9].setI(tm_tm.tm_isdst);
prms[9].setModify(); }
    return 0;
}
// string strftime(int sec, string form = "%Y-%m-%d %H:%M:%S") - перетворює
абсолютний час <sec> у рядок формату <form>
// sec - час у секундах від 1.1.1970
// form - вихідний форматований рядок
if(iid == "strftime" && !prms.empty())
{
    time_t tm_t = prms[0].getI();
    struct tm tm_tm;
    localtime_r(&tm_t, &tm_tm);
    char buf[1000];

```

```

    int rez = strftime(buf, sizeof(buf), (prms.size()>=2) ?
prms[1].getS().c_str() : "%Y-%m-%d %H:%M:%S", &tm_tm);
    return (rez>0) ? string(buf,rez) : "";
}
// int strtptime(string str, string form = "%Y-%m-%d %H:%M:%S") - повертає
час у секундах від of 1/1/1970,
// базується на запису рядку часу <str>, відповідно до вказаного
шаблону <form>
// str - джерело часу у рядку
// form - рядки часу у форматі POSIX-функцій "strptime"
if(iid == "strptime" && !prms.empty())
{
    struct tm stm;
    stm.tm_isdst = -1;
    strtptime(prms[0].getS().c_str(), (prms.size()>=2) ? prms[1].getS().c_str()
: "%Y-%m-%d %H:%M:%S", &stm);
    return (int)mkttime(&stm);
}
// int cron(string cronreq, int base = 0) - повертає час , планується у
форматі стандартного Cron <cronreq>,
// початок від основного часу<base> або від поточного, якщо основа не
вказана
// cronreq - розповсюджений у стандартному форматі Cron
// base - основний час
if(iid == "cron" && !prms.empty())
    return (int)cron(prms[0].getS(), (prms.size()>=2) ? prms[1].getI() : 0);
// string strFromCharCode(int char1, int char2, int char3, ...) - створює
рядок з кодових символів
// char1, char2. char3 - кодові символи
if(iid == "strFromCharCode")
{
    string rez;
    for(unsigned i_p = 0; i_p < prms.size(); i_p++)
        rez += (unsigned char)prms[i_p].getI();
    return rez;
}
// string strCodeConv( string src, string fromCP, string toCP ) - Текстовий
рядок перекодує з кодової сторінки <fromCP> до кодової сторінки <toCP>.
// src - source text;
// fromCP - з кодової сторінки , порожньої для внутрішньої кодової сторінки
;
// toCP - до кодової сторінки , порожньої для внутрішньої кодової сторінки
.
if(iid == "strCodeConv" && prms.size() >= 3)
    return Mess->codeConv((prms[1].getS().size() ? prms[1].getS() : Mess-
>charset()),
                        (prms[2].getS().size() ? prms[2].getS() : Mess->charset()),
prms[0].getS());

return TCntrNode::objFuncCall(iid,prms,user);
}

void TSYS::cntrCmdProc( XMLNode *opt )
{
    char buf[STR_BUF_LEN];

    //Беремо сторінку інформації
    if(opt->name() == "info")
    {
        TCntrNode::cntrCmdProc(opt);
        snprintf(buf,sizeof(buf),_("%s station:
\"%s\""),PACKAGE_NAME,name().c_str());
        ctrMkNode("IoT_cntr",opt,-1,"/",buf,R_R_R_);
        if(ctrMkNode("branches",opt,-1,"/br","",R_R_R_))
            ctrMkNode("grp",opt,-
1,"/br/sub_",_("Subsystem"),R_R_R_,"root","root",1,"idm","1");
        if(TUIS::icoPresent(id())) ctrMkNode("img",opt,-1,"/ico","",R_R_R_);
        if(ctrMkNode("area",opt,-1,"/gen",_("Station"),R_R_R_))
        {

```

```

        ctrMkNode("fld",opt,-
1, "/gen/id",_("ID"),R_R_R_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-
1, "/gen/stat",_("Station"),RWRWR_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-
1, "/gen/prog",_("Program"),R_R_R_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-
1, "/gen/ver",_("Version"),R_R_R_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-1, "/gen/host",_("Ім'я
хосту"),R_R_R_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-1, "/gen/user",_("Користувач
системи"),R_R_R_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-1, "/gen/sys",_("Операційна
система"),R_R_R_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-1, "/gen/frq",_("Частота
(MHZ)"),R_R_R_,"root","root",1,"tp","real");
        ctrMkNode("fld",opt,-1, "/gen/clk_res",_("Значення годинника реального
часу"),R_R_R_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-1, "/gen/in_charset",_("Внутрішній набір
символів"),R_R_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-1, "/gen/config",_("Конфігураційний
файл"),R_R_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-1, "/gen/workdir",_("Робоча
директорія"),RWRW_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-1, "/gen/icodir",_("Директорія
іконок"),RWRW_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-1, "/gen/moddir",_("Директорія
модулів"),RWRW_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-1, "/gen/wrk_db",_("Робоча база
даних"),RWRWR_,"root","root",4,"tp","str","dest","select","select","/db/list",
        "help",_("Адрес робочої бази даних у форматі [<БД module>.<БД
name>].\n Змінюємо ці поля якщо необхідно зберегти або завантажити усю систему з
іншої БД."));
        ctrMkNode("fld",opt,-1, "/gen/saveExit",_("Збереження змін у системі
перед виходом"),RWRWR_,"root","root",2,"tp","bool",
        "help",_("Обираємо автоматичне збереження системи до БД перед
виходом."));
        ctrMkNode("fld",opt,-1, "/gen/savePeriod",_("Збереження періоду роботи
у системі "),RWRWR_,"root","root",2,"tp","dec",
        "help",_("Використовуємо нульовий період (секунди) для періодичного
збереження змін частин системи у БД."));
        ctrMkNode("fld",opt,-
1, "/gen/lang",_("Language"),RWRWR_,"root","root",1,"tp","str");
        ctrMkNode("fld",opt,-1, "/gen/baseLang",_("Базова мова тексту
змінних"),RWRWR_,"root","root",5,"tp","str","len","2","dest","sel_ed","select","
/gen/baseLangLs",
        "help",_("Мультимовність для змінного тексту підтримки для вибору
базової мови."));
        if(ctrMkNode("area",opt,-1, "/gen/mess",_("Повідомлення"),R_R_R_))
        {
            ctrMkNode("fld",opt,-1, "/gen/mess/lev",_("Найменший
рівень"),RWRWR_,"root","root",3,
            "tp","dec","len","1","help",_("Повідомлення найменшого рівня для
відображення процесів, які відбуваються у системі."));
            ctrMkNode("fld",opt,-1, "/gen/mess/log_sysl",_("To
syslog"),RWRWR_,"root","root",1,"tp","bool");
            ctrMkNode("fld",opt,-1, "/gen/mess/log_stdio",_("To
stdout"),RWRWR_,"root","root",1,"tp","bool");
            ctrMkNode("fld",opt,-1, "/gen/mess/log_stde",_("To
stderr"),RWRWR_,"root","root",1,"tp","bool");
            ctrMkNode("fld",opt,-1, "/gen/mess/log_arch",_("До
архіву"),RWRWR_,"root","root",1,"tp","bool");
        }
    }
    if(ctrMkNode("area",opt,-1, "/subs",_("Підсистема")))
        ctrMkNode("list",opt,-
1, "/subs/br",_("Підсистеми"),R_R_R_,"root","root",3,"idm","1","tp","br","br_pref
","sub_");
        if(ctrMkNode("area",opt,-1, "/tasks",_("Tasks"),R_R_))

```

```

        if(ctrMkNode("table",opt,-
1, "/tasks/tasks",_ ("Завдання"),RWRW_,"root","root",2,"key","path",
        "help",!multCPU()?":_ ("Для CPU встановлюємо рядок номерів
процесорів використання, відокремлений символом:'.\n"
        "CPU починаємо з 0.)))
    {
        ctrMkNode("list",opt,-
1, "/tasks/tasks/path",_ ("Path"),R_R_,"root","root",1,"tp","str");
        ctrMkNode("list",opt,-
1, "/tasks/tasks/thrd",_ ("Поток"),R_R_,"root","root",1,"tp","str");
        ctrMkNode("list",opt,-
1, "/tasks/tasks/tid",_ ("TID"),R_R_,"root","root",1,"tp","dec");
        ctrMkNode("list",opt,-
1, "/tasks/tasks/stat",_ ("Статус"),R_R_,"root","root",1,"tp","str");
        ctrMkNode("list",opt,-
1, "/tasks/tasks/plc",_ ("Політика"),R_R_,"root","root",1,"tp","str");
        ctrMkNode("list",opt,-
1, "/tasks/tasks/prior",_ ("Prior."),R_R_,"root","root",1,"tp","dec");
#ifdef __GLIBC__
        if(multCPU())
            ctrMkNode("list",opt,-1, "/tasks/tasks/cpuSet",_ ("CPU
встановлення"),RWRW_,"root","root",1,"tp","str");
#endif
    }
    if( !cntrEmpty() && ctrMkNode("area",opt,-1, "/cntr",_ ("Країна")) )
        if( ctrMkNode("table",opt,-
1, "/cntr/cntr",_ ("Counters"),R_R_,"root","root") )
        {
            ctrMkNode("list",opt,-
1, "/cntr/cntr/id", "ID",R_R_,"root","root",1,"tp","str");
            ctrMkNode("list",opt,-
1, "/cntr/cntr/vl",_ ("Value"),R_R_,"root","root",1,"tp","real");
        }
        if( ctrMkNode("area",opt,-1, "/hlp",_ ("Help"),R_R_ ) )
            ctrMkNode("fld",opt,-1, "/hlp/g_help",_ ("Опції
допомоги"),R_R_,"root","root",3,"tp","str","cols","90","rows","10");
        return;
    }

//Процес управління на сторінці
string a_path = opt->attr("path");
if(a_path == "/ico" && ctrChkNode(opt))
{
    string itp;
    opt->setText(TSYS::strEncode(TUIS::icoGet(id(),&itp),TSYS::base64));
    opt->setAttr("tp",itp);
}
else if(a_path == "/gen/host" && ctrChkNode(opt)) opt->setText(host());
else if(a_path == "/gen/sys" && ctrChkNode(opt))
{
    utsname ubuf; uname(&ubuf);
    opt->setText(string(ubuf.sysname)+"-"+ubuf.release);
}
else if(a_path == "/gen/user" && ctrChkNode(opt)) opt->setText(mUser);
else if(a_path == "/gen/prog" && ctrChkNode(opt)) opt->setText(PACKAGE_NAME);
else if(a_path == "/gen/ver" && ctrChkNode(opt)) opt->setText(VERSION);
else if(a_path == "/gen/id" && ctrChkNode(opt)) opt->setText(id());
else if(a_path == "/gen/stat")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText(name());
    if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) setName(opt->text());
}
else if(a_path == "/gen/frq" && ctrChkNode(opt)) opt->
setText(TSYS::real2str((float)sysClk()/1000000.,6));
else if(a_path == "/gen/clk_res" && ctrChkNode(opt))
{
    struct timespec tmval;
    clock_getres(CLOCK_REALTIME,&tmval);

```

```

    opt->setText(TSYS::time2str(1e-3*tmval.tv_nsec));//
    TSYS::real2str((float)tmval.tv_nsec/1000000.,4));
}
else if(a_path == "/gen/in_charset" && ctrChkNode(opt)) opt->setText(Mess-
>charset());
else if(a_path == "/gen/config" && ctrChkNode(opt))          opt-
>setText(mConfFile);
else if(a_path == "/gen/wrk_db" )
{
    if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt-
>setText(mWorkDB);
    if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) setWorkDB(opt-
>text());
}
else if(a_path == "/gen/saveExit")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText(
int2str(saveAtExit() ));
    if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) setSaveAtExit(
atoi(opt->text().c_str() ));
}
else if(a_path == "/gen/savePeriod")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText(
int2str(savePeriod() ));
    if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) setSavePeriod(
atoi(opt->text().c_str() ));
}
else if(a_path == "/gen/workdir")
{
    if(ctrChkNode(opt,"get",R_R___,"root","root",SEC_RD)) opt-
>setText(workDir());
    if(ctrChkNode(opt,"set",R_R___,"root","root",SEC_WR)) setWorkDir(opt-
>text().c_str());
}
else if(a_path == "/gen/icodir")
{
    if(ctrChkNode(opt,"get",R_R___,"root","root",SEC_RD)) opt-
>setText(icoDir());
    if(ctrChkNode(opt,"set",R_R___,"root","root",SEC_WR)) setIcoDir(opt-
>text().c_str());
}
else if(a_path == "/gen/moddir")
{
    if(ctrChkNode(opt,"get",R_R___,"root","root",SEC_RD)) opt-
>setText(modDir());
    if(ctrChkNode(opt,"set",R_R___,"root","root",SEC_WR)) setModDir(opt-
>text().c_str());
}
else if(a_path == "/gen/lang")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText(Mess-
>lang());
    if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->setLang(opt-
>text());
}
else if(a_path == "/gen/baseLang")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText(Mess-
>lang2CodeBase());
    if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess-
>setLang2CodeBase(opt->text());
}
else if(a_path == "/gen/baseLangLs" && ctrChkNode(opt))
{
    opt->childAdd("el")->setText(Mess->lang2Code());
    if(!Mess->lang2CodeBase().empty() && Mess->lang2CodeBase() != Mess-
>lang2Code())
        opt->childAdd("el")->setText(Mess->lang2CodeBase());
}

```

```

    opt->childAdd("el")->setText("");
}
else if(a_path == "/gen/mess/lev")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText(TSYS::int2str(Mess->messLevel()));
    if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->setMessLevel(atoi(opt->text().c_str()));
}
else if(a_path == "/gen/mess/log_sysl")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText((Mess->logDirect() & 0x01)?"1":"0");
    if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->setLogDirect(atoi(opt->text().c_str())?Mess->logDirect() | 0x01: Mess->logDirect() & (~0x01) );
}
else if(a_path == "/gen/mess/log_stdio")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText((Mess->logDirect() & 0x02)?"1":"0");
    if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->setLogDirect(atoi(opt->text().c_str())?Mess->logDirect() | 0x02: Mess->logDirect() & (~0x02) );
}
else if(a_path == "/gen/mess/log_stde")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText((Mess->logDirect() & 0x04)?"1":"0");
    if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->setLogDirect(atoi(opt->text().c_str())?Mess->logDirect() | 0x04: Mess->logDirect() & (~0x04) );
}
else if(a_path == "/gen/mess/log_arch")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root","root",SEC_RD)) opt->setText((Mess->logDirect() & 0x08)?"1":"0");
    if(ctrChkNode(opt,"set",RWRWR_,"root","root",SEC_WR)) Mess->setLogDirect(atoi(opt->text().c_str())?Mess->logDirect() | 0x08: Mess->logDirect() & (~0x08) );
}
else if((a_path == "/br/sub_" || a_path == "/subs/br") &&
ctrChkNode(opt,"get",R_R_R_,"root","root",SEC_RD))
{
    vector<string> lst;
    list(lst);
    for(unsigned i_a=0; i_a < lst.size(); i_a++)
        opt->childAdd("el")->setAttr("id",lst[i_a])->setText(at(lst[i_a]).at().subName());
}
else if(a_path == "/tasks/tasks")
{
    if(ctrChkNode(opt,"get",RWRW__, "root","root"))
    {
        XMLNode *n_path = ctrMkNode("list",opt,-1,"/tasks/tasks/path","",R_R__, "root","root");
        XMLNode *n_thr = ctrMkNode("list",opt,-1,"/tasks/tasks/thrd","",R_R__, "root","root");
        XMLNode *n_tid = ctrMkNode("list",opt,-1,"/tasks/tasks/tid","",R_R__, "root","root");
        XMLNode *n_stat = ctrMkNode("list",opt,-1,"/tasks/tasks/stat","",R_R__, "root","root");
        XMLNode *n_plc = ctrMkNode("list",opt,-1,"/tasks/tasks/plc","",R_R__, "root","root");
        XMLNode *n_prior = ctrMkNode("list",opt,-1,"/tasks/tasks/prior","",R_R__, "root","root");
        XMLNode *n_cpuSet = (multCPU() ? ctrMkNode("list",opt,-1,"/tasks/tasks/cpuSet","",RWRW__, "root","root") : NULL);

        ResAlloc res(taskRes,false);
        for(map<string,STask>::iterator it = mTasks.begin(); it != mTasks.end(); it++)
        {

```

```

        if(n_path) n_path->childAdd("el")->setText(it->first);
        if(n_thr) n_thr->childAdd("el")->setText(TSYS::uint2str(it-
>second.thr));
        if(n_tid) n_tid->childAdd("el")->setText(TSYS::int2str(it-
>second.tid));
        if(n_stat)
        {
            int64_t tm_beg = 0, tm_end = 0, tm_per = 0;
            for(int i_tr = 0; tm_beg == tm_per && i_tr < 2; i_tr++)
            { tm_beg = it->second.tm_beg; tm_end = it->second.tm_end; tm_per
= it->second.tm_per; }
            XMLNode *cn = n_stat->childAdd("el");
            if(it->second.flgs&STask::FinishTask) cn->setText(_("Закінчено.
"));
            if(tm_beg && tm_beg < tm_per)
                cn->setText(cn->text()+TSYS::strMess(_("Останій: %s.
Завантажено: %3.1f% (%s з %s)"),
                    time2str((time_t)(1e-9*tm_per),"%d-%m-%Y
%H:%M:%S").c_str(), 100*(double)(tm_end-tm_beg)/(double)(tm_per-tm_beg),
                    time2str(1e-3*(tm_end-tm_beg)).c_str(), time2str(1e-
3*(tm_per-tm_beg)).c_str()));
        }
        if(n_plc)
        {
            string plcV1 = _("Стандартний");
            if(it->second.policy == SCHED_RR) plcV1 = _("Кругова система
");
            #if __GLIBC__PREREQ(2,4)
                if(it->second.policy == SCHED_BATCH) plcV1 = _("Style
\"batch\"");
            #endif
            n_plc->childAdd("el")->setText(plcV1);
        }
        if(n_prior) n_prior->childAdd("el")->setText(TSYS::int2str(it-
>second.prior));
        if(n_cpuSet) n_cpuSet->childAdd("el")->setText(it-
>second.cpuSet);
    }
}
#if __GLIBC__PREREQ(2,4)
    if(multCPU() && ctrChkNode(opt,"set",RWRW__,"root","root",SEC_WR) && opt-
>attr("col") == "cpuSet")
    {
        ResAlloc res(taskRes,true);
        map<string,STask>::iterator it = mTasks.find(opt->attr("key_path"));
        if(it == mTasks.end()) throw TError(nodePath().c_str(),_("Не
представлене завдання '%s'."));
        if(it->second.flgs & STask::Detached) return;

        it->second.cpuSet = opt->text();

        cpu_set_t cpuset;
        CPU_ZERO(&cpuset);
        string sval;
        for(int off = 0; (sval=TSYS::strParse(it-
>second.cpuSet,0,":",&off)).size(); )
            CPU_SET(atoi(sval.c_str()),&cpuset);
        int rez = pthread_setaffinity_np(it->second.thr, sizeof(cpu_set_t),
&cpuset);
        res.release();
        TBDS::genDBSet(nodePath()+"CpuSet:"+it->first,opt->text());
        if(rez == EINVAL && opt->text().size()) throw
TError(nodePath().c_str(),_("Не встановлено жодних дозволених процесорів."));
        if(rez && opt->text().size()) throw TError(nodePath().c_str(),_("CPU
установки для потоку помилкові."));
    }
}
#endif
}

```

```

    if(!cntrEmpty() && a_path == "/cntr/cntr" &&
ctrChkNode(opt,"get",R_R___,"root","root"))
    {
        XMLNode *n_id      = ctrMkNode("list",opt,-
1,"/cntr/cntr/id","",R_R___,"root","root");
        XMLNode *n_vl     = ctrMkNode("list",opt,-
1,"/cntr/cntr/vl","",R_R___,"root","root");

        ResAlloc res( nodeRes(), false );
        for(map<string,double>::iterator icnt = mCntrs.begin(); icnt !=
mCntrs.end(); icnt++)
        {
            if(n_id)      n_id->childAdd("el")->setText(icnt->first);
            if(n_vl)      n_vl->childAdd("el")->setText(TSYS::real2str(icnt-
>second));
        }
        else if(a_path == "/hlp/g_help" &&
ctrChkNode(opt,"get",R_R___,"root","root",SEC_RD)) opt->setText(optDescr());
        else TCntrNode::cntrCmdProc(opt);
    }
}

```

K6П3\_2024

## resalloc.cpp - Розподіл ресурсів у системі

```

//Інтелектуальна система підтримки життєвого циклу з використанням IoT системний
файл: resalloc.cpp

#include "errno.h"

#include "tsys.h"
#include "resalloc.h"

using namespace IoT;

//*****
//* Об'єкт ресурсу *
//*****
Res::Res( )
{
#if !__GLIBC_PREREQ(2,4)
    wThr = 0;
#endif
    if(pthread_rwlock_init(&rw, NULL))
        throw TError("ResAlloc",_("Помилка відкриття семафору!"));
}

Res::~Res( )
{
    pthread_rwlock_wrlock(&rw);
    pthread_rwlock_destroy(&rw);
}

void Res::resRequestW( unsigned short tm )
{
    int rez = 0;
#if !__GLIBC_PREREQ(2,4)
    //EDEADLK імітація
    if(wThr && wThr == pthread_self()) rez == EDEADLK;
    else
#endif
    if(!tm) rez = pthread_rwlock_wrlock(&rw);
    else
    {
        timespec wtm;
        clock_gettime(CLOCK_REALTIME, &wtm);
        wtm.tv_nsec += 1000000*(tm%1000);
        wtm.tv_sec += tm/1000 + wtm.tv_nsec/1000000000; wtm.tv_nsec =
wtm.tv_nsec%1000000000;
        rez = pthread_rwlock_timedwrlock(&rw, &wtm);
    }
    if(rez == EDEADLK) throw TError(10,"ResAlloc",_("Ресурс пробує заблокувати
потік!"));
    else if(tm && rez == ETIMEDOUT) throw TError("ResAlloc",_("Ресурс не
відповідає!"));
#if !__GLIBC_PREREQ(2,4)
    wThr = pthread_self();
#endif
}

bool Res::resTryW( )
{
    int rez = pthread_rwlock_trywrlock(&rw);
    if(rez == EBUSY) return false;
    else if(rez == EDEADLK) throw TError(10,"ResAlloc",_("Ресурс пробує
заблокувати потік!"));
    return true;
}

void Res::resRequestR( unsigned short tm )

```

```

{
    int rez = 0;
#ifdef !_GLIBC_PREREQ(2,4)
    //EDEADLK імітація
    if(wThr && wThr == pthread_self()) rez == EDEADLK;
    else
#endif
    if(!tm) rez = pthread_rwlock_rdlock(&rw);
    else
    {
        timespec wtm;
        clock_gettime(CLOCK_REALTIME, &wtm);
        wtm.tv_nsec += 1000000*(tm%1000);
        wtm.tv_sec += tm/1000 + wtm.tv_nsec/1000000000; wtm.tv_nsec =
wtm.tv_nsec%1000000000;
        rez = pthread_rwlock_timedrdlock(&rw, &wtm);
    }
    if(rez == EDEADLK) throw TError(10, "ResAlloc", _("Ресурс пробує заблокувати
потік!"));
    else if(tm && rez == ETIMEDOUT) throw TError("ResAlloc", _("Ресурс не
відповідає!"));
}

bool Res::resTryR( )
{
    int rez = pthread_rwlock_tryrdlock(&rw);
    if(rez == EBUSY) return false;
    else if(rez == EDEADLK) throw TError(10, "ResAlloc", _("Ресурс пробує
заблокувати потік!"));
    return true;
}

void Res::resRelease( )
{
    pthread_rwlock_unlock(&rw);
#ifdef !_GLIBC_PREREQ(2,4)
    if(wThr == pthread_self()) wThr = 0;
#endif
}

//*****
/* Автоматичний розподіл ресурсів *
//*****
ResAlloc::ResAlloc( Res &rid ) : mId(rid), mAlloc(false)
{
}

ResAlloc::ResAlloc( Res &rid, bool write, unsigned short tm ) : mId(rid),
mAlloc(false)
{
    request(write, tm);
}

ResAlloc::~ResAlloc( )
{
    if(mAlloc) release();
}

void ResAlloc::request( bool write, unsigned short tm )
{
    if(mAlloc) release();
    mAlloc = false;
    try
    {
        if(write) mId.resRequestW(tm);
        else mId.resRequestR(tm);
        mAlloc = true;
    }catch(TError err) { if(err.cod!=10) throw; }
}

```

```

}

void ResAlloc::release()
{
    if(!mAlloc) return;
    mId.resRelease( );
    mAlloc = false;
}

//*****
//* Рядок + ресурс для *
//*****
ResString::ResString( const string &vl )
{
    pthread_mutex_init(&mRes, NULL);
    setVal(vl);
}

ResString::~ResString( )
{
    pthread_mutex_lock(&mRes);
    pthread_mutex_destroy(&mRes);
}

size_t ResString::size( ) { return getVal().size(); }

bool ResString::empty( ) { return getVal().empty(); }

void ResString::setVal( const string &vl )
{
    pthread_mutex_lock(&mRes);
    str = vl;
    pthread_mutex_unlock(&mRes);
}

string ResString::getVal( )
{
    string rez;
    pthread_mutex_lock(&mRes);
    rez = str;
    pthread_mutex_unlock(&mRes);
    return rez;
}

ResString &ResString::operator=( const string &val )
{
    setVal(val);
    return *this;
}

```

## tarchives.cpp - Робота з архівом

```

//Інтелектуальна система підтримки життєвого циклу з використанням IoT системний
файл: tarchives.cpp

#include <unistd.h>
#include <getopt.h>
#include <signal.h>
#include <sys/time.h>
#include <string.h>
#include <algorithm>

#include "tsys.h"
#include "tarchives.h"

#define BUF_SIZE_DEF 500
#define BUF_SIZE_MAX 100000

using namespace IoT;

//*****
//* Підсистема архівування *
//*****

//*****
//* TArchiveS *
//*****
TArchiveS::TArchiveS( ) :
    TSubSYS(SARH_ID,"Archives",true), elMess(""), elVal(""), elAval(""),
    bufErr(0), mMessPer(10), prcStMess(false),
    headBuf(0), headLstread(0), mValPer(1000), mValPrior(10), prcStVal(false),
    endrunReqVal(false)
{
    mAval = grpAdd("va_");

    //> архіватор повідомлення у структурі БД
    elMess.fldAdd( new TFld("ID",_("ID")),TFld::String,TCfg::Key,"20" );
    elMess.fldAdd( new TFld("MODUL",_(" Ім'я модуля
(плагіна)")),TFld::String,TCfg::Key,"20" );
    elMess.fldAdd( new
TFld("NAME",_(" Ім'я")),TFld::String,TCfg::TransltText,"50" );
    elMess.fldAdd( new
TFld("DESCR",_("Дескриптор")),TFld::String,TCfg::TransltText,"200" );
    elMess.fldAdd( new TFld("START",_("Початок архіву")),TFld::Boolean,0,"1" );
    elMess.fldAdd( new TFld("CATEG",_("Категорії
повідомлень")),TFld::String,0,"100" );
    elMess.fldAdd( new TFld("LEVEL",_("Рівні
повідомлень")),TFld::Integer,0,"1","","0;7" );
    elMess.fldAdd( new TFld("ADDR",_("Адреса")),TFld::String,0,"100" );

    //> Значення архіватора у структурі БД
    elVal.fldAdd( new TFld("ID",_("ID")),TFld::String,TCfg::Key,"20" );
    elVal.fldAdd( new TFld("MODUL",_(" Ім'я модуля
(плагіна)")),TFld::String,TCfg::Key,"20" );
    elVal.fldAdd( new TFld("NAME",_(" Ім'я
")),TFld::String,TCfg::TransltText,"50" );
    elVal.fldAdd( new
TFld("DESCR",_("Дескриптор")),TFld::String,TCfg::TransltText,"200" );
    elVal.fldAdd( new TFld("START",_("Початок архіву")),TFld::Boolean,0,"1","0"
);
    elVal.fldAdd( new TFld("ADDR",_("Адреса")),TFld::String,0,"50" );
    elVal.fldAdd( new TFld("V_PER",_("Value period
(sec)")),TFld::Real,0,"12.6","1","0;1000000" );
    elVal.fldAdd( new TFld("A_PER",_("Період архівації
(sec)")),TFld::Integer,0,"4","60","0;1000" );

    //> Значення архіву у структурі БД
    elAval.fldAdd( new TFld("ID",_("ID")),TFld::String,TCfg::Key,"20" );

```

```

    elAval.fldAdd( new TFld("NAME",_(" Ім'я
"),TFld::String,TCfg::TransltText,"50") );
    elAval.fldAdd( new
TFld("DESCR",_("Дескриптор"),TFld::String,TCfg::TransltText,"200") );
    elAval.fldAdd( new TFld("START",_("Початок архіву"),TFld::Boolean,0,"1","0")
);
    elAval.fldAdd( new TFld("SrcMode",_("Режим джерела"),TFld::Integer,0,"1") );
    elAval.fldAdd( new TFld("Source",_("Джерело"),TFld::String,0,"100") );
    elAval.fldAdd( new TFld("VTYPE",_("Тип значення"),TFld::Integer,0,"1") );
    elAval.fldAdd( new TFld("BPER",_("Період буферизації
(sec)"),TFld::Real,0,"9.6","1","0;10000") );
    elAval.fldAdd( new TFld("BSIZE",_("Розмір
буферу()"),TFld::Integer,0,"6","100","0;1000000") );
    elAval.fldAdd( new TFld("BHGRD",_("Буфер у режимі ґрид-
системи"),TFld::Boolean,0,"1","1") );
    elAval.fldAdd( new TFld("BHRES",_("Значення буфера останім
часом"),TFld::Boolean,0,"1","0") );
    elAval.fldAdd( new TFld("ArchS",_("Процес
архівування"),TFld::String,0,"500") );

    setMessBufLen( BUF_SIZE_DEF );

    //> Створення повідомлення часу архівування
    struct sigevent sigev;
    memset(&sigev,0,sizeof(sigev));
    sigev.sigev_notify = SIGEV_THREAD;
    sigev.sigev_value.sival_ptr = this;
    sigev.sigev_notify_function = ArhMessTask;
    sigev.sigev_notify_attributes = NULL;
    timer_create(CLOCK_REALTIME,&sigev,&tmIdMess);
}

TArchiveS::~TArchiveS( )
{
    //> Повідомлення про закінчення архівування
    timer_delete(tmIdMess);

    //> Переривання архівування
    if(prcStVal) SYS->taskDestroy(nodePath('.',true)+".vals", &endrunReqVal);

    //> Визволення усіх ресурсів
    nodeDelAll();
}

int TArchiveS::valPeriod( )          { return vmax(1,mValPer); }

void TArchiveS::setValPrior( int ivl ) { mValPrior = vmax(-1,vmin(99,ivl));
modif(); }

void TArchiveS::load_( )
{
    //> Завантажуємо параметри з командного рядка
    int next_opt;
    const char *short_opt="h";
    struct option long_opt[] =
    {
        {"help"      ,0,NULL,'h'},
        {NULL        ,0,NULL,0 }
    };

    optind=0,opterr=0;
    do
    {
        next_opt=getopt_long(SYS->argc,(char * const *)SYS-
>argv,short_opt,long_opt,NULL);
        switch(next_opt)
        {
            case 'h': fprintf(stdout,"%s",optDescr().c_str()); break;
            case -1 : break;

```

```

    }
    } while(next_opt != -1);

    //> Завантажуємо параметри
    setMessBufLen (
atoi (TBDS::genDBGet (nodePath ()+"MessBufSize", TSYs::int2str (messBufLen ())) .c_str (
)) );
    setMessPeriod (
atoi (TBDS::genDBGet (nodePath ()+"MessPeriod", TSYs::int2str (mMessPer)) .c_str ()) );
    setValPeriod (
atoi (TBDS::genDBGet (nodePath ()+"ValPeriod", TSYs::int2str (mValPer)) .c_str ()) );
    setValPrior (
atoi (TBDS::genDBGet (nodePath ()+"ValPriority", TSYs::int2str (mValPrior)) .c_str ())
);

    //> LidDB
    //>> Повідомлення завантаження архіватора
    string id,type;
    map<string, bool> itReg;
    try
    {
        TConfig c_el (&elMess);
        c_el.cfgViewAll (false);
        vector<string> db_ls;

        //>> Шукаємо у БД і створюємо новий архів
        SYS->db ().at ().dbList (db_ls, true);
        db_ls.push_back ("");
        for (unsigned i_db = 0; i_db < db_ls.size (); i_db++)
            for (int fld_cnt=0; SYS-
>db ().at ().dataSeek (db_ls [i_db] + "." + subId () + "_mess_proc", nodePath () + subId () + "_me
ss_proc", fld_cnt++, c_el); )
            {
                id = c_el.cfg ("ID").getS ();
                type = c_el.cfg ("MODUL").getS ();
                if (modPresent (type) && !at (type).at ().messPresent (id))
                    at (type).at ().messAdd (id, (db_ls [i_db] == SYS-
>workDB ()) ? "*" : db_ls [i_db]);
                itReg [type + "." + id] = true;
            }

        //>>> Перевіряємо для видалення з БД
        if (!SYS->selDB ().empty ())
        {
            vector<string> m_ls;
            modList (m_ls);
            for (unsigned i_m = 0; i_m < m_ls.size (); i_m++)
            {
                at (m_ls [i_m]).at ().messList (db_ls);
                for (unsigned i_it = 0; i_it < db_ls.size (); i_it++)
                    if (itReg.find (m_ls [i_m] + "." + db_ls [i_it]) == itReg.end () &&
SYS->chkSelDB (at (m_ls [i_m]).at ().messAt (db_ls [i_it]).at ().DB ()))
                        at (m_ls [i_m]).at ().messDel (db_ls [i_it]);
            }
        }
    } catch ( TError err )
    {
        mess_err (err.cat.c_str (), "%s", err.mess.c_str ());
        mess_err (nodePath ().c_str (), _ ("Повідомлення вомилки завантаження
архіватора."));
    }

    //>> Завантажуємо значення архіватора
    try
    {
        TConfig c_el (&elVal);
        c_el.cfgViewAll (false);
        vector<string> db_ls;
        itReg.clear ();
    }

```

```

//>> Шукаємо у БД та створюємо новий архів
SYS->db().at().dbList(db_ls,true);
db_ls.push_back("<cfg>");
for(unsigned i_db = 0; i_db < db_ls.size(); i_db++)
    for(int fld_cnt=0; SYS-
>db().at().dataSeek(db_ls[i_db]+"."+subId()+"_val_proc",nodePath()+subId()+"_val
_proc",fld_cnt++,c_el); )
    {
        id = c_el.cfg("ID").getS();
        type = c_el.cfg("MODUL").getS();
        if(modPresent(type) && !at(type).at().valPresent(id))
            at(type).at().valAdd(id,(db_ls[i_db]==SYS-
>workDB())?"*.*":db_ls[i_db]);
        itReg[type+"."+id] = true;
    }

//>>> Перевіряємо для видалення з БД
if(!SYS->selDB().empty())
    {
        vector<string> m_ls;
        modList(m_ls);
        for(unsigned i_m = 0; i_m < m_ls.size(); i_m++)
            {
                at(m_ls[i_m]).at().valList(db_ls);
                for(unsigned i_it = 0; i_it < db_ls.size(); i_it++)
                    if(itReg.find(m_ls[i_m]+"."+db_ls[i_it]) == itReg.end() &&
SYS->chkSelDB(at(m_ls[i_m]).at().valAt(db_ls[i_it]).at().DB()))
                        at(m_ls[i_m]).at().valDel(db_ls[i_it]);
            }
    }
} catch( TError err )
{
    mess_err(err.cat.c_str(),"%s",err.mess.c_str());
    mess_err(nodePath().c_str(),"Помилка завантаження значення
архіватора.");
}

//>> Завантажуємо значення архіватора
try
{
    TConfig c_el(&selAval);
    c_el.cfgViewAll(false);
    vector<string> db_ls;
    itReg.clear();

    //>> Шукаємо у БД та створюємо новий архів
SYS->db().at().dbList(db_ls,true);
db_ls.push_back("<cfg>");
for(unsigned i_db = 0; i_db < db_ls.size(); i_db++)
    for(int fld_cnt=0; SYS-
>db().at().dataSeek(db_ls[i_db]+"."+subId()+"_val",nodePath()+subId()+"_val",fld
_cnt++,c_el); )
    {
        id = c_el.cfg("ID").getS();
        if(!valPresent(id)) valAdd(id,(db_ls[i_db]==SYS-
>workDB())?"*.*":db_ls[i_db]);
        itReg[id] = true;
    }

    //>>> Перевіряємо для видалення з БД
    if(!SYS->selDB().empty())
        {
            valList(db_ls);
            for(unsigned i_it = 0; i_it < db_ls.size(); i_it++)
                if(itReg.find(db_ls[i_it]) == itReg.end() && SYS-
>chkSelDB(valAt(db_ls[i_it]).at().DB()))
                    valDel(db_ls[i_it]);
        }
    }
}

```

```

    }catch(TError err)
    {
        mess_err(err.cat.c_str(),"%s",err.mess.c_str());
        mess_err(nodePath().c_str(),_("Помилка завантаження значення
архіватора."));
    }
}

void TArchiveS::save_( )
{
    vector<string> t_lst, o_lst;

    //> Зберігаємо параметри
    TBDS::genDBSet (nodePath()+"MessBufSize",TSYS::int2str(messBufLen()));
    TBDS::genDBSet (nodePath()+"MessPeriod",TSYS::int2str(messPeriod()));
    TBDS::genDBSet (nodePath()+"ValPeriod",TSYS::int2str(valPeriod()));
    TBDS::genDBSet (nodePath()+"ValPriority",TSYS::int2str(valPrior()));
}

void TArchiveS::valAdd( const string &iid, const string &idb )
{
    if( valPresent(iid) ) return;
    chldAdd(mAval,new TVArchive(iid,idb,&aVale()));
}

string TArchiveS::optDescr( )
{
    char buf[STR_BUF_LEN];
    snprintf(buf,sizeof(buf),_(
        "===== Підсистема \"Архів\" Опції
=====\\n"
        "----- Параметри частини '%s' у конфігураційний файл -----\\n"
        "MessBufSize <items>      Повідомлення розміру буферу.\\n"
        "MessPeriod <sec>          Повідомлення періоду архівування.\\n"
        "ValPeriod <msec>          Значення періоду архівування.\\n"
        "ValPriority <level>       Значення завдання пріоритетного рівня.\\n"
        "MaxReqMess <items>       Повідомлення максимального запиту.\\n"
        "MaxReqVals <items>       Значення максимального запиту.\\n\\n"
    ),nodePath().c_str());

    return buf;
}

void TArchiveS::subStart( )
{
    mess_info(nodePath().c_str(),_("Запускаємо підсистеми."));

    SubStarting = true;

    vector<string> t_lst, o_lst;

    modList(t_lst);
    for(unsigned i_t = 0; i_t < t_lst.size(); i_t++)
    {
        AutoHD<TTipArchivator> mod = modAt(t_lst[i_t]);

        //> Повідомлення про початок роботи архіватора
        mod.at().messList(o_lst);
        for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
        {
            AutoHD<TMArchivator> mess = mod.at().messAt(o_lst[i_o]);
            if( /*!mess.at().startStat() &&*/ mess.at().toStart() )
                try{ mess.at().start(); }
                catch(TError err)
                {
                    mess_err(err.cat.c_str(),"%s",err.mess.c_str());
                    mess_err(nodePath().c_str(),_("Повідомлення про помилку роботи
архіватора."),o_lst[i_o].c_str());
                }
        }
    }
}

```

```

}
//> Значення початку роботи архіватора
mod.at().valList(o_lst);
for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
{
    AutoHD<TVArchivator> val = mod.at().valAt(o_lst[i_o]);
    if( /*!val.at().startStat() &&*/ val.at().toStart() )
        try{ val.at().start(); }
        catch(TError err)
        {
            mess_err(err.cat.c_str(), "%s", err.mess.c_str());
            mess_err(nodePath().c_str(), _("Помилка початку роботи
архіватора."), val.at().workId().c_str());
        }
    }
}

//> Значення початку роботи архіватора
vallList(o_lst);
for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
{
    AutoHD<TVArchive> aval = valAt(o_lst[i_o]);
    if( /*!aval.at().startStat() &&*/ aval.at().toStart() )
        try{ aval.at().start(); }
        catch(TError err)
        {
            mess_err(err.cat.c_str(), "%s", err.mess.c_str());
            mess_err(nodePath().c_str(), _("Помилка початку роботи
архіватора"), o_lst[i_o].c_str());
        }
    }

//> Повідомлення про початок роботи інтервального таймера
struct itimerspec itval;
itval.it_interval.tv_sec = itval.it_value.tv_sec = messPeriod();
itval.it_interval.tv_nsec = itval.it_value.tv_nsec = 0;
timer_settime(tmIdMess, 0, &itval, NULL);

//> Значення початку роботи завдань
if(!prcStVal) SYS->taskCreate(nodePath('.', true) + ".vals", valPrior(),
TArchiveS::ArhValTask, this);

TSubSYS::subStart( );

SubStarting = false;
}

void TArchiveS::subStop( )
{
    mess_info(nodePath().c_str(), _("Зупинка підсистеми."));

    TSubSYS::subStop( );

    vector<string> t_lst, o_lst;

    //> Зупинка інтервального таймера для періодичних потоків, для створення
повідомлення архівації структур;
    itval.it_interval.tv_sec = itval.it_value.tv_sec =
        itval.it_interval.tv_nsec = itval.it_value.tv_nsec = 0;
    timer_settime(tmIdMess, 0, &itval, NULL);
    if(TSYS::eventWait( prcStMess, false, nodePath() + "mess_stop", 10))
        throw TError(nodePath().c_str(), _("Архівація повідомлень потоків не може
бути зупинена!"));

    //> Значення зупинки роботи завдань
    if(prcStVal) SYS->taskDestroy(nodePath('.', true) + ".vals", &endrunReqVal);

    //> Виклик останнього повідомлення архіватора
    sigval obj; obj.sival_ptr = this;

```

```

ArhMessTask(obj);

//> Зупинка архіватора
modList(t_lst);
for(unsigned i_t = 0; i_t < t_lst.size(); i_t++)
{
    AutoHD<TTipArchivator> mod = modAt(t_lst[i_t]);
    //Значення зупинки архіватора
    mod.at().vallList(o_lst);
    for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
    {
        AutoHD<TVArchivator> val = mod.at().valAt(o_lst[i_o]);
        if( val.at().startStat() )
            try{ val.at().stop(); }
            catch(TError err)
            {
                mess_err(err.cat.c_str(), "%s", err.mess.c_str());
                mess_err(nodePath().c_str(), _("Значення архіватора '%s' stop
error."), o_lst[i_o].c_str());
            }
    }
    // Повідомлення про зупинку архіватора
    mod.at().messList(o_lst);
    for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
    {
        AutoHD<TMArchivator> mess = mod.at().messAt(o_lst[i_o]);
        if( mess.at().startStat() )
            try{ mess.at().stop(); }
            catch(TError err)
            {
                mess_err(err.cat.c_str(), "%s", err.mess.c_str());
                mess_err(nodePath().c_str(), _("Повідомлення про помилку зупинки
архіватора"), o_lst[i_o].c_str());
            }
    }
}

//> Значення зупинки архіватора
vallList(o_lst);
for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
{
    AutoHD<TVArchive> aval = valAt(o_lst[i_o]);
    if( aval.at().startStat() )
        try{ aval.at().stop(); }
        catch(TError err)
        {
            mess_err(err.cat.c_str(), "%s", err.mess.c_str());
            mess_err(nodePath().c_str(), _("Значення помилки зупинки
архіватора"), o_lst[i_o].c_str());
        }
}
}

void TArchiveS::messPut( time_t tm, int utm, const string &categ, int8_t level,
const string &mess )
{
    //> Відправляємо повідомлення у буфер
    ResAlloc res(mRes, true);
    mBuf[headBuf].time = tm;
    mBuf[headBuf].utime = utm;
    mBuf[headBuf].categ = categ;
    mBuf[headBuf].level = (TMess::Type)abs(level);
    mBuf[headBuf].mess = mess;
    if( ++headBuf >= mBuf.size() ) headBuf = 0;
    //> Пеервіряємо, чи це не повідомлення архіватора
    if( headBuf == headLstread )
    {
        if( !(bufErr&0x01) )
        {

```

```

        bufErr |= 0x01;
        res.release();
        mess_err(nodePath().c_str(),_("Буфер заповнений. Останнє
повідомлення!"));
        res.request(true);
    }
    if( ++headLstread >= mBuf.size() ) headLstread = 0;
}
//> Перевіряємо швидкість заповнення буфера.
else if( headBuf-headLstread > messBufLen( )/2 )
{
    if( !(bufErr&0x02) )
    {
        bufErr |= 0x02;
        res.release();
        mess_warning(nodePath().c_str(),_("Повідомлення про т, що швидкість
заповнення буфера дуже висока!"));
        res.request(true);
    }
}
else bufErr = 0;

//> Обробка тривоги. Для рівня менше 0 тривоги встановлено
map<string, TMess::SRec>::iterator p;
if( level < 0 ) mAlarms[categ] =
TMess::SRec(tm,utm,categ, (TMess::Type)abs(level),mess);
else if( (p=mAlarms.find(categ)) != mAlarms.end() ) mAlarms.erase(p);
}

void TArchiveS::messPut( const vector<TMess::SRec> &recs )
{
    for(unsigned i_r = 0; i_r < recs.size(); i_r++)
        messPut(recs[i_r].time,recs[i_r].utime,recs[i_r].categ,recs[i_r].level,recs[i_r].mess);
}

void TArchiveS::messGet( time_t b_tm, time_t e_tm, vector<TMess::SRec> & recs,
const string &category, int8_t level, const string &arch, time_t upTo )
{
    recs.clear();

    ResAlloc res(mRes,false);
    if(!upTo) upTo = time(NULL)+STD_INTERF_TM;
    TRegExp re(category, "p");

    //> Отримуємо записи з буфера
    unsigned i_buf = headBuf;
    while(level >= 0 && (!arch.size() || arch==BUF_ARCH_NM) && time(NULL) <
upTo)
    {
        if(mBuf[i_buf].time >= b_tm && mBuf[i_buf].time != 0 && mBuf[i_buf].time
<= e_tm &&
            mBuf[i_buf].level >= level && re.test(mBuf[i_buf].categ))
            recs.push_back(mBuf[i_buf]);
        if(++i_buf >= mBuf.size()) i_buf = 0;
        if(i_buf == headBuf) break;
    }

    //> Отримуємо записи з архівів
    vector<string> t_lst, o_lst;
    modList(t_lst);
    for(unsigned i_t = 0; level >= 0 && i_t < t_lst.size(); i_t++)
    {
        at(t_lst[i_t]).at().messList(o_lst);
        for(unsigned i_o = 0; i_o < o_lst.size() && time(NULL) < upTo; i_o++)
        {
            AutoHD<TMArchivator> archtor = at(t_lst[i_t]).at().messAt(o_lst[i_o]);
            if(archtor.at().startStat() && (!arch.size() ||
arch==archtor.at().workId()))

```

```

        archtor.at().get(b_tm,e_tm,recs,category,level);
    }
}

//> Запит процесу тривоги
if(level < 0)
{
    vector< pair<int64_t,TMess::SRec* > > mb;
    for(map<string,TMess::SRec>::iterator p = mAlarms.begin(); p !=
mAlarms.end() && time(NULL) < upTo; p++)
        if((p->second.time >= b_tm || b_tm == e_tm) && p->second.time <= e_tm
&&
            p->second.level >= abs(level) && re.test(p->second.categ))
            mb.push_back(pair<int64_t,TMess::SRec* >(FTM(p->second), &p-
>second));
    sort(mb.begin(),mb.end());
    for(unsigned i_b = 0; i_b < mb.size(); i_b++)
recs.push_back(*mb[i_b].second);
}
}

time_t TArchiveS::messBeg( const string &arch )
{
    time_t rez = 0;
    ResAlloc res(mRes,false);
    if(arch.empty() || arch == BUF_ARCH_NM)
    {
        unsigned i_buf = headBuf;
        while(!arch.size() || arch == BUF_ARCH_NM)
        {
            rez = rez ? vmin(rez,mBuf[i_buf].time) : mBuf[i_buf].time;
            if(++i_buf >= mBuf.size()) i_buf = 0;
            if(i_buf == headBuf) break;
        }
        if( !arch.empty() ) return rez;
    }

    //- Отримуємо записи з архівів -
    vector<string> t_lst, o_lst;
    modList(t_lst);
    AutoHD<TMArchivator> archtor;
    for(unsigned i_t = 0; i_t < t_lst.size(); i_t++)
    {
        at(t_lst[i_t]).at().messList(o_lst);
        for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
        {
            archtor = at(t_lst[i_t]).at().messAt(o_lst[i_o]);
            if(archtor.at().startStat() && (!arch.size() ||
arch==archtor.at().workId()))
                rez = rez ? vmin(rez,archtor.at().begin()) : archtor.at().begin();
        }
    }

    return rez;
}

time_t TArchiveS::messEnd( const string &arch )
{
    time_t rez = 0;
    ResAlloc res(mRes,false);
    if(arch.empty() || arch == BUF_ARCH_NM)
    {
        unsigned i_buf = headBuf;
        while(!arch.size() || arch == BUF_ARCH_NM)
        {
            rez = rez ? vmax(rez,mBuf[i_buf].time) : mBuf[i_buf].time;
            if(++i_buf >= mBuf.size()) i_buf = 0;
            if(i_buf == headBuf) break;
        }
    }
}

```

```

    if(!arch.empty()) return rez;
}

//> Отримуюмо записи з архівів
vector<string> t_lst, o_lst;
modList(t_lst);
AutoHD<TMArchivator> archtor;
for(unsigned i_t = 0; i_t < t_lst.size(); i_t++)
{
    at(t_lst[i_t]).at().messList(o_lst);
    for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
    {
        archtor = at(t_lst[i_t]).at().messAt(o_lst[i_o]);
        if(archtor.at().startStat() && (!arch.size() ||
arch==archtor.at().workId()))
            rez = rez ? vmax(rez,archtor.at().end()) : archtor.at().end();
    }
}

return rez;
}

void TArchiveS::setMessBufLen(unsigned len)
{
    ResAlloc res(mRes,true);
    len = vmin(BUF_SIZE_MAX,vmax(BUF_SIZE_DEF,len));
    while(mBuf.size() > len)
    {
        mBuf.erase(mBuf.begin() + headBuf);
        if(headBuf >= mBuf.size()) headBuf = 0;
        if(headLstread >= mBuf.size()) headLstread = mBuf.size()-1;
    }
    while(mBuf.size() < len) mBuf.insert(mBuf.begin() + headBuf, TMess::SRec());
    modif();
}

void TArchiveS::setActValArch( const string &id, bool val )
{
    unsigned i_arch;

    ResAlloc res(vRes,true);
    for( i_arch = 0; i_arch < actUpSrc.size(); i_arch++ )
        if( actUpSrc[i_arch].at().id() == id ) break;

    if( val && i_arch >= actUpSrc.size() )
        actUpSrc.push_back(valAt(id));
    if( !val && i_arch < actUpSrc.size() )
        actUpSrc.erase(actUpSrc.begin()+i_arch);
}

void TArchiveS::setMessPeriod( int ivl )
{
    mMessPer = ivl;
    modif();

    if( subStartStat( ) )
    {
        struct itimerspec itval;
        itval.it_interval.tv_sec = itval.it_value.tv_sec = mMessPer;
        itval.it_interval.tv_nsec = itval.it_value.tv_nsec = 0;
        timer_settime(tmIdMess, 0, &itval, NULL);
    }
}

void TArchiveS::ArhMessTask( union signal obj )
{
    TArchiveS &arh = *(TArchiveS *)obj.sival_ptr;
    if( arh.prcStMess ) return;
    arh.prcStMess = true;
}

```

```

//> Читаємо повідомлення з буфера
if( arh.headLstread != arh.headBuf )
  try
  {
    ResAlloc res(arh.mRes,false);

    //> Беремо нове повідомлення
    unsigned new_headLstread = arh.headBuf;
    unsigned i_m = arh.headLstread;
    vector<TMess::SRec> o_mess;
    while( i_m != new_headLstread )
    {
      o_mess.push_back(arh.mBuf[i_m]);
      if( ++i_m >= arh.mBuf.size() ) i_m = 0;
    }
    arh.headLstread = new_headLstread;

    res.release();

    //> Архівуємо
    vector<string> t_lst, o_lst;
    arh.modList(t_lst);
    for(unsigned i_t = 0; i_t < t_lst.size(); i_t++)
    {
      arh.at(t_lst[i_t]).at().messList(o_lst);
      for(unsigned i_o = 0; i_o < o_lst.size(); i_o++)
        if(arh.at(t_lst[i_t]).at().messAt(o_lst[i_o]).at().startStat())
          arh.at(t_lst[i_t]).at().messAt(o_lst[i_o]).at().put(o_mess);
    }
  }
  catch(TError err)
  {
    mess_err(err.cat.c_str(),"%s",err.mess.c_str());
    mess_err(arh.nodePath().c_str(),"Помилка читання повідомлення з
буфера.");
  }

  arh.prcStMess = false;
}

void *TArchiveS::ArhValTask( void *param )
{
  TArchiveS &arh = *(TArchiveS *)param;
  arh.endrunReqVal = false;
  arh.prcStVal = true;

  while( !arh.endrunReqVal )
  {
    int64_t work_tm = SYS->curTime();

    arh.vRes.resRequestR( );
    for(unsigned i_arh = 0; i_arh < arh.actUpSrc.size(); i_arh++)
      try
      {
        {
          if( work_tm/arh.actUpSrc[i_arh].at().period() >
arh.actUpSrc[i_arh].at().end()/arh.actUpSrc[i_arh].at().period() )
            arh.actUpSrc[i_arh].at().getActiveData();
        }
        catch(TError err)
        { mess_err(err.cat.c_str(),"%s",err.mess.c_str()); }
        arh.vRes.resRelease( );

        TSYS::taskSleep((int64_t)arh.valPeriod()*1000000);
      }

    arh.prcStVal = false;

    return NULL;
  }
}

```

```

}

TVariant TArchiveS::objFuncCall( const string &iid, vector<TVariant> &prms,
const string &user )
{
    // Array messGet(int btm, int etm, string cat = "", int lev = 0, string arch
= ""); - запит системних повідомлень, для часу з <btm>
    // до <etm> для категорії <cat>, рівня <lev> та архіву <arch>
    // btm - час початку
    // etm - час закінчення
    // cat - категорія повідомлення
    // lev - рівні повідомлень
    // arch - архівація повідомлення
    if( iid == "messGet" && prms.size() >= 2 )
    {
        vector<TMess::SRec> recs;
        messGet( prms[0].getI(), prms[1].getI(), recs, ((prms.size())>=3) ?
prms[2].getS() : string(""),
            ((prms.size())>=4) ? prms[3].getI() : 0, ((prms.size())>=5) ?
prms[4].getS() : string("")) );
        TArrayObj *rez = new TArrayObj();
        for(unsigned i_m = 0; i_m < recs.size(); i_m++)
        {
            TVarObj *am = new TVarObj();
            am->propSet("tm", (int)recs[i_m].time);
            am->propSet("utm", recs[i_m].utime);
            am->propSet("categ", recs[i_m].categ);
            am->propSet("level", recs[i_m].level);
            am->propSet("mess", recs[i_m].mess);
            rez->propSet(TSYS::int2str(i_m), am);
        }
        return rez;
    }

    return TCntrNode::objFuncCall(iid, prms, user);
}

void TArchiveS::cntrCmdProc( XMLNode *opt )
{
    string a_path = opt->attr("path");
    //> Сервіс управління процесами
    if(a_path == "/serv/mess") //Повідомлення доступу
    {
        if(ctrChkNode(opt, "info", RWRWRW, "root", SARH_ID, SEC_RD))
        //Інформаційні повідомлення
        {
            string arch = opt->attr("arch");
            opt->setAttr("end", TSYS::uint2str(messEnd(arch)));
            opt->setAttr("beg", TSYS::uint2str(messBeg(arch)));
        }
        else if(ctrChkNode(opt, "get", RWRWRW, "root", SARH_ID, SEC_RD)) //Запит
значення дати
        {
            time_t tm = strtoul(opt->attr("tm").c_str(), 0, 10);
            time_t tm_grnd = strtoul(opt->attr("tm_grnd").c_str(), 0, 10);
            string arch = opt->attr("arch");
            string cat = opt->attr("cat");
            int lev = atoi(opt->attr("lev").c_str());
            vector<TMess::SRec> rez;
            messGet( tm_grnd, tm, rez, cat, (TMess::Type)lev, arch );
            for(unsigned i_r = 0; i_r < rez.size(); i_r++)
                opt->childAdd("el")->
                    setAttr("time", TSYS::uint2str(rez[i_r].time))->
                    setAttr("utime", TSYS::uint2str(rez[i_r].utime))->
                    setAttr("cat", rez[i_r].categ)->
                    setAttr("lev", TSYS::int2str(rez[i_r].level))->
                    setText(rez[i_r].mess);
        }
    }
    return;
}

```

```

}

//> Беремо сторінку інформації
if(opt->name() == "info")
{
    TSubSYS::cntrCmdProc(opt);
    ctrMkNode("grp",opt,-1,"/br/va_",_("Архів
значень"),RWRWR_,"root",SARH_ID,2,"idm","1","idSz","20");
    if(ctrMkNode("area",opt,1,"/m_arch",_("Архів
повідомлень"),R_R_R_,"root",SARH_ID))
    {
        ctrMkNode("fld",opt,-1,"/m_arch/size",_("Повідомлення розміру
буферу"),RWRWR_,"root",SARH_ID,2,"tp","dec","min",TSYS::int2str(BUF_SIZE_DEF).c_
str());
        ctrMkNode("fld",opt,-1,"/m_arch/per",_("Період архівування
(s)"),RWRWR_,"root",SARH_ID,1,"tp","dec");
        if(ctrMkNode("area",opt,-1,"/m_arch/view",_("Перегляд
повідомлень"),R_R___,"root",SARH_ID))
        {
            ctrMkNode("fld",opt,-
1,"/m_arch/view/tm",_("Time"),RWRW___,"root",SARH_ID,1,"tp","time");
            ctrMkNode("fld",opt,-1,"/m_arch/view/size",_("Розмір
(s)"),RWRW___,"root",SARH_ID,1,"tp","dec");
            ctrMkNode("fld",opt,-1,"/m_arch/view/cat",_("Зразок
категорії"),RWRW___,"root",SARH_ID,2,"tp","str","help",
_("Шаблон категорії повідомлень або регулярне вираження.\n"
_("Використовуйте тимчасові символи для групового
виділення:\n ' ' - будь-які підрядки;\n '?' - будь-які символи.\n"
_("Регулярне вираження складається з символів/'/
(/mod_(System|LogicLev)/)."));
            ctrMkNode("fld",opt,-
1,"/m_arch/view/lvl",_("Рівень"),RWRW___,"root",SARH_ID,4,"tp","dec","min","-
7","max","7",
            "help",_("Отримуємо повідомлення для рівня більшого і
дорівнюючого цьому."));
            ctrMkNode("fld",opt,-
1,"/m_arch/view/archtor",_("Архіватор"),RWRW___,"root",SARH_ID,4,"tp","str","dest
","select","select","/m_arch/lstAMess",
            "help",_("Архівація повідомлення.\n Не встановлюємо архіватор
для процесів у буфері та інших архіваторів.\nВстановлюємо '<buffer>' для процесів
у буфері ."));
            if(ctrMkNode("table",opt,-
1,"/m_arch/view/mess",_("Messages"),R_R___,"root",SARH_ID))
            {
                ctrMkNode("list",opt,-
1,"/m_arch/view/mess/0",_("Time"),R_R___,"root",SARH_ID,1,"tp","time");
                ctrMkNode("list",opt,-
1,"/m_arch/view/mess/0a",_("mcsec"),R_R___,"root",SARH_ID,1,"tp","dec");
                ctrMkNode("list",opt,-
1,"/m_arch/view/mess/1",_("Category"),R_R___,"root",SARH_ID,1,"tp","str");
                ctrMkNode("list",opt,-
1,"/m_arch/view/mess/2",_("Lev."),R_R___,"root",SARH_ID,1,"tp","dec");
                ctrMkNode("list",opt,-
1,"/m_arch/view/mess/3",_("Message"),R_R___,"root",SARH_ID,1,"tp","str");
            }
        }
    }
    if(ctrMkNode("area",opt,2,"/v_arch",_("Архів
значень"),R_R_R_,"root",SARH_ID))
    {
        ctrMkNode("fld",opt,-1,"/v_arch/per",_("Беремо дані періоду
(ms)"),RWRWR_,"root",SARH_ID,1,"tp","dec");
        ctrMkNode("fld",opt,-1,"/v_arch/prior",_("Беремо дані завдання
пріоритетного рівня"),RWRWR_,"root",SARH_ID,1,"tp","dec");
        ctrMkNode("fld",opt,-
1,"/v_arch/nmb",_("Number"),R_R_R_,"root",SARH_ID,1,"tp","str");
        ctrMkNode("list",opt,-1,"/v_arch/archs",_("Архів
значень"),RWRWR_,"root",SARH_ID,5,"tp","br","idm","1","s_com","add,del","br_pref
","va_","idSz","20");
    }
}

```

```

    }
    ctrMkNode("fld",opt,-1,"/help/g_help",_("Опції
допомоги"),R_R____,"root",SARH_ID,3,"tp","str","cols","90","rows","10");
    return;
}

//> Процес управління на сторінці
if(a_path == "/m_arch/per")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))    opt->
>setText(TSYS::int2str(messPeriod()));
    if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))
    setMessPeriod(atoi(opt->text().c_str()));
}
else if(a_path == "/m_arch/size")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))    opt->
>setText(TSYS::int2str(messBufLen()));
    if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))
    setMessBufLen(atoi(opt->text().c_str()));
}
else if(a_path == "/m_arch/view/tm")
{
    if(ctrChkNode(opt,"get",RWRW____,"root",SARH_ID,SEC_RD))
    {
        opt->setText(TBDS::genDBGet(nodePath()+"messTm","0",opt->
>attr("user")));
        if(!atoi(opt->text().c_str()))    opt->
>setText(TSYS::int2str(time(NULL)));
    }
    if(ctrChkNode(opt,"set",RWRW____,"root",SARH_ID,SEC_WR))
        TBDS::genDBSet(nodePath()+"messTm",(atoi(opt->
>text().c_str())>=time(NULL))?"0":opt->text(),opt->attr("user"));
}
else if(a_path == "/m_arch/view/size")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))    opt->
>setText(TBDS::genDBGet(nodePath()+"messSize","60",opt->attr("user")));
    if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))
        TBDS::genDBSet(nodePath()+"messSize",opt->text(),opt->attr("user"));
}
else if(a_path == "/m_arch/view/cat")
{
    if(ctrChkNode(opt,"get",RWRW____,"root",SARH_ID,SEC_RD))    opt->
>setText(TBDS::genDBGet(nodePath()+"messCat","",opt->attr("user")));
    if(ctrChkNode(opt,"set",RWRW____,"root",SARH_ID,SEC_WR))
        TBDS::genDBSet(nodePath()+"messCat",opt->text(),opt->attr("user"));
}
else if(a_path == "/m_arch/view/lvl")
{
    if(ctrChkNode(opt,"get",RWRW____,"root",SARH_ID,SEC_RD))    opt->
>setText(TBDS::genDBGet(nodePath()+"messLev","0",opt->attr("user")));
    if(ctrChkNode(opt,"set",RWRW____,"root",SARH_ID,SEC_WR))
        TBDS::genDBSet(nodePath()+"messLev",opt->text(),opt->attr("user"));
}
else if(a_path == "/m_arch/view/archtor")
{
    if(ctrChkNode(opt,"get",RWRW____,"root",SARH_ID,SEC_RD))    opt->
>setText(TBDS::genDBGet(nodePath()+"messArch","",opt->attr("user")));
    if(ctrChkNode(opt,"set",RWRW____,"root",SARH_ID,SEC_WR))
        TBDS::genDBSet(nodePath()+"messArch",opt->text(),opt->attr("user"));
}
else if(a_path == "/m_arch/lstAMess" && ctrChkNode(opt,"get",R_R____))
{
    opt->childAdd("el")->setText("");
    opt->childAdd("el")->setText(BUF_ARCH_NM);
    vector<string> lsm, lsa;
    modList(lsm);
    for(unsigned i_m = 0; i_m < lsm.size(); i_m++)

```

```

    {
        at(lsm[i_m]).at().messList(lsa);
        for(unsigned i_a = 0; i_a < lsa.size(); i_a++)
            opt->childAdd("el")->setText(lsm[i_m]+"."+lsa[i_a]);
    }
}
else if(a_path == "/m_arch/view/mess" &&
ctrChkNode(opt,"get",R_R____,"root",SARH_ID))
{
    vector<TMess::SRec> rec;
    time_t gtm = atoi(TBDS::genDBGet(nodePath()+"messTm","0",opt-
>attr("user")).c_str());
    if(!gtm) gtm = time(NULL);
    int gsz = atoi(TBDS::genDBGet(nodePath()+"messSize","60",opt-
>attr("user")).c_str());
    messGet(gtm-gsz, gtm, rec,
            TBDS::genDBGet(nodePath()+"messCat","",opt->attr("user")),
            atoi(TBDS::genDBGet(nodePath()+"messLev","0",opt-
>attr("user")).c_str()),
            TBDS::genDBGet(nodePath()+"messArch","",opt->attr("user")) );

    XMLNode *n_tm      = ctrMkNode("list",opt,-
1,"/m_arch/view/mess/0","",R_R____,"root",SARH_ID);
    XMLNode *n_tmu     = ctrMkNode("list",opt,-
1,"/m_arch/view/mess/0a","",R_R____,"root",SARH_ID);
    XMLNode *n_cat     = ctrMkNode("list",opt,-
1,"/m_arch/view/mess/1","",R_R____,"root",SARH_ID);
    XMLNode *n_lvl     = ctrMkNode("list",opt,-
1,"/m_arch/view/mess/2","",R_R____,"root",SARH_ID);
    XMLNode *n_mess    = ctrMkNode("list",opt,-
1,"/m_arch/view/mess/3","",R_R____,"root",SARH_ID);
    for(int i_rec = rec.size()-1; i_rec >= 0; i_rec--)
    {
        if(n_tm)      n_tm->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].time));
        if(n_tmu)     n_tmu->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].utime));
        if(n_cat)     n_cat->childAdd("el")->setText(rec[i_rec].categ);
        if(n_lvl)     n_lvl->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].level));
        if(n_mess)    n_mess->childAdd("el")->setText(rec[i_rec].mess);
    }
}
else if(a_path == "/v_arch/per")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))      opt-
>setText(TSYS::int2str(valPeriod()));
    if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))
        setValPeriod(atoi(opt->text().c_str()));
}
else if(a_path == "/v_arch/prior")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))      opt-
>setText(TSYS::int2str(valPrior()));
    if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))
        setValPrior(atoi(opt->text().c_str()));
}
else if(a_path == "/v_arch/nmb" && ctrChkNode(opt))
{
    vector<string> list;
    vallist(list);
    unsigned e_c = 0;
    for(unsigned i_a = 0; i_a < list.size(); i_a++)
        if(valAt(list[i_a]).at().startStat()) e_c++;
    opt->setText(TSYS::strMess_("All: %d; Enabled: %d"),list.size(),e_c);
}
else if(a_path == "/br/va_" || a_path == "/v_arch/archs")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))

```

```

    {
        vector<string> list;
        valList(list);
        for(unsigned i_a=0; i_a < list.size(); i_a++)
            opt->childAdd("el")->setAttr("id",list[i_a])-
>setText(valAt(list[i_a]).at().name());
    }
    if(ctrChkNode(opt,"add",RWRWR_,"root",SARH_ID,SEC_WR)
    {
        string vid = TSYS::strEncode(opt->attr("id"),TSYS::oscdID);
        valAdd(vid); valAt(vid).at().setName(opt->text());
    }
    if(ctrChkNode(opt,"del",RWRWR_,"root",SARH_ID,SEC_WR)
    chldDel(mAval,opt->attr("id"),-1,1);
    }
    else if(a_path == "/help/g_help" &&
ctrChkNode(opt,"get",R_R___,"root",SARH_ID)    opt->setText(optDescr());
    else TSubSYS::cntrCmdProc(opt);
    }

//*****
/* TTipArchivator *
//*****
TTipArchivator::TTipArchivator( const string &id ) : TModule(id)
{
    mVal = grpAdd("val_");
    mMess = grpAdd("mess_");
}

TTipArchivator::~~TTipArchivator()
{
    nodeDelAll();
}

TArchiveS &TTipArchivator::owner()
{
    return (TArchiveS &)TModule::owner();
}

void TTipArchivator::messAdd(const string &name, const string &idb )
{
    chldAdd(mMess, AMess(name,idb));
}

void TTipArchivator::valAdd( const string &iid, const string &idb )
{
    chldAdd(mVal, AVal(iid,idb));
}

void TTipArchivator::cntrCmdProc( XMLNode *opt )
{
    //> Беремо сторінку інформації
    if(opt->name() == "info")
    {
        TModule::cntrCmdProc(opt);
        ctrMkNode("area",opt,0,"/arch",_("Архіватори"));
        ctrMkNode("grp",opt,-1,"/br/mess_",_("Повідомлення
архіватора"),RWRWR_,"root",SARH_ID,2,"idm","1","idSz","20");
        ctrMkNode("grp",opt,-1,"/br/val_",_("Значення
архіватора"),RWRWR_,"root",SARH_ID,2,"idm","1","idSz","20");
        ctrMkNode("list",opt,-1,"/arch/mess",_("Повідомлення
архіваторів"),RWRWR_,"root",SARH_ID,5,"tp","br","idm","1","s_com","add,del","br_
pref","mess_","idSz","20");
        ctrMkNode("list",opt,-1,"/arch/val",_("Значення
архіваторів"),RWRWR_,"root",SARH_ID,5,"tp","br","idm","1","s_com","add,del","br_
pref","val_","idSz","20");
        return;
    }
    //> Процес управління на сторінці

```

```

string a_path = opt->attr("path");
if(a_path == "/br/mess_" || a_path == "/arch/mess")
{
    if(ctrChkNode(opt, "get", RWRWR_, "root", SARH_ID, SEC_RD)
    {
        vector<string> list;
        messList(list);
        for( unsigned i_a=0; i_a < list.size(); i_a++ )
            opt->childAdd("el")->setAttr("id", list[i_a])->
>setText(messAt(list[i_a]).at().name());
    }
    if(ctrChkNode(opt, "add", RWRWR_, "root", SARH_ID, SEC_WR)
    {
        string vid = TSYS::strEncode(opt->attr("id"), TSYS::oscdID);
        messAdd(vid); messAt(vid).at().setName(opt->text());
    }
    if(ctrChkNode(opt, "del", RWRWR_, "root", SARH_ID, SEC_WR)      messDel(opt->
>attr("id"), true);
    }
    else if(a_path == "/br/val_" || a_path == "/arch/val")
    {
        if(ctrChkNode(opt, "get", RWRWR_, "root", SARH_ID, SEC_RD)
        {
            vector<string> list;
            valList(list);
            for(unsigned i_a=0; i_a < list.size(); i_a++)
                opt->childAdd("el")->setAttr("id", list[i_a])->
>setText(valAt(list[i_a]).at().name());
        }
        if(ctrChkNode(opt, "add", RWRWR_, "root", SARH_ID, SEC_WR)
        {
            string vid = TSYS::strEncode(opt->attr("id"), TSYS::oscdID);
            valAdd(vid); valAt(vid).at().setName(opt->text());
        }
        if(ctrChkNode(opt, "del", RWRWR_, "root", SARH_ID, SEC_WR)      valDel(opt->
>attr("id"), true);
        }
        else TModule::cntrCmdProc(opt);
    }
}

//*****
/* Повідомлення архіватора *
//*****

//*****
/* TMArchivator *
//*****
TMArchivator::TMArchivator(const string &iid, const string &idb, TElem *cf_el) :
    TConfig( cf_el ), run_st(false),
    m_id(cfg("ID").getSd()), m_name(cfg("NAME").getSd()),
m_dscr(cfg("DESCR").getSd()), m_addr(cfg("ADDR").getSd()),
    m_cat_o(cfg("CATEG").getSd()), m_start(cfg("START").getBd()),
m_level(cfg("LEVEL").getId()), m_db(idb)
{
    m_id = iid;
}

TCntrNode &TMArchivator::operator=( TCntrNode &node )
{
    TMArchivator *src_n = dynamic_cast<TMArchivator*>(&node);
    if( !src_n ) return *this;

    //> Конфігурація копіювання
    string tid = id();
    *(TConfig*)this = *(TConfig*)src_n;
    cfg("MODUL").setS(owner().modId());
    m_id = tid;
    m_db = src_n->m_db;
}

```

```

    if( src_n->startStat() && toStart() && !startStat() )
        start( );

    return *this;
}

void TMArchivator::postEnable( int flag )
{
    cfg("MODUL").setS(owner().modId());
}

void TMArchivator::preDisable( int flag )
{
    if( startStat() ) stop( );
}

void TMArchivator::postDisable(int flag)
{
    try
    {
        if( flag )
            SYS->db().at().dataDel(fullDB(),SYS-
>archive().at().nodePath()+tbl(),*this,true);
    }catch(TError err)
    { mess_warning(err.cat.c_str(),"%s",err.mess.c_str()); }
}

TTipArchivator &TMArchivator::owner( )    { return *(TTipArchivator*)nodePrev(); }

string TMArchivator::workId( )            { return owner().modId()+"."+id(); }

string TMArchivator::name( )              { return (m_name.size())?m_name:m_id; }

string TMArchivator::tbl( )               { return
owner().owner().subId()+"_mess_proc"; }

void TMArchivator::load_( )
{
    if( !SYS->chkSelDB(DB()) ) return;
    SYS->db().at().dataGet(fullDB(),SYS->archive().at().nodePath()+tbl(),*this);
}

void TMArchivator::save_( )
{
    SYS->db().at().dataSet(fullDB(),SYS->archive().at().nodePath()+tbl(),*this);
}

void TMArchivator::categ( vector<string> &list )
{
    list.clear();
    string c_vl;
    for( int i_off = 0; (c_vl=TSYS::strSepParse(m_cat_o,0,',',&i_off)).size(); )
        list.push_back(c_vl);
}

bool TMArchivator::chkMessOK( const string &icateg, TMess::Type ilvl )
{
    vector<string> cat_ls;

    categ(cat_ls);

    if(ilvl >= level())
        for(unsigned i_cat = 0; i_cat < cat_ls.size(); i_cat++)
            if(TRegExp(cat_ls[i_cat], "p").test(icateg))
                return true;
    return false;
}

```

```

TVariant TMArchivator::objFuncCall( const string &iid, vector<TVariant> &prms,
const string &user )
{
    // bool status() - беремо статус архівування.
    if(iid == "status") return startStat();
    // int end() - беремо дані архівування, час закінчення.
    if(iid == "end") return (int)end();
    // int begin() - беремо дані архівування, час початку.
    if(iid == "begin") return (int)begin();

    //> Виклик функцій конфігурації
    TVariant cfRez = objFunc(iid, prms, user);
    if(!cfRez.isNull()) return cfRez;

    return TCntrNode::objFuncCall(iid, prms, user);
}

void TMArchivator::cntrCmdProc( XMLNode *opt )
{
    //> Беремо сторінку інформації
    if(opt->name() == "info")
    {
        TCntrNode::cntrCmdProc(opt);
        ctrMkNode("IoT_cntr",opt,-1,"/","_("Повідомлення архіватора:
")+name(),RWRWR_,"root",SARH_ID);
        if(ctrMkNode("area",opt,-1,"/prm","_("Архіватор")))
        {
            if(ctrMkNode("area",opt,-1,"/prm/st","_("Стан")))
            {
                ctrMkNode("fld",opt,-
1,"/prm/st/st","_("Running"),RWRWR_,"root",SARH_ID,1,"tp","bool");
                ctrMkNode("fld",opt,-1,"/prm/st/db","_("Архіватор
БД"),RWRWR_,"root","root",4,
"tp","str","dest","select","select","/db/list","help",TMess::labDB());
                ctrMkNode("fld",opt,-
1,"/prm/st/end","_("End"),R_R_R_,"root","root",1,"tp","time");
                ctrMkNode("fld",opt,-
1,"/prm/st/beg","_("Begin"),R_R_R_,"root","root",1,"tp","time");
            }
            if(ctrMkNode("area",opt,-1,"/prm/cfg","_("Конфігурація")))
            {
                ctrMkNode("fld",opt,-
1,"/prm/cfg/id",cfg("ID").fld().descr(),R_R_R_,"root",SARH_ID,1,"tp","str");
                ctrMkNode("fld",opt,-
1,"/prm/cfg/nm",cfg("NAME").fld().descr(),RWRWR_,"root",SARH_ID,2,"tp","str","le
n","50");
                ctrMkNode("fld",opt,-
1,"/prm/cfg/dscr",cfg("DESCR").fld().descr(),RWRWR_,"root",SARH_ID,3,"tp","str",
"cols","90","rows","3");
                ctrMkNode("fld",opt,-
1,"/prm/cfg/addr",cfg("ADDR").fld().descr(),RWRWR_,"root",SARH_ID,1,"tp","str");
                ctrMkNode("fld",opt,-
1,"/prm/cfg/lvl",cfg("LEVEL").fld().descr(),RWRWR_,"root",SARH_ID,2,"tp","dec",
"help","_("Отримуємо повідомлення для рівня більшого і
дорівнюючого цьому."));
                ctrMkNode("fld",opt,-
1,"/prm/cfg/cats",cfg("CATEG").fld().descr(),RWRWR_,"root",SARH_ID,2,"tp","str",
"help","_("Шаблон категорії повідомлень або регулярне вираження у
процесі архівування, відокремлюються символом';'.\n"
"Використовуйте тимчасові символи для групового
виділення:\n '*' - будь-які підрядки;\n '?' - будь-які символи.\n"
"Регулярне вираження складається з символів'/')
(/mod_(System|LogicLev)/)."));
                ctrMkNode("fld",opt,-1,"/prm/cfg/start","_("To
start"),RWRWR_,"root",SARH_ID,1,"tp","bool");
            }
        }
    }
}

```

```

        if(run_st && ctrMkNode("area",opt,-
1, "/mess",_("Messages"),R_R___,"root",SARH_ID)
        {
            ctrMkNode("fld",opt,-
1, "/mess/tm",_("Time"),RWRW___,"root",SARH_ID,1,"tp","time");
            ctrMkNode("fld",opt,-1, "/mess/size",_("Розмір
(s)"),RWRW___,"root",SARH_ID,1,"tp","dec");
            ctrMkNode("fld",opt,-1, "/mess/cat",_("Зразок
катеропії"),RWRW___,"root",SARH_ID,2,"tp","str","help",
            _("Шаблон катеропії повідомлень або регулярне вираження.\n"
            "Використовуйте тимчасові символи для групового виділення:\n
'*' - будь-які підрядки;\n '?' - будь-які символи.\n"
            "Регулярне вираження складається з символів'/
(/mod_(System|LogicLev)/)."));
            ctrMkNode("fld",opt,-
1, "/mess/lvl",_("Level"),RWRW___,"root",SARH_ID,4,"tp","dec","min","0","max","7",
            "help",_("Отримуємо повідомлення для рівня більшого і дорівнюючого
цьому."));
            if(ctrMkNode("table",opt,-
1, "/mess/mess",_("Messages"),R_R___,"root",SARH_ID)
            {
                ctrMkNode("list",opt,-
1, "/mess/mess/0",_("Час"),R_R___,"root",SARH_ID,1,"tp","time");
                ctrMkNode("list",opt,-
1, "/mess/mess/0a",_("mcsec"),R_R___,"root",SARH_ID,1,"tp","dec");
                ctrMkNode("list",opt,-
1, "/mess/mess/1",_("Катеропія"),R_R___,"root",SARH_ID,1,"tp","str");
                ctrMkNode("list",opt,-
1, "/mess/mess/2",_("Рівень"),R_R___,"root",SARH_ID,1,"tp","dec");
                ctrMkNode("list",opt,-
1, "/mess/mess/3",_("Повідомлення"),R_R___,"root",SARH_ID,1,"tp","str");
            }
            return;
        }
        //> Процес управління на сторінці
        string a_path = opt->attr("path");
        if(a_path == "/prm/st/st")
        {
            if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)
startStat() ? "1" : "0" );
            if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR)
atoi(opt-
>text().c_str()) ? start() : stop());
        }
        else if(a_path == "/prm/st/db")
        {
            if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)
opt->setText(
DB() );
            if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR)
setDB( opt-
>text() );
        }
        else if(a_path == "/prm/st/end" && ctrChkNode(opt)
opt->setText(
TSYS::int2str(end()) );
        else if(a_path == "/prm/st/beg" && ctrChkNode(opt)
opt->setText(
TSYS::int2str(begin()) );
        else if(a_path == "/prm/cfg/id" && ctrChkNode(opt)
opt->setText(
id() );
        else if(a_path == "/prm/cfg/nm" )
        {
            if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)
opt->setText(
name() );
            if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR)
setName( opt-
>text() );
        }
        else if(a_path == "/prm/cfg/dscr")
        {
            if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD)
opt->setText(
dscr() );

```

```

        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))      setDscr( opt-
>text() );
    }
    else if(a_path == "/prm/cfg/addr")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))      opt->setText(
addr() );
        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))      setAddr( opt-
>text() );
    }
    else if(a_path == "/prm/cfg/lvl")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))      opt-
>setText(TSYS::int2str(level()));
        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))      setLevel(
atoi(opt->text().c_str()));
    }
    else if(a_path == "/prm/cfg/start")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))      opt->setText(
toStart() ? "1" : "0" );
        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))      setToStart(
atoi(opt->text().c_str()));
    }
    else if(a_path == "/prm/cfg/cats")
    {
        if(ctrChkNode(opt,"get",RWRWR_,"root",SARH_ID,SEC_RD))      opt-
>setText(m_cat_o);
        if(ctrChkNode(opt,"set",RWRWR_,"root",SARH_ID,SEC_WR))      { m_cat_o =
opt->text(); modif(); }
    }
    else if(a_path == "/mess/tm")
    {
        if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))
        {
            opt->setText(TBDS::genDBGet(nodePath()+"messTm","0",opt-
>attr("user")));
            if(!atoi(opt->text().c_str())) opt-
>setText(TSYS::int2str(time(NULL)));
        }
        if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
            TBDS::genDBSet(nodePath()+"messTm", (atoi(opt-
>text().c_str())>time(NULL))?"0":opt->text(),opt->attr("user"));
    }
    else if(a_path == "/mess/size")
    {
        if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))      opt-
>setText(TBDS::genDBGet(nodePath()+"messSize","10",opt->attr("user")));
        if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
            TBDS::genDBSet(nodePath()+"messSize",opt->text(),opt->attr("user"));
    }
    else if(a_path == "/mess/cat")
    {
        if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))      opt-
>setText(TBDS::genDBGet(nodePath()+"messCat","",opt->attr("user")));
        if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
            TBDS::genDBSet(nodePath()+"messCat",opt->text(),opt->attr("user"));
    }
    else if(a_path == "/mess/lvl")
    {
        if(ctrChkNode(opt,"get",RWRW__, "root", SARH_ID, SEC_RD))      opt-
>setText(TBDS::genDBGet(nodePath()+"messLev","0",opt->attr("user")));
        if(ctrChkNode(opt,"set",RWRW__, "root", SARH_ID, SEC_WR))
            TBDS::genDBSet(nodePath()+"messLev",opt->text(),opt->attr("user"));
    }
    else if(a_path == "/mess/mess" && run_st &&
ctrChkNode(opt,"get",R_R____,"root",SARH_ID))
    {
        vector<TMess::SRec> rec;

```

```

        time_t end = atoi(TBDS::genDBGet(nodePath()+"messTm","0",opt-
>attr("user")).c_str());
        if( !end ) end = time(NULL);
        time_t beg = end - atoi(TBDS::genDBGet(nodePath()+"messSize","10",opt-
>attr("user")).c_str());
        string cat = TBDS::genDBGet(nodePath()+"messCat","",opt->attr("user"));
        char lev = atoi(TBDS::genDBGet(nodePath()+"messLev","0",opt-
>attr("user")).c_str());

        get( beg, end, rec, cat, lev );

        XMLNode *n_tm      = ctrMkNode("list",opt,-
1, "/mess/mess/0","",R_R____,"root",SARH_ID);
        XMLNode *n_tmu     = ctrMkNode("list",opt,-
1, "/mess/mess/0a","",R_R____,"root",SARH_ID);
        XMLNode *n_cat     = ctrMkNode("list",opt,-
1, "/mess/mess/1","",R_R____,"root",SARH_ID);
        XMLNode *n_lvl     = ctrMkNode("list",opt,-
1, "/mess/mess/2","",R_R____,"root",SARH_ID);
        XMLNode *n_mess    = ctrMkNode("list",opt,-
1, "/mess/mess/3","",R_R____,"root",SARH_ID);
        for(int i_rec = rec.size()-1; i_rec >= 0; i_rec--)
        {
            if(n_tm)      n_tm->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].time));
            if(n_tmu)     n_tmu->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].utime));
            if(n_cat)     n_cat->childAdd("el")->setText(rec[i_rec].categ);
            if(n_lvl)     n_lvl->childAdd("el")-
>setText(TSYS::int2str(rec[i_rec].level));
            if(n_mess)    n_mess->childAdd("el")->setText(rec[i_rec].mess);
        }
    }
    else TCntrNode::cntrCmdProc(opt);
}

```

## tmodule.cpp - робота з модулями

```

//Інтелектуальна система підтримки життєвого циклу з використанням IoT системний
файл: tmodule.cpp

#include <sys/types.h>
#include <sys/stat.h>
#include <stdarg.h>
#include <unistd.h>
#include <dlfcn.h>
#include <string.h>
#include <libintl.h>

#include "tsys.h"
#include "terror.h"
#include "tmess.h"
#include "tsubsys.h"
#include "tmodule.h"

using namespace IoT;

//*****
/* TModule
//*****
const char *TModule::l_info[] =
    {"Модуль", "Ім'я", "Тип", "Джерело", "Версія", "Автор", "Дескриптор", "Ліцензія"};

TModule::TModule( const string &id ) : mId(id)
{
    lc_id = string("oscd_")+mId;
    bindtextdomain(lc_id.c_str(), LOCALEDIR);

    //> Переведення динамічного рядка
#ifdef 0
    char mess[][100] = { _("Автор"), _("Ліцензія") };
#endif
}

TModule::~TModule( )
{
    //> Очищення списку експортуємих функцій
    for(unsigned i = 0; i < m_efunc.size(); i++)
        delete m_efunc[i];
}

string TModule::modName()
{
    return mName;
}

void TModule::postEnable( int flag )
{
    if(flag&TCntrNode::NodeRestore) return;

    mess_info(nodePath().c_str(), _("З'єднання з модулем!"));
}

TSubSYS &TModule::owner( )    { return *(TSubSYS*)nodePrev(); }

void TModule::modFuncList( vector<string> &list )
{
    list.clear();
    for(unsigned i = 0; i < m_efunc.size(); i++)
        list.push_back(m_efunc[i]->prot);
}

bool TModule::modFuncPresent( const string &prot )
{

```

```

    for(unsigned i = 0; i < m_efunc.size(); i++)
        if(m_efunc[i]->prot == prot)
            return true;
    return false;
}

TModule::ExpFunc &TModule::modFunc( const string &prot )
{
    for(unsigned i = 0; i < m_efunc.size(); i++)
        if(m_efunc[i]->prot == prot) return *m_efunc[i];
    throw TError(nodePath().c_str(),_("Функція '%s' не представлена у
модулі!"),prot.c_str());
}

void TModule::modFunc( const string &prot, void (TModule::*offptr)() )
{
    *offptr = modFunc(prot).ptr;
}

void TModule::modInfo( vector<string> &list )
{
    for(unsigned i_opt = 0; i_opt < sizeof(l_info)/sizeof(char *); i_opt++)
        list.push_back(l_info[i_opt]);
}

string TModule::modInfo( const string &name )
{
    string info;

    if(name == l_info[0]) info = mId;
    else if(name == l_info[1]) info = mName;
    else if(name == l_info[2]) info = mType;
    else if(name == l_info[3]) info = mSource;
    else if(name == l_info[4]) info = mVers;
    else if(name == l_info[5]) info = mAuthor;
    else if(name == l_info[6]) info = mDescr;
    else if(name == l_info[7]) info = mLicense;

    return info;
}

void TModule::cntrCmdProc( XMLNode *opt )
{
    //> Беремо сторінку інформації
    if(opt->name() == "info")
    {
        TCntrNode::cntrCmdProc(opt);
        ctrMkNode("IoT_cntr",opt,-1,"/","Module: "+modId(),R_R_R_);
        ctrMkNode("branches",opt,-1,"/br","","R_R_R_");
        if(TUIS::icoPresent(owner().subId()+"."+modId())) ctrMkNode("img",opt,-
1,"/ico","","R_R_R_");
        if(ctrMkNode("area",opt,-1,"/help",_("Help")))
            if(ctrMkNode("area",opt,-1,"/help/m_inf",_("Модуль інформації")))
            {
                vector<string> list;
                modInfo(list);
                for(unsigned i_l = 0; i_l < list.size(); i_l++)
                    ctrMkNode("fld",opt,-
1,(string("/help/m_inf/"+list[i_l]).c_str(),_(list[i_l].c_str()),R_R_R_,"root",
"root",1,"tp","str");
            }
        return;
    }

    //> Процес управління на сторінці
    string a_path = opt->attr("path");
    if(a_path == "/ico" && ctrChkNode(opt))
    {
        string itp;

```

```
    opt-
>setText(TSYS::strEncode(TUIS::icoGet(owner().subId()+".")+modId(),&itp),TSYS::ba
se64));
    opt->setAttr("tp",itp);
    }
    else if(a_path.substr(0,11) == "/help/m_inf" && ctrChkNode(opt))
        opt->setText(modInfo(TSYS::pathLev(a_path,2)));
    else TCntrNode::cntrCmdProc(opt);
}

const char *TModule::I18N( const char *mess )
{
    const char *rez = Mess->I18N(mess,lc_id.c_str());
    if( !strcmp(mess,rez) ) rez = _(mess);
    return rez;
}
```

К6П3\_2024

## tparamcontr.cpp - параметри управління системою

```

//Інтелектуальна система підтримки життєвого циклу з використанням IoT системний
файл: tparamcontr.cpp

#include "tbds.h"
#include "tsys.h"
#include "tmess.h"
#include "tdaqs.h"
#include "tcontroller.h"
#include "ttipdaq.h"
#include "ttipparam.h"
#include "tparamcontr.h"

using namespace IoT;

//*****
//* TParamContr *
//*****
TParamContr::TParamContr( const string &name, TTipParam *tpprm ) :
TConfig(tpprm), m_en(false), tipparm(tpprm)
{
    setId(name);
    setName(name);
}

TParamContr::~TParamContr( )
{
    nodeDelAll();
}

TCntrNode &TParamContr::operator=( TCntrNode &node )
{
    TParamContr *src_n = dynamic_cast<TParamContr*>(&node);
    if(!src_n) return *this;

    //> Перевіряємо тип параметрів й змінюємо їх, якщо вони змінні, або нижчі
    if(type().name != src_n->type().name && owner().owner().tpPrmToId(src_n-
>type().name) >= 0)
    {
        if(enableStat()) disable();
        setType(src_n->type().name);
    }

    //> Конфігурація копіювання
    string tid = id();
    *(TConfig*)this = *(TConfig*)src_n;
    setId(tid);

    //> Дозволяємо нові параметри
    if(src_n->enableStat() && toEnable( ) && !enableStat())enable();

    return *this;
}

TController &TParamContr::owner( ) { return *(TController*)nodePrev(); }

string TParamContr::name( ) { string nm = cfg("NAME").getS(); return nm.size()
? nm : id(); }

void TParamContr::setName( const string &inm ) { cfg("NAME").setS(inm); }

string TParamContr::descr( ) { return cfg("DESCR").getS(); }

void TParamContr::setDescr( const string &idsc ){ cfg("DESCR").setS(idsc); }

void TParamContr::postEnable(int flag)
{

```

```

TValue::postEnable(flag);

if(!vlCfg())  setVlCfg(this);
if(!vlElemPresent(&SYS->daq().at().errE()))
    vlElemAtt(&SYS->daq().at().errE());
}

void TParamContr::preDisable(int flag)
{
    //> Видаляємо або зупиняємо архівування
    vector<string> a_ls;
    vlList(a_ls);

    for(unsigned i_a = 0; i_a < a_ls.size(); i_a++)
        if(!vlAt(a_ls[i_a]).at().arch().freeStat())
            {
                string arh_id = vlAt(a_ls[i_a]).at().arch().at().id();
                if(flag) SYS->archive().at().valDel(arh_id,true);
                else SYS->archive().at().valAt(arh_id).at().stop();
            }

    if(enableStat())  disable();
}

void TParamContr::postDisable(int flag)
{
    if(flag)
        {
            //> Видаляємо параметри з БД
            try
                {
                    SYS->db().at().dataDel(owner().DB()+"."+owner().cfg(type().db).getS(),
owner().owner().nodePath()+owner().cfg(type().db).getS(),*this,true);
                }catch(TError err) { mess_err(err.cat.c_str(),"%",err.mess.c_str()); }
            }
        }

void TParamContr::load_( )
{
    if(!SYS->chkSelDB(owner().DB())) return;

    cfgViewAll(true);
    SYS->db().at().dataGet(owner().DB()+"."+owner().cfg(type().db).getS(),
owner().owner().nodePath()+owner().cfg(type().db).getS(),*this);
}

void TParamContr::save_( )
{
    SYS->db().at().dataSet( owner().DB()+"."+owner().cfg(type().db).getS(),
owner().owner().nodePath()+owner().cfg(type().db).getS(),*this );

    //> Зберігаємо архіви
    vector<string> a_ls;
    vlList(a_ls);
    for(unsigned i_a = 0; i_a < a_ls.size(); i_a++)
        if(!vlAt(a_ls[i_a]).at().arch().freeStat())
            vlAt(a_ls[i_a]).at().arch().at().save();
}

bool TParamContr::cfgChange( TCfg &cfg ) { modif( ); return true; }

TParamContr & TParamContr::operator=( TParamContr & PrmCntr )
{
    TConfig::operator=(PrmCntr);

    return *this;
}

```

```

}

void TParamContr::enable()
{
    m_en = true;
}

void TParamContr::disable()
{
    m_en = false;
}

void TParamContr::vlGet( TVal &val )
{
    if( val.name() == "err" )
    {
        if( !enableStat() ) val.setS(_("1:Параметри заборонені."),0,true);
        else if( !owner().startStat( ) ) val.setS(_("2:Контролер
зупинено."),0,true);
        else val.setS("0",0,true);
    }
}

void TParamContr::setId( const string &vl )
{
    cfg("SHIFR").setS(vl);
}

void TParamContr::setType( const string &tpId )
{
    if(enableStat() || tpId == type().name ||
!owner().owner().tpPrmPresent(tpId)) return;

    setNodeMode(TCntrNode::Disable);

    try
    {
        //> Чекаємо поки роз'єднаються інші
        while(nodeUse(true) > 1) usleep(1000);
        //> Видаляємо з БД
        postDisable(true);

        //> Створюємо тимчасову структуру
        TConfig tCfg(&type());
        tCfg = *(TConfig*)this;

        //> Встановлюємо нову конфігурацію структури
        tpparm = &owner().owner().tpPrmAt(owner().owner().tpPrmToId(tpId));
        setElem(tpparm);

        //> Відновлюємо конфігурацію
        *(TConfig*)this = tCfg;
    }catch(...) { }
    setNodeMode(TCntrNode::Enable);
    setVlCfg(this);
    modif();
}

TVariant TParamContr::objFuncCall( const string &iid, vector<TVariant> &prms,
const string &user )
{
    //> Виклик функцій конфігурації
    TVariant cfRez = objFunc(iid, prms, user);
    if(!cfRez.isNull()) return cfRez;
    return TValue::objFuncCall(iid, prms, user);
}

void TParamContr::cntrCmdProc( XMLNode *opt )
{
    string a_path = opt->attr("path");
    //> Сервіс управління процесами

```

```

if(a_path.substr(0,6) == "/serv/") { TValue::cntrCmdProc(opt); return; }
//> Беремо сторінку інформації
if(opt->name() == "info")
{
    TValue::cntrCmdProc(opt);
    ctrMkNode("IoT_cntr",opt,-1,"/","_("Параметр:
")+name(),RWRWR_,"root",SDAQ_ID);
    if(ctrMkNode("area",opt,0,"/prm","_("Parameter")))
    {
        if(ctrMkNode("area",opt,-1,"/prm/st","_("Стан")))
        {
            if(!enableStat() && owner().owner().tpPrmSize() > 1)
                ctrMkNode("fld",opt,-
1,"/prm/st/type","_("Тип"),RWRWR_,"root",SDAQ_ID,4,"tp","str","dest","select","se
lect","/prm/tpLst",
                "help","_("Змінюємо тип керівництва до останніх даних для
специфічних конфігурацій."));
            else ctrMkNode("fld",opt,-
1,"/prm/st/type","_("Тип"),R_R_R_,"root",SDAQ_ID,1,"tp","str");
            if(owner().enableStat())
                ctrMkNode("fld",opt,-
1,"/prm/st/en","_("Дозволено"),RWRWR_,"root",SDAQ_ID,1,"tp","bool");
        }
        if(ctrMkNode("area",opt,-1,"/prm/cfg","_("Configuration")))
            TConfig::cntrCmdMake(opt,"/prm/cfg",0,"root",SDAQ_ID,RWRWR_);
    }
    return;
}
//> Процес управління на сторінці
if(a_path == "/prm/st/type")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SDAQ_ID,SEC_RD)) opt-
>setText(type().name);
    if(ctrChkNode(opt,"set",RWRWR_,"root",SDAQ_ID,SEC_WR)) setType(opt-
>text());
}
else if(a_path == "/prm/st/en")
{
    if(ctrChkNode(opt,"get",RWRWR_,"root",SDAQ_ID,SEC_RD)) opt-
>setText(enableStat()?"1":"0");
    if(ctrChkNode(opt,"set",RWRWR_,"root",SDAQ_ID,SEC_WR))
    {
        if(!owner().enableStat()) throw TError(nodePath().c_str(),"_("Контролер
не запустився!"));
        else atoi(opt->text().c_str())?enable():disable();
    }
}
else if(a_path.substr(0,8) == "/prm/cfg")
TConfig::cntrCmdProc(opt,TSYS::pathLev(a_path,2),"root",SDAQ_ID,RWRWR_);
else if(a_path == "/prm/templList" && ctrChkNode(opt))
{
    vector<string> lls, ls;
    SYS->daq().at().templLibList(lls);
    for(unsigned i_l = 0; i_l < lls.size(); i_l++)
    {
        SYS->daq().at().templLibAt(lls[i_l]).at().list(ls);
        for(unsigned i_t = 0; i_t < ls.size(); i_t++)
            opt->childAdd("el")->setText(lls[i_l]+"."+ls[i_t]);
    }
}
else if(a_path == "/prm/tpLst" && ctrChkNode(opt))
    for(unsigned i_tp = 0; i_tp < owner().owner().tpPrmSize(); i_tp++)
        opt->childAdd("el")->setAttr("id",owner().owner().tpPrmAt(i_tp).name)-
>setText(owner().owner().tpPrmAt(i_tp).descr);
    else TValue::cntrCmdProc(opt);
}

```