

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут автоматики та електротехніки

Кафедра: Морського приладобудування

«Допущений до захисту»

В.о. завідувача кафедри



Гайдай Г.Ю.

«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «магістр»

**на тему: «Інформаційно-вимірювальна система моніторингу технічного стану
двигунів у виробничих умовах»**

Виконав: студент групи 6631м



Гайдаєнко В.А.

(підпис)

Керівник роботи:

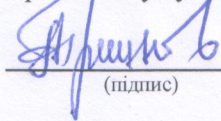
доц.

кафедри

Морського

приладобудування

Грєшинов А.Ю.



(підпис)

Грєшинов А.Ю.

Миколаїв – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут автоматики та електротехніки

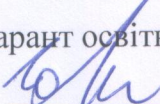
Кафедра морського приладобудування

Спеціальність 175 «Інформаційно-вимірювальні технології»

Освітня програми «Вимірювально-інформаційні системи»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми


_____ **Гордєєв Б.М.**
(підпис)

«__» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «магістр»

Студенту _____ **Гайдаєнко В'ячеславу Андрійовичу**
(Прізвище, ім'я, по батькові)

1. Тема роботи: «Інформаційно-вимірювальна система моніторингу технічного стану двигунів у виробничих умовах»

Керівник роботи _____ **Грешнов Андрій Юрійович, кандидат технічних наук, доцент**

Затверджені наказом ректора № 1217-уч від «14» _____ 10 _____ 2025 року

2. Термін подання роботи: _____ **17.12.25**

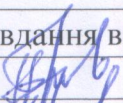
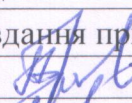
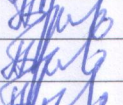
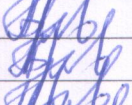
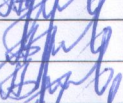
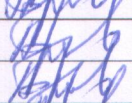
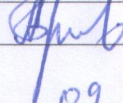
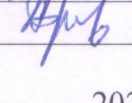
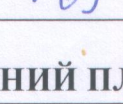
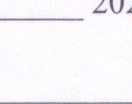
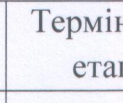
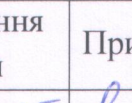
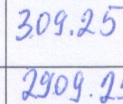
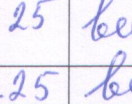
3. Вихідні дані по роботі: ДСТУ щодо обробки інформації, літературні джерела, технічна документація системи, матеріали практики.

4. Перелік питань, що належать до розробки (найменування розділів): 1. Аналіз існуючих систем моніторингу двигунів; 2. Розробка технічного завдання;

3. Розробка структурної схеми установки для визначення відхилення 4. Розробка електричної
принципової схеми; 5. Метрологічне забезпечення; 6. Техніко-економічне обґрунтування розробки;
7. Охорона праці. 8. Охорона навколишнього середовища

5. Перелік презентаційних матеріалів: 1. Концепція системи; 2. Архітектура розробки; 3.
Алгоритм локальної схеми калібрування; 4. Мнемосхема об'єкту управління; 5. Структурна схема
перетворювача; 6. Тренд запуску; 7. Екранні форми розробки. 8. Апробація дослідження; Висновки.

6. Консультанти розділів роботи

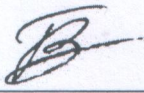
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Грешнов А.Ю., доцент		
2	Грешнов А.Ю., доцент		
3	Грешнов А.Ю., доцент		
4	Грешнов А.Ю., доцент		
5	Грешнов А.Ю., доцент		
6	Тубальцев А.М., доцент		
7	Тубальцев А.М., доцент		

7. Дата видачі завдання « 3 » 09 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз існуючих систем моніторингу двигунів	30.09.25 – 29.10.25	век.
2.	Розробка технічного завдання	29.10.25 – 20.11.25	век.
3.	Розробка структурної схеми установки для визначення відхилення двигунів	20.11.25 – 27.12.25	век.
4.	Розробка електричної принципової схеми сполучення перетворювача з ЕОМ	27.12.25 – 10.01.26	век.
5.	Метрологічне забезпечення	10.01.26 – 17.02.26	век.
6.	Охорона праці	17.02.26 – 24.03.26	век.
7.	Охорона навколишнього середовища	24.03.26 – 18.04.26	век.

Студент

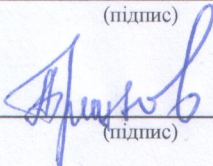


(підпис)

Гайдаєнко В.А.

(ПІБ)

Керівник роботи



(підпис)

Грешнов А.Ю.

(ПІБ)

Анотація

Робота присвячена розробці інформаційно-управляючої системи віддаленого моніторингу та діагностики двигуна. Спроектована ІС та програмне забезпечення призначене для автоматичного діагностування технічного стану двигуна в режимі реального часу, збору та збереження інформації для наступного вивчення зміни характеристик устаткування з метою підвищення їхньої надійності.

У роботі проведені дослідження процесів інформаційно управляючої системи віддаленого моніторингу та діагностики двигуна.

У роботі розроблено проект інформаційно управляючої системи віддаленого моніторингу та діагностики двигуна.

Під час проектування розглянута сучасна концепція підвищення експлуатаційних характеристик електропривідних установок з використанням інформаційно-управляючої системи віддаленого моніторингу та діагностики газотурбінних двигунів, описані структура та основні програмно-технічні компоненти центра віддаленого моніторингу, наведена схема взаємодії центра віддаленого моніторингу та системи технічної діагностики, сформульовані мета та завдання інформаційно-управляючої системи віддаленого моніторингу та діагностики газотурбінних двигунів, наведений перелік алгоритмів з викладом суті їхні роботи, показані методи відображення діагностичної інформації.

Приводяться рисунки, що ілюструють роботу програми, а також вхідні дані. Описуються результати роботи.

Програма розроблена в середовищі Python і працює під керуванням операційної системи Windows NT.

Дипломна робота викладена на 127 аркушах, містить 32 рисунків, 7 таблиць, 3 додатки. Список використаних джерел містить 15 найменувань.

Ключові слова: моніторинг стану двигунів, інформаційно-вимірювальні системи, діагностика обладнання, вібраційний аналіз, технічний стан машин, сенсорні технології, прогнозування відмов

Annotation

The thesis is devoted to the development of an information-control system for remote monitoring and diagnostics of an engine. The designed information system and software are intended for automatic real-time diagnostics of the engine's technical condition, as well as for collecting and storing data for further analysis of changes in equipment characteristics in order to increase its reliability.

The work includes research on the processes of the information-control system for remote monitoring and diagnostics of the engine. A project of an information-control system for remote monitoring and diagnostics of the engine has been developed.

During the design stage, a modern concept of improving the operational characteristics of electric drive installations using an information-control system for remote monitoring and diagnostics of gas-turbine engines was considered. The structure and main software-hardware components of the remote monitoring center are described. The interaction scheme between the remote monitoring center and the technical diagnostics system is provided. The goals and tasks of the information-control system for remote monitoring and diagnostics of gas-turbine engines are formulated. A list of algorithms is presented with an explanation of their functioning, and the methods of visualizing diagnostic information are demonstrated.

The thesis includes figures illustrating the operation of the program, as well as input data. The results of the system's operation are described.

The software was developed in the Python environment and operates under the Windows NT operating system.

The thesis is presented on 127 pages and contains 32 figures, 7 tables, and 3 appendices. The list of references includes 15 sources.

Keywords: engine condition monitoring, information-measuring systems, equipment diagnostics, vibration analysis, machine technical state, sensor technologies, failure prediction

ЗМІСТ

Основні поняття та визначення	5
Вступ	6
РОЗДІЛ 1. ДОСЛІДЖЕННЯ СИСТЕМ МОНІТОРИНГУ ТА ДІАГНОСТИКИ ЕТУ	9
1.1. Опис існуючих систем моніторингу технічного стану двигунів	9
1.2 MitsubishiRMS (Система віддаленого моніторингу)	11
1.3 Siemens. OnlineMonitoringofGasTurbinePowerPlants.....	13
1.4 General Electric. iCenter and Remote Monitoring & Diagnostics (RM&D).	14
1.5 Призначення та задачі модуля інформаційно-управляючої системи моніторингу	16
1.6 Оперативний тренд-аналіз і прогнозування параметрів ЕТУ	18
1.7 Техніко-економічне обґрунтування розробки інформаційно-управляючої системи моніторингу та діагностики ЕТУ	19
1.8 Концепція системи	20
1.9 Постановка задачі	25
Висновки до розділу 1.	27
РОЗДІЛ 2. ФОРМУЛЮВАННЯ ВИМОГ ТА КОНЦЕПЦІЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ ВІДДАЛЕНОГО МОНІТОРИНГУ ТА ДІАГНОСТИКИ ЕТУ	29
2.1 Технічні засоби	29
2.1.1 Вибір головного серверу даних	29
2.2 Програмні засоби	31
2.2.1 Вибір середі розробки	31
2.3 Формування вимог до інформаційно-управляючої системи моніторингу	33
2.3.1 Основні положення	33
2.3.3. Склад і призначення системи вилученого моніторингу	35
2.3.4. Очікувані результати реалізації концепції вилученого моніторингу	36
2.3.5 Напрямок розвитку системи вилученого моніторингу	36
2.4 Розробка програмного забезпечення інформаційно-управляючої системи.....	37
2.4.1 Розробка інтерфейсу	37
2.5.1 Модуль відправки даних	41
2.6 Модуль передачі даних у СУБД.....	53
2.7 Модуль управління потоком даних СТД.....	54
2.7 Модуль збору й обробки даних	63
2.8 Розробка модулю передачі даних у систему технічної діагностики	67
Висновки до розділу 2	70
РОЗДІЛ 3. РОЗРОБКА ПРОЄКТУ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ ВІДДАЛЕНОГО МОНІТОРИНГУ ТА ДІАГНОСТИКИ ЕТУ	72
3.1 Алгоритм локальної схеми калібрування інформаційно-виміральної системи	72
3.2 Установка СУБД	75
Висновки до розділу 3	81
РОЗДІЛ 4. РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ ВІД ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ МОНІТОРИНГУ	82
4.1 Загальна характеристика	82
4.2 Розрахунок вартості створення програмного засобу	83

4.3 Розрахунок економічної ефективності розробки і впровадження програмного продукту	86
Висновки до розділу 4	87
РОЗДІЛ 5. ОХОРОНА ПРАЦІ ПРИ РОБОТІ З ПК У ВІДДІЛІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	88
5.1 Загальні принципи охорони праці	88
5.3 Аналіз небезпечних і шкідливих факторів при роботі з ПК у відділі програмного забезпечення	89
5.4 Розрахунок системи загального освітлення приміщення відділу програмного забезпечення	92
5.2 Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів	94
Висновки до розділу 5	94
6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА.....	96
6.1 Вступ.....	96
6.2 Забруднення навколишнього середовища суднобудівними заводами	98
6.3 Заходи щодо зменшення забруднення	99
6.4 Аналіз ризиків ПЗ методом побудови дерева подій.....	100
Висновки до розділу 6	103
Висновки	105
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	106
Додаток А – Технічне завдання.....	108
Додаток Б - Інструкція користувача	111
Додаток В – Фрагмент програмного коду	114

ОСНОВНІ ПОНЯТТЯ ТА ВИЗНАЧЕННЯ

БД – база даних

ВМ - віддалений моніторинг

ЕТУ - електропривідні технологічні установки

ОС – операційна система

ПЗ – програмне забезпечення

ЦВМ – центр віддаленого моніторингу

САУ – система автоматизованого управління

СЕТУ - суднові газотурбінні енергетичні установки

СД – система діагностики

СЗ – система змащення

СУ – силові установки

ТЗ – технічне завдання

ТО – технічне обслуговування

ТП – температурне поле

ТС – технічний стан

ODBC – Open DataBase Connectivity – відкритий інтерфейс зв'язку з базою даних

RAID - Redundant Array of Independent Disks – матриця незалежних дискових накопичувачів з надмірністю

SQL – Select Query Language – мова запитів до бази даних

ВСТУП

Сучасні виробничі підприємства дедалі більше залежать від безперебійної роботи технологічного обладнання, значну частку якого становлять електричні та механічні двигуни різних типів. Відмова двигуна або його робота у ненормованих режимах може призвести не лише до втрати продуктивності, а й до аварійних ситуацій, значних фінансових збитків та ризиків для безпеки персоналу. Саме тому питання оперативного, точного та надійного контролю технічного стану двигунів є критично важливим у контексті сучасних інтелектуалізованих виробничих систем.

Традиційні методи діагностики, що базуються на періодичних оглядах або ручному зборі даних, не забезпечують належного рівня точності та своєчасності виявлення несправностей. У виробництвах, де обладнання працює в інтенсивних та змінних умовах, виникає потреба у впровадженні систем безперервного моніторингу, здатних у режимі реального часу збирати, обробляти та аналізувати дані для раннього виявлення аномалій. Такий підхід дозволяє переходити від реактивного обслуговування до прогностичного, що значно знижує ризик відмов і підвищує ефективність експлуатації обладнання.

Інформаційно-вимірювальні системи, побудовані на сучасних сенсорних технологіях, методах цифрової обробки сигналів та інтелектуальних алгоритмах аналізу, відкривають нові можливості для підвищення надійності та продуктивності підприємств. Їх використання забезпечує створення єдиного цифрового середовища для моніторингу та підтримки прийняття рішень щодо технічного стану обладнання.

Актуальність теми полягає у зростаючій потребі промислових підприємств у системах, що здатні забезпечити високоточний контроль стану обладнання, мінімізувати простой та підвищити рівень технічної безпеки. Розробка такої системи сприяє оптимізації виробничих процесів, зниженню витрат на ремонт і обслуговування, а також створенню умов для переходу до концепції розумного виробництва.

Об'єкт дослідження є процес моніторингу технічного стану промислових двигунів у виробничих умовах.

Предметом дослідження є методи, алгоритми та технічні засоби інформаційно-вимірювальної системи, призначеної для вимірювання, збору, оброблення та аналізу параметрів технічного стану двигунів.

Метою магістерської роботи є підвищення ефективності моніторингу технічного стану двигуна за рахунок розробки інформаційно-вимірювальної системи моніторингу технічного стану двигунів, яка забезпечує збір, вимірювання, обробку та інтелектуальний аналіз параметрів роботи у виробничих умовах. У роботі досліджуються сучасні методи сенсорного вимірювання, моделі прогнозування відмов, структури побудови IoT-орієнтованих систем та засоби інтеграції з виробничими процесами.

Наукова новизна роботи полягає у розробленні та обґрунтуванні комплексного підходу до побудови інформаційно-вимірювальної системи, що забезпечує інтеграцію багатоканальних сенсорних даних у режимі реального часу. Застосування адаптивних алгоритмів аналізу технічного стану двигунів на основі виявлення відхилень та трендових змін. Використання методів машинного навчання для прогнозування відмов та формування ранніх попереджень. Підвищення точності та надійності оцінювання стану двигунів у складних виробничих умовах з високим рівнем шумів і зовнішніх впливів.

Завдання дослідження

1. Проаналізувати сучасні методи та технічні засоби моніторингу технічного стану промислових двигунів.
2. Визначити перелік ключових параметрів (вібрація, температура, струм, шум, тиск тощо), що характеризують технічний стан двигуна.
3. Розробити концептуальну модель інформаційно-вимірювальної системи моніторингу.
4. Спроектувати архітектуру системи з урахуванням вимог реального часу та виробничих обмежень.
5. Обґрунтувати вибір сенсорів та засобів збору даних, включно з частотою дискретизації, точністю та протоколами передавання.
6. Розробити алгоритми оброблення даних, включаючи фільтрацію, нормалізацію, виділення ознак та виявлення аномалій.

7. Створити математичну модель аналізу технічного стану двигуна із використанням елементів машинного навчання або методів діагностики.
8. Реалізувати програмно-апаратний прототип або модель системи.
9. Провести експериментальні дослідження, оцінити точність, швидкодію та достовірність роботи системи.
10. Розробити рекомендації щодо впровадження системи у виробниче середовище.

Таким чином, обрана тема є важливою як з точки зору наукового дослідження, так і з практичного боку, оскільки дозволяє вирішити актуальні проблеми промислових підприємств, пов'язані з діагностикою та підтримкою працездатності двигунів у режимі реального часу.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ СИСТЕМ МОНІТОРИНГУ ТА ДІАГНОСТИКИ ЕТУ

1.1. Опис існуючих систем моніторингу технічного стану двигунів

Сучасні виробничі підприємства активно впроваджують автоматизовані системи моніторингу технічного стану двигунів, оскільки від їхньої надійності та стабільності роботи значною мірою залежить ефективність технологічних процесів. Такі системи виконують функції безперервного вимірювання параметрів роботи двигуна, аналізу одержаних даних та діагностування потенційних відхилень у реальному часі. На ринку представлено широкий спектр рішень, що відрізняються функціональністю, архітектурою, рівнем автоматизації та вартістю впровадження [1].

Одним із найпоширеніших типів є вібраційні системи моніторингу, що базуються на аналізі амплітудно-частотних характеристик вібрацій. Такі системи застосовуються для виявлення дисбалансу, осьового зміщення, зносу підшипників і нерівномірності механічних вузлів. Комерційні рішення, зокрема SKF IMx та PRÜFTECHNIK Online Condition Monitoring Systems, забезпечують багатоканальний збір даних, автоматичну діагностику та передачу інформації на віддалені сервери [2].

Іншим класом є температурні та термографічні системи, що здійснюють моніторинг нагріву елементів двигуна. Вони використовуються для раннього виявлення перегрівів, локальних дефектів, порушень теплового балансу та нештатних режимів роботи. Системи цього типу уміють інтегруватися зі SCADA-платформами для створення централізованих панелей контролю [3].

Електричні системи діагностики базуються на аналізі струмових та напругових характеристик — спектрального аналізу струму, виявлення гармонічних спотворень, фіксації стрибків навантаження та визначення потенційних дефектів обмоток, коротких замикань чи зміни фазових параметрів. До поширених систем цього класу належать Motor Current Signature Analysis (MCSA) та комплексні рішення Fluke [4].

Попри значні переваги, існуючі системи мають низку обмежень таких як високу вартість, складність інтеграції з наявним обладнанням, потребу у

спеціалізованих каналах зв'язку та недосконалість моделей прогнозування в умовах нестабільних виробничих середовищ. Також типовою проблемою є недостатня адаптивність під конкретні типи двигунів та необхідність експертного налаштування діагностичних параметрів [5].

Таким чином, аналіз існуючих рішень свідчить про необхідність створення інформаційно-вимірювальної системи з покращеною функціональністю, здатною адаптуватися до особливостей виробничих умов, підтримувати інтелектуальний аналіз параметрів та забезпечувати високу точність оцінювання технічного стану двигунів у режимі реального часу [6].

1. SCADA-система (Supervisory Control And Data Acquisition)

SCADA — це комплексна система для збору, обробки та візуалізації даних із датчиків, контролерів і виробничого обладнання. Вона дозволяє операторам у режимі реального часу контролювати процеси та приймати рішення.

Моніторинг параметрів температура, тиск, швидкість конвеєра, рівень Вріх у томатній пасті.

Віддалений контроль обладнання запуск, зупинка, регулювання.

Аварійна сигналізація система повідомляє про перевищення критичних параметрів.

Збір історичних даних і формування звітів.

Переваги для виробництва томатної пасти полягають в тому що є змога точно контролювати етапи випарювання і згущення пасти.

Оптимізує роботу електродвигунів і насосів, щоб уникнути перевантаження.

Забезпечує простежуваність якості traceability.

Predictive Maintenance System система предиктивного обслуговування на базі IoT + Big Data + AI

Система передбачає несправності обладнання на основі аналізу даних з датчиків (вібрація, температура, шум, струм). Використовує методи машинного навчання для прогнозу відмов.

Постійний моніторинг технічного стану електродвигунів, насосів, редукторів.

Виявлення аномалій за допомогою алгоритмів AI.

Прогноз терміну служби вузлів і агрегатів.

Планування ремонту й технічного обслуговування без зупинки виробництва.

Переваги для виробництва томатної пасти полягає в зменшенні ризику аварій та незапланованих простоїв.

Економія витрат на ремонт та енергоносії.

Підвищення надійності лінії випарювання та пакування.

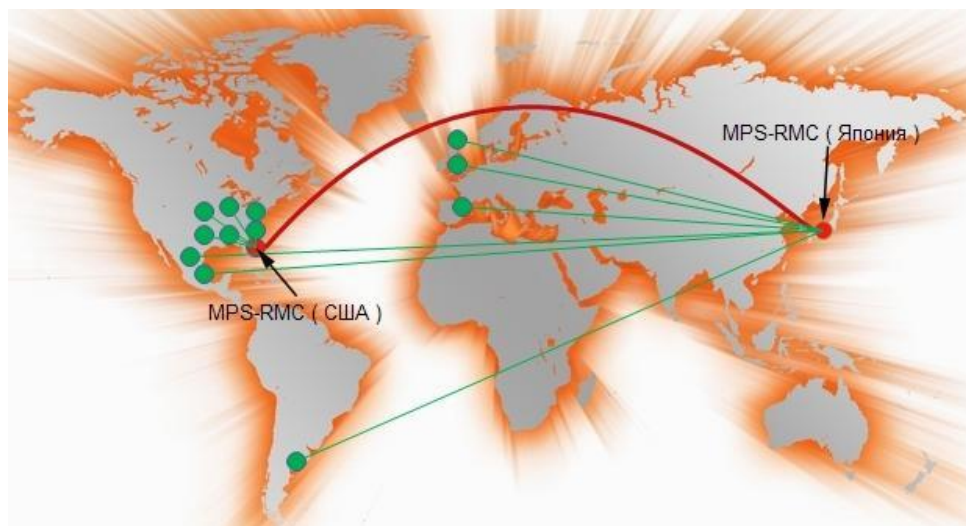
Плавне керування обладнанням для підтримки стабільної якості продукції [9].

1.2 MitsubishiRMS (Система віддаленого моніторингу)

MitsubishiPowerSystems пропонує контроль у режимі реального часу, та передбачення потенційних проблем за допомогою двох центрів, розташованих в Орландо (Флорида) та Такасаго (Японія).

Збір усіх даних з кожного пункту прямує до центру кожну секунду або три, та зберігається не менш як 2 роки, що робить цю систему дуже динамічною для виявлення будь яких проблем.

На малюнку 1.1 зображена мережа системи віддаленого моніторингу MitsubishiRMS:



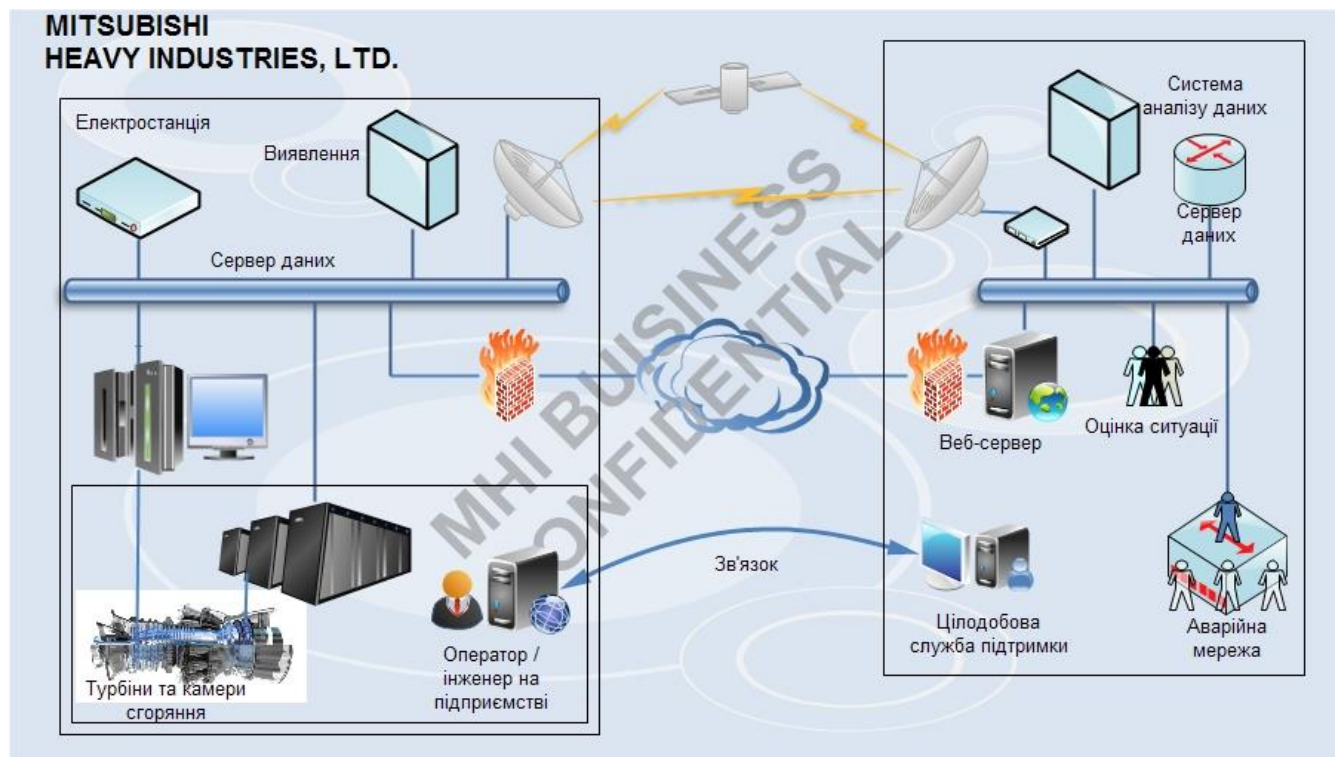
Малюнок 1.1 – зображення MitsubishiRMS

Стислий перелік можливостей:

- Підтримка клієнтів
 - ✓ моніторинг роботи, аналіз даних за історією розвитку процесів і графіків;

- ✓ управління високотемпературними частинами та планування технічного обслуговування;
- ✓ усунення неполадок в аварійних ситуаціях;
- Проводиться моніторинг наступних параметрів:
 - ✓ швидкості та потужності генератора;
 - ✓ сигналів управління;
 - ✓ температури на вході та виході з компресора, тиску на виході з компресору;
 - ✓ температури у тракті турбіни та викидів оксидів азоту;
 - ✓ вібрації ротору;
 - ✓ зміни тиску у камері згоряння;

На малюнку 1.2 представлена структура віддаленої системи моніторингу:



Малюнок 1.2 – Структура Mitsubishi RMS

Також компанія надає послугу сервісного обслуговування за рахунок якого мінімізуються експлуатаційні ризики та максимально збільшується коефіцієнт готовності електростанції. Пакети обслуговувань представлені на малюнку 1.3:



Малюнок 1.3 – Пакети обслуговувань

1.3 Siemens. OnlineMonitoringofGasTurbinePowerPlants

На рисунку 1.4 зображено структуру віддаленого моніторингу.

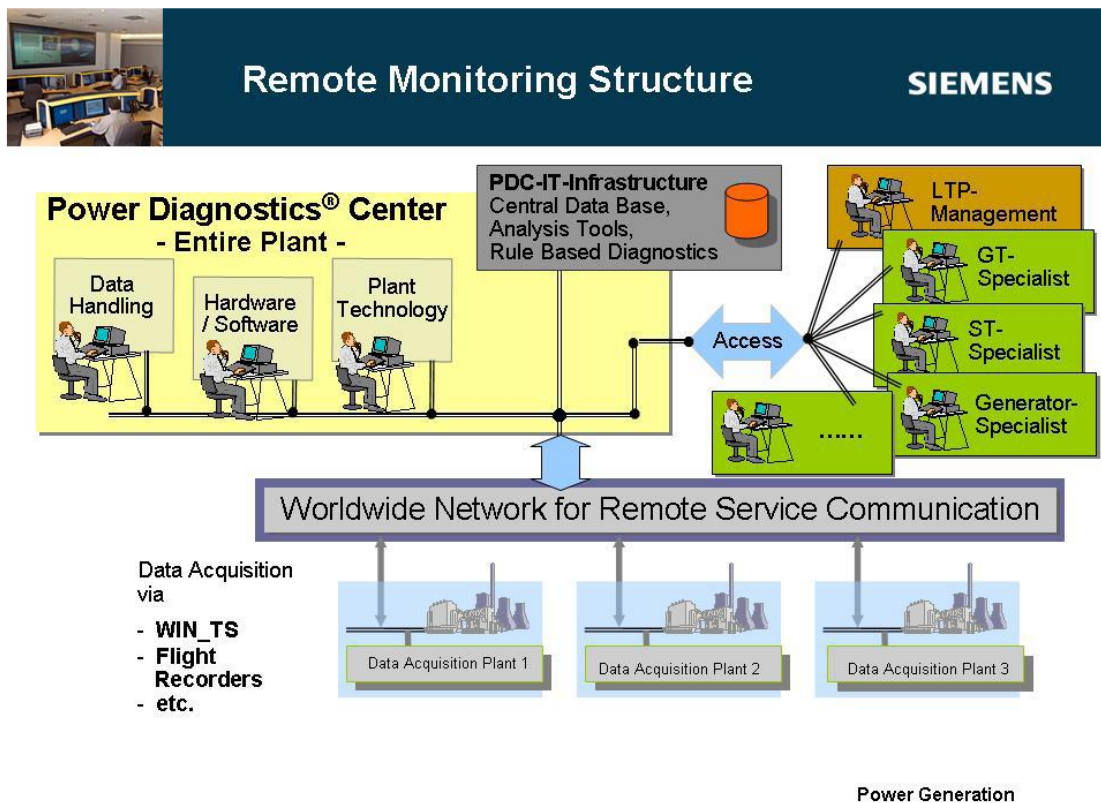


Рисунок 1.4 - OnlineMonitoringofGasTurbinePowerPlants

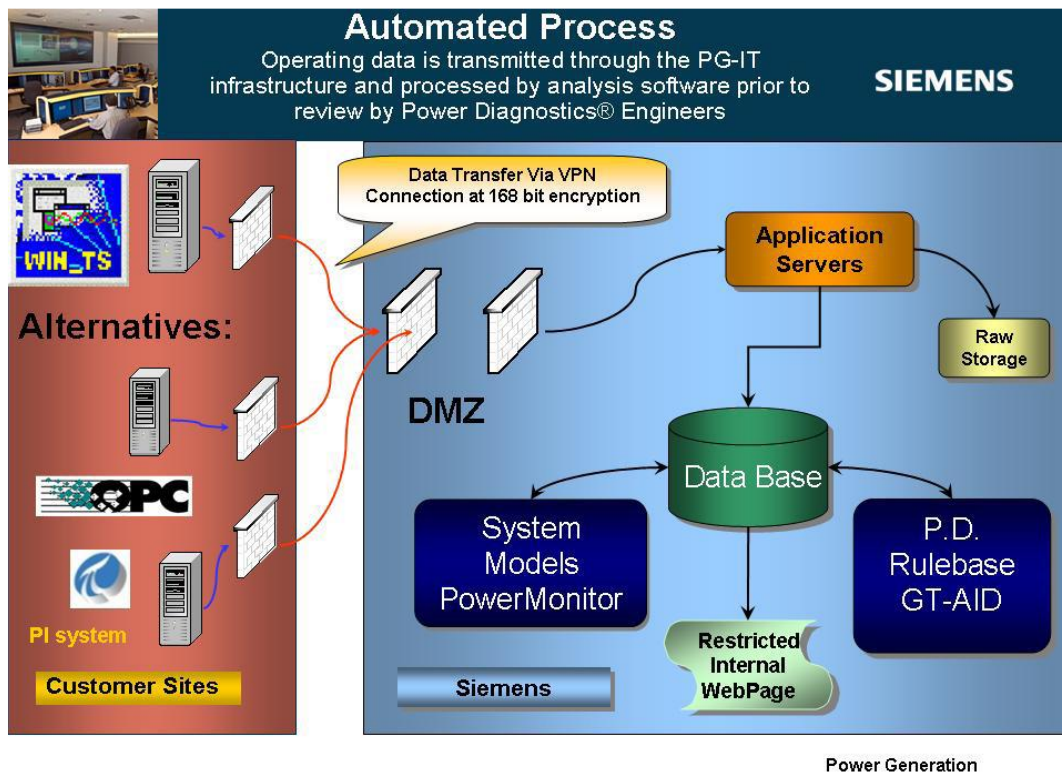


Рисунок 1.5 – Автоматизований процес моніторингу

1.4 General Electric. iCenter and Remote Monitoring & Diagnostics (RM&D).

iCenter – новітній набір послуг високих технологій, розроблених, щоб оптимізувати придатність підприємств і захисту життєвого циклу.

iCenter був розроблений так, щоб збирати інформацію з певної ділянки та відправляти її до штабу розташованого у Флорентії (FlorenceHeadquartersTeam).

Різнорічні послуги розроблені, щоб оцінювати, підтримувати та аналізувати різні дані за допомогою різнорічних моделей аналізу, додаючи цінність крізь прямий зв'язок із нафтовою та газовою командою.

BiCenter використовується програмне забезпечення, щоб забезпечити:

- ✓ віддалена оцінка стану двигунів і пошук несправностей, рисунок 1.6;
- ✓ оцінка роботи двигуна та порівняння рисунок 1.7;
- ✓ налагодження систем згорання рисунок 1.8;
- ✓ прогнозування стану ЕТУ рисунок 1.9.

Power vs Ambient Temperature

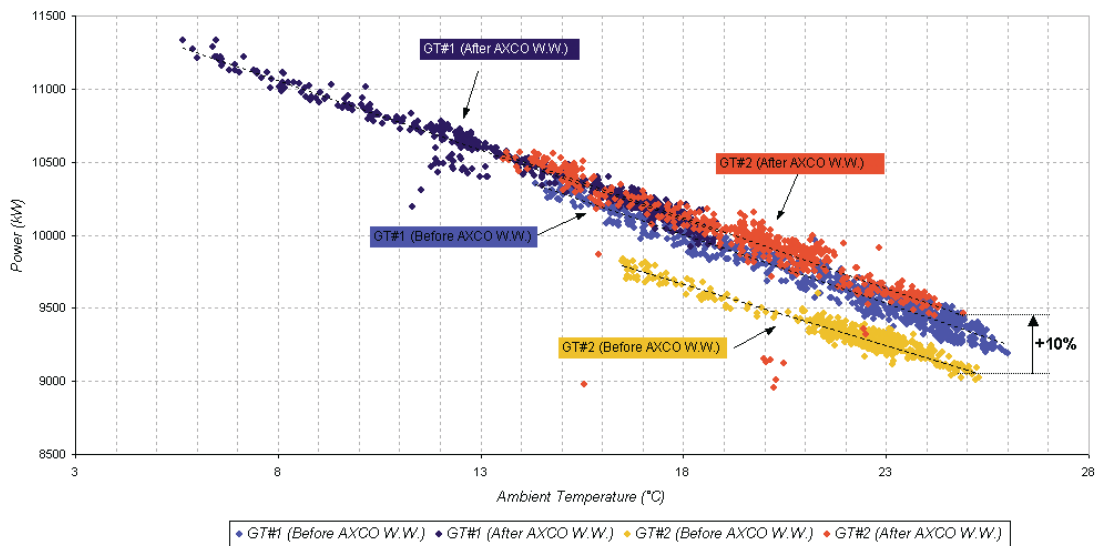


Рисунок 1.6 - Віддалена оцінка стану двигунів і пошук несправностей

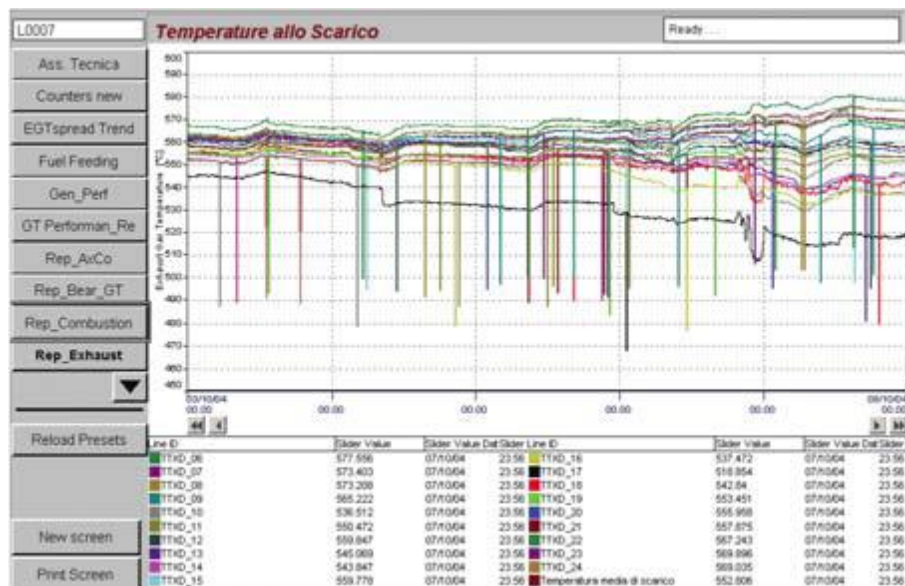


Рисунок 1.7 - Оцінка роботи двигуна та порівняння

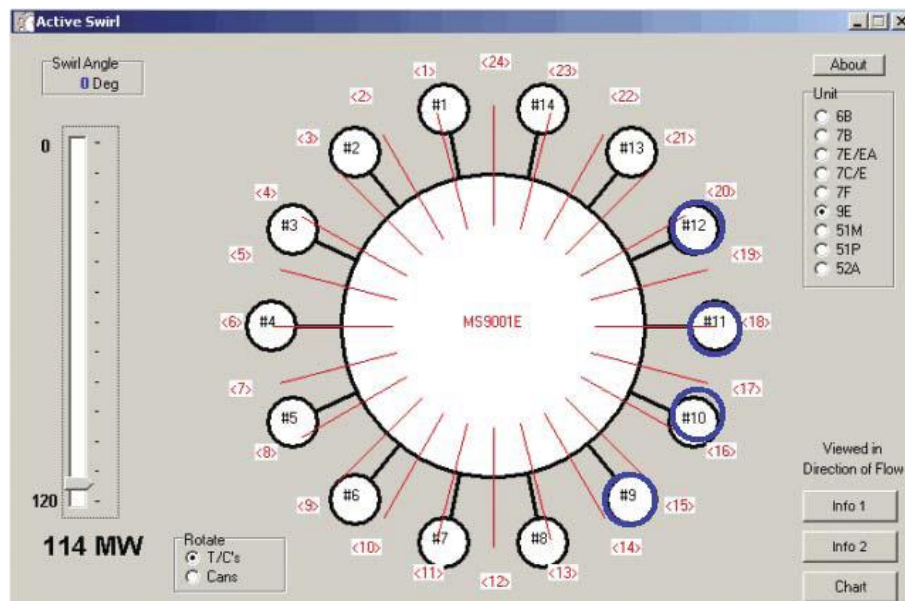


Рисунок 1.8 - Налаштування систем згорання

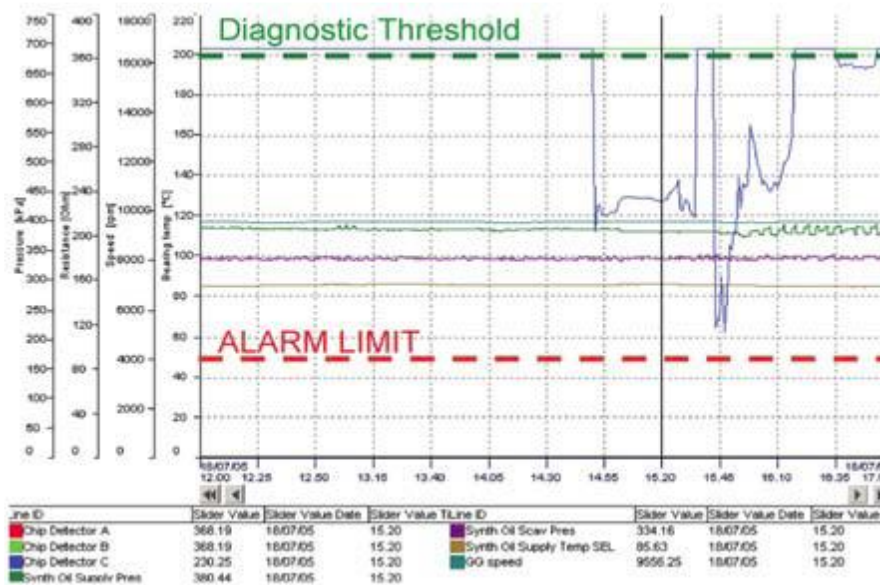


Рисунок 1.9 - Прогнозування стану ЕТУ

1.5 Призначення та задачі модуля інформаційно-управляючої системи моніторингу

Система як розробляється з технічної (параметричної) діагностики - СТД, призначена для постійного безперервного контролю технічного стану (ТС) електропривідних установок будь-якого типу й схеми по параметрах, реєструючих САУ, а також для виявлення несправностей двигуна і його підсистем на ранній стадії розвитку, з метою недопущення виникнення вторинних дефектів, прогнозування ТС двигуна, підтримки оператора при прийнятті рішень по проведенню заходів технічного обслуговування.

Особливістю даної системи є те, що вона інваріантна стосовно типу діагностуючого устаткування (енергетичної установки). Система є що набудовується й легко адаптується стосовно до індивідуальних особливостей конкретного об'єкта.

Зазначені функції реалізовані завдяки застосуванню одного з методів штучного інтелекту - універсального математичного апарата штучних обсмектувати мереж (ИНС), на базі якого побудована більша частина алгоритмів СТД. ИНС реалізують новітні досягнення інформаційних комп'ютерних технологій і успішно застосовуються при рішенні широкого класу науково-технічних задач, таких, наприклад, як апроксимація багатомірних нелінійних функцій, прогнозування в часі процесів, що залежать від багатьох змінних, класифікація за

багатьма ознаками, що забезпечує розбивку вхідного простору на області, засіб розпізнавання образів і ін.

Програмне забезпечення СТД складається з наступних основних підпрограм і блоків:

Програма, що здійснює зв'язок і одержання параметрів із САУ.

Програма збору, первинної обробки й архівації параметрів. Містить у собі:

- перевірку вхідних параметрів на вірогідність;
- виявлення стаціонарності режиму;
- розрахунок не регистрируемых параметрів;
- запис вступників параметрів у базу даних СТД.

Приватна програма визначення технічного стану ЕТУ і його систем.

Складається з підпрограм визначення ТС ЕТУ по:

- термогазодинамическим параметрам (ТГП);
- температурному полю;
- параметрам системи змащення;
- параметрам паливної системи;
- тиску в розвантажувальній порожнині;
- параметрам ЕТУ на пуску;
- тривалості вибігу роторів.

Екранний інтерфейс оператора, виконує функції візуалізації результатів по:

- ідентифікації моделі нормального стану встаткування;
- тренд - аналізу розвитку технічного стану встаткування;
- прогнозу технічного стану встаткування;
- розпізнаванню несправностей, а також надає доступ до електронної версії інструкції для експлуатації.

Програма забезпечення інтерфейсу вилученого моніторингу.

Система параметричної діагностики може бути доповнена модулем вібраційної діагностики для забезпечення можливості більше повного діагностичного аналізу експлуатованого встаткування.

1.6 Оперативний тренд-аналіз і прогнозування параметрів ЕТУ

Алгоритм аналізує відхилення від індивідуальних норм параметрів ЕТУ і у випадку виявлення тренда робить прогнозування.

Вхідні параметри.

Такими є всі параметри, з наведених у додатку 4, для яких заповнені поля в стовпцях «ТРЕНД» і «ПРОГНОЗ»

Таблиця 1.1

Повідомлення про виявлений тренд

Позначення	Розмірність	Значення	Найменування
300401	година	20	Інтервал навчання
300402	-	30	Мінімальна кількість аналізованих значень
300403	година	30	Інтервал прогнозу
300404	година	48	Граничний час до виходу прогнозованого параметра за допуск

Вихідні параметри алгоритму є ознаки повідомлень про виявлений тренд відповідного параметра (див. Таблицю 1.1) і про прогнозований вихід за допуск діагностичного параметра (див. Таблицю 1.1).

Умови входу в алгоритм.

Алгоритм вступає в роботу при наявності ознаки реєстрації годинного зрізу.

Опис алгоритму.

Для кожного з контрольованих параметрів алгоритм аналізує вибірку з $n=300401$ останніх значень X_i , $i = 1, \dots, n$, зареєстрованих у базі даних вартових зрізів. Для виявлення тренда (тобто статистично значимої зміни параметра) застосований непараметричний r -критерій Хальда-Аббе, що базується на аналізі наступного відносного параметра:

$$r = \frac{\sum_{i=2}^n (X_i - X_{i-1})^2}{\sum_{i=1}^n (X_i - \bar{X})^2};$$

чисельник цього відношення є сумою квадратів послідовних приростів параметрів, а знаменник - сумою квадратів відхилень від середнього значення для даної послідовності. Початкове значення критерію для аналізу визначається після нагромадження 300402 перших значень. Якщо накопичено менше значень, аналіз

не виконується. Розраховане значення рівняється із критичним для прийнятого рівня значимості (алгоритм настроєний на рівень значимості 0.9) і, якщо розраховане значення менше критичного, то приймається гіпотеза про наявність тренда й формується ознака відповідної ситуації.

Примітка: зазначені вище параметри надходять у даний алгоритм не безпосередньо з інших алгоритмів ППД, а з бази даних варткових зрізів.

У випадку виявлення тренда на інтервалі прогнозування, рівному C00403 інтервалам реєстрації в базу даних, визначаються рівновіддалені моменти часу t_i і для кожного з них обчислюються значення прогнозованого параметра $Y_{i\text{ пр}} = a + b \cdot t_i$. Коефіцієнти визначаються методом найменших квадратів по n вихідних значеннях параметра. Значення ширини довірчого інтервалу прогнозу $\delta Y_{i\text{ ін}}$, що відповідає довірчої ймовірності 0.95, обчислюється по формулі:

$$\delta Y_{i\text{ пр}} = 1.95 \cdot \sigma \cdot \sqrt{1 - \frac{(Y_{i\text{ пр}} - \bar{Y})^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}};$$

де σ - середньоквадратична помилка прогнозу, обчислена для відхилень від лінійного тренда по формулі

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - X_{i\text{ пр}})^2}{n - 1}};$$

Далі для кожного аналізованого параметра визначається час виходу прогнозного значення за допуск: $t = \frac{Y_{\text{дп}} - \bar{a}}{b}$. Якщо час менше C00404 годин, формується ознака відповідної ситуації.

Частота виконання, що рекомендується, алгоритму - 0.1 Гц.

1.7 Техніко-економічне обґрунтування розробки інформаційно-управляючої системи моніторингу та діагностики ЕТУ

Використання СД дозволяє організувати автоматизоване діагностичне обслуговування ЕТУ у складі газоперекачиваючих агрегатів (ГПА) для наступного переходу на проведення ТЕ по стані, а не по строках. Підвищити економічність і контролепригодність газотурбінної техніки, що як перебуває в експлуатації так і знову створюваної. Підвищити ефективність використання установки ГПА й зниження експлуатаційних витрат, у т.ч. зниження витрат на паливо у зв'язку з неухильним ростом цін на енергоносії. Підвищити зручність і ефективності роботи

обслуговуючого персоналу ЕТУ. Збільшити надійність двигунів. Підвищити ефективність технологічного процесу перекачування газу. Збільшити ступінь готовності установки.

Використання системи ВМ надає замовникові системи ВМ наступні додаткові переваги в експлуатації в порівнянні із традиційним підходом:

- послуги експертів по уточненню й верифікації результатів діагностичного контролю ЕТУ, пошуку несправностей, наданню консультацій по конструктивних особливостях двигуна, по досвіду експлуатації різних ЕТУ і т.д.
- підтримка з боку підприємства-виготовлювача в прийнятті рішень відносно оптимального часу проведення заходів щодо ТЕ й ремонту;
- можливість покласти функції оператора СД на співробітників ЦУМ-А;
- постійний безперервний контроль технічного стану ЕТУ;
- виявлення дефектів на ранній стадії розвитку до виникнення вторинних дефектів і своєчасного проведення заходів щодо їхнього усунення;
- збільшення надійності двигунів і підвищення ступеня готовності ЕТУ;
- можливість проведення ТЕ по стані, а не по строках;
- зниження експлуатаційних витрат за рахунок оптимізації простоїв і своєчасного проведення заходів щодо ТЕ (наприклад, промивань, що дозволяє значно знизити витрати на паливо);
- зручний у роботі контекстно-контекстно-залежний електронний варіант інструкції для експлуатації.
- видача замовникові більше обміркованих рекомендацій для прийняття рішень про режими роботи установки, проведенні ТЕ, ремонтах або планових оглядах, внаслідок кращого розуміння підприємством-виготовлювачем того, що відбувається із двигуном в експлуатації.
- удосконалювання й доведення як експлуатованих, так і випуску знову ЕТУ;

перепризначення ресурсу підприємством-виготовлювачем на підставі інформації про параметри й ТС ЕТУ прихожої з експлуатації в ЦУМ і ін.

1.8 Концепція системи

Електрична принципова схема інформаційно-вимірювальної системи моніторингу роботи двигуна складається з кількох функціональних блоків,

об'єднаних у єдину систему збору, обробки та передавання параметрів технічного стану.

1. Сенсорний рівень

До двигуна підключено комплекс вимірювальних датчиків:

Вібраційний датчик (акселерометр)

Підключається безпосередньо до корпусу двигуна. Формує аналоговий сигнал пропорційно рівню вібрації.

Температурний датчик термопара/терморезистор

Вимірює температуру критичних зон двигуна.

Струмова шупа (датчик струму) дозволяє визначати навантаження на електродвигун та ідентифікувати перевантаження, перекози фаз або внутрішні несправності.

Вихідні сигнали датчиків надходять на блок попередньої обробки.

2. Блок аналогової обробки сигналів

Включає:

Підсилювач сигналу (інструментальний / операційний підсилювач)

Підсилює слабкі сигнали від сенсорів, забезпечує фільтрацію та нормалізацію рівня сигналу.

Аналого-цифровий перетворювач (ADC)

Перетворює аналогові дані у цифрову форму для подальшої обробки мікроконтролером.

У схему ADC можуть входити антиаліасингові фільтри, мультиплексори, схеми опорної напруги.

3. Обчислювальний модуль

Ключовий елемент — мікроконтролер, до якого підключені цифрові виходи ADC. Блок оперативної пам'яті (EEPROM / Flash / SD-карта). Бездротовий комунікаційний модуль. Інтерфейси керування панеллю оператора.

Функції мікроконтролера збір даних у реальному часі, фільтрація та попередня діагностика, контроль порогів небезпечних параметрів, архівація та буферизація даних, передача інформації на зовнішні системи.

4. Модуль зв'язку

Забезпечує передавання даних на сервер або SCADA-платформу Wi-Fi, LoRaWAN, Bluetooth, GSM/4G/NB-IoT (за потреби)

Передає поточні параметри, аварійні сповіщення, накопичену діагностичну інформацію.

5. Блок індикації та керування (панель оператора). Панель керування містить: світлодіодну або графічну індикацію, кнопки/сенсорні елементи для зміни режимів, інтерфейс відображення поточних значень, індикатори аварійного стану двигуна.

Цей блок отримує сигнали від мікроконтролера та відображає стан обладнання у зручному вигляді.

6. Джерело живлення включає стабілізатор напруги, фільтри, захист від перенапруги та короткого замикання.

Живить датчики, підсилювачі, ADC, мікроконтролер, комунікаційний модуль.

Електрична принципова схема відображає взаємодію всіх основних елементів Датчики → Підсилювач → ADC → Мікроконтролер → Пам'ять → Комунікаційний модуль → Панель оператора

Система, рисунок 1.10, призначена для високоточного вимірювання параметрів роботи двигуна, локальної діагностики, передачі даних у реальному часі, забезпечення безперервного моніторингу стану обладнання.

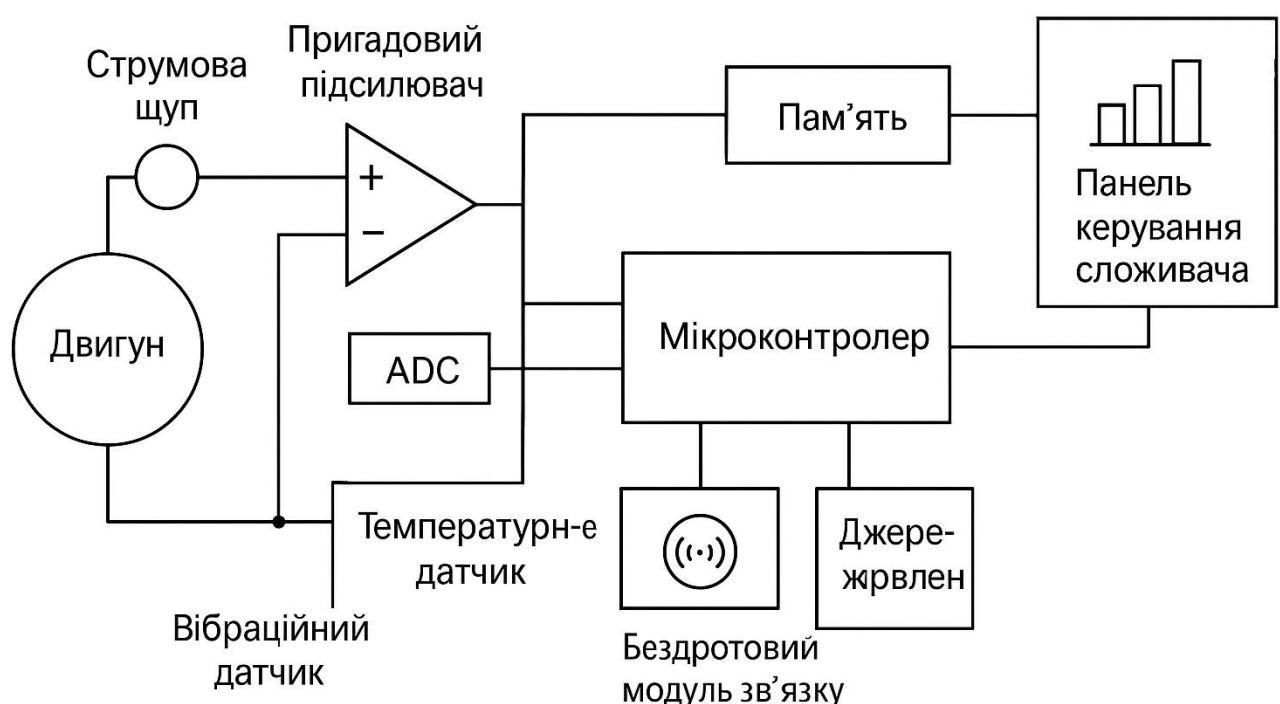


Рисунок 1.10 – Інформаційно-вимірювальна система моніторингу роботи двигуна

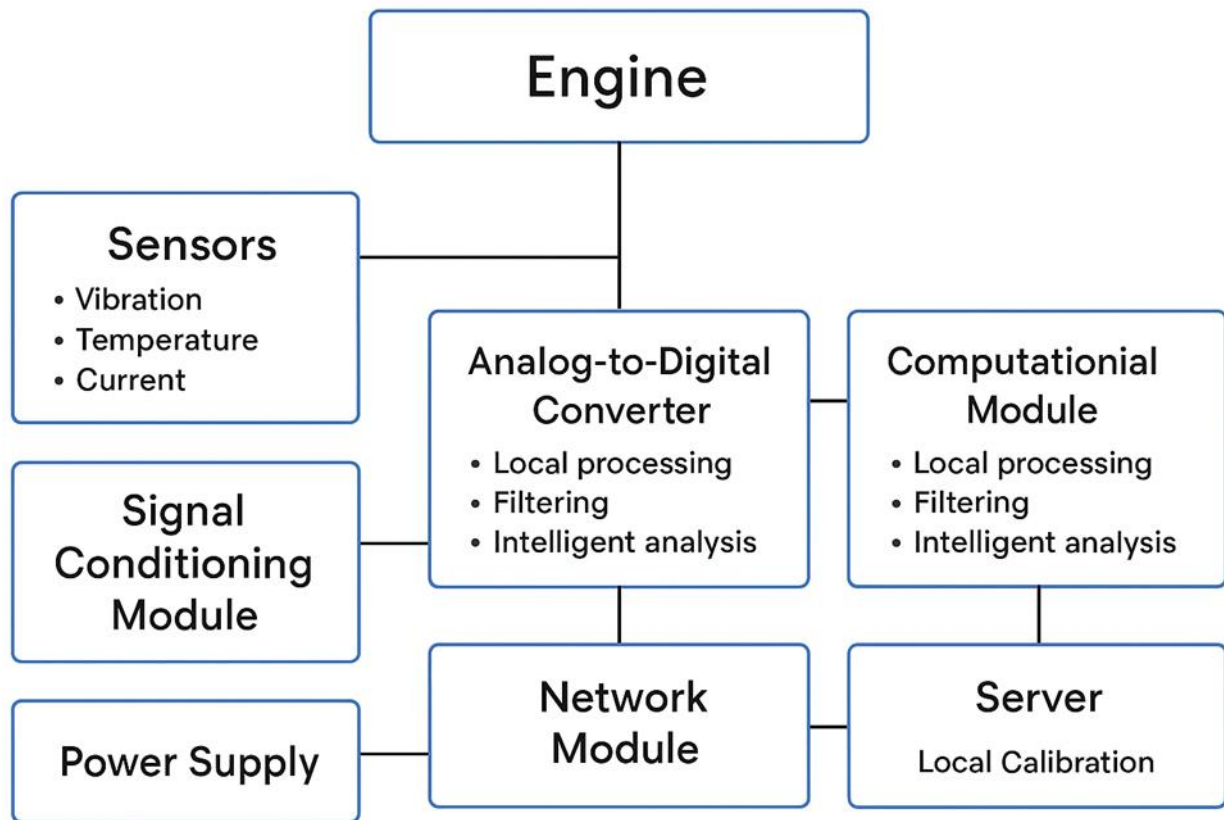


Рисунок 1.11 - Комплексна, багаторівнева вимірювальну систему з високою точністю, адаптивністю та можливістю інтеграції алгоритмів машинного навчання

Електрична принципова схема інформаційно-вимірювальної системи (ІВС), рисунок 1.11, моніторингу технічного стану двигуна складається з взаємопов'язаних апаратних модулів, кожен з яких виконує окрему функцію зі збору, перетворення, обробки та передавання діагностичних сигналів. Система реалізує адаптивну архітектуру, що забезпечує високу точність вимірювань, стійкість до промислових завад та можливість інтеграції алгоритмів машинного аналізу.

1. Первинні вимірювальні перетворювачі (сенсорний рівень)

На двигуні розміщується набір промислових датчиків.

Вібраційні акселерометри (ICP/IEPE)

– підключення до схеми здійснюється через схемні підсилювачі з константним струмом живлення (4–20 мА);

– вихідні сигнали – аналогові, у діапазоні мВ/г.

Температурні датчики (RTD Pt100/Pt1000 або термопари типу К)

– термопари вмикаються через компенсаційний модуль холодного спаю;

– RTD працюють у схемі 4-провідного підключення для забезпечення високої точності.

Датчики струму двигуна (трансформатори струму або датчики Холла)

– вихідні сигнали

– аналогові, що відповідають амплітуді навантаження.

Датчики частоти обертання (оптичні / магнітні) – формують цифрові імпульси TTL або індуктивні синусоїдальні сигнали.

Усі первинні перетворювачі з'єднані екранованими кабельними лініями для мінімізації електромагнітних завад.

2. Модуль попередньої обробки сигналів (Signal Conditioning Unit)

Сигнали від сенсорів надходять до блоку попередньої фільтрації та нормалізації. У схемі присутні операційні підсилювачі низького шуму для вібраційних каналів, фільтри нижніх та верхніх частот (LPF/HPF) для виділення інформативних компонент, аналогові активні фільтри 2-го/4-го порядку для придушення промислових перешкод (50/100 Гц), прецизійні підсилювачі термопар (типу AD8495 або MAX31855), гальванічна розв'язка (оптрони, ізоляційні DC/DC-конвертори) для захисту від перенапруг.

Цей модуль забезпечує стабільність і точність сигналів перед подачею їх у АЦП.

3. Аналого-цифрове перетворення (АЦП-модуль)

Передбачено використання багатоканального 24-розрядного АЦП (типу ADS1256 або LTC2486), підключеного через SPI-інтерфейс до контролера. Особливості в високій роздільній здатності для вібраційних сигналів, можливість синхронного дискретування декількох каналів, підтримка частоти вибірки до 50 кГц.

4. Обчислювальний модуль (контролер/мікропроцесор)

До схеми входить мікроконтролер STM32 / ESP32 / ARM Cortex-M, або одноплатний комп'ютер (Raspberry Pi / Jetson Nano) для задач машинного навчання.

Його роль зчитування цифрових потоків, виконання алгоритмів локальної фільтрації та FFT-аналізу, первинна діагностика (виявлення ударів, розбалансування, перегрівів), адаптивне налаштування чутливості сенсорних каналів, як частина концепції «самокалібрування».

Комунікаційні інтерфейси: I²C, SPI, UART, Modbus RTU/TCP, Ethernet, Wi-Fi.

5. Мережевий модуль передачі даних

Передбачено два варіанти локальна передача, RS-485 → локальний контролер або SCADA, Ethernet → сервер заводу, віддалена передача, 4G/LTE-модем, MQTT-клієнт для пересилання телеметрії, TLS-шифрування для захисту каналів

6. Серверно-аналітичний рівень

Фізично не входить до електричної схеми, але логічно пов'язаний з нею. Сервер бази даних (PostgreSQL/InfluxDB). Модуль машинного навчання (платформи TensorFlow, PyTorch). Веб-панель моніторингу та візуалізації.

Модуль аналітики, що реалізує прогнозування відмов, класифікацію аномалій, аналіз трендів.

7. Силові та захисні елементи у схемі застосовані стабілізовані джерела живлення 12/24 В, фільтри завад ЕМІ, варистори й TVS-діоди для захисту від перепадів напруги, автоматичні запобіжники на кожному сенсорному каналі, захисне заземлення всього комплексу.

8. Інтелектуальний блок локальної калібровки як окремий контур електричної схеми відповідає за автоматичне визначення дрейфу сигналів, підстроювання коефіцієнтів підсилення, перевірку цілісності сенсорних ліній, адаптацію порогів та частотних фільтрів під змінні виробничі умови.

Цей блок реалізує концепцію «самодіагностики» ІВС.

Описана електрична принципова схема являє собою комплексну, багаторівневу вимірювальну систему з високою точністю, адаптивністю та можливістю інтеграції алгоритмів машинного навчання. Вона забезпечує стабільний моніторинг технічного стану двигунів у режимі реального часу, навіть в умовах підвищених завад і складних виробничих середовищ.

1.9 Постановка задачі

У сучасних виробничих умовах забезпечення безперервної та надійної роботи двигунів є критично важливим фактором ефективності технологічних процесів. Відмова електричних, газотурбінних або поршневих двигунів може

спричинити не лише значні прості та економічні втрати, а й створити потенційні небезпеки для персоналу та обладнання. Тому актуальним є створення інформаційно-вимірювальної системи, здатної виконувати постійний моніторинг параметрів роботи двигунів, оперативно виявляти відхилення та забезпечувати діагностику технічного стану в режимі, максимально наближеному до реального часу.

Ціль роботи є розроблення інформаційно-вимірювальної системи моніторингу технічного стану двигунів, яка дозволяє автоматизувати процес контролю, зменшити вплив людського фактора та підвищити точність оцінювання технічного стану обладнання.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Проаналізувати існуючі методи, системи та технічні підходи до моніторингу стану двигунів у промисловості, визначити їхні недоліки та обмеження.
2. Визначити перелік контрольованих параметрів (вібраційні, температурні, електричні, акустичні тощо), які найбільш інформативні для оцінювання технічного стану.
3. Сформулювати вимоги до апаратного та програмного забезпечення інформаційно-вимірювальної системи.
4. Розробити архітектуру системи, яка включатиме підсистему збору даних, оброблення сигналів, аналізу технічного стану, зберігання даних і візуалізації результатів.
5. Створити концептуальну, логічну та фізичну модель бази даних, що забезпечить збереження історії вимірювань і діагностичних висновків.
6. Розробити алгоритми оброблення вимірювальної інформації, включаючи фільтрацію шумів, екстракцію ознак та оцінку стану на основі діагностичних параметрів.
7. Реалізувати програмне забезпечення, що забезпечуватиме збір, аналіз і представлення даних у зручному вигляді для оператора або інженера-діагнosta.
8. Здійснити тестування системи на модельних або реальних даних для оцінювання точності діагностики та працездатності прототипу.

9. Проаналізувати ефективність розробленої системи, визначити її переваги та можливі напрямки модернізації.

Таким чином, постановка задачі передбачає створення комплексної інформаційно-вимірювальної системи, яка здатна підвищити надійність двигунів у виробничих умовах за рахунок безперервного моніторингу, своєчасного виявлення відхилень і формування діагностичних рекомендацій.

Висновки до розділу 1.

У розділі виконано системний аналіз сучасних технологій та платформ моніторингу технічного стану двигунів, що застосовуються у промислових та енергетичних установках. Розглянуті рішення провідних світових виробників — Mitsubishi RMS, Siemens Online Monitoring, GE iCenter / RM&D — демонструють високий рівень автоматизації, використання віддаленої діагностики, цифрових двійників та прогнозової аналітики, що підтверджує актуальність переходу до інтелектуальних систем підтримки технічної експлуатації.

Проведене дослідження засвідчило, що сучасні системи моніторингу орієнтовані на безперервний збір даних, їх онлайн-аналіз, раннє виявлення ознак деградації та підтримку прийняття рішень на основі моделей машинного навчання. Разом з тим, наявні платформи характеризуються обмеженою адаптивністю до специфічних виробничих умов, високою вартістю впровадження та недостатньою інтегрованістю у локальні інформаційно-керуючі комплекси.

Визначено призначення та функціональні задачі модуля інформаційно-управляючої системи моніторингу, серед яких: оперативний тренд-аналіз, автоматизоване формування прогнозів параметрів ЕТУ, підтримка діагностичних рішень та підвищення точності оцінювання технічного стану. Техніко-економічне обґрунтування довело доцільність створення такої системи як засобу зниження експлуатаційних витрат, підвищення надійності обладнання та попередження аварійних режимів.

Сформована концепція передбачає розроблення гнучкої, масштабованої та адаптивної інформаційно-вимірювальної системи, здатної працювати в реальному часі та забезпечувати інтелектуальний аналіз даних. У постановці задачі визначено ключові напрями розробки, що включають алгоритми оброблення сигналів, методи

прогнозної діагностики, інтеграцію з локальними системами керування та створення інтерфейсу для операторів.

Отримані результати формують наукове підґрунтя для подальшої розробки та реалізації інформаційно-управляючої системи моніторингу та діагностики ЕТУ з підвищеною функціональністю та точністю.

РОЗДІЛ 2. ФОРМУЛЮВАННЯ ВИМОГ ТА КОНЦЕПЦІЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ ВІДДАЛЕНОГО МОНІТОРИНГУ ТА ДІАГНОСТИКИ ЕТУ

2.1 Технічні засоби

2.1.1 Вибір головного серверу даних

Сервер зберігання даних Flagman S1208

Основною особливістю даного сервера є дискова підсистема великої ємності, завдяки якій, він може використовуватися як універсальна альтернатива множині файл-серверів або NAS. Маючи процесор компанії Intel® з використанням технології Dual/Quad Core™ сервер здатний виконувати ефективну обробку даних, що перебувають на ньому.

Надійність забезпечується завдяки використанню серверних компонентів, пам'яті з корекцією помилок (ECC), високоякісного джерела харчування з можливістю дублювання й гарячої заміни, жорстких дисків з можливістю гарячої заміни, апаратному RAID-Контролеру, а також ефективної системи охолодження.

Сервер укомплектований дисковою підсистемою SerialATA з 8-ма дисковими відсіками для забезпечення максимальної ємності даних. Ємність дискової системи моделі S1208 може досягати 8Тб.

Сервер перукарки для установки в 19" стійку. Займана висота в стійці - 2U

Типові задачі:

Файл-Сервер

Сервер зберігання аудіо-відео інформації

Сервер відеоспостереження

Основні характеристики:двох-/чотирьох- ядерний процесор Intel® Core™ 2 / Pentium® Dual-Core EM64T

до 8Gb оперативної пам'яті DDR2 PC2-5300 ECC

дискова підсистема SATA RAID 0,1,10,5,50,6 до 8Тб

два інтегрованих мережних адаптери 10/100/1000Mbit Ethernet

система керування сервером IPMI 2.0 Server Management (опціонально)

корпус 2U Rackmount

Корпус	Chassis	2U Rackmount
Процесор	CPU	2.66GHz Intel® Core™ 2 Quad Q9450 EM64T QuadCore 1333MHz FSB, 2 x 6144Kb L2 cache
Набір мікросхем	Chipset	Intel® 3210 (Bigby-P) Server chipset
Установлена пам'ять	DIMM 1	2048Mb DDR-II ECC PC2-6400 DIMM
	DIMM 2	2048Mb DDR-II ECC PC2-6400 DIMM
	DIMM 3	(вільно)
	DIMM 4	(вільно)
Інтегровані пристрої	Мережний адаптер	Intel® 82573V PCI-E Gigabit LAN Network Interface Controller
	Мережний адаптер	Intel® 82573L PCI-E Gigabit LAN Network Interface Controller
	Відеоадаптер	ATI® ES1000 32Mb integrated PCI Graphics Controller
	Контролер SATA	Intel® ICH9R 6-channel SATA-II 300 w/integrated chipset RAID 0/1/10/5 (software) support
Слоти розширення	Slot 1 LP (PCI-X 100Mhz)	Серверний мережний адаптер 1000Мбіт Intel Server PRO/1000
	Slot 2 LP (PCI-X 100Mhz)	Серверний мережний адаптер 1000Мбіт Intel Server PRO/1000
	Slot 4 LP (PCI-E x4)	8-internal channel SAS/SATA RAID (0,1,1EE,10,5,5EE,50,6,60) controller 256Mb cache
	Slot 5 LP (PCI-X 133Mhz)	(вільно)
	Slot 6 LP (PCI-X 133Mhz)	(вільно)
	Slot 7 LP (PCI-E x8)	(вільно)
Дисковий кошик	HotSwap	8 дискових відсіків HotSwap 3.5" SerialATA/SAS
Установлені HDD	Відітнув 1	HDD 1000Gb SerialATA 7200rpm
	Відітнув 2	HDD 1000Gb SerialATA 7200rpm
	Відітнув 3	HDD 1000Gb SerialATA 7200rpm
	Відітнув 4	HDD 1000Gb SerialATA 7200rpm
	Відітнув 5	(вільно)
	Відітнув 6	(вільно)
	Відітнув 7	(вільно)
	Відітнув 8	(вільно)

Периферійні відсіки	3.5" external slim	(вільно)
	5.25" external slim	(вільно)
Джерело харчування	PSU	Відмовостійкий БП із можливістю гарячої заміни 1+1 700Вт
Система охолодження	Middle FANs	3 x 80mm HotSwap High speed FAN with Speed control
Вилучене керування	Management	Basic Management Software
Додаткове встаткування	Key/Mouse	Клавіатура 104кн + Миша оптична 2кн+скр (PS/2)
Операційна система	OS	Microsoft Windows Server Standard 2003 R2 w/SP2 64bit Russian 1-4CPU 5 Client
Додаткові ліцензії на підключення до OS	User CALs	немає
	Device CALs	немає
Сервісне обслуговування	Warranty	36 місяців безкоштовний гарантійний ремонт (перші 12 місяців з можливістю виїзду на місце установки для Москви й МО)

2.2 Програмні засоби

2.2.1 Вибір серед розробки

Delphi - це комбінація декількох найважливіших технологій:

- Високопродуктивний компілятор у машинний код
- Об'єктно-орієнтована модель компонентів
- Візуальне (а, отже, і швидкісне) побудова додатків із програмних прототипів
- Надобіддя засоби для побудови баз даних
- Компілятор у машинний код

Компілятор, убудований в Delphi, забезпечує високу продуктивність, необхідну для побудови додатків в архітектурі " клієнт-сервер". Цей компілятор у цей час є найшвидшим у світі, його швидкість компіляції становить понад 120 тисяч рядків у хвилину на комп'ютері 486DX33. Він пропонує легкість розробки й швидкий час перевірки готового програмного блоку, характерного для мов

четвертого покоління (4GL) і в той же час забезпечує якість коду, характерного для компілятора 3GL. Крім того, Delphi забезпечує швидку розробку без необхідності писати вставки на Си або ручного написання коду (хоча це можливо).

У процесі побудови додатка розроблювач вибирає з палітри компонентів готові компоненти як художник, що робить великі мазки кистю. Ще до компіляції він бачить результати своєї роботи - після підключення до джерела даних їх можна бачити відображеними на формі, можна переміщатися за даними, представляти їх у тім або іншому виді. У цьому змісті проектування в Delphi мало чим відрізняється від проектування в інтерпретуючому середовищі, однак після виконання компіляції ми одержуємо код, що виконується в 10-20 разів швидше, ніж те ж саме, зроблене за допомогою інтерпретатора. Крім того, компілятор в Delphi компіляція виробляється безпосередньо в рідний машинний код, у той час як існують компілятори, що перетворюють програму в так званий р-код, що потім інтерпретується віртуальною р-машиною. Це не може не позначитися на фактичній швидкодії готового додатка.

Об'єктно-орієнтована модель програмних компонентів

Основний упор цієї моделі в Delphi робиться на максимальному перезастосуванні коду. Це дозволяє розроблювачам будувати додатки досить швидко із заздалегідь підготовлених об'єктів, а також дає їм можливість створювати свої власні об'єкти для середовища Delphi. Ніяких обмежень по типах об'єктів, які можуть створювати розроблювачі, не існує. Дійсно, усе в Delphi написано на ньому ж, тому розроблювачі мають доступ до тих же об'єктам і інструментам, які використовувалися для створення середовища розробки. У результаті немає ніякої різниці між об'єктами, що поставляються Borland або третіми фірмами, і об'єктами, які ви можете створити.

У стандартну поставку Delphi входять основні об'єкти, які утворюють вдало підібрану ієрархію з 270 базових класів. Для початку - непогано. Але якщо виникне необхідність у рішенні якоїсь специфічної проблеми на Delphi, радимо, перш ніж спробувати починати вирішувати проблему "з нуля", переглянути список вільно розповсюджуваних або комерційних компонентів, розроблених третіми фірмами, кількість цих фірм у цей час перевищує число 250, хоча, можливо, я не про усіх знаю. Скептики, можливо, не повірять мені, коли я скажу, що на Delphi

можна однаково добре писати як додатка до корпоративних баз даних, так і, приміром, ігрові програми. Проте, це так. Багато в чому це пояснюється тим, що традиційно в середовищі Windows було досить складно реалізовувати користувальницький інтерфейс. Подійна модель в Windows завжди була складна для розуміння й налагодження. Але саме розробка інтерфейсу в Delphi є найпростішою задачею для програміста.

2.3 Формування вимог до інформаційно-управляючої системи моніторингу

Концепція віддаленого моніторингу включає ряд складових частин.

2.3.1 Основні положення

Концепція інформаційно-управляючої системи віддаленого моніторингу ЕТУ і силових установок (СУ) на їхній основі визначає основні напрямки організаційної, науково-технічної й економічної політики всіх учасників процесу розробки, виготовлення й експлуатації ЕТУ, по формуванню єдиного інформаційного простору, реалізації заходів, по створенню мережі діагностичних засобів в експлуатуючих організаціях каналів зв'язку й центрів вилученого моніторингу, створенню необхідного методичного, наукового, алгоритмічного й програмного забезпечення.

Нормативною базою концепції є «Тимчасове положення про встановлення та збільшення ресурсів і термінів служби електропривідних установок цивільної авіації, їхніх агрегатів і комплектуючих виробів», державні та галузеві нормативні документи та стандарти, нормативні документи підприємств, що беруть участь у реалізації концепції.

Кінцевою метою реалізації концепції є створення умов для експлуатації ЕТУ і СУ на їхній основі по технічному стану до виробітку фактичного ресурсу при дотриманні екологічних обмежень і за умови економічної доцільності.

Концепція відбиває погоджені позиції та цільові настанови її організацій, що реалізують.

Передумови реалізації системи інформаційно-управляючої системи віддаленого моніторингу та діагностики ЕТУ.

Обмеження та недоліки діючих інформаційно-управляючих систем моніторингу та діагностики ЕТУ.

Моніторинг технічного стану ЕТУ, реально здійснюваний у цей час, має ряд глибоких системних недоліків, зокрема:

- не ведеться історія кожного конкретного двигуна, а тільки історія конкретного дефекту;
- нормативно не визначений мінімально необхідний состав параметрів ЕТУ для рішення діагностичних завдань, інтервали дискретизації таких параметрів, вимоги до точності вимірювальних каналів;
- не визначені вимоги до повноти, составу, точності математичних моделей ЕТУ, їхньої параметричної ідентифікації для обліку індивідуальних особливостей конкретного двигуна.
- не встановлені вимоги до состава, повноті й можливостям методологічного й алгоритмічного забезпечення, зокрема, по процедурах виділення трендів параметрів ЕТУ, використання математичних моделей;
- діагностична інформація належним чином не систематизується та не доводить до розроблювачів ЕТУ для виробітку рішень по поліпшенню характеристик двигуна;
- апаратна й програмна реалізація систем моніторингу й діагностики виконується різними фірмами, найчастіше досить далекими від специфіки галузі, по окремим ТЗ, не завжди погодженими з розроблювачем двигуна.

Таким чином, по даним існуючих діагностичних систем, що перебувають в експлуатації, установити поточний реальний стан ЕТУ практично важко, або неможливо. Тому більшість ремонтів двигунів виконується у відповідності регламентом (капітальний) або після аварії. Отже, реалізація стратегії експлуатації ЕТУ по технічному стані за допомогою існуючих діагностичних засобів не представляється можливою.

Передумови до створення технологій віддаленого моніторингу ЕТУ.

- наявність і доступність сучасних інформаційних технологій збору, обробки, передачі даних практично необмеженого обсягу;
- наявність наукового, науково-технічного й методичного заділу по діагностуванню стану складних об'єктів, у тому числі ЕТУ (ЦИАМ, ХАІ, НАУ, ІПМЭ, ГП ІПКГ «Зоря-Машпроект», ВАТ «Елемент» і інші);

- наявність техніко-економічних умов, що забезпечують конкурентоспроможність інформаційних технологій і послуг ЦУМ.

2.3.3. Склад і призначення системи вилученого моніторингу

Інформаційна технологія вилученого моніторингу організована за радіальною структурою в складі ЦУМ і об'єктних станцій моніторингу й діагностики (ОСМД), зв'язаних між собою цифровими каналами зв'язку (Рис.1). Комунікаційний канал вилученого моніторингу створюється за допомогою організації виділеної лінії на базі супутникового оптоволоконного, радіо або змішаного (гібридного) каналу зв'язку (КС).

Об'єктна станція моніторингу й діагностики складається з робочої станції діагностики (РСД) і сервера параметричної діагностики (СПД), зв'язаних як із ЦУМ, так і з технічними засобами, установленими на об'єктах для рішення локальних завдань виміру, контролю параметрів і керування (САУК) кожної із силових установок.

Сервер параметричної діагностики, виконаний на ПЕВМ промислового типу, призначений для збору, первинного відпрацьовування, архівації параметрів ЕТУ. Сервер надає також мережні інтерфейси для вилученого доступу до діагностичних параметрів, результатам роботи алгоритмів, до баз даних (БД) і іншим діагностичним даним. Зв'язок з технічними засобами системи керування здійснюється по погодженому з розроблювачем САУК інтерфейсу. Робоча станція діагностики надає операторові графічний інтерфейс до системи моніторингу й діагностики й, відповідно, інтерфейс вилученого доступу до екрана оператора.

У ЦУме встановлюються вилучені термінали, кожний з яких є повноцінною робочою станцією, що шляхом вилученого підключення до мережних інтерфейсів сервера СПД дає можливість спостерігати й діагностувати встаткування із ЦУМ. Термінали ЦУМ об'єднані в мережу, ізольовану від зовнішніх мереж з метою запобігання несанкціонованого доступу.

Состав ЦУМ визначається специфікою організації, його що створює. Як функціонально необхідні підрозділи можна вказати наступні:

- технічної підтримки КС і обслуговування мережного встаткування;
- адаптації алгоритмічного й розробки програмного забезпечення системи моніторингу й діагностування;

- установки, налагодження, введення в експлуатацію й супроводи ОСМД на об'єктах;
- обробки даних моніторингу, формування архівів параметрів і баз даних.

2.3.4. Очікувані результати реалізації концепції вилученого моніторингу

Реалізація концепції вилученого моніторингу дозволить:

- об'єднати інтелектуальні, інформаційні й технічні ресурси всіх ділянок створення й експлуатації ЕТУ і силових установок на їхній основі;
- підвищити надійність і вірогідність діагностичних висновків і рекомендацій;
- забезпечити розроблювачів двигуна об'єктивною інформацією про експлуатаційні характеристики встаткування для своєчасного внесення необхідних змін;
- знизити ймовірність аварійних ситуацій, позапланових відключень і простоїв, зменшити екологічний ризик;

У цілому реалізація Концепції створює організаційну, методичну й технічну основу для реалізації стратегій експлуатації ЕТУ по технічному стані.

Системи показників і цінкових індикаторів реалізації Концепції.

Система техніко-економічних показників і цінкових індикаторів реалізації Концепції встановлюється підприємством - організатором ЦУМ.

2.3.5 Напрямок розвитку системи вилученого моніторингу

У частині апаратних засобів:

- послідовний переклад САУ запрошених ЕТУ у профіль FADEC;
- удосконалювання каналів зв'язку;
- застосування нових технологій діагностики (вихретокових, електронних, SMART – датчиків і т.п.).
- застосування сертифікованих технічних засобів.

У частині методів діагностування:

- застосування в ЦУМ експертних систем і технологій штучного інтелекту;

- удосконалювання математичних моделей діагностуємих моделей і на їхній основі методів довгострокової діагностики трендів;
- розробка ефективних методів сертифікації аномалій у фіксуємих тимчасових рядах дані реєстрації.

У частині програмного забезпечення:

- застосування спеціалізованих верифіцированих і сертифікованих програмних продуктів;
- адаптація до завдань ЦУМ універсальних програмних інструментів.

2.4 Розробка програмного забезпечення інформаційно-управляючої системи

2.4.1 Розробка інтерфейсу

Відповідно до вимог технічного завдання інтерфейс повинен у себе включати наступні компоненти візуалізації й керування:

- мнемосхема об'єкта керування;
- область виводу інформаційних повідомлень;
- область виводу інформації про контрольовані параметри;
- подання необхідної інформації у вигляді графіків (температурне поле, вібрація, пульсація);
- візуалізація інформації отриманої з модуля діагностики;

індикатори стану програми.

Тому що об'єктом керування обраний газотурбінний двигун, то мнемосхема об'єкта реалізована за допомогою наступних компонентів середовища розробки Borland Delphi:

- **TImage** відображає на формі графічне зображення. Властивість Picture типу TPicture містить відображуваний графічний об'єкт типу бітової матриці, піктограми, метафайла або певного користувачем типу. Властивість Canvas дозволяє створювати й редагувати зображення.
- **TLabel** використовується для відображення тексту, що відіграє роль мітки й не змінюється користувачем. Текст мітки задається властивістю Caption. Шрифт напису визначається властивістю Font, колір тла - властивістю Color, а колір напису - підвластивістю Color властивості Font. Розмір міток TLabel визначається властивістю AutoSize.

- **TGroupBox** контейнер з рамкою й написом, що поєднує групу зв'язаних органів керування.

Приклад мнемосхеми представлений на малюнку 2.1. На мнемосхемі відображені газотурбінний двигун і основні параметри, такі як: частота обертання компресора низького тиску, частота обертання компресора високого тиску, частота обертання силової турбіни, тиск повітря за компресором високого тиску, тиску й температура масла на вході й виході, а також положення паливорегулюючих клапанів.

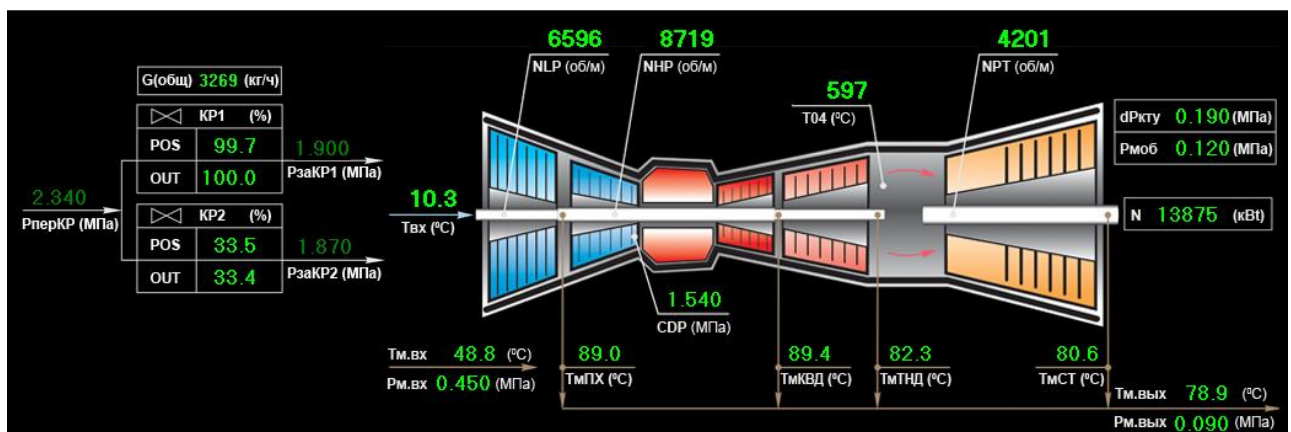


Рисунок 2.1 – Мнемосхема об'єкту управління.

Область виводу інформаційних повідомлень реалізована за допомогою компонента TListBox, що відображає список рядків і дозволяє користувачеві вибрати з нього необхідні рядки. У список автоматично додаються смуги прокручування, якщо всі рядки не містяться у вікні компонента. Область виводу інформації представлена на рисунку 2.2.

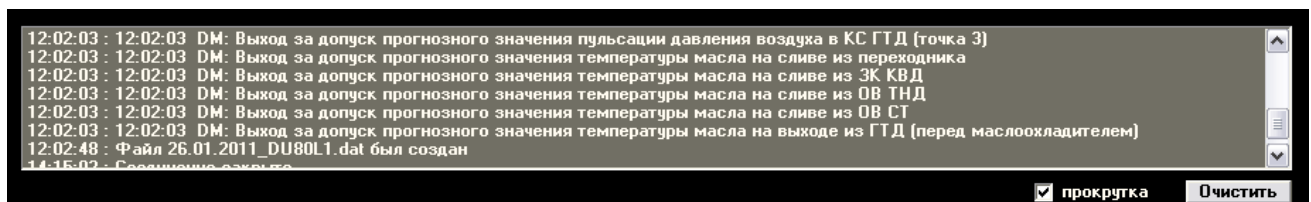


Рисунок 2.2 – Область вывода информационных сообщений

Область виводу інформації про контрольовані параметри реалізована за допомогою компонента ListView, що дозволяє відображати дані у вигляді списків,

таблиць, великих і дрібних піктограм. Область виводу інформації про контрольовані параметри представлена на рисунку 2.3.

Number	CTD	Name	Value	Description	Topic	Tag	Ftag	Gain	Offset	MinValue	MaxValue
1	1	EN110	6441	Частота вращения КНД, об/мин (точка 1)	[LC]	AppGT...	AppGT...	1	0	0	10000
2	0	EN120	6441	Частота вращения КНД, об/мин (точка 2)	[LC]	AppGT...	AppGT...	1	0	0	10000
3	2	EN210	8562	Частота вращения КВД, об/мин (точка 1)	[LC]	AppGT...	AppGT...	1	0	0	11000
4	0	EN220	8562	Частота вращения КВД, об/мин (точка 2)	[LC]	AppGT...	AppGT...	1	0	0	11000
5	3	EN310	3705	Частота вращения СТ, об/мин (точка 1)	[LC]	AppGT...	AppGT...	1	0	0	6000
6	0	EN320	3705	Частота вращения СТ, об/мин (точка 2)	[LC]	AppGT...	AppGT...	1	0	0	6000
7	4	EP010	1.396	Давление воздуха за КВД, кПа (точка 1)	[LC]	AppGT...	EP030.F	1000	0	0	2.5
8	0	EP110	1.394	Давление воздуха за КВД, кПа (точка 2)	[LC]	AppGT...	EP050.F	1000	0	0	2.5
9	5	EP030	0.223	Давление воздуха в разгрузочной полости КВД, кПа	[LC]	EP030...	EP030.F	1000	0	0	1
10	6	EP050	0.218	Давление воздуха в разгрузочной полости КНД, МПа	[LC]	EP050...	EP050.F	1000	0	0	0.4
11	7	EP120	0.279	Давление воздуха за КНД, кПа	[LC]	EP120...	EP120.F	1000	0	0	1
12	8	IP210	101	Барометрическое давление воздуха, кПа	[LC]	IP210...	IP210.F	10	0	0	140
13	9	T1_avr	11	Температура воздуха на входе в ГТД, °C (среднее)	[LC]	T1_avr...	T1_avr.F	10	-100	-100	100
14	10	T2_calc	178	Температура воздуха за КНД °C (среднее)	[LC]	T2_cal...	T2_calc.F	10	0	0	600
15	11	ET230	567	Температура газа за турбиной низкого давления, °C (точк...	[LC]	AppGT...	AppGT...	1	0	0	1300
16	12	ET240	562	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET240...	AppGT...	1	0	0	1300
17	13	ET250	577	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET250...	AppGT...	1	0	0	1300
18	14	ET260	570	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET260...	AppGT...	1	0	0	1300
19	15	ET270	566	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET270...	AppGT...	1	0	0	1300
20	16	ET280	575	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET280...	AppGT...	1	0	0	1300
21	17	ET290	574	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET290...	AppGT...	1	0	0	1300
22	18	ET300	565	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET300...	AppGT...	1	0	0	1300
23	19	ET310	573	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET310...	AppGT...	1	0	0	1300
24	20	ET320	573	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET320...	AppGT...	1	0	0	1300
25	21	ET330	570	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET330...	AppGT...	1	0	0	1300
26	22	ET340	557	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET340...	AppGT...	1	0	0	1300
27	23	ET350	563	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET350...	AppGT...	1	0	0	1300
28	24	ET360	578	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET360...	AppGT...	1	0	0	1300
29	25	ET370	552	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET370...	AppGT...	1	0	0	1300
30	26	ET380	540	Температура газа за турбиной низкого давления, °C (точк...	[LC]	ET380...	AppGT...	1	0	0	1300
31	27	T04SR	565	Температура газа за турбиной низкого давления, °C (сред...	[LC]	EGT_calc		1	0	0	1300
32	28	EB010	4.133	Вибрация корпуса ГТД (ПФ КНД), мм/с (СКЗ)	[LC]	EB010...	EB010.F	10	0	0	50
33	29	EB020	2.362	Вибрация корпуса ГТД (ЗФ переходника), мм/с (СКЗ)	[LC]	EB020...	EB020.F	10	0	0	50
34	30	EB030	1.843	Вибрация корпуса ГТД (ЗФ КС), мм/с (СКЗ)	[LC]	EB030...	EB030.F	10	0	0	50
35	31	EB040	2.841	Вибрация корпуса ГТД (ЗФ ОБ ТНД), мм/с (СКЗ)	[LC]	EB040...	EB040.F	10	0	0	50
36	32	EB050	1.651	Вибрация корпуса ГТД (ПФ ОБ СТ), мм/с (СКЗ)	[LC]	EB050...	EB050.F	10	0	0	50
37	33	EB060	1.392	Пульсация давления воздуха в камере сгорания ГТД, кПа...	[LC]	EB060...	EB060.F	10	0	0	50
38	34	EB070	1.562	Пульсация давления воздуха в камере сгорания ГТД, кПа...	[LC]	EB070...	EB070.F	10	0	0	50
39	35	EB080	1.750	Пульсация давления воздуха в камере сгорания ГТД, кПа...	[LC]	EB080...	EB080.F	10	0	0	50
40	0	EF110	-14.983	Частота тока в цепи РЧТин №1, Гц	[LC]	EF110...	EF110.F	10	0	0	60
41	0	EF210	-14.982	Частота тока в цепи РЧТин №2, Гц	[LC]	EF210...	EF210.F	10	0	0	60
42	36	FP010	1.737	Давление топливного газа за КР1, МПа	[LC]	FP010...	AppGT...	1000	0	0	4
43	37	FP020	2.346	Давление топливного газа перед КР1, МПа	[LC]	FP020...	AppGT...	1000	0	0	4
44	38	FP030	1.680	Давление топливного газа за КР2, МПа	[LC]	FP030...	AppGT...	1000	0	0	4
45	39	FP040	2.346	Давление топливного газа перед КР2, МПа	[LC]	FP040...	AppGT...	1000	0	0	4
46	0	FT010	12	Температура газа на входе в систему, °C	[LC]	FT010...	FT010.F	10	-100	-100	100
47	40	FA010	100	Положение КР1, %	[LC]	FA010...		10	0	0	100
48	41	FA020	22	Положение КР2, %	[LC]	FA020...		10	0	0	100
49	0	FA016	100	Управляющий сигнал на КР1, %	[LC]	FA016...		10	0	0	100
50	0	FA026	22	Управляющий сигнал на КР2, %	[LC]	FA026...		10	0	0	100
51	42	FQ010	-500	Расход топливного газа, кг/ч	[LC]	FQ010...	FQ010.F	1	0	0	2000
52	43	LP010	0.445	Давление масла на входе в ГТД, МПа	[LC]	LP010...	LP010.F	1000	0	0	0.065
53	44	LP020	0.090	Давление масла на выходе из ГТД, МПа	[LC]	LP020...	LP020.F	1000	0	0	0.4
54	0	LP110	0.188	Перепад давления между полостью КНД и МОБ, МПа	[LC]	LP110...	LP110.F	1000	0	0	0.25
55	45	LP130	0.093	Давление в МОБ, МПа	[LC]	LP130...	LP130.F	1000	0	0	0.4
56	46	LP200	0.530	Давление в полости КВД, МПа	[LC]	LP200...	LP200.F	1000	0	0	0.4

Рисунок 2.3 – Область вывода информации о контролируемых параметрах

У графічній формі представлена наступна інформація: показання температурного поля, значення вібрацій і пульсацій. Для виводу даної інформації використовувався компонент **TChart**, що є панеллю, на якій можна створювати діаграми й графіки різних типів. Вивід інформації у вигляді графіків представлений на рисунку 2.4.

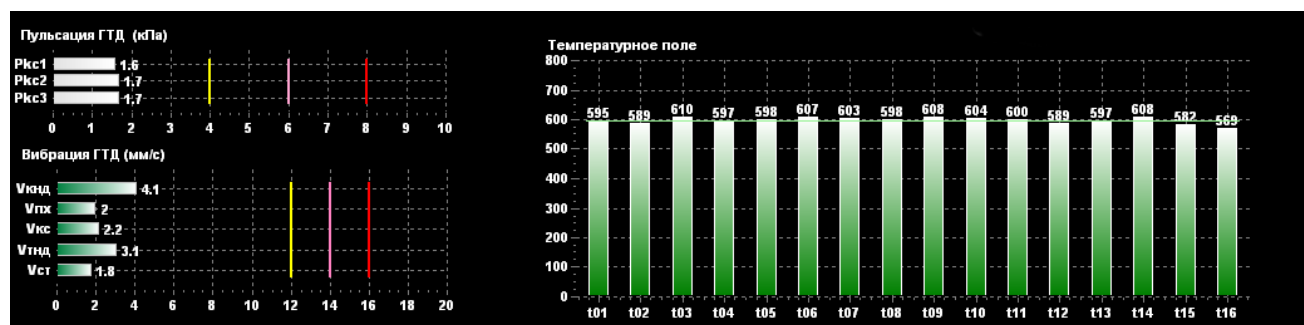


Рисунок 2.4 – График температурного поля, пульсация, вибрация.

Візуалізація інформації отриманої з модуля діагностики представлена за допомогою розробленого компонента TIndicate, що дозволяє відображати відхилення від норм, розрахункові показники, і значення транспаранта, а також компонента TChart (рис 2.5). У даному розділі реалізована гнучкість програми, тому що користувач має можливість конфігурувати кількість діагностуючих параметрів для відображення. У цьому випадку існує можливість одночасно відображати до 10 індикаторів одночасно й групувати їх у групи (до 20 груп). Реалізовано можливість видаляти як індикатори так і групи. Налаштування про конфігурацію індикаторів зберігаються в конфігураційному файлі.

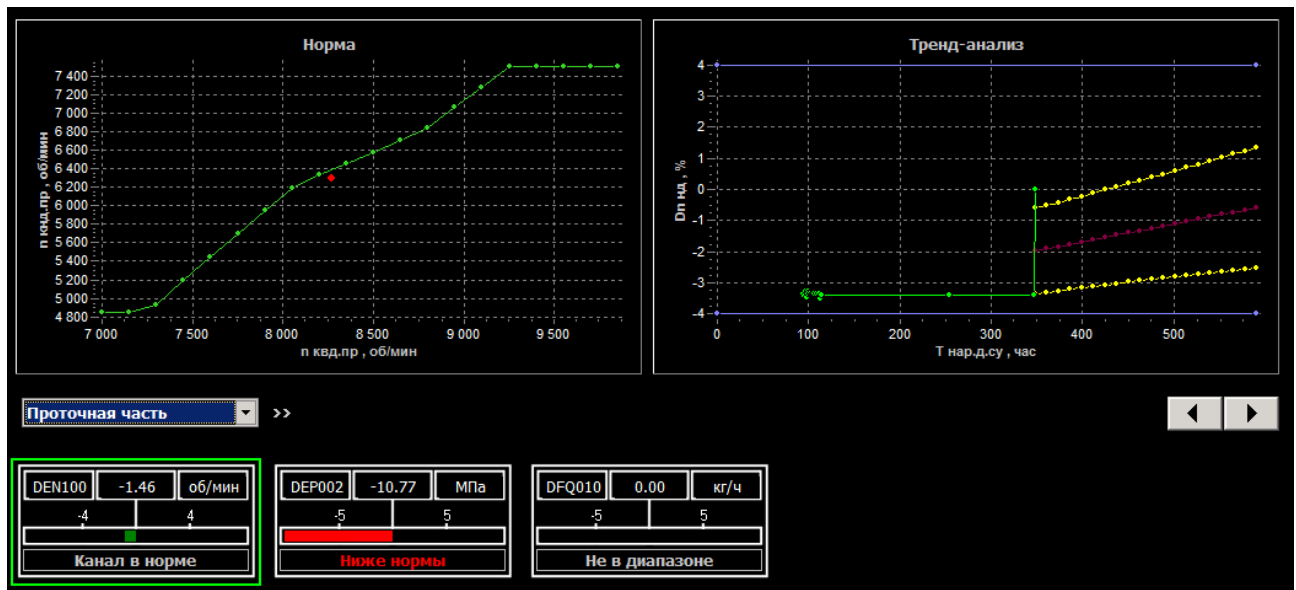


Рисунок 2.5 – Діагностична інформація.

Індикатори стану програми, реалізовані у вигляді картинок, що інформують про дії програми. Реалізовано наступні індикатори:

- читання даних з контролера;
- запис даних у файл;
- запис даних у БД;
- астросфера файлу;
- режим on-line;
- діагностика.

Загальний вид інтерфейсу представлений на рисунку 2.6.

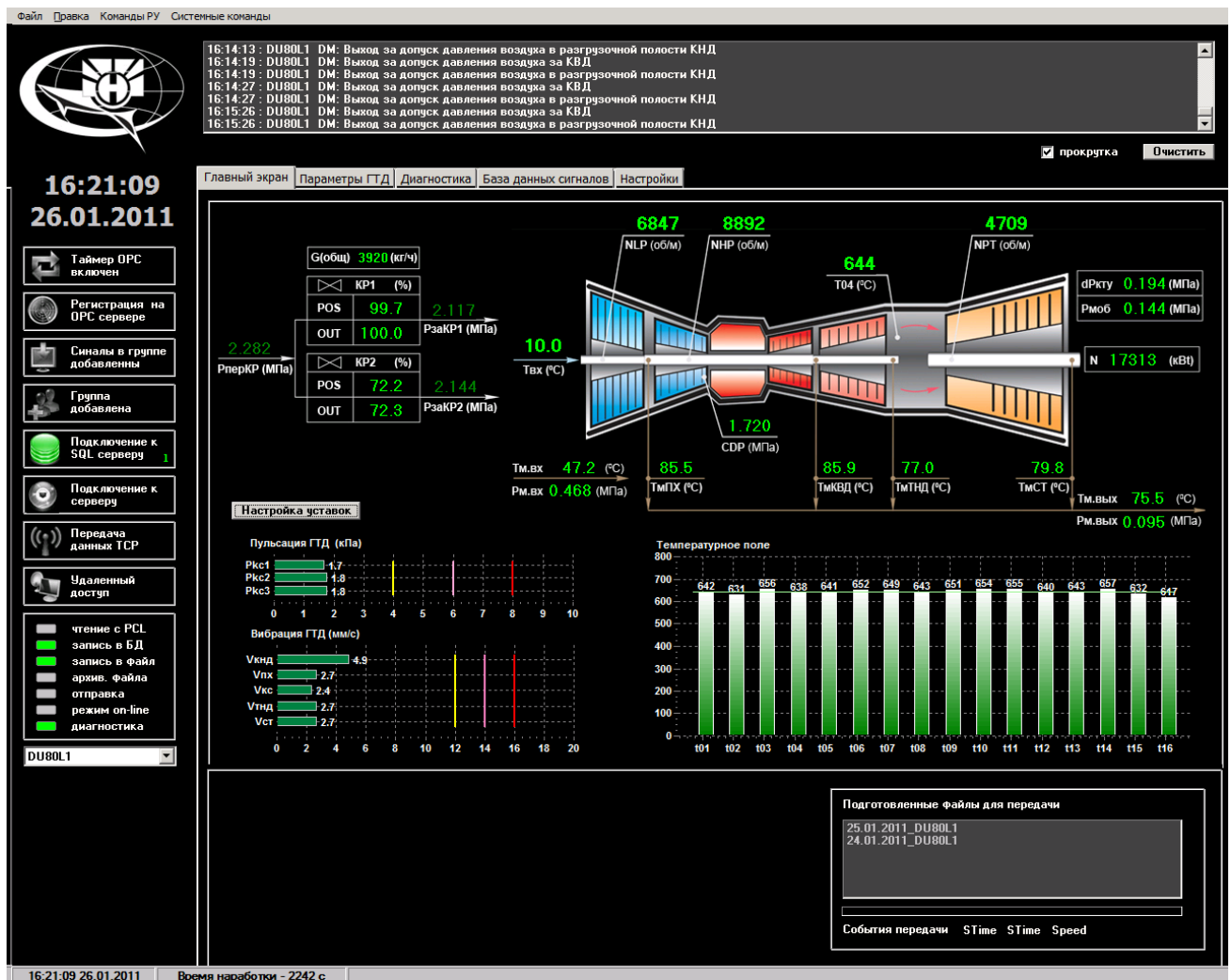


Рисунок 2.6 – Загальний вид інтерфейсу.

2.5.1 Модуль відправки даних

Система автоматичного керування обраного об'єкта, побудована на комплексі програмно-технічних засобів Series, що призначений для побудови керування турбомашинними, з метою регулювання й захисту турбоагрегатів практично всіх відомих типів. Основною функціональною одиницею Системи керування турбомашинними об'єктами Series є пристрій, іменоване функціонально-прикладним модулем (AFM). По призначенню AFM є аналогом традиційному регулятору, але набору й характеру своїх функцій це набагато більше універсальний засіб. AFM являють собою пристрою, "адаптовані" до конкретних застосувань

Виконуючі алгоритми керування, обробку вхідних вихідних сигналів і обмін інформацією із пристроями, що займають більше високі рівні в ієрархії керування. Концепція побудови функціонально-прикладного модуля ілюструється схемою, показаної на рисунку 2.7.

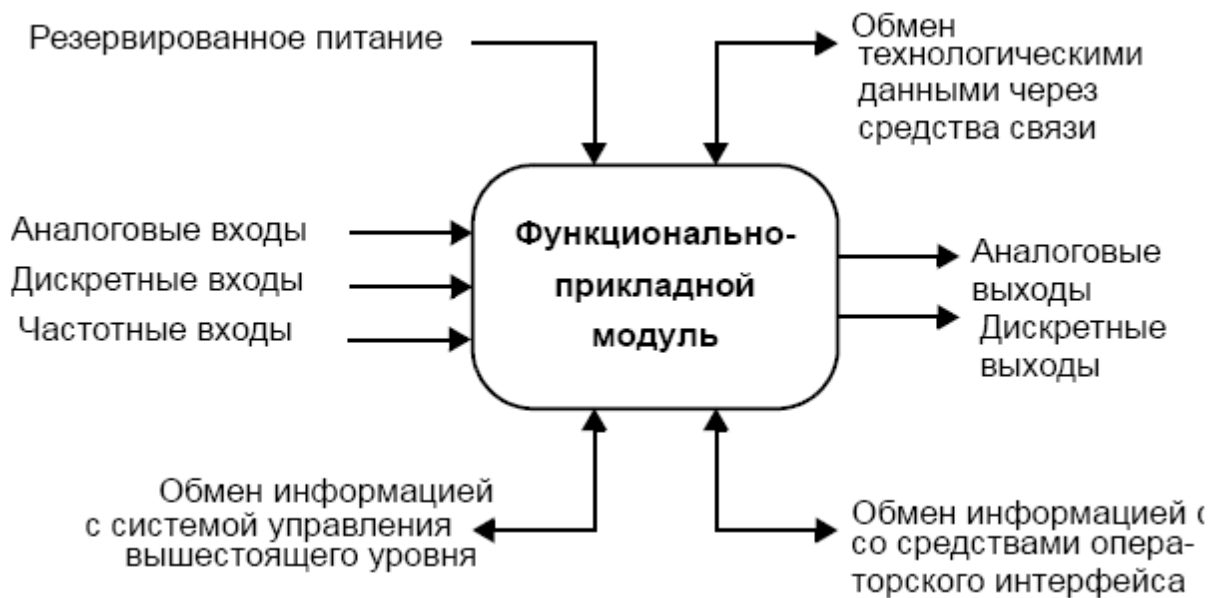


Рисунок 2.7 – Функционально-прикладный модуль Series

Кожний AFM оснащується двома асинхронними послідовними портами. Вони можуть бути сконфігуровані на роботу з одному із трьох перерахованих нижче протоколів інформаційного обміну:

- Протокол Series 4 - забезпечує підключення AFM до станцій операторського інтерфейсу (OIS).
- Протокол Modbus - забезпечує підключення AFM до розподілених систем керування й інших хост-системам.
- Протокол шини міжмодульного зв'язку (IMCB) - забезпечує підключення AFM до інших AFM системи.

Із представленого вище, можна зробити вивід, що обмін інформацією між програмно технічними засобами системи автоматичного керування й програмних засобів інформаційно-управляючої системи можна реалізувати по промисловому протоколі Modbus.

Modbus — комунікаційний протокол, заснований на архітектурі « клієнт-сервер ». Широко застосовується в промисловості для організації зв'язку між електронними пристроями. Може використовувати для передачі даних через послідовні лінії зв'язку RS-485, RS-422, RS-232, а також мережі TCP/IP (Modbus TCP).

Modbus ставиться до протоколів прикладного рівня мережної моделі OSI. Контролери на шині Modbus взаємодіють, використовуючи клієнт-клієнт-серверну модель, засновану на транзакціях, які складаються із запиту й відповіді.

Звичайно в мережі є тільки один сервер, так зване, «головне» (англ. master) пристрій, і трохи клієнтів — «підлеглих» (slaves) пристроїв. Головний пристрій ініціює шипика (передає запити). Підлегли пристрої передають запитувані головним пристроєм дані, або роблять запитувані дії. Головний може адресуватися індивідуально до підлеглого або ініціювати передачу широкомовного повідомлення для всіх підлеглих пристроїв. Підлегли й пристрій формує повідомлення й повертає його у відповідь на запит, адресований саме йому. При одержанні широкомовного запиту відповідне повідомлення не формується.

Специфікація Modbus описує структуру запитів і відповідей. Їхня основа — елементарний пакет протоколу, так званий PDU (Protocol Data Unit). Структура PDU не залежить від типу лінії зв'язку й містить у собі код функції й поле даних. Код функції кодується однобайтовим полем й може приймати значення в діапазоні 1...127. Діапазон значень 128...255 зарезервований для кодів помилок. Поле даних може бути змінної довжини. Розмір пакета PDU обмежений 253 байтами.

Modbus PDU	
номер функції	дані
1 байт	N < 253 (байт)

Для передачі пакета по фізичних лініях зв'язку PDU міститься в інший пакет, що містить додаткові поля. Цей пакет зветься ADU (Application Data Unit). Формат ADU залежить від типу лінії зв'язку.

Існують три основних реалізації протоколу Modbus, дві для передачі даних по послідовних лініях зв'язку, як мідним EIA/TIA-232-E (RS-232), EIA-422, EIA/TIA-485-A (RS-485), так і оптичним і радіо:

- Modbus ASCII — для обміну використовуються тільки ASCII символи. Для перевірки цілісності використовується алгоритм en:Longitudinal redundancy check. Повідомлення розділяється на стовпці за допомогою символу «:» і закінчується символами нового рядка CR/LF.

- Modbus RTU
- і для передачі даних по мережах Ethernet поверх TCP/IP:
- Modbus TCP.

Загальна структура ADU наступна (залежно від реалізації, деякі з полів можуть бути відсутні):

Адреса введеного пристрою	Код функції	Дані	Блок виявлення помилок
--	--------------------	-------------	-------------------------------

де

- Адреса веденого пристрою – адреса підлеглого пристрою, до якого адресований запит. Ведені пристрої відповідають тільки на запити, що надійшли на їх адресу. Відповідь також починається з адреси що відповідає веденому пристрою, що може змінюватися від 1 до 247. Адреса 0 використовується для широкомовної передачі, його розпізнає кожний пристрій, адреси в діапазоні 248...255 - зарезервовані;

- номер функції - це наступне однобайтне поле кадру. Воно говорить веденому пристрою, які дані або виконання якої дії жадає від його провідний пристрій;

- дані - поле містить інформацію, необхідну веденому пристрою для виконання заданої майстром функції або містить дані, передані введеним пристроєм у відповідь на запит ведучого. Довжина й формат поля залежить від номера функції;

- блок виявлення помилок - контрольна сума для перевірки відсутності помилок у кадрі.

Максимальний розмір ADU для послідовних мереж RS232/RS485 - 256 байт, для мереж TCP - 260 байт.

Для Modbus TCP ADU виглядає в такий спосіб:

ід транзакції	ід протокола	довжина пакету	адрес ведомого пристрою	код функції	дані
----------------------	---------------------	-----------------------	--------------------------------	--------------------	-------------

де

- ід транзакції - два байти, звичайно нулі
- ід протоколу - два байти, нулі

- довжина пакета - два байти, старший потім молодший, довжина наступної за цим полем частини пакета
- адреса веденого пристрою – адреса підлеглого пристрою, до якого адресований запит. Звичайно ігнорується, якщо з'єднання встановлене з конкретним пристроєм.

Поле контрольної суми в Modbus TCP відсутній.

Категорії кодів функцій

У діючій у цей час специфікації протоколу визначаються три категорії кодів функцій:

Стандартні команди

Їхній опис повинне бути опубліковане й затверджене Modbus-IDA. Ця категорія містить у собі як уже певні, так і вільні в цей час куркульки.

Користувальницькі команди

Два діапазони кодів (від 65 до 72 і від 100 до 110), для яких користувач може реалізувати довільну функцію. При цьому не гарантується, що якийсь інший пристрій не буде використовувати той же самий код для виконання іншої функції.

Зарезервовані

У цю категорію входять коди функцій, що не є стандартними, але вже використовувані в пристроях, вироблених різними компаніями. Це коди 9, 10, 13, 14, 41, 42, 90, 91, 125, 126 і 127.

Модель даних

Одне з типових застосувань протоколу - читання й запис даних у регістри контролерів. Специфікація протоколу визначає чотири таблиці даних:

Таблиця	Тип елемента	Тип доступу
Дискретні входи (Discrete Inputs)	Один біт	Тільки читання
Регістри флагов (Coils)	Один біт	Читання та запис
Регістри вводу (Input Registers)	16-бітне слово	Тільки читання
Регістри зберігання (Holding Registers)	16-бітне слово	Читання та запис

Доступ до елементів у кожній таблиці здійснюється за допомогою 16-бітної адреси, першому осередку відповідає адреса 0. Таким чином, кожна таблиця може містити до 65536 елементів. Специфікація не визначає, що фізично повинні являти

собою елементи таблиць і по яких внутрішніх адресах пристрою вони повинні бути доступні.

Читання даних

Для читання значень із перерахованих вище таблиць даних використовуються функції з кодами 1—4 (шістнадцяткові значення 0x01-0x04):

- 1 (0x01) — читання значень із декількох регістрів прапорів (Read Coil Status)
- 2 (0x02) — читання значень із декількох дискретних входів (Read Discrete Inputs)
- 3 (0x03) — читання значень із декількох регістрів зберігання (Read Holding Registers)
- 4 (0x04) — читання значень із декількох регістрів введення (Read Input Registers)

Запит складається з адреси першого елемента таблиці, значення якого потрібно прочитати, і кількості зчитувальних елементів. Адреса й кількість даних задаються 16-бітними числами, старший байт кожного з них передається першим.

У відповіді передаються запитані дані. Кількість байт даних залежить від кількості запитаних елементів. Перед даними передається один байт, значення якого дорівнює кількості байт даних.

Значення регістрів зберігання й регістрів введення передаються починаючи із зазначеної адреси, по двох байта на регістр, старший байт кожного регістра передається першим:

байт 1	байт 2	байт 3	байт 4	...	байт N-1	байт N
RA,1	RA,0	RA+1,1	RA+1,0	...	RA+Q-1,1	RA+Q-1,0

Значення прапорів і дискретних входів передаються в упакованому виді: по одному біті на прапор. Одиниця означає включений стан, нуль — виключене. Значення запитаних прапорів заповнюють спочатку перший байт, починаючи з молодшого біта, потім наступні байти, також від молодшого біта до старшого. Молодший біт першого байта даних містить значення прапора, зазначеного в поле «адреса». Якщо запитано кількість прапорів, не кратне восьми, то значення зайвихбітов заповнюються нулями:

байт 1								...	байт N							
FA+7	FA+6	FA+5	FA+4	FA+3	FA+2	FA+1	FA	...	0	...	0	FA+Q-1	FA+Q-2	...		

У протоколі ModbusRTU реалізованим AFM модулем Stries передбачено два типи даних: двійкові (дискретні) і цілочислові. Існує пряма кореляція між чотирма типами Modbus-Даних і двома типами даних AFM:

- Котушки - еквівалентом котушки в AFM Series 4 є двійкова (однобітова) змінна, значення якої може бути лічено головним пристроєм мережі Modbus або змінене (записано) цим пристроєм.

- Дискретні входи - еквівалентом дискретного входу в AFM є двійкова (однобітова) змінна, значення якому може бути тільки лічено, але не може бути змінено головним пристроєм мережі.

- Вхідні регістри - еквівалентом вхідного регістра в AFM є ціло числове (12-бітова) змінна, значення якої може бути тільки лічено, але не може бути змінено головним пристроєм мережі. Найчастіше вхідні регістри використовуються головним пристроєм для одержання з підлеглого пристрою 12-бітових цілих чисел, однак можливо їхнє застосування для одержання значень дванадцяти 1-бітовихзмінних, "упакованих" в 12-бітові слова (іноді такі утворення йменуються пакетуванням двійковими даними).

- Регістри проміжного зберігання - еквівалентом регістра проміжного зберігання в AFM є цілочислова змінна, значення якої може бути лічено головним пристроєм мережі Modbus або змінене (записано) цим пристроєм.

Відповідно до структури зберігання даних Modbus логічного контролера була створена аналогічна структура системи вилученого моніторингу, для спрощення прив'язки сигналів. Дана структура представлена нижче.

TIOMData=record

Coils: array[0..384] of byte;

Discrete: array[0..600] of byte;

Holding_Reg: array[0..192] of word;

Input_Reg: array[0..256] of word;

end;

PLCData: array[1..4] of TIOMData;

Для забезпечення спрощення підключення інформаційної системи до програмно логічного контролера, спрощення програмного коду, створення розгалуженої системи, запити до програмно логічного контролера будуть формуватися в режимі Modbus/TCP, канал передачі даних Ethernet. Для реалізації протоколу Modbus/TCP, був використаний модуль Modbus/TCP, що розроблений на базі компонента TIdUDPClient. Компонент TIdTCPClient (Indy клієнт TCP) використовується для створення з'єднання по протоколі TCP, має ряд функції по відправленню й прийому різної інформації. Для сумісності протоколів використовується комунікаційний шлюз MOXA MB3180.

MOXA MB3180 стандартні Modbus шлюз (малюнку 7), що конвертує Modbus TCP і Modbus RTU/ASCII протоколи. Він підтримує до 16 одночасних Modbus TCP "майстрів" (masters) і до 31 RTU/ASCII "ведених" (slaves) на послідовний порт. Для RTU/ASCII "майстрів" здійснюється підтримка до 32 TCP "ведених".



Рисунок 2.8 – MOXA MB3180

Спецификация MOXA MB3180.

■ Сеть

Ethernet	10/100 Mbps, RJ45, Auto DI/MDIX
Захист	Вбудована 1.5 KV magnetic isolation

■ Последовательный интерфейс

Интерфейс	RS-232/422/485 програмно обраний
Число портів	MB3180: 1 MB3280: 2 MB3480: 4 Male DB9

Сигнали	RS-232: TxD, RxD, RTS, CTS, DTR, DSR, DCD, GND RS-422: Tx+, Tx-, Rx+, Rx-, GND RS-485 (2-проводний): Data+, Data-, GND RS-485 (4-проводний): Rx+, Tx-, Rx+, GND
Захист	15 KV ESD для всех сигналів

■ Конфігураційні

Паритет	None, Even, Odd, Space, Mark
Біт даних	7, 8
Стоп біти	1 2
Керування потоком	RTS/CTS, XON/XOFF
Швидкість	от 50 bps до 921.6 Kbps

■ Програмне забезпечення

Modbus протоколи	RTU Slave, RTU Master, ASCII Slave, ASCII Master
Утиліти	MGate Manager Suite для Winows 98/ME/NT/2000XP/2003/Vista
Multi-master и multi-drop	Master mode: 32 TCP slaves. Slave mode: 16 TCP masters, 32 запиту для кожного "мастера"

■ Живлення та оточуюче середовище

Рабочая температура	от 0 до 55°C
Відносна вологість	от 5 до 95% RH
Температура зберігання	от -20 до 80°C
Матеріал корпусу	MB3180: Алюміній (1 мм) MB3280: Алюміній(1 мм) MB3480: SECC (0.8 mm)
Живлення	От 12 до 48 VDC
Підключення живлення	MB3180: разєм MB3280: разєм і термінальний блок MB3480: разєм і термінальний блок
Живлення, захист	1 KV Burst (ET), EN61000-4-4 0.5 KV Surge, EN61000-4-5

■ Сертифікати

EMC	CE: EN550022 Class A / EN550024 FCC: FCC Part 15 subpart B, Class A
Safety	TUV: EN60950-1 Shock: IEC60068-2-27 reefall: IEC60068-2-23 Vibration: IEC60068-2-6

Схема підключення системи автоматичного керування й системи вилученого моніторингу представлена на рисунку 2.8

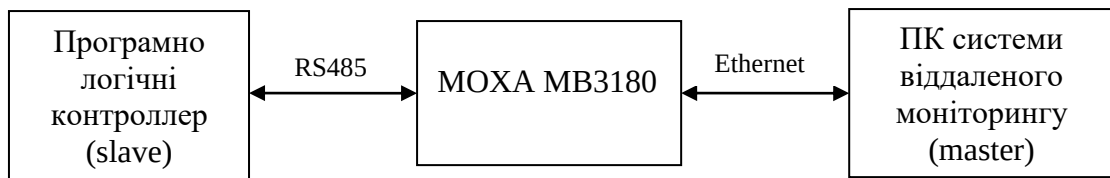


Рисунок 2.8 – Схема підключення

Структура даних для зберігання, що надходить із системи автоматичного керування, представлені у вигляді:

TSigR=record

Time:TDateTime;

AValue:array[1..80] of Smallint;

DValue:array[1..20] of Byte;

MLIM:array[0..19] of Longword;

end;

У дану структуру можливо записати до 80 показань аналогових сигналів, до 20 показань дискретних сигналів і 640 дискретних сигналів сформованих по 32 біта.

Для правильного формування обміну даними були зроблені наступні дії:

- Сконфігуровано порт логічного контролера САУ для роботи в режимі Modbus.
- Прив'язано сигнали відповідно до переліку (таблиця 1) у призначені регістри буфера Modbus.
- Сформовано запити відповідно до протоколу Modbus/TCP на зчитування ряду регістрів.
- Сформовано команди для читання результатів запитуваних функцій.

Тому що вихідні регістри котролера масштабують сигнали в діапазоні від 0..4095, то при зчитуванні показань виробляється обробка даних згідно формули.

$$\text{SigValue} = \frac{\text{ModbusValue} * (\text{SigMaxValue} - \text{SigMinValue})}{4095} + \text{SigOffset}$$

де

ModbusValue: значення отримані з Modbus регістра;

SigMaxValue: максимальне значення сигналу по переліку;

SigMinValue: мінімальне значення сигналу по переліку;

SigOffset: зсув відносно 0 (SigOffset= SigMinValue).

Блок схема роботи модуля обміну даними по протоколі Modbus представлена на малюнку 9, лістинг представлений у додатку А.

Таблиця 2.1 – перелік зчитувальних сигналів

Имя по перечню	Описание сигнала	Gain	Offset	Min Value	Max Value
EN110	Частота обертання КНД, об/мин (т1)	2.442	0	0	10000
EN120	Частота обертання КНД, об/мин (т2)	2.442	0	0	10000
EN210	Частота обертання КВД, об/мин (т1)	2.442	0	0	10000
EN220	Частота обертання КВД, об/мин (т2)	2.442	0	0	10000
EN310	Частота обертання СТ, об/мин (т1)	1.465	0	0	6000
EN320	Частота обертання СТ, об/мин (т2)	1.465	0	0	6000
EP010	Давление воздуха за КВД, кПа (т1)	0.610	0	0	2500
EP110	Давление воздуха за КВД, кПа (т2)	0.610	0	0	2500
EP050	Давление воздуха в разгрузочной полости КНД, МПа	9.768E-05	0	0	0.4
EP120	Тиск повітря за КНД, кПа	0.0002442	0	0	1
IP210	Барометрическое давление воздуха, кПа	0.034188	0	0	140
T1_avg	Температура воздуха на входе в ЕТУ, °C (средняя)	-99.95116	-100	-100	100
T2_calc	Температура воздуха за КНД, °C (средняя)	0.14652015	0	0	600
ET230	Температура газа за ТНД, °C (т1)	0.31746032	0	0	1300
ET240	Температура газа за ТНД, °C (т2)	0.31746032	0	0	1300
ET250	Температура газа за ТНД, °C (т3)	0.31746032	0	0	1300
ET260	Температура газа за ТНД, °C (т4)	0.31746032	0	0	1300
ET270	Температура газа за ТНД, °C (т5)	0.31746032	0	0	1300
ET280	Температура газа за ТНД, °C (т6)	0.31746032	0	0	1300
ET290	Температура газа за ТНД, °C (т7)	0.31746032	0	0	1300
ET300	Температура газа за ТНД, °C (т8)	0.31746032	0	0	1300
ET310	Температура газа за ТНД, °C (т9)	0.31746032	0	0	1300
ET320	Температура газа за ТНД, °C (т10)	0.31746032	0	0	1300
ET330	Температура газа за ТНД, °C (т11)	0.31746032	0	0	1300
ET340	Температура газа за ТНД, °C (т12)	0.31746032	0	0	1300
ET350	Температура газа за ТНД, °C (т13)	0.31746032	0	0	1300
ET360	Температура газа за ТНД, °C (т14)	0.31746032	0	0	1300
ET370	Температура газа за ТНД, °C (т15)	0.31746032	0	0	1300
ET380	Температура газа за ТНД, °C (т16)	0.31746032	0	0	1300
T04SR	Температура газа за ТНД, °C (средняя)	0	0		
EB010	Вибрация корпуса ЕТУ (ПФ КНД), мм/с (СКЗ)	0.01221001	0	0	50
EB020	Вибрация корпуса ЕТУ (ЗФ переходника), мм/с (СКЗ)	0.01221001	0	0	50
EB030	Вибрация корпуса ЕТУ (ЗФ КС), мм/с (СКЗ)	0.01221001	0	0	50
EB040	Вибрация корпуса ЕТУ (ЗФ ОВ ТНД), мм/с (СКЗ)	0.01221001	0	0	50
EB050	Вибрация корпуса ЕТУ (ПФ ОВ СТ), мм/с (СКЗ)	0.01221001	0	0	50
EB060	Пульсация давления воздуха в КС ЕТУ, кПа	0.01221001	0	0	50
EB070	Пульсация давления воздуха в КС ЕТУ, кПа	0.01221001	0	0	50
EB080	Пульсация давления воздуха в КС ЕТУ, кПа	0.01221001	0	0	50
EF110	Частота тока в цепи РЧТиН №1, Гц	0.01465201	0	0	60
EF210	Частота тока в цепи РЧТиН №2, Гц	0.01465201	0	0	60
FP010	Тиск палива газа за КР1, кПа	0.97680098	0	0	4000
FP020	Тиск палива газа перед КР1, кПа	0.97680098	0	0	4000
FP030	Тиск палива газа за КР2, кПа	0.97680098	0	0	4000
FP040	Тиск палива газа перед КР2, кПа	0.97680098	0	0	4000

FT010	Температура газу на вході в систему, °C	-99.95116	-100	-100	100
FA010	Положення КР1, %	0.02442002	0	0	100
FA020	Положення КР2, %	0.02442002	0	0	100
FA016	Керуючий сигнал на КР1, %	0.02442002	0	0	100
FA026	Керуючий сигнал на КР2, %	0.02442002	0	0	100
FQ010	Расход топливного газа, кг/ч	0.48840049	0	0	2000
LP020	Тиск масла на виході з ЕТУ, МПа	9.768E-05	0	0	0.4
LP110	Перепад давления КНД-МОБ, МПа	9.768E-05	0	0	0.4
LP130	Давление в МОБ, МПа	9.768E-05	0	0	0.4
LP200	Давление масла на входе в ЕТУ (перед фильтром), МПа	9.768E-05	0	0	0.4
LT010	Температура масла на вході в ЕТУ (в баке), МПа	-98.4119121	-98.46	-98.46	98.46
LT020	Температура масла на сливе из переходника, МПа	0.04806105	0	0	196.81
LT030	Температура масла на сливі з ЗККВД, МПа	0.04806105	0	0	196.81
LT040	Температура масла на сливі з ОВ ТНД, МПа	0.04806105	0	0	196.81
LT050	Температура масла на сливі з ОВ СТ, МПа	0.04806105	0	0	196.81
LT060	Температура масла на виході з ЕТУ (перед маслоохладителем), МПа	0.04806105	0	0	196.81
LT070	Температура масла за маслоохолоджувачем, МПа	0.04884005	0	0	200

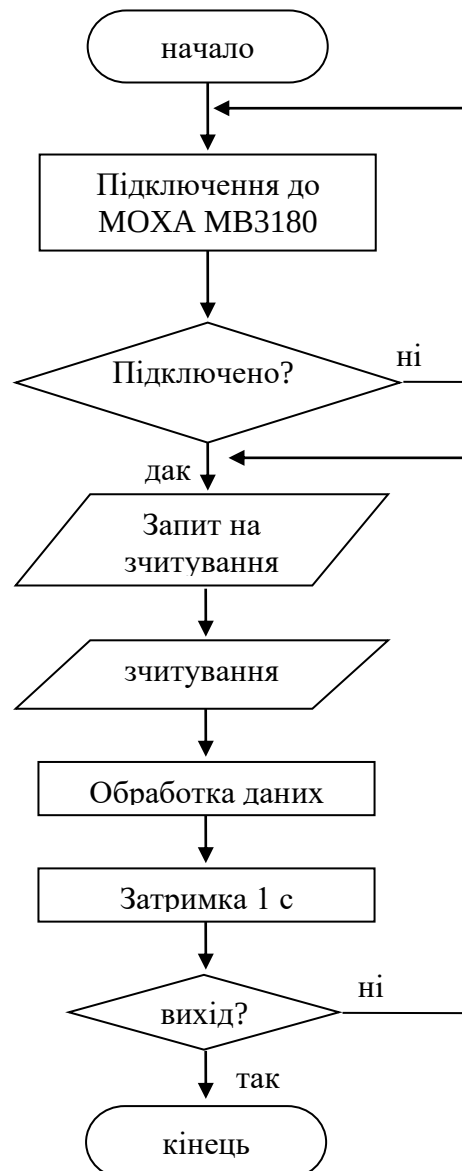


Рисунок 2.9 – Блок-схема обміну даними за протоколом Modbus

2.6 Модуль передачі даних у СУБД

Модуль керування потоком даних призначений для керування потоком даних отриманих з контролера, основними функціями модуля є:

- Обробка даних отриманих із програмно логічного контролера;
- Підготовка даних для запису із СУБД;
- Підготовка даних для обміну із системою параметричної діагностики;
- Підготовка архівного файлу для відправлення на сервер зберігання даних;
- Підготовка даних для передачі в режимі прямого підключення зовнішніми клієнтами;
- Підготовка й візуалізація даних для оператора.

Функціональна схема модуля керування даними представлена на малюнку Модуль представлений у вигляді процедур і функцій, лістинг представлений у додатку А.

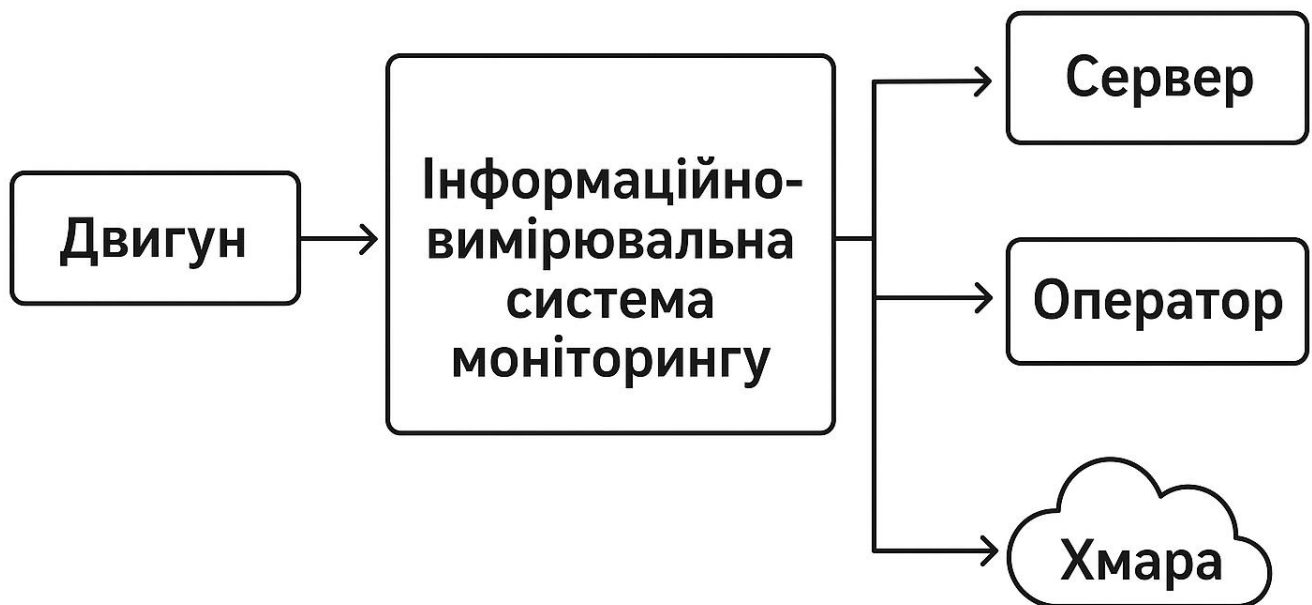


Рисунок 2.10 – Функціональна схема модуля управління даними

2.7 Модуль управління потоком даних STD

Модуль передачі даних у СУБД реалізований з використанням базових компонентів ADO:

- **TADOConnection**: здійснює зв'язок інших компонентів з ADO. Для цих цілей у нього є рядок `ConnectionString`. Після того, як за допомогою цього рядка зв'язок з даними встановлена, на компонент можуть посылатися інші ADO-Компоненти, розділяючи встановлену їм зв'язок.
- **TADOCommand**: призначений, в основному, для реалізації SQL-Запитів, що не повертають ніяких даних (підмножину Data Definition Language - мова визначення даних мови структурованих запитів SQL).
- **TADOQuery**: призначений для одержання набору записів з однієї або декількох таблиць БД. У нього включений специфічний метод `ExecSQL`, за допомогою якого компонент може виконувати пропозиції DDL мови SQL.

Всі дані, які постукають із САУ, а так само ті дані, які надходять із модуля діагностики записуються у відповідну базу даних. Дані зберігаються в таблицях. Щодоби створюється нова таблиця, що у назві включає дату й унікальний номер двигуна.

MySQL підтримує наступні типи таблиць:

1. MyISAM.

MyISAM прийнятий за замовчуванням в MySQL версії 3.23. Він ґрунтується на коді ISAM і володіє в порівнянні з ним більшою кількістю корисних доповнень. Індекс зберігається у файлі з розширенням `.MYI` (MYIndex), а дані - у файлі з розширенням `.MYD` (MYData). Таблиці MyISAM можна перевіряти/відновлювати за допомогою утиліти `myisamchk`. Таблиці MyISAM можна стискати за допомогою команди `myisampack`, після чого вони будуть займати набагато менше місця.

В MyISAM підтримується три різних типи таблиць. Два з них вибираються автоматично, залежно від типу використовуваних стовпців. Третій - стислі таблиці - може бути створений тільки за допомогою інструмента `myisampack`.

У файлі MyISAM зберігається прапор, що вказує, про правильність закритті таблиці. У випадку запуску `mysqld` з параметром `myisam recover` таблиці MyISAM

будуть автоматично перевірятися й/або відновлюватися при відкритті, якщо таблиця була закрита неправильно.

За допомогою команди INSERT можна вставляти нові рядки в таблицю, у середині файлу даних якої немає вільних блоків, у той час як інші потоки зчитують із таблиці інформацію (сполучена вставка). Вільний блок може бути отриманий при відновленні рядка з динамічною довжиною, коли більша кількість даних замінюється меншою кількістю або при видаленні рядків. Коли вільних блоків не залишається, всі наступні блоки знову будуть вставлятися як сполучені.

MyISAM підтримує формат більших файлів (63 біта) у файлові/операційних системах, які підтримують більші файли.

Зберігання всіх даних здійснюється з першим молодшим байтом. Це робить дані незалежними від операційної системи. Єдина вимога - у комп'ютері повинні застосовуватися доповнені до двох байтів цілі числа зі знаком і формат із плаваючої одиничної коми IEEE. Єдиними комп'ютерами, які можуть не підтримувати бінарну сумісність, є убудовані системи (оскільки в них іноді застосовуються спеціальні процесори). При зберіганні даних з першим молодшим байтом не відбувається зниження швидкості. Звичайно байти в рядку таблиці не вирівняні й немає великої різниці в тім, як прочитати невирівняний байт - у прямої послідовності або у зворотній. Фактичний час добування значення стовпця також не критично в порівнянні згодом виконання іншого коду.

Всі ключі номерів зберігаються з першим старшим байтом, щоб стиск індексів був більше ефективним.

Внутрішня обробка стовпця AUTO_INCREMENT. MyISAM автоматично оновлює його при виконанні команд INSERT/UPDATE.

Стовпці BLOB і TEXT можуть бути проіндексовані.

В індексних стовпцях дозволені значення NULL. Вони займають 0-1 байта на ключ.

За замовчуванням максимальна довжина ключа становить 500 байтів (це значення може бути змінене при повторній компіляції). У випадках, коли ключі більше 250 байтів, для них використовуються більші розміри блоку ключа, чим передбачені за замовчуванням 1024 байта.

За замовчуванням у таблиці може бути не більше 32 ключів. Це значення можна збільшити до 64 без повторної компіляції `myisamchk`.

Рядка з динамічним розміром будуть менш фрагментованими, чим при змішуванні видалень із відновленнями й вставками. Це здійснюється шляхом автоматичного сполучення вилучених суміжних блоків і розширенням блоків, якщо наступний блок вилучений.

MyISAM також підтримує наступні функції, які можна буде використовувати в MySQL у найближчому майбутньому:

Підтримка типу `VARCHAR`; стовпець `VARCHAR` починається з довжини, що зберігається в 2 байтах.

Таблиці з `VARCHAR` можуть мати фіксовану або динамічну довжину записів.

`VARCHAR` і `CHAR` можуть бути до 64 Кб довжиною. У всіх ключових сегментів є свої власні визначення мови. Це дозволяє задавати в MySQL різні визначення мови для кожного стовпця.

Таблиці `MERGE` (об'єднання) є нововведенням версії MySQL 3.23.25. У цей час куркульок перебуває ще на стадії розробки, але, проте, повинен бути досить стабільний.

2. Merge.

Таблиця `MERGE` (або таблиця `MRG_MyISAM`) являє собою сукупність ідентичних таблиць `MyISAM`, які можуть використовуватися як одна таблиця. До сукупності таблиць можна застосовувати тільки команди `SELECT`, `DELETE` і `UPDATE`. Якщо ж спробувати застосувати до таблиці `MERGE` команду `DROP`, вона подіє тільки на визначення `MERGE`.

Під ідентичними таблицями маються на увазі таблиці, створені з однаковою структурою й ключами. Не можна поєднувати таблиці, у яких стовпці стислі різними методами або не збігаються, або ключі розташовані в іншому порядку.

При створенні таблиці `MERGE` будуть утворені файли визначень таблиць `.frm` і списку таблиць `.MRG`. Файл `.MRG` містить список індексних файлів (файли `.MYI`), робота з якими повинна здійснюватися як з єдиним файлом. Всі використовувані таблиці повинні розміщатися в тій же базі даних, що й таблиця `MERGE`.

На даний момент стосовно таблиць, які необхідно перетворити в таблицю MERGE, необхідно мати привілеї SELECT, UPDATE і DELETE.

Нижче перераховані можливості, які забезпечують таблиці MERGE:

Просте керування набором файлів журналів.

Збільшення швидкості роботи. Більшу таблицю можна розділити за деяким критерієм, а потім помістити різні частини таблиці на різні диски. У цьому випадку таблиця MERGE може оброблятися набагато швидше, ніж звичайна більша таблиця (можна, звичайно, скористатися дисковим масивом RAID, щоб одержати ті ж переваги).

Більше ефективний пошук. Їсти можливість робити пошук по певних запитах тільки в одній із тридцятилітніх таблицю MERGE таблиць, одночасно використовуючи таблицю MERGE для інших запитів.

Більше просте відновлення. Більше ефективніше відновити окремі файли, які перетворені у файл MERGE, чим відновлювати великий файл.

Швидка обробка великої кількості файлів як одного. Для таблиці MERGE використовуються індекси окремих таблиць; підтримувати для неї один великий індекс немає необхідності. Завдяки цьому створення або зміна таблиць MERGE здійснюється дуже швидко.

Швидкодія й економія дискового простору досягається шляхом створення таблиці MERGE, якщо потрібно об'єднати кілька таблиць в одну більшу таблицю. Таблиці MERGE дозволяють обходити обмеження на розмір файлів в операційних системах.

Можливість створення псевдонімів/синонімів для таблиці - для цього потрібно просто застосувати MERGE до однієї таблиці. Помітного падіння продуктивності при цьому не спостерігатися.

3. Heap.

Для HEAP-Таблиць використовуються хеш-індекси, ці таблиці зберігаються в пам'яті. Завдяки цьому обробка їх здійснюється дуже швидко, однак у випадку збою MySQL будуть втрачені всі дані, які в них зберігалися. Тип HEAP добре підходить для тимчасових таблиць.

Для внутрішніх HEAP-Таблиць в MySQL використовується 100%-е динамічне хешування без областей переповнення; додатковий простір для вільних списків

не потрібно. Відсутні при використанні HEAP-Таблиць і проблеми з командами видалення й вставки, які часто застосовуються в хешированих таблицях:

При використанні HEAP-Таблиць необхідно звертати увагу на наступні моменти:

- необхідно завжди вказувати параметр `MAX_ROWS` в операторі `CREATE`, щоб випадковим образом не зайняти всю пам'ять.
- в HEAP-Таблицях для пошуку рядка можуть використовуватися тільки повні ключі, у той час як для таблиць `MyISAM` при пошуку рядків може застосовуватися будь-який префікс ключа.
- для HEAP-Таблиць використовується формат з фіксованою довжиною запису.
- для HEAP-Таблиць не підтримуються стовпці формату `BLOB/TEXT`.
- для HEAP-Таблиць не підтримуються стовпці формату `AUTO_INCREMENT`.
- до версії 4.0.2 для HEAP-Таблиць не підтримуються індекси в стовпцях формату `NULL`.
- в HEAP-Таблицях можуть зустрічатися співпадаючі ключі (що не є нормою для хешированих таблиць).
- HEAP-Таблиці використовуються спільно всіма клієнтами (як і всі інші таблиці).
- не можна робити пошук наступного запису в порядку проходження (тобто використовувати індекс у команді `ORDER BY`).
- дані HEAP-Таблиць розташовані в маленьких блоках. Таблиці на 100% є динамічними (при вставці). Немає необхідності ні в областях переповнення, ні в додаткових ключах. Вилучені рядки містяться у зв'язаний список і використовуються при вставці в таблицю нових даних.
- варто подбати про те, щоб була достатня кількість додаткової пам'яті для всіх HEAP-Таблиць, які будуть використовуватися одночасно.
- щоб звільнити пам'ять, необхідно запустити команду `DELETE FROM heap_table`, `TRUNCATE heap_table` або `DROP TABLE heap_table`.
- `MySQL` не може підрахувати, скільки рядків перебуває між двома значеннями (використовується оптимізатором діапазонів для вибору

використовуваного індексу). Це може вплинути на деякі запити, якщо перетворити таблицю MyISAM у формат HEAP.

- при створенні розмір таблиці HEAP не може перевищувати `max_heap_table_size`; це зроблено для того, щоб забезпечити захист від випадкових некваліфікованих дій.

4. InnoDB

Таблиці InnoDB в MySQL постачені оброблювачем таблиць, що забезпечують безпечні шипики (рівня ACID) з можливостями фіксації шипика, відкоту й відновлення після збою. Для таблиць InnoDB здійснюється блокування на рівні рядка, а також використовується метод читання без блокувань у команді SELECT. Перераховані функції дозволяють поліпшити взаємну сумісність і підвищити продуктивність у багатокористувальницькому режимі. В InnoDB немає необхідності в розширенні блокування, тому що блоки рядків в InnoDB займають дуже мало місця. Для таблиць InnoDB підтримуються обмежуючі умови FOREIGN KEY.

InnoDB призначається для одержання максимальної продуктивності при обробці більших обсягів даних. По ефективності використання процесора цей тип набагато перевершує інші моделі реляційних баз даних з пам'яттю на дисках.

Технічно InnoDB є завершеною системою керування базою даних у рамках MySQL. В InnoDB є свій власний буферний пул для кешування даних і індексів в основній пам'яті. Таблиці й індекси InnoDB зберігаються в спеціальному просторі пам'яті, що може складатися з декількох файлів. У цьому полягає відмінність InnoDB від, наприклад, таблиць MyISAM: кожна таблиця MyISAM зберігається в окремому файлі. Таблиці InnoDB можуть бути будь-якого розміру навіть у тих операційних системах, де встановлене обмеження файлу в 2 Гб.

У даний момент (жовтень 2001 року) таблиці InnoDB застосовуються там, де важлива висока продуктивність.

Таблиці InnoDB входять у дистрибутив вихідних текстів MySQL, починаючи з версії 3.23.34a; вони активізовані у виконується коді, щоМ, MySQL-Max. Для Windows виконуються коды, щоМ, -Max перебувають у стандартному дистрибутиві.

Проаналізувавши основні типи таблиць, які підтримує MySQL можна зробити вивід, що для зберігання даних, що надходять від інформаційно-управляючої системи, найбільш підходящими є таблиці MyISAM, які мають надійність, швидкодією, відновленням.

Для зберігання інформації, відповідно до структури даних, програмними засобами інформаційно-управляючої системи створюється таблиця:

```
CREATE TABLE `18012011_du8011` (  
  `Time` time default NULL,  
  `EN110` float default NULL,  
  `EN120` float default NULL,  
  `EN210` float default NULL,  
  `EN220` float default NULL,  
  `EN310` float default NULL,  
  `EN320` float default NULL,  
  `EP010` float default NULL,  
  `EP110` float default NULL,  
  `EP030` float default NULL,  
  `EP050` float default NULL,  
  `EP120` float default NULL,  
  `IP210` float default NULL,  
  `T1_avr` float default NULL,  
  `T2_calc` float default NULL,  
  `ET230` float default NULL,  
  `ET240` float default NULL,  
  `ET250` float default NULL,  
  `ET260` float default NULL,  
  `ET270` float default NULL,  
  `ET280` float default NULL,  
  `ET290` float default NULL,  
  `ET300` float default NULL,  
  `ET310` float default NULL,  
  `ET320` float default NULL,  
  `ET330` float default NULL,  
  `ET340` float default NULL,  
  `ET350` float default NULL,  
  `ET360` float default NULL,  
  `ET370` float default NULL,  
  `ET380` float default NULL,  
  `T04SR` float default NULL,  
  `EB010` float default NULL,  
  `EB020` float default NULL,  
  `EB030` float default NULL,  
  `EB040` float default NULL,  
  `EB050` float default NULL,  
  `EB060` float default NULL,
```

```
`EB070` float default NULL,
`EB080` float default NULL,
`EF110` float default NULL,
`EF210` float default NULL,
`FP010` float default NULL,
`FP020` float default NULL,
`FP030` float default NULL,
`FP040` float default NULL,
`FT010` float default NULL,
`FA010` float default NULL,
`FA020` float default NULL,
`FA016` float default NULL,
`FA026` float default NULL,
`FQ010` float default NULL,
`LP010` float default NULL,
`LP020` float default NULL,
`LP110` float default NULL,
`LP130` float default NULL,
`LP200` float default NULL,
`LT010` float default NULL,
`LT020` float default NULL,
`LT030` float default NULL,
`LT040` float default NULL,
`LT050` float default NULL,
`LT060` float default NULL,
`LT070` float default NULL,
`LL100` float default NULL,
`FQ110` float default NULL,
`RB040` float default NULL,
`N` float default NULL,
`F68` float default NULL,
`F69` float default NULL,
`F70` float default NULL,
`F71` float default NULL,
`F72` float default NULL,
`F73` float default NULL,
`F74` float default NULL,
`F75` float default NULL,
`F76` float default NULL,
`F77` float default NULL,
`F78` float default NULL,
`F79` float default NULL,
`F80` float default NULL,
`LS010` tinyint(4) default NULL,
`LS020` tinyint(4) default NULL,
`LS030` tinyint(4) default NULL,
`LS040` tinyint(4) default NULL,
`LS050` tinyint(4) default NULL,
`LS060` tinyint(4) default NULL,
```

```

`CK010` tinyint(4) default NULL,
`EK020` tinyint(4) default NULL,
`EK010` tinyint(4) default NULL,
`stControl` tinyint(4) default NULL,
`stGTD` tinyint(4) default NULL,
`D12` tinyint(4) default NULL,
`D13` tinyint(4) default NULL,
`D14` tinyint(4) default NULL,
`D15` tinyint(4) default NULL,
`D16` tinyint(4) default NULL,
`D17` tinyint(4) default NULL,
`D18` tinyint(4) default NULL,
`D19` tinyint(4) default NULL,
`D20` tinyint(4) default NULL,
`Lim00` int(10) unsigned default NULL,
`Lim01` int(10) unsigned default NULL,
`Lim02` int(10) unsigned default NULL,
`Lim03` int(10) unsigned default NULL,
`Lim04` int(10) unsigned default NULL,
`Lim05` int(10) unsigned default NULL,
`Lim06` int(10) unsigned default NULL,
`Lim07` int(10) unsigned default NULL,
`Lim08` int(10) unsigned default NULL,
`M10` int(10) unsigned default NULL,
`M11` int(10) unsigned default NULL,
`M12` int(10) unsigned default NULL,
`M13` int(10) unsigned default NULL,
`M14` int(10) unsigned default NULL,
`M15` int(10) unsigned default NULL,
`DMess1` int(10) unsigned default NULL,
`DMess2` int(10) unsigned default NULL,
`DMess3` int(10) unsigned default NULL,
`DMess4` int(10) unsigned default NULL,
`DMess5` int(10) unsigned default NULL
) ENGINE=MyISAM DEFAULT CHARSET=cp1251
ROW_FORMAT=COMPACT;

```

Як видно з вище представленої команди, назви стовпців відповідають назвам сигналів згідно переліку, дана команда формується динамічно шляхом зчитування інформації з локальної бази даних сигналів. Це дає можливість не мене програмного коду створювати різні таблиці, у випадки зміни переліку, додавання або видалення деяких сигналів. Поля таблиці мають наступну розмірність:

- 8 байт: поле часу;
- 4 байти формату float: поле зберігання аналогових сигналів;
- 1 байт формату int: поле зберігання дискретних сигналів;

- 4 байти формату int: поле зберігання повідомлень (побітно), упакованих у форматі 32 біт.

Запис у таблицю здійснюється за допомогою команди INSERT, що формується так само динамічно, ім'я бази даних є унікальний номер двигуна. Блок схема роботи модуля передачі даних у СУБД представлена на малюнку 2.11, лістинг модуля представлений у додатку А

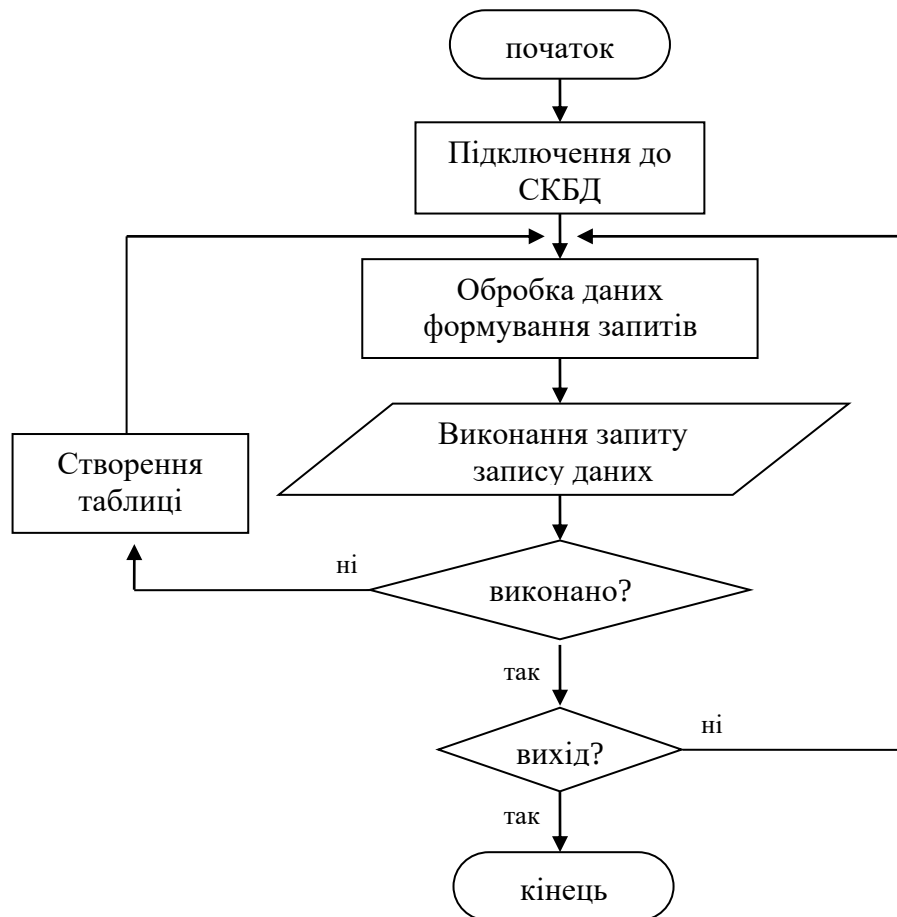


Рисунок 2.11 – Блок схема роботи модуля передачі даних в СУБД

2.7 Модуль збору й обробки даних

Модуль відправлення даних призначений для передачі добового архіву на головний сервер зберігання даних. Даний модуль заснований на клієнт-серверної технології, з використанням базового компонента TIdTCPClient .

Клієнт-Сервер (англ. Client-server) — обчислювальна або мережна архітектура, у якій завдання або мережне навантаження розподілені між постачальниками послуг (сервісів), називаними серверами, і замовниками послуг, називаними клієнтами. Нерідко клієнти й сервери взаємодіють через комп'ютерну

мережу й можуть бути як різними фізичними пристроями, так і програмним забезпеченням.

TIdTCPClient має ряд параметрів, що набудовуються, які доступні для користувача. Інтерфейс налаштування параметрів модуля передачі даних представлений на малюнку 2.12.



Параметр	Значение
IPServer	213.186.98.95
Port	6060
ReadTimeOut	25000
MinBufferSize	2048
MaxBufferSize	16000
Control Time	1500
Connection Try	15

Рисунок 2.12 – Налаштовуючі параметри модуля передачі даних

Представлено наступні параметри:

- IPServer: IP адреса сервера збору даних;
- Port: порт сервера збору даних;
- ReadTimeOut: час очікування відгуку сервера;
- MinBufferSize: мінімальний розмір буфера, що посилається;
- MaxBufferSize: максимальний розмір буфера, що посилається;
- ControlTime: контрольний час відправлення даних;
- ConnectionTry: кількість спроб підключення до сервера.

Принцип роботи модуля відправлення даних. При виклику модуля відправлення даних виробляється спроба підключення до сервера зберігання даних. Якщо спроба не вдала, то виробляється перепідключення через певний час, кількість спроб налаштовується користувачем. При перевищенні кількості спроб, підключиться можливо буде через добу. Якщо підключення успішно, то передається інформація про переданий файл у форматі:

TFileInfoSend=record

```
Name: String[30]; // ім'я файлу;  
Extension:String[4]; // розширення;  
Size:LongWord; //розмір файлу;  
end;
```

У відповідь клієнт одержує підтверджувальну інформацію про готовність прийняти файл. Надалі відбувається передача файлу по наступному алгоритмі: переданий файл заноситься в буфер і в розділ передається на сервер. У першу чергу відправляється розмір мінімального буфера відправлення, і сам буфер (частина файлу). Якщо час відправлення менше контрольованого, то буфер збільшується на певний розмір, якщо час відправлення менше контрольованого то буфер зменшується. Якщо час відправлення критично, то розмір буфера, що відправляється, стає мінімальним. Сервер у свою чергу створює буфер розміром як у переданого файлу, а так само буфер для прийняття частин файлу. Сервер приймає файл у вигляді частин, які формуються в єдиний файл. Надалі сервер розпаковує переданий файл, і дані заносить у СУБД.

Блок схема модуля відправлення даних представлена на малюнку 13, лістинг представлений у додатку А.

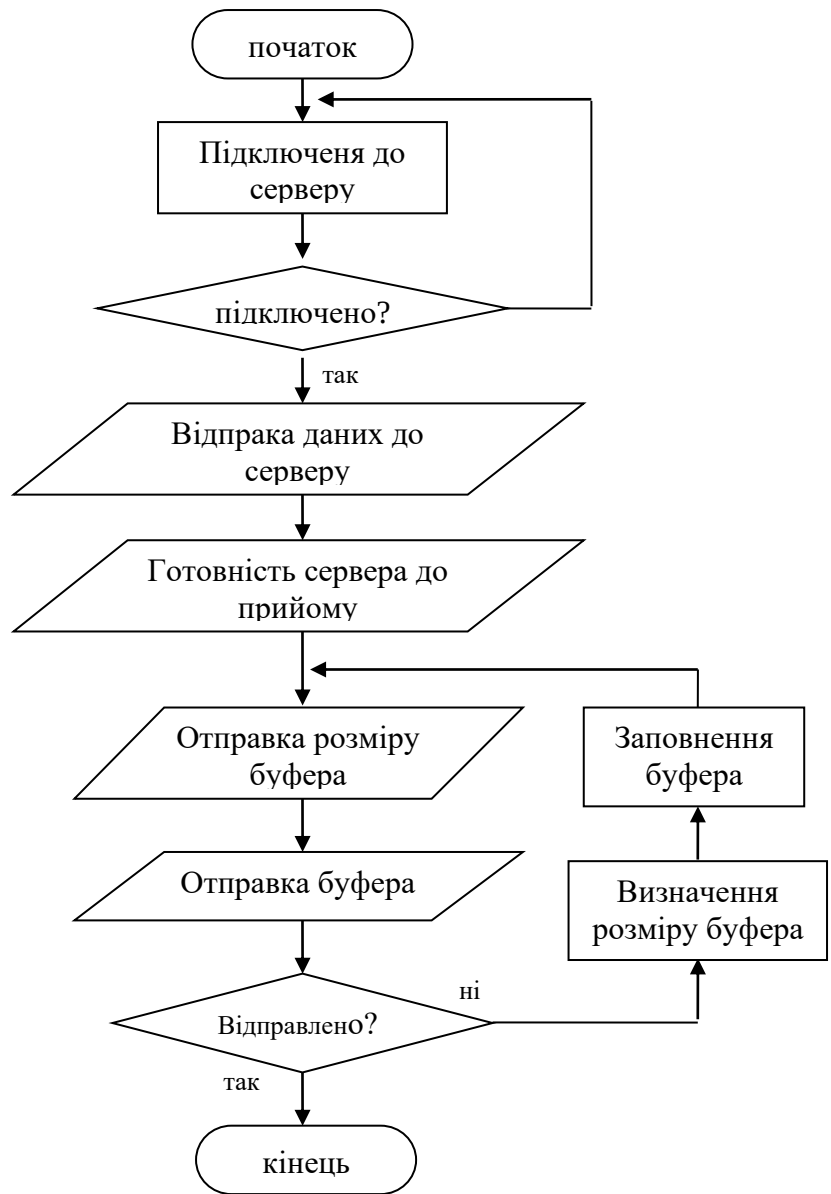


Рисунок 2.13 – Блок схема модулю відправки даних

2.8 Розробка модулю передачі даних у систему технічної діагностики

Обмін даними між клієнтами СТД і властиво СТД виробляється по протоколі TCP/IP.

СТД є сервером TCP/IP і забезпечує прийом вихідних даних від клієнтської частини, обробку даних алгоритмами діагностики й передачу вихідних діагностичних параметрів клієнтської частини.

Всі передані параметри СТД розділені по типах (категоріям). Кожний переданий параметр є екземпляром призначеного йому типу.

Пакет вхідних параметрів, прийнятих СТД від клієнта, являє собою набір масивів аналогового й бінарного типів. Розмірність масивів і порядок проходження параметрів строго погоджено.

```
TAValue=record
```

```
    Name:string[10];
```

```
    AVal:Single;
```

```
    FVal:byte;
```

```
end;
```

```
TDDataSend=record
```

```
    ASignal:array[1..80] of TAValue;
```

```
    DValue:array[1..20] of Byte;
```

```
end;
```

Пакет вихідних параметрів, переданих СТД по запиті клієнта, являє собою набір масивів аналогового типу, масивів бінарного типу, а також масив типу ситуацій і масив типу відхилень. Розмірність масивів і порядок проходження параметрів строго погоджено.

```
TDeviation = record
```

```
    d_Name:String[10];    // Имя контролируемого параметра
```

```
    d_calc:byte;          ///Признак с расчетным контр.  
параметром отклонения 1-да, 0-нет
```

```
    d_order:byte;        //порядковый номер отклонения
```

```
    //Отклонение
```



```

d_DUse:byte;      //Признак расчета отклонения
d_ParVal:Single;   //Значение отклонения
d_MinTolerance:Single; //Минимальный допуск
d_MaxTolerance:Single; //Максимальный допуск
d_MinChnlTolerance:Single; //Минимальный допуск
канала
d_MaxChnlTolerance:Single; //Максимальный допуск
канала
d_State:byte;     //Состояние для транспаранта.
d_NrmVal:Single; //Значение нормы отклонения
d_MonVal:Single; //Значение контролируемого параметра
d_DevMonUnit:String[10]; //единица измерения
контролируемого параметра

//Отображение на графике норм
d_GrUse:byte;     //Признак построения графика нормы
d_GrNameGrArg:string[10]; //Имя аргумента на графике
d_GrNameGrArgUnit:string[10]; //Единица измерения
аргумента
d_GrArgLimDn:Single; //Левая граница аргумента для
отображения на графике
d_GrArgLimUp:Single; //Правая граница аргумента для
отображения на графике
d_GrArgValue:Single; //Текущее значение аргумента для
отображения точки
d_GrNameGrPar:string[10]; //Имя параметра на графике
d_GrNameGrParUnit:string[10]; //Единица измерения
параметра
d_gGrParLimDn:Single; //Нижняя граница параметра для
отображения на графике
d_GrParLimUp:Single; //Верхняя граница параметра для
отображения на графике
d_GrParValue:Single; //Текущее значение параметра для
отображения точки
d_GrXArr:array[1..20] of single; //Массив точек оси X
d_GrYArr:array[1..20] of single; //Массив точек оси Y

//Тренд-анализ (Прогноз)
d_TrUse:byte;     //Признак построения графика тренда
d_TrNameGrNar:string[10]; //Имя наработки на графике
d_TrNameGrNarUnit:string[10]; //Единица измерения
наработки
d_TrNameGrPar:string[10]; //Имя параметра на графике
d_TrNameGrParUnit:string[10]; //Единица измерения
параметра
d_TrGrLimDnX:Single; //Левая граница графика по оси
X
d_TrGrLimUpX:Single; //Правая граница графика по
оси X

```

```

        d_TrGrLimDnY:Single;      //Нижня граница графика по
оси Y
        d_TrGrLimUpY:Single;      //Верхня граница графика по
оси Y
        d_TrDevXArrStr:array[1..20] of single;    //Массив
точек оси X для ActiveX
        d_TrNumDevPoints:byte;    //Число точек графика допуска
        d_TrDevUpYArr:single;     //Верхний допуск отклонения
        d_TrDevDnYArr:single;     //Нижний допуск отклонения
        d_TrTrendXArr:array[1..20] of single; //Массив точек
тренда по оси X
        d_TrTrendYArr:array[1..20] of single; //Массив точек
тренда по оси Y

        //Линия прогноза
        d_TrNumTrendPoints:byte; //Число точек графика Тренд
        d_TrCastXArr:array[1..20] of single; //Массив точек
прогноза по оси X
        d_TrCastYArr:array[1..20] of single; //Массив точек
прогноза по оси Y
        d_TrCastUpYArr:array[1..20] of single;    //Массив
точек ВП прогноза по оси Y
        d_TrCastDnYArr:array[1..20] of single;    //Массив
точек НП прогноза по оси Y
        d_TrNumCastPoints: byte; //Количество точек для
отображения графика прогноза
    end;

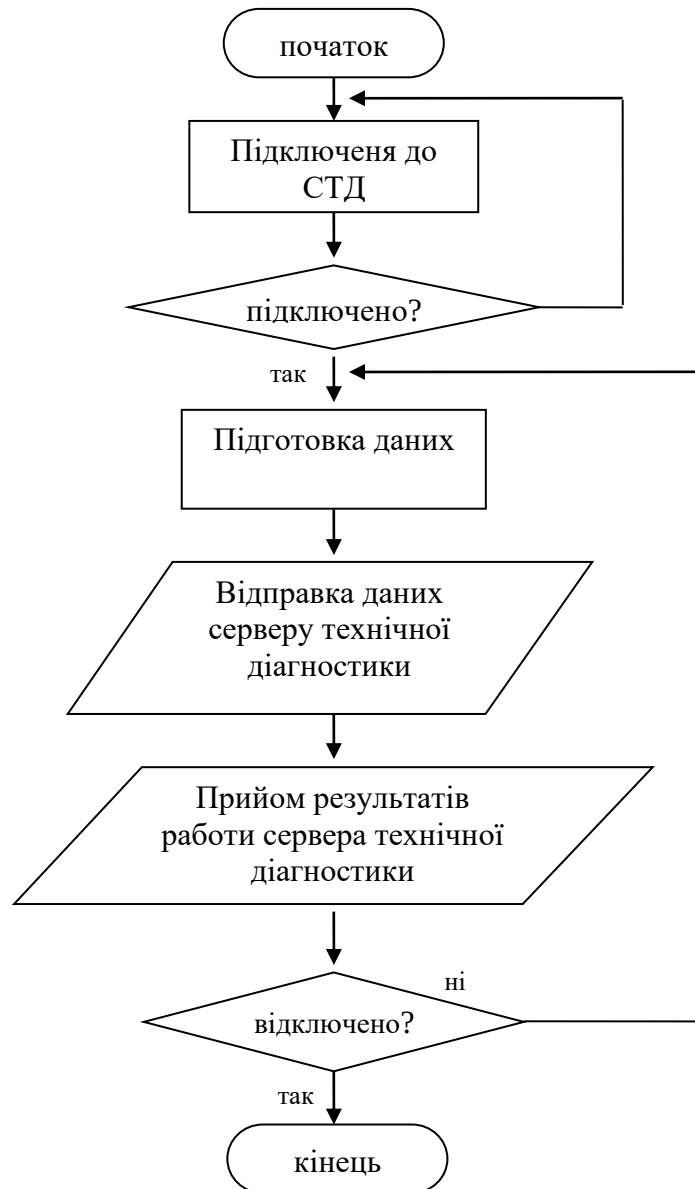
    TDDataGet=record
        Alarms:array[0..4] of Longword;
        Deviation:array[1..50] of TDeviation;
        DASignal:array[1..20] of Single;
        DDSignal:array[1..20] of byte;
    end;

```

Принцип роботи модуля передачі даних у систему технічної діагностики полягає в наступному:

- підключення до сервера технічної діагностики;
- підготовка даних для відправлення;
- відправлення даних;
- прийом результатів роботи сервера технічної діагностики;
- затримка 1 с., повтор вищевказаних дій до відключення.

Блок схема модуля передачі даних у систему технічної діагностики представлений на малюнку 2.14, лістинг представлений у додатку



А.

Рисунок 2.14 – Блок схема модулю передачі даних у систему технічної діагностики

Висновки до розділу 2

У розділі проведено системний аналіз сучасних підходів і технологій моніторингу технічного стану двигунів, що застосовуються у провідних промислових рішеннях. Розглянуті системи Mitsubishi RMS, Siemens Online Monitoring та GE RM&D демонструють високий рівень цифровізації та інтелектуалізації процесів оцінювання стану енерготехнічних установок,

орієнтуючись на безперервний контроль, віддалену діагностику та прогнозування відмов.

Проаналізовані функціональні можливості цих систем дозволили окреслити ключові принципи, які мають бути враховані при створенні інформаційно-управляючої системи моніторингу: модульність, масштабованість, підтримка трендового аналізу, інтеграція алгоритмів прогнозування та забезпечення оперативної реакції на критичні зміни технічних параметрів.

Уточнено призначення та завдання цільового модуля моніторингу, який має забезпечувати автоматизоване збирання, обробку та інтерпретацію діагностичних даних у режимі реального часу. Проведений техніко-економічний аналіз довів доцільність розроблення системи, здатної підвищити надійність експлуатації двигунів, мінімізувати ризики аварійних зупинок та оптимізувати витрати на технічне обслуговування.

Запропонована концепція майбутньої системи базується на принципах інтелектуального аналізу, адаптації до виробничих умов та застосуванні сучасних інформаційно-комунікаційних технологій. Постановка задачі визначає необхідність створення комплексного програмно-технічного рішення, що поєднуватиме функції вимірювання, діагностики, прогнозування та управління інформаційними потоками.

РОЗДІЛ 3. РОЗРОБКА ПРОЄКТУ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ ВІДДАЛЕНОГО МОНІТОРИНГУ ТА ДІАГНОСТИКИ ЕТУ

3.1 Алгоритм локальної схеми калібрування інформаційно-вимірювальної системи

Локальна схема калібрування виконується безпосередньо на робочому вузлі (локальному модулі збору даних), який встановлений у виробничих умовах поруч із двигуном. Мета алгоритму — забезпечити коректність, точність і адаптивність вимірювань у реальному часі, враховуючи температурні градієнти, вібраційні навантаження та старіння сенсорів.

1. Ініціалізація вимірювальної підсистеми

Система виконує запуск локального контролера.

Визначаються підключені сенсори: вібраційні, температурні, тиску, струму тощо.

Завантажуються останні збережені калібрувальні коефіцієнти.

Встановлюється базовий режим реального часу для збору даних.

2. Перевірка працездатності сенсорів

Система проводить самодіагностику: перевіряє інтегральні параметри сенсорів (шум, зсув нуля, стабільність).

Якщо знайдено аномалії (надмірний шум, пропуски у даних, нестабільність сигналу), модуль переходить у режим "повторна калібрування".

Якщо всі сенсори коректні — система переходить до калібрувальних вимірювань.

3. Збір контрольних даних для калібрування

На короткому часовому інтервалі система виконує опорні вимірювання за умов сталого режиму роботи двигуна.

Вимірюються базові параметри вібраційне прискорення, температура корпусу, частота обертання, тиск мастила.

Дані порівнюються з референсними значеннями (еталонні діапазони).

4. Обчислення калібрувальних коефіцієнтів виконується корекція нульового зміщення сигналу. Розраховуються поправки чутливості сенсорів залежно від температури та навантаження.

Для інтелектуальної корекції використовується машинне навчання, яке аналізує історичні дані, тренди відхилень, характеристики старіння сенсорів.

Формується набір коефіцієнтів компенсації для кожного каналу вимірювання.

5. Адаптивне узгодження параметрів. Алгоритм адаптації визначає, чи відповідають нові коефіцієнти трендовим моделям. Якщо зміни аномальні — система позначає сенсор як потенційно несправний. Якщо зміни коректні — калібрування приймається та фіксується локально.

6. Оновлення конфігурації та перехід у робочий режим. Оновлені параметри зберігаються у локальній базі даних. Модуль вимірювання перемикається в режим реального часу. Інтелектуальні моделі оцінювання технічного стану отримують оновлені дані. Система починає моніторинг із підвищеною точністю.

7. Передача даних до центральної системи. Локальний модуль надсилає звіт про проходження калібрування.

До центру моніторингу передаються нові коефіцієнти, оцінка достовірності сенсорів, виявлені аномалії, історія калібрування.

8. Постійна фонові перевірка якості вимірювань. У процесі роботи система продовжує оцінювати стабільність сигналів. Якщо виявляється деградація чи відхилення — запускається автоматичне локальне перекалібрування. Це дозволяє системі адаптуватися до змін у виробничих умовах.

Алгоритм локальної схеми калібрування, рисунок 3.1, забезпечує високу точність оцінювання технічного стану двигуна, адаптивність до змін навколишнього середовища, надійність і самокорекцію вимірювальної підсистеми, інтелектуальну обробку даних за допомогою алгоритмів машинного навчання, мінімізацію помилок вимірювань, спричинених старінням сенсорів чи вібраційними навантаженнями.

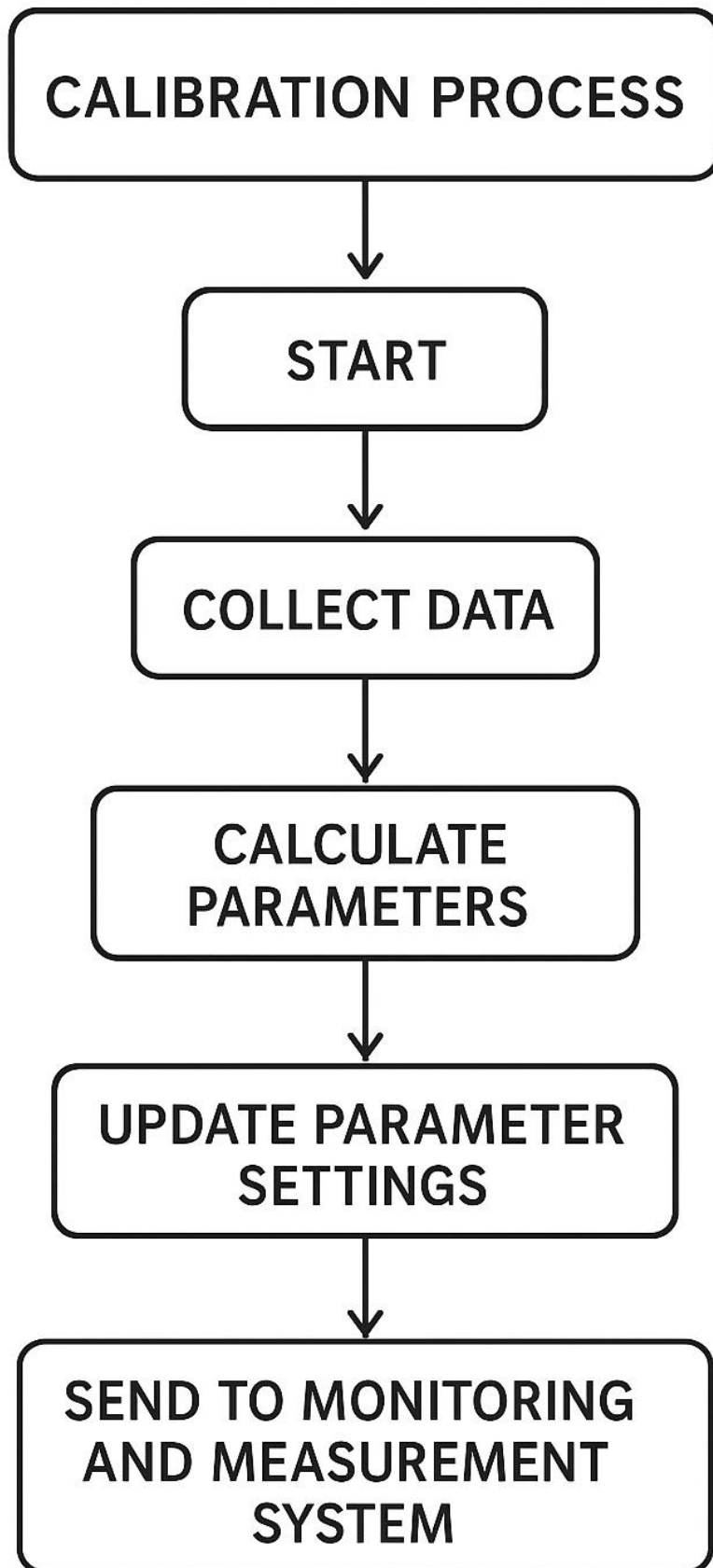


Рисунок 3.1 - Алгоритм локальної схеми калібрування

У ході виконання дипломного проекту були виконані наступні роботи:

- установка й налаштування системи керування базами даних MySQL;
- установка й налаштування розробленого програмного забезпечення;
- перевірка працездатності розробленого ПО разом із СУБД MySQL;

- перевірка працездатності розробленого ПО разом із програмно технічними засобами CCC “Series”;
- перевірка працездатності основних модулів програми.

3.2 Установка СУБД

Установка СУБД вироблялася з дистрибутива, для операційної системи Windows, із вказівкою директорії установки. Після установки, були виконані ліхословити MySQL, за допомогою утиліти ліхословити. На рисунках 3.1-3.8 представлені всі етапи налаштування.

Здійснено вибір режиму сервера MySQL, що визначає використання пам'яті й процесорного часу додатком. У цьому випадку був обраний тип режиму розроблювача, що приводить до мінімальних витрат ресурсів для роботи СУБД.

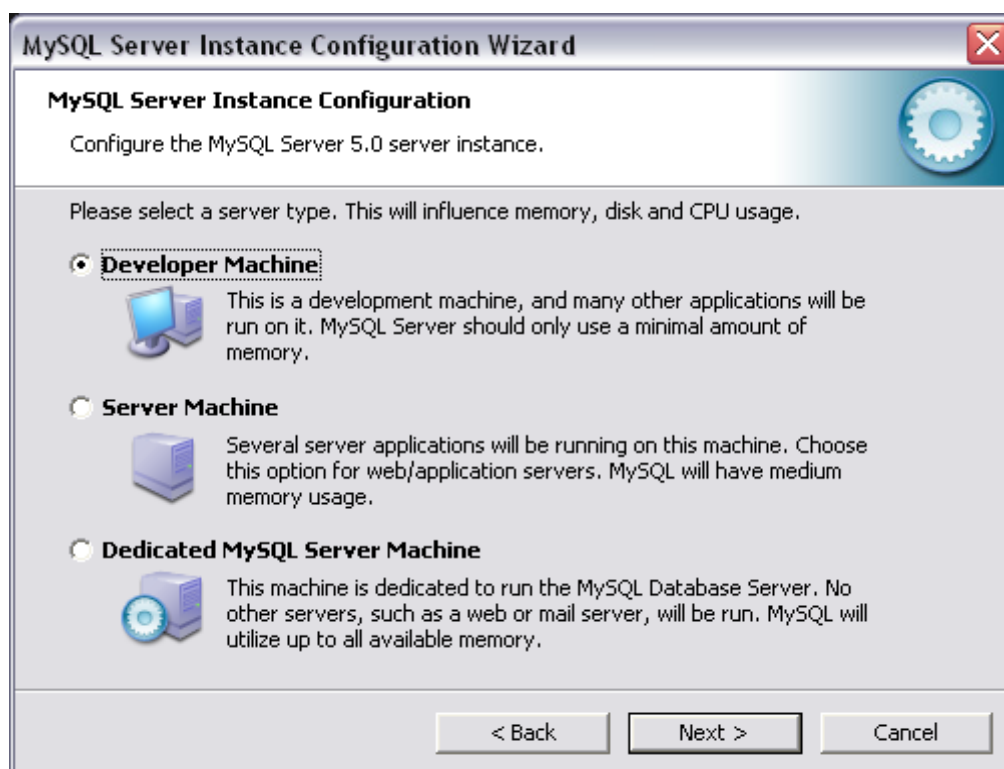


Рисунок 3.1 – Конфігурування MySQL

Вибір використання бази даних. Був здійснений вибір без транзакційних баз даних, при цьому активуються тільки таблиці MyISAM.

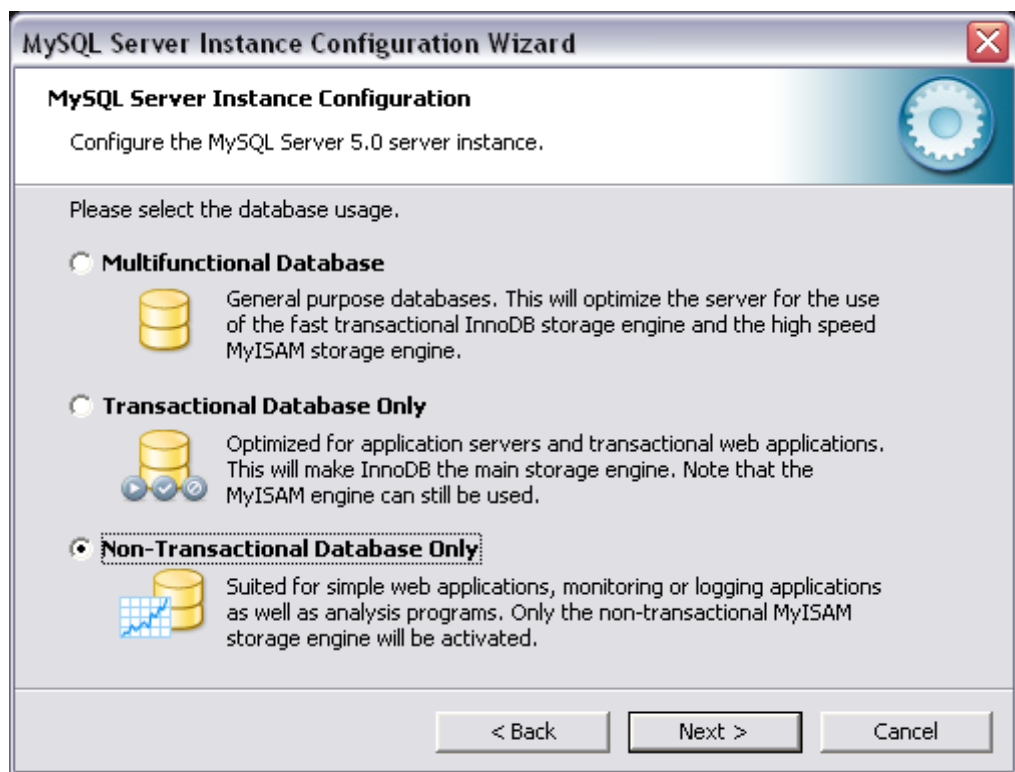


Рисунок 3.2 – Конфігурування MySQL

Кількість користувачів, що підключаються, обмежуємося 10 і встановлюємо це вручну.

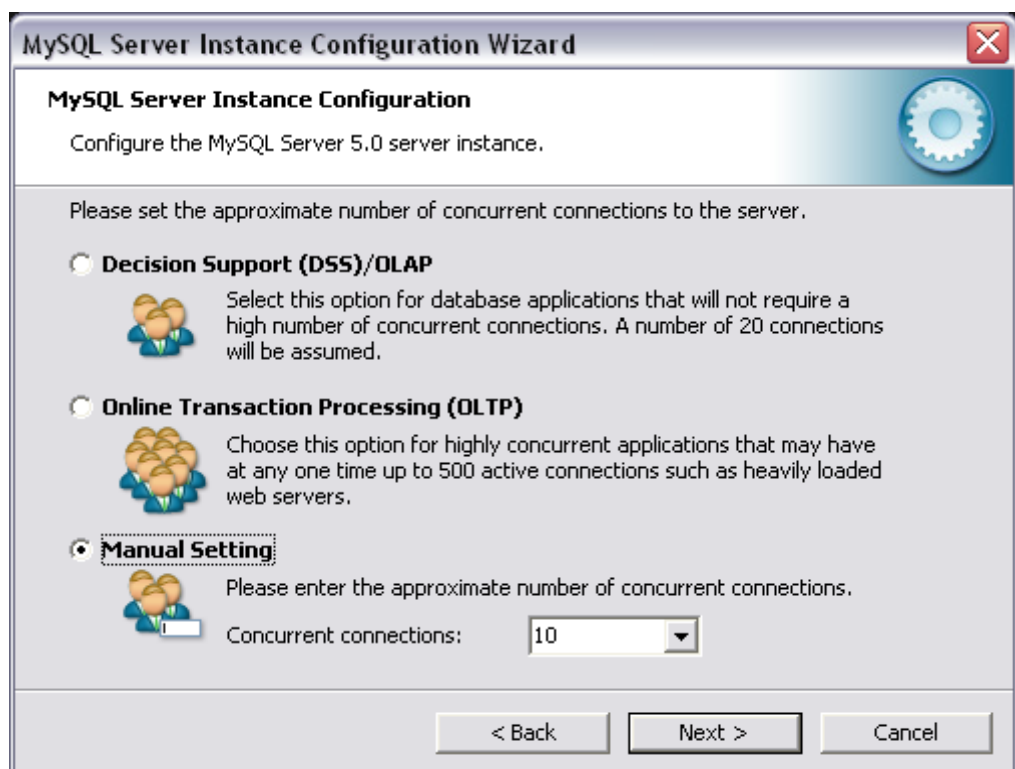


Рисунок 3.3 – Конфігурування MySQL

Налаштування мережі. Порт вибраний за замовчуванням, але якщо є необхідність, тобто можливість призначити інший порт, відповідні налаштування необхідно провести у програмному забезпеченні віддаленого моніторингу.

Одночасно встановлюємо виняток на вибраний порт, що унеможливило його блокування міжмережовим екраном.

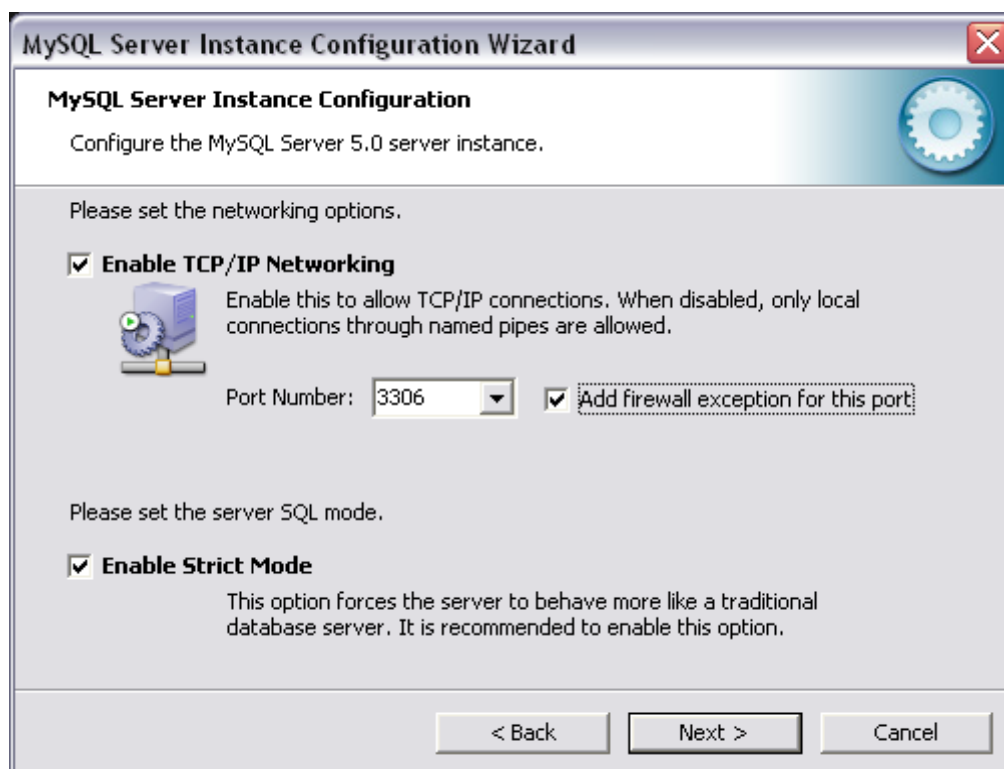


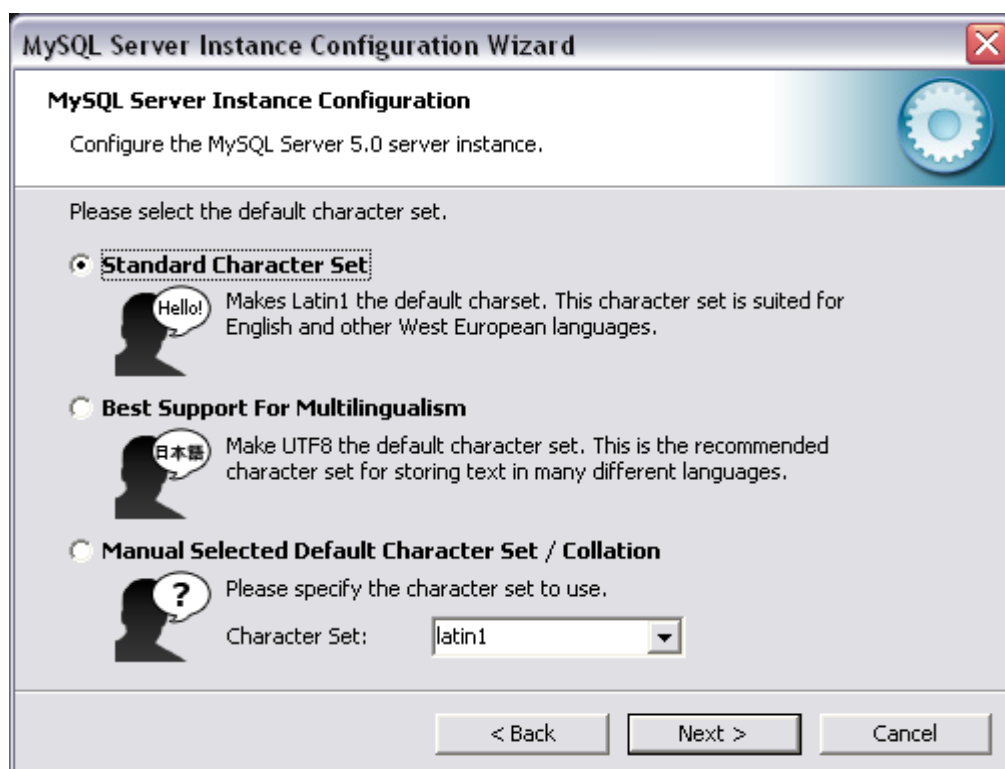
Рисунок 3.4 – Конфігурування MySQL

Кодування

символів

вибрано

за



замовчуванням.

Рисунок 3.5 – Конфігурування MySQL

Робота MySQL обрана як служба операційної системи, з автоматичним запуском під час завантаження системи.

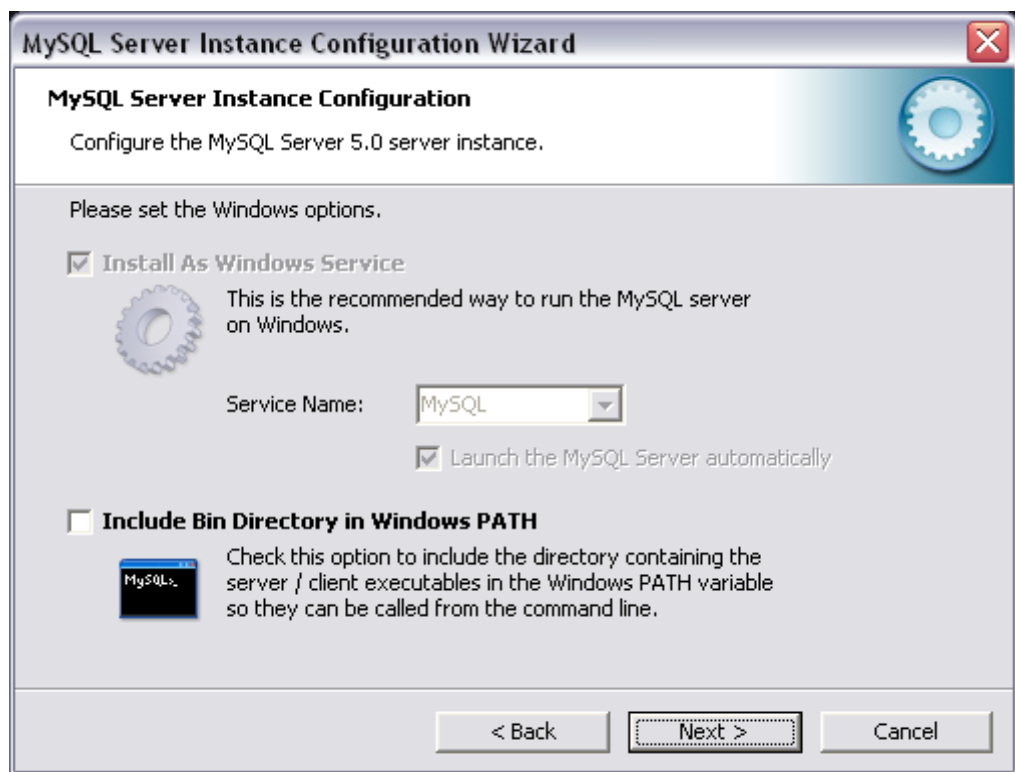


Рисунок 3.6 – Конфігурування MySQL

Налаштування безпеки. Були встановлені паролі адміністратора та можливість доступу з віддаленого комп'ютера..



Рисунок 3.7 – Конфігурування MySQL

Після завершення конфігурації проводиться збереження налаштувань і перезапуск сервісу, після чого СУБД готова до використання.

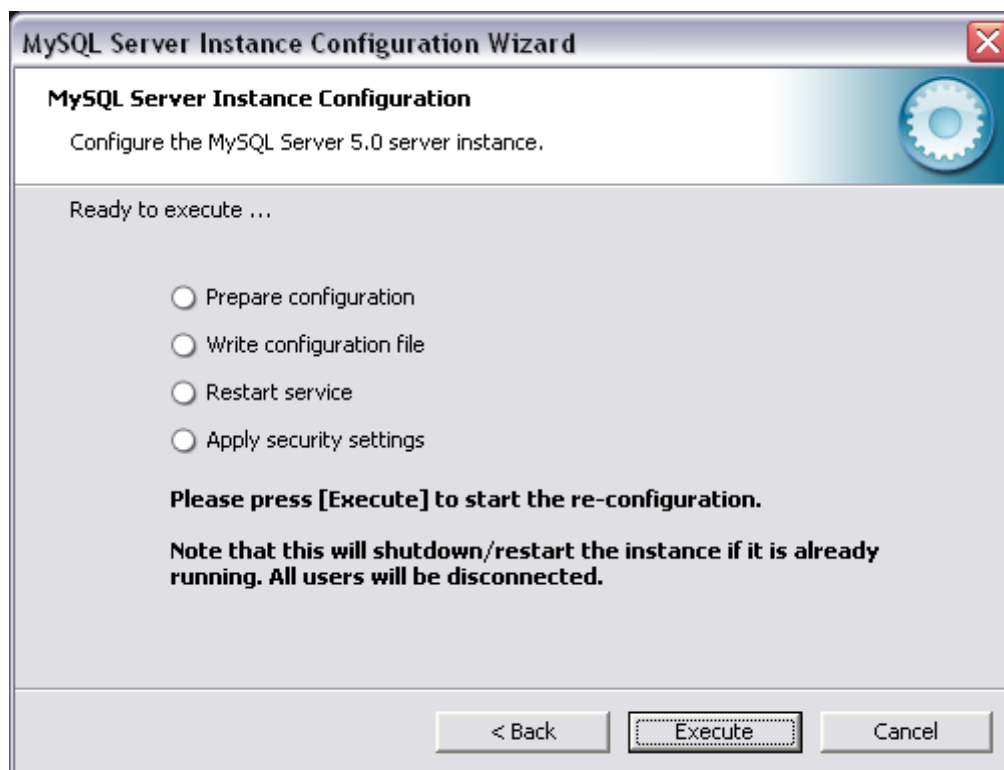


Рисунок 3.8 –Конфігурування MySQL

Для підключення програми вилученого моніторингу до СУБД MySQL необхідно внести наступні налаштування: номер IP адреси й порту системи керування базами даних, ім'я й пароля користувача. Після внесення даних програма має повний доступ до запитів SQL.

За замовчуванням програма вилученого моніторингу зберігає дані, отримані з контролера, цілодобово в директорії DataFiles, заархівовані (підготовлені для передачі) файли зберігаються в директорії ArchiveFiles.

Дане програмне забезпечення вилученого моніторингу було протестировано в іспитовому цеху 100 ГП НПКГ «Зоря» - «Машпроект». За результатами випробування можна зробити наступні висновки:

- для забезпечення автономності програмного продукту необхідно максимально забезпечити програмну обробку критичних помилок таких як: помилка підключення, обривши зв'язку, помилка запитів SQL, помилки введення висновку і т.і.
- для зменшення розміру переданого файлу, була розроблена наступна структура: файл розбитий на дві частини - заголовок і дані. Заголовок несе в собі інформацію про кожний сигнал - ім'я, множник, зсув. Всі дані речовинного типу, які займають б'ємо в 4 байти, представлені

целочисленними значеннями розміром в 2 байти. Перетворення таких значень виробляється за допомогою зазначених множників, зсувів. Дане перетворення дає можливість зменшити об'єм даних у середньому на 13 МВ.

- розмір добового архіву за результатами випробування склав: для архівного не стислого файлу 22.4 МБ, для архівного стислого файлу (файл який передається головному серверу зберігання даних) у середньому 3,74 МБ, об'єм даних зберігаючихся в таблицях (добова таблиця) склав 36,1 МБ. Ці показання дають нам можливість оцінювати конфігурації персональних комп'ютерів для зберігання даних, а також вибирати пропускну здатність каналів зв'язку, для усталеної роботи додатків.

У цілому в результаті тестування було виявлено працездатність всіх модулів програми, були дороблені деякі функціональні підпрограми, що дозволило поліпшити надійність, швидкодію програми, а також зменшити розмір вихідного коду шляхом правильного вибору типів даних і способів обробки інформації.

Отримані дані в результаті випробувань являють собою архів параметрів двигуна й результати роботи технічної діагностики, що зберігаються в базі даних MySQL, що дає можливість багатьом користувачам у зручній формі працювати з даними для перегляду й аналізу. Приклад використання даних представлений на рисунку 3.9.

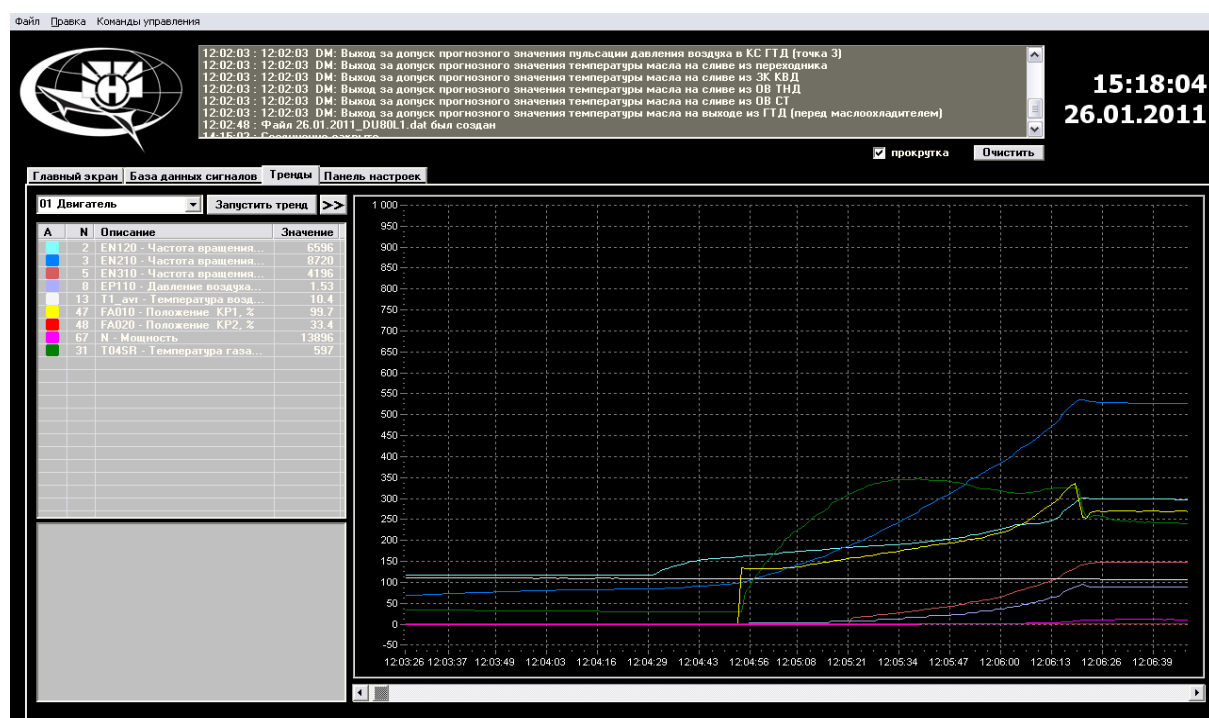


Рисунок 3.9 – Тренд запуска ЕТУ.

Висновки до розділу 3

У третьому розділі було здійснено розроблення проєкту інформаційно-управляючої системи віддаленого моніторингу й діагностики електротехнічних установок (ЕТУ), орієнтованої на роботу в промислових умовах та забезпечення високої точності вимірювальних процесів.

Опрацьований **алгоритм локальної схеми калібрування** дозволив формалізувати процедуру забезпечення достовірності вимірюваних параметрів, мінімізувати похибки сенсорних каналів та підвищити стабільність функціонування системи в умовах виробничих завад. Запропонована схема враховує відхилення фізичних перетворювачів, адаптується до режимів роботи обладнання та забезпечує автоматизоване оновлення калібрувальних коефіцієнтів у реальному часі.

Проведена **інсталяція та налаштування СУБД** створила основу для побудови централізованого сховища технологічних даних, оптимізованого для потокового надходження, обробки та архівації великих обсягів вимірювальної інформації. Обрана архітектура підтримує масштабованість, інтеграцію зі зовнішніми аналітичними модулями та формування історичних вибірок для подальшого застосування методів прогнозування стану ЕТУ.

У результаті виконання розділу сформовано цілісну технічну та алгоритмічну основу системи, що забезпечує її подальшу реалізацію, інтеграцію та впровадження у виробничі процеси.

РОЗДІЛ 4. РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ ВІД ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНО-УПРАВЛЯЮЧОЇ СИСТЕМИ МОНІТОРИНГУ

4.1 Загальна характеристика

На даний момент сучасні обчислювальні засоби широко використовуються в повсякденній діяльності людини. Сьогодні обчислювальна техніка дозволяє вирішувати найрізноманітніші задачі: від звільнення людини від монотонної ручної праці до складних технічних розрахунків, що не під силу виконати одній людині. Сучасні засоби програмування дозволяють вирішувати задачі, що стояли в недавньому минулому перед цілими колективами наукових співробітників.

Розвиток нових технологій і ріст застосування обчислювальної техніки привели до широкого впровадження ЕОМ для вирішення проблем, пов'язаних з обробкою інформації, її контролем і аналізом. Сьогодні на основі засобів обчислювальної техніки розробляються і впроваджуються різні автоматизовані інформаційні системи (управління, проектування, технологічної підготовки виробництва). Успіх у розробці цих систем, їхня роль в інтенсифікації розвитку народного господарства нашої країни багато в чому залежить від фахівців програмістів, які знають методику аналізу та проектування цих систем, можливості обчислювальної техніки і володіють математичними методами, що використовуються при постановці та вирішенні задач.

Даний дипломний проект присвячений розробці програмного забезпечення для якісного аналізу собівартості побудови корпусу судна, який дозволяє отримати якісну оцінку варіанта-сценарію та прийняти рішення щодо ефективності будівництва корпусу судна.

Найбільш важливим моментом для розробника, з економічної точки зору, є процес формування вартості системи. Очевидно, що вона являє собою дуже специфічний товар з безліччю особливостей.

Створення програмного забезпечення вимагає одноразових витрат на його розробку, придбання необхідних технічних засобів і поточних витрат на функціонування. Економія від функціонування програмного забезпечення визначається з урахуванням витрат на його експлуатацію. Відношення цієї економії до витрат на створення програмного забезпечення характеризує

економічну ефективність капітальних вкладень. Економічні показники визначаються по діючим на момент розрахунку оптовим цінам, тарифам і ставкам заробітної плати.

4.2 Розрахунок вартості створення програмного засобу

Витрати на розробку системи складаються з витрат на:

- заробітну платню розробника;
- амортизацію ЕОМ, на якій виконується розробка;
- експлуатацію цієї ЕОМ;
- засоби розробки;
- матеріали і комплектуючі.

Виходячи з того, що основна заробітна платня розробників програмного забезпечення складає 2000 грн./міс., вартість сучасної ПЕОМ складає 6000 грн. і вартість кіловат-години електроенергії складає 0,2436 грн., розрахуємо вартість розробки системи.

Розрахунок заробітної плати:

$$З_{зп} = ЗП_{роз} \cdot T_{роз} \quad (3.1)$$

де $ЗП_{роз}$ – зарплатня розробника, за місяць; $T_{роз}$ – тривалість розробки (дослідження, створення, налагодження і впровадження).

Для розробки даної системи необхідно 4 місяці (за експертною оцінкою часу на розробку аналогічних систем). Кількість розробників – 1 людина.

З цього випливає, що загальна сума витрат на заробітну платню складе:

$$З_{зп} = ЗП_{роз} \cdot T_{роз} = 2000 \cdot 4 = 8000 (\text{грн.})$$

Витрати на амортизацію ЕОМ, на якій виконується розробка, розраховується за формулою:

$$З_{амор} = Ц_{варт} \cdot A \cdot T_{роз} \quad (3.2)$$

де $Ц_{варт} = 6000$ грн. – балансова вартість ЕОМ; $A = 60\%$ - амортизація за рік; термін служби ЕОМ – 5 років; $T_{роз} = 0,33$ року – час, необхідний для розробки системи.

$$З_{амор} = 6000 \cdot 0,6 \cdot 0,33 = 1188 (\text{грн.})$$

Витрати на експлуатацію ЕОМ, на якій виконується розробка, полягають в оплаті споживаної нею електричної енергії і розраховуються по формулі:

$$З_{Амор} = P_{ЕОМ} \cdot T_{розр} \cdot N_{р\delta} \cdot N_{рч} \cdot E_{ел} \quad (3.3)$$

де $P_{ЕОМ} = 0,4$ кВт./год. – потужність ЕОМ; $T_{розр} = 4$ місяці – тривалість розробки; $N_{р\delta} = 22$ дні – число робочих днів у місяці; $N_{рч} = 8$ годин – число годин у робочому дні; $E_{ел} = 0,2436$ грн. – вартість 1 кВт/год електроенергії.

$$З_{Амор} = 0,4 \cdot 4 \cdot 22 \cdot 8 \cdot 0,2436 = 68,59 \text{ (грн.)}$$

Витрати на матеріали і комплектуючі вироби враховують витрати на папір для друкувальних пристроїв, на картридж і тонер для принтера, а також на непередбачені витрати. Розрахунок приведений у таблиці 1 (ціни взяті відповідно до прайс-листів та договірних цін на даний вид матеріалів).

Таблиця 3.1 – Розрахунок витрат на матеріали і комплектуючі вироби

№ з/п	Найменування виробів	Вартість, грн.
1	Папір для принтера	40,00
2	Картридж та тонер для принтера	250,00
3	Література (доступ до Internet)	100,00
4	Непередбачені витрати	60,00
Разом		450,00

Загальний кошторис витрат на створення системи з урахуванням вищевказаного представлений у таблиці 3.2.

Таблиця 3.2 – Загальний кошторис витрат на створення системи

№ з/п	Найменування витрат	Вартість, грн.
1	Заробітна платня основна	8 000,00
2	Заробітна плата додаткова (20% від п.1)	2 000,00
3	Відрахування на соціальне страхування (38% від пп. 1 та 2)	4560
4	Витрати на амортизацію ЕОМ	1 188,00
5	Витрати на експлуатацію ЕОМ	68,59
6	Витрати на матеріали і комплектуючі вироби	450,00
7	Адміністративні витрати (50% від основної заробітної платні)	4 000,00
Разом на створення системи		20 266,59

До витрат на впровадження системи можна віднести витрати на придбання технічного забезпечення, вартість програмного забезпечення, вартість навчання кадрів, витрати на монтаж и настроювання мережі.

Оскільки параметри технічних засобів, які вже є, відповідають вимогам, то їх вартість при розрахунку витрат враховувати не будемо.

Виходячи з вимог до програмного забезпечення, а також проаналізувавши цінову політику, можемо прийняти наступне (табл. 3.3).

Таблиця 3.3 – Перелік програмного забезпечення, необхідного для впровадження системи

№	Найменування ПО	Кількість	Вартість, USD	Вартість, грн.
1	MSWindowsXP	1	190	1520,00
Разом			1520,00	

Витрати на програмне забезпечення склали 1520 грн.

Згідно з досвідом створення аналогічних програм, приймаємо, що вартість підготовки кадрів дорівнює 300,00 грн. Витрати на супроводження програмного забезпечення дорівнюватимуть 1% від вартості програми и становлять 202,67 грн.

Інвестиційні витрати на впровадження системи з урахуванням вартості навчання кадрів і витрати на супровід представлені в таблиці 3.4.

Таблиця 3.4 – Інвестиційні витрати на впровадження системи

№	Найменування інвестицій	Вартість, грн.
1	Вартість устаткування	0
2	Вартість програмного забезпечення	1520
3	Вартість підготовки кадрів	300,00
5	Супроводження системи	202,67
Разом на впровадження системи		2022,67

Разом загальна сума витрат на створення і впровадження системи складає 22 289,26 грн.

4.3 Розрахунок економічної ефективності розробки і впровадження програмного продукту

Основним показником економічної ефективності функціонування програмного забезпечення є підвищення ефективності керування інформацією у вигляді зниження витрат на керування при одночасному збільшенні швидкості і якості одержання потрібного результату.

Економічна ефективність впровадження програмного засобу «Analysis» очікується за рахунок вивільнення робочого часу працівників та підвищення продуктивності праці. Крім того, не піддається прямій грошовій оцінці зменшення кількості помилкових та необережних рішень, підвищення оперативності керування, поліпшення організації роботи та своєчасне отримання необхідної інформації про пріоритетність варіантів рішень тощо.

Використання даної програми дозволяє вивільнити 0,5 робочого часу. Так як з системою працює один робітник, то система умовно вивільнить 0,5 робітника.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження системи вивільнить 0,5 робітника.

Зарплата 0,5 робітника у рік складає $\Delta C = 3000 \cdot 12 \cdot 0,5 = 18000$ (грн.).

Річний економічний ефект розраховується за формулою 3.4.

$$E_{\text{рік}} = \Delta C - C_{\text{супр}} - E_n \cdot k, (3.4)$$

де $C_{\text{супр}}$ - вартість супроводження системи; E_n – нормативний коефіцієнт економічної ефективності капітальних вкладень у галузь – для обчислювальної техніки приймається $= 0,5$; k - додаткові капітальні вкладення з урахуванням витрат на проектування, створення і функціонування системи.

$$E_{\text{рік}} = 18000 - 202,67 - 0,5 \cdot 22289,26 = 6652,7.$$

Строк окупності системи розраховується за формулою 3.5.

$$I = \frac{k}{\Delta C - C_{\text{супр}}}. \quad (3.5)$$

$$I = \frac{22289,26}{18000 - 202,67} = 1,25 (\text{року}).$$

Тобто, система окупиться через 1,25 року, що приблизно дорівнює 15 місяцям.

Висновки до розділу 4

У четвертому розділі було проведено комплексний економічний аналіз ефективності впровадження інформаційно-управляючої системи моніторингу технічного стану електротехнічних установок (ЕТУ). На основі зібраних вихідних даних сформовано узагальнену характеристику проєкту, яка включає оцінку трудових, матеріальних та експлуатаційних ресурсів, необхідних для створення і підтримки програмного забезпечення.

Виконано детальний розрахунок вартості розроблення програмного продукту, що охоплює витрати на заробітну плату розробників, супутні накладні витрати, амортизацію технічних засобів та інші статті фінансових затрат. Отримані результати дозволили визначити повну собівартість створення програмного забезпечення.

Проведений розрахунок економічної ефективності показав, що впровадження розробленої інформаційно-управляючої системи забезпечує суттєве зниження операційних витрат, пов'язаних з технічним обслуговуванням і діагностикою ЕТУ, підвищує точність діагностичних рішень та зменшує ризик аварійних простоїв. Розраховані показники рентабельності, строку окупності та економічного ефекту свідчать про доцільність і перспективність застосування запропонованої системи в реальних виробничих умовах.

Таким чином, результати розділу підтверджують економічну обґрунтованість створення та впровадження інформаційно-управляючої системи моніторингу, що забезпечує підвищення технологічної надійності, ефективності експлуатації обладнання та оптимізацію витрат підприємства.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ ПРИ РОБОТІ З ПК У ВІДДІЛІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Загальні принципи охорони праці

Згідно до ст.1 Закону "Про охорону праці", охорона праці є системою правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людей в процесі праці. Як правовий інститут охорона праці визначає цілий комплекс правових норм.

Ст.4 Закону "Про охорону праці" формулює принципи державної політики у сфері охорони праці. Ці принципи в силу вказаного їх визначення, слід кваліфікувати і як принципи інституту охорони праці, які підлягають застосуванню при відсутності відповідних конкретних правових норм, і як такі, що визначають напрямки подальшого розвитку інституту охорони праці, і обов'язки держави, які вона взяла на себе і має додержувати при здійсненні своїй політики у майбутньому. З урахуванням викладеного, принципи, зазначені ст. 4 Закону "Про охорону праці", мають певне регулятивне значення, названій статті закріплені такі принципи:

- пріоритет життя і здоров'я працівників відносно результатів виробничої діяльності підприємства, повна відповідальність власника за створення безпечних і нешкідливих умов праці;
- комплексне вирішення завдань охорони праці на засадах національних програм з цих питань, а також з урахуванням інших напрямків економічної і соціальної політики, досягнень в галузі науки, техніки охорони здоров'я;
- соціальний захист працівників, повне відшкодування шкоди особам потерпілим від нещасних випадків на виробництві та професійних захворювань;
- установлення єдиних нормативів з охорони праці для всіх підприємств незалежно від форм власності і видів їх діяльності;
- використання економічних методів управління охороною праці. проведення політики пільгового оподаткування, то сприяє створенню

безпечних та нешкідливих умов праці, участь держави в фінансуванні заходів з охорони праці;

- здійснення навчання населення, професійної підготовки та підвищення кваліфікації працівників з питань охорони праці;

- забезпечення координації діяльності державних органів, установ, організацій та об'єднань громадян, що вирішують різні проблеми охорони здоров'я, гігієни та безпеки праці, а також співробітництво.

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. Власник або уповноважений ним орган повинен впроваджувати сучасні засоби техніки безпеки, які запобігають виробничому травматизмові, і забезпечувати санітарно-гігієнічні умови, що запобігають виникненню професійних захворювань працівників.

На власника або уповноважений ним орган покладається систематичне проведення інструктажу (навчання) працівників з питань охорони праці, протипожежної охорони.

Охорона праці - один з центральних інститутів трудового права. Він має виключно практичне значення. Недодержання вимог охорони праці створює небезпеку для здоров'я і життя працівників. У свою чергу і ті, кого законодавець називає власником або уповноваженим ним органом, несуть сувору, у тому числі й кримінальну, відповідальність за порушення правил охорони праці.

5.3 Аналіз небезпечних і шкідливих факторів при роботі з ПК у відділі програмного забезпечення

З точки зору впливу небезпечних і шкідливих факторів при роботі з ПК у відділі програмного забезпечення можна виділити наступні:

- 1) вплив електромагнітного та електростатичного випромінювань;
- 2) вплив недостатньої освітленості робочого місця;
- 3) вплив електричного струму;
- 4) вплив шуму;
- 5) вплив несприятливих умов мікроклімату;

б) вплив нераціонального розташування устаткування і неправильної організації робочого місця.

Тривале перебування людини в зоні комбінованого впливу різних несприятливих факторів може привести до професійного захворювання. Розглянемо більш детально вплив означених факторів на користувача ПК.

Вплив електромагнітного та електростатичного випромінювань.

При роботі ПК виникають два типи випромінювань: електростатичне та електромагнітне. Навколо електростатично зарядженого монітора підвищується концентрація пилу. Електромагнітне випромінювання створюється магнітними котушками, що знаходяться біля цокольної частини ЕПТ. Результати медичних досліджень показують, що електризований пил викликає запалення шкіри, а електромагнітне випромінювання приводить до порушення біологічних процесів (порушення сну, зниження кров'яного тиску, уповільнення серцевої діяльності, зміни складу крові).

Вплив недостатньої освітленості робочого місця.

Напружена зорова робота внаслідок нераціонального освітлення може бути причиною функціональних порушень у зоровому аналізаторі і привести до розладу зору, а у важких випадках – і до повної втрати. Втома органів зору залежить від ступеня напруженості процесів, що супроводжують зорове сприйняття. При роботі оператора ПК одним з основних навантажень на організм є навантаження на зір, тому дуже важливо приділити особливу увагу освітленості робочого місця.

У залежності від джерела світла виробниче висвітлення може бути трьох видів: природним, штучним і сполученим. При роботі з ПК вдень у ясну погоду використовується природне освітлення, яке здійснюється через світлові віконні прорізи. При нестачі природного світла, а також при роботі у вечірній час використовується штучне освітлення, джерелами якого є газорозрядні лампи. При такому освітленні величезне значення має правильне розташування джерел світла, їхній тип і кількість.

Правильна освітленість робочого місця сприяє високої працездатності користувача ПК і зводить до мінімуму шкідливий вплив на органи зору. Освітленість робочого місця, що рекомендується для користувача ПК складає 400–750 лк.

Вплив електричного струму.

Робота комп'ютерів і периферійних пристроїв можлива завдяки електричному струмові, до джерел якого вони підключаються, тому існує небезпека ураження користувача струмом. Щоб уникнути нещасних випадків варто строго дотримуватися правил електробезпечності.

Небезпека електричного струму на відміну від інших небезпек збільшується тим, що людина не в змозі без спеціальних приладів виявити напругу дистанційно, а також швидкоплинністю поразки – небезпека виявляється, коли людина вже уражена. Ураження струмом приводить до різних порушень в організмі, викликаючи як місцеву поразку тканин і органів, так і загальне ураження організму. Людина відчуває дратівну дію перемінного струму промислової частоти силою 0,6–1,5 мА і постійного струму 5–7 мА. Ці струми не представляють серйозної небезпеки для організму людини, тому що при такій силі струму можливе самостійне звільнення людини від контакту зі струмоведучими частинами. Для перемінного струму промислової частоти сила невідпускаючого струму знаходиться в межах 6–20 мА. Постійний струм не викликає невідпускаючого ефекту, але приводить до сильних болючих відчуттів, сила такого струму 15–80 мА і більше.

Вплив шуму.

Шумом прийнято називати всякий небажаний для людини звук, що заважає сприйняттю корисних сигналів. Шум являє собою безладне сполучення звуків різної інтенсивності і частоти. Він викликає труднощі з розпізнанням кольірних сигналів, знижує швидкість сприйняття кольору, гостроту зору, зорову адаптацію, порушує сприйняття візуальної інформації, зменшує на 5–12% продуктивність праці. Тривалий вплив шуму з рівнем звукового тиску 90 дБ знижує продуктивність праці на 30–60%.

Робота ПК може супроводжуватися шумами, викликаними деякими рухливими пристроями усередині комп'ютера, а також підключеними до нього. До таких пристроїв можна віднести вентилятори, пристрої читання дисків, принтери. Виникаючий шум, у залежності від інтенсивності, може привести до підвищення нервової напруги і зниження працездатності.

Вплив несприятливих умов мікроклімату.

Мікрокліматичні параметри (температура, відносна вологість і швидкість руху повітря) впливають на самопочуття і здоров'я людини, а також на надійність роботи засобів обчислювальної техніки.

При підвищеній температурі і вологості в приміщенні відбувається перегрівання тіла, що погіршує самопочуття і працездатність. Варто враховувати, що ПК, які знаходяться в приміщенні, сприяють збільшенню температури.

Для комфортної роботи приймається температура $23 (\pm 2)^{\circ}\text{C}$ й $18 (\pm 2)^{\circ}\text{C}$ в теплий і холодний час року відповідно, при відносній вологості $55 (\pm 5)\%$.

Вплив нераціонального розташування устаткування і неправильної організації робочого місця.

Робоче місце – це частина простору, у якій користувач ПК здійснює трудову діяльність і проводить велику частину робочого часу. Важливу роль грає планування робочого місця, яке повинне задовольняти вимогам зручності виконання робіт і економії енергії і часу користувача, зручності обслуговувань ЕОМ, дотримання правил охорони праці.

При плануванні робочого місця необхідно враховувати зони досяжності при розташуванні моніторів, клавіатури, маніпулятора «миша». Ці зони встановлюються на підставі антропологічних даних людського тіла. Під час роботи користувач ПК не повинен піддаватися впливові сторони подразників, якими можуть бути неадекватне фарбування пристроїв ЕОМ або приміщення.

5.4 Розрахунок системи загального освітлення приміщення відділу програмного забезпечення

В приміщенні відділу програмного забезпечення необхідно організувати змішане освітлення (комбінування природного і штучного освітлення). Природне освітлення створюється бічним освітленням через вікно. Штучне освітлення використовується при недостатньому природному освітленні. У даному приміщенні використовується загальне штучне освітлення. Розрахунок виконаємо за методом світлового потоку з урахування потоку, віддзеркаленого від стін та стелі.

Необхідно зробити розрахунок загального освітлення виробничої приміщення. Довжина $A = 40$ м, ширина $B = 16$ м, висота $H = 4,5$ м. Висота робочої поверхні $h_p = 1$ м. Для освітлення використовується світильні УТІМ-5. Мінімальна освітленість лампи наколювання по нормах $E_{\min} = 100$ лк. Коефіцієнт відображення стелі $S_n = 50\%$, стін $S_c = 30\%$, робочої поверхні $S_p = 10\%$. Напруга мережі 220 В.

Відстань від стелі до робочої поверхні:

$$H_0 = H - h_p = 4,5 - 1 = 3,5 \text{ м.}$$

Відстань від стелі до світильника:

$$h_c = 0,2 H_0 = 0,2 * 3,5 = 0,7 \text{ м.}$$

Висота підвісу світильника над освітленою поверхнею:

$$h_c = 0,2 H_0 = 0,2 * 3,5 = 0,7 \text{ м.}$$

Висота підвісу світильника над підлогою:

$$H_n = h + h_p = 2,8 + 1 = 3,8 \text{ м.}$$

Для досягнення найбільшої рівномірності освітлення приймаємо відношення $L/h = 1,5$.

Відстань між центрами світильників:

$$L = 1,5h = 1,5 * 2,8 = 4,2 \text{ м.}$$

Необхідна кількість світильників:

$$N = S / L^2 = (40 * 16) / (4,2)^2 \approx 36.$$

Приймаємо кількість світильників: 36 (3 ряди по 12 світильників).

Індекс приміщення:

$$i = (A B) / (h (A + B)) = (40 * 16) / (2,8 (40 + 16)) = 4,08.$$

Коефіцієнт використання світлового потоку $\eta = 0,66$.

Світловий потік однієї лампи:

$$\Phi = (E_{\min} S K_z Z) / (N \eta) = (100 * 640 * 1,3 * 1,15) / (36 * 0,66) = 4027 \text{ лм.}$$

Вибираємо (у таблиці) лампу, що має світловий потік 4610 лм:

$$\Gamma 215-225-300.$$

Фактична освітленість:

$$E_{\phi} = E_{\min} \Phi_{\phi} / \Phi_p = 100 * 4610 / 4027 = 114 \text{ лк.}$$

Загальна потужність:

$$P_{\text{общ}} = P_{\phi} N = 300 * 36 = 10800 \text{ Вт} = 10,8 \text{ кВт.}$$

Таким чином, для оптимальної роботи в приміщенні необхідно встановити 3 ряди по 12 світильників типу Г215-225-300. Загальна потужність усіх світильників складе 10,8 кВт.

5.2 Розробка заходів щодо зменшення впливу небезпечних і шкідливих факторів

Оскільки найбільшу загрозу при роботі на ЕОМ встановлює електростатичне та електромагнітне випромінювання. Заходи щодо зниження впливу. Для зниження величини виникаючих зарядів статичної електрики покриття технологічних підлог варто виконувати з одношарового полівінілхлоридного антистатичного лінолеуму. Іншим методом захисту є нейтралізація заряду статичної електрики іонізованим газом. У промисловості широко застосовуються радіоактивні нейтралізатори. До загальних мір захисту від статичної електрики можна віднести загальне і місцеве зволоження повітря.

Окрім цього необхідно розполагати монітори ПК таким чином, щоб електромагнітне випромінювання від їх електронно-променевих трубок не було направлено на людей. Також слід використовувати захисні екрани, що зменшують електростатичне випромінювання моніторів. Крім цього, на сьогоднішній день існують суміші які при обробці ним екранів також зменшують електростатичне поле, що направлено безпосередньо на людину. Крім того необхідно дотримуватися загальної техніки безпеки при користуванні електричною технікою.

Висновки до розділу 5

У п'ятому розділі було проведено системний аналіз умов праці працівників відділу програмного забезпечення та визначено основні небезпечні й шкідливі виробничі фактори, характерні для роботи з персональним комп'ютером. На основі чинних нормативно-правових документів охорони праці сформульовано комплекс вимог щодо організації безпечного робочого середовища та забезпечення оптимальних ергономічних параметрів робочих місць.

Проаналізовано потенційні ризики, пов'язані з тривалим статичним навантаженням, впливом електромагнітного випромінювання, нераціональним мікрокліматом, недостатнім рівнем освітленості, а також психофізіологічним

перевантаженням працівників. Виконаний розрахунок загального освітлення дозволив визначити необхідні світлотехнічні параметри для забезпечення нормативного рівня зорового комфорту та зменшення втоми операторів.

Розроблений комплекс профілактичних заходів — оптимізація робочих місць, нормування режимів праці та відпочинку, впровадження засобів захисту, покращення вентиляції та освітлення — спрямований на мінімізацію ризиків і підвищення загального рівня безпеки праці. Запропоновані рекомендації забезпечують створення здорових і продуктивних умов роботи для спеціалістів програмного забезпечення, що відповідає сучасним вимогам охорони праці та ергономіки.

Результати розділу підтверджують, що дотримання відповідних норм та впровадження превентивних заходів дозволяє значно знизити вплив небезпечних факторів, підвищити ефективність діяльності працівників та забезпечити високий рівень виробничої безпеки в умовах інтенсивної роботи з ПК.

6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вступ

Охорони навколишнього середовища, раціональне використання природних ресурсів, забезпечення екологічної безпеки життєдіяльності людини - невід'ємна умова сталого економічного та соціального розвитку країни. З цією метою країна здійснює на своїй території екологічну політику, спрямовану на збереження безпечного для існування живої і неживої природи навколишнього середовища, захисту життя і здоров'я населення від негативного впливу, зумовленої забрудненням навколишнього природного середовища, досягнення гармонійної взаємодії суспільства і природи, охорону, раціональне використання і відтворення природних ресурсів.

Завданням законодавства про охорону навколишнього природного середовища є регулювання відносин у галузі охорони, використання і відтворення природних ресурсів, забезпечення екологічної безпеки, запобігання і ліквідації негативного впливу господарської та іншої діяльності на навколишнє природне середовище, збереження природних ресурсів, генетичного фонду живої природи, ландшафтів та інших природних комплексів, унікальних територій та природних об'єктів, пов'язаних з історико-культурною спадщиною.

Основними принципами охорони навколишнього природного середовища є:

- дотримання права людини на сприятливе навколишнє середовище;
- забезпечення сприятливих умов життєдіяльності людини;
- науково обґрунтоване поєднання екологічних, економічних і соціальних інтересів людини, суспільства і держави з метою забезпечення сталого розвитку і сприятливого навколишнього середовища;
- охорона, відтворення та раціональне використання природних ресурсів як необхідні умови забезпечення сприятливого навколишнього середовища та екологічної безпеки;
- відповідальність органів державної влади України, органів державної влади суб'єктів України, органів місцевого самоврядування за забезпечення сприятливого навколишнього середовища та екологічної безпеки на відповідних територіях;

- платність природокористування і відшкодування шкоди навколишньому середовищу;
- незалежність контролю в області охорони навколишнього середовища;
- забезпечення наявності екологічної безпеки запланованої господарської та іншої діяльності;
- обов'язковість оцінки впливу на навколишнє середовище при прийнятті рішень про здійснення господарської та іншої діяльності;
- обов'язковість проведення державної екологічної експертизи проектів та іншої документації, що обґрунтовують господарську та іншу діяльність, яка може мати негативний вплив на навколишнє середовище, створити загрозу життю, здоров'ю та майну громадян;
- облік природних і соціально - економічних особливостей територій при плануванні та здійсненні господарської та іншої діяльності;
- пріоритет збереження природних екологічних систем, природних ландшафтів і природних комплексів;
- допустимість впливу господарської та іншої діяльності на природне середовище, виходячи з вимог в області охорони навколишнього середовища;
- забезпечення зниження негативного впливу господарської та іншої діяльності на навколишнє середовище відповідно до нормативів в області охорони навколишнього середовища, якого можна досягти на основі використання найкращих існуючих технологій з урахуванням економічних і соціальних чинників;
- обов'язковість участі в діяльності з охорони навколишнього середовища органів державної влади України, органів державної влади суб'єктів України, органів місцевого самоврядування, громадських та інших некомерційних об'єднань, юридичних і фізичних осіб;
- збереження біологічного різноманіття;
- забезпечення інтегрованого та індивідуального підходів до встановлення вимог в області охорони навколишнього середовища до суб'єктів господарської та іншої діяльності, що здійснюють таку діяльність або планувальним здійснення такої діяльності;

- заборона господарської та іншої діяльності, наслідки впливу якої, непередбачувані для навколишнього середовища, а також реалізації проектів, які можуть призвести до деградації природних екологічних систем, зміни та (або) знищення генетичного фонду рослин, тварин та інших організмів, виснаження природних ресурсів і іншим негативних змін навколишнього середовища;
- дотримання права кожного на отримання достовірної інформації про стан навколишнього середовища, а також участь громадян у прийнятті рішень, що стосуються їх прав на сприятливе навколишнє середовище, відповідно до законодавства;
- відповідальність за порушення законодавства в галузі охорони навколишнього середовища;
- організація та розвиток системи екологічної освіти, виховання і формування екологічної культури;
- участь громадян, громадських та інших некомерційних об'єднань у вирішенні завдань охорони навколишнього середовища;
- міжнародне співробітництво України в області охорони навколишнього середовища.

6.2 Забруднення навколишнього середовища суднобудівними заводами

Встановлено, що одним з найпотужніших ядер забруднення є суднобудівні та судноремонтні заводи. Найбільш це забруднення стосується прісних то морських вод.

В Україні суднобудівні заводи спричиняють найбільшу загрозу для навколишнього середовища у таких містах, як Севастополь, Миколаїв та Херсон.

У Миколаєві суднобудівні та судноремонтні підприємства розташовані безпосередньо в Бузькому лимані. Сюди ж скидаються стічні води міста та підприємств, розташованих вздовж узбережжя. Вода Бузького лиману має стійке забруднення від Миколаєва аж до впадання в Чорне море. Концентрації нафтопродуктів та органічних сполук перевищують допустимі норми в багато разів. Для Бузького лиману характерні згінно-нагінні явища, які ускладнюють екологічну ситуацію в гирлі Південного Бугу.

Справжнім лихом для морського середовища в районі Севастополя є відпрацьовані судна. Сотні великих та малих посудин кинуті в прибережних водах Севастопольської, Південної та Балаклавської бухт. Друге лихо – доволі висока забрудненість моря навіть в районі міських пляжів, явища, які ускладнюють екологічну ситуацію в гирлі Південного Бугу.

У Херсоні суднобудівні та судноремонтні підприємства розташовані в Кошовій протоці, гідравлічне зв'язаній з Дніпром. Вода протоки покрита товстим шаром нафтової плівки. В протоку скидаються міські стічні води. Мають місце непоодинокі випуски виробничих стічних вод. Вода в нижній течії Дніпра забруднена багатьма токсичними речовинами, вміст яких перевищує гранично допустимі концентрації.

6.3 Заходи щодо зменшення забруднення

Згідно з Постановою Кабінету Міністрів від 29 лютого 1996 р. №269 «Про і затвердження правил охорони внутрішніх морських вод і територіального моря забруднення та засмічення» суднобудівні і судноремонтні заводи повинні:

- бути укомплектовані суднами-збирачами, спеціальними засобами для локалізації та ліквідації наслідків надзвичайних ситуацій;
- мати берегові приймальні очисні споруди, систему каналізації, об'єкти для збирання та знешкодження відходів;
- мати засоби для накопичення зворотних вод з подальшою передачею їх на очисні споруди у разі відсутності приймальних очисних споруд та системи каналізації.

Усі операції з забруднюючими речовинами, водами, що їх містять, та сміттям, які проводяться на приймальних очисних спорудах та об'єктах поводження з відходами у портах, на судноремонтних та суднобудівних заводах, підлягають обов'язковій реєстрації в установленому порядку. Підприємства морської галузі узгоджують технологію перевантаження щодо кожного виду вантажів з державними інспекціями охорони Чорного та Азовського морів Мінприроди.

У разі порушення вимог природоохоронного законодавства України діяльність об'єктів може бути обмежена, тимчасово заборонена (зупинена) чи припинена в установленому порядку.

Державний контроль за охороною внутрішніх морських вод і територіального моря України від забруднення здійснюється у порядку, встановленому Мінприроди.

6.4 Аналіз ризиків ПЗ методом побудови дерева подій

На сучасному етапі розвитку людства, коли людина майже повністю опанувала природу, виявила закономірності розвитку зірок і сягнула глибин мирового океану проблема надзвичайних ситуацій стає все більш актуальною. Із зростанням потужностей техніки зростає і кількість людей що отримали ушкодження або і загинули через аварію або недотримання техніки безпеки(відомо, що наприклад від наслідків аварій на дорогах за рік страждає більше людей ніж від нападів злочинців). І тому зрозуміло, що процес передбачення і запобігання надзвичайної ситуації коштує набагато дешевше ніж усунування її наслідків. Окрім того запобігання ЧС дає змогу зберегти життя та здоров'я людей, що є безцінним.

Для того, щоб оцінити ризик виникнення надзвичайної ситуації користуються багатьма методами. В інформаційних системах нам треба перш за все оцінити надійність системи, тобто здатність її працювати без відмов. А також ймовірність відмови. При виконанні аналізу ризиків доводиться розв'язувати ряд складних проблем, наприклад оцінка коштовності ресурсів і побудова списку загроз. Процес цей є креативним так як ризики у кожному випадку різні і не піддаються формальному аналізу.

Процес аналізу ризиків поділяється на декілька етапів:

- ідентифікація інформаційних ресурсів;
- вибір критеріїв оцінки і визначення потенційного негативного впливу на ресурси і програмами;
- оцінка загроз;
- оцінка вразливостей;
- оцінка ризиків;
- оцінка ефективності існуючих засобів забезпечення інформаційної безпеки

На основі аналізу ризиків обираються засоби, що забезпечують режим безпеки. При оцінці ризиків враховується потенційний негативний вплив від

загроз, ризик характеризує загрозу якої може бути повержена система або організація що її використовує

Ступінь ризику залежить від ряду факторів:

- цінності ресурсів; ймовірності реалізації загроз;
- простоти використання вразливості для реалізації загроз;
- існуючих чи планованих к застосуванню засобів забезпечення безпеки,

що зменшують число вразливосте, ймовірність виникнення загроз та можливість негативних впливів.

Одним з методів аналізу ймовірностей ризиків є метод побудови дерева подій. Метод побудови дерева подій полягає у переборі усіх варіантів виникнення надзвичайної ситуації і оцінці ймовірності її виникнення.

Ці оцінки ґрунтуються на методах теорії ймовірності. Насамперед в них використовують ся поняття незалежних та несумісних подій. Суми та добутку подій. Події називаються несумісними якщо здійснення однієї події виключає виникнення останніх, події називаються незалежними якщо ймовірність добутку подій дорівнює добутку ймовірностей подій.

Як відомо події, що виходять з одного листа дерева подій є незалежними. Тобто ймовірність виникнення надзвичайної ситуації вершини дерева дорівнює сумі подій. А ймовірність надзвичайної ситуації проходу по шляху дерева добутку подій – цін кожної вершини шляху. На малюнку 5.1 представлене дерево відмов, поряд з кожною вершиною дерева подана ймовірність виникнення відмови через певну подію. У таблиці 5.2 подане дерево подій, що обраховує ймовірність виникнення відмови через похибку у модулі даних.

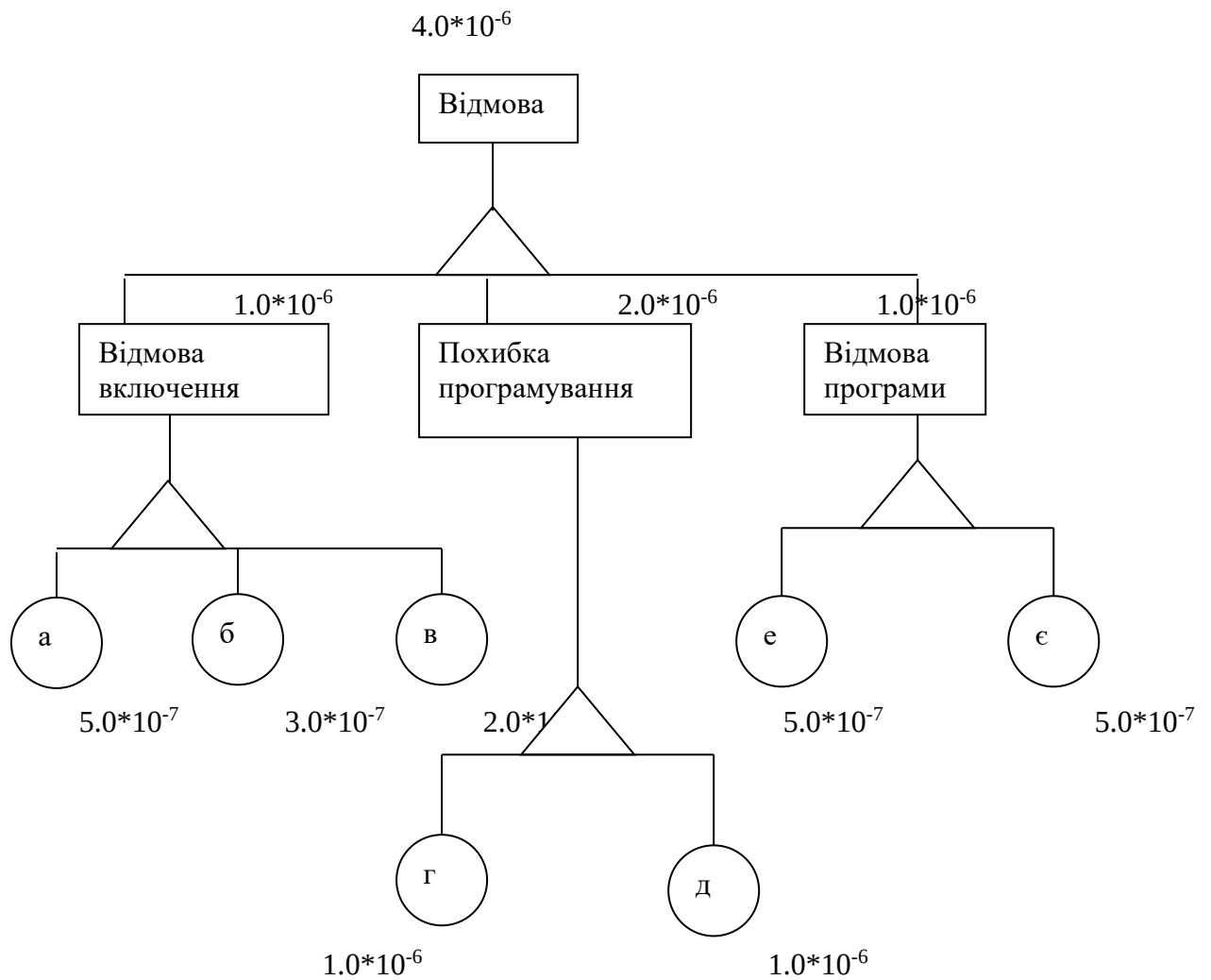


Рисунок 6.1 – Дерево відмов

Були виділені такі листові події:

- а) поломка кнопки включення;
- б) поломка мережі електроструму;
- в) відключення електромережі з технічних причин;
- г) похибка при роботі з бібліотеками
- д) похибка в основному коді;
- е) неможливість запуску;
- є) «зависання» програмного продукту.

Побудуємо дерево подій (табл. 6.1).

Таблиця 6.1 – Дерево подій

Похибка у основному коді	Похибка у модулях	Похибка програмування
$P(A)=5.0 \cdot 10^{-7}$	$P(A1)=1.0 \cdot 10^{-6}$	$P(A2)=2.0 \cdot 10^{-6}$
		$P(B2)=8.0 \cdot 10^{-6}$
	$P(B1)=9.0 \cdot 10^{-6}$	$P(A2)=2.0 \cdot 10^{-6}$
		$P(B2)=8.0 \cdot 10^{-6}$

Обчислимо ймовірність відмови через похибку в основному коді, користуючись формулами ймовірностей суми і добутку подій

$$P(\text{відказу}) = P(A) \cdot P(A1) \cdot P(A2) + P(A) \cdot P(B1) \cdot P(A2) = 5.0 \cdot 10^{-7} \cdot 1.0 \cdot 10^{-6} \cdot 2.0 \cdot 10^{-6} + 5.0 \cdot 10^{-7} \cdot 9.0 \cdot 10^{-6} \cdot 2.0 \cdot 10^{-6} = 0.10 \cdot 10^{-16}$$

По обчисленим даним можна зробити висновок, що ймовірність відмови значно менше верхньої припустимої межі ймовірності відмови.

Тому в межах припустимої похибки ми вважаємо систему безвідмовною, тобто ймовірність відказу щезаючи мала.

Висновки до розділу 6

У результаті проведених досліджень встановлено, що діяльність суднобудівних підприємств є суттєвим джерелом антропогенного впливу на довкілля, зокрема у частині викидів забруднюючих речовин, утворення промислових відходів та негативного впливу на водні екосистеми. Аналіз

існуючих екологічних проблем дозволив визначити ключові фактори ризику та класифікувати їх за рівнем небезпеки.

Проведений огляд сучасних підходів до екологічної безпеки дав змогу обґрунтувати необхідність упровадження комплексу технологічних і організаційних заходів, спрямованих на мінімізацію впливу виробничих процесів. Запропоновані рішення охоплюють удосконалення систем очищення викидів, оптимізацію поводження з відходами, модернізацію виробничих ліній та підвищення контролю за використанням ресурсів.

Застосування методу аналізу ризиків із використанням дерева подій дозволило структуровано оцінити ймовірність виникнення екологічно небезпечних ситуацій та визначити критичні точки, що потребують першочергової уваги. Отримані результати показали, що реалізація превентивних заходів істотно знижує ймовірність настання аварійних подій та сприяє підвищенню загального рівня екологічної безпеки.

Таким чином, комплексна оцінка впливів, аналіз ризиків і сформовані рекомендації створюють методичне підґрунтя для підвищення екологічної стійкості виробничих процесів і підтверджують доцільність адаптації сучасних екологічних стандартів на суднобудівних підприємствах.

ВИСНОВКИ

У ході роботи було проведено комплексне дослідження сучасних методів і технічних засобів моніторингу технічного стану промислових двигунів, що дозволило визначити оптимальні підходи до побудови ефективної інформаційно-виміральної системи. Проаналізовано ключові параметри стану двигуна — вібрація, температура, струм, шум, тиск та інші, що формують основу для діагностики його працездатності.

Сформовано концептуальну модель ІВС, яка охоплює усі етапи: від збору первинних сигналів до автоматизованого аналізу та прийняття рішень. На її основі спроектовано архітектуру системи, адаптовану до вимог реального часу, обмежень виробничого середовища та надійності роботи.

Обґрунтовано вибір сенсорів і апаратних засобів збору даних, визначено необхідні характеристики для забезпечення точності вимірювань. Розроблено алгоритми оброблення даних та методи виявлення аномалій, що підвищують інформативність вимірювань та дозволяють своєчасно виявляти відхилення у роботі двигуна.

Створено математичну модель діагностики технічного стану, що може бути реалізована як у вигляді класичних методів аналізу, так і з використанням ML-підходів. Реалізовано прототип системи та проведено експериментальні дослідження, які підтвердили її працездатність, достатню точність і стабільність у виробничих умовах.

За результатами роботи сформовано рекомендації щодо впровадження системи в промислове середовище, що дозволяють підвищити ефективність технічного обслуговування, мінімізувати ризики відмов та оптимізувати витрати на експлуатацію двигунів.

Програма розроблена в середовищі об'єктно-орієнтованого програмування Python.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. IEC 60034-1:2017. Rotating electrical machines – Part 1: Rating and performance. – International Electrotechnical Commission, 2017. – 124 p.
2. Fluke Corporation. Motor & Drive Troubleshooting Guide. – Everett, WA, USA: Fluke, 2021. – 72 p.
3. Randall, R. B. *Vibration-based Condition Monitoring: Industrial, Aerospace and Automotive Applications*. – Wiley, 2021. – 450 p.
4. Mobley, R. K. *Maintenance Engineering Handbook*. – McGraw-Hill, 2020. – 1184 p.
5. Бесекерський В. А., Попов Е. П. Теория автоматического управления. — СПб.: Питер, 2018. – 448 с.
6. Holownia, P., Kuziak, R. *Condition Monitoring Systems in Industrial Applications*. – Springer, 2020. – 312 p.
7. Siemens AG. *MindSphere: Industrial IoT Monitoring Platform Technical Overview*. – Siemens, 2021. – 84 p.
8. Mitsubishi Electric. *Remote Monitoring System (RMS) Technical Manual*. – Tokyo: Mitsubishi Corp., 2019. – 102 p.
9. Грешнов А. Ю., Гайдаєнко В. А. Особливості використання інформаційно-вимірювальних систем у виробництві томатної пасти. // Інновації в суднобудуванні та океанотехніці: XVI Міжнародна науково-технічна конференція: матеріали (Миколаїв, 2025). – Миколаїв: НУК, 2025. – С. 652–654.
10. General Electric. *Remote Monitoring & Diagnostics for Power Systems – Technical Report*. – GE Power, 2020. – 56 p.
11. Іванченко Ф. К., Дубровін В. О. Системи технічної діагностики промислового обладнання. – К.: Ліра-К, 2019. – 268 с.
12. Поляк М. П. Математичні методи оцінки технічного стану електричних машин. – Х.: Нац. техн. ун-т «ХПІ», 2018. – 176 с.
13. Montgomery, D. C., Runger, G. C. *Applied Statistics and Probability for Engineers*. – Wiley, 2021. – 728 p.
14. Bishop, C. M. *Pattern Recognition and Machine Learning*. – Springer, 2016. – 738 p.

15. Іванов О. П., Шарий О. Г. Інтелектуальні вимірювальні системи в промисловості. – К.: Техніка, 2020. – 222 с.

ДОДАТОК А – Технічне завдання

1. Загальні положення

1.1. Найменування розробки.

Інформаційно-вимірвальна система моніторингу технічного стану двигунів у виробничих умовах (ІВС-МТСД).

1.2. Підстава для розроблення.

Необхідність забезпечення безперервного контролю й оперативної діагностики технічного стану двигунів у промисловому середовищі, зниження витрат на ремонт та підвищення рівня безпеки експлуатації.

1.3. Замовник

Промислове підприємство (виробничий цех), яке експлуатує електричні та механічні двигуни.

1.4. Розробник

Фахівець з інформаційних технологій / кафедра комп'ютерних систем і технологій.

1.5. Мета створення системи

Створення автоматизованої ІВС, здатної здійснювати безперервний збір параметрів роботи двигуна, їх аналіз, виявлення відхилень, прогнозування несправностей і забезпечення прийняття рішень у режимі реального часу.

2. Призначення системи

Система призначена для автоматизованого вимірювання параметрів роботи двигунів (вібрація, струм, напруга, температура, тиск тощо). Обробки, зберігання та аналізу вимірювань у реальному часі. Виявлення аномалій, діагностики стану та прогнозування можливих відмов. Формування сповіщень, технічних звітів і рекомендацій щодо технічного обслуговування. Інтеграції з системами SCADA, MES, ERP та хмарними аналітичними платформами.

3. Вимоги до функціональності системи

3.1. Основні функції системи

Збір даних із датчиків у реальному часі (частота знімання – до 1 кГц залежно від каналу). Попередня обробка сигналів (фільтрація, нормалізація, спектральний

аналіз). Визначення ключових діагностичних ознак (RMS, THD, FFT-спектри, температурні тренди). Автоматичне виявлення несправностей за ознаками.

Прогнозування залишкового ресурсу на основі моделей машинного навчання. Ведення бази даних історичних вимірювань та технічного стану обладнання. Формування звітів, графіків та інтерактивних панелей моніторингу. Генерація сповіщень при критичних відхиленнях. Підтримка локального та віддаленого доступу.

3.2. Додаткові функції

Адаптивне калібрування датчиків та каналів вимірювання. Захист даних та авторизація користувачів. Можливість масштабування на необмежену кількість двигунів. API для інтеграції з іншими інформаційними системами.

4. Вимоги до апаратного забезпечення

4.1. Вимірювальні датчики

Датчики струму (до 600 А). Датчики напруги (до 1000 В). Вібраційні датчики (акселерометри 0–10 kHz). Температурні датчики (термопари, ІЧ-сенсори). Датчики тиску (за потреби).

4.2. Обчислювальні вузли

Контролер збору даних (промисловий PLC або IoT-модуль). Локальний сервер/хмарна платформа для зберігання та аналізу даних. Мережеве обладнання (Ethernet/Wi-Fi/Industrial IoT).

5. Програмні вимоги

5.1. Системне програмне забезпечення. Операційна система Linux/Windows Server. Система керування базами даних: PostgreSQL / MySQL / MS SQL Server.

5.2. Прикладне програмне забезпечення

Модулі збору даних із датчиків. Модуль аналітики (FFT, ML-моделі). Модуль відображення (панель моніторингу, графіки, таблиці). Модуль керування подіями й оповіщеннями. Модуль експорту звітів (PDF/CSV). Мобільний/веб-інтерфейс.

5.3. Вимоги до розробки

Мова реалізації: Python/Java/C#. Забезпечення кросплатформенності. Розробка графічного інтерфейсу, адаптованого під виробничі потреби.

6. Вимоги до надійності та безпеки

Стійкість до відмов та автоматичне відновлення після збоїв. Захищені канали передачі даних (TLS, SSH). Резервне копіювання бази даних не рідше ніж раз на 24 години. Контроль доступу до системи за ролями. Відповідність базовим вимогам кібербезпеки промислових систем.

7. Вимоги до експлуатації.

Робоча температура апаратних модулів: $-10...+60$ °C. Робота в умовах підвищеної вібрації та пилу. Легкість монтажу та технічного обслуговування. Можливість оновлення ПЗ без зупинки системи.

8. Вимоги до документації

Розробник повинен надати технічний опис системи, інструкції користувача, опис протоколів обміну даними, регламент технічного обслуговування, паспорт системи та гарантійну документацію.

9. Етапи розробки

Аналіз вимог та технічних умов. Розробка концептуальної та фізичної архітектури ІВС. Проєктування бази даних. Розробка модулів збору даних. Розробка аналітичних модулів. Створення інтерфейсу користувача. Інтеграційні випробування. Тестування у виробничих умовах. Підготовка технічної документації. Введення системи в експлуатацію.

10. Порядок приймання

Система підлягає приймально-здавальним випробуванням, що включають тестування коректності вимірювань, перевірку працездатності кожного модуля, оцінку стійкості та точності аналітики, перевірку швидкості обробки сигналів, перевірку безпеки та надійності. Після успішного проходження випробувань складається акт приймання системи.

Додаток Б - Інструкція користувача

1. Загальні положення

Ця інструкція призначена для операторів, інженерів та технічних фахівців, які працюють з інформаційно-вимірювальною системою моніторингу технічного стану двигунів (ІВС-МТСД). Документ описує основні правила роботи, порядок запуску, налаштування, перегляду параметрів, інтерпретації даних та дії у разі виникнення нештатних ситуацій.

ІВС-МТСД забезпечує збір параметрів у реальному часі (вібрація, температура, струм, напруга, тиск, швидкість). Автоматичну обробку та первинну фільтрацію сигналів. Діагностику технічного стану. Прогнозування можливих відмов. Формування звітів і журналів подій. Передавання даних на центральний сервер, у SCADA або хмарну аналітичну платформу.

2. Увімкнення та запуск системи

2.1 Підготовка до роботи

Перед запуском необхідно перевірити підключення датчиків до вузлів збору даних. Забезпечити живлення вимірювальних модулів. Переконайтеся у наявності зв'язку з сервером або локальним контролером. Перевірити коректність встановлених калібрувальних коефіцієнтів.

2.2 Запуск програмного забезпечення

Увімкніть робочу станцію оператора. Запустіть програмне забезпечення IVS-EngineMonitor через робочий стіл або меню «Запуск». Виконайте авторизацію, вказавши логін та пароль. Після входу система автоматично завантажить останню конфігурацію датчиків та активних діагностичних каналів.

3. Робота з головним інтерфейсом

Головне вікно містить такі основні елементи: панель статусів обладнання – відображає реальний стан двигуна (норма, попередження, аварія). Онлайн-тренди параметрів – графіки вібрації, температури, струму тощо. Лог подій – список попереджень, помилок та спрацювань порогів. Меню налаштувань – конфігурація каналів, порогів, частоти опитування. Модуль аналітики – результати діагностики та прогнозування. Кнопки експорту – формування звітів (PDF, CSV).

4. Перегляд параметрів та аналітичних даних

4.1 Онлайн-моніторинг

У вкладці «Реальний час» користувач може, переглядати актуальні значення вимірюваних параметрів, відстежувати графічні тренди за останні хвилини, години або добу, контролювати стан двигуна за кольорами індикаторів.

4.2 Аналітика та діагностика

Модуль аналітики надає спектральний аналіз сигналів вібрації, аналіз гармонік струму, результати машинного прогнозування (ML-моделі), оцінку ймовірності наближеного відмовлення, рекомендації щодо технічного обслуговування. Відображення доступне у вигляді діаграм, таблиць та діагностичних рішень.

5. Налаштування системи

5.1 Калібрування

У вкладці «Калібрування» користувач може вводити коефіцієнти для датчиків, запускати локальну процедуру калібрування за алгоритмом, переглядати історію змін параметрів калібрування.

5.2 Параметри опитування

Доступно налаштування частоти вимірювання, фільтрів попередньої обробки, амплітудних, спектральних та інтегральних порогів, алгоритмів ML-прогнозування.

6. Робота зі звітами та журналами

Система підтримує автоматичне формування щоденних/щотижневих звітів, ручний експорт даних у PDF, CSV. Створення технічних висновків за шаблоном.

У модулі «Архів» користувач може обирати період збережених даних, переглядати чек-пункти технічних станів, завантажувати довготривалі тренди.

7. Дії у разі нештатних ситуацій

7.1 Попередження

Жовтий індикатор означає незначне відхилення параметра, можливе зношення підшипника, підвищену температуру вузла.

Дії користувача ознайомитися з описом події в журналі. Перевірити тренди за останній період. За потреби знизити навантаження або виконати технічну перевірку.

7.2 Аварія

Червоний індикатор означає критичне перевищення параметрів, підозру на міжвиткове замикання, високу температуру корпусу або обмоток. Різке зростання вібрації. Дії користувача негайно зупинити двигун згідно з інструкціями підприємства. Сповістити відповідального інженера. Переглянути діагностичні дані та сформувати аварійний звіт.

8. Забезпечення безпеки при роботі

Користувач зобов'язаний дотримуватися інструкцій з охорони праці. Не відкривати вимірювальні модулі під напругою. Не змінювати конфігурацію датчиків без дозволу. Працювати у захисних засобах (підприємство визначає перелік).

9. Завершення роботи

Закрити всі відкриті вікна та зберегти зміни. Вийти з облікового запису через меню «Вихід». Закрити програму. В разі потреби вимкнути робочу станцію.

10. Технічна підтримка

З метою усунення збоїв або виконання системного обслуговування користувач може звернутися до адміністратора системи, служби технічної підтримки підприємства, розробника ПЗ через вказані контактні канали.

ДОДАТОК В – Фрагмент програмного коду

```
unitUnit1;
{$IFDEFVER150}
{$WARNUNSAFE_CASTOFF}
{$WARNUNSAFE_TYPEOFF}
{$ENDIF}
{$IFDEFVER170}
{$WARNUNSAFE_CASTOFF}
{$WARNUNSAFE_CODEOFF}
{$ENDIF}

interface

uses
{$IFDEFVER140}
Variants,
{$ENDIF}
{$IFDEFVER150}
Variants,
{$ENDIF}
{$IFDEFVER170}
Variants,
{$ENDIF}
Windows, Forms, ComObj, SysUtils, OPCtypes,
  OPCDA, OPCutils, Printers, Messages, Classes, Graphics, Controls,
  Dialogs, StdCtrls, Mask, ExtCtrls, ComCtrls, ToolWin, ActnMan,
  ActnColorMaps, Menus, ImgList, Buttons, ActnCtrls, TabNotBk, ActnList,
  jpeg, Tabs, DB, DBTables, DBCtrls, Grids, DBGrids, StdActns, ValEdit, IdIntercept, IdComponent,
  IdIPWatch, IdCoderXXE, IdCoder, Gauges, ADODB, IdBaseComponent,
  IdCoder3to4, IdCoderUUE, Registry, ShellApi, SyncObjs, IdTCPConnection,
  IdTCPClient, TeEngine, Series, TeeProcs, Chart, Math, RM_Type, DBClient,
  dblookup, IdTCPServer, Indicator, ActiveX, DateUtils;

type

  TIndicate = class(TThread)
  private
    { Private declarations }
  protected
    procedure Execute; override;
  end;

  TArhive = class(TThread)
  private
    { Private declarations }
  protected
    procedure Execute; override;
  end;

  TTCPIPSend = class(TThread)
  private
    { Private declarations }
```

```
protected
procedure Execute; override;
end;
```

```
TSimulator = class(TThread)
private
    { Private declarations }
protected
procedure Execute; override;
end;
```

```
TDTCPTThread= class(TThread)
private
    { Private declarations }
protected
procedure Execute; override;
end;
```

```
TForm1 = class(TForm)
    OPCtimer: TTimer;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    MCommand: TMenuItem;
    N3: TMenuItem;
    N11: TMenuItem;
    StatusBar1: TStatusBar;
    ActionList1: TActionList;
    Timer: TTimer;
    DataSource1: TDataSource;
    Signals: TTable;
    GridPopupmenu: TPopupMenu;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    DelRec: TAction;
    InsRec: TAction;
    EditRec: TAction;
    EditCopy1: TEditCopy;
    EditPaste1: TEditPaste;
    Image1: TImage;
    N7: TMenuItem;
    N8: TMenuItem;
    Edit4: TMenuItem;
    N12: TMenuItem;
    Paste2: TMenuItem;
    Copy2: TMenuItem;
    Cut1: TMenuItem;
    N14: TMenuItem;
    Undo1: TMenuItem;
    EditUndo1: TEditUndo;
    EditDelete1: TEditDelete;
    EditCut1: TEditCut;
    PageControl: TPageControl;
```


TabSheet1: TTabSheet;
TabSheet3: TTabSheet;
DBGrid1: TDBGrid;
TabSheet5: TTabSheet;
IdIPWatch1: TIdIPWatch;
IdEncoderXXE1: TIdEncoderXXE;
IdDecoderXXE1: TIdDecoderXXE;
GroupBox2: TGroupBox;
GroupBox3: TGroupBox;
GBox1: TGroupBox;
SQLValueList: TValueListEditor;
Label6: TLabel;
GBox2: TGroupBox;
TCPValueList: TValueListEditor;
Label7: TLabel;
GBox3: TGroupBox;
Label12: TLabel;
Button7: TButton;
MOPC: TMenuItem;
MGroup: TMenuItem;
MSQL: TMenuItem;
N9: TMenuItem;
MRun: TMenuItem;
GroupBox1: TGroupBox;
SendFList: TListBox;
Label15: TLabel;
EvScroll: TCheckBox;
IdTCPClient1: TIdTCPClient;
TCPTimer: TTimer;
N2: TMenuItem;
N10: TMenuItem;
BitBtn3: TBitBtn;
ProgressBar2: TProgressBar;
Label21: TLabel;
L1: TLabel;
L2: TLabel;
L3: TLabel;
Duplicatefield: TMenuItem;
SignalLabel: TValueListEditor;
GroupBox8: TGroupBox;
IDP5: TImage;
Label4: TLabel;
GroupBox7: TGroupBox;
Label3: TLabel;
IDP4: TImage;
IDP41: TImage;
GroupBox6: TGroupBox;
IDP3: TImage;
Label5: TLabel;
GroupBox5: TGroupBox;
IDP2: TImage;
Label2: TLabel;
GroupBox4: TGroupBox;
IDP1: TImage;
Label1: TLabel;
GroupBox9: TGroupBox;

IDP6: TImage;
Label16: TLabel;
GroupBox10: TGroupBox;
IDP7: TImage;
Label18: TLabel;
TimeDataL: TLabel;
Page1: TAction;
Page2: TAction;
Page3: TAction;
Page4: TAction;
ListLabelInd: TValueListEditor;
Label19: TLabel;
Label17: TLabel;
RMAValueList: TValueListEditor;
ServerRMAccess: TIdTCPServer;
GroupBox11: TGroupBox;
IDP8: TImage;
Label22: TLabel;
GroupBox12: TGroupBox;
Label11: TLabel;
CompID5: TShape;
Label10: TLabel;
CompID4: TShape;
Label9: TLabel;
CompID3: TShape;
Label8: TLabel;
CompID2: TShape;
CompID6: TShape;
Label23: TLabel;
TabSheet2: TTabSheet;
SignalsView: TListView;
Page5: TAction;
GroupBox13: TGroupBox;
ChartP: TChart;
HorizBarSeries1: THorizBarSeries;
LineSeries1: TLineSeries;
LineSeries2: TLineSeries;
Series5: TLineSeries;
ChartV: TChart;
BarSeries1: THorizBarSeries;
Series3: TLineSeries;
Series4: TLineSeries;
ChartT: TChart;
Series1: TBarSeries;
GroupBox14: TGroupBox;
Image3: TImage;
LTin: TLabel;
LTmIN: TLabel;
LPmIN: TLabel;
LTmPX: TLabel;
LCDP: TLabel;
LNLP: TLabel;
LNHP: TLabel;
LT04: TLabel;
LNTP: TLabel;
LTmKBD: TLabel;

LTmTND: TLabel;
LTmCT: TLabel;
LPmOut: TLabel;
LTmOut: TLabel;
LdPkty: TLabel;
LPmob: TLabel;
LPower: TLabel;
GroupBox15: TGroupBox;
LKR2out: TLabel;
LKR2pos: TLabel;
LKR1pos: TLabel;
LKR1out: TLabel;
Image2: TImage;
LG: TLabel;
LPperKP: TLabel;
LPzaKP1: TLabel;
LPzaKP2: TLabel;
Series2: TLineSeries;
Button1: TButton;
Panel1: TPanel;
SPointValueList: TValueListEditor;
Button3: TButton;
GroupBox16: TGroupBox;
IDP0: TImage;
Button4: TButton;
DBNavigator1: TDBNavigator;
Button6: TButton;
ProgressBar1: TProgressBar;
Button8: TButton;
CompID7: TShape;
Label25: TLabel;
DBGT: TTable;
DataSource2: TDataSource;
GroupBox18: TGroupBox;
DBGrid2: TDBGrid;
AutoRun: TCheckBox;
WriteDB: TCheckBox;
SendTime: TMaskEdit;
ArhivTime: TMaskEdit;
ComboBox1: TComboBox;
Label14: TLabel;
Label13: TLabel;
Label20: TLabel;
Button10: TButton;
DBNavigator2: TDBNavigator;
GroupBox17: TGroupBox;
BitBtn2: TBitBtn;
BitBtn1: TBitBtn;
BitBtn4: TBitBtn;
Button2: TButton;
Button9: TButton;
GTSelect: TComboBox;
LSQLConnect: TLabel;
TabSheet4: TTabSheet;
Series6: TLineSeries;
SetSIM: TCheckBox;

PDMenu: TPopupMenu;
N13: TMenuItem;
N15: TMenuItem;
N16: TMenuItem;
N17: TMenuItem;
DTCPTimer: TTimer;
Events: TListBox;
GroupDInd: TComboBox;
SpeedButton1: TSpeedButton;
GroupBox19: TGroupBox;
Button11: TButton;
Button12: TButton;
Button13: TButton;
Button5: TButton;
SetSigToDInd: TComboBox;
DChart1: TChart;
Series7: TLineSeries;
Series8: TPointSeries;
DChart2: TChart;
LineSeries3: TLineSeries;
Series9: TLineSeries;
Series10: TLineSeries;
Series11: TLineSeries;
Series12: TLineSeries;
Series13: TLineSeries;
DSignalsView: TListView;
UpDown1: TUpDown;
Dchart3: TChart;
Series14: TBarSeries;
CompID1: TShape;
Label26: TLabel;

```
// procedure SendGetDiagnosticData(G:byte);  
procedure DeInit;  
// procedure DMesageToMRec(G:byte;DMCount:Byte);  
procedure WWriteFile(G:byte);  
procedure AddDataFileName(G:byte);  
procedure WriteSQL(G:byte);  
procedure ReadOPCData(G:byte);  
procedure DataToIndicators(G:byte);  
procedure DataToDIndicators(G:byte);  
procedure EventsAdd(wrf:boolean;st:string);  
procedure Registration;  
procedure AddToGroupeSig(G:byte);  
procedure run();  
procedure OPCtimerTimer(Sender: TObject);  
procedure FormResize(Sender: TObject);  
procedure TimerTimer(Sender: TObject);  
procedure Button4Click(Sender: TObject);  
procedure DelRecExecute(Sender: TObject);  
procedure InsRecExecute(Sender: TObject);  
procedure EditRecExecute(Sender: TObject);  
procedure N8Click(Sender: TObject);  
procedure Button6Click(Sender: TObject);  
procedure BitBtn2Click(Sender: TObject);  
procedure BitBtn1Click(Sender: TObject);
```

```

procedure BitBtn3Click(Sender: TObject);
procedure ADOConnectionAfterConnect(Sender: TObject);
procedure ADOConnectionAfterDisconnect(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure MOPCClick(Sender: TObject);
procedure MGroupClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure MSQLErrorClick(Sender: TObject);
procedure MRunClick(Sender: TObject);
procedure EvScrollClick(Sender: TObject);
procedure IdTCPClient1Connected(Sender: TObject);
procedure IdTCPClient1Disconnected(Sender: TObject);
procedure TCPTimerTimer(Sender: TObject);
procedure N2Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure N11Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Page4Execute(Sender: TObject);
procedure AutoRunClick(Sender: TObject);
procedure DuplicatefieldClick(Sender: TObject);
procedure WriteDBClick(Sender: TObject);
procedure Page1Execute(Sender: TObject);
procedure Page2Execute(Sender: TObject);
procedure Page3Execute(Sender: TObject);
procedure ServerRMAccessExecute(AThread: TIdPeerThread);
procedure CompID4MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Page5Execute(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure SetSIMClick(Sender: TObject);
procedure N13Click(Sender: TObject);
procedure N15Click(Sender: TObject);
procedure SetSigToDIndSelect(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure N17Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure Button12Click(Sender: TObject);
procedure Button13Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure DTCPTimerTimer(Sender: TObject);
procedure GroupDIndSelect(Sender: TObject);
procedure SetSigToDIndCloseUp(Sender: TObject);
procedure DChart1DbClick(Sender: TObject);
procedure DChart2DbClick(Sender: TObject);
procedure UpDown1Click(Sender: TObject; Button: TUDBtnType);
procedure ServerRMAccessConnect(AThread: TIdPeerThread);
procedure ServerRMAccessDisconnect(AThread: TIdPeerThread);

```

private

```
{ Private declarations }  
public
```

```
{ Public declarations }
```

```
Sig:array[1..GTCountMax,1..120] of TSig; // массивданныхсигналовСАУ
```

```
DevIn: array[1..GTCountMax,1..50] of TDevIn; // массивданныхвходящихсигналовдиагностики
```

```
DSig:array[1..GTCountMax,1..40] of TDSig; // массивданныхсигналовДиагностики
```

```
Location: array[1..GTCountMax,1..120] of byte;
```

```
LabelInd: array[1..50] of byte;
```

```
//1- NLP
```

```
//2- NHP
```

```
//3- NPT
```

```
//4- T04
```

```
//5- CDP
```

```
//6-21 T01 - T16
```

```
//22-Start
```

```
//23- Stop
```

```
GT_Count:byte; // Кол-во элементов в массиве
```

```
flags:array[0..30] of boolean; //Флагисостояния
```

```
// 0 - Ошибка считывания БД GT
```

```
// 1 - Ошибка считывания БД Сигналов одного из GT
```

```
// 2 - Настройки считаны
```

```
// 3 - Подключение к БД
```

```
// 4 - регистрация на сервере
```

```
// 5 - добавление группы
```

```
// 6,7 - добавление сигналов в группу
```

```
// 8,9 - Считывание данных с OPC (1 ни один сигнал не считан, 2 сигналы считаны частично)
```

```
// 10 - Запись SQL
```

```
// 11 - Запись в файл
```

```
// 12 - Архивирование файлов
```

```
// 13 - Ошибка создания архива!
```

```
// 14 - Авто скрол журнала событий
```

```
// 15 - Отправка файла
```

```
// 16 - режим он лайн
```

```
// 17 - RM запущен
```

```
// 18 - DTCPTimer работает
```

```
// 19 - TDTCPThread.Execute!!!
```

```
// 20 - Чтение буфера данных диагностики!
```

```
// 21 - zoom Dchart1!
```

```
// 22 - zoom Dchart2!
```

```
// 23 - TCPTimer работает
```

```
// 24 - Ошибка в работе диагностики
```

```
Settings:Tsettings;
```

```
GT:array [1..GTCountMax] of TGT;
```

```
DIndConfig:array[0..CountOfDIndGr-1] of TDIndGr;
```

```
DDataSend: TDDataSend;
```

```
DClient: array[1..GTCountMax] of TIdTCPClient;
```

```
end;
```

```
procedure CompressFile(const inFile, outFile: String); external 'ZLib.dll';
```

```
// procedure DecompressFile(const inFile, outFile: String); external 'ZLib.dll';
```

```
const
```

```
key='SOFTWARE\RM_Client';  
ServerProgID = 'RSLinx OPC Server'; // НазваниеOPCсервера  
GroupName = 'RMClient'; // Названиегруппысвязи  
RPC_C_AUTHN_LEVEL_NONE = 1;  
RPC_C_IMP_LEVEL_IMPERSONATE = 3;  
EOAC_NONE = 0;  
DMessageCount=150;
```

```
var
```

```
Form1: TForm1;  
ServerIf: IOPCServer;  
GroupIf: IOPCItemMgt; // интерфейс  
GroupHandle: OPCHANDLE; //DWORD  
StatType: TVarType; //ActiveX WORD  
StatQuality: Word;  
Timer:byte; // флагвключениятаймера  
HR,HR2: HRESULT; // Временнаяпеременнаясвязи (System) HRESULT = type Longint; { from  
WTYPES.H }  
ServerRMAccessUsersCount:byte; // Счетчик подключенных пользователей  
s: string; // Временное хранилище строк  
p: TextFile; // Переменная для принтера  
T,D:string; // Время и дата
```

```
TimeCount:LongWord;  
SendError:integer;  
//LogFileName:string;  
SemaforFileWR:boolean;
```

```
//RMessage:TRMessage;  
DMessage:array[1..GTCountMax,0..4] of TDBlock;
```

```
// потоки
```

```
Indicate:TIndicate;  
Arhive:TArhive;  
TCPIPSend:TTCPIPSend;  
Simulator:TSimulator;  
DTCPTThread:TDTCPTThread;
```

```
SR:array[1..80] of TFastLineSeries;  
FHeader: array [1..120] of TFHeader;  
SimData:array[1..120] of single;
```

```
DIndicator:array[1..10] of TIndicator;  
DIndicatorActiveTR:byte; // Номер индикатора на графиках  
DIndicatorActive:byte; // Активныйиндикатор  
DHandle:array[1..6] of THandle;
```

implementation

uses Unit2;

{ \$R *.dfm }

```
procedure SaveStrItemToReg(ItName:string; dataS:string);
var Registry: TRegistry;
begin
  { создаём объект TRegistry }
  Registry := TRegistry.Create;
  { устанавливаем корневой ключ; }
  Registry.RootKey := hkey_local_machine;
  { открываем и создаём ключ }
  Registry.OpenKey(key,true);
  { записываем значение }
  Registry.WriteString(ItName,dataS);
  { закрываем и освобождаем ключ }
  Registry.CloseKey;
  Registry.Free;
end;
```

```
procedure TForm1.EventsAdd(wrf:boolean;st:string);
label l1;
var F: textfile;
begin
  l1:
  sleep(10);
  if SemaforFileWR=true then goto l1;
  SemaforFileWR:=true;
```

//если больше 200 записей в листе то удалить 0й

```
if Events.Items.Count>200 then Events.Items.Delete(0);
```

```
try
  Events.Items.Add(T+' : '+st);
  //Скрол окна события вниз
  sleep(10);
  if flags[14]=true then Events.Perform(WM_VScroll,SB_LineDOWN,0) else
  Events.Perform(WM_VScroll,SB_ENDSCROLL,0) ;
  application.ProcessMessages;
except
end;
```

// запись в файл!!!!

```
if wrf then begin //wrf
assignFile(F,'LogFiles\'+D[1]+d[2]+'log');
try
  append(F);
except
  rewrite(f);
```



```
end;  
writeln(f,T+' : '+st);  
closefile(f);
```

```
    end; //wrf
```

```
SemaforFileWR:=false;  
end;
```