

Міністерство освіти і науки України  
Криворізький національний університет  
Факультет інформаційних технологій  
Кафедра автоматизації, комп'ютерних наук і технологій

## **КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття ступеня вищої освіти – бакалавр  
за освітньо-професійною програмою  
«Автоматизація та комп'ютерно-інтегровані технології»

зі спеціальності  
151 – Автоматизація та комп'ютерно-інтегровані технології

тема роботи:

***«Розробка веб-орієнтованої системи диспетчеризації та керування  
газовою котельнею з адаптивними алгоритмами роботи в умовах  
аварійного електропостачання»***

Виконав студент гр. АКІТ-22 \_\_\_\_\_ Шеремета Д.А.

Керівник \_\_\_\_\_ Тиханський М.П.

Нормоконтроль \_\_\_\_\_ Маринич І. А.

Завідувач кафедри \_\_\_\_\_ Рубан С. А.

# КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: інформаційних технологій

Кафедра: автоматизації, комп'ютерних наук і технологій

Ступінь вищої освіти: Бакалавр

Спеціальність: 151 - Автоматизація та комп'ютерно-інтегровані технології

**ЗАТВЕРДЖУЮ**

Зав. кафедрою: к.т.н. Рубан С.А.

---

« 26 » січня 2026 р.

## **ЗАВДАННЯ**

**на кваліфікаційну роботу бакалавра**

студентові групи АКІТ-22 Шереметі Дмитру Андрійовичу

**1. Тема кваліфікаційної роботи:** *«Розробка веб-орієнтованої системи диспетчеризації та керування газовою котельнею з адаптивними алгоритмами роботи в умовах аварійного електропостачання»*

затверджено наказом по університету № 171с від 30.03.2026 р.

**2. Термін здачі кваліфікаційної роботи:** 10.06.2026 р.

**3. Склад кваліфікаційної роботи:** Пояснювальна записка., презентація у Microsoft PowerPoint в електронному та друкованому вигляді

**4. Консультанти кваліфікаційної роботи:**

---

Розділ 1-2

доц. Тиханський М.П.

---

Нормоконтроль

доц. Маринич І. А.

## 5. Календарний план:

| № | Етапи роботи                                          | Термін виконання |
|---|-------------------------------------------------------|------------------|
| 1 | <i>Вступ</i>                                          | <i>01.03.26</i>  |
| 2 | <i>Розділ 1</i>                                       | <i>05.04.26</i>  |
| 3 | <i>Розділ 2</i>                                       | <i>01.05.26</i>  |
| 4 | <i>Висновки</i>                                       | <i>25.05.26</i>  |
| 5 | <i>Оформлення кваліфікаційної роботи</i>              | <i>28.05.26</i>  |
| 6 | <i>Підготовка презентації та графічного матеріалу</i> | <i>20.05.26</i>  |
| 7 | <i>Підготовка доповіді до захисту</i>                 | <i>05.06.25</i>  |

6. Дата видачі завдання: 28.01.2026р.

Керівник \_\_\_\_\_ / Тиханський М.П. /

7. Запевнення: Я, Шеремета Дмитро Андрійович, запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомлений.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Студент \_\_\_\_\_ / Шеремета Д.А./

## АНОТАЦІЯ

Шеремета Д.А. Розробка веб-орієнтованої системи диспетчеризації та керування газовою котельнею з адаптивними алгоритмами роботи в умовах аварійного електропостачання: кваліфікаційна робота бакалавра : 151 – Автоматизація, комп'ютерно-інтегровані технології. Кривий Ріг. Криворізький національний університет, 2026. 78 с.

Об'єктом дослідження є процес автоматизованого керування технологічними параметрами газової котельні в умовах нестабільного електропостачання та тривалих відключень електроенергії.

Метою роботи є розробка програмно-апаратного комплексу диспетчеризації з адаптивними алгоритмів енергозбереження, які забезпечують суттєве подовження часу автономної роботи системи від джерела безперебійного живлення без порушення контурів протиаварійного захисту.

У першому розділі проведено аналіз існуючих підходів до автоматизації об'єктів критичної інфраструктури. Обґрунтовано доцільність впровадження архітектури ПоТ та сформовано вимоги до апаратного забезпечення, мережевих протоколів і систем резервного живлення.

У другому розділі проведено комплексне проектування та імітаційне моделювання системи. У середовищі Node-RED реалізовано адаптивну логіку енергозбереження та зв'язок із ПЛК за протоколом Modbus TCP. Розроблено клієнтський веб-інтерфейс на базі React для моніторингу, аналітики трендів та фіксації подій. Тестування підтвердило здатність системи ефективно знижувати енергоспоживання при знеструмленні та гарантувати безпечну зупинку обладнання при критичному розряді акумулятора.

БЛЕКАУТ, ГАЗОВА КОТЕЛЬНЯ, АДАПТИВНЕ КЕРУВАННЯ, HMI, WEB SCADA, NODE-RED, MODBUS TCP, WEBSOCKET, ДБЖ.

|           |      |                 |        |      |                         |  |  |             |      |         |   |
|-----------|------|-----------------|--------|------|-------------------------|--|--|-------------|------|---------|---|
|           |      |                 |        |      | КНУ КРБ.151.26.13.00.ПЗ |  |  |             |      |         |   |
|           |      |                 |        |      |                         |  |  |             |      |         |   |
| Змн.      | Арк. | № докум.        | Підпис | Дата |                         |  |  |             |      |         |   |
| Розроб.   |      | ШПеремета Д.А.  |        |      | АНОТАЦІЯ                |  |  | Лім.        | Арк. | Аркушів |   |
| Перевір.  |      | Тиханський М.П. |        |      |                         |  |  |             |      | 4       | 1 |
|           |      |                 |        |      |                         |  |  | КНУ АКІТ-22 |      |         |   |
| Н. Контр. |      | Маринич І.А.    |        |      |                         |  |  |             |      |         |   |
| Затверд.  |      | Маринич І.А.    |        |      |                         |  |  |             |      |         |   |

## ЗМІСТ

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| ВСТУП.....                                                                           | 7  |
| РОЗДІЛ 1 АНАЛІЗ ГАЗОВОЇ КОТЕЛЬНІ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....                       | 10 |
| 1.1 Характеристика об'єкту автоматизації.....                                        | 10 |
| 1.2 Опис технологічного процесу теплопостачання .....                                | 11 |
| 1.3 Особливості роботи котельні в умовах аварійного електропостачання .....          | 12 |
| 1.4 Аналіз існуючих рішень .....                                                     | 14 |
| 1.5 Експлуатаційні вимоги до системи автоматизації.....                              | 17 |
| 1.5.1 Основні функції системи.....                                                   | 17 |
| 1.6 Постановка завдань дослідження .....                                             | 20 |
| РОЗДІЛ 2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ДОСЛІДЖУВАНОЇ СИСТЕМИ ТА ЇЇ МОДЕЛЮВАННЯ.....   | 22 |
| 2.1 Вибір технологічного обладнання та засобів автоматизації .....                   | 22 |
| 2.2 Формалізація об'єкта керування та вибір змінних стану .....                      | 27 |
| 2.3 Математична модель підсистеми автономного енергозабезпечення .....               | 28 |
| 2.4 Математична модель енергоспоживання циркуляційного насоса.....                   | 30 |
| 2.5 Математична модель теплової динаміки об'єкта керування .....                     | 31 |
| 2.6 Імітаційне моделювання об'єкта в середовищі MATLAB/Simulink .....                | 32 |
| 2.7 Алгоритмічне забезпечення системи диспетчеризації та адаптивного керування ..... | 37 |
| 2.8 Програмна реалізація у Node-RED.....                                             | 41 |

|           |      |                 |        |      |                                |      |         |
|-----------|------|-----------------|--------|------|--------------------------------|------|---------|
|           |      |                 |        |      | <i>КНУ КРБ.151.26.13.00.ПЗ</i> |      |         |
| Змн.      | Арк. | № докум.        | Підпис | Дата | <i>ЗМІСТ</i>                   |      |         |
| Розроб.   |      | ШІремента Д.А.  |        |      |                                |      |         |
| Перевір.  |      | Тиханський М.П. |        |      |                                |      |         |
|           |      |                 |        |      |                                |      |         |
| Н. Контр. |      | Маринич І.А.    |        |      |                                |      |         |
| Затверд.  |      | Маринич І.А.    |        |      | <i>КНУ АКІТ-22</i>             |      |         |
|           |      |                 |        |      | Літ.                           | Арк. | Аркушів |
|           |      |                 |        |      |                                | 5    | 2       |

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| 2.8.1 Загальна архітектура та концептуальна структура потоків у Node-RED..... | 41 |
| 2.8.2 Інтеграція взаємодії з ПЛК у Node-RED .....                             | 42 |
| 2.8.3 Організація двостороннього обміну даними через WebSockets .....         | 47 |
| 2.8.4 Реалізація алгоритмів керування та цифрового двійника .....             | 49 |
| 2.9 Архітектура та програмна реалізація НМІ .....                             | 54 |
| 2.9.1 Вибір стеку веб-технологій та мікрофронтендна архітектура               | 54 |
| 2.9.2 Візуалізація технологічного процесу та структура НМІ .....              | 56 |
| 2.9.3 Функціональне призначення та опис технологічних екранів .               | 57 |
| 2.9.4 Апробація роботи системи та моделювання нештатних ситуацій .....        | 62 |
| 2.10 Адаптивність кросплатформного користувацького інтерфейсу.....            | 68 |
| 2.11 Шляхи подальшого вдосконалення системи.....                              | 70 |
| ВИСНОВКИ.....                                                                 | 73 |
| СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....                                          | 76 |

## ВСТУП

Автоматизація — це застосування комплексу засобів, що дозволяють здійснювати виробничі процеси без безпосередньої фізичної участі людини, але під її постійним контролем. Автоматизація виробничих процесів призводить до збільшення продуктивності, зниження собівартості й поліпшення якості продукції, зменшує чисельність обслуговуючого персоналу, підвищує надійність і довговічність обладнання, дає економію енергоресурсів, а також суттєво поліпшує умови праці й техніки безпеки [1, 2].

В автоматизованому процесі виробництва роль людини зводиться до налагодження, конфігурування систем, обслуговування засобів автоматизації та високорівневого моніторингу за їхньою дією [17]. Сучасна автоматизація має на меті полегшити не тільки фізичну, але й розумову працю оператора шляхом впровадження інтелектуальних інформаційних систем та сучасних людино-машинних інтерфейсів (HMI) [24]. Експлуатація таких систем вимагає від обслуговуючого персоналу високої технічної кваліфікації.

За рівнем автоматизації теплоенергетика займає одне з провідних місць серед інших галузей промисловості. Теплоенергетичні установки характеризуються безперервністю процесів, що протікають у них. При цьому генерація теплової енергії в будь-який момент часу повинна відповідати навантаженню та вимогам споживачів. Більшість операцій на теплоенергетичних установках швидкоплинні, а перехідні процеси в них розвиваються динамічно, що унеможливлює виключно ручне керування.

В умовах сучасних викликів, пов'язаних із нестабільністю енергосистеми та загрозами тривалих блекаутів, забезпечення безперебійної роботи критичної інфраструктури набуває першочергового значення. Традиційні системи керування газовими котельнями орієнтовані на штатні режими роботи і не здатні автономно адаптувати своє енергоспоживання під час живлення від

|           |      |                  |        |      |                         |  |  |             |      |         |   |
|-----------|------|------------------|--------|------|-------------------------|--|--|-------------|------|---------|---|
|           |      |                  |        |      | КНУ КРБ.151.26.13.00.ПЗ |  |  |             |      |         |   |
|           |      |                  |        |      |                         |  |  |             |      |         |   |
| Змн.      | Арк. | № докум.         | Підпис | Дата |                         |  |  |             |      |         |   |
| Розроб.   |      | ШПеремета Д.А.   |        |      | ВСТУП                   |  |  | Лім.        | Арк. | Аркушів |   |
| Перевір.  |      | Тиханський М. П. |        |      |                         |  |  |             |      | 7       | 3 |
|           |      |                  |        |      |                         |  |  |             |      |         |   |
| Н. Контр. |      | Маринич І.А.     |        |      |                         |  |  | КНУ АКІТ-22 |      |         |   |
| Затверд.  |      | Маринич І.А.     |        |      |                         |  |  |             |      |         |   |

резервних джерел. Водночас стрімкий розвиток індустріального інтернету речей, периферійних обчислень та веб-технологій відкриває нові можливості для створення інтелектуальних цифрових двійників. Розробка адаптивної системи керування, яка здатна математично оптимізувати роботу обладнання для максимального подовження часу автономної роботи, є своєчасним, економічно обґрунтованим та актуальним науково-практичним завданням.

Метою роботи є підвищення енергоефективності та надійності роботи газової котельні в умовах аварійних відключень електроенергії шляхом розробки та впровадження адаптивної системи диспетчеризації на базі IoT-архітектури та цифрового двійника.

Об'єктом дослідження є процес автоматизованого керування теплогенеруючим комплексом (газовою котельнею) в режимі живлення від обмежених резервних джерел електроенергії.

Предметом дослідження виступають методи диспетчеризації на базі індустріальних веб-технологій та алгоритми адаптивного керування теплогенеруючим обладнанням в умовах нестабільного енергозабезпечення.

Поставлена мета досягається шляхом розв'язання таких основних завдань:

- Провести комплексний аналіз архітектур існуючих автоматизованих систем керування теплоенергетичними об'єктами та ідентифікувати їхні вразливості в умовах тривалих блекаутів.
- Розробити логічну структуру системи керування з використанням трирівневої моделі (ПЛК — IoT-шлюз — Веб-інтерфейс).
- Розробити адаптивні алгоритми виживання, здатні динамічно оптимізувати продуктивність циркуляційного обладнання залежно від залишкової ємності джерел безперебійного живлення.

- Здійснити програмну імплементацію серверного логічного ядра в середовищі Node-RED та створити клієнтський веб-інтерфейс із використанням мікрофронтендної архітектури React.
- Налаштувати асинхронну двонаправлену комунікацію між апаратними та програмними рівнями комплексу за допомогою індустріального стандарту Modbus TCP і мережевого протоколу WebSockets

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.00.ПЗ | Арк. |
|      |      |          |        |      |                         | 9    |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

## РОЗДІЛ 1

### АНАЛІЗ ГАЗОВОЇ КОТЕЛЬНОЇ ТА ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

#### 1.1 Характеристика об'єкту автоматизації

Газова котельня — це багатокомпонентний технологічний об'єкт, головним завданням якого є безперебійне виробництво та постачання теплової енергії споживачам. З точки зору управління, ця система має велику кількість взаємопов'язаних параметрів (температура, тиск, витрата води), а використання природного газу диктує підвищені вимоги до надійності контурів безпеки.

Специфічною рисою водогрійного котла є його значна теплова інерційність та затримки у реакції на керуючі команди. На стабільність технологічного процесу постійно впливають зовнішні фактори: зміна погодних умов, коливання споживання тепла в мережі, а також раптові перепади тиску газу та нестабільність напруги в електромережі. З огляду на це, система автоматизації повинна оперативно реагувати на збурення за допомогою гнучких адаптивних алгоритмів.

Апаратна архітектура типової котельні складається з кількох ключових вузлів. Основну функцію виконує сам котел із газовим пальником, де відбувається нагрівання теплоносія. За безпечну подачу палива відповідає газова лінія, оснащена фільтрами та електромагнітними клапанами, які миттєво блокують подачу газу під час аварійних ситуацій. Регулювання цільової температури води здійснюється за допомогою триходових клапанів із сервоприводами, що дозволяють точно змішувати гарячий та зворотний потоки.

Для забезпечення роботи в умовах блекаутів критично важливою ланкою є підсистема резервного живлення на базі ДБЖ та акумуляторних батарей. Вона гарантує безперебійну роботу контролерів та аварійної арматури при

|           |      |                  |        |      |                         |      |         |  |  |  |
|-----------|------|------------------|--------|------|-------------------------|------|---------|--|--|--|
|           |      |                  |        |      | КНУ КРБ.151.26.13.01.ПЗ |      |         |  |  |  |
|           |      |                  |        |      |                         |      |         |  |  |  |
| Змн.      | Арк. | № докум.         | Підпис | Дата | РОЗДІЛ 1                |      |         |  |  |  |
| Розроб.   |      | ШПеремета Д.А.   |        |      |                         |      |         |  |  |  |
| Перевір.  |      | Тиханський М. П. |        |      |                         |      |         |  |  |  |
|           |      |                  |        |      |                         |      |         |  |  |  |
| Н. Контр. |      | Маринич І.А.     |        |      |                         |      |         |  |  |  |
| Затверд.  |      | Маринич І.А.     |        |      |                         |      |         |  |  |  |
|           |      |                  |        |      | Літ.                    | Арк. | Аркушів |  |  |  |
|           |      |                  |        |      |                         |      | 1012    |  |  |  |
|           |      |                  |        |      | КНУ АКІТ-22             |      |         |  |  |  |

знеструмленні. Збір оперативних даних для керування всім процесом забезпечують контрольно-вимірювальні прилади: датчики температури, тиску, рівня теплоносія та сигналізатори загазованості, які безперервно передають інформацію на головний контролер.

## 1.2 Опис технологічного процесу теплопостачання

Технологічний цикл газової котельні ґрунтується на безперервному перетворенні хімічної енергії палива в теплову з подальшим транспортуванням нагрітого теплоносія до кінцевих споживачів. Головна мета цього процесу — суворе підтримання заданих температурних і гідравлічних графіків. На початковій стадії охолоджена вода з тепломережі потрапляє до зворотних колекторів. Циркуляційні агрегати забезпечують її постійний рух замкненим контуром, створюючи необхідний робочий напір. У цей час датчики безперервно моніторять тиск, щоб не допустити його виходу за безпечні межі, запобігаючи виникненню кавітаційних явищ у насосах та пошкодженню магістральних трубопроводів.

Процес активної генерації тепла запускається автоматично, коли температура теплоносія падає нижче встановленого диспетчером порогу. Логічний контролер генерує керуючу команду на розпал газового пальника, ініціюючи чітку послідовність дій: перевірку наявності тяги, попередню вентиляцію камери згоряння, відкриття електромагнітних клапанів та подачу іскри. Під час горіння вивільнена теплова енергія передається воді через металеві стінки теплообмінника. Аби уникнути руйнівних термічних напружень у конструкції котла, швидкість нагріву плавно регулюється, паралельно автоматика безпеки відстежує стабільність полум'я та робочий тиск газу в колекторі.

У штатному експлуатаційному режимі система самостійно стабілізує всі технологічні показники. Контролер циклічно зчитує дані з датчиків

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.01.ПЗ | Арк. |
|      |      |          |        |      |                         | 11   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

температури прямої та зворотної ліній, а також трансмітерів тиску, реалізуючи надійний алгоритм двопозиційного регулювання. Коли теплоносії нагрівається до верхньої межі уставки, подача палива припиняється; при охолодженні до нижньої межі — цикл теплогенерації відновлюється. Якщо в системі фіксується витік та падіння тиску, автоматика самостійно відкриває лінію підживлення для поповнення втраченого об'єму води.

Особлива увага в технологічному процесі приділяється реакції на нештатні та аварійні події. Будь-яке критичне відхилення, таке як зникнення факела, загазованість приміщення метаном [16] або критичний рівень теплоносія, призводить до миттєвого апаратного блокування процесу горіння. Визначальною інноваційною рисою розробленої системи є її здатність адаптуватися до умов нестабільного енергопостачання. У разі знеструмлення та автоматичного переходу на живлення від акумуляторів ДБЖ, логічний контролер змінює пріоритети керування. Він переводить циркуляційні насоси в режим суттєво зниженого енергоспоживання та оптимізує графіки нагріву. Такий підхід гарантує безпечне завершення робочих циклів, збереження безперервної передачі телеметричних даних та максимальне подовження часу автономної роботи об'єкта критичної інфраструктури.

### 1.3 Особливості роботи котельні в умовах аварійного електропостачання

В умовах сучасних енергетичних викликів ключовою проблемою експлуатації теплогенеруючих об'єктів стали регулярні та довготривалі перебої в роботі централізованих електромереж. Попри те, що базова теплова енергія видобувається шляхом спалювання природного газу, загальна працездатність комплексу має абсолютну залежність від стабільного електроживлення. Електричний струм є базовою вимогою для функціонування мікропроцесорної техніки, генерації іскри розпалу, утримання запірної газової арматури в робочому відкритому положенні та, що є найбільшим споживачем кіловат-

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.01.ПЗ | Арк. |
|      |      |          |        |      |                         | 12   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

годин, — для забезпечення безперервного обертання масивних роторів мережевих насосів.

У момент раптової втрати основної напруги автоматика зобов'язана миттєво перемикає апаратну частину на резервні інверторні лінії з акумуляторними блоками. Однак застарілі або жорстко запрограмовані системи керування мають суттєвий недолік: вони не здатні отримувати зворотний зв'язок про поточний стан батарей і продовжують експлуатувати виконавчі механізми на номінальних потужностях. За таких обставин акумуляований запас енергії вичерпується вкрай швидко, що неминуче призводить до повної і неконтрольованої зупинки процесу теплогенерації. Під час опалювального сезону такий сценарій створює пряму загрозу розмерзання тепломереж із подальшим незворотним руйнуванням гідравлічної інфраструктури трубопроводів.

З огляду на вищезазначене, виникає гостра необхідність у модернізації об'єкта шляхом впровадження інтелектуальної цифрової платформи, яка буде не лише керувати агрегатами, але й безперервно відстежувати енергетичний баланс ДБЖ. Адаптивне програмне забезпечення повинно вміти оперативно переформатовувати базову логіку поведінки системи одразу після реєстрації блекауту. Ключовим механізмом тут стає примусове коригування температурних уставок до мінімальних меж (так званого режиму "виживання") та синхронне зниження продуктивності циркуляційних насосів, що дозволяє радикально зменшити навантаження на інвертор. Реалізація такого підходу гарантує захист від раптових відключень, багаторазово збільшує часове вікно автономності до моменту відновлення живлення, а також забезпечує диспетчерський персонал інструментами дистанційного контролю в кризових обставинах.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.01.ПЗ | Арк. |
|      |      |          |        |      |                         | 13   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

## 1.4 Аналіз існуючих рішень

Еволюція сучасних комплексів диспетчеризації та автоматизації (АСУ ТП) чітко демонструє тенденцію до відмови від застарілих монолітних архітектур на користь гнучких, децентралізованих рішень на базі концепції промислового інтернету речей. Для теплогенеруючих підприємств, зокрема газових котелень, ця зміна парадигми продиктована суворими реаліями сьогодення, де на перше місце виходить здатність інфраструктури зберігати експлуатаційну стійкість в умовах довготривалих перебоїв з електроживленням.

Традиційно базовим інструментом для керування такими об'єктами слугували класичні SCADA-платформи, такі як Siemens WinCC, Wonderware InTouch або MasterSCADA. Сильною стороною подібних продуктів історично вважається висока відмовостійкість, наявність широкої бібліотеки інтегрованих драйверів для взаємодії з промисловим обладнанням, а також гарантований час реакції та захищеність від кібератак завдяки фізичному сегментуванню технологічних мереж.

Незважаючи на доведену роками надійність, практичний досвід застосування цих платформ оголює низку фундаментальних обмежень у контексті новітніх кризових сценаріїв. Насамперед проблемою є архітектурна негнучкість, оскільки диспетчерські інтерфейси переважно формуються як масивні десктопні додатки, фізично прив'язані до конкретної комп'ютерної станції в операторній. Організація безпечного віддаленого доступу в таких умовах є обтяжливою і часто вимагає значних фінансових вкладень у додаткове ліцензування. Окрім того, монолітна природа коду суттєво ускладнює процес модернізації: навіть мінімальні візуальні коригування на мнемосхемі змушують інженерів повністю зупиняти, перекомпілювати та перезапускати весь диспетчерський проєкт. Проте найбільш вразливим місцем класичних систем виявилася повна відсутність механізмів енергетичної адаптивності. Оскільки

логічне ядро традиційних SCADA створювалося з розрахунком на гарантоване енергозабезпечення, вони не містять вбудованих інтелектуальних сценаріїв, здатних динамічно оптимізувати споживання ресурсів та подовжити життєвий цикл обладнання від резервних акумуляторів під час раптових блекаутів.

На противагу традиційним підходам, актуальний вектор розвитку індустріальної автоматики та патентні розробки провідних інтеграторів однозначно вказують на міграцію в бік периферійних обчислень та веб-орієнтованих інтерфейсів. Делегування функцій первинної обробки телеметрії на рівень промислових IoT-шлюзів дозволяє кардинально знизити навантаження на мережеву інфраструктуру, тоді як впровадження сучасних веб-технологій відкриває шлях до відмови від громіздкого стаціонарного програмного забезпечення на користь легких кросплатформних браузерних застосунків.

Зважаючи на виявлені обмеження класичних систем, фундаментальною базою для реалізації даного проєкту стала сучасна трирівнева архітектура промислового інтернету речей формату Web SCADA, рисунок Рисунок 1.1. Такий концептуальний підхід гармонійно поєднує апаратну надійність програмованих логічних контролерів із винятковою гнучкістю та масштабованістю сучасного веб-стеку.

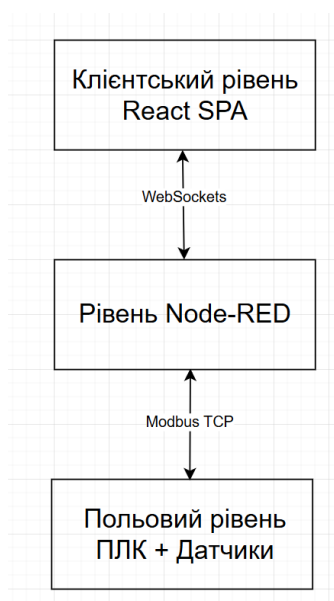


Рисунок 1.1— Трирівнева IoT-архітектура Web SCADA

На нижньому рівні обміну даними між польовим обладнанням та контролером використано індустріальний стандарт Modbus TCP [6], що забезпечує детерміновану та безперебійну трансляцію керуючих директив у межах локального сегмента мережі. Функції проміжної обчислювальної ланки покладено на подієво-орієнтоване середовище Node-RED, де розгорнуто ядро адаптивних алгоритмів. Саме цей шлюзовий вузол здійснює безперервний моніторинг енергетичного статусу резервних джерел живлення та автономно перераховує логіку поведінки виконавчих механізмів, не вимагаючи внесення ризикованих змін у жорстко детермінований код ПЛК.

Проблематику закритості та монолітності диспетчерського інтерфейсу було успішно подолано шляхом розробки клієнтського середовища на базі технології React із застосуванням концепції мікрофронтендів. Відповідна архітектура дозволяє генерувати ізольовані інтерфейсні віджети та гарантує повну апаратну незалежність: оператор отримує можливість здійснювати комплексний контроль над технологічним об'єктом з будь-якого доступного смарт-пристрою, перебуваючи в укритті або за межами диспетчерської.

З метою мінімізації мережевих затримок та усунення недоліків класичного синхронного опитування (HTTP-polling), транспортний зв'язок між шлюзом Node-RED та React-додатком побудовано на базі повнодуплексного протоколу WebSockets. Таке рішення гарантує миттєву, асинхронну реплікацію аварійних сповіщень та технологічних трендів у режимі реального часу. Синтез вищезазначених інформаційних технологій дозволив спроектувати високонадійний, адаптивний диспетчерський комплекс, який не лише нівелює ризики при енергетичних кризах, але й суттєво перевершує застарілі SCADA-моноліти за показниками гнучкості та оперативності.

## 1.5 Експлуатаційні вимоги до системи автоматизації

Система повинна бути побудована як багаторівнева автоматизована система керування технологічним процесом (АСУ ТП). Перелік основних автоматизованих трактів об'єкта:

- гідравлічний контур циркуляції та дистрибуції теплоносія
- паливна магістраль подачі та регулювання тиску природного газ;
- тракт подачі повітря та відводу димових газів;
- тракт автономного енергозабезпечення (моніторинг ДБЖ).

Логіка керування покликана підтримувати стабільність теплогенераційного циклу, надаючи диспетчерському персоналу інструментарій для глибокого віддаленого контролю та оперативного коригування технологічних уставок через захищений веб-інтерфейс. Оскільки експлуатаційне середовище апаратної частини (мережевих шлюзів та мікропроцесорів) характеризується високою ймовірністю критичних просадок або повної втрати мережевої напруги, апаратно-програмний комплекс оснащується алгоритмами автономного виживання. При фіксації блекауту та перемиканні на живлення від інвертора система здатна без жодного втручання людини активувати протоколи адаптивного енергозбереження, гарантуючи збереження керованості та працездатності критичної інфраструктури.

### 1.5.1 Основні функції системи

Система автоматизації та диспетчеризації виконує комплекс функцій, які концептуально поділяються на три основні групи: функції автоматичного регулювання, дистанційного керування (НМІ) та технологічного захисту.

Функції автоматичного регулювання та управління. В штатному режимі роботи головним завданням ПЛК є підтримання заданої температури теплоносія на виході з котла шляхом двопозиційного регулювання потужності газового пальника (циклічне вмикання та вимикання агрегату залежно від

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.01.ПЗ | Арк. |
|      |      |          |        |      |                         | 17   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

температурних уставок). Замість складного гідравлічного регулювання клапанами підживлення, система зосереджена на пріоритеті безпеки: здійснюється безперервний моніторинг тиску в контурі з ініціацією аварійної зупинки у разі його виходу за допустимі межі. Ключовою інноваційною функцією є адаптивне керування, яке передбачає динамічну зміну цільової уставки температури та частоти обертання циркуляційних насосів залежно від наявності напруги в основній мережі та поточного рівня заряду джерела безперебійного живлення (ДБЖ).

Функції дистанційного керування та візуалізації. Веб-орієнтований інтерфейс верхнього рівня, побудований на базі сучасних веб-технологій та компонентної архітектури, забезпечує відображення поточного стану всіх агрегатів та значень вимірювальних датчиків у реальному часі. Інтерфейс надає диспетчеру інструменти для дистанційного запуску або зупинки механізмів у ручному режимі, а також забезпечує ведення безперервного архіву аварійних подій і графіків зміни технологічних параметрів.

Технологічні захисти, блокування та сигналізація. Найважливішою функцією нижнього рівня системи є забезпечення абсолютної безпеки об'єкта. При виході контрольованих параметрів за допустимі межі, контролер зобов'язаний миттєво відпрацювати алгоритм захисту. Перелік основних аварійних ситуацій та відповідних дій автоматики зведено у таблицю 1.1

Таблиця 1.1 — Матриця технологічних захистів, блокувань та адаптивних реакцій

| Контрольований параметр | Умова спрацьовування (аварійна межа) | Дія системи автоматизації                  | Тип дії      |
|-------------------------|--------------------------------------|--------------------------------------------|--------------|
| Зв'язок із контролером  | Відсутність відповіді від ПЛК        | Тривога на дашборді, запис у журнал подій. | Сигналізація |

Продовження таблиці 1.1

| Контрольований параметр | Умова спрацьовування (аварійна межа)       | Дія системи автоматизації                                     | Тип дії             |
|-------------------------|--------------------------------------------|---------------------------------------------------------------|---------------------|
| Рівень загазованості    | Перевищення ГДК газу в приміщенні          | Закриття ввідного клапана, старт аварійної вентиляції.        | Блокування          |
| Напруга мережі          | Втрата фази або падіння нижче норми        | Перехід на ДБЖ, економ-режим (зниження $t^{\circ}$ , насоси). | Адаптивне керування |
| Рівень заряду ДБЖ       | Зниження заряду нижче критичного           | Збереження уставок в пам'ять, штатна зупинка.                 | Блокування          |
| Зв'язок із контролером  | Відсутність відповіді від ПЛК              | Тривога на дашборді, запис у журнал подій.                    | Сигналізація        |
| Температура теплоносія  | Перевищення максимально допустимої уставки | Відключення пальника, робота насоса (охолодження).            | Блокування          |
| Тиск води в контурі     | Вихід за допустимі межі                    | Зупинка пальника та насосів, запуск підживлення.              | Блокування          |

## 1.6 Постановка завдань дослідження

Предмет дослідження: методи диспетчеризації на базі індустріальних веб-технологій та алгоритми адаптивного керування теплогенеруючим обладнанням в умовах нестабільного енергозабезпечення

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- Провести комплексний аналіз архітектур існуючих автоматизованих систем керування теплоенергетичними об'єктами та ідентифікувати їхні вразливості в умовах тривалих блекаутів.
- Розробити логічну структуру системи керування з використанням трирівневої моделі (ПЛК — IoT-шлюз — Веб-інтерфейс).
- Розробити адаптивні алгоритми виживання, здатні динамічно оптимізувати продуктивність циркуляційного обладнання залежно від залишкової ємності джерел безперебійного живлення.
- Здійснити програмну імплементацію серверного логічного ядра в середовищі Node-RED та створити клієнтський веб-інтерфейс із використанням мікрофронтендної архітектури React.
- Налагодити асинхронну двонаправлену комунікацію між апаратними та програмними рівнями комплексу за допомогою індустріального стандарту Modbus TCP і мережевого протоколу WebSockets.

*Висновки за розділом:*

У першому розділі виконано комплексний системний аналіз газової котельні як об'єкта автоматизації та управління. Детально досліджено технологічний процес теплопостачання та виявлено критичні фактори, що впливають на безпеку об'єкта, зокрема в умовах тривалих аварійних відключень електроенергії. Визначено, що раптова зупинка циркуляційних насосів та відмова автоматики управління можуть призвести до незворотних наслідків:

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.01.ПЗ | Арк. |
|      |      |          |        |      |                         | 20   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

локального закипання теплоносія в теплообміннику, стрибків тиску та подальшого розмерзання зовнішніх трубопроводних мереж.

Проведений аналітичний огляд існуючих систем диспетчеризації показав, що класичні монолітні SCADA-рішення не повною мірою відповідають сучасним інженерним вимогам. Вони характеризуються надмірною ресурсоемністю, високою вартістю ліцензування, складністю інтеграції з хмарними сервісами та відсутністю вбудованих гнучких алгоритмів для роботи в умовах жорсткого енергетичного дефіциту.

На основі виявлених недоліків обґрунтовано доцільність переходу до сучасної розподіленої IoT-архітектури з використанням концепції периферійних обчислень. Доведено, що перенесення аналітичної обробки даних (зокрема, алгоритмів «виживання» та цифрового двійника) безпосередньо на локальний шлюз дозволяє забезпечити автономність об'єкта навіть за повної відсутності інтернет-з'єднання.

Результатом проведеного дослідження стало чітке формування мети та технічного завдання дипломного проекту: розробка апаратно-програмного комплексу, здатного автономно адаптувати енергоспоживання (керуючи частотою насосів та температурними уставками), прогнозувати час роботи резервних джерел живлення та забезпечувати дуплексний обмін телеметрією із сучасним кросплатформним веб-інтерфейсом управління

## РОЗДІЛ 2

### РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ДОСЛІДЖУВАНОЇ СИСТЕМИ ТА ЇЇ МОДЕЛЮВАННЯ

#### 2.1 Вибір технологічного обладнання та засобів автоматизації

Для побудови математичної моделі та подальшого проєктування компонентів цифрового двійника [5] у середовищі Web SCADA необхідно визначити параметри фізичного прототипу об'єкта керування. Враховуючи специфіку українського теплоенергетичного сектору та сучасні тенденції модернізації муніципальних котелень, у якості базового теплогенераційного вузла обрано промисловий газовий водогрійний котел КОЛВІ 250 [12], рисунок Рисунок 2.1, номінальною тепловою потужністю 250 кВт. Даний агрегат масово впроваджується для автономного опалення об'єктів соціальної інфраструктури та житлових комплексів завдяки своїй надійності. Фізичні та теплотехнічні параметри котла, які є критично важливими для розрахунку коефіцієнтів термодинамічної моделі, зведено у таблицю 2.1.



Рисунок 2.1— Водогрійний котел КОЛВІ 250

|           |      |                  |        |      |                         |  |  |  |             |      |         |
|-----------|------|------------------|--------|------|-------------------------|--|--|--|-------------|------|---------|
|           |      |                  |        |      | КНУ КРБ.151.26.13.02.ПЗ |  |  |  |             |      |         |
|           |      |                  |        |      |                         |  |  |  |             |      |         |
| Змн.      | Арк. | № докум.         | Підпис | Дата | РОЗДІЛ 2                |  |  |  | Літ.        | Арк. | Аркушів |
| Розроб.   |      | ШПеремета Д.А.   |        |      |                         |  |  |  |             |      |         |
| Перевір.  |      | Тиханський М. П. |        |      |                         |  |  |  |             | 22   | 51      |
|           |      |                  |        |      |                         |  |  |  | КНУ АКІТ-22 |      |         |
| Н. Контр. |      | Маринич І.А.     |        |      |                         |  |  |  |             |      |         |
| Затверд.  |      | Маринич І.А.     |        |      |                         |  |  |  |             |      |         |

Таблиця 2.1 — Технічні характеристики водогрійного котла КОЛВІ 250

| Найменування параметра                         | Значення  | Одиниці вимірювання |
|------------------------------------------------|-----------|---------------------|
| Номінальна теплова потужність                  | 250       | кВт                 |
| Коефіцієнт корисної дії                        | 92        | %                   |
| Максимальна температура теплоносія             | 95        | °С                  |
| Водяний об'єм котлового контура                | 350       | літр                |
| Номінальна витрата газу при макс. навантаженні | 28.5      | м³/год              |
| Робочий тиск у контурі                         | 0.3 (3.0) | МПа (бар)           |

Забезпечення безперервної циркуляції теплоносія в мережевому контурі покладено на високоефективний регульований насос із мокрим ротором Wilo Stratos MAXO 40/0.5-12 [13], рисунок Рисунок 2.2. Наявність інтегрованих частотних перетворювачів дозволяє гнучко керувати продуктивністю агрегату, що є фундаментальною умовою для імплементації алгоритмів зниженого енергоспоживання при перемиканні котельні на живлення від ДБЖ. Паспортні параметри насоса представлено у таблиці 2.2.



Рисунок 2.2— Насос із мокрим ротором Wilo Stratos MAXO 40/0.5-12

Таблиця 2.2 — Технічні характеристики циркуляційного насоса Wilo Stratos MAXO 40/0.5-12

| Найменування параметра                      | Значення         | Одиниці вимірювання |
|---------------------------------------------|------------------|---------------------|
| Максимальна продуктивність                  | 18.5             | м³/год              |
| Максимальний гідравлічний напір             | 12               | метрів вод. ст.     |
| Номінальна електрична потужність            | 410              | Вт                  |
| Мінімальна споживана потужність (еко-режим) | 15               | Вт                  |
| Напруга живлення                            | 230              | В                   |
| Інтерфейси передачі даних                   | Modbus TCP / RTU | -                   |

Комплекс засобів автоматизації розподілено за ієрархічними рівнями АСУ ТП. На польовому рівні для збору телеметрії та безпосереднього керування виконавчою арматурою обрано промисловий програмований логічний контролер Siemens SIMATIC S7-1200 (CPU 1214C) [11], рисунок Рисунок 2.3. Цей мікропроцесорний пристрій гарантує детерміноване виконання контурів протиаварійного захисту та підтримує обмін даними за стандартом Modbus TCP



Рисунок 2.3— Контролер Siemens SIMATIC S7-1200 (CPU 1214C)

Функції проміжного рівня, зокрема розгортання логічного ядра у Node-RED та агрегацію телеметрії для клієнтського веб-інтерфейсу, реалізовано на базі індустріального IoT-шлюзу Siemens SIMATIC IOT2050, рисунок Рисунок 2.4. Дане апаратне рішення адаптоване для цілодобової експлуатації в умовах промислових приміщень і забезпечує надійну обробку адаптивних сценаріїв виживання об'єкта під час блекаутів.



Рисунок 2.4— Індустріальний IoT-шлюз Siemens SIMATIC IOT2050

Важливим елементом нижнього рівня АСУ ТП є контрольно-вимірювальні прилади. Для моніторингу теплових параметрів обрано термоперетворювачі опору РТ100 [15], рисунок Рисунок 2.5, а для контролю гідравлічного тиску — перетворювачі із стандартним вихідним сигналом 4-20 мА. Забезпечення протиаварійного захисту реалізовано за допомогою промислового сигналізатора загазованості ВАРТА 1-03, рисунок Рисунок 2.6, який фіксує перевищення допустимої концентрації метану ( $CH_4$ ) у приміщенні та передає дискретний сигнал на ПЛК для негайного закриття головного ввідного газового клапана-відсікача.



Рисунок 2.5— Термоперетворювач опору РТ100



Рисунок 2.6— Сигналізатор загазованості ВАРТА 1-03

Оскільки ключовим завданням системи є експлуатація в умовах нестабільного електропостачання, серцевиною апаратної частини виступає потужна підсистема автономного живлення. Для об'єкта обрано гібридний інвертор промислового класу Victron Energy MultiPlus-II 48/3000, рисунок Рисунок 2.7, у комплексі з масивом літій-залізо-фосфатних ( $\text{LiFePO}_4$ ) акумуляторних батарей Pylontech US3000C (4 модулі, 48 В) загальною ємністю 300 А·год, рисунок Рисунок 2.8. Інвертор підтримує протокол Modbus TCP, що дозволяє IoT-шлюзу безперервно зчитувати рівень заряду батарей та динамічно адаптувати продуктивність циркуляційних насосів для максимізації часу автономної роботи котельні в період тривалих блекаутів



Рисунок 2.7— Гібридний інвертор промислового класу Victron Energy MultiPlus-II 48/3000



Рисунок 2.8— Масив LiFePO4 акумуляторних батарей Pylontech US3000C

## 2.2 Формалізація об'єкта керування та вибір змінних стану

Для розробки математичної моделі адаптивного керування газовою котельнею необхідно формалізувати систему, визначивши ключові параметри, які описують її стан у штатному та аварійному режимах. Об'єкт керування розглядається як складна багатовимірна динамічна система, стан якої визначається векторами вхідних впливів, змінних стану та керуючих сигналів.

Оскільки в рамках даної роботи застосовується двопозиційне регулювання пальника та фокус зміщено на адаптивне керування в умовах блекауту, формалізований перелік параметрів оптимізовано відповідно до розробленого цифрового двійника. Стан системи описується наступними векторами даних.

Вектор збурюючих впливів  $F$  включає параметри, на які система автоматизації не може впливати безпосередньо, але які визначають логіку роботи:

- $T_{ext}$  — температура зовнішнього повітря, °C (впливає на вибір температурних уставок під час роботи від ДБЖ);
- $U_{grid}$  — напруга основної живильної мережі, В (дискретний параметр: 1 — наявна, 0 — блекаут).

Вектор змінних стану, контрольовані параметри,  $Y$  містить інформацію про поточний теплотехнічний та енергетичний статус об'єкта:

- $T_{sup}$  — температура теплоносія в подавальному трубопроводі (на виході з котла), °C;
- $T_{ret}$  — температура теплоносія у зворотному трубопроводі (на вході в котел), °C;
- $T_{room}$  — усереднена температура повітря всередині приміщення, °C;
- SOC (поточний рівень заряду акумуляторних батарей ДБЖ, %;
- $P_{water}$  — тиск теплоносія в гідравлічному контурі, Bar
- $C_{CH_4}$  — концентрація метану в приміщенні, % (контроль загазованості та блокування системи);
- $T_{rem}$  — розрахунковий час автономної роботи, хв (віртуальний параметр, що обчислюється системою на основі поточного енергоспоживання

Вектор керуючих впливів  $U$  складається з вихідних параметрів ПЛК, що безпосередньо змінюють робочий режим виконавчих механізмів:

- $S_{heater}$  — стан газового пальника (дискретний параметр: 1 — розпал/горіння, 0 — зупинка).
- $\omega$  — відносна частота обертання ротора циркуляційного насоса, % (змінюється алгоритмом для економії електроенергії)

### 2.3 Математична модель підсистеми автономного енергозабезпечення

Оскільки ключовою інновацією даної роботи є адаптація роботи котельні до тривалих вимкнень електропостачання, фундаментальною частиною математичного опису є модель динаміки розряду акумуляторної батареї. З огляду на використання акумуляторів у комплексі з гібридним інвертором,

динаміка рівня заряду описується як дискретний процес інтегрування енергії, що споживається обладнанням котельні за певний проміжок часу.

Загальна електрична потужність  $P_{total}$ , яка споживається котельнею в режимі автономного живлення від інвертора, є сумою потужностей основних виконавчих механізмів та засобів автоматизації, що є змінною величиною і залежить від поточної частоти насоса  $\omega$  (2.1).

$$P_{total}(t) = P_{pump}(\omega) + P_{burner\_solenoid} + P_{auto} \quad (2.1)$$

де  $P_{pump}(\omega)$  — потужність циркуляційного насоса, що є нелінійною функцією від відносної частоти обертання  $\omega$ ,  $P_{burner\_solenoid}$  — потужність, необхідна для утримання газових клапанів у відкритому стані під час горіння пальника (діє при  $S_{heater} = 1$ );  $P_{auto}$  — постійна потужність споживання засобів автоматизації, що є незмінною.

Математична модель зміни рівня заряду акумуляторної батареї за інтервал часу моделювання описується дискретним рівнянням (2.2).

$$SOC_{t+1} = SOC_t - \frac{P_{total}(t) \cdot \Delta t}{E_{ups\_max}} \cdot 100\% \quad (2.2)$$

де  $SOC_{t+1}$  — рівень заряду акумулятора на наступному кроці моделювання, %;  $SOC_t$  — поточний рівень заряду, %;  $\Delta t$  — крок інтегрування моделей (часовий інтервал), год;  $E_{ups\_max}$  — максимальна енергоємність встановленого масиву акумуляторних батарей.

Для розрахунку віртуального параметра  $T_{rem}$  (розрахунковий час автономної роботи, що залишився), система автоматизації використовує наступну математичну залежність (2.3).

$$T_{rem}(t) = \frac{SOC_t \cdot E_{ups\_max}}{P_{total}(t) \cdot 100\%} \cdot 60 \quad (2.3)$$

Це дозволяє системі динамічно перераховувати залишковий час автономності при кожній зміні частоти насоса  $\omega$ , що реалізується адаптивними алгоритмами. Отримані рівняння (2.2) та (2.3) закладені в математичне ядро цифрового двійника для імітації енергетичних процесів під час блекаутів

## 2.4 Математична модель енергоспоживання циркуляційного насоса

Основною проблемою забезпечення життєздатності котельні під час тривалих відключень електроенергії є стрімке виснаження ємності джерел безперебійного живлення. Найбільш енергоємним вузлом системи, який вимагає безперервного функціонування для запобігання тепловим ударами та замерзанню контуру, є циркуляційний насос.

У класичних системах насос працює на номінальній потужності незалежно від умов. Адаптивний алгоритм, що розробляється в даній роботі, передбачає динамічне керування продуктивністю насоса за допомогою частотно-регульованого приводу (ЧРП). Згідно з законами подібності для відцентрових насосів, споживана потужність насоса  $P$  має кубічну залежність від відносної частоти обертання його робочого колеса  $\omega$ .

$$P(\omega) = P_{nom} \cdot \left(\frac{\omega}{\omega_{nom}}\right)^3 \quad (2.4)$$

де  $P_{nom}$  - номінальна електрична потужність насоса, Вт;  $\omega_{nom}$  - номінальна частота обертання (або 100% керуючого сигналу ШІМ);  $\omega$  — поточна задана алгоритмом частота обертання.

Аналіз рівняння (2.4) яскраво демонструє високий потенціал енергозбереження: зниження швидкості насоса лише на 20% ( $\omega = 0.8 * \omega_{nom}$ ) призводить до падіння миттєвого енергоспоживання майже вдвічі.

Оскільки адаптивний алгоритм керування постійно змінює швидкість обертання насоса залежно від умов, загальне споживання електроенергії  $E_{pump}$  не є статичним. Його необхідно розраховувати як інтеграл миттєвої потужності за весь час автономної роботи  $t_{auto}$ .

$$E_{pump} = \int_0^{t_{auto}} P(\omega(t)) dt \quad (2.5)$$

Кінцевою метою розрахунку є визначення прогнозованого часу автономної роботи котельні  $t_{auto}$ . Він обчислюється як відношення доступної енергії акумуляторних батарей до сумарної динамічної потужності системи:

$$t_{auto} = \frac{E_{ups} \cdot SOC \cdot \eta_{inv}}{P_{sys} + \frac{1}{t_{auto}} \cdot \int_0^{t_{auto}} P(\omega(t)) dt} \quad (2.6)$$

де  $E_{ups}$  — номінальна енергоемність батареї, Вт·год; SOC — поточний рівень заряду (від 0 до 1);  $\eta_{inv}$  — ККД інвертора;  $P_{sys}$  — базова незмінна потужність автоматики (ПЛК, клапани, шлюз), Вт.

## 2.5 Математична модель теплової динаміки об'єкта керування

Застосування адаптивних алгоритмів зі зниженням температурних уставок та зменшенням продуктивності циркуляційних насосів змінює температурний режим об'єкта. У фазах, коли алгоритм двопозиційного регулювання вимикає пальник для економії енергії, система переходить у режим вільного охолодження.

Для забезпечення безпеки трубопровідної мережі від замерзання теплоносія, необхідно безперервно аналізувати швидкість втрати тепла. Процес охолодження приміщення або теплоносія в замкнутому контурі описується диференціальним рівнянням теплового балансу на основі закону охолодження Ньютона-Ріхмана (2.7).

$$\frac{dT_{room}}{dt} = -k \cdot (T_{room}(t) - T_{ext}) \quad (2.7)$$

де  $T_{room}(t)$  — поточна температура всередині приміщення(або температура теплоносія  $T_{in}$ ), °C;  $T_{ext}$  — температура зовнішнього середовища, °C;  $k$  — зведений коефіцієнт швидкості охолодження (величина, обернена до постійної часу теплової інерції будівлі), що комплексно враховує площу огорожувальних конструкцій, теплоізоляцію та масу теплоносія, 1/год;  $t$  — час, що минув з моменту зупинки генерації тепла, год.

Розв'язком цього диференціального рівняння є експоненціальна функція (2.8), яка дозволяє розрахувати температуру всередині об'єкта в будь-який момент часу  $t$ .

$$T_{room}(t) = T_{ext} + (T_{init} - T_{ext}) \cdot e^{-k \cdot t} \quad (2.8)$$

де  $T_{init}$  – початкова температура в приміщенні на момент виникнення аварії.

З наведеної експоненціальної залежності випливає, що швидкість падіння температури прямо пропорційна різниці температур ( $T_{init} - T_{ext}$ ). Чим нижча температура на вулиці, тим стрімкіше остигає система.

З іншого боку, з позицій класичної теорії автоматичного управління [10], для синтезу алгоритмів та налаштування регуляторів котел розглядається як динамічна ланка спрямованої дії. Динаміка нагрівання теплоносія при активації газового пальника математично описується передавальною функцією аперіодичної ланки першого порядку із транспортним запізненням (2.9).

$$W(s) = \frac{T_{sup}(s)}{S_{heater}(s)} = \frac{K}{T \cdot s + 1} e^{-\tau s} \quad (2.9)$$

де  $K$  – коефіцієнт підсилення об'єкта (відображає максимальну зміну температури при зміні потужності), °C;  $T$  — постійна часу теплової інерції котла, що залежить від водяного об'єму, с;  $\tau$  — час чистого транспортного запізнення, зумовлений швидкістю руху теплоносія від теплообмінника до датчиків температури, с.

## 2.6 Імітаційне моделювання об'єкта в середовищі MATLAB/Simulink

Для верифікації розроблених математичних моделей та наочної візуалізації перехідних процесів було застосовано середовище візуального імітаційного моделювання MATLAB/Simulink. Такий підхід дозволяє перевірити адекватність теоретичних розрахунків та підтвердити ефективність запропонованих алгоритмів керування до їх імплементації на реальному апаратному забезпеченні. Імітаційна модель розділена на дві незалежні

підсистеми: модель теплової динаміки котла та модель енергоспоживання системи автономного живлення.

Для побудови кривої розгону котла КОЛВІ 250 було зібрано структурну схему на базі передавальної функції об'єкта, рисунок Рисунок 2.9.

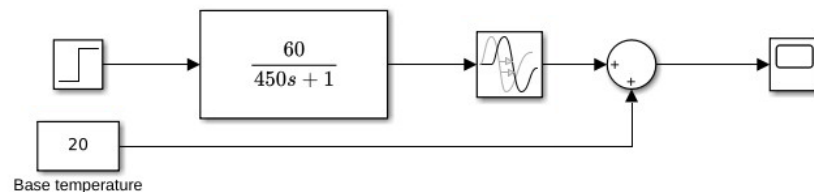


Рисунок 2.9— Simulink-модель для побудови кривої розгону котла КОЛВІ 250

Керуючий вплив (увімкнення пальника) імітується блоком генерації одиничного стрибка *Step*. Теплова інерція маси теплоносія реалізована за допомогою блоку *Transfer Fcn* (аперіодична ланка першого порядку  $W(s) = \frac{60}{450s+1}$ ), а транспортне запізнення в трубах — блоком *Transport Delay*. Для приведення результату до реальних фізичних умов за допомогою суматора до графіка додається початкова температура теплоносія (20 °C). Отриманий перехідний процес, зображений на рисунку Рисунок 2.10, підтверджує плавний вихід об'єкта на робочий температурний режим без перерегулювання, що є критично важливим для запобігання тепловим ударам у контурі.

Числові значення параметрів передавальної функції для моделювання були визначені аналітичним шляхом на основі паспортних даних обраного обладнання. Коефіцієнт підсилення  $K = 60$  відповідає розрахунковому нагріву теплоносія від кімнатної температури до робочої уставки (з 20°C до 80°C). Постійна часу  $T = 450$  с відображає теплову інерцію нагрівання 350 літрів води у котлі КОЛВІ 250 при номінальній потужності пальника. Транспортне запізнення  $\tau = 30$  с визначено виходячи з середньої швидкості циркуляції теплоносія від теплообмінника до вимірювального зонда датчика РТ100.

Для демонстрації енергоспоживання циркуляційного насоса було побудовано імітаційну модель кубічної залежності споживаної потужності від

частоти обертання ротора, рисунок Рисунок 2.11. Блок часу імітує зміну частоти від 0 до 50 Гц. За допомогою математичних операторів реалізується кубічна залежність за законом афінної подібності: відносна частота підноситься до третього степеня та множиться на номінальну потужність насоса (410 Вт).

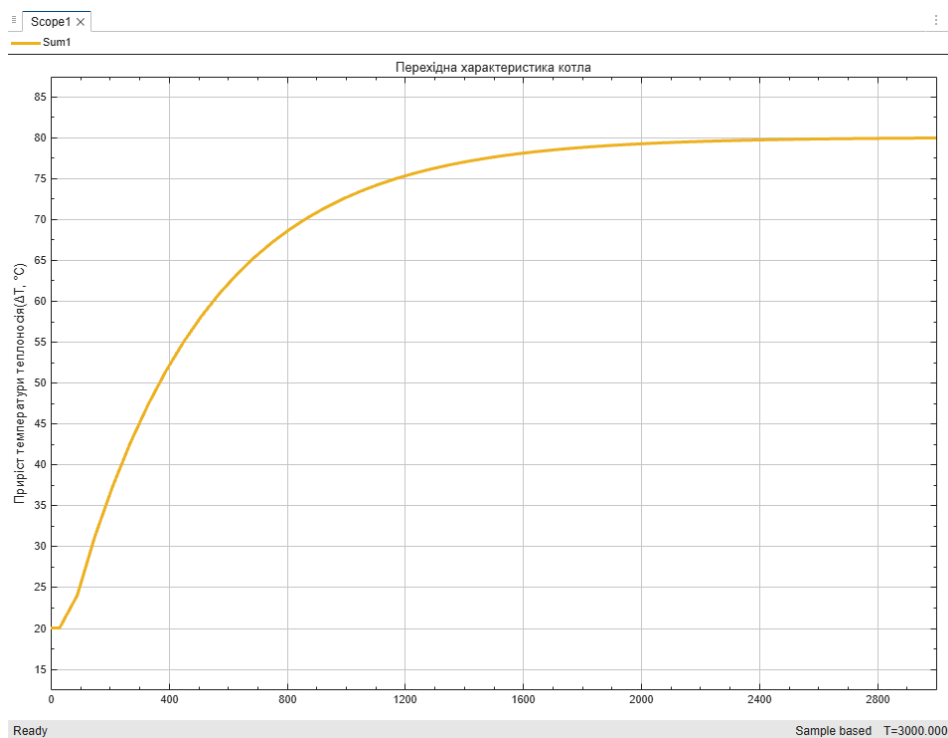


Рисунок 2.10— Перехідна характеристика нагрівання теплоносія в котлі

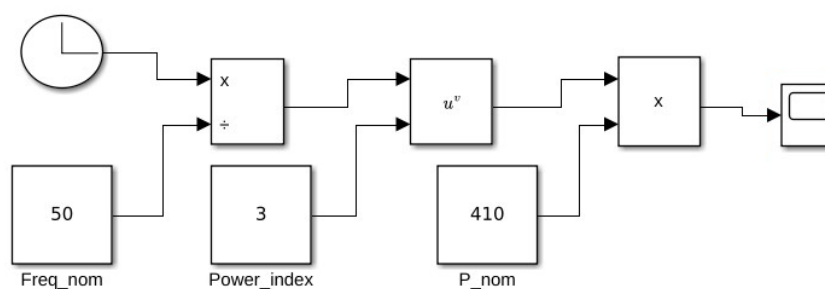


Рисунок 2.11— Simulink-модель для моделювання кубічної залежності споживаної потужності від частоти обертання ротора

Як видно з рисунку Рисунок 2.12, при номінальній частоті 50 Гц насос споживає паспортні 410 Вт. Програмне зниження керуючого впливу до 30 Гц (60% продуктивності) призводить до стрімкого падіння електричного

навантаження до 88,56 Вт. Саме ця нелінійна різниця є фізичним підґрунтям для забезпечення більш тривалої автономної роботи котельні у порівнянні з класичним релейним керуванням.

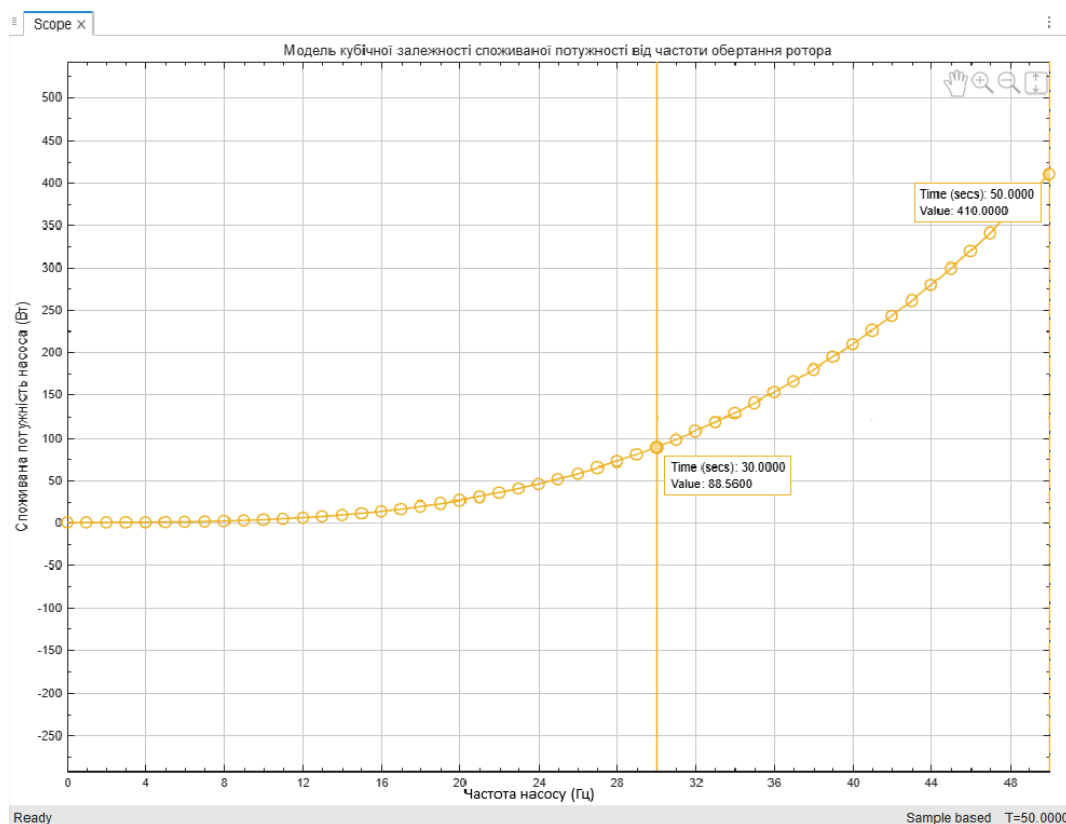


Рисунок 2.12— Результати моделювання споживаної потужності від частоти обертання ротора

Щоб продемонструвати, як саме кубічна залежність споживаної потужності відцентрового насоса від частоти обертання його ротора, була розроблена структурна схема, що паралельно моделює розряд акумуляторного масиву в двох режимах: класичному та адаптивному. Структурна схема моделі представлена на рисунку Рисунок 2.13.

Основу моделі складає математичний апарат інтегрування потужності за часом (блок *Integrator 1/s*), що дозволяє обчислювати спожиту енергію. Блоки підсилення *Gain* виконують масштабування отриманих значень з перетворенням у відсотки заряду з урахуванням загальної ємності батарей. Для імітації фізичних обмежень акумулятора застосовано блоки *Saturation*, які

жорстко обмежують рівень заряду в межах від 0% до 100%. Ключовим елементом схеми, що моделює логіку протоколу *Safe Shutdown*, є блок комутатора *Switch*. У разі, коли поточний заряд падає до критичної позначки 10%, комутатор автоматично відсікає навантаження циркуляційного насоса, переводячи систему в режим мінімального енергоспоживання (150 Вт для підтримки працездатності ПЛК та шлюзу).

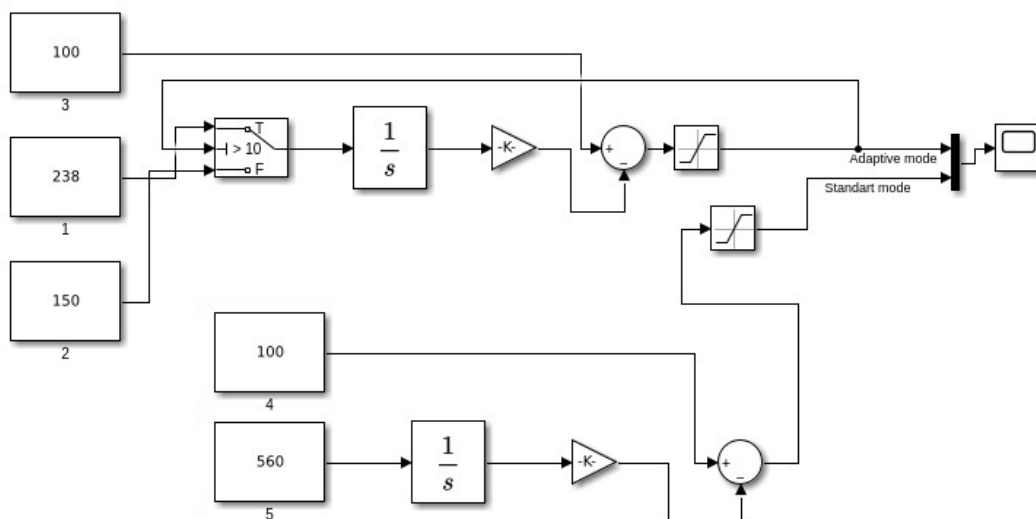


Рисунок 2.13— Simulink-модель для моделювання тривалості роботи від акумулятору

Результати імітаційного моделювання зображено на рисунку Рисунок 2.14. Синя крива, *Standart mode*, ілюструє класичний релейний режим керування на 100 % потужності, за якого повне виснаження резервної ємності настає приблизно за 25 годин. Жовта крива, *Adaptive mode*, демонструє роботу розробленого адаптивного алгоритму зі зниженням частоти насоса до 30 Гц. Завдяки оптимізації енергоспоживання час автономної роботи системи збільшується більш ніж удвічі, досягаючи 54 годин до моменту спрацьовування протоколу безпечної зупинки.

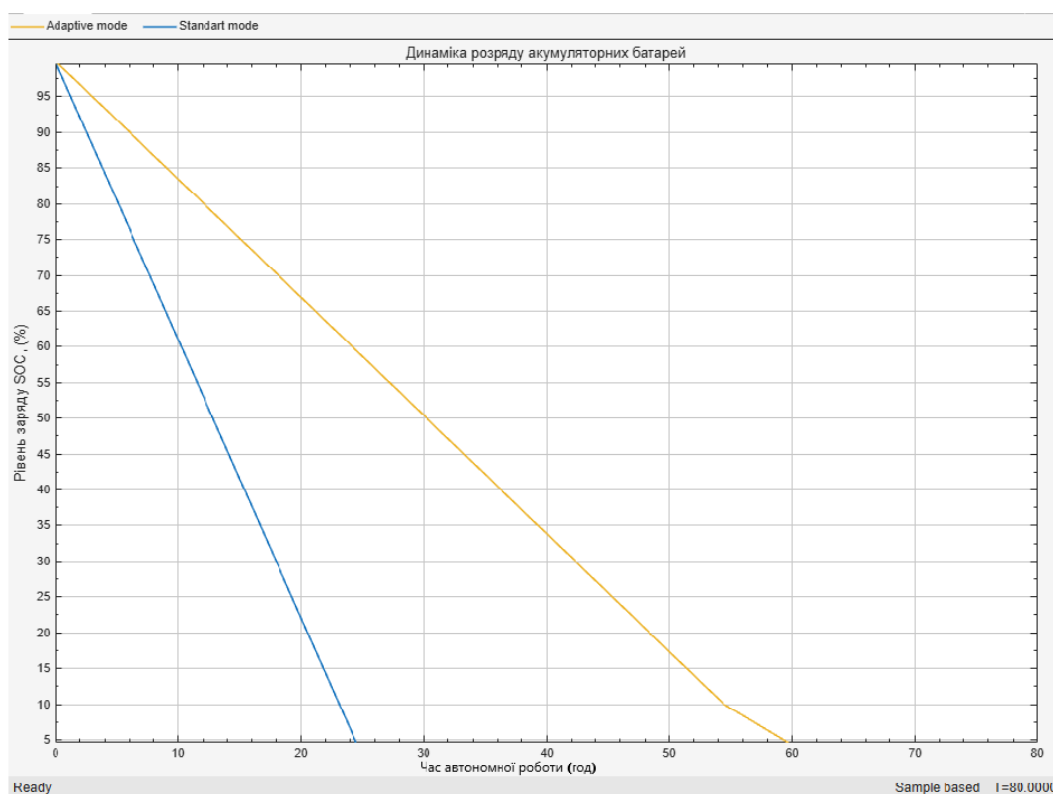


Рисунок 2.14— Моделювання динаміки розряду акумулятора ДБЖ

## 2.7 Алгоритмічне забезпечення системи диспетчеризації та адаптивного керування

На основі розробленої математичної моделі та визначених векторів стану було розроблено комплекс алгоритмів, який програмно реалізує логіку керування газовою котельнею. Архітектура програмного забезпечення побудована за модульним принципом і складається з головного циклу (диспетчера задач) та підпорядкованих підпрограм, що відповідають за конкретні технологічні вузли.

Логічна структура головного циклу керування представлена на рисунку Рисунок 2.15. Фундаментальним принципом побудови даного алгоритму є абсолютний пріоритет функцій безпеки над функціями регулювання. На кожному такті виконання контролер першочергово опитує датчики аварійних станів: перевірка, чи натиснена кнопки аварійної зупинки, контроль загазованості в приміщенні, перевірка гідравлічного тиску в системі на

знаходження в допустимих межах. У разі спрацювання хоча б одного з цих тригерів, виконання алгоритму негайно переривається і система переводиться в стан аварійної зупинки з блокуванням виконавчих механізмів. Якщо параметри безпеки в нормі, система перевіряє наявність напруги в основній мережі. За її наявності та активному автоматичному режимі викликається базова підпрограма двопозиційного регулювання температури. В ручному режимі – очікування команди оператора.

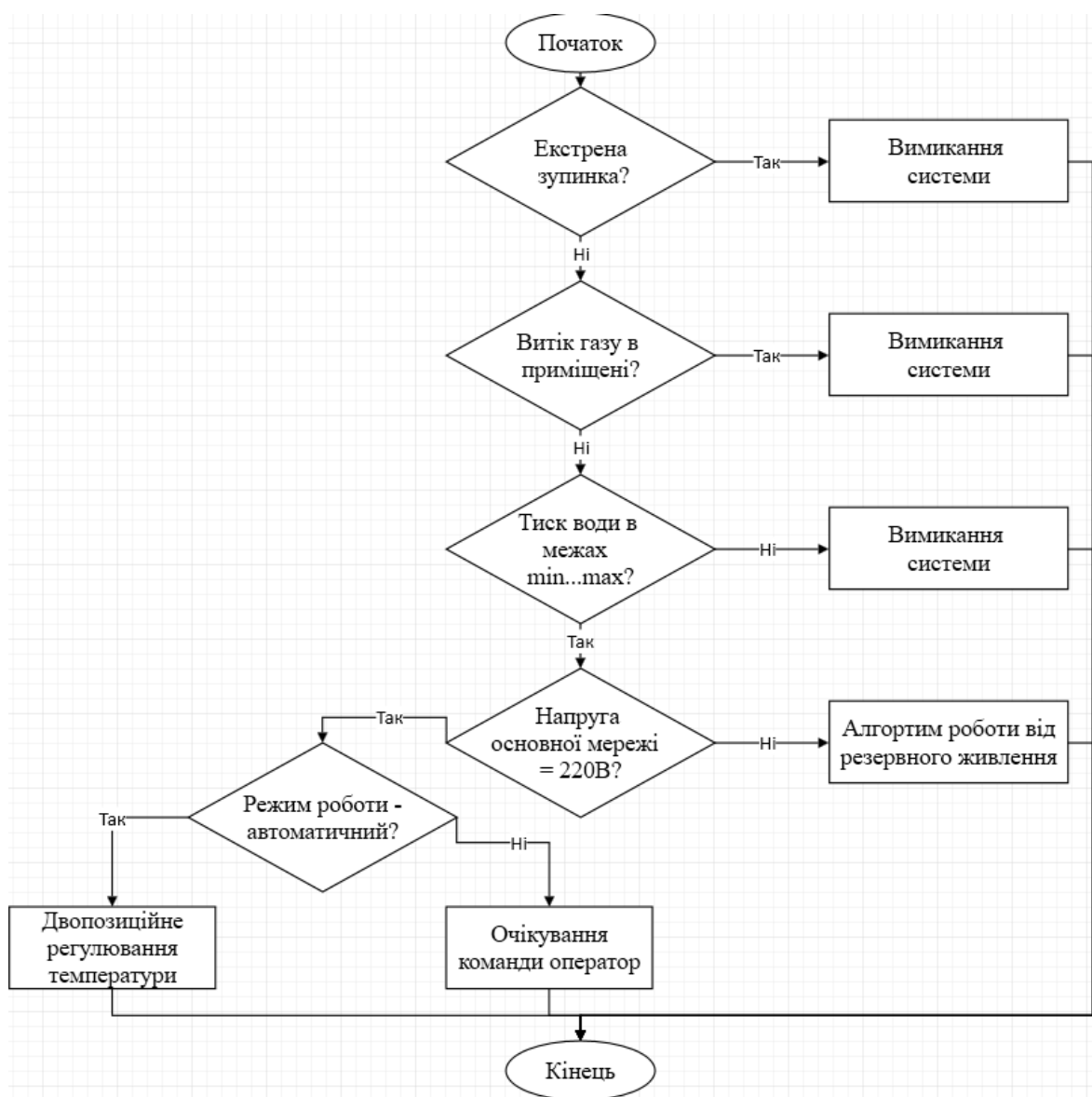


Рисунок 2.15— Блок-схема головного циклу керування котельнею

Двопозиційне регулювання температури, рисунок Рисунок 2.16, реалізує класичний релейний закон керування з гістерезисом. Алгоритм порівнює

поточну температуру теплоносія з нижньою та верхньою уставками. При падінні температури нижче мінімальної ініціюється виклик підпрограми розпалювання котла. При досягненні максимальної уставки — нагрів примусово припиняється.

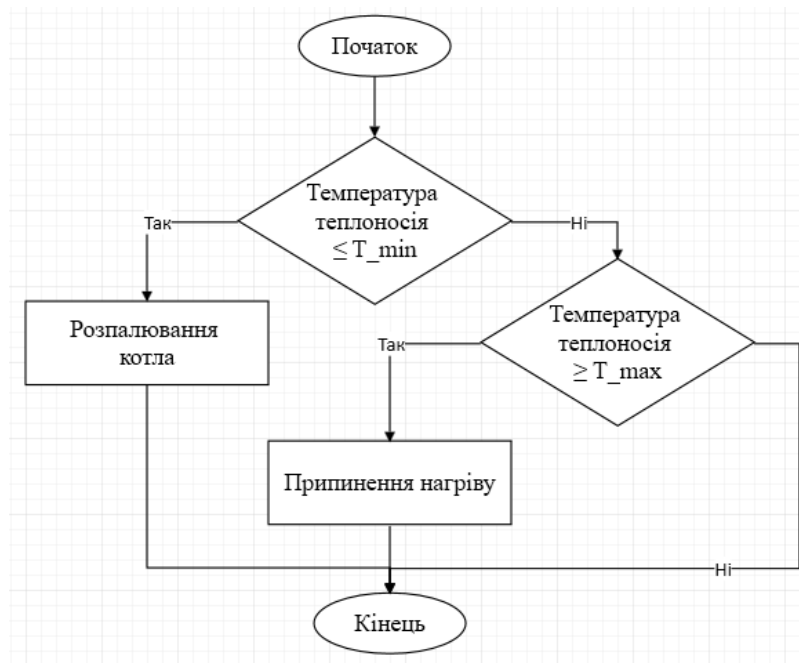


Рисунок 2.16— Блок-схема підпрограми двопозиційного регулювання температури

Розпалювання котла, рисунок Рисунок 2.17, містить критично важливий для газового обладнання таймер затримки. Після подачі команди на відкриття газового клапана та іскроутворення, алгоритм очікує підтвердження наявності факела від оптичного або іонізаційного датчика. Якщо по завершенню таймера полум'я не зафіксовано, система діагностує зрив факела, припиняє подачу газу та блокується для запобігання вибухонебезпечної ситуації.

У випадку зникнення напруги в основній мережі головний цикл передає керування спеціалізованому адаптивному алгоритму роботи від резервного живлення, рисунок Рисунок 2.18. Цей модуль є ключовим інноваційним елементом розробленої системи. Його логіка базується на каскадному опитуванні рівня заряду ДБЖ (SOC) та динамічній зміні режимів роботи.

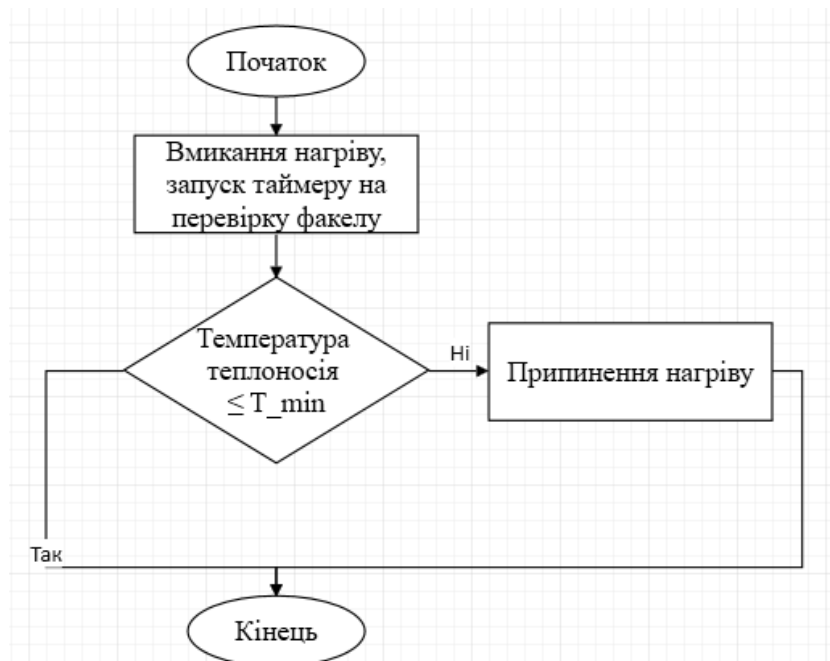


Рисунок 2.17— Блок-схема підпрограми розпалювання котла

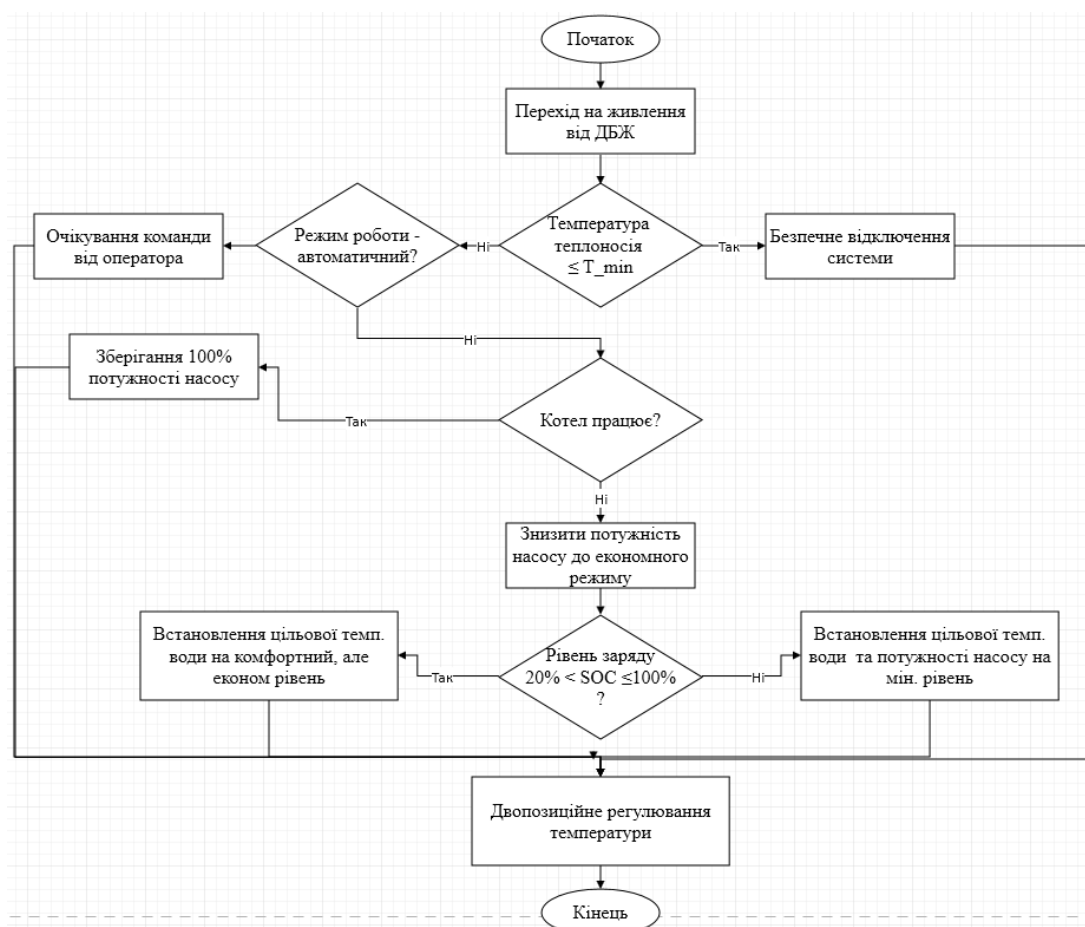


Рисунок 2.18— Блок-схема підпрограми алгоритму роботи від резервного живлення

*Критичний рівень,  $SOC \leq 10\%$ .* Система ініціює процедуру безпечного відключення. Автоматика блокується, відключаються всі енергоємні споживачі, зберігається лише живлення контролера для передачі телеметрії на верхній рівень.

*Енергозберігаючий рівень,  $10\% < SOC \leq 20$ :* Уставка цільової температури теплоносія примусово знижується до "рівня виживання", щоб запобігти аваріям внаслідок замерзання теплоносія, а потужність циркуляційного насоса обмежується до мінімально допустимої частоти обертання.

*Економний комфорт,  $20\% < SOC \leq 100\%$ .* Цільова температура встановлюється на знижений, але комфортний рівень. Потужність насоса переводиться в економний режим.

*Блокування зниження потужності.* Важливим елементом безпеки є перевірка стану котла. Якщо в момент блекауту пальник активно працює, алгоритм тимчасово зберігає 100% потужності насоса для зняття теплового навантаження з теплообмінника, щоб запобігти локальному закипанню води, і лише після вимкнення пальника переходить в економний режим.

## 2.8 Програмна реалізація у Node-RED

### 2.8.1 Загальна архітектура та концептуальна структура потоків у Node-RED

Ядром системи диспетчеризації та обробки адаптивних алгоритмів виступає середовище візуального подієво-орієнтованого програмування Node-RED. У рамках даного проєкту воно виконує роль багатофункціонального індустріального IoT-шлюзу, центрального процесора логіки "виживання" котельні та високопродуктивного сервера зв'язку з НМІ.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.02.ПЗ | Арк. |
|      |      |          |        |      |                         | 41   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

Вибір платформи Node-RED для реалізації керуючого рівня обумовлений її архітектурними перевагами, що базуються на середовищі виконання Node.js. Завдяки асинхронній однопотоковій моделі введення-виведення, що працює в неблокуючому режимі, середовище здатне обробляти тисячі паралельних подій із мінімальними витратами системних ресурсів. Це є критично важливим для систем автоматизації, де необхідно одночасно виконувати високочастотне опитування промислових регістрів Modbus TCP, проводити інтенсивні математичні розрахунки цифрового двійника та підтримувати стабільні дуплексні WebSocket-з'єднання з багатьма клієнтськими веб-додатками.

Загальна архітектура розроблених потоків (flows) побудована за суворим модульним принципом. Повна логічна схема взаємодії вузлів представлена на рисунку Рисунок 2.19.

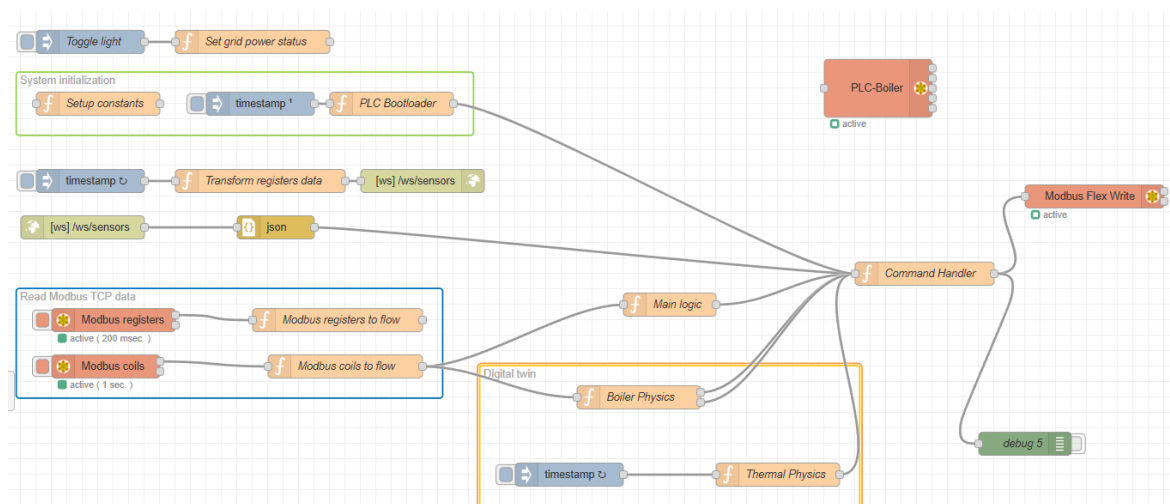


Рисунок 2.19— Загальна структура потоків у середовищі Node-RED

## 2.8.2 Інтеграція взаємодії з ПЛК у Node-RED

Основою взаємодії з нижнім рівнем автоматики є протокол Modbus TCP. Для його реалізації в середовище було інтегровано спеціалізований пакет розширень *node-red-contrib-modbus*. Це дозволило не лише реалізувати функції клієнта для опитування ПЛК, але й розгорнути віртуальний сервер контролера, вузол *PLC-Boiler*. Його налаштування імітують поведінку реального апаратного контролера. Як показано на рисунку, у параметрах вузла задано локальний

мережевий порт 8502 та виділено необхідний об'єм пам'яті. Для всіх типів регістрів (Coils, Holdings, Inputs, Discretes) зарезервовано масиви по 10000 адрес, що з надлишком покриває потреби розробленої карти пам'яті та залишає простір для подальшого масштабування системи.

Рисунок 2.20— Вікно налаштувань віртуального сервера ПЛК

Процес збору телеметрії винесено в окремий асинхронний блок *Read Modbus TCP data*, рисунок 2.21. Відповідні клієнтські вузли зчитування налаштовані на підключення до локального хоста (127.0.0.1) за тим самим портом 8502.

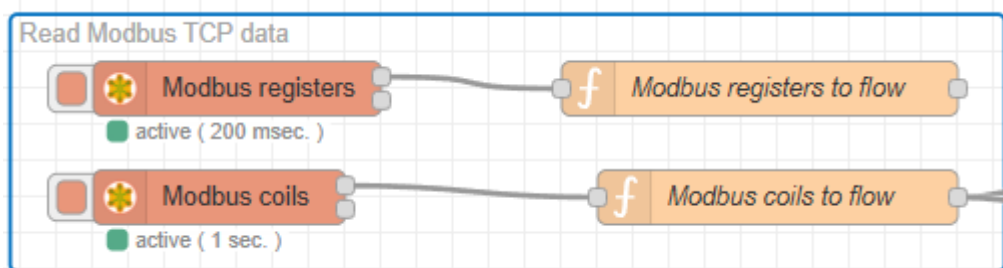


Рисунок 2.21— Вузли Node-RED, відповідальні за збір даних з ПЛК

Опитування контролера розділено на два паралельні потоки залежно від типу даних. Для зчитування аналогових величин(температури, швидкість насоса, рівень заряду) та дискретних станів(статус пальника, наявність напруги мережі) використовується вузол *Modbus Read*. Проте, в першому випадку, його налаштовано на використання стандартної функції *Modbus FC3 (Read Holding Registers)* та високу частототу опитування — кожні 200 мілісекунд, рисунок 2.22.

Рисунок 2.22— Конфігурація вузла зчитування аналогових регістрів

В іншому випадку, вузол використовує функції *FC1 (Read Coils)*, рисунок 2.23. Оскільки ці параметри є менш динамічними, періодичність опитування знижено до 1 секунди задля оптимізації мережевого трафіку.

Отримані масиви даних не передаються одразу в алгоритми керування, а проходять через вузли-обробники (*Modbus registers to flow* та *Modbus coils to flow*), де вони розбираються та записуються у глобальний контекст потоку (*flow.set*). Таке архітектурне рішення гарантує цілісність даних: воно ізолює процес мережевого опитування від логіки обробки, запобігає втраті інформації при перезавантаженнях окремих вузлів та дозволяє будь-якому елементу

системи миттєво отримати актуальний "зріз" стану котельні без очікування наступного циклу опитування.

Рисунок 2.23— Конфігурація вузла зчитування дискретних станів

Для підвищення читабельності коду та забезпечення масштабованості системи в Node-RED реалізовано блок ініціалізації *System initialization*, рисунок 2.24. При запуску системи або розгортанні потоку одноразово спрацьовує вузол *Setup constants*. Він формує карту пам'яті Modbus, закріплюючи за кожним технологічним параметром його фізичну адресу в контролері, після чого ці константи зберігаються у глобальний контекст потоку. Завдяки цьому підходу наступні алгоритмічні вузли звертаються до даних не за "магічними числами", а за семантично зрозумілими іменами (наприклад, *registerKeys.PUMP\_SPEED* замість просто адреси 2). Це мінімізує ймовірність помилок при зверненні до пам'яті ПЛК та значно спрощує подальшу підтримку програмного коду.

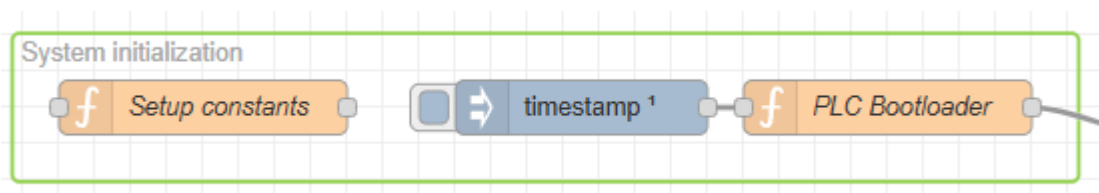


Рисунок 2.24— Блок ініціалізації

Після обробки телеметрії та прийняття рішень, система повинна надсилати керуючі команди назад на виконавчі механізми котельні. Враховуючи, що система підтримує як автоматичний, так і ручний режим керування, виникає ризик конфліктів при одночасному записі даних у ПЛК. Для вирішення цієї проблеми архітектуру потоку побудовано за патерном "єдиної точки входу". Усі команди, незалежно від їхнього джерела, направляються виключно у вузол *Command Handler*.

Цей вузол виконує роль маршрутизатора та нормалізатора. Коли він отримує стандартизовану команду від клієнта (JavaScript-об'єкт із полями цільової дії та значення), він звертається до попередньо ініціалізованого словника констант *modbusRegistersKeys*. Цей словник виконує роль транслятора: він динамічно перетворює отриману текстову команду на конкретну фізичну адресу комірки пам'яті ПЛК. Після успішного знаходження адреси формується специфічний об'єкт даних *payload*, готовий для передачі у вузол *Modbus Flex Write*. Приклад програмної реалізації зображена на рисунку 2.25.

```
const modbusRegistersKeys = flow.get('modbusRegistersKeys');
const data = msg.payload;

if (modbusRegistersKeys && modbusRegistersKeys[data.command] !== undefined) {
  msg.payload = {
    value: data.value,
    fc: 6,
    unitid: 1,
    address: modbusRegistersKeys[data.command],
    quantity: 1
  };
  return msg;
}
```

Рисунок 2.25— Обробка команди на зміну параметра та формування об'єкта даних для Modbus Flex Write

У наведеному фрагменті параметр *fc: 6* вказує на використання функції Modbus для запису аналогових величин у 16-бітні регістри зберігання (*Holding Registers*). Для керування дискретними станами формування пакета відбувається з використанням коду функції *fc: 5* (*Write Single Coil*) та окремого

словника катушок. Такий програмний "контракт" робить вузол повністю універсальним: для додавання нових елементів керування на НМІ-панелі достатньо лише розширити відповідний словник, не втручаючись у базову логіку запису Node-RED.

### 2.8.3 Організація двостороннього обміну даними через WebSockets

Для забезпечення диспетчера актуальною інформацією в режимі реального часу та миттєвої реакції на його команди, класичний механізм HTTP-запитів є неефективним через високу затримку та надлишковий трафік. Тому обмін даними між сервером Node-RED та клієнтським веб-додатком реалізовано поверх дуплексного протоколу WebSockets [4]. Точкою входу для клієнтських підключень виступають вузли *WebSocket in* та *WebSocket out*, налаштовані на прослуховування єдиного маршруту. Налаштування вузла *WebSocket in* зображено на рисунку 2.26. Відповідні параметри встановлені і для *WebSocket out*.

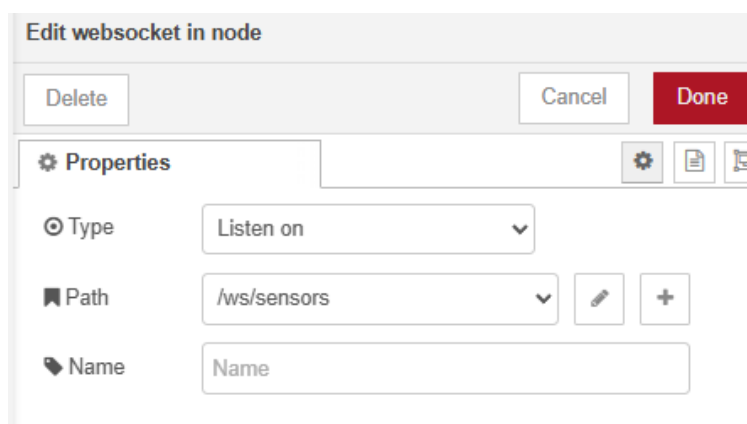


Рисунок 2.26— Налаштування вузла WebSocket in

Процес отримання команд від оператора побудовано за наступним алгоритмом: веб-додаток серіалізує об'єкт команди в JavaScript-об'єкт і відправляє його через сокет. Оскільки мережевий протокол передає дані у вигляді рядків, у Node-RED цей потік проходить через вузол-парсер *json*. Він виконує десеріалізацію, відновлюючи структуру JavaScript-об'єкта, після чого

валідовані дані передаються до маршрутизатора *Command Handler* для запису в контролер.

Процес відправки телеметрії на фронтенд реалізовано через вузол агрегації *Transform registers data*. Він ініціюється вузлом *timestamp* раз на секунду. Це дозволяє уникнути надмірного навантаження на мережу та запобігає "мерехтінню" інтерфейсу, оскільки людське око не здатне ефективно сприймати інформацію, що змінюється частіше.

Під час спрацювання таймера вузол агрегації збирає актуальні значення з глобального контексту потоку та формує єдиний пакет даних для відправки. Код формування даних зображено на рисунку 2.27.

```
1 // Retrieve initialized memory maps and system limits from global context
2 const registerKeys = flow.get('registerKeys');
3 const softwareKeys = flow.get('softwareKeys');
4 const limits = flow.get('systemLimits');
5
6 if (!registerKeys || !softwareKeys) return null;
7
8 const {
9   TEMPERATURE_SUPPLY, TEMPERATURE_RETURN,
10  TEMPERATURE_OUTDOOR, TEMPERATURE_INDOOR,
11  PRESSURE, BATTERY_LEVEL,
12  PUMP_ACTIVE, PUMP_SPEED, HEATER_ACTIVE,
13  FLAME_ACTIVE
14 } = registerKeys;
15
16 const {
17   SYSTEM_STATE, HAS_GRID_POWER, BATTERY_TIME_REMAINING,
18   IS_AUTO_MODE, EMERGENCY_STOP, TEMPERATURE_SETPOINT
19 } = softwareKeys;
20
21 // Determine current temperature setpoints (dynamic from UPS mode or default limits)
22 const currentTemperatureSetpoints = flow.get('ups-temperature-setpoint') ||
23   flow.get(TEMPERATURE_SETPOINT) ||
24   [limits.TEMP_MIN, limits.TEMP_MAX];
25
26 msg.payload = {
27   tempSupply: flow.get(TEMPERATURE_SUPPLY) || 0,
28   tempReturn: flow.get(TEMPERATURE_RETURN) || 0,
29   tempOutdoor: flow.get(TEMPERATURE_OUTDOOR) || 0,
30   tempIndoor: flow.get(TEMPERATURE_INDOOR) || 0,
31   pressure: flow.get(PRESSURE),
32   batteryLevel: flow.get(BATTERY_LEVEL),
33   pumpActive: flow.get(PUMP_ACTIVE) || false,
34   heaterActive: flow.get(HEATER_ACTIVE),
35   flameActive: flow.get(FLAME_ACTIVE),
36   pumpSpeed: flow.get(PUMP_SPEED),
37
38   systemState: flow.get(SYSTEM_STATE),
39   hasGridPower: flow.get(HAS_GRID_POWER),
40   isAutoMode: flow.get(IS_AUTO_MODE),
41   isEmergencyStop: flow.get(EMERGENCY_STOP),
42   temperatureSetpoint: currentTemperatureSetpoints,
43   batteryTimeRemaining: flow.get(BATTERY_TIME_REMAINING)
44 };
45 return msg;
```

Рисунок 2.27— Формування даних системи для відправки по WebSockets

#### 2.8.4 Реалізація алгоритмів керування та цифрового двійника

Логіка адаптивного керування, блок-схема якої зображена на рисунку 2.15, інтегрована в систему у вигляді JavaScript-функції всередині вузла *Main logic*. Даний скрипт циклічно викликається при кожному оновленні вектора телеметрії з програмованого логічного контролера. Замість використання громіздких візуальних гілок базових вузлів Node-RED, уся аналітична частина зосереджена в єдиному вузлі, що значно підвищує швидкодію системи та дозволяє використовувати складні математичні умови без затримок. Основою алгоритму є відстеження статусу основної мережі та каскадна перевірка рівня заряду акумуляторів для динамічної зміни уставок. Фрагмент коду, що ілюструє перехід системи в режим резервного живлення та розрахунок нової частоти насоса, наведено на рисунку 2.28.

```
if (!isGridNormal) {
  if (batteryLevel <= 10) {
    flow.set(softwareKeys.IS_AUTO_MODE, false); // Force manual mode override
    flow.set(softwareKeys.SYSTEM_STATE, "SAFE SHUTDOWN (SOC <= 10%)");
    node.warn("[BLACKOUT] Battery depleted. Initiating safe shutdown sequence.");
    return turnOffSystem();
  }

  // Adaptive setpoint adjustment based on battery capacity and outdoor conditions
  if (batteryLevel > 20) {
    if (tempOutdoor < -10) {
      targetTempMin = 65;
      targetTempMax = 70;
      currentStateStr = "RESERVE (ECO + FROST)";
    } else {
      targetTempMin = 60;
      targetTempMax = 65;
      currentStateStr = "RESERVE (ECO)";
    }
  } else {
    targetTempMin = 40;
    targetTempMax = 45;
    currentStateStr = "RESERVE (CRITICAL)";
  }

  flow.set('ups-temperature-setpoint', [targetTempMin, targetTempMax]);

  // Dynamic circulation pump capacity allocation
  if (isHeaterActive) {
    targetPumpSpeed = 50;
  } else {
    targetPumpSpeed = 20;
  }
  flow.set('ups-pump-speed', targetPumpSpeed);
} else {
  // Restore default setpoints upon grid recovery
  flow.set('ups-temperature-setpoint', undefined);
  flow.set('ups-pump-speed', undefined);
}
```

Рисунок 2.28— Реалізація блок-схеми підпрограми “Алгоритм роботи від резервного живлення”

Оскільки налагодження та випробування адаптивних алгоритмів безпосередньо на діючому технологічному об'єкті пов'язане з високими ризиками виникнення аварійних ситуацій та необґрунтованим зносом обладнання, у середовищі Node-RED було розроблено комплексний імітаційний модуль, що виконує функцію цифрового двійника газової котельні та складається з двох взаємопов'язаних обчислювальних вузлів: *Boiler Physics* та *Thermal Physics*, рисунок 2.29.

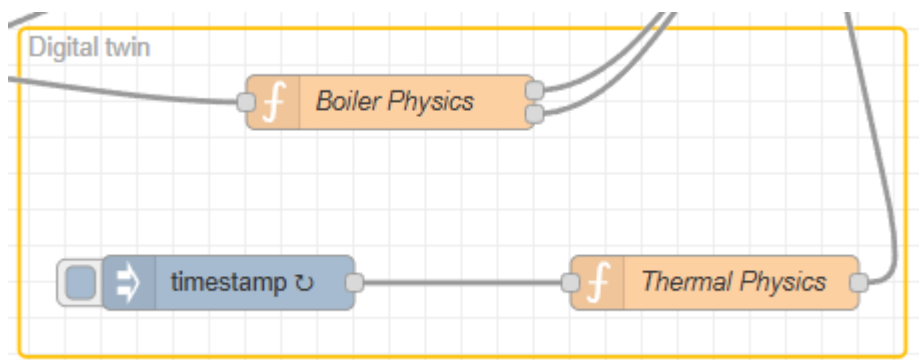


Рисунок 2.29— Вузли цифрового двійника газової котельні

Вузол *Boiler Physics* призначений для моделювання фізичних затримок виконавчих механізмів, які завжди присутні в реальному обладнанні (зокрема, у газовому пальнику котла КОЛВІ 250), але відсутні в ідеальній математичній моделі. Головним завданням вузла є реалістична імітація процесу розпалювання пальника та генерація сигналу наявності факела для ПЛК. Логіка роботи вузла використовує внутрішній контекст пам'яті Node-RED (*context.get* / *context.set*) для реалізації програмного таймера, який зберігає свій стан між тактами виконання програми (інтервал — 1 секунда). Алгоритм відпрацьовує два основні технологічні сценарії: ініціалізація розпалювання та гасіння пальника.

При отриманні від ПЛК команди на активацію нагріву, за умови відсутності полум'я, запускається лічильник часу. Фізика процесу враховує затримку, необхідну для продувки камери згоряння, відкриття газового клапана та іскроутворення. Датчик факела переводиться в активний стан строго на 3-ю

секунду після початку циклу, що дозволяє повноцінно протестувати захисні таймери контролера. В свою чергу, при знятті сигналу нагріву таймер розпалювання миттєво скидається в нуль, а датчик факела вимикається без затримки, що відповідає фізичному процесу перекриття подачі газу. Фрагмент коду вузла *Boiler Physics* зображено на рисунку 2.30.

```

1  const registerKeys = flow.get('registerKeys');
2  const coilsKeys = flow.get('coilsKeys');
3  if (!registerKeys) return null;
4
5  const isHeaterActive = flow.get(registerKeys.HEATER_ACTIVE) === true;
6  const isFlameActive = flow.get(registerKeys.FLAME_ACTIVE) === true;
7
8  // Node-specific memory context to track ignition timing across execution ticks
9  let ignitionTimer = context.get('ignitionTimer') || 0;
10
11  if (isHeaterActive && !isFlameActive) {
12      ignitionTimer += 1;
13      context.set('ignitionTimer', ignitionTimer);
14
15      // Simulate physical delay
16      if (ignitionTimer >= 3) {
17          return { payload: { command: 'flame-sensor', value: true } };
18      }
19  }
20
21  if (!isHeaterActive) {
22      // Reset ignition timer for subsequent cycles
23      context.set('ignitionTimer', 0);
24
25      if (isFlameActive) {
26          return { payload: { command: 'flame-sensor', value: false } };
27      }
28  }
29
30  return null;

```

Рисунок 2.30— Реалізація вузла імітації процесу розпалювання пальника

Вузол *Thermal Physics* є математичним серцем цифрового двійника. Він виконує чисельне інтегрування рівнянь стану системи на кожному такті симуляції, відтворюючи реальну динаміку котельні. У даному вузлі розраховуються теплова динаміка теплообмінника та приміщення, фізика тиску з імітацією закипання, а також енергетична модель акумуляторного масиву ДБЖ, рисунок 2.31.

При активному факелі температура подачі зростає, апроксимуючи розраховану раніше перехідну характеристику аперіодичної ланки (з коефіцієнтом підсилення  $K=60$  та постійною часу  $T=450$ ). У разі вимкнення пальника алгоритм моделює процеси охолодження. Якщо насос працює, відбувається швидке зниження температури за рахунок підмішування холодної води зі зворотної лінії. Якщо насос зупинено — імітується повільне охолодження теплоносія в замкнутому контурі відповідно до закону охолодження Ньютона-Ріхмана. Також розраховуються постійні тепловтрати приміщення через огорожувальні конструкції залежно від різниці внутрішньої та зовнішньої температур.

```
// === 1. HEAT EXCHANGER DYNAMICS ===
if (isFlameActive) {
    tSupply += 1.0;
} else {
    if (isPumpActive) {
        if (tSupply > tReturn) tSupply -= 0.5;
    } else {
        if (tSupply > tIndoor) tSupply -= 0.05;
    }
}

// === 2. HYDRAULIC DISTRIBUTION & CONVECTION ===
if (isPumpActive) {
    let targetReturn = tSupply - 15; // Target delta T

    if (tReturn < targetReturn) {
        tReturn += 0.5;
    } else if (tReturn > targetReturn && tReturn > tIndoor) {
        tReturn -= 0.5;
    }

    if (tReturn > tIndoor) tIndoor += 0.05;
} else {
    // Natural convection simulation (pump stopped)
    if (tSupply > tReturn + 5) {
        tReturn += 0.05;
    } else if (tReturn > tIndoor) {
        tReturn -= 0.05;
    }
}

// === 3. AMBIENT HEAT LOSS ===
if (tIndoor > tOutdoor) {
    tIndoor -= 0.02; // Thermal dissipation through building envelope
}
```

Рисунок 2.31— Імітація теплової динаміки теплообмінника

Алгоритм безперервно розраховує робочий гідравлічний тиск, який пропорційно залежить від теплового розширення рідини, рисунок 2.32. Критично важливим елементом віртуального середовища є програмна імітація аварійного закипання: при перевищенні температурою теплоносія позначки 95 °C починається інтенсивне пароутворення, що викликає різкий нелінійний стрибок тиску. Це дозволяє в безпечному режимі перевірити реакцію автоматики ПЛК на загрозу розриву трубопроводів.

```
// === 4. PRESSURE PHYSICS & BOILING SIMULATION ===
let avgWaterTemp = (tSupply + tReturn) / 2;
let calculatedPressure = 1.2 + (avgWaterTemp * 0.015);

if (avgWaterTemp > 95) {
    calculatedPressure += (avgWaterTemp - 95) * 0.15; // Critical boiling pressure spike
}
```

Рисунок 2.32— Імітація фізика тиску та закипання

Обчислення поточного рівня заряду батареї (SOC) повністю базується на рівняннях енергетичного балансу підсистеми живлення (2.4, 2.5, 2.6), рисунок 2.33. Під час блекауту розраховується струм розряду, який складається з базового споживання електроніки та динамічної потужності циркуляційного насоса. Використання законів афінної подібності (кубічної залежності споживаної потужності від частоти обертання  $\omega$ ) дозволяє в реальному часі симулювати економію енергії. Алгоритм підтверджує, що кероване зниження швидкості насоса радикально зменшує енергоспоживання і подовжує час автономної роботи об'єкта.

Після розрахунків вузол виконує масштабування даних до промислового стандарту (наприклад, множення температур на 10 для запису у 16-бітні цілочисельні регістри) і формує масив повідомлень для оновлення пам'яті віртуального контролера.

```

// === 6. BATTERY DISCHARGE PHYSICS & ESTIMATION ===
let exactSoc = flow.get('internal-soc') ?? flow.get(registerKeys.BATTERY_LEVEL) ?? 100
const isGridNormal = flow.get(softwareKeys.HAS_GRID_POWER) !== false;

if (isGridNormal) {
  // Battery charging phase
  if (exactSoc < 100) {
    exactSoc += 0.5;
    if (exactSoc > 100) exactSoc = 100;
  }
  flow.set(softwareKeys.BATTERY_TIME_REMAINING, null);
} else {
  // Blackout discharge calculation
  let baseDischarge = 0.05; // Base inverter consumption

  if (isFlameActive) baseDischarge += 0.1; // Boiler electronics

  if (isPumpActive) {
    // Cubic power dependency based on affinity laws
    let pumpRatio = currentPumpSpeed > 0 ? (currentPumpSpeed / 50) : 0;
    let pumpFactor = Math.pow(pumpRatio, 3);
    baseDischarge += pumpFactor * 0.25;
  }

  exactSoc -= baseDischarge;
  if (exactSoc < 0) exactSoc = 0;

  // Remaining runtime estimation
  if (baseDischarge > 0) {
    let virtualMinutesRemaining = Math.round(exactSoc / baseDischarge);
    flow.set(softwareKeys.BATTERY_TIME_REMAINING, virtualMinutesRemaining);
  }
}

flow.set('internal-soc', exactSoc);
let roundedSoc = Math.floor(exactSoc);
flow.set(registerKeys.BATTERY_LEVEL, roundedSoc);

```

Рисунок 2.33— Імітація енергетичної моделі акумулятора ДБЖ

## 2.9 Архітектура та програмна реалізація НМІ

### 2.9.1 Вибір стеку веб-технологій та мікрофронтендна архітектура

Для забезпечення диспетчерського контролю за станом газової котельні було відкинуто концепцію використання класичних десктопних SCADA-систем на користь сучасного веб-орієнтованого підходу. Клієнтський додаток реалізовано у форматі *Single Page Application* (SPA) з використанням бібліотеки React [8].

Вибір React є стратегічним рішенням для систем промислового моніторингу. Завдяки механізму *Virtual DOM*, бібліотека здатна точково оновлювати лише ті елементи інтерфейсу (наприклад, значення температури чи тиску), які фактично змінилися, без перемальовування всієї сторінки. Це гарантує відсутність візуального "мерехтіння" та мінімізує навантаження на процесор клієнтського пристрою навіть при високочастотному надходженні телеметрії. У якості бібліотеки UI-компонентів використано *Ant Design*. Її строгий, індустріальний дизайн та наявність готових високонадійних

компонентів (модальні вікна, сітки, індикатори) дозволили значно прискорити розробку інтерфейсу та забезпечити його кросплатформеність.

Ключовою інноваційною особливістю розробленого НМІ є відмова від монолітної структури на користь архітектури мікрофронтендів [26]. Цей підхід реалізовано за допомогою інструменту збірки *Vite* та плагіна *Vite Plugin Federation* [9]. Інтерфейс диспетчера фізично розділений на незалежні модулі (модуль керування насосами, модуль моніторингу ДБЖ), які розробляються та компілюються окремо, а під час роботи безшовно інтегруються в єдину оболонку через механізм *remoteEntry*. Це забезпечує високу відмовостійкість системи на рівні UI: у випадку критичної помилки або падіння одного з віджетів, решта інтерфейсу продовжує функціонувати в штатному режимі. На даному етапі життєвого циклу додатка для повної ініціалізації відновленого модуля передбачено оновлення сесії (перезавантаження сторінки клієнта), що є допустимим компромісом для статичної резолюції федеративних модулів.

Для забезпечення суворої типізації та консистентності даних між різними мікрофронтендами було створено спільний робочий простір — модульну директорію *packages*. У ній зосереджені єдині TypeScript-інтерфейси, типи телеметрії та константи. Завдяки цьому будь-який віджет системи завжди оперує актуальними контрактами даних.

Як вже було зазначено у пункті 2.8.3, для встановлення дуплексного зв'язку з сервером Node-RED на клієнті використовується нативний API WebSockets. А усі керуючі впливи від оператора формуються у вигляді звичайного JavaScript-об'єкта, який містить ідентифікатор команди та її значення: Щоб забезпечити коректну відправку команд та стандартизувати звернення до Node-RED, було створено TypeScript інтерфейси-контракти, рисунок 2.34. Такий типізований підхід дозволяє серверу Node-RED миттєво розпізнати команду через словник констант та виконати безпечний запис у відповідний регістр ПЛК. Крім того, на етапі компіляції клієнтського додатка TypeScript гарантує, що розробник не зможе, наприклад, випадково передати

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.02.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                         | 55   |

текст замість числа при встановленні температурної уставки, що додатково підвищує загальну надійність людино-машинного інтерфейсу.

```
export const SocketEvent = {
  CONNECTION: "connection",
  DISCONNECT: "disconnect",
  SET_HEATER: "set-heater",
  SET_PUMP: "set-pump",
  SET_TEMPERATURE_SETPOINT: "set-temperature-setpoint",
  SET_AUTO_MODE: "set-auto-mode",
  EMERGENCY_STOP: "emergency-stop",
} as const;

export type SocketCommand = (typeof SocketEvent)[keyof typeof SocketEvent];
export type SocketMessageValue = number | string | boolean | [number, number];

export interface SocketMessage {
  command: SocketCommand;
  value: SocketMessageValue;
}
```

Рисунок 2.34— TypeScript інтерфейси-контракти для формування повідомлень до Node-RED

## 2.9.2 Візуалізація технологічного процесу та структура НМІ

Для забезпечення максимальної ергономіки та зниження когнітивного навантаження на диспетчера, архітектуру користувацького інтерфейсу побудовано за класичними стандартами проєктування промислових SCADA-систем [24]. Робочий простір веб-додатка розділено на три основні функціональні зони: глобальну панель керування, *Header*, навігаційну панель, *Sidebar*, та головну робочу область для відображення динамічних мікрофронтендів, *Main Workspace*.

Верхня панель інтерфейсу зафіксована на екрані та залишається видимою незалежно від обраної вкладки. Вона містить критично важливі елементи керування та індикації, що потребують миттєвого доступу:

- *Системний час та статус зв'язку*, дозволяє оператору контролювати актуальність даних. У разі обриву WebSocket-з'єднання індикатор миттєво сигналізує про втрату зв'язку з контролером.

- *Індикатор ДБЖ* для відображення поточного відсотка заряду акумулятора. Оскільки робота в умовах блекауту є критичним режимом, цей параметр винесено в глобальну зону видимості.
- *Селектор режиму керування (Auto / Manual)*, завдяки якому диспетчер може перехопити керування системою для проведення сервісних робіт або примусового коригування алгоритмів.
- *Кнопка аварійної зупинки*, натискання якої генерує команду екстреного блокування виконавчих механізмів, ігноруючи поточні уставки.

Ліва панель забезпечує перемикання між основними екранами системи. Маршрутизація реалізована на базі компонентів *React Router*, що дозволяє миттєво підміняти контент головної робочої області. Окрім базового екрана диспетчеризації, бокове меню надає доступ до спеціалізованої вкладки моніторингу системи резервного живлення, де зосереджена детальна інформація про поточний стан акумуляторних батарей та розрахунковий час автономної роботи об'єкта. Для глибокого аналізу технологічних процесів розроблено модуль тренд-аналітики, який відображає історичну динаміку зміни параметрів у вигляді інтерактивних графіків. Додатково в системі виділено окрему сторінку журналу подій, що призначена для зручної табличної фіксації попереджень, критичних аварійних повідомлень датчиків безпеки та повної хронології дій оператора.

### 2.9.3 Функціональне призначення та опис технологічних екранів

Головний *Dashboard* слугує базовою точкою входу і призначений для оперативного контролю за всіма ключовими параметрами котельні, рисунок 2.35. Зважаючи на необхідність швидкого зчитування інформації диспетчером, його інтерфейс побудовано за панельно-блоковим принципом. Такий підхід дозволяє уникнути візуального перевантаження, притаманного застарілим

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.02.ПЗ | Арк. |
|      |      |          |        |      |                         | 57   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

мнемосхемам. Інформація на екрані логічно згрупована за технологічними вузлами в окремі картки.

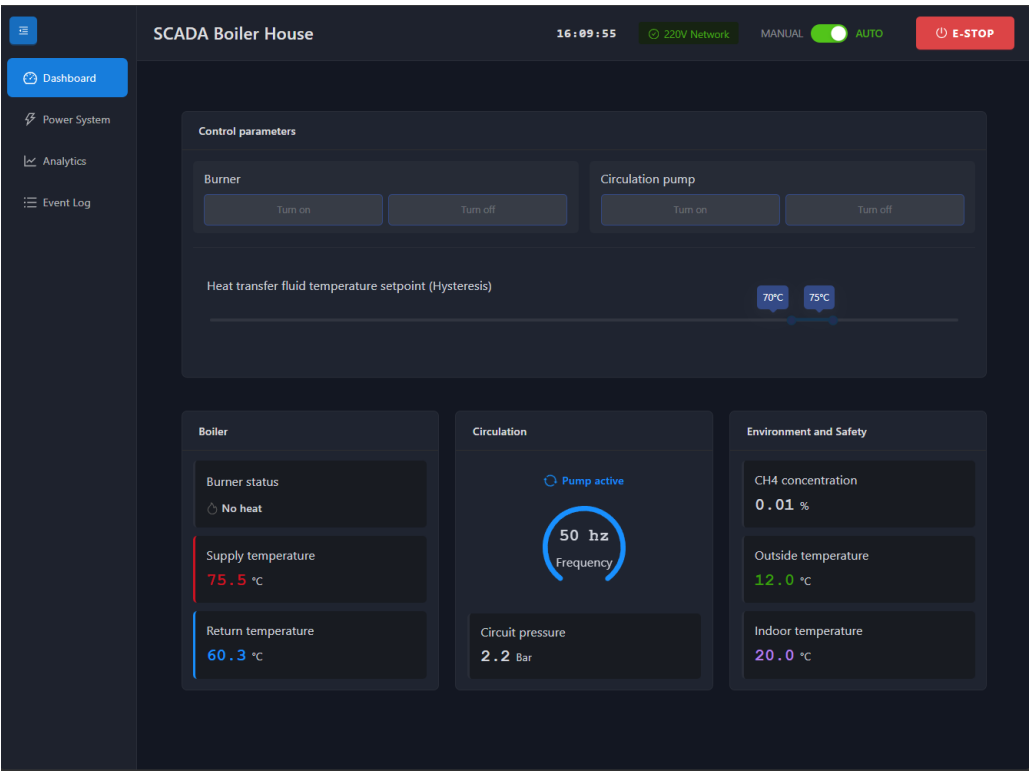


Рисунок 2.35— Інтерфейс головного дашборда оператора

Зона температурного режиму відображає поточні значення на подавальному та зворотному трубопроводах, а також температурний фон всередині приміщення і на вулиці з використанням адаптивного кольорового кодування. Блок гідравліки та контурів безпеки містить дані про актуальний тиск теплоносія в системі та показання датчика загазованості метаном, причому наближення цих параметрів до критичних або аварійних меж супроводжується візуальною пульсацією відповідних компонентів для привернення уваги. Окремо виведено інформаційну панель стану виконавчих механізмів, яка відображає дискретні статуси роботи газового пальника та активності циркуляційного насоса.

Відповідно до розробленої концепції експлуатаційної безпеки, на головному дашборді свідомо відсутні елементи тонкого ручного налаштування, зокрема регулятори частоти обертання насоса. Це архітектурне рішення втілює

програмний патерн захисту від некомпетентного втручання: оскільки оптимізація енергоспоживання під час тривалих блекаутів повністю покладена на серверний адаптивний алгоритм, диспетчер не має можливості випадково порушити енергетичний баланс будівлі. Натомість оператору доступна загальна інформація про поточну алгоритмічну уставку цільових температур та режим роботи системи.

Інтерфейсний модуль *Power System* призначений для детального аналізу стану підсистеми автономного енергозабезпечення котельні в умовах аварійного відключення основного живлення, рисунок 2.36. Візуалізація цього екрана повністю сфокусована на відображенні динамічних параметрів джерела безперебійного живлення та акумуляторних батарей. Центральним елементом інтерфейсу виступає графічний віджет поточного рівня заряду акумуляторів, інтегрований із колірною індикацією стану, яка динамічно змінює свій колір залежно від ступеня деградації резерву енергії.

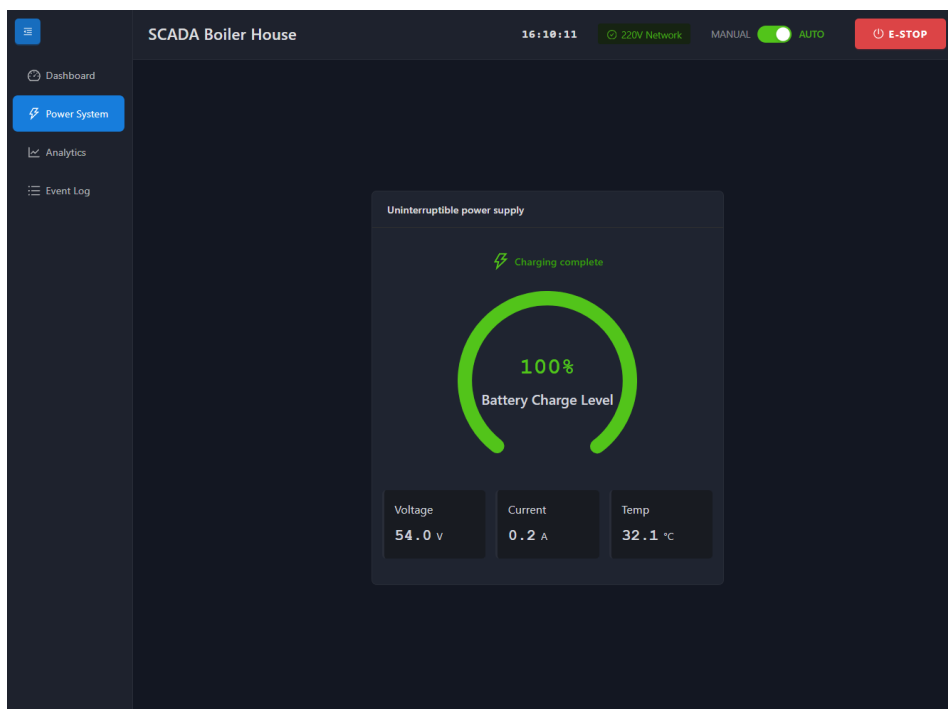


Рисунок 2.36— Інтерфейс вкладки моніторингу енергосистеми котельні

Головною аналітичною та інженерною особливістю цього мікрофронтенду є виведення інтелектуального прогнозу часу автономної

роботи, що залишився до повного виснаження системи. Цей показник розраховується на основі раніше описаного математичного ядра в Node-RED, яке враховує нелінійне кубічне зниження споживання циркуляційного насоса при адаптивному керуванні. Диспетчер має можливість у реальному часі спостерігати за процесом накопичення енергії або за швидкістю розряду системи під час блекауту. Таке архітектурне рішення забезпечує оператора вичерпною інформацією про поточний енергетичний баланс об'єкта, дозволяючи точно прогнозувати час безпечної роботи котельні в автономному режимі.

Наступний модуль графічного аналізу *Analytics* забезпечує оператора інструментарієм для глибокого моніторингу динаміки технологічних процесів, рисунок 2.37. На відміну від головного дашборда, який відображає виключно миттєвий зріз поточних значень, сторінка трендів акумулює часові ряди даних та будує на їх основі інтерактивні графіки. Головними технологічними параметрами для відстеження обрано температурні показники подавальної та зворотної ліній теплоносія, а також криву деградації заряду акумуляторних батарей. Візуалізація саме цих величин дозволяє наочно оцінити адекватність роботи цифрового двійника та ефективність адаптивних алгоритмів економії енергії.

Останнім модулем інтерфейсу є сторінка *Event Log*, рисунок 2.38. Вона призначена для ретроспективного аналізу роботи системи, проведення аудиту дій експлуатаційного персоналу та фіксації хронології виникнення аварійних режимів. Інтерфейс сторінки виконано у вигляді динамічної таблиці, яка автоматично оновлюється при надходженні нових сповіщень.

Кожен запис у журналі має сувору уніфіковану структуру даних, яка містить точну дату та час події з точністю до секунди, категорію або тип команди, а також ініціатора події. У якості автора фіксується або диспетчер, або автоматизована система. Аварійні ситуації, такі як критичний рівень загазованості приміщення метаном або глибокий розряд акумуляторних

батареї, обробляються як окремий пріоритетний клас подій і обов'язково містять текстовий опис конкретної першопричини активізації захисту для полегшення діагностики.

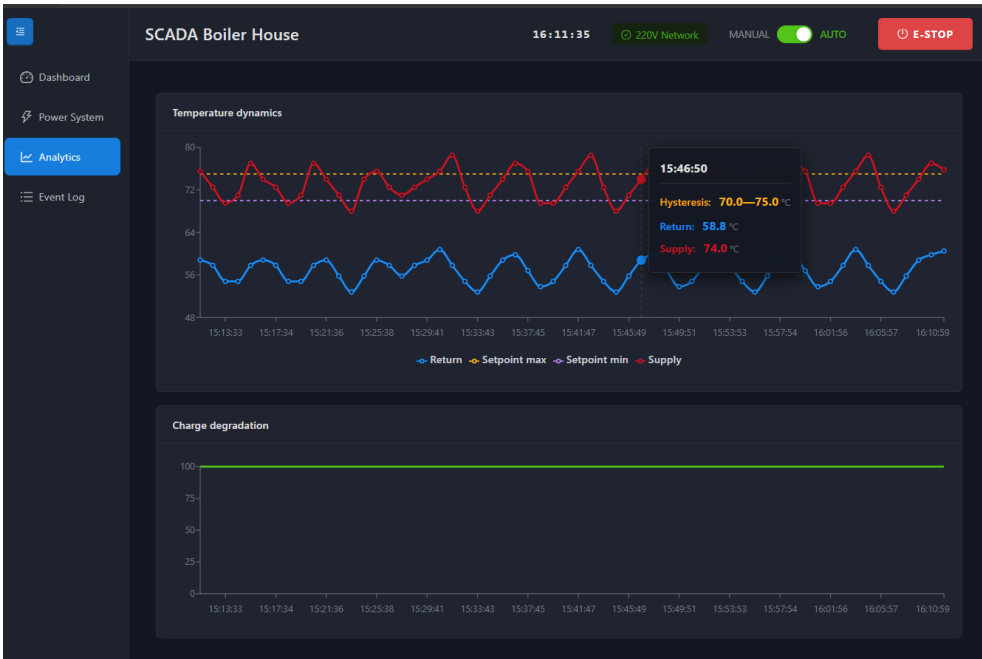


Рисунок 2.37— Інтерфейс модуля тренд-аналітики технологічних параметрів

The figure shows a SCADA interface for a boiler house. The main table, titled 'Event Log', displays a list of events. The columns are: Date and time, Category, Initiator, and Message (Event description). The events are as follows:

| Date and time          | Category | Initiator  | Message (Event description)                      |
|------------------------|----------|------------|--------------------------------------------------|
| 15:51:53<br>16.05.2026 | INFO     | AUTOMATION | Flame sensor: IGNITION DETECTED                  |
| 15:51:50<br>16.05.2026 | COMMAND  | AUTOMATION | Equipment control: Main Boiler Heater turned ON  |
| 15:51:31<br>16.05.2026 | INFO     | AUTOMATION | Flame sensor: NO FLAME                           |
| 15:51:30<br>16.05.2026 | COMMAND  | AUTOMATION | Equipment control: Main Boiler Heater turned OFF |
| 15:51:20<br>16.05.2026 | INFO     | AUTOMATION | Flame sensor: IGNITION DETECTED                  |
| 15:51:17<br>16.05.2026 | COMMAND  | AUTOMATION | Equipment control: Main Boiler Heater turned ON  |
| 15:50:58<br>16.05.2026 | INFO     | AUTOMATION | Flame sensor: NO FLAME                           |
| 15:50:57<br>16.05.2026 | COMMAND  | AUTOMATION | Equipment control: Main Boiler Heater turned OFF |
| 15:50:47<br>16.05.2026 | INFO     | AUTOMATION | Flame sensor: IGNITION DETECTED                  |

At the bottom of the table, there is a pagination bar showing 'Total records: 100' and a page selector with options 1, 2, 3, 4, 5, 6, 7, and a '15 / page' button.

Рисунок 2.38— Інтерфейс табличного журналу подій та системних аварій

Формування часових міток відбувається безпосередньо на стороні сервера Node-RED у моменти перезапису Modbus-регістрів або спрацювання програмних тригерів. У рамках даної дипломної роботи повідомлення акумулюються у циклічному буфері оперативної пам'яті сервера в межах поточного сеансу, чого цілком достатньо для демонстрації роботи логіки.

#### 2.9.4 Апробація роботи системи та моделювання нештатних ситуацій

Для підтвердження працездатності розроблених алгоритмів та оцінки надійності НМІ було проведено серію імітаційних випробувань. Тестування охоплювало як штатні експлуатаційні режими, так і критичні нештатні ситуації, що вимагають миттєвої реакції автоматики.

*Сценарій 1, перехід у режим ручного керування.* В умовах нормального електропостачання диспетчер має змогу перевести систему з автоматичного в ручний режим роботи за допомогою відповідного селектора на глобальній панелі НМІ. При активації цього режиму на головному дашборді розблоковуються елементи прямого впливу на виконавчі механізми, зокрема кнопки примусового запуску газового пальника та циркуляційного насоса, рисунок 2.39. У цьому стані адаптивний алгоритм керування призупиняє свою автономну дію, передаючи повний контроль над технологічним процесом оператору. Такий функціонал є критично необхідним для проведення пусконаладжувальних робіт, перевірки працездатності окремих агрегатів або виконання регламентного сервісного обслуговування.

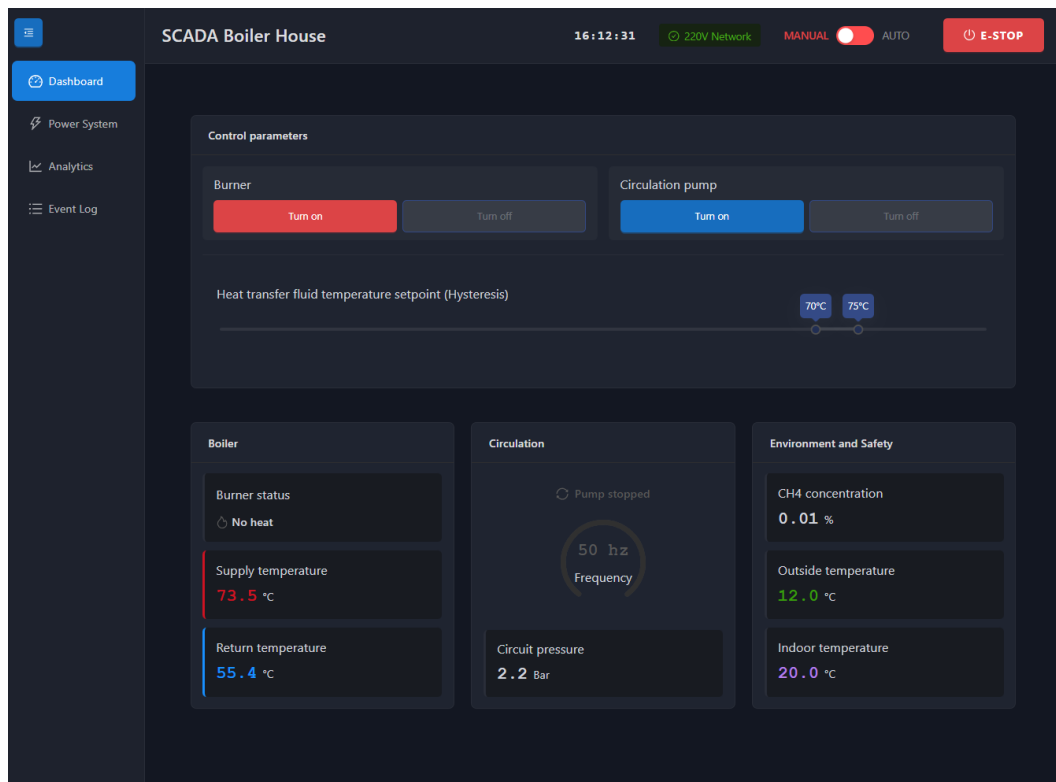


Рисунок 2.39— Інтерфейс дашборда під час активного режиму ручного керування

*Сценарій 2, апаратне блокування при аварії в ручному режимі.*

Фундаментальною вимогою до сучасних систем автоматизації є безумовний пріоритет контурів безпеки над діями експлуатаційного персоналу. Для перевірки надійності цього механізму було змодельовано ситуацію, коли система знаходиться під ручним управлінням оператора, але в контурі виникає критичне відхилення. Результати тестування підтвердили, що незалежно від активного режиму та наявності ручних команд на запуск обладнання, серверне ядро негайно фіксує перетин аварійної межі та примусово знімає керуючі сигнали з пальника і насоса. У цей же момент на інтерфейсі диспетчера спрацьовує візуальна сигналізація тривоги, рисунок 2.40, а в системний журнал автоматично вноситься відповідний запис із точною вказівкою першопричини екстреного блокування, рисунок 2.41.

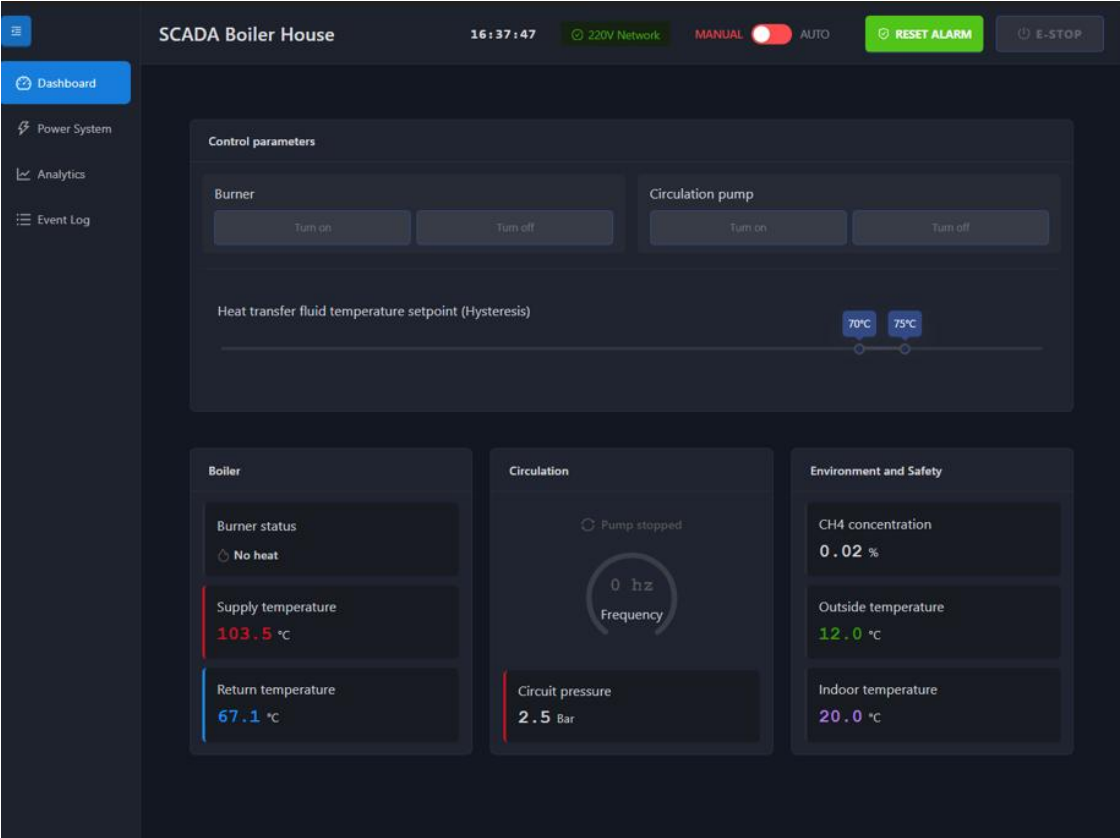


Рисунок 2.40— Візуалізація аварійної зупинки обладнання

| Date and time          | Category | Initiator  | Message (Event description)                             |
|------------------------|----------|------------|---------------------------------------------------------|
| 16:37:32<br>16.06.2026 | INFO     | AUTOMATION | Flame sensor: NO FLAME                                  |
| 16:37:31<br>16.06.2026 | COMMAND  | AUTOMATION | Equipment control: Circulation Pump turned OFF          |
| 16:37:31<br>16.06.2026 | COMMAND  | AUTOMATION | Equipment control: Main Boiler Heater turned OFF        |
| 16:37:31<br>16.06.2026 | CRITICAL | АВТОМАТИКА | EMERGENCY PROTECTION: Critically HIGH pressure: 2.5 Bar |
| 16:36:51<br>16.06.2026 | INFO     | AUTOMATION | Flame sensor: IGNITION DETECTED                         |
| 16:36:48<br>16.06.2026 | COMMAND  | OPERATOR   | Equipment control: Main Boiler Heater turned ON         |
| 16:36:30<br>16.06.2026 | COMMAND  | OPERATOR   | Operation mode changed: MANUAL                          |
| 16:36:19<br>16.06.2026 | INFO     | AUTOMATION | Flame sensor: NO FLAME                                  |
| 16:36:18<br>16.06.2026 | COMMAND  | AUTOMATION | Equipment control: Main Boiler Heater turned OFF        |

Рисунок 2.41— Фіксація аварійної події у журналі

Сценарій 3. Реакція системи на втрату електропостачання та адаптивне енергозбереження. Ключовим етапом апробації стала імітація повного знеструмлення об'єкта з метою перевірки адекватності реакції адаптивного алгоритму виживання. У момент втрати напруги в основній мережі серверне ядро миттєво фіксує зміну статусу живлення, рисунок 2.42.

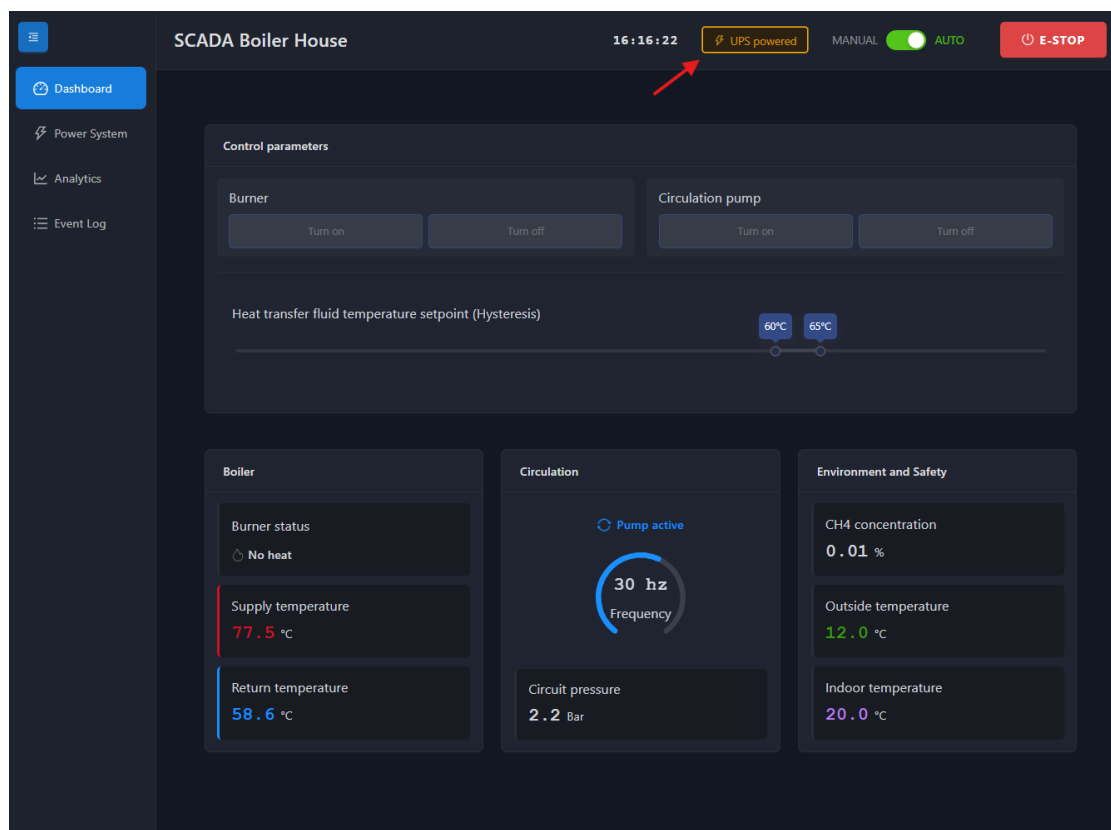


Рисунок 2.42— Відображення переходу на резервне живлення

Одразу після фіксації аварії живлення алгоритм активує протокол енергозбереження, ключовою дією якого є зниження частоти обертання циркуляційного насоса для мінімізації навантаження на інвертор. Наслідки цього рішення оператор має змогу детально проаналізувати на спеціалізованій вкладці моніторингу енергосистеми, рисунок 2.43. На екрані чітко прослідковується, як завдяки кубічній залежності споживаної потужності від швидкості насоса, зниження його продуктивності призводить до стрімкого збільшення розрахункового часу автономної роботи котельні порівняно зі штатним режимом.

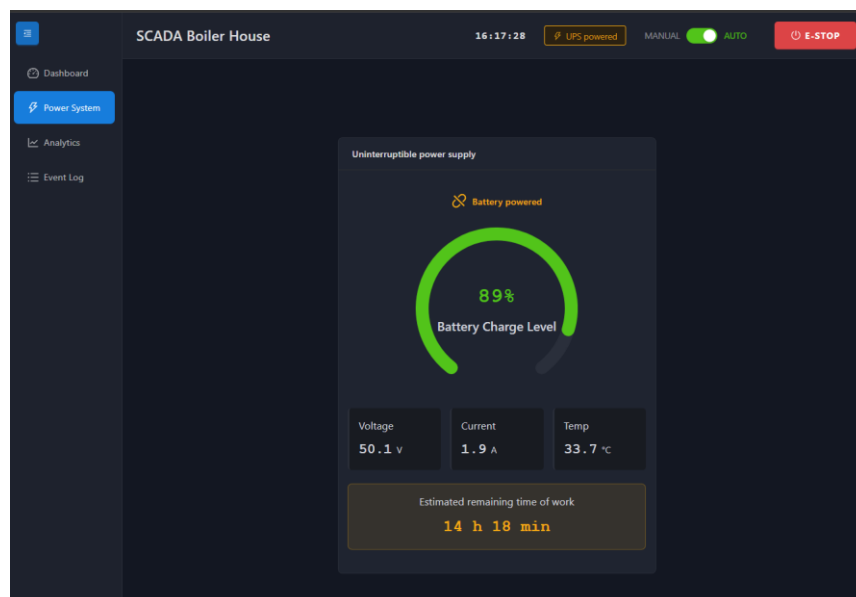


Рисунок 2.43— Розрахунок нового часу автономної роботи

Додатковим інструментом для перевірки ефективності розробленої логіки виступає підсистема тренд-аналітики, рисунок 2.44. На графіку розряду акумуляторних батарей наочно демонструється зміна кута нахилу кривої деградації заряду: після моменту знеструмлення та ініціалізації режиму економії графік стає значно пологішим. Це є беззаперечним графічним підтвердженням того, що втручання адаптивного алгоритму суттєво розтягує наявний запас енергії у часі, запобігаючи швидкому виснаженню ДБЖ.



Рисунок 2.44— Аналіз ефективності алгоритму збереження енергії

У модулі журналу подій детально фіксуються послідовні записи про факт зникнення напруги в мережі, перехід системи керування у відповідний стан резервного функціонування, а також точні системні повідомлення про алгоритмічне зниження потужності насоса із зазначенням нових значень у відсотках, рисунок 2.45.

| Date and time          | Category | Initiator  | Message (Event description)                                |
|------------------------|----------|------------|------------------------------------------------------------|
| 16:17:11<br>16.06.2026 | INFO     | AUTOMATION | Flame sensor: NO FLAME                                     |
| 16:17:11<br>16.06.2026 | COMMAND  | AUTOMATION | Parameter adjusted: Circulation Pump Speed set to 30 Hz    |
| 16:17:10<br>16.06.2026 | COMMAND  | AUTOMATION | Equipment control: Main Boiler Heater turned OFF           |
| 16:17:00<br>16.06.2026 | INFO     | AUTOMATION | Flame sensor: IGNITION DETECTED                            |
| 16:16:58<br>16.06.2026 | COMMAND  | AUTOMATION | Parameter adjusted: Circulation Pump Speed set to 50 Hz    |
| 16:16:57<br>16.06.2026 | COMMAND  | AUTOMATION | Equipment control: Main Boiler Heater turned ON            |
| 16:16:21<br>16.06.2026 | COMMAND  | AUTOMATION | Parameter adjusted: Circulation Pump Speed set to 30 Hz    |
| 16:16:21<br>16.06.2026 | WARNING  | AUTOMATION | Adaptive temperature setpoints updated: Min 60°C, Max 65°C |
| 16:16:21<br>16.06.2026 | WARNING  | AUTOMATION | 220V grid power lost. Switching to battery power.          |

Рисунок 2.45— Реєстрація послідовності активації алгоритмів виживання у системному журналі

Окремим критичним етапом у межах даного сценарію є досягнення акумуляторною батареєю порогу глибокого розряду, що встановлений на позначці 10% і нижче від номінальної ємності. При перетині цієї межі адаптивний алгоритм виживання ініціює фінальну стадію протоколу захисту — режим безпечного відключення об'єкта. У цьому випадку алгоритм примусово скидає прапорець автоматичного режиму, переводячи систему в ручний стан, та повністю деактивує газовий пальник і циркуляційний насос, рисунок 2.46. Таке рішення запобігає аварійному та непередбачуваному вимкненню обладнання в момент повної втрати живлення, захищаючи ПЛК та аналогові датчики від апаратних збоїв. Одночасно з цим у системному журналі подій з'являється запис з попередженням про вичерпання резерву енергії. Наявність

цього кейсу в імітаційній моделі підтверджує, що розроблений комплекс диспетчеризації здатний не лише оптимізувати споживання ресурсів, а й гарантувати безаварійну зупинку теплогенеруючого об'єкта за будь-яких умов автономної роботи.

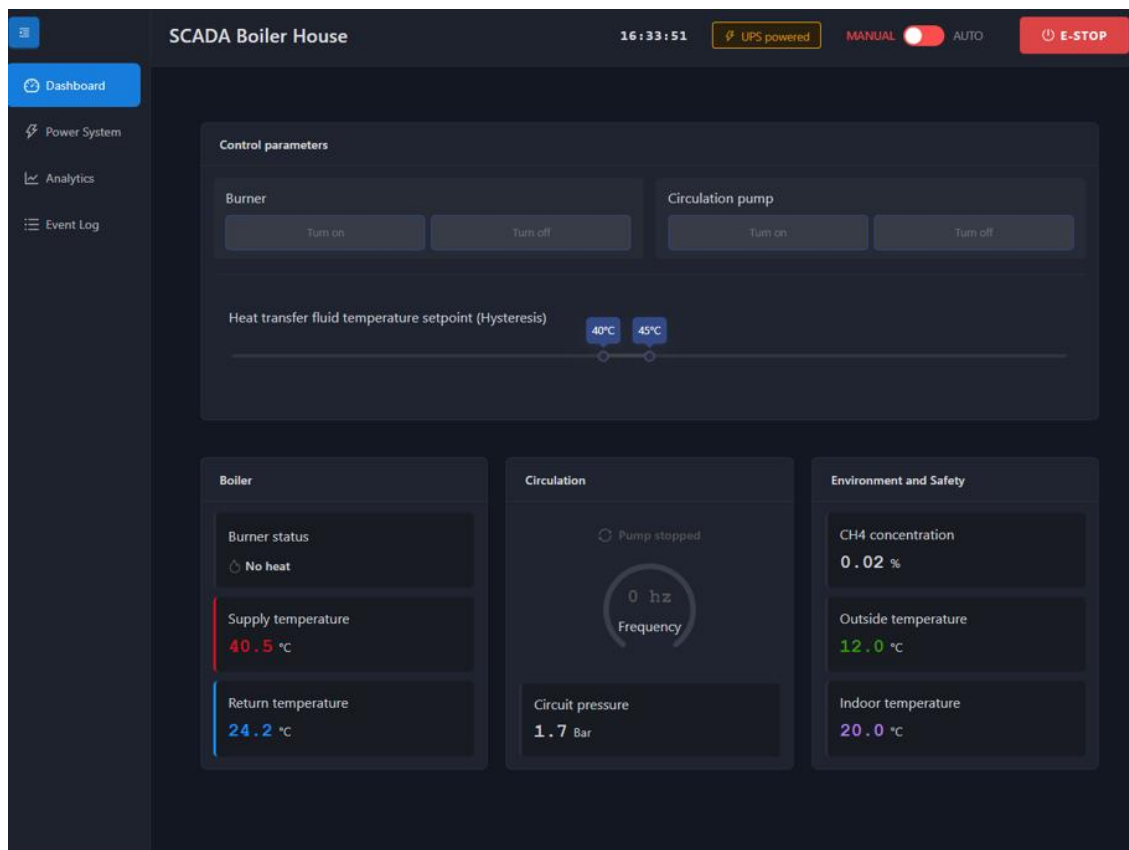


Рисунок 2.46— Інтерфейс дашборда при критичному розряді АКБ

## 2.10 Адаптивність кросплатформного користувацького інтерфейсу

Однією з ключових вимог до сучасних систем класу Web SCADA є забезпечення безперервного та повноцінного доступу диспетчерського персоналу до керування об'єктом незалежно від типу використовуваного апаратного забезпечення. Для реалізації цієї вимоги розроблений клієнтський додаток побудовано за принципами адаптивного веб-дизайну *Responsive Web Design*.

При відкритті диспетчерського додатка на екрані планшету візуальний макет автоматично оптимізує корисний простір. Масштабування відбувається

таким чином, щоб зберегти комфортне відображення глобальної панелі керування та карток з ключовими показниками телеметрії. Розміри інтерактивних елементів, зокрема кнопок та візуальних індикаторів, адаптуються для зручної взаємодії за допомогою сенсорного екрана без необхідності горизонтального прокручування. Інтерфейс додатку для планшета зображено на рисунку 2.47.

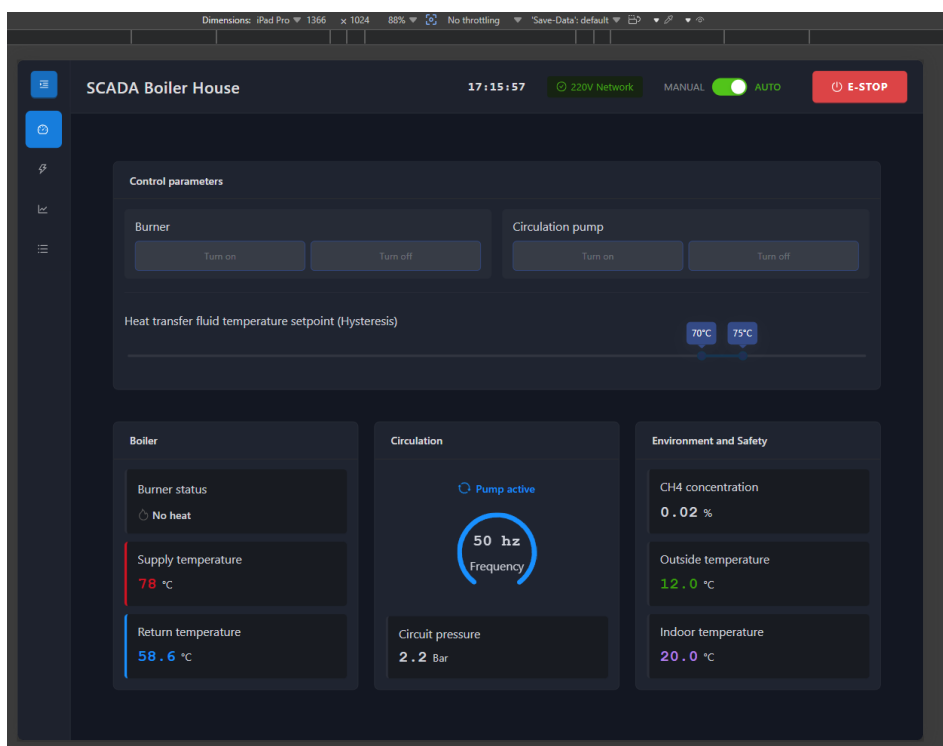


Рисунок 2.47— Адаптація інтерфейсу для планшета

У випадку доступу до системи з мобільного смартфона, що є актуальним під час фізичної відсутності оператора на робочому місці, інтерфейс зазнає структурної трансформації. Багатостовпковий макет робочої області автоматично перебудовується в єдину вертикальну стрічку, рисунок 2.48. Інформаційні блоки, елементи керування та графіки розташовуються послідовно один за одним, що забезпечує зручний доступ до всіх функцій системи за допомогою стандартного вертикального скролінгу.

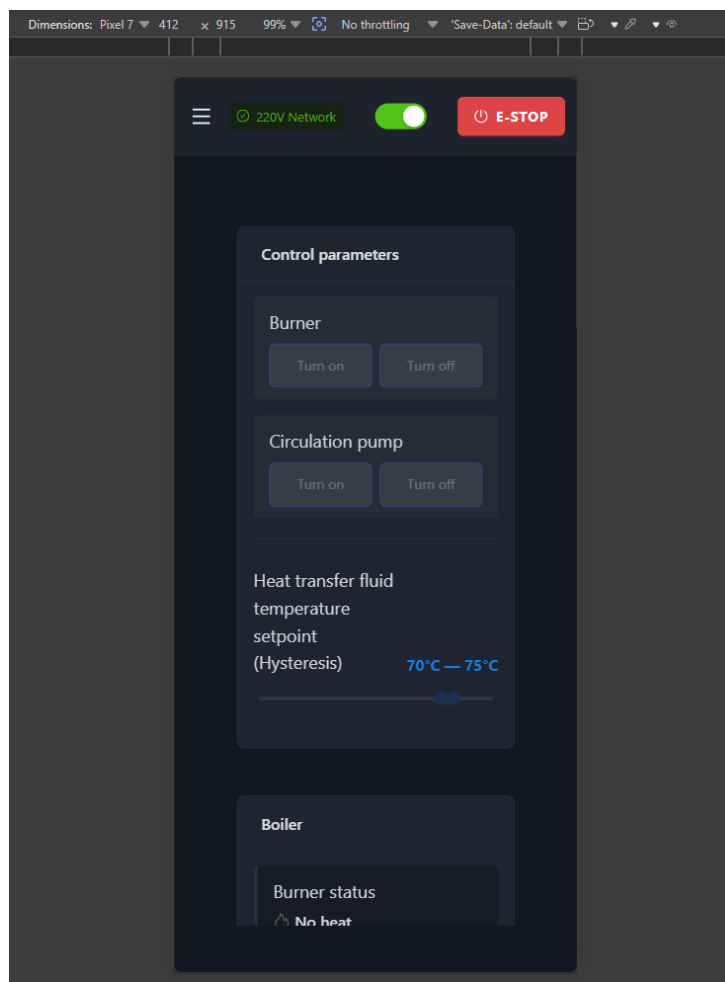


Рисунок 2.48— Адаптація інтерфейсу для мобільних пристроїв

Такий базовий підхід до проєктування фронтенд-частини гарантує, що у разі виникнення нештатної ситуації або блекауту, інженер матиме змогу оперативно оцінити стан системи, проаналізувати поточний заряд акумуляторних батарей та відправити керуючі команди безпосередньо зі свого смартфона, використовуючи інтуїтивно зрозуміле розташування елементів

## 2.11 Шляхи подальшого вдосконалення системи

Хоча розроблений програмно-апаратний комплекс повністю виконує поставлені завдання в рамках поточного проєкту, для його повноцінного промислового впровадження доцільно реалізувати ряд архітектурних покращень. По-перше, перехід від імітаційної моделі цифрового двійника до

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.02.ПЗ | Арк. |
|      |      |          |        |      |                         | 70   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

керування реальним технологічним об'єктом потребуватиме лише зміни мережевих налаштувань підключення до фізичного ПЛК за стандартом Modbus TCP та відключення віртуальних симуляторів фізики. По-друге, для забезпечення довгострокового та надійного зберігання великих масивів журналу подій і трендів необхідно розширити архітектуру інтеграцією зовнішньої реляційної бази даних, наприклад PostgreSQL. По-третє, клієнтський React-додаток для підвищення загальної продуктивності, безпеки та стійкості до високих навантажень має бути скомпільований у статичні файли та розгорнутий на виділеному веб-сервері промислового рівня.

#### *Висновки за розділом:*

У другому розділі здійснено комплексне проектування та програмну імплементацію інтелектуальної системи керування газовою котельнею, що адаптована до умов тривалих блекаутів. На основі паспортних даних обраного технологічного обладнання — водогрійного котла КОЛВІ 250 (номінальна потужність 250 кВт, водяний об'єм 350 л) та циркуляційного насоса Wilo Stratos MAHO (максимальна електрична потужність 410 Вт) — було розроблено математичні моделі теплової та енергетичної динаміки об'єкта. Перехідний процес нагрівання теплоносія математично описано аперіодичною ланкою першого порядку із транспортним запізненням, де розрахунковий коефіцієнт підсилення становить  $K = 60$ , постійна часу теплової інерції  $T = 450$  с, а чисте запізнення  $\tau = 30$  с.

Фундаментом розробленого алгоритму енергозбереження стали закони афінної подібності, зокрема — кубічна залежність споживаної потужності відцентрового насоса від частоти обертання його ротора. Математичне та імітаційне моделювання в середовищі MATLAB/Simulink довело, що програмне обмеження продуктивності насоса до 60% (еквівалент 30 Гц) призводить до нелінійного падіння його енергоспоживання з 410 Вт до розрахункових 88,5 Вт.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.02.ПЗ | Арк. |
|      |      |          |        |      |                         | 71   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

Результати імітаційного моделювання динаміки розряду акумуляторних батарей підтвердили високу ефективність розробленої системи: у класичному релейному режимі обладнання повністю виснажує резерв ємності за 25 годин, тоді як імплементація адаптивного алгоритму подовжує час автономної роботи об'єкта до 54 годин. При досягненні критичного рівня заряду в 10% алгоритм автоматично ініціює протокол Safe Shutdown, відсікаючи силове навантаження та зберігаючи залишковий резерв енергії виключно для живлення мікропроцесорної автоматики контролера та засобів зв'язку (постійне споживання 150 Вт).

Практичну реалізацію логіки керування та створення цифрового двійника котельні виконано в середовищі Node-RED, яке забезпечило надійну взаємодію з ПЛК нижнього рівня за промисловим протоколом Modbus TCP. Цифровий двійник дозволив безпечно симулювати фізичні апаратні затримки розпалювання пальника, тепловіддачу приміщення за законом Ньютона-Ріхмана та аварійні процеси.

Клієнтський НМІ розроблено на базі бібліотеки React із застосуванням сучасної мікрофронтендної архітектури. Обмін телеметричними даними між Edge-шлюзом та веб-інтерфейсом забезпечується через дуплексний протокол WebSockets, що гарантує високу швидкість реакції інтерфейсу.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.02.ПЗ | Арк. |
|      |      |          |        |      |                         | 72   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

## ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне завдання щодо розробки системи автоматизованого керування газовою котельнею, здатної забезпечити стійке, енергоефективне та безпечне функціонування теплогенеруючого об'єкта в умовах тривалих відключень основної мережі електропостачання.

Обґрунтовано вибір комплексу технічних засобів автоматизації, що базується на використанні промислового програмованого логічного контролера Siemens серії S7-1200 як пристрою нижнього рівня. У якості об'єктів керування обрано реальне технологічне обладнання: газовий водогрійний котел КОЛВІ 250 та інтелектуальний циркуляційний насос Wilo Stratos MAXO.

На основі детального аналізу існуючих рішень та технологічних вимог було розроблено імітаційні математичні моделі термодинамічних процесів теплообміну, гідравліки та енергетичного балансу резервного джерела живлення. Теоретичним підґрунтям для створення адаптивних алгоритмів виживання стало застосування законів афінної подібності, що дозволило математично обґрунтувати концепцію динамічного енергозбереження. Було доведено можливість суттєвого подовження часу автономної роботи котельні за рахунок використання нелінійної кубічної залежності споживаної потужності циркуляційного насоса від частоти його обертання.

Проведене математичне та імітаційне моделювання у програмному комплексі MATLAB/Simulink повністю підтвердело теоретичні розрахунки. Моделювання динаміки розряду акумуляторних батарей довело, що примусове програмне зниження частоти керуючого впливу до 30 Гц призводить до нелінійного падіння електричного навантаження насоса з номінальних 410 Вт до 88,5 Вт. Це, у свою чергу, дозволяє збільшити загальний час автономної роботи критичної інфраструктури котельні з 25 до 54 годин порівняно з класичним релейним алгоритмом керування. Перехідні теплові процеси нагріву

|           |      |                  |        |      |                                |  |  |  |                    |      |         |   |  |
|-----------|------|------------------|--------|------|--------------------------------|--|--|--|--------------------|------|---------|---|--|
|           |      |                  |        |      | <i>КНУ КРБ.151.26.13.00.ПЗ</i> |  |  |  |                    |      |         |   |  |
|           |      |                  |        |      |                                |  |  |  |                    |      |         |   |  |
| Змн.      | Арк. | № докум.         | Підпис | Дата |                                |  |  |  |                    |      |         |   |  |
| Розроб.   |      | ШПеремета Д.А.   |        |      | <i>ВИСНОВКИ</i>                |  |  |  | Літ.               | Арк. | Аркушів |   |  |
| Перевір.  |      | Тиханський М. П. |        |      |                                |  |  |  |                    |      | 73      | 3 |  |
|           |      |                  |        |      |                                |  |  |  | <i>КНУ АКИТ-22</i> |      |         |   |  |
| Н. Контр. |      | Маринич І.А.     |        |      |                                |  |  |  |                    |      |         |   |  |
| Затверд.  |      | Маринич І.А.     |        |      |                                |  |  |  |                    |      |         |   |  |

успішно апроксимовано передавальною функцією аперіодичної ланки першого порядку з транспортним запізненням.

Програмна реалізація логічного ядра системи була виконана у середовищі візуального подієво-орієнтованого програмування Node-RED, яке ефективно виконало функції промислового IoT-шлюзу та комунікаційного сервера. У рамках розробки було створено програмний модуль цифрового двійника котельні, що дозволило провести безпечне та комплексне імітаційне моделювання фізичних і аварійних процесів без ризику пошкодження реального обладнання. Запропонована архітектура цифрового двійника здатна в реальному часі відтворювати апаратні затримки розпалювання пальника, розраховувати тепловтрати приміщення за законом охолодження Ньютона-Ріхмана, а також імітувати аварійні стрибки тиску при досягненні теплоносієм температури кипіння. Інтеграція серверного ядра з віртуальним програмованим логічним контролером була успішно реалізована за стандартом промислового зв'язку Modbus TCP, що гарантує високу масштабованість проєкту та його готовність до швидкого перенесення на реальні фізичні об'єкти після мінімальної реконфігурації мережевих параметрів.

Значним інноваційним досягненням роботи є розробка сучасного клієнтського людино-машинного інтерфейсу на базі екосистеми React із застосуванням мікрофронтендної архітектури. Відмова від застарілих монолітних SCADA-систем на користь передових веб-технологій та організація двостороннього асинхронного обміну даними з використанням WebSockets дозволили створити відмовостійкий, швидкодіючий та інтуїтивно зрозумілий диспетчерський додаток. Створений інтерфейс забезпечує оперативний моніторинг ключових технологічних показників, інтелектуальне прогнозування часу роботи об'єкта від акумуляторних батарей, глибоку тренд-аналітику параметрів та автоматизоване ведення журналу подій.

Проведені імітаційні випробування розробленого програмно-апаратного комплексу в умовах штатних режимів та змодельованих критичних нештатних ситуацій підтвердили його абсолютну працездатність та високу надійність.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.00.ПЗ | Арк. |
|      |      |          |        |      |                         | 74   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

Система успішно продемонструвала здатність автономно переходити в режим адаптивного енергозбереження під час знеструмлення, гарантуючи при цьому беззаперечний пріоритет контурів протиаварійного захисту над діями оператора та забезпечуючи безпечну зупинку обладнання при критичному виснаженні ДБЖ. Таким чином, мету дипломного проєкту досягнуто в повному обсязі, а запропоновані рішення становлять високу практичну цінність для впровадження у сфері комунальної теплоенергетики.

|      |      |          |        |      |                         |      |
|------|------|----------|--------|------|-------------------------|------|
|      |      |          |        |      | КНУ КРБ.151.26.13.00.ПЗ | Арк. |
|      |      |          |        |      |                         | 75   |
| Змн. | Арк. | № докум. | Підпис | Дата |                         |      |

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ельперін І. В., Пупена О. М., Сідлецький В. М., Швед М. М. Автоматизація виробничих процесів: підручник. Київ: Видавництво «Ліра-К», 2015. 378 с.
2. Левченко О. І., Сідлецький В. М. Основи автоматизації теплоенергетичних процесів та установок: навчальний посібник. Київ: НУХТ, 2014. 227 с.
3. Boyes H. et al. The industrial internet of things (IIoT): An analysis framework // Computers in Industry. 2018. Vol. 101. P. 1-12.
4. Fette I., Melnikov A. The WebSocket Protocol. Internet Engineering Task Force (IETF), RFC 6455, 2011. 71 p.
5. Grieves M. Digital Twin: Manufacturing Excellence through Virtual Factory Replication. White Paper, 2014.
6. Modbus Application Protocol Specification V1.1b3. Modbus Organization, 2012. 50 p. : URL: (Дата звернення 10.03.2026).
7. Node-RED Documentation. Node-RED : URL: <https://nodered.org/docs/> (Дата звернення 10.03.2026).
8. React Official Documentation. React : URL: <https://react.dev/learn> (Дата звернення 13.04.2026).
9. Vite: Next Generation Frontend Tooling. ViteJS : URL: <https://vitejs.dev/guide/> (Дата звернення 15.04.2026)
10. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. Київ: Либідь, 2007. 656 с.
11. SIMATIC S7 S7-1200 Programmable controller. System Manual. Siemens AG, 2022. 1826 p. URL: <https://support.industry.siemens.com/> (Дата звернення: 20.04.2026).

|           |      |                 |        |      |                         |  |  |      |      |         |   |
|-----------|------|-----------------|--------|------|-------------------------|--|--|------|------|---------|---|
|           |      |                 |        |      | КНУ КРБ.151.26.13.00.ПЗ |  |  |      |      |         |   |
|           |      |                 |        |      |                         |  |  |      |      |         |   |
| Змн.      | Арк. | № докум.        | Підпис | Дата |                         |  |  |      |      |         |   |
| Розроб.   |      | Шеремета Д.А.   |        |      | СПИСОК ЛІТЕРАТУРИ       |  |  | Літ. | Арк. | Аркушів |   |
| Перевір.  |      | Тиханський М. . |        |      |                         |  |  |      |      | 76      | 3 |
|           |      |                 |        |      |                         |  |  |      |      |         |   |
| Н. Контр. |      | Маринич І.А.    |        |      | КНУ АКІТ-22             |  |  |      |      |         |   |
| Затвердив |      | Маринич І.А.    |        |      |                         |  |  |      |      |         |   |

12. Котел газовий водогрійний КОЛВІ 250. Паспорт та керівництво з експлуатації. ТОВ «Колві», 2020. 45 с.

13. Wilo-Stratos MAXO. Інструкція з монтажу та експлуатації. Wilo SE, 2021. 110 с.

14. Система контролю загазованості ВАРТА 1-03. Настанова щодо експлуатації. ТОВ «Новатор», 2021. 24 с

15. ДСТУ EN 60751:2010. Термоперетворювачі опору промислові платинові та чутливі елементи платинові. Київ: Держспоживстандарт України, 2010. 26 с.

16. НПАОП 0.00-1.76-15. Правила безпеки систем газопостачання. Київ: Міненерговугілля України, 2015. 112 с.

17. Жученко А. І., Осипенко В. А. Автоматизація технологічних процесів: Підручник. Київ: КПІ ім. Ігоря Сікорського, 2019. 412 с.

18. Сторчак К.П., Тушич А.М., Срібна І.М., Яковенко Н.Д., Кравець Д.В. Технології Інтернет речей. Навч. посібник підготовлено для студентів вищих навчальних закладів – Київ: ДУТ, 2021. – 68 с.

19. Node.js Official Documentation. Node.js Foundation. URL: <https://nodejs.org/docs/> (Дата звернення: 18.04.2026).

20. MDN Web Docs: WebSockets API. Mozilla Foundation. URL: [https://developer.mozilla.org/en-US/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API) (Дата звернення: 19.04.2026).

21. Пупена О. М. Довідник з Node-RED URL: <https://pupenasan.github.io/NodeREDGuidUKR/> (Дата звернення: 25.04.2026).

22. Пупена О. М. Автоматизація типових технологічних процесів URL: <https://asu-in-ua.github.io/atpv/> (Дата звернення: 25.04.2026).

23. Пупена О. М. Автоматизація виробничих процесів URL: <https://pupenasan.github.io/avpbook/> (Дата звернення: 25.04.2026).

24. Пупена О. М. Людино-машинні інтерфейси URL: <https://pupenasan.github.io/hmibook/> (Дата звернення: 25.04.2026).

25. ДСТУ EN 61131-3:2014 Контролери програмовані. Частина 3. Мови програмування (EN 61131-3:2013, IDT). Київ: Держспоживстандарт України, 2015. 248 с.

26. Мартин Р. Чиста архітектура. Мистецтво розроблення програмного забезпечення. Харків: Фабула, 2019. 368 с.

27. Kurose J. F., Ross K. W. Computer Networking: A Top-Down Approach (8th Edition). Pearson, 2020. 832 с.

28. Дакетт Д. HTML і CSS. Розробка та дизайн веб-сайтів. Харків: Клуб сімейного дозвілля, 2016. 480 с.

29. Моркун Н. В., Маринич І. А. Методичні вказівки до виконання кваліфікаційної роботи бакалавру для студентів спеціальності 151 “Автоматизація та комп’ютерно-інтегровані технології”. Кривий Ріг : Видавничий центр КНУ, 2019. 50 с.

30. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. Київ, ДП «УкрННЦ», 2015. 26с. (Інформація та документація).

31. ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання Київ, ДП «УкрННЦ», 2016. 16 с. (Інформація та документація).

32. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила. Київ, ДП «УкрННЦ», 2013. 23 с. (Інформація та документація)