

ХЕРСОНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ІНЖЕНЕРІЇ ТА ТРАНСПОРТУ

КАФЕДРА АВТОМАТИЗАЦІЇ, РОБОТОТЕХНІКИ І МЕХАТРОНИКИ

Пояснювальна записка

до кваліфікаційної роботи магістра

на тему: «Автоматизована система поквартирного розподілу і обліку теплової енергії»

«Automated metering system of apartment heat distribution and consumption»

Виконав: студент 6 курсу, групи БА
спеціальність 174 – «Автоматизація,
комп'ютерно-інтегровані технології та
робототехніка»

Вяткін О.В.

Керівник: к.т.н., доцент Поливода О.В.

Рецензент к.т.н., доцент Хохлов В.А.

(прізвище та ініціали)

Херсон – 2025 рік

Херсонський національний технічний університет
Факультет Інженерії та транспорту
Кафедра Автоматизації, робототехніки і мехатроніки
Ступінь вищої освіти магістр
Спеціальність 174 – «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»

ЗАТВЕРДЖУЮ:
Завідувач кафедри автоматизації,
робототехніки і мехатроніки
Селіверстов І.А.
«__» _____ 2025 р.

ЗАВДАННЯ

на дипломну роботу студенту

Вяткіну Олексію Владиславовичу

1. Тема проекту: Автоматизована система поквартирного розподілу і обліку теплової енергії
Automated metering system of apartment heat distribution and consumption
керівник проекту: к.т.н., доцент Поливода О.В.
затверджена наказом вищого навчального закладу від 28.08.2025 р. № 364-с
2. Строк подання студентом проекту «15» грудня 2025 р.
3. Вихідні дані до проекту: довідникова література
4. Зміст розрахунково-пояснювальної записки:
 1. Теоретичні основи автоматизованих систем обліку теплової енергії.
 2. Постановка задачі та розробка концепції структури автоматизованої системи.
 3. Технічне забезпечення системи.
 4. Програмне забезпечення автоматизованої системи
5. Перелік графічного матеріалу:
 1. Огляд існуючих приладів обліку.
 2. Система поквартирного обліку теплової енергії.
 3. Алгоритми розрахунку спожитої теплової енергії.
 4. Терморегулятори та значення у системі.
 5. Технологічна схема системи обліку розподілу і обліку теплової енергії.
 6. Веб-застосунок.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	Поливода О.В., к.т.н., доцент		

7. Дата видачі завдання

«02» вересня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Підбір і огляд літератури	03.09-24.09	
2	Аналіз існуючих методів обліку тепла	25.09-03.10	
3	Пошук обладнання	05.10-13.10	
4	Аналіз і порівняння	14.10-21.10	
5	Формування ідеї для програми з обліку тепла	22.10-29.10	
6	Створення веб-застосунку	30.10-25.11	
7	Оформлення ПЗ і графічного матеріалу	26.11-12.12	

Студент

Вяткін О.В.

_____ (підпис)

Керівник проекту

Поливода О.В.

_____ (підпис)

РЕФЕРАТ

Кваліфікаційна робота магістра: 67 сторінок, 13 рисунків. Графічна частина – 6 аркуша формату А1.

Кваліфікаційна робота магістра присвячена аналізу існуючих систем поквартирного обліку теплопостачання та створенню диспетчерської системи для моніторингу її.

Були проведені дослідницькі роботи по аналізу сучасних тенденцій в автоматизації по енергозбереженні, методам обліку та існуюче обладнання. Була проведена постановка відповідної задачі для створення системи обліку тепла, формування вимог, підбір архітектури. Проведенно порівняння різних лічильників, мікроконтролерів та мережевих технологій

Був розроблений веб-застосунок для моніторингу тепла в квартирах і регулювання потоку тепла в квартири.

SCADA, ТЕПЛОЛІЧИЛЬНИК, МІКРОКОНТРОЛЕР, АПРОКСИМАЦІЯ,
ВЕБ-ЗАСТОСУНОК, API

					ХНТУ 174.КРМ.25.003 РФ			
Зм.	Арк.	№ докум	Підпис	Дата				
Розроб.	Вяткін О.В				Реферат	Літ.	Аркуш	Аркушів
Перевір.	Полівода О.В.						5	67
Реценз.								
Н. Контр.	Поліщук В.М.							
Затверд.	Селівєрстов І.А.							
						ХНТУ, гр. 2Амб		

ABSTRACT

Master's qualification work: 67 pages, 13 figures. Graphic part - 6 sheets of A1 format.

The master's qualification work is devoted to the analysis of existing systems of apartment heat metering and the creation of a dispatching system for monitoring it. Research work was carried out on the analysis of modern trends in automation for energy saving, metering methods and existing equipment. A corresponding task was formulated for the creation of a heat metering system, requirements were formed, and architecture was selected. A comparison of various meters, microcontrollers and network technologies was carried out.

A web application was developed for monitoring heat in apartments and regulating the flow of heat into apartments.

SCADA, HEAT METER, MICROCONTROLLER, APPROXIMATION, WEB APPLICATION, API

					ХНТУ 174.КРМ.25.003 РФ	Арк.
						6
Зм.	Арк.	№ докум	Підпис	Дата		

ЗМІСТ

	Стор.
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	9
ВСТУП.....	10
1 ТЕОРЕТИЧНІ ОСНОВИ АВТОМАТИЗОВАНИХ СИСТЕМ ОБЛІКУ ТЕПЛОВОЇ ЕНЕРГІЇ.....	12
1.1 Аналіз сучасного стану систем теплозабезпечення в багатоквартирних будинках	12
1.2 Методи та принципи обліку теплової енергії	15
1.3 Існуючі прилади обліку та їх характеристика.....	17
1.4 Сучасні тенденції в автоматизації та енергозбереженні	21
2. ПОСТАНОВКА ЗАДАЧІ ТА РОЗРОБКА КОНЦЕПЦІЇ СТРУКТУРИ АВТОМАТИЗОВАНОЇ СИСТЕМИ.....	24
2.1 Формулювання вимог до системи	24
2.2 Вибір загальної архітектури системи	28
2.3 Структурна схема системи поквартирного обліку теплової енергії.....	31
2.4 Аналіз можливих варіантів обліку теплопостачання	33
3. ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	35
3.1 Огляд та вибір теплотічильників і датчиків	35
3.2 Використання мікроконтролерів та вузлів збору даних.....	38
3.3 Мережеві технології для передачі даних	40
4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ	42
4.1 Архітектура програмного забезпечення.....	42
4.2 Алгоритми розрахунку спожитої теплової енергії	44
4.3 Розробка користувацького інтерфейсу.....	47
4.4 Веб-інтерфейс	49
4.5 Серверна частина веб-додатку.....	54
4.6 SCADA-система для диспетчерського контролю.....	57
4.7 База даних та обмін через API	59

Зм.	Арк.	№ докум	Підпис	Дата

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	64
Додаток А. Код веб-інтерфейсу.....	68
Додаток Б. Код відкриття та закриття клапану	79

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>

ХНТУ 174.КРМ.25.003 ПЗ

Арк.

8

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

АСКТП-Автоматизована Система Контролю Технологічним Процесом

HTML- HyperText Markup Language

CSS- Cascading Style Sheets

SCADA- Supervisory Control and Data Acquisition

REST- Representational State Transfer

СУБД-Система Управління Базой Даних

SSE- Server-Sent Events

MQTT Message Queue Telemetry Transport-

HTTP- HyperText Transfer Protocol

TCP/IP- Transmission Control Protocol/Internet Protocol

AMS- Automated Metering System

AMI- Advanced Metering Infrastructure

Зм.	Арк.	№ докум	Підпис	Дата

ХНТУ 174.КРМ.25.003 ПЗ

Арк.

9

ВСТУП

Сучасний розвиток технологій та суспільства характеризується зростанням потреби в енергетичних ресурсах і особливо їх раціонального використання. В умовах енергетичної кризи, підвищення цін на енергетичні носії та зростання уваги до проблем енергоефективності питання обліку і оптимального використання теплової енергії набуває дуже сильного та особливої актуальності. Традиційна система централізованого тепlopостачання, яка присутня в майже більшості багатоквартирних будинків України, має доволі багато недоліків. Зокрема, відсутність індивідуального обліку споживання енергоносіїв та тепла призводить до нерівномірного розподілу витрат між мешканцями, неможливість контролю їх витрат, як наслідок, зниження мотивації до її економії

Впровадження індивідуальних приладів обліку теплової енергії є важливим кроком на шляху до створення фундаменту прозорих та справедливого нарахування платежів за теплоносії. Хоча, наявність окремих лічильників в приміщенні самі по собі не вирішують більшу частину проблем, оскільки виникає потреба у зборі, передачі та у подальшому обробці та аналізі даних. Тому на сцену виходять автоматизовані системи поквартирного розподілу і обліку теплової енергії, які поєднують у собі сучасні вимірювальні прилади, мікроконтролери та обов'язково засоби передачі інформації та додаткове програмне забезпечення яке необхідне для цієї системи. Такі системи дозволяють нам здійснювати облік теплоенергії в реальному часі, формувати аналітичні звіти, а також інтегруватись з інформаційними системами тепlopостачання різних підприємств якщо на це буде потреба.

Крім суто технічних переваг, такі системи мають і дуже велику соціально-економічну перевагу для самої людини та постачальником. По-перше, вони забезпечують прозорість у взаємовідносинах між споживачем та постачальником теплоенергії. По-друге, надають можливість мешканцям

					ХНТУ 174.КРМ.25.003 ПЗ	Арк.
						10
Зм.	Арк.	№ докум	Підпис	Дата		

можливість аналізувати витрати на тепло і якщо на це буде потреба, регулювати споживання, а саме збільшувати або зменшувати використання теплоенергії що сприяє для економії коштів або для комфортного життя. По-третє, широке впровадження таких систем стимулює загальне зменшення споживання енергоресурсів і, як, наслідок зниження викидів у навколишнє середовище.

Таки чином, важливість обраної теми зумовлена потребою у підвищенні ефективності використання а і у подальшому можливості економії теплової енергії, впроваджені інноваційних технологій в житлово-комунальну сферу.

Об'єктом цього дослідження є процес розподілу та споживання теплової енергії у багатоквартирних житлових будинках.

Предметом дослідження виступають методи, технічні засоби та програмне забезпечення для автоматизованого поквартирного обліку теплової енергії.

Метою цього дипломного проекту є створення веб-застосунку та дослідження автоматизованої системи, яка забезпечує поквартирний облік теплової енергії, можливість моніторингу споживання.

					ХНТУ 174.КРМ.25.003 ПЗ	Арк.
						11
Зм.	Арк.	№ докум	Підпис	Дата		

ті, що розатовашні всередині будівлі, а мешканці які самостійно утеплюють житло, змушені сплачувати стільки ж, скільки й сусіди без утеплення.

Сучасна тенденція у сфері енергозбереження полягає у переході до індивідуального поквартирного обліку теплової енергії. У багатьох країнах ЄС така практика є обов'язковою та регламентується відповідно законодавчою ініціативою. Вона дозволяє не лише справедливо розподіляти витрати, але й стимулює до більш економного використання теплової енергії. За статистикою після встановлення індивідуальних теплотічників споживання теплової енергії може зменшуватись на 15-30% завдяки більш раціональному використанню ресурсів. [2]

В Україні процес модернізації та переходу на індивідуальний облік перебуває на етапі активного впровадження, але існують значні організаційні та технічні бар'єри. До основних проблем можна віднести високу вартість обладнання, модернізацію вже значнозастарілих систем опалення, обмеженообізнаністю мешканців про переваги таких систем та відсутність стандартів інтеграції приладів обліку з інформаційними системами постачальників тепла.

У цьому контексті на перший план виходить створення автоматизованих систем поквартирного розподілу і обліку теплової енергії, які дозволяють комплексно вирішити вище зазначені проблеми. Такі системи дозволяють не тільки вимірювати а й автоматизованно передавати данні до єдиної диспетчерського центру. Це мінімізує людський фактор, спрощує процес зчитування показників і дозволяє формувати детальну аналітику як для мешканців, так і для теплопостачальних організацій.

Отже, сучасний стан теплозабезпечення в багатоквартирних будинках України характеризується поступовим переходом від колективного до індивідуального обліку тепла. Проте цей процес потребує комплексного підходу, що включає технічні рішення, нормативно-правове забезпечення та

інформаційні технології. Саме тому розробка автоматизованих систем поквартирного обліку є своєчасним і перспективним завданням, яке має важливе значення як для окремих споживачів, так і для держави в цілому.

					ХНТУ 174.КРМ.25.003 ПЗ	Арк.
						14
Зм.	Арк.	№ докум	Підпис	Дата		

1.2 Методи та принципи обліку теплової енергії

Облік теплової енергії на даний момент є одним із ключових елементів систем теплозабезпечення, адже саме на основі даних про фактичне споживання здійснюються розрахунки між споживачами та постачальниками послуг. Від правильності й точності вимірювань залежить не тільки економічна справедливість, але й ефективність управління всією системою тепlopостачання. [3]

Тому в загальній практиці сформулювалися різні методи та принципи обліку тепла, які відрізняються як за технічними рішеннями, так і за рівнем автоматизації.

Прямий метод обліку передбачає в собі встановлення ідивідуальних теплолічильників у квартирах чи будинках. Прилади вимірюють витрату теплоносія та температуру на вході й виході, після чого вбудований мікроконтролер враховує кількість спожитої теплової енергії. Цей метод є найбільш точним та прозорим, проте потребує значних капітальних витрат на обладнання та монтажні роботи.

Непрямий метод або розрахунковий використовується тоді, коли встановлення теплолічильника неможливе або економічно недоцільне. У такому випадку розподіл загального споживання будинку здійснюється пропорційно площі квартир, кількості мешканців або інших критеріїв. Недоліком цього методу є низька точність і несправедливість нарахувань, оскільки не враховує реальні умови споживання[4]

Метод розподілювачів тепла на радіаторах часто застосовується у Європі, прилади які розподіляють тепло безпосередньо встановлюється на опалювальні прилади. Вони фіксують відносну кількість тепла, яку віддав кожен радіатор, і дозволяють оцінити розподіл енергії між квартирами. Хоча він є не точним у порівнянні з прямим вимірюванням, але він є відповідно дешевшим і простішим у впровадженні.

Комбінований метод у собі поєднує використання будинкового теплолічильника та індивідуальних приладів або розподільвачів. Загальне споживання тепла визначається будинковим лічильником, далі ж розподіляється між квартирами з урахуванням показників індивідуальних приладів.

Останіми роками все більшого поширення набувають автоматизовані системи комерційного обліку теплової енергії. Завдяки автоматичним зчитуванням даних із теплолічильників, централізованим збереженням даних та обробка їх у подальшому це є набагато перспективнішою системою керування. Головна перевага автоматизованих систем полягає в мінімізації людського фактора, тобто відсутня потреба у щомісячному ручному знятті показників, знісється ймовірність помилок та сфабрикування показників, забезпечується можливість моніторингу споживання у реальному часі. [6]

Таким чином, методи обліку теплової енергії можна класифікувати за рівнем точності та автоматизації. Традиційні розрахункові методи зникають, на їх же місце приходять більш точні та автоматизовані методи обліку. Саме технології є непохитним фундаментом для створення ефективних рішень у сфері енергозбереження та прозорих розрахунків за тепло.

1.3 Існуючі прилади обліку та їх характеристика

У системах теплопостачання точність обліку теплової енергії безпосередньо залежить від типу та якості застосованих вимірювальних приладів. Сучасний ринок пропонує широкий вибір теплотічильників, сенсорів та модулів збору даних, які відрізняються за принципом дії, конструкцією, класам точності та здатністю інтегруватись в автоматизовані системи

Основним елементом систем обліку є сам теплотічильник-цей прилад, призначений для визначення кількості теплової енергії, що споживається опалювальною системою будівлі або окремої квартири. Самі теплотічильники виконуються вимірювання витрат теплоносія, температури на вході та виході а потім обчислює теплову енергію за заданим алгоритмом. Усі прилади які починають з 21 століття оснащуються мікроконтролерами, що забезпечують цифрову обробку сигналів та збереження результатів у внутрішній пам'яті.

Основними видами теплотічильників за класифікацією можна виділити пару видів: Механічні(тахометричні) або ультразвукові. [7]

Механічні за своєю будовою є найпростіші та найдешевші прилади, в яких вимірювання витрат визнається завдяки турбінам під дією поток теплоносія, якщо бути точніше, частота обертання пропорційна витрат рідини. Основним їх недоліком є чутливість до забрудження та зменшення точності визначень через повітря у системі і через ці моменти вони потребують більш частого обслуговування. Але через низьку ціну та простоту конструкцій цей тип часто є улюбленцем для домогосподарства. До механічних можна віднести Gross WMZ-UA Dn15 MBUS 110мм (Рис.1.) та Metron WMZ-UA 15/0.6 Mbus Class3 (Рис.2.)



Рис.1. Механічний теплотічильник Gross WMZ-UA Dn15 MBUS 110мм



Рис.2. Механічний теплотічильник Gross WMZ-UA 15/0.6 Mbus Class3

Ультразвукові в свою чергу вимірюють витрати теплоносія шляхом порівняння часу проходження ультразвукових сигналів уздовж і проти потоку. Завдяки цьому методу різниця часу дозволяє обчислити швидкість поток і, відповідно, витрату. Недоліком їх є також чутливість до повітря у воді і ступені її забруднення. Найголовнішим з усіх також є те що вони є набагато дорожче у порівнянні з механічними і це може спровокувати людей на покупку більш дешевих механічних теплотічильників. До ультразвукових теплотічильників можна віднести QALCOSONIC E3 M-BUS L270 Q15-DN50 FL (Рис.3.) та SHARKY 775 10,0 м³/ч DN40 (Рис.4.).

Зм.	Арк.	№ докум	Підпис	Дата

ХНТУ 174.КРМ.25.003 ПЗ

Арк.

18



Рис.3. Ультразвуковий теплोलічильник QALCOSONIC E3 M-BUS L270 Q15-DN50 FL



Рис.4. Ультразвуковий теплोलічильник SHARKY 775 10,0 м3/ч DN40

До найважливіших технічних характеристик будь то механічні чи ультразвукові теплोलічильник є:

- Визначення мінімального та максимального значення об'єму теплоносія, яке може коректно фіксувати теплोलічильник
- Визначення робочого температурного рівня
- Допустима похибка на цьому обладнанні

Зм.	Арк.	№ докум	Підпис	Дата

ХНТУ 174.КРМ.25.003 ПЗ

Арк.

19

- Інтерфейс обміну даних
- Міжповірочний інтервал-період, протягом якого лічильник зберігає метрологічні характеристики без повторного калібрування

Система обліку теплоенергії складається не тільки з самого теплолічильника, а також додатково з датчику температури, перетворювача витрат, комунікаційного та обчислювального модуля. [9]

Сучасні теплолічильники активно інтегруються в концепцію “розумного будинку” та IoT. Це дозволяє організувати віддалений моніторинг і керування системами опалення, створювати адаптивні моделі споживання енергії, а також автоматично формувати звітність для комунальних підприємств.

Завдяки використанню енергоефективних мікроконтролерів і бездротових технологій обміну даними, такі пристрої можуть працювати на батарейному живленні протягом 8-12 років без заміни елементів живлення. Крім того, деякі моделі підтримують функцію виявлення витоків теплоносія або несанкціонованого втручання у роботу приладу, що підвищує рівень безпеки та достовірність даних.

А отже, сучасні прилади обліку теплової енергії є високотехнологічними комплексами, що поєднують у собі точні сенсорні елементи, електронні обчислювальні вузли та комунікаційні засоби, їх застосування у складі автоматизованих систем дозволяє створити ефективну, надійну та прозору інфраструктуру енергоменеджменту на рівні окремої квартири або будинку. [11]

1.4 Сучасні тенденції в автоматизації та енергозбереженні

У сучасному світі питання енероефективності та автоматизації процесів управління споживанням ресурсів стає одним із ключових напрямів розвитку як промислових, так і побутових систем. Зростання вартості енергоресурсів, необхідність зниження навантаження на довкілля, а також прагнення до комфорту та цифровізації сприяли появі нових технологій у сфері обліку, регулювання та оптимізації теплової енергії.

Тенденції автоматизації нерозривно пов'язані з розвитком концепцій “розумного будинку” “розумного міста” та “Інтернету речей”. Завдяки цим технологіям можливо реалізувати не лише дистанційний контроль споживання енергії, а й інтелектуальні алгоритми, які самостійно приймають рішення на основі аналізу даних про температуру, вологість, погодні умови чи присутність людей у приміщенні.

Одним з ключових напрямів є впровадження АСКТП, яке дозволяє здійснювати моніторинг, регулювання та оптимізацію параметрів систем опалення, вентиляції та кондиціонування. Такі системи будуються на базі контролерів, сенсорів, виконавчих механізмів і програмного забезпечення, що забезпечує централізоване управління всіма підсистемами будівлі.

Особливо значення для цього є SCADA-системи, які забезпечують збір, візуалізацію та збереження даних у реальному часі. Завдяки ним можливо не лише спостерігати за роботою системи, але й формувати звіти про споживання енергоресурсів, виявляти неефективні ділянки та приймати управлінські рішення для зниження витрат. Сучасні SCADA-рішення підтримують інтеграцію з тепловими лічильниками, датчиками температури, тиску, вологості, а також із хмарними сервісами, що дозволяє віддалено керувати об'єктами через мережу Інтернет. Це особливо актуально для великих житлових комплексів, адміністративних будівель або промислових підприємств, де контроль над усіма вузлами теплопостачання здійснюється централізовано.

					<i>ХНТУ 174.КРМ.25.003 ПЗ</i>	Арк.
						21
Зм.	Арк.	№ докум	Підпис	Дата		

Інтернет речей відкриває новий рівень автоматизації, де всі пристрої взаємодіють між собою без участі людини. Наприклад, система опалення може автоматично зменшувати подачу тепла у години, коли мешканці відсутні, або навпаки – збільшувати подачу перед їх поверненням. Такі функції реалізуються за допомогою “розумних” терморегуляторів та сенсорних вузлів, які підключаються до єдиної мережі управління. Завдяки розвитку бездротових протоколів зв’язку, до прикладу Wi-Fi, Bluetooth, LoRa, ZigBee, NB-IoT, можливо створювати системи моніторингу, які не потребують прокладання додаткових кабелів. Це значно зменшує вартість впровадження автоматизації у все існуючі будівлі. Додатковим напрямом розвитку є використання хмарних технологій і великих даних для аналітики енергоспоживання. Зібрані дані аналізуються за допомогою спеціалізованих алгоритмів, що дозволяють прогнозувати потребу в енергії, визначати аномалії у споживанні та навіть прогнозувати поломки обладнання. [18]

Новітні розробки у галузі теплотехніки передбачають застосування адаптивних регуляторів, які використовують принципи машинного навчання для визначення оптимального режиму турботи. Такі системи враховують погодні дані, теплову інерційність будівлі, інтенсивність сонячного випромінювання та кількість людей у приміщенні. У результаті досягається мінімізація споживання тепла без зниження комфортних умов. Наприклад розумні терморостати, здатні автоматично формувати графік обігріву, на основі звичок користувачів. Вони мають доступ до мережі Інтернет, тому можуть отримувати дані про прогноз погоди, синхронізуватись з мобільним телефоном користувача та автоматично регулювати роботи опалювальної системи.

Сучасна енергетично політика більшості країн, у тому числі України, передбачає поступовий перехід до енергоефективного будівництва та управління. Це включає утеплення будівель, модернізацію систем опалення, застосування конденсаційних котлів, теплових насосів і тому подібне враховуючи також автоматизацію процесів споживання тепла.

Таким чином, автоматизація в енергозбереженні вже давно перестала бути лише технологічною новинкою – сьогодні це обов’язковий елемент сталого розвитку. Поєднання приладів обліку, систем автоматизованого керування, аналітичних платформ і алгоритмів дозволяє створювати комплексні рішення, що забезпечують економію ресурсів, екологічну безпеку та високий рівень технологісності.[20]

					ХНТУ 174.КРМ.25.003 ПЗ	Арк.
						23
Зм.	Арк.	№ докум	Підпис	Дата		

2. ПОСТАНОВКА ЗАДАЧІ ТА РОЗРОБКА КОНЦЕПЦІЇ СТРУКТУРИ АВТОМАТИЗОВАНОЇ СИСТЕМИ

2.1 Формулювання вимог до системи

Розробка автоматизованої системи поквартирного розподілу та обліку тепла потребує ретельного аналізу та постановки завдань, які вона має вирішувати, а також визначення вимог до її функціонування. Правильно сформовані вимоги для системи є головною основою для подальшого її проектування. Цей етап є найважливішим у процесі створення автоматизованої системи, адже він визнає загальні рамки, у межах яких буде здійснюватися реалізація проєкту.[8]

Вимоги формуються на основі аналізу об'єкта автоматизації, особливостей технологічного процесу теплопостачання, а також нормативних документів, які регламентують порядок обліку теплової енергії в житлових будівлях. Варто також враховувати потреби кінцевих користувачів-мешканців квартир, обслуговуючих організацій та постачальників теплової енергії.

АСОТЕ повина забезпечити комплексне вирішення всіх цих завдань, особливо якщо вони пов'язані з точним вимірюванням, збором, передачею, зберіганням та аналізом інформації про споживання тепла в межах квартири та усєї будівлі. Система має бути сконструювана на базі сучасних засобів вимірювання, мікропроцесорної техніки, систем зв'язку та обчислювальних засобів які здатні забезпечити повну автоматизацію процесів обліку тепла.

Якщо ж характеризувати більш детально завдання які мають виконувати ці вимоги, тоді можна сказати що:

– Висока точність вимірювань: система повинна забезпечувати достовірність даних які буде отримувати наша система враховуючи відповідно метрологічні вимоги та стандарти. Завдяки цьому моменту ми зможе уникнути помилок при розрахунках спожитої енергії

– Надійсність та безперервність роботи: система наша має функціонувати в стані цілодобовості, також потрібно врахувати щоб система могла продовжувати працювати при короткочасних перебоях живлення або зв'язку

– Гнучкість та масштабованість: Сама система має бути простою із можливістю розширення кількості підключених приладів без повної реконструкції нашої системи.

– Автоматизованість: усі процеси які були зазначені раніше має виконуватись автоматично без додаткового втручання людини в саму систему.

Функціональні вимоги визначають, що саме повинна робити система, які завдання виконувати і яким чином взаємодіяти з користувачами та зовнішніми системами. Як раз такими завданнями займається наше АСОТЕ. [13]

Якщо ж коротко описати головне завдання АСОТЕ, то вона збирає усі показники з кожного прибору обліку тепла, передає інформацію, архівірує їх, формує звітність.

Також варто не забувати і про технічні вимоги до нашої системи. Технічні вимоги є також важливим фундаментом вже після функціональних вимог, оскільки саме такі вимоги визначають набір обладнання, умови експлуатації, характеристики надійності та точності які нам знадобляться для будови нашої системи. Для систем поквартирного обліку тепла технічні вимоги мають врахуват як специфіку теплотехнічних процесів, так і обмеження, пов'язані з розміщенням обладнання в житлових приміщеннях. У контексті даної системи ключовими компонентами для цієї системи можна назвати вже раніше згадані мною пристрої: теплотічильники, контролери, комунікаційні модулі, джерела живлення, сервери збору даних та канали зв'язку між усіма елементами.

Теплотічильники повинні мати можливість вимірювати об'єм теплоносія, температуру подаючого та зворотного трубопроводів і обчислювати спожити

теплову енергію з високою точністю. Варто зазначити, що вони мабуть також додатково бути оснащеним інтерфейсом передачі даних, це можуть бути або Modbus, RS-485, M-Bus або інші інтерфейси.

Особливу увагу при побудові системи варто приділити увагу системі збору даних. Вона повинна забезпечувати стійке з'єднання з кожним лічильником, навіть за наявності перешкод або втрати сигналу. Для цього використовуються буферні алгоритми збереження даних у пам'яті пристрою до моменту відновлення зв'язку

Комунікаційну ж інфраструктуру можна реалізувати двома способами.

- Дротовим(M-Bus,RS-485) - для будинків з готовою інфраструктурою або невеликою кількістю пристроїв;
- Бездротовим(GSM,ZigBee, LoRaWAN) – більш характерна для нових об'єктів або місць, де прокладання кабелів неможливе або економічно не вигідне;

З раніше мною згаданою автономністю, потрібно також врахувати потребу у обладненні яке зпроможне працювати при відсутності на деякий час світла. Для деяких лічильників передбачаються літєві батареї з терміном служби до 6 років. Як що ж казати про вузли збору даних, то в них відсутне будь який акумулятор, тому для них буде характерні блоки безперебійного живлення або окремі резервні акумулятори. Наявність цих речей дасть стабільність роботи при аварійному відключенні електропостачання. [16]

Програмне забезпечення для систем обліку – це “мозок” автоматизованої системи, який відповідає за логіку збору, обробки, передачі, зберігання та візуалізації даних. Від якості написаного коду, налаштованої логіки, зручності інтерфейсу залежить ефективність системи а також можливості її контролювання. Перш за все, уся програмна частина повина мати модульну структуру, тобто бути побудованою за принципом розподілення функцій між окремими підсистемами, наприклад окремі модулі які відповідають за збір даних, комунікацій та адміністрування усієї системи. Кожен з усі цих моделей

повинен мати в собі чітко визначенні взаємодії, яке дозволяють при будь яких проблемах оновлювати або замінювати без повного переписування усієї системи з нуля. Також таке програмне забезпечення повино мати в собі підтримку будь якого протоколу зв'язку, вони забезпечують сумісність з більшістю теплотічильників і контролерів. Для більш централізованого збору даних нам потрібно використовувати базу даних, це може бути як MySQL, PostgreSQL або SQLite), яка зможе забезпечити цілісність інформації і швидкий доступ до неї. Разом зі збору даних має бути також і обробка, особливо не тільки зчитування показників, а й по можливості додатково визначати добові, місячні та сезонні витрати тепла, формувати графіки споживання, виявляти відхилення від норми, аналізувати динаміку тощо. Для цього можна застосовувати аналітичні алгоритми, що дозволяють прогнозувати споживання тепла на основі попередніх даних.

Інтерфейс користувача має бути інтуїтивно зрозумілим. Для мешканців це може бути звичайний кабінет, де відображаються споживання за день, тиждень, місяць тощо. Для обслуговуючого персоналу-це повноцінна модель моніторингу, яка дає можливість бачити всі прилади, їх стан, аномалії в тих чи інших зонах. Також найголовнішим для моделі моніторингу є аварійне сповіщення про наявність аварій на тій чи іншій ділянці.

Якщо робити висновки стосовно аналізу вимог до системи, то можна підсумувати що усі вимоги охоплюють як технічну частину, користувацьку та функціональну. Адекватна реалізація цих систем дозволить нам створювати надійні та сучасні системи обліку енергії які можуть бути зручні як для мешканця так і організація з надання тепlopостачання.

					<i>ХНТУ 174.КРМ.25.003 ПЗ</i>	Арк.
						27
Зм.	Арк.	№ докум	Підпис	Дата		

2.2 Вибір загальної архітектури системи

Вибір архітектури системи під час проектування є важливим кроком і найвідповідальнішим, адже від нього залежить не лише функціональність а й стабільність роботи усієї системи, можливості подальшої модернізації, розширення та інтеграції з іншими системами. Архітектура в собі визначає спосіб організації обміну даними між вузлами, характер взаємодії один з одним та вимоги до обчислювальних ресурсів. Для систем поквартирного обліку теплової енергії можна розглядати три основні архітектурні підходи: централізовану, децентралізовану та гібридну. Кожен з цих підходів має як переваги так і обмеження, які варто обдумати під час вибору оптимальних рішень.

Централізована архітектура передбачає наявність єдиного центрального вузла, який виконує усі функції, а це збір, обробка, зберігання та аналіз інформації. Усі поквартирні лічильники, датчики, вузли вимірювання та інші пристрої будуть підключатись до єдиного центрального вузла безпосередньо або через проміжні концентратори даних. Основа її полягає в тому, що усі процеси контролю та прийняття рішень здійснюються з єдиного центру, а периферійні пристрої виконують лише функції вимірювання та передачі сигналів. Типовим прикладом такої архітектури є SCADA-системи або центрального серверного комплексу, який приймає дані через дротові мережі M-BUS чи Ethernet від усіх вузлів обліку. Основним же його плюсом є повний контроль та моніторинг усіх параметрів у реальному часі та зручність збору даних, формування звітів. Але з недоліків це є залежність від головного вузла, якщо він відмовить то тоді буде втрачений повний контроль на усією системою, а також обмежена масштабованість при наявності великої кількості пристроїв. Така архітектура підійде більше невеликими житловим комплексам і будинкам, де кількість вузлів не перевищує більше сотні. Такий підхід забезпечує стабільну роботу, але в той же час потребує дуже якісної серверної

									Арк.
									28
Зм.	Арк.	№ докум	Підпис	Дата	ХНТУ 174.КРМ.25.003 ПЗ				

інфраструктури, резервного живлення та безперебійного інтернет-з'єднання. Також можливо проблема з витоком даних, бо усі данні сконцетровані в одному місці, що є гарною мішенню для тих кому ці данні будуть дуже потрібні.

Централізована архітектура передбачає розподілення функцій збору та обробки даних між кількома локальними вузлами або мікроконтролерами, які працюють автономно, обмінюючись лише необхідною інформацією з центральною частиною системи. [26]

Кожен квартирний або поверхневий вузол у цьому випадку є інтелектуальним модулем, здатним самостійно робити вимірювання, зберігати дані, проводити обробку інформації і передавати їх на центральний сервер за потребою.

Основною перевагою її є підвищена надійність, зниження навантаженість на вузли та гнучке налаштування. Але з мінусів це висока складність адміністрування та підвищена вимога до локальних контролерів.

З технічної частини, така архітектура все частіше реалізується завдяки використанню розподілених контролерів, до прикладу Siemens, WAGO, або бездротовими модулями передачі даних.

У такій системі навіть при відсутності зв'язку з сервером дані продовжують зберігатися локально і можуть бути передані пізніше, що робить систему стійкою до збоїв. Також така система дозволяє знизити споживання електрики завдяки тому що обробка даних відбувається на місці що дозволяє скоротити обсяг переданої інформації.

Гібридна архітектура поєднує в собі переваги двох раніше згаданих архітектур, що дозволяє забезпечити оптимальний баланс між централізованим контролем і розподіленим збором даних. У такій системі локальні вузли виконують попередню обробку та накопичення інформації, а центральний сервер відповідає за глобальний моніторинг, аналітику та адміністрування. Наприклад, кожен під'їзд може мати локальний концентратор даних, який зчитує показники з лічильників через M-Bus, а потім передає агреговану

інформацію на головний сервер завдяки Ethernet. Таким чином, навантаження на центральну систему значно зменшується, а стійкість і гнучкість системи підвищуються.

Основною її ж перевагою є поєднання стабільності з автономністю, можливість масштабування та зниження навантаження на мережу завдяки локальній обробці даних.

З мінусів це складна реалізація та обслуговування і висока початкова вартість через необхідність в додаткових контролерах.

З практичного погляду, гібридна система є найбільш доцільною і цікавою для сучасних систем поквартирного обліку теплової енергії у багатоповерхових будинках. Вона зможе забезпечувати необхідну точність, гнучкість, високу швидкість передачі даних і мінімальні ризики втрати інформації.

Також такий підхід створює умови для подальшої інтеграції з системами “розумного будинку”, де важливим є обмін інформацією між різними підсистемами-теплопостачанням, водопостачання та електропостачанням.

					ХНТУ 174.КРМ.25.003 ПЗ	Арк.
						30
Зм.	Арк.	№ докум	Підпис	Дата		

2.3 Структурна схема системи поквартирного обліку теплової енергії

Структурна схема є одні із найважливіших елементів проекту нашої системи, оскільки вона відображає логічну взаємодію всіх компонентів. Визначає потоки інформації, рівні ієрархії та функціональні зв'язки між технічними засобами. На основі структурної схеми здійснюється подальше розроблення апаратної частини, програмного забезпечення, а також формування алгоритмів обробки даних.

Система поквартирного розподілу теплової енергії за своєю структурою належить багаторівневих автоматизованих систем управління. Вона включає три основні рівні: польовий, рівень збору та передавання даних, диспетчерський. Кожен з них виконує власні функції, але всі вони заємопов'язані між собою через канали обміну інформацією.(Рис.5.).



Рис 5. Структурна схема системи поквартирного обліку теплової енергії

На польовому рівні розташовані всі вимірювальні прилади та датчики, що здійснюють безпосередній контроль параметрів теплоносія в кожній квартирі. Основними елементами в польовому рівні можна віднести теплотічильники, датчики температури, витратоміри, датчики тиску.

Усі ці елементи забезпечують безперервний збір даних у реальному часі. Залежно від моделі теплотічильників або іншого обладнання, передача інформації може здійснюватися як дротовими так і бездротовими технологіями як було раніше мною зазначено. Тож, основне завдання цього рівня полягає у зборі інформації попередній обробці а потім передачі її на наступний рівень.

Основним компонентом цього рівня є локальні контролери які зчитують показники усіх приладів які підключені до нього. Ці контролери можуть працювати на базі програмованих логічних контролерів або спеціальних модулів збору даних. Такими приладами можуть виступати WAGO 750, Siemens LOGO, або комунікаційний модуль типу Kamstrup MULCICAL.

На рівні збору та передачі даних відбувається логіка локального управління, тобто якщо відсутній зв'язок з центральним сервером, контролер продовжує працювати автономно, накопичуючи дані для подальшої передачі після відновлення зв'язку. Передавання даних до диспетчерського рівня може здійснюватися різними способами які я вже раніше зазначав.

Диспетчерський рівень є інформаційно-аналітичним центром системи. Тут же відбувається остаточна обробка, аналіз, зберігання даних а також можливостей коригування відповідних рішень в системі в операторській. Найголовнішими елементами тут є SCADA-система, робоче місце оператора та програмне забезпечення. Саме ж програмне забезпечення цього рівня можна реалізувати як через SCADA, використовуючи TraceMode та MasterSCADA, так і як веб-платформка з хмарним зберіганням даних. Важливим плюсом останньої дає доступ до інформації з будь якого пристрою до якою є підключення Інтернет. [29]

					<i>ХНТУ 174.КРМ.25.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		32

2.4 Аналіз можливих варіантів обліку теплопостачання

В цьому етапі необхідно здійснити детальний аналіз можливих варіантів обліку теплопостачання, визначити їх переваги, недоліки та обґрунтувати вибір оптимального рішення з технічної частини, економічної доцільності, надійності та зручності.

Першим ж можливим варіантом є спрощена система. В цьому випадку облік в кожній квартирі присутній теплолічильник, який веде облік спожитої енергії та

зберігає дані локально. Зчитування показників робиться вручну. Основною її перевагою є простота, адже ця система не вимагає дуже складної і дорогавартісної інфраструктури. Хоч цей варіант і є доволі простим у реалізації особливо з економічної точки зору, але він має в собі багато недоліків – неоперативність отримання даних, вплив людського фактору під час зчитування показників, відсутність можливостей дистанційного моніторингу та обмеженість у проведенні автоматичного аналізу споживання.

Другий варіант це викорисатння вже системи але з радіоканальною передачею даних. В даному випадку теплолічильники оснащуються радіомодулями, які передають показники на центральний вузол збору даних – контролер або концентратор, той в свою випадку передає інформацію вже на сервер. Такий підхід дозволяє автоматизувати вже процес зчитування показників у реальному часі, зменшує ймовірність помилок і виключає необхідність прокладання дротової мережі або зчитувати показники вручну. Ця система вже є набагато зручною для подальшої модернізації та забезпечує інтеграцію з програмними засобами моніторингу. Однак, її недоліками є обмежена дальність передачі сигналу, якість передачі даних залежить від радіо'зв'язку та перешкод особливо якщо це стосується багатоповерхівок, а також необхідності заміни елементів живлення радіомодулів. Таким чином, цей варіант вже є набагато цікавим та гнучким в плані модернізації, але потребує

					<i>ХНТУ 174.КРМ.25.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		33

додаткових витрат на обслуговування і не завжди гарантує стабільність передачі даних.

Третім варіантом є провідної системи збору даних, вона побудована на базі стандартних інтерфейсів, таких як RS-485, M-Bus або інші. У такій системі усі теплотічильники об'єднуються у єдину дротову мережу, з якої інформація надходить до контролера збору даних, а потім – на центральний сервер для збереження обробки. Цей підхід характеризується високою стабільністю сигналу та можливістю централізованого керування усіма приладами обліку. Крім того, система забезпечує масштабованість, тобто дає змогу додавати нові вузли без повної реконструкції. Основним недоліком цього рішення є потреба у прокладанні кабельної інфраструктури, що ускладнює монтаж у вже збудованих будинках, а також вищі початкові витрати на матеріали, обладнання та роботу. Хоча якщо аналізувати такі системи в перспективі, вони повністю компенсують усі витрачені на неї ресурси. [30]

Порівняння кожного способу обліку показує, що найбільш сучасним та вигідним є дротова система збору даних. Її головною перевагою серед інших є як раніше було зазначено це точність та стабільність передачі інформації. Воно ж дозволяє здійснювати моніторинг у реальному часі, вести архіви споживання, генерувати звіти та контролювати стан обладнання. Це дозволяє зменшити витрати на обслуговуючий персонал якому необхідно постійно ходити та збирати відповідні показники. Тому можна зробити висновки що дротова система має в собі усі необхідні переваги для системи обліку теплопостачання які навіть перебивають мінуси які характерні більше для короткострокової дії, тому така система буде дуже зручною для багатоквартирних будівлях.

3. ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Огляд та вибір теплोलічильників і датчиків

Як раніше мною билу зазначено, у процесі розбροці системи поквартиного розподілу і обліку теплової енергії є вибір технічного обладнання. Саме від точності, стабільності та надійності цього обладнання залежить ефективність та якість функціонування цієї системи. Прилади, які можна використовувати для обліку теплової енергії, мають відповідати низці технічних вимог на основі нашої ідеї. Зокрема, вони повинні забезпечувати можливість високоточного витрат теплоносія, також підтримувати стандартні інтерфейси для передачі даних до систем збору та обробки інформації. Для забезпечення сумісності всіх компонентів, усі вони мають бути сертифіковані до вимог чинних стандартів, наприклад ДСТУ EN 1434, який регламентує метрологічні характеристики теплोलічильників. [12]

Також умови експлуатації відіграють важливу роль у підборі обладнання. У багатоквартирних будинках головним критерієм є простота монтажу, низькі вимоги до обслуговування та стабільність показників упродовж тривалого часу та у подальшому інтеграції з автоматизованими системами.

Гідравлічні та температурні коливання грають хоч і меншу але тим не менш свою роль у підборі обладнання. Найголовніше в ньому є те, щоб теплोलічильник забезпечував точність вимірювань при умові нестабільності тиску або коливань температури.

Серед усього асортименту лічильників, до системи можна взяти або Danfoss SonoSelect 10 (Рис. 6.) або QUNDIS Q heat 5 (Рис. 7.). Серед усіх представлених лічильників на ринку, ці два показали більшу надійність та готовність до інтеграції автоматизованої системи.

Обидва прилади – це сучасні електронні теплोलічильники які призначені для обліку теплової енергії. При виборі цих лічильників важливо

					<i>ХНТУ 174.КРМ.25.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		35

також враховувати їх сумісність з інтерфейсами збору даних, можливості діагностики, вимоги до монтажу й сервісу, а також вартість і доступність документації та інструментів для інтеграції.



Рис. 6. Теплолічильник Danfoss SonoSelect 10

Danfoss SonoSelect 10 реалізує ультразвуковий принцип вимірювання, що забезпечує відсутність рухомих частин, високу стабільність показників. Тому він є стійким до механічного зносу та тим часом підвищує термін служби в експлуатації. QUNDIS Q heat 5 в свою також чергу використовує ультразвукове вимірювання але його характеристики більш характерні для обліку з акцентом на енергоефективність та низьке власне енергоспоживання.

Якщо казати за інтерфейси зв'язку до у Danfoss є зазвичай доступний M-Bus, або опціонально RS-485/ModBus через шлюзи. Також він доволі простий в діагностиці, що сильно полегшує вирішення несправностей та додає віддалений моніторинг. QUNDIS в свою чергу також має ці інтерфейси зв'язку але він має в собі свою екосистему для масових зборів даних, тобто “End to End” рішення яке гарне підходить для ЖКГ, тому його набагато простіше підключити до існуючих AMS/AMI рішень і готових хмарних платформ.

Зм.	Арк.	№ докум	Підпис	Дата

ХНТУ 174.КРМ.25.003 ПЗ

Арк.

36

Процес монтажу цих двох є в цілому однаковий, але Danfoss зазвичай постачає такі прилади в модульному промисловому корпусі, іноді навіть з більшим діапазоном номінальних діаметрів. QUNDIS в свою чергу орієнтується на компактність та легкість монтажу в типових вузлах опалення, що робить його зручним для поквартирного встановлення без втручання у інженерні мережі.



Рис 7. Теплолічильник QUNDIS Q heat 5

Тому якщо аналізувати ті два лічильника то для нашої системи підходять обидва варіанти, але QUNDIS Q heat 5 є набагато оптимальнішим вибором для створення сучасної та надійної системи обліку теплової енергії у будинку. Він забезпечує точні вимірювання, є сумісним з більшістю систем автоматизації та є простим у встановленні.

Зм.	Арк.	№ докум	Підпис	Дата

ХНТУ 174.КРМ.25.003 ПЗ

Арк.

37

3.2 Використання мікроконтролерів та вузлів збору даних

В будь якій автоматизованій системі обліку обробка інформації, вузли збору даних є важливою частиною всієї системи, яка забезпечує інтеграцію вимірювальних приладів з центральною системою моніторингу. В таких системах мікроконтролери найголовніші, вони як раз таки виконують функцію збору показників, обробка та фільтрація від шумів а потім передача вже через дротовий або бездротовий спосіб інформації результатів. [17]

Для реалізації необхідної нам системи ми можемо використовувати мікроконтролери сімейства STM32 або ESP32. Обидва варіанти мають високі обчислювальні можливості, широкий набір периферійних інтерфейсів і доволі таки непогану підтримку сучасних технологій зв'язку.

Мікроконтролери STM32 на базі архітектури ARM Cortex-M характеризуються низьким енергоспоживанням, високою продуктивністю та стабільною роботою в умовах промислових середовищ. Вони підтримують апаратну реалізацію протоколів UART та SPI, завдяки цьому, це дозволяє нам легко підключати різноманітні сенсори та комунікаційні модулі. Крім того, такі мікроконтролери мають іноді вбудовані таймери, аналогоцифрові перетворювачі і можливості працювати в енергозберігаючому режимі.

У свою чергу, ESP32 - це сучасні мікроконтролери із двоядерним процесором та вбудованими модулями Wi-Fi та Bluetooth, що робить його ідеальним рішенням для реалізації бездротового обміну даними. Завдяки підтримці протоколів MQTT, HTTP та TCP/IP ці контролери можуть безпосередньо передавати інформацію на сервери моніторингу або у хмарні сервіси без додаткового обладнання та налаштування. Його обчислювальні можливості дозволяють виконувати не лише збір інформації з датчиків, а й локальну обробку цієї інформації, наприклад фільтрацію аномальних показників або виявлень збоїв у роботі лічильників. Використання мікроконтролера в системі поквартирного обліку забезпечує гнучкість

конфігурації та масштабованість. За потреби можна легко розширити кількість підключених приладів, змінити частоту збору даних або підключити нові модулі, до прикладу через LoRaWAN або NB-IoT. Мікроконтролерна архітектура дозволяє створювати як локальні вузли збору інформації так і централізовані пункти збору даних. Крім того, важливим аспектом є енергоефективність вузлів збору даних. [19]

Більшість сучасних мікроконтролерів, зокрем STM 32 або ESP32, мають розширені можливості енергозбереження, в яких соживання струму знижується до одиниць мікроампер. Це дозволяє забезпечити автономну роботу пристрою від батарей протягом, кількох років, що є важливою перевагою при впровадженні систем у житлових будинках, де доступ до обладнання може бути обмежений. Вибір мікроконтролера також визначається експлуатаційними вимогами. У разі централізованого встановлення вузла збору даних у щитових або технічних приміщеннях, краще використовувати STM32, завдяки його високій стабільності та промисловій надійності. Якщо ж передбачено бездротову передачу даних без проводів, то оптимальним є ESP32.

Таким чином, у розроблюваній системі доцільно використовувати мікроконтролер ESP32, який забезпечує необхідний рівень обчислювальної потужності, підтримку бездротових технологій зв'язку, простота програмування та висока енергоефективність. Застосування ESP32 дозволяє реалізувати сучасну та масштабну систему яка буде відповідати усім вимогами поквартирного обліку теплової енергії у будинку.

3.3 Мережеві технології для передачі даних

Передавання даних у різних системах може реалізовуватись як за допомогою провідних, так і бездротових технологій як було зазначено раніше. Кожен з цих підходів має свої особливості, переваги, недоліки.

Провідні технології традиційно вважаються більш надійними завдяки фізичному з'єднанню пристроїв, що мінімізує вплив зовнішніх чинників і гарантує стабільну передачу даних. У промислових та житлових системах обліку частіше використовують RS-485, який за своєю характеристикою є простою структурою з широкою підтримкою сучасних контролерів. Цей інтерфейс забезпечує диференційний спосіб передачі сигналу, завдяки чому інформація може передаватися на відстань до 1200 метрів без суттєвої втрати якості. Це робить технологію зручною для застосування у великих будинках.

Комунікація за стандартом RS-485 зазвичай відбувається за допомогою протоколу ModBus RTU, який є одним із найпоширеніших у системах автоматики. Його просто дозволяє швидко реалізовувати обмін даними між контролерами та периферійними пристроями. Також є можливість інтегрування системи в існуючі вже мережі диспетчеризації або обліку. Водночас недоліком його як мною згадані раніше будь які дротові технології це складність монтажу. Серед сучасних бездротових технологій у сфері автоматизації варто виділити Wi-Fi, LoRaWAN та NB-IoT. Кожна з них має свої технічні особливості і особливо це важливо якщо враховувати різні умови експлуатації. [21]

Технологія Wi-Fi є найпоширенішою у побутовому середовищі. Вона забезпечує високу швидкість обміну даними, що дозволяє передавати навіть великі обсяги інформації у режимі реального часу. Використання Wi-Fi у системах обліку теплової енергії є зручним, якщо в будинку наявна стабільна мережа з достатнім покриттям. У такому випадку можна використовувати контролери ESP32, завдяки підключенню безпосередньо до локальної мережі та надсилати показники на сервер без потреби у додаткових систем. Основним недоліком

його це високе споживання електроенергії, через що Wi-Fi менш підходить для автономних пристроїв, які працюють від батареї. На відміну Wi-Fi, технологія LoRaWAN призначення для передачі невеликих обсягів даних на великі відстані. Вона характеризується надзвичайно низьким споживанням, що дозволяє працювати роками без заміни елементів живлення. Завдяки великому радіусу дій LoRaWAN є оптимальним рішенням, коли потрібно охопити значну кількість квартир або будинків у межах декількох під'їздів. Її основна перевага-стабільна робота навіть у складних умовах забудови та можливості централізованого збору інформації з великої території. [22]

Ще одним сучасним підходом до передачі даних є NB-IoT – це технологія, що базується на інфраструктурі мобільних операторів. Вона забезпечує високу надійність зв'язку та стабільність роботи навіть у місцях зі слабким покриттям. Ця технологія дозволяє передавати показники лічильників безпосередньо на сервер, минаючи також проміжні вузли збору. До її переваг можна віднести низьке енергоспоживання та можливість підключення великої кількості пристроїв на одній базі. Але з головних її недоліком є потреба у SIM-картах або спеціальних тарифах операторів, що дещо збільшує експлуатаційні витрати. Таким чином, для реалізації системи обліку та розподілу тепла можна використовувати будь яку технологію, але варто відштовхуватись від багатьох моментів які були мною зазначені раніше, особливо від виду контролера. В нашому ж випадку можна використовувати Wi-Fi або LoRaWAN, вони є для нас більш сучасними та доступними технологіями для реалізації автоматизованої системи. [25]

4. ПРОГРАМНЕ ЗАПЕЧЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

4.1 Архітектура програмного забезпечення

Архітектура програмного забезпечення в автоматизованій системі, визначає спосіб взаємодії між апаратною частиною, алгоритмами обробки, базами даних та користувацьким інтерфейсом. Грамотно спроектована архітектура забезпечує стабільність роботи системи, її можливості до масштабування, а також простоту подальшого обслуговування та модернізації. У сучасних системах обліку перевага надається модульному або багаторівневому підходу, що дозволяє розділити програмний комплекс на окремі логічні блоки з чітко визначеними функціями.

Загалом архітектуру даної системи доцільно реалізувати у вигляді трьох основних рівнів :рівень збору даних, обробки та відображення результату.

На першому рівні функціонують програмні модулі, що взаємодіють із фізичним обладнанням: теплові лічильники, сенсори температури, датчики витрати тощо. Це може бути реалізовані через драйвери, протоколи зв'язку які ми раніше зазначали, або адаптери, які перетворюють в цифровий сигнал. Такий рівень іноді можна зустріти під назвою “інтерфейс з обладнанням”, адже він забезпечує електронний міст між реальним фізичним процесом і програмною частиною системи.[24]

Другий рівень є центральним елементом архітектури та містить логіку обробку отриманих даних. Тут же відбувається валідація показників, виявлення аномалій, перевірка на відповідність, усереднення даних та подальша їх трансформація у формати, які будуть придатні для розрахунків. На цьому етапі виконуються алгоритми визначення спожитої теплової енергії, її розподілу між квартирами, формування аналітичних звітів, прогнозування витрат та інше. Важливо підкреслити, що ефективність функціонування системи значною

мірою залежить саме від правильно реалізованої бізнес-логіки, адже диктує порядок обчислень та забезпечує коретність кінцевих результатів. [5]

Третій рівень - це рівень візуалізації та користувацької взаємодії, на якому інформація становиться зрозумілою для кінцевого користувача. Тут застосовуються веб-технології (HTML,CSS,JAVASCRIPT), конструктори графіків, дашборди та інтерактивні елементи, що дозволяють переглядати історію споживання, формувати звіти, експортувати дані або керувати налаштуваннями систем. Користувач отримує можливість віддалено контролювати стан обладнання, аналізувати показники та приймати чітко обґрунтовані рішення щодо теплового споживання. Цей рівень також відповідає за безпеку доступу, авторизацію та розмежує права доступу. [10]

Окремої уваги заслуговує питання масштабованості та гнучкості, яке також закладається на рівні архітектурного проектування. Система повинна підтримувати можливість додавання нових пристроїв або будинків без додаткових витрат або кардинально переписування усієї системи. Для цього як раз може застосовуватись мікросервісний або модульний підхід, за кожен компонент працює як окрема служба, що може оновлюватися незалежно від інших. Такий принцип дозволяє легко адаптувати системи під нові стандарти обліку або змінювати структуру бази даних без повного перезапуску. Для другого рівня буде важливий спосіб обміну даними через API та основною логікою взаємодією з базою даних, де можуть зберігатися архівні дані, конфігурації системи, журнали подій тощо. Зв'язок можна здійснювати через RESTful або WebSocket, що дозволяє реалізувати як одноразовий запит, так і постійний потік даних у реальному часі. У разі інтеграцій з зовнішніми системами, така архітектура дозволяє забезпечити високий рівень сумісності та стандартизації інформації.[14]

Таким чином, архітектура програмного забезпечення автоматизованої системи має ключове значення для адекватної роботи, можливості розвитку, масштабування з іншими системами.

4.2 Алгоритми розрахунку спожитої теплової енергії

Алгоритми розрахунку спожитої енергії визначають фінальне значення енергоспоживання кожного житлового приміщення та формують основу для розрахунку вартості опалення та дозволяє здійснювати справедливий і юридично коректний розподіл теплової енергії між мешканцями будинку. Результати таких обчислень повинні бути не лише точними, але й відтворюваними, тобто такими, що можуть бути перевірені і підтвердженні у разі необхідності, наприклад, під час аудиту або вирішення спірних ситуацій.

Для визначення кількості теплової енергії зазвичай використовується стандартна фізична формула:

$$Q = m * c * (t_1 - t_2)$$

Де

Q - кількість теплової енергії(кВт * год або Гкал),

m - маса або витрата теплоносія (кг або м³),

c - теплоємність теплоносія,

t₁ і t₂ - температури теплоносія на вході та виході системи

У реальних системах часто використовується витрата або одиницю часу, тому формула набуває інтегрального характеру, і розрахунок ведеться не разово, а з урахуванням безперервного потоку даних. У цьому разі доцільно застосовувати наступний вираз:

$$Q = \int_{t_0}^{t_n} G(t) * c * (T_{\text{подача}}(t) - T_{\text{зворотне}}(t)), dt$$

Зм.	Арк.	№ докум	Підпис	Дата

Де $G(t)$ - миттєва витрата теплоносія, що змінюється у часі. Такий підхід дозволяє максимально точно врахувати зміни в тепловому навантаженні і забезпечує достовірність даних при поквартирному обліку. [15]

Сучасні лічильники тепла нерідко вже містять вбудовані алгоритми розрахунку, і система отримує готові цифрові значення. Однак у випадках поквартирного розподілу часто виникає потреба у вторинних обчисленнях, наприклад, для коригування результатів відповідно до фактичних умов або нормативів. Система має враховувати дуже багато параметрів такі як температура, режим роботи, періоди неперервного або переривчастого користування, тепловтрат.

На етапі розрахунку може застосовуватися метод балансового розподілу, коли спочатку визначається кількість теплової енергії, що увійшла до будинку, після чого відбувається її пропорційний розподіл між квартирами. Формула має вигляд:

$$Q_i = Q_{\text{загал}} * \frac{A_i}{\sum A_i}$$

Q_i - споживання теплової енергії в i -ій квартирі,

$Q_{\text{загал}}$ - загальне споживання будинку,

A_i - коригуючий коефіцієнт (площа, тепловтрати, кількість кімнат тощо).

Такий підхід дозволяє врахувати різницю умов проживання в різних квартирах і підвищити точність розподілу.

У деяких випадках застосовуються предиктивні алгоритми, що базуються на методах машинного навчання або статистичного аналізу. Наприклад, на основі історичних даних можна здійснювати прогнозування споживання на наступний день або місяць, що допомагає оптимізувати роботу теплового пункту. Для цього можуть використовуватися моделі ARIMA, методи ковзного середнього або нейронні мережі. Використання таких алгоритмів підвищує нергоефективність будинку та дозволяє попереджати перевитрати ресурсів.

Зм.	Арк.	№ докум	Підпис	Дата

Також важливим елементом є перевірка обчислень на коректність. У системі повинна бути реалізована логіка, яка порівнює отримані значення з типовими, і у випадку різких відхилень формує повідомлення про можливу помилку, аварію або втручання в систему. Це сприяє підвищенню надійності обліку та захисту від неасанкціонованого впливу на показники. [19]

Таким чином, алгоритми розрахунку спожитої теплової енергії є не просто математичними інструментом, а комплексною системою процедур, що забезпечують юридично коректний, технічно обґрунтований та економічно справедливий облік теплової енергії на рівні окремої квартири. Вони дають можливість реалізувати принцип “платиш лише за своє”, що відповідає сучасним європейським стандартам енергофактивності та прозорості у сфері комунального господарства.

Зм.	Арк.	№ докум	Підпис	Дата

4.3 Розробка користувацького інтерфейсу

Користувацький інтерфейс є частиною взаємодії користувача з системою, адже саме через нього здійснюється взаємодія між людиною та програмним забезпеченням. Незалежно від складності внутрішніх алгоритмів обробки даних чи серверної архітектури, кінцевий користувач отримує результат лише через інтерфейс, тому зручність, логічність та зрозумілість його структури є визначальними факторами ефективності всієї системи. У сучасних автоматизованих комплексах присутній принцип “user-friendly”, він є обов’язковою, бо користувачі можуть мати різний рівень технічної підготовки та досвід роботи з подібними рішеннями.

Розробка інтерфейсу розпочинається зі структурування інформаційних блоків, визначення типів користувачів та їхніх сценаріїв використання. Наприклад, оператор теплопункту, мешканець квартири та диспетчер комунального підприємства можуть мати принципово різні ролі та рівні доступу. Це означає, що інтерфейс повинен бути адаптивним та здатним підлаштовуватися під тип користувача: одні мають доступ до графіків та прогнозів, інші - лише до загальної інформації та історії споживання. Такий підхід дозволяє уникнути перевантаження зайвими функціями і, водночас, підвищує точність управління системою.

На концептуальному рівні інтерфейс можна поділити на інформаційний, аналітичний та керуючий блок. Інформаційний блок відображає поточні показники лічильників, статус обладнання та попередження про можливі несправності. Аналітичний блок включає в собі графіки, таблиці, статистичні параметри та історичні дані, що дозволяє оцінити динаміку споживання теплової енергії. Керуючий блок, у свою чергу, може давати змогу змінювати параметри системи, виконувати дистанційні команди або налаштовувати періодичність зчитування даних або інші елементи. Важливо, щоб взаємодія

між блоками не вимагала складної технічної підготовки, а логіка навігації була інтуїтивно зрозумілою.

Особливе значення має візуалізації даних, оскільки графічне представлення є найбільш доступним способом сприйняття інформації. Наприклад, динаміка споживання тепла за день або місяць у графічному форматі часто є більш наочною, ніж числова таблиця. Це дозволяє швидко виявити пікові навантаження, періоди неефективного використання або потенційні тепловтрати. Використання кольорових індикаторів, дашбордів та інтерактивних елементів робить систему ще більш зручною для прийняття, що особливо важливо для оперативного управління тепловими ресурсами. Ще одним критичним аспектом є логіка авторизації та безпеки доступу.[27] Інтерфейс повинен підтримувати розмежування прав на перегляд, редагування та адміністрування, щоб уникнути несанкціонованого втручання. Для цього застосовуються ролі користувачів з відповідними дозволами. Кожна дія в системі повинна бути логованою, що дозволяє вести прозорий облік подій та забезпечує юридичну значимість усіх операцій. Не менш важливою характеристикою є адаптивність інтерфейсу, тобто здатність правильно відображатися на різних типах пристроїв - від персональних комп'ютерів до планшетів або смартфонів. Такий підхід відповідає сучасним тенденціям і дозволяє користувачу отримувати доступ до системи в будь-який час та з будь-якого місця. У багатьох випадках саме мобільний інтерфейс стає основним каналом взаємодії з системою, особливо для мешканців квартир.[23]

Важливо підкреслити, що інтерфейс не є лише графічною оболонкою, а й виконує також функцію посередника між користувачем та алгоритмами системи. Саме через нього реалізуються сценарії роботи, формується логіка використання, здійснюється аналіз даних та прийняття рішень. Фактично, інтерфейс - це інструмент який перетворює складну технічну систему на доступний і зрозумілий інструмент для широкого кола користувачів.[28]

4.4 Веб-інтерфейс

Веб-інтерфейс як було зазначено нами раніше, є головним посередником між користувачем та системою. Використання стеку HTML, CSS та JavaScript дозволяє створити кросплатформене рішення з широкими можливостями для візуалізації, інтерактивності та адаптації під різні типи екранів. Веб-технології забезпечують швидку розробку та простоту інтеграції з серверними API, що робить їх оптимальним вибором для реалізації дашбордів обліку теплових ресурсів.

Перша версія нашого Dispatcher Dashboard Heat Energy(Додаток А) була побудована як веб-інтерфейс диспетчера для контролю обліку теплової енергії в квартирах(Рис 8.), із можливістю відображення даних у реальному часі. Важливою частиною сайту є сервер на Node.js, який відповідає за отримання і зберігання даних про показники квартир, розподіл теплової енергії та стан клапанів. Всі дані споживання тепла, площу квартири і алокацію зберігаються в масиві об'єктів, кожен з яких містить унікальний ідентифікатор квартири, її назву, показник обліку тепла. Саме цей масив використовується для формування таблиці на сайті, а також для розрахунку пропорційного, рівномірного або розподілу за площею.

Для відображення даних у реальному часі був реалізований механізм Server-sent Events, який дозволяє серверу надсилати оновлення фронтенду без необхідності постійного опитування. Кожна подія, що передається через SSE, обгортається в спеціальний формат, де вказується тип події, час її виникнення та дані, наприклад зміни у розподілі тепла, стан клапана або аварійні показники температури. Це дозволяє відображати в окремому журналі подій усі важливі зміни та аварійні стани, що є критично важливим для диспетчера.

Наш веб-інтерфейс складається з кількох блоків. Перший блок(Рис. 9) - це загальна інформація про будинок, де відображається кількість квартир, загальний облік та розподілений обсяг.

Використання fetch API дозволяє робити запити до серверу для отримання актуальних даних або відправки команд без перезавантаження сторінки(Рис. 14). SSE забезпечує миттєве оновлення відображення без потреби у ручному оновленні. Inline-стилі та структура HTML були підібрані таким чином, щоб забезпечити чітку візуалізацію інформацію, легку навігацію і швидкий доступ до ключових системних налаштувань для диспетчера. Таким чином, перший сайт являє собою інтегровану систему диспетчерського контролю, де фронтенд і бекенд працюють як єдине ціле для забезпечення реального часу відображення даних, управління тепловими потоками, ведення журналу подій і графічної аналітики. Кожен блок коду виконує свою важливу функцію, а їхнє поєднання дозволяє створити ефективну диспетчерську систему для контролю теплової енергії.

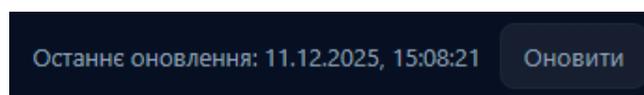


Рис. 13. Кнопка оновлення з інформацією

4.5 Серверна частина веб-додатку

Наш Node.js - сервер виконує центральну роль у системі диспетчерського контролю теплової енергії. Він забезпечує зберігання даних про квартири, їхні показники обліку тепла, площу та вже виділену алокацію. Кожна квартиру описується об'єктом з полями id, name, reading, area та allocated. Масив таких об'єктів дозволяє легко отримувати інформацію для формування таблиць, графіків і SCADA-панелі, а також служить основою для розрахунку пропорційного, рівномірного або площового розподілу тепла.

Для інтерактивності та роботи в реальному часі на сервері реалізований механізм. Кожен підключений клієнт отримує події у форматі text/event-stream. Сервер відправляє повідомлення при зміні стану клапана, зміні розподілу тепла або при виникненні аварійної ситуації, наприклад перегріву або низької температури в квартирі. Функція broadcastEvent форматує дані у вигляді JSON і надсилає їх усім підключеним клієнтам. Завдяки цьому фронтенд автоматично оновлює показники без ручного опитування серверу.

Сервер також додатково обробляє POST-запити для керування клапаном та ручного або автоматичного розподілу тепла. Коли диспетчер натискає кнопку відкриття або закриття клапана(Додаток Б), відправляється POST-запит з параметром action. Север обробляє його, змінює стан клапана і відразу транслює оновлення всім клієнтам через SSE. Таким чином, система забезпечує миттєве відображення стану і дозволяє віддалено регулювати потоки тепла без фізичного доступу.

Алгоритми розподілу тепла реалізовані у вигляді функцій(Додаток В), що виконуються при отриманні POST-запиту на /api/auto-allocate. Якщо обрано пропорційний метод, сервер підсумовує всі показники квартирних лічильників і визначає частку кожної квартири у загальному обсязі доступного тепла. Для рівномірного методу серверу ділить загальний доступний обсяг на кількість квартир, а для методу за площею обчислюється частка кожної квартири від

Зм.	Арк.	№ докум	Підпис	Дата

загальної площі. Після розрахунку оновлені дані квартир знову транслюються через SSE для фронтенду, що забезпечує синхронність даних на всіх відкритих сторінках.

SCADA-модуль на сервері реалізований як масив об'єктів, що містить параметри квартир, включно з температурою, потоком та станом системи. Значення температури та потоку моделюються з випадковими коливаннями, щоб імітувати реальну роботу теплової системи. Якщо температура виходить за допустимі межі, сервер генерує аварійну подію і передає її через SSE. Це дозволяє диспетчеру оперативно реагувати на критичні ситуації, відображати їх у журналі подій і SCADA-панелі. Таким чином, SCADA моделює реальні умови роботи системи і забезпечує інтерактивним контроль за параметрами кожної квартири.

Ще однією важливою особливістю є модуль журналу подій. Кожна зміна стану клапана, розподілу тепла або аварійна подія записується в масив подій, який транслюється на фронтенд. Це дозволяє вести історію дій і подій, аналізувати роботу системи, оцінювати ефективність розподілу тепла та виявляти квартири з підвищеним ризиком аварій. Журнал подій є критично важливим для диспетчера, оскільки він дозволяє відстежувати причини відхилень і своєчасно реагувати на будь-які проблеми.

Сервер також включає періодичне оновлення SCADA-даних за допомогою `setInterval`, що імітує зміну температури та потоку в реальному часі. Алгоритм перевіряє температуру кожної квартири, змінює статус на `OVERHEAT` або `LOW_TEMP` при виході за допустимі межі і генерує відповідну подію для всіх клієнтів. Це дозволяє диспетчеру не тільки бачити поточної значення, а й отримувати сигнал про аварійну ситуацію миттєво. У цілому, Node.js-сервер і SCADA-модуль працюють як єдине ціле, забезпечуючи повну інтеграцію фронтенду і бекенду, миттєве оновлення даних, віддалене керування системою, автоматичний і ручний розподіл теплової енергії. Така архітектура забезпечує надійність, гнучкість і масштабованість системи,

дозволяє легко додавати нові квартири, підключати додаткові сенсори або інтегрувати зовнішні системи моніторингу, що робить її ефективним інструментом для диспетчера теплової енергії.

<i>Зм.</i>	<i>Арк.</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>

ХНТУ 174.КРМ.25.003 ПЗ

Арк.

56

4.6 SCADA-система для диспетчерського контролю

SCADA-система в нашій системі створена як частина веб-інтерфейсу диспетчера, де кожна квартира візуалізується як окремий блок із показниками температури, потоку та стану системи. Дані SCADA моделюються на сервері у вигляді масиву об'єктів, де кожен об'єкт містить ід квартири, температури потім і статус які ми визначали раніше.

На сервері кожні кількі секунд виконується перерахунок значень SCADA(Додаток Г). Алгоритм оновлює температуру і потік для кожної квартири і визначає статус квартири залежно від температури: якщо температура перевищує 28 градусів, статус має OVERHEAT, якщо падає нижче 18 - LOW_TEMP, інакше залишається ОК. У випадку виникнення аварійного стану сервер формує спеціальне SSE-повідомлення, яке містить тип події, назву квартири тощо. Це дозволяє фронтенду оперативно отримувати інформацію про критичні ситуації і відображати їх у SCADA-панелі та журнал подій.

Фронтенд обробляє SSE-події і оновлює SCADA-панель у реальному часі. Кожен блок квартир на панелі має змінювати колір або відображати індикатор у залежності від статусу: зелений ОК, червоний OVERHEAT, синій відтінок для LOW-TEMP. Таким чином, диспетчер миттєво бачить проблемні квартири і може швидко реагувати на аварії, відкриваючи або закриваючи клапани через відповідні кнопки керування. SCADA також інтегрується з модулем розподілу тепла і клапанами. Коли відбувається автоматичний або ручний розподіл тепла, сервер оновлює масив SCADA, включаючи значення алокації для кожної квартири. Це дає змогу диспетчеру бачити не лише температуру а й потік, і реальний розподіл тепла. Інтеграція з клапанами може бути реалізована через POST-запити на сервер; зміна стану клапана відразу відображається у SCADA через SSE. Таким чином забезпечується повна синхронізація даних між фізичними параметрами системи, розподілом тепла та візуалізацією на панелі.

					<i>ХНТУ 174.КРМ.25.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		57

Ще однією важливою функцією SCADA є журнал подій. Кожна аварійна ситуація або критичне відхилення від нормальних параметрів і записується і передається на фронтенд. Диспетчер бачить у журналі всі події із зазначенням часу, квартири та типу події як дозволяють їх відслідковувати та реагувати на відповідні зміни. Комбінування SCADA-панелі та журналу подій забезпечує повний контроль і зручний огляд стану будинку в цілому. Також є можливість моделювання та візуалізації параметрів у режимі реального часу. Це досягається завдяки регулярному оновленню сервером значень температури і потоку, обчисленню статусу квартир і трансляції через SSE.

Таким чином SCADA-система у нашому проєкті виступає ядром диспетчерського контролю. Вона забезпечує реальний час моніторингу температури, потоку і стану кожної квартири, інтеграцію з алокацією тепла і керування клапанами, а також ведення журналу подій. Це дозволяє диспетчеру ефективно управляти системою, своєчасно реагувати на критичні ситуації, оцінювати ефективність розподілу тепла і підпримувати безпеку так комфорт мешкаців. SCADA поєднує в собі моделювання, відображення і контроль у єдну інтегровану систему, що є ключовим компонентом у сучасних диспетчерських рішеннях для житлових комплексів.

4.7 База даних та обмін через API

База даних в нашому випадку слугує центральним вузлом, у якому акумулюються, впорядковуються та зберігаються всі необхідні дані: від сирової телеметрії з теплोलічильників до результатів розрахунку фактичного теплового споживання кожної окремої квартири.

З огляду на характер процесів у системі, база даних повинна забезпечувати стабільність, відмовостійкість, масштабованість та можливість оброблення великих масивів даних у режимі близькому до реального часу. Дані, що надходять від теплोलічильників, зазвичай мають поштовий характер - тобто передаються періодично, але з фіксованою частотою (наприклад, кожні 5–15 хвилин). У межах багатоквартирного будинку це може становити сотні або тисячі записів протягом доби. Тому вибор СУБД є важливим етапом проектування.

Для подібних систем доцільним є застосування реляційної бази даних, наприклад PostgreSQL або MySQL, оскільки облік теплової енергії передбачає структуровані взаємопов'язані сутності: будинки, квартири, вузли обліку, користувачі, покази лічильників, розрахункові значення. Реляційна модель дозволяє чітко визначити зв'язки «один-до-багатьох», «багато-до-одного» та забезпечує логічну цілісність даних, що є критично важливим при формуванні офіційної звітності для енергетичних компаній або ОСББ.

У спрощеному вигляді структурна модель бази даних складається з таких основних таблиць:

- buildings - містить загальну інформацію про будинок: адресу, кількість квартир, технічні характеристики системи опалення;
- apartments - інформація про кожну квартиру, включаючи належність до будинку та унікальний ідентифікатор користувача;
- meters - опис теплोलічильників, їх тип, серійний номер, дата встановлення та поточний стан;

- measurements - основна таблиця для накопичення показів: температура подачі, температура звороту, витрата теплоносія, час зчитування, похибки, службові позначення;
- heat_calculations - результати математичної обробки та розрахунків, включаючи обсяг переданої теплоти, коефіцієнти корекції тощо;
- users - облікові записи мешканців або операторів системи;
- roles - дозволи та рівні доступу до інформації.

Важливо підкреслити, що кожен запис показів у таблиці measurements має містити не лише основні параметри, а й часову мітку, адже на її основі будуються графіки, аналітика та звіти, а також розраховується інтегральний обсяг спожитої теплоти за період. Це забезпечує можливість гнучкого аналізу даних за будь-який проміжок часу.

Оскільки система містить клієнтську частину (веб-інтерфейс), серверну частину (Node.js / Python Flask / Django), SCADA- диспетчерський модуль та модуль збору телеметрії з лічильників, взаємодія між ними потребує єдиного уніфікованого протоколу обміну. Таку роль відіграє API (Application Programming Interface). У даному проєкті використовується REST-архітектурний підхід до побудови API, оскільки він дозволяє організувати зрозумілий, масштабований і стандартизований обмін даними між усіма компонентами системи. Основні принципи REST - це незалежність клієнта і сервера, використання HTTP-методів (GET, POST, PUT, DELETE), а також передавання інформації у форматі JSON.

Системний API можна розділити на кілька груп:

- API збору даних від лічильників - використовується модулем зв'язку, який передає покази на сервер.
- API розрахункових операцій - надає можливість запуску алгоритмів або отримання результатів.

- API для користувацького веб-інтерфейсу - забезпечує роботу особистого кабінету.
- Адміністративний API для диспетчерської SCADA-системи - для перегляду аварійних станів, останніх телеметричних даних та графічного моніторингу.

REST - підхід дозволяє гнучко масштабувати систему: наприклад, при збільшенні кількості будинків можна просто підняти додаткові інстанси серверного застосунку без зміни архітектури.

					<i>ХНТУ 174.КРМ.25.003 ПЗ</i>	Арк.
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		61

ВИСНОВКИ

У ході виконання даного проєкту була створена комплексна автоматизована диспетчерська система, що дозволяє здійснювати контроль, моніторинг та керування тепловою енергією на рівні окремих квартир. Основною метою роботи було не лише аналіз комплектуючих для таких систем, а й створення програмного інтерфейсу для відображення даних обліку. Розроблений веб-сайт диспетчера виконує роль центрального вузла для збору, обробки та візуалізації даних. Він дозволяє відображати показники теплової енергії, обсяг споживання, розподіл між квартирами у реальному часі.

Веб-інтерфейс включає зручні таблиці, інтерактивні графіки та SCADA-панель, що дозволяє візуалізувати параметри кожної квартири, такі як потік, та стан системи. Завдяки цьому оператор може оперативнo реагувати на будь-які відхиленн та приймати рішення щодо коригування розподілу тепла. Особлива увага приділена була реалізації SCADA-модулю, який забезпечує віддалений моніторинг та керування системою в режимі реального часу.

Для забезпечення повноцінної роботи системи була інтегрована серверна частина на базі Node.js, яка відповідає за зберігання даних, обробку запитів від клієнтів, розрахунок алокацій теплової енергії за різними методами та реалізацію SSE для оновлення даних у реальному часі. Використання Node.js дозволило організувати ефективну взаємодію між фронтендом і бекендом, забезпечуючи стабільну роботу системи навіть при високій частоті оновлень SCADA-даних.

Розроблена система також включає модуль обробки подій та журналування, що дозволяє вести детальний облік усіх дій користувача та змін стану системи. Це важливо як для аналітики, так і для подальшого удосконалення алгоритмів розподілу та виявлення можливих проблем.

У процесі реалізації були враховані принципи масштабованості та модульності. Веб-інтерфейс і серверна частина побудована таким чином, що до

					<i>ХНТУ 174.КРМ.25.003 ПЗ</i>	Арк.
						62
Зм.	Арк.	№ докум	Підпис	Дата		

них легко можна додати нові функції, інтегрувати додаткові квартири або підключати зовнішні системи збору даних. Це забезпечує довгострокову перспективу розвитку системи та можливість її адаптації під різні типи житлових комплексів.

Таким чином, підсумовуючи виконану роботу, можна зробити що створена диспетчерська система є повноцінним інструментом для контролю та управління тепловою енергією на рівні квартир. Вона поєднує в собі функції обліку, розподілу, моніторингу та SCADA-контролю що робить її ефективним засобом для забезпечення комфорту мешканців, оптимізації використання енергоресурсів та зниження ризиків аварійних ситуацій.

					ХНТУ 174.КРМ.25.003 ПЗ	Арк.
						63
Зм.	Арк.	№ докум	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бондар-Підгурська О. В. Науково-методичні підходи до оцінки енергоефективності як фактора конкурентоспроможності промислової продукції в інноваційній моделі розвитку України. Наукові праці Кіровоградського національного технічного університету. Економічні науки. Кіровоград, 2012. Вип. 22, ч.ІІ. С. 470.
2. ДСТУ 3755-98. Енергозбереження. Номенклатура показників енергоефективності та порядок їхнього внесення у нормативну документацію. [Чинний від 1999-07-01]. Вид. офіц. Київ : Мінрегіонбуд України, 1999. 8 с.
3. Енергоефективність як ресурс інноваційного розвитку: Національна доповідь про стан та перспективи реалізації державної політики енергоефективності у 2008 році / С.Ф. Єрмілов, В.М. Геєць, Ю.П. Яценко, В.В. Григоровський, В.Е. Лір та ін. Київ : НАЕР, 2009. 93 с.
4. Рубан-Максимець О.О. Особливості розрахунку показників енергетичної ефективності на базі статистичної звітності України. Проблеми загальної енергетики. Київ, 2009. № 20. С. 21–26.
5. ДБН В.2.6-31:2016. Теплова ізоляція будівель. – К. : Міністерство регіонального розвитку, будівництва, архітектури та житлово-комунального господарства України, 2017. – 30 с.
6. Фаренюк Г.Г. Основи забезпечення енергоефективності будинків та теплової надійності огорожувальних конструкцій. Київ : Гама - Принт, 2019. 216 с.
7. Про затвердження Методики визначення енергетичної ефективності будівель : наказ Міністерства регіонального розвитку, будівництва та житлово-комунального господарства України від 11.07.2018 р. №169. Офіційний вісник України. 2018. № 55. С. 301.

					<i>ХНТУ 174.КРМ.25.003 ПЗ</i>	Арк.
Зм.	Арк.	№ докум	Підпис	Дата		64

8. Ковалко М.П., Денисюк С.П. Енергозбереження – пріоритетний напрямок державної політики України : Українські енциклопедичні знання. Київ, 1998. 512 с.
9. ДБН В.2.6-33:2018. Конструкції зовнішніх стін із фасадною теплоізоляцією. Вимоги до проєктування. Київ : Мінрегіонбуд України. 2018. 25 с.
10. ДСТУ 9190:2022. Енергетична ефективність будівель. Метод розрахунку енергоспоживання під час опалення, охолодження, вентиляції, освітлення та гарячого водопостачання. Київ : Мінрегіонбуд України. 2023. 156 с.
11. Про енергетичну ефективність будівель : Закон України від 22.06.2017 р. № 2118-VIII. Голос України. 2017. №134. С. 18.
12. ДСТУ Б В.2.5-38:2008 «Улаштування блискавкозахисту будівель та споруд».
13. ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою»;
14. Боженко М.Ф. Джерела теплопостачання та споживачі теплоти / М.Ф.Боженко, В.П.Сало. – К.: ІВЦ «Видавництво «Політехніка»», 2004. – 192с.
15. Боженко М.Ф. Джерела теплопостачання та споживачі теплоти. Методичні вказівки. – К.: НТУУ «КПІ», ТЕФ. 2008. – 64с.
16. Прядко, М. О. Теплові мережі : навчальний посібник / М. О. Прядко, В. І. Павелко, С. М. Василенко – К.: Алерта, 2005. – 227 с.
17. Поржезінський Ю.Г. Основи проєктування водо підготовки ТЕЦ і котелень харчових підприємств: Навч. посіб. – К.: НУХТ, 2008. – 206с.
18. Філоненко В.Н. Нагнітачі та теплові двигуни. – Мет. Вказ. до вивчення дисципліни для студентів спеціальності “Теплоенергетики” ден. та заочн. форм навчання.– К.: НУХТ. – 2004.– 50с.
19. A Reliable Communication Framework and Its Use in Internet of Things (IoT) / Tanweer Alam // Department of Computer Science, Faculty of Computer and

Зм.	Арк.	№ докум	Підпис	Дата

- Information Systems, Islamic University of Madinah, Saudi Arabia, - 2018. - №5: International Journal of Scientific Research in Computer Science, Engineering and Information Technilogy. – p.450-456
20. Узденов Т. А. Алгоритми диспетчеризації для grid систем з невідчужуваними ресурсами // Збірник матеріалів VIII Всеукраїнської науково-практичної конференції молодих вчених «Наукова молодь-2020» (Київ, 21 жовтня 2020 р.). – К.: ФОП Ямчинський О.В., 2020. – с. 186
21. Bejarano A , Fernandez B , Jimeno M , Salazar A, and Wightman P. Towards the Evolution of Smart Home Environments: A Survey [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/307557956_Towards_the_Evolution_of_Smart_Home_Environments_A_Survey
22. Кирик В.В. Мікропроцесорна техніка: Навчальний посібник з дисципліни для всіх форм навчання та студентів іноземців напряму підготовки 6.050701 “Електротехніка та електротехнології” .- К.: ІВЦ «Видавництво «Політехніка», 2014.- 183 с.
23. Боженко М.Ф. Джерела теплопостачання та споживачі теплоти: навч. посіб. / М.Ф. Боженко, В.П. Сало. – Київ : ІВЦ «Видавництво «Політехніка», 2004. – 192 с.
24. Matias Linan-Reyes , Joaquin Garrido-Zafra , Aurora Gil-de-Castro and Antonio Moreno-Munoz Article Energy Management Expert Assistant, a New Concept // Departamento de Ingeniería Electrónica y de Computadores, Escuela Politécnica Superior de Córdoba, Campus de Rabanales, Universidad de Córdoba, Edificio Leonardo Da Vinci, E-14071 Cordoba, Spain. – 2021. – 38 p.
25. Узденов Т. А. Алгоритми диспетчеризації для grid систем з невідчужуваними ресурсами // Збірник матеріалів VIII Всеукраїнської науково-практичної конференції молодих вчених «Наукова молодь-2020» (Київ, 21 жовтня 2020 р.). – К.: ФОП Ямчинський О.В., 2020. – с. 186

Зм.	Арк.	№ докум	Підпис	Дата

26. Технології інтернету речей. Навчальний посібник [Електронний ресурс]: навч. посіб. для студ. спеціальності 126 «Інформаційні системи та технології», спеціалізація «Інформаційне забезпечення робототехнічних систем» / Б. Ю. Жураковський, І.О. Зенів; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 12,5 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 271 с
27. Автоматичне керування системою опалення [Електронний ресурс]. – Режим доступу: <https://reneco.com.ua/stati-ua-novini/avtomatichnekeruvannya-sistemoju-opalennya/>
28. Одарущенко О.Б. Верифікація програмного забезпечення програмованих логічних контролерів із використанням математичних блоків дискретного перетворення інформації // Збірник наукових праць науковопрактичної конференції професорськовикладацького складу Полтавської державної аграрної академії за підсумками науково-дослідної роботи в 2018 році (м. Полтава, 16-17 травня 2019 року). – Полтава : РВВ ПДАА, 2019. – с. 52
29. Енергетична ефективність України. Кращі проектні ідеї [електронне видання] : Проект «Професіоналізація та стабілізація енергетичного менеджменту в Україні» / Уклад.: С.П. Денисюк, О.В. Коцар, Ю.В. Чернецька. – К. : КПІ ім. Ігоря Сікорського, 2016. – 79 с
30. Пуховий І.І. Теплонасосне та безпосереднє використання теплової енергії довкілля і її потенціал в Україні / І.І. Пуховий // Енергетика: економіка, технології, екологія. – 2005. – № 1. – 92–97 с.

Зм.	Арк.	№ докум	Підпис	Дата

Додаток А. Код веб-інтерфейсу

```
<!doctype html>
<html lang="uk">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>Диспетчер — Облік та розподіл теплової енергії</title>
  <!-- Прості стилі: сучасна, чиста панель -->
  <style>
    :root{--bg:#0f1724;--card:#0b1220;--muted:#94a3b8;--accent:#fb923c}
    *{box-sizing:border-box;font-family:Inter,Segoe UI,Roboto,Arial,sans-serif}
    body{margin:0;background:linear-gradient(180deg,#071022 0%,#081226
100%);color:#e6eef8}
    header{display:flex;align-items:center;justify-content:space-between;padding:18px
28px;border-bottom:1px solid rgba(255,255,255,0.03)}
    .brand{display:flex;gap:12px;align-items:center}
    .logo{width:44px;height:44px;border-radius:8px;background:linear-
gradient(135deg,var(--accent),#ffb86b);display:flex;align-items:center;justify-
content:center;color:#07203a;font-weight:700}
    .title{font-size:18px;font-weight:600}
    .controls{display:flex;gap:10px;align-items:center}
    .btn{background:rgba(255,255,255,0.06);border:1px solid
rgba(255,255,255,0.04);padding:8px 12px;border-radius:8px;color:var(--
muted);cursor:pointer}
    .container{padding:20px 28px;display:grid;grid-template-columns:300px 1fr;gap:20px}
    .card{background:linear-
gradient(180deg,rgba(255,255,255,0.02),rgba(255,255,255,0.01));padding:16px;border-
radius:12px;border:1px solid rgba(255,255,255,0.03)}
    .sidebar .section{margin-bottom:14px}
```

```

.kpi {display:flex;flex-direction:column;gap:10px}
.kpi .item {display:flex;justify-content:space-between;align-
items:center;padding:8px;border-radius:8px;background:rgba(255,255,255,0.01)}
main .top {display:flex;gap:16px;margin-bottom:14px}
.chart-wrap {flex:1;min-height:220px}
.list-wrap {width:420px}
table {width:100%;border-collapse:collapse;font-size:14px}
th,td {padding:10px;text-align:left;border-bottom:1px dashed rgba(255,255,255,0.03)}
th {color:var(--muted);font-size:13px}
tr:hover td {background:rgba(255,255,255,0.01)}
.small {font-size:13px;color:var(--muted)}
.muted {color:var(--muted)}
.pill {background:rgba(255,255,255,0.03);padding:6px 8px;border-radius:999px;font-
weight:600}
.actions {display:flex;gap:8px}
input[type=number] {width:100px;padding:6px;border-radius:6px;border:1px solid
rgba(255,255,255,0.04);background:transparent;color:inherit}
.footer {padding:12px 28px;color:var(--muted);font-size:13px}
/* Responsive */
@media (max-width:920px) { .container {grid-template-columns:1 fr;} .list-
wrap {width:100%;} }
</style>
</head>
<body>
<header>
<div class="brand">
<div class="logo">TE</div>
<div>
<div class="title">Диспетчер — Облік теплової енергії</div>
<div class="small">Панель управління розподілом та моніторингом</div>

```

```
</div>
</div>
<div class="controls">
  <div class="small muted">Останнє оновлення: <span id="lastUpdated">—
</span></div>
  <button class="btn" id="refreshBtn">Оновити</button>
  <button class="btn" id="exportBtn">Експорт CSV</button>
</div>
</header>

<div class="container">
  <!-- Бічна панель: загальні KPI і фільтри -->
  <aside class="sidebar">
    <div class="card section">
      <div style="font-weight:700;margin-bottom:6px">Загальна інформація</div>
      <div class="kpi">
        <div class="item"><div>Всього квартир</div><div class="pill" id="totalApts">—
</div></div>
        <div class="item"><div>Загальний облік (Гкал)</div><div class="pill"
id="totalEnergy">—</div></div>
        <div class="item"><div>Розподілено (Гкал)</div><div class="pill"
id="allocatedEnergy">—</div></div>
      </div>
    </div>

    <div class="card section">
      <div style="font-weight:700;margin-bottom:8px">Фільтри</div>
      <div class="small">Швидко відобразити квартири за станом</div>
      <div style="display:flex;gap:8px;margin-top:8px">
        <button class="btn" onclick="filter('all')">Всі</button>
```

```
<button class="btn" onclick="filter('underconsumption')">Невелике  
споживання</button>
```

```
<button class="btn" onclick="filter('overconsumption')">Високе  
споживання</button>
```

```
</div>
```

```
</div>
```

```
<div class="card section">
```

```
<div style="font-weight:700;margin-bottom:8px">Автоматичний розподіл</div>
```

```
<div class="small">Розподілати за пропорцією показників або вручну.</div>
```

```
<div style="display:flex;flex-direction:column;gap:8px;margin-top:10px">
```

```
<label class="small">Метод розподілу</label>
```

```
<select id="allocMethod" class="btn" style="padding:10px">
```

```
<option value="proportional">Пропорційно до показників</option>
```

```
<option value="equal">Рівномірно</option>
```

```
<option value="area">За площею квартири</option>
```

```
</select>
```

```
<button class="btn" id="runAlloc">Запустити розподіл</button>
```

```
</div>
```

```
</div>
```

```
</aside>
```

```
<!-- Головна частина: графік + список квартир -->
```

```
<main>
```

```
<div class="top">
```

```
<div class="card chart-wrap">
```

```
<div style="display:flex;justify-content:space-between;align-items:center;margin-  
bottom:8px">
```

```
<div style="font-weight:700">Динаміка споживання (останні 7 днів)</div>
```

```
<div class="small muted">Гкал</div>
```

```
</div>
<canvas id="consumptionChart" style="width:100%;height:220px"></canvas>
</div>

<div class="card list-wrap">
  <div style="display:flex;justify-content:space-between;align-items:center;margin-
bottom:8px">
    <div style="font-weight:700">Останні події</div>
    <div class="small muted">Події системи</div>
  </div>
  <div id="eventsList" class="small muted">Немає подій</div>
</div>

<div class="card">
  <div style="font-weight:700;margin-bottom:10px">Список квартир та
лічильників</div>
  <div style="overflow:auto;max-height:420px">
    <table id="apartmentsTable">
      <thead>
        <tr><th>Квартира</th><th>Показник (Гкал)</th><th>Площа
(м2)</th><th>Алокація (Гкал)</th><th></th></tr>
      </thead>
      <tbody></tbody>
    </table>
  </div>
</div>
</main>
</div>
```

```
<div class="footer">Побудовано як прототип для дипломної роботи  
"Автоматизована система поквартирного розподілу і обліку теплової енергії" — тут  
реалізовано: перевірка показників, автоматичний розподіл, експорт даних, базові  
візуалізації.</div>
```

```
<!-- Скрипти: використовуємо Chart.js CDN для графіку. Логіка — вбудована JS  
для демонстрації. -->  
<script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.0/dist/chart.umd.min.js"></script>  
<script>  
  // --- Демонстраційні дані ---  
  const apartments = [  
    {id:1,name:'Кв. 1',reading:1.24,area:48,allocated:0},  
    {id:2,name:'Кв. 2',reading:0.96,area:36,allocated:0},  
    {id:3,name:'Кв. 3',reading:2.18,area:72,allocated:0},  
    {id:4,name:'Кв. 4',reading:0.50,area:30,allocated:0},  
    {id:5,name:'Кв. 5',reading:1.02,area:40,allocated:0}  
  ];  
  
  let totalAvailable = 6.0; // Гкал — загальний обсяг теплової енергії для розподілу  
(можна отримувати з АСКОЕ)  
  
  // Ініціалізація панелі  
  function init(){  
    renderKPI();  
    renderTable(apartments);  
    renderChart();  
    updateEvents('Система ініціалізована');  
    document.getElementById('lastUpdated').textContent = new Date().toLocaleString();  
  }  
</script>
```

```

function renderKPI(){
  document.getElementById('totalApts').textContent = apartments.length;
  const total = apartments.reduce((s,a)=>s+a.reading,0).toFixed(2);
  document.getElementById('totalEnergy').textContent = total;
  const allocated = apartments.reduce((s,a)=>s+a.allocated,0).toFixed(2);
  document.getElementById('allocatedEnergy').textContent = allocated;
}

```

```

function renderTable(list){
  const tbody = document.querySelector('#apartmentsTable tbody');
  tbody.innerHTML = "";
  list.forEach(a=>{
    const tr = document.createElement('tr');
    tr.innerHTML = `
      <td><div style="font-weight:600">${a.name}</div><div class="small
muted">ID:${a.id}</div></td>
      <td>${a.reading.toFixed(2)}</td>
      <td>${a.area}</td>
      <td><input type="number" step="0.01" value="${a.allocated.toFixed(2)}" data-
id="${a.id}" class="allocInput"></td>
      <td class="actions"><button class="btn"
onclick="applyManual(${a.id})">Застосувати</button></td>
    `;
    tbody.appendChild(tr);
  });
  // підключаємо обробник для зміни вручну

```

```

document.querySelectorAll('.allocInput').forEach(inp=>inp.addEventListener('change',e=>
{

```

```
    const id=+e.target.dataset.id; const val=parseFloat(e.target.value)||0; const
ap=apartments.find(x=>x.id===id); ap.allocated=val; renderKPI();
  }));
}
```

```
function applyManual(id){
  const ap = apartments.find(x=>x.id===id);
  updateEvents(`Вручну задано алокацію для ${ap.name}: ${ap.allocated.toFixed(2)}
Гкал`);
  renderKPI();
}
```

```
function updateEvents(text){
  const el = document.getElementById('eventsList');
  const ts = new Date().toLocaleTimeString();
  el.innerHTML = `<div style=\"margin-bottom:8px\"><strong>${ts}</strong> —
${text}</div>` + el.innerHTML;
}
```

// --- Розподіл: реалізовано 3 методи ---

```
function runAllocation(){
  const method = document.getElementById('allocMethod').value;
  if(method==='proportional') allocateProportional();
  else if(method==='equal') allocateEqual();
  else if(method==='area') allocateByArea();
  renderTable(apartments); renderKPI(); renderChart();
  updateEvents(`Розподіл виконано методом: ${method}`);
}
```

```
function allocateProportional(){
```

```

const sumRead = apartments.reduce((s,a)=>s+a.reading,0);
apartments.forEach(a=>{ a.allocated = sumRead? (a.reading/sumRead)*totalAvailable
: 0; });
}
function allocateEqual(){
const per = totalAvailable / apartments.length;
apartments.forEach(a=>a.allocated = per);
}
function allocateByArea(){
const sumArea = apartments.reduce((s,a)=>s+a.area,0);
apartments.forEach(a=>{ a.allocated = sumArea? (a.area/sumArea)*totalAvailable : 0;
});
}

```

```

document.getElementById('runAlloc').addEventListener('click',runAllocation);
document.getElementById('refreshBtn').addEventListener('click',()=>{
document.getElementById('lastUpdated').textContent=new Date().toLocaleString();
updateEvents('Оновлення даних (симуляція)'); });
document.getElementById('exportBtn').addEventListener('click',exportCSV);

```

```

function exportCSV(){
const rows = [['id','name','reading_gcal','area_m2','allocated_gcal]];
apartments.forEach(a=>rows.push([a.id,a.name,a.reading.toFixed(3),a.area,a.allocated.toFixed(3)]));
const csv = rows.map(r=>r.join(',')).join('\n');
const blob = new Blob([csv],{type:'text/csv;charset=utf-8;'});
const url = URL.createObjectURL(blob);
const a = document.createElement('a'); a.href=url;
a.download='apartments_allocation.csv'; a.click(); URL.revokeObjectURL(url);

```

```

    updateEvents('Експортовано CSV');
  }

function filter(mode){
  if(mode==='all') renderTable(apartments);
  else if(mode==='underconsumption')
renderTable(apartments.filter(a=>a.reading<1.0));
  else if(mode==='overconsumption') renderTable(apartments.filter(a=>a.reading>1.5));
  updateEvents('Застосовано фільтр: '+mode);
}

// --- Графік ---
let chart=null;
function renderChart(){
  const labels = Array.from({length:7},(_,i)=>{ const d=new Date();
d.setDate(d.getDate()-6+i); return d.toLocaleDateString(); });
  const data = labels.map( (_,i)=>{
    // синтетичні значення: сума показників + випадкове коливання
    const base = apartments.reduce((s,a)=>s+a.reading,0);
    return parseFloat((base + Math.sin(i/2)*0.2).toFixed(2));
  });
  const ctx = document.getElementById('consumptionChart');
  if(chart) chart.destroy();
  chart = new Chart(ctx, {
    type:'line',
    data: {labels, datasets:[{label:'Споживання',data,fill:true,tension:0.35}]},
    options: {scales: {y: {beginAtZero:true}}}
  });
}

```

```
// --- Ініціалізація ---  
init();  
</script>  
</body>  
</html>
```

Додаток Б. Код відкриття та закриття клапану

```
let valveState = 1;

app.post('/api/valve', (req, res) => {
  const { action } = req.body;

  if (action === 'open') valveState = 1;
  else if (action === 'close') valveState = 0;
  else return res.status(400).json({ error: 'unknown action' });

  broadcastEvent({
    type: 'valve',
    valve: valveState,
    ts: new Date()
  });

  res.json({ status: 'ok', valve: valveState });
});
```

Додаток В. Код з імітації зміни температури

```
const totalAvailable = 6.0;
```

```
app.post('/api/auto-allocate', (req, res) => {
```

```
  const method = req.body.method;
```

```
  if (method === 'proportional') {
```

```
    const sum = apartments.reduce((s, a) => s + a.reading, 0);
```

```
    apartments.forEach(a => {
```

```
      a.allocated = sum ? +(a.reading / sum * totalAvailable).toFixed(3) : 0;
```

```
    });
```

```
  }
```

```
  if (method === 'equal') {
```

```
    const per = +(totalAvailable / apartments.length).toFixed(3);
```

```
    apartments.forEach(a => a.allocated = per);
```

```
  }
```

```
  if (method === 'area') {
```

```
    const sumArea = apartments.reduce((s, a) => s + a.area, 0);
```

```
    apartments.forEach(a => {
```

```
      a.allocated = sumArea ? +(a.area / sumArea * totalAvailable).toFixed(3) : 0;
```

```
    });
```

```
  }
```

```
  broadcastEvent({
```

```
    type: 'allocation',
```

```
    method,
```

```
    apartments,
```

```
    ts: new Date()  
  });
```

```
  res.json(apartments);  
});
```

Додаток Г. Код розподілу тепла

```
const express = require("express");
const app = express();
const PORT = 3000;

app.use(express.json());

let mockData = [
  { id: 1, apartment: 101, temperature: 22.5, flow: 12.3, status: "OK" },
  { id: 2, apartment: 102, temperature: 29.1, flow: 10.1, status: "OVERHEAT" },
  { id: 3, apartment: 103, temperature: 18.2, flow: 9.7, status: "LOW_TEMP" }
];

// API для SCADA-панелі
app.get("/api/scada-data", (req, res) => {
  res.json(mockData);
});

// Симуляція оновлення даних (імітація реального контролера)
setInterval(() => {
  mockData.forEach(d => {
    d.temperature += (Math.random() - 0.5);
    d.flow += (Math.random() - 0.5);

    if (d.temperature > 28) d.status = "OVERHEAT";
    else if (d.temperature < 18) d.status = "LOW_TEMP";
    else d.status = "OK";
  });
}, 3000);
```

```
app.listen(PORT, () => console.log(`SCADA сервер працює на порту ${PORT}`));
```