

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра електроніки та телекомунікацій

(повне найменування кафедри)

## КВАЛІФІКАЦІЙНА РОБОТА ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «МАГІСТР»

**ІНТЕЛЕКТУАЛЬНА СИСТЕМА КЕРУВАННЯ ДОМАШНІМ  
ОПАЛЕННЯМ З ВИКОРИСТАННЯМ RASPBERRY PI 4**

**INTELLIGENT HOME HEATING CONTROL SYSTEM USING  
RASPBERRY PI 4**

спеціальність 171 Електроніка  
(шифр і назва спеціальності)

освітня програма «Електроніка»  
(назва освітньої програми)

Виконав: здобувач вищої освіти  
групи ЕЛМ-21  
Сидорук Віталій Володимирович

\_\_\_\_\_  
(підпис)

Керівник: к.ф.-м.н., доцент  
Хвищун Микола Вячеславович

\_\_\_\_\_  
(підпис)

Кваліфікаційну роботу  
допущено до захисту  
«   » грудня 2025 р.  
Гарант освітньої програми:  
к.т.н., доцент  
Заблоцький Валентин Юрійович

\_\_\_\_\_  
(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра електроніки та телекомунікацій

Ступінь вищої освіти: магістр

Галузь знань: 17 Електроніка та телекомунікації

Спеціальність: 171 Електроніка

Освітня програма: «Електроніка»

ЗАТВЕРДЖУЮ

Завідувач кафедри

«\_\_» \_\_\_\_\_ 2025 р.

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*Сидоруку Віталію Володимировичу*

1. Тема кваліфікаційної роботи: *Інтелектуальна система керування домашнім опаленням з використанням Raspberry Pi 4*

Керівник роботи: *к.ф.-м.н., доцент Хвищун Микола Вячеславович*

затвержені наказом закладу вищої освіти від «14» січня 2025 р. № 20/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи: 05.12.2025 р.

3. Вихідні дані до роботи: *інформаційні матеріали для одноплатного комп'ютера Raspberry Pi 4, набір давачів, технічні характеристики периферійних інтерфейсів (UART, SPI, I<sup>2</sup>C, USB, CAN). Структурований план розробки апаратної та програмної частини системи*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

*Вступ*

*Розділ 1 Аналіз вихідних даних та обґрунтування теми кваліфікаційної роботи магістра*

*Розділ 2 Теоретична частина*

*Розділ 3 Комп'ютерне моделювання системи*

*Розділ 4 Спеціальна частина*

*Висновки*

5. Перелік графічного (ілюстративного) матеріалу

*Презентація PowerPoint 12 слайдів.*

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз вихідних даних та обґрунтування теми кваліфікаційної роботи магістра</i>	<i>Хвищун М. В., доцент</i>		
<i>Теоретична частина</i>	<i>Хвищун М. В., доцент</i>		
<i>Комп'ютерне моделювання системи</i>	<i>Хвищун М. В., доцент</i>		
<i>Спеціальна частина</i>	<i>Хвищун М. В., доцент</i>		
<i>Висновки</i>	<i>Хвищун М. В., доцент</i>		
<i>Нормоконтроль</i>	<i>Селепина Й. Р., доцент</i>		
<i>Гарант ОП</i>	<i>Заблоцький В. Ю., доцент</i>		
<i>Показник запозичень тексту</i>		_____ %	
<i>Академічна доброчесність</i>	<i>Хвищун М. В., доцент</i>		

7. Дата видачі завдання 03.02.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Розділ 1 Аналіз вихідних даних та обґрунтування теми кваліфікаційної роботи магістра</i>	до 18.09.2025 р.	
2.	<i>Розділ 2 Теоретична частина</i>	до 26.10.2025 р.	
3.	<i>Розділ 3 Технічне забезпечення вирішення поставленого завдання</i>	до 02.11.2025 р.	
4.	<i>Розділ 4 Спеціальна частина</i>	до 09.11.2025 р.	
5.	<i>Висновки</i>	до 16.11.2025 р.	
6.	<i>Формування списку використаних джерел</i>	до 23.11.2025 р.	
7.	<i>Оформлення ілюстративного матеріалу</i>	до 28.11.2025 р.	
8.	<i>Нормоконтроль</i>	до 01.12.2025 р.	
9.	<i>Інструментальна перевірка на академічний плагіат</i>	до 05.12.2025 р.	
10.	<i>Представлення кваліфікаційної роботи магістра до захисту</i>	до 30.12.2025 р.	

**Здобувач вищої освіти**

\_\_\_\_\_ (підпис)

**Сидорук В. В.**

\_\_\_\_\_ (прізвище, ініціали)

**Керівник кваліфікаційної роботи**

\_\_\_\_\_ (підпис)

**Хвищун М. В.**

\_\_\_\_\_ (прізвище, ініціали)

## АНОТАЦІЯ

Сидорук В. В. Інтелектуальна система керування домашнім опаленням з використанням Raspberry Pi 4. Рукопис. Кваліфікаційна робота магістра ОП «Електроніка» спеціальності 171 «Електроніка». Луцький національний технічний університет. Луцьк, 2025. 80 с.

Кваліфікаційна робота магістра складається зі вступу, чотирьох розділів, висновків, переліку використаних джерел, додатів.

У магістерській роботі розроблено прототип інтелектуальної системи керування домашнім опаленням, що забезпечує автоматизований контроль температурного режиму, моніторинг параметрів мікроклімату та оптимізацію енергоспоживання з використанням одноплатного комп'ютера Raspberry Pi 4.

Система реалізує збір даних із температурних, вологісних та інфрачервоних датчиків, здійснює аналіз отриманих показників і формує керуючі сигнали до виконавчих елементів (реле, електрокотел, сервоприводи клапанів), забезпечуючи точне регулювання роботи опалювального обладнання.

Розроблена система може бути інтегрована у житлові будинки та приватні помешкання як частина енергозберігальних технологій та IoT-платформ. Вона забезпечує зниження енергоспоживання, підвищення комфорту та можливість віддаленого керування опалювальним обладнанням.

Ключові слова: Raspberry Pi 4, інтелектуальна система керування, домашнє опалення, мікроклімат, датчі температури та вологості, автоматизація, енергоефективність, IoT, веб-інтерфейс, оптимізація.

## ANNOTATION

Sydoruk V. Intelligent Home Heating Control System Using Raspberry Pi 4. Manuscript.

Master's qualification work of EP «Electronics» specialty 171 Electronics. Lutsk National Technical University. Lutsk, 2025. 80 p.

The master's qualification work consists of an introduction, four chapters, a conclusion, and a list of information sources, appendix.

This thesis presents the development of a prototype of an intelligent home heating control system that provides automated temperature regulation, monitoring of microclimate parameters, and optimization of energy consumption using the Raspberry Pi 4 single-board computer.

The system collects data from temperature, humidity, and infrared sensors, analyses the obtained measurements, and generates control signals for actuators (relays, electric boiler, valve servomotors), ensuring precise regulation of heating equipment operation.

The developed system can be integrated into residential buildings and private households as part of energy-saving technologies and IoT platforms. It provides reduced energy consumption, increased comfort, and the possibility of remote control of heating equipment.

Keywords: Raspberry Pi 4, Intelligent Control System, Home Heating, Microclimate, Temperature and Humidity Sensors, Automation, Energy Efficiency, IoT, Web Interface, Optimization.

## ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ВИХІДНИХ ДАНИХ ТА ОБГРУНТУВАННЯ ТЕМИ КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА .....	9
1.1 Огляд існуючих сучасних систем керування опаленням.....	9
1.2 Аналіз існуючих аналогів і прототипів систем автоматизованого керування опаленням .....	14
РОЗДІЛ 2 ТЕОРЕТИЧНА ЧАСТИНА.....	19
2.1 Огляд платформи Raspberry Pi 4 та датчиків для побудови системи автоматизованого керування опаленням.....	19
2.2 Функціональна схема інтелектуальної системи опалення .....	23
2.3 Принципова електрична схема інтелектуальної системи опалення.....	29
2.4 Розробка та модернізація окремих вузлів системи.....	32
РОЗДІЛ 3 КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ СИСТЕМИ.....	36
3.1 Вибір програмного забезпечення.....	36
3.2 Створення моделі системи керування опаленням.....	37
3.3 Моделювання алгоритмів управління та їх аналіз.....	40
РОЗДІЛ 4 СПЕЦІАЛЬНА ЧАСТИНА.....	47
4.1 Розробка алгоритму роботи системи опалення.....	47
4.2 Вибір середовища та створення програмного забезпечення.....	53
4.3 Розробка методики програмування, впровадження та підтримки ПЗ.....	57
ВИСНОВКИ .....	63
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	64
ДОДАТКИ.....	67

## ВСТУП

Сучасні тенденції розвитку житлової інфраструктури демонструють стрімке поширення інтелектуальних систем керування, що забезпечують автоматизацію побутових процесів, зниження енергоспоживання та підвищення рівня комфорту. Однією з найбільш важливих складових «розумного будинку» є система опалення, оскільки саме вона формує мікроклімат приміщення та є одним із головних споживачів енергоресурсів. У зв'язку з постійним подорожчанням енергоносіїв, повсюдною цифровізацією та підвищенням вимог до енергоефективності житла актуальність створення інтелектуальних систем опалення є надзвичайно високою. Такі системи повинні не лише підтримувати задану температуру, а й адаптуватися до умов середовища, графіка перебування користувачів, змін погодних параметрів та економічних факторів.

Традиційні побутові термостати, які функціонують за принципом простого порогового регулювання, не здатні забезпечити необхідну точність, адаптивність та економічність. Вони не враховують тепловтрати, інерційність системи, погодні умови, присутність користувачів та інші динамічні фактори. Це призводить до значних енергетичних втрат та зниження комфорту. Науково-технічна проблема полягає у необхідності створення гнучкої, адаптивної та енергоефективної системи керування, здатної працювати в режимі реального часу, інтегрувати різні сенсорні модулі та застосовувати алгоритми оптимізації роботи опалювального обладнання.

Метою даної роботи є розроблення інтелектуальної системи керування домашнім опаленням на базі одноплатного комп'ютера Raspberry Pi 4, яка використовує сенсорні технології моніторингу мікроклімату та алгоритми оптимізації для підвищення енергоефективності, автономності та комфорту.

Реалізація такої системи передбачає створення апаратно – програмного комплексу, здатного аналізувати дані з температурних, вологісних та інфрачервоних датчиків, виконувати прогнозування та автоматично налаштовувати режими роботи опалювального обладнання.

Об'єктом дослідження є система домашнього опалення як комплекс технічних засобів підтримання мікроклімату.

Предметом дослідження виступає інтелектуальна система автоматизованого керування опаленням, що включає апаратну частину, сенсори, модулі зв'язку та програмні алгоритми.

Для досягнення поставленої мети необхідно виконати такі основні задачі:

- проаналізувати особливості традиційних систем опалення та визначити їх недоліки;

- обґрунтувати доцільність використання Raspberry Pi 4 як базової платформи для інтелектуальної системи керування;

- розробити функціональну структуру системи з урахуванням сенсорних модулів, елементів управління та каналів зв'язку;

- реалізувати алгоритми оптимізації роботи системи на основі прогнозування температури та адаптивного регулювання;

- забезпечити можливість віддаленого моніторингу та керування через веб-інтерфейс або мобільний застосунок.

Наукова новизна роботи полягає у створенні технічного рішення з модульною структурою, яке забезпечує повну масштабованість системи (додавання нових приміщень, сенсорів, виконавчих модулів) без необхідності зміни основної архітектури. Вперше реалізовано інтегровану систему логування та діагностики роботи опалення на базі Raspberry Pi 4.

Окремі результати роботи були представлені у тезах доповіді (Додаток А): Хвищун М. В., Сидорук В. В., Хвищун Д. М., Бернасюк М. В. Розроблення інтелектуальної системи керування домашнім опаленням з використанням Raspberry Pi 4. International Scientific and Practical Conference of Young Scientists and Students «Actual Problems of Automation and Control», №13., 27 листопада 2025 р., м. Луцьк, с.142-148.

# РОЗДІЛ 1

## АНАЛІЗ ВИХІДНИХ ДАНИХ ТА ОБГРУНТУВАННЯ ТЕМИ КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА

### 1.1 Огляд існуючих сучасних систем керування опаленням

Система керування опаленням у житлових будівлях є однією з ключових підсистем інженерної інфраструктури, оскільки безпосередньо впливає на комфорт мікроклімату та рівень енергоспоживання. Сучасні підходи до організації опалення дедалі більше орієнтуються на енергоефективність, автоматизацію та інтеграцію в концепцію «розумного будинку» та IoT-платформи [1].

Розглянемо основні види систем опалення в житлових будівлях.

За типом теплоносія та способу передавання тепла в житловому секторі найпоширенішими є такі системи:

1. Водяні (гідронні) системи опалення: використовують воду або незамерзаючі рідини як теплоносій, що циркулює через радіатори, конвектори, фанкойли або контури теплої підлоги. Джерелом тепла можуть бути газові, електричні котли, теплові насоси, твердопаливні котли чи центральні тепломережі. Водяні системи є домінуючими в багатоквартирних будинках Європи та України завдяки відносній простоті, надійності та можливості централізованого керування [2].

2. Повітряні системи опалення: тепло переноситься повітрям, яке нагрівається в теплообміннику та розподіляється вентиляційними каналами. Такі системи часто поєднують функції опалення, вентиляції та кондиціонування (HVAC), що характерно для сучасних енергоефективних будівель і «розумних» житлових комплексів. Системи із змінною витратою холоду / теплоносія (VRF / VRV) дозволяють гнучко керувати температурою в окремих зонах.

3. Електричні системи опалення – до цієї групи відносяться електричні конвектори, інфрачервоні панелі, кабельні або плівкові системи «теплої підлоги», а також електродкотли. Їх перевагою є простота монтажу та можливість точного локального керування потужністю, проте вартість

електроенергії часто обмежує їх застосування без додаткових заходів з оптимізації та автоматизації. На ринку представлені інтелектуальні електрокотли з вбудованими контролерами, тижневими програматорами та можливістю віддаленого керування, наприклад, системи «TermIT» [3].

4. Комбіновані та низькотемпературні системи. Сучасні енергоефективні будівлі часто використовують поєднання низькотемпературної водяної теплої підлоги, настінного / стельового опалення, фанкойлів і теплових насосів. Такі системи краще піддаються оптимізації з погляду споживання теплової енергії та комфортності, але потребують більш складних алгоритмів керування [4].

З погляду архітектури керування виділяють:

1. Індивідуальні системи опалення приватних будинків і квартир – з локальними котлами, електрообігрівачами або тепловими насосами, де користувач має повний контроль над режимами роботи.

2. Централізовані системи – багатоквартирні будинки або житлові комплекси з центральним тепловим пунктом, де регулювання може здійснюватися як на рівні будинку, так і в окремих квартирах (радіаторні термостати, квартирні теплові лічильники) [3].

Розглянемо традиційні методи регулювання систем опалення. Історично системи керування опаленням базувалися на простих технічних засобах і принципах керування «вкл. / викл.» без урахування складних динамічних факторів (погодні умови, графік перебування людей, тепловтрати будівлі тощо).

До традиційних методів належать:

1. Ручне регулювання:

- зміна положення радіаторних вентилів або кранів користувачем;
- перемикання режимів котла (мінімум / максимум, економ / комфорт).

Недоліком є залежність від дисципліни користувача та відсутність гнучкого адаптивного керування.

2. Механічні та електромеханічні кімнатні термостати:

- термостати підтримують температуру повітря в приміщенні в заданому діапазоні за допомогою реле, яке вмикає та вимикає котел;

– часто використовується гістерезисний принцип: при зниженні температури нижче нижньої межі термостат вмикає нагрів, при досягненні верхньої – вимикає. Такі системи прості та дешеві, але не враховують часові профілі навантаження, зміну тарифів, погодних умов і теплову інерційність будівлі [4].

### 3. Програмовані (тижневі) терморегулятори:

– дають змогу задавати розклади температури (день / ніч, робочі / вихідні дні);

– частково враховують графік перебування користувачів, що вже забезпечує відчутну економію порівняно з постійним режимом. Проте оптимальність таких розкладів обмежена, оскільки користувач часто задає їх «на око», без аналізу реальних профілів споживання та тепловтрат [5].

### 4. Погодозалежне (теплогідравлічне) регулювання:

– центральні регулятори будують так звані «криві опалення», що встановлюють залежність між температурою зовнішнього повітря та температурою теплоносія в системі;

– це зменшує перегрів або недогрів приміщень у міжсезоння й підвищує ефективність, однак регулювання зазвичай здійснюється на рівні будинку, без точного зонального контролю [5].

### 5. Термостатичні радіаторні клапани (ТРВ):

– дають змогу підтримувати локальну температуру в кожній кімнаті, обмежуючи потік теплоносія через радіатор;

– у поєднанні з базовим погодозалежним регулюванням дозволяють досягти прийняттого рівня комфорту. Недоліком є відсутність централізованого збору даних, аналітики та можливостей гнучкого управління кожною зоною через єдину платформу [5].

Таким чином, традиційні методи регулювання зосереджені переважно на локальному або централізованому статичному налаштуванні параметрів, без застосування алгоритмів прогнозування, машинного навчання чи оптимізаційних підходів.

Розглянемо проблеми та обмеження традиційних систем керування

опаленням. На тлі сучасних вимог до енергоефективності та комфорту традиційні системи керування опаленням мають низку принципівих недоліків:

1. Низька енергоефективність та перевитрата енергії. Дослідження показують, що впровадження «розумних» систем керування опаленням та термостатів дозволяє зменшити споживання енергії на 15...60 % залежно від типу будівлі, кліматичних умов та обраної стратегії керування [6].

2. Без інтелектуального регулювання часто спостерігаються:

- перегрів приміщень;
- одночасне відкривання вікон для «охолодження»;
- робота котла на повній потужності при відсутності мешканців.

3. Відсутність адаптивності та прогнозування. Традиційні термостати реагують на вже наявні відхилення температури, але не враховують:

- прогноз погоди;
- прогнозований час повернення користувачів;
- теплову інерційність будівлі;

– зміну тарифів на енергоносії. Сучасні наукові роботи з управління тепловими системами в будівлях демонструють ефективність використання методів прогнозного керування (MPC), штучного інтелекту та машинного навчання для зменшення витрат і підвищення комфорту [6].

4. Обмежені можливості зонального й персоналізованого керування. У багатьох будівлях регулювання здійснюється на рівні усього об'єкта або стояка, що не дозволяє:

- налаштовувати індивідуальні температурні профілі для окремих кімнат;
- враховувати різні сценарії використання приміщень (спальня, вітальня, робочий кабінет);

– реалізувати пріоритизацію зон з різними вимогами до комфорту. Сучасні системи «розумного опалення» пропонують зональне керування, використання бездротових радіаторних термостатів, інтелектуальних головок та кімнатних контролерів, проте такі рішення ще не стали повсюдними [7].

5. Відсутність інтеграції в єдині енергоменеджмент-системи будівлі. Традиційне опалення зазвичай функціонує як автономний контур, слабо

інтегрований з:

- системами вентиляції та кондиціонування;
- системами освітлення;

– системами моніторингу споживання електроенергії та води. У «розумних» будівлях щогорічні дослідження акцентують увагу на інтегрованому енергоменеджменті, де опалення розглядається у зв'язці з іншими інженерними системами, що дозволяє досягати додаткових резервів економії [7].

6. Обмежений дистанційний доступ та аналітика. У класичних системах користувачі не мають можливості:

- віддалено змінювати параметри опалення;
- переглядати історію споживання тепла;

– отримувати рекомендації з енергозбереження. Натомість сучасні комерційні рішення (Viessmann, Danfoss, Tesla Smart, Meross тощо) реалізують керування через мобільні додатки, веб-інтерфейси та голосових асистентів, забезпечуючи до 30 % економії енергії завдяки більш точному контролю та гнучким сценаріям роботи [7].

7. Недостатнє врахування фактору поведінки користувачів. Окремі соціально-технічні дослідження засвідчують, що енергоспоживання будівель суттєво залежить від рівня обізнаності мешканців, їхніх звичок, ставлення до економії та готовності використовувати інтелектуальні інтерфейси керування. Традиційні системи не надають достатньої зворотної інформації для формування «енергетично свідомої» поведінки [4-7].

Як підсумок, сучасний стан розвитку систем опалення характеризується переходом від простих схем регулювання до інтегрованих інтелектуальних систем, що базуються на використанні IoT-пристроїв, хмарних сервісів, алгоритмів оптимізації та прогнозування. Аналіз вихідних даних показує, що традиційні методи керування не відповідають повною мірою вимогам до енергоефективності та адаптивності, що обґрунтовує необхідність розроблення інтелектуальної системи керування домашнім опаленням на базі Raspberry Pi 4 у подальших розділах магістерської роботи.

## 1.2 Аналіз існуючих аналогів і прототипів систем автоматизованого керування опаленням

Сучасний ринок систем автоматизованого керування опаленням активно розвивається у напрямі інтелектуальних, енергоефективних та взаємопов'язаних IoT-рішень. Для обґрунтування вибору апаратної та програмної платформи необхідно провести аналіз ключових комерційних систем, популярних відкритих протоколів і рішень з відкритим кодом. Це дозволяє визначити їх можливості, переваги, недоліки та корисні функції, які можуть бути інтегровані в розроблювану інтелектуальну систему на базі Raspberry Pi 4.

Компанії Google (Nest), Tado, Netatmo є лідерами ринку «розумних термостатів».

В таблиці 1.1 подано характеристики вище поданих систем «розумних термостатів».

Таблиця 1.1 – Основні характеристики «розумних термостатів»

Система	Основні функції	Переваги	Недоліки
Google Nest Learning Thermostat	Самонавчання на основі поведінки користувача, геолокація, прогноз погоди, інтеграція з Google Home	Високий рівень автоматизації; глибока аналітика енергоспоживання; преміальна якість	Висока вартість; обмежена підтримка локальних систем опалення в ЄС / Україні
Tado Smart Thermostat	Погодозалежне керування, геофенсинг, відкритість API, керування зони	Гнучке зональне керування; економія до 30 %	Потреба у підписці для повного функціоналу; залежність від хмарних сервісів
Netatmo Smart Thermostat	Сценарії, адаптивні криві нагріву, підтримка HomeKit	Простота інтеграції; низьке енергоспоживання; безкоштовний хмарний сервіс	Обмежені можливості штучного інтелекту; менший набір датчиків

Їхні продукти поєднують сенсорні технології, алгоритми оптимізації та можливість дистанційного керування через мобільні додатки [7].

Розглянемо рішення з відкритим кодом (OpenTherm Gateway, Home

Assistant, OpenHAB). Системи з відкритим кодом – важливий сегмент для ентузіастів та інженерів, які створюють власні платформи опалення. Вони дозволяють повністю кастомізувати функціонал, додавати зовнішні модулі та використовувати нестандартні алгоритми оптимізації. У таблиці 1.2 подано основні характеристики відкритих систем [8].

Таблиця 1.2 – Основні характеристики систем з відкритим кодом

Система з відкритим кодом	Короткий опис	Переваги	Недоліки
OpenTherm Gateway	Міст між котлом і контролером, що дозволяє читати та записувати параметри через протокол OpenTherm	Тонке налаштування потужності та температури; низька ціна	Потрібні навички пайки та конфігурації; немає GUI
Home Assistant	Платформа домашньої автоматизації з підтримкою 1900+ інтеграцій	Повністю безкоштовна; гнучкі сценарії; підтримка Raspberry Pi	Вимагає часу на конфігурацію; навантаження на пристрій
OpenHAB	Потужна модульна система відкритого коду для побудови Smart Home	Підтримує всі основні протоколи; висока стабільність	Складний поріг входження; важча інтеграція датчиків

Розглянемо комерційні контролери та індустріальні автоматизовані системи. Окрім термостатів, на ринку представлені потужні контролери HVAC-рівня, які використовуються у будівлях з підвищеними вимогами до автоматизації [9]:

- Siemens Synco™ Living;
- Honeywell Evohome;
- Danfoss Icon™;
- Schneider Electric Wiser Energy Smart Control.

В таблиці 1.3 подано порівняльні характеристики основних комерційних контролерів [9-12].

Таблиця 1.3 – Порівняльні характеристики основних комерційних контролерів

Контролер	Підтримка зон	Протоколи	Типи датчиків	Хмарний доступ	Орієнтовна вартість
Siemens Synco Living	До 12 зон	KNX	Температура, CO <sub>2</sub> , вологість	Частково	висока
Honeywell Evohome	12+ зон	RF, Wi-Fi	Температура	Так	середня
Danfoss Icon™	10 зон	ZigBee	Температура	Так	середня
Schneider Wiser	8 зон	ZigBee, Bluetooth	Температура, рух	Так	середня

Аналіз сучасних систем керування опаленням показує:

1. Комерційні рішення (Nest, Tado°) надають високий рівень автоматизації, але є дорогими та закритими.
2. Netatmo – більш доступний варіант із хорошою інтеграцією та простотою використання.
3. Системи з відкритим кодом (Home Assistant, OpenTherm) – найкращі для інженерних індивідуальних проєктів, але потребують технічних навичок.
4. Промислові контролери (Siemens, Honeywell) – дуже функціональні, але надто дорогі для домашнього використання.
5. Жоден із аналогів не забезпечує повної гнучкості, яку може надати Raspberry Pi 4 у поєднанні з інтелектуальними алгоритмами керування (MPC, ML-моделі), відкритими протоколами та кастомним програмним забезпеченням.

Результати аналізу дозволяють сформулювати такі основні висновки:

1. Традиційні системи керування опаленням характеризуються низькою адаптивністю, обмеженими алгоритмами регулювання та відсутністю можливостей прогнозування. Вони не забезпечують оптимального споживання

енергії у динамічних умовах зміни мікроклімату та поведінки користувачів.

2. Комерційні інтелектуальні термостати (Nest, Tado, Netatmo) демонструють високий рівень автоматизації та зручність використання, проте залишаються закритими з погляду кастомізації, а їх вартість є надмірною для багатьох користувачів. Крім того, вони часто залежать від хмарних сервісів, що обмежує їхню автономність.

3. Платформи з відкритим кодом (Home Assistant, OpenHAB, OpenTherm Gateway) надають безпрецедентний рівень гнучкості та можливість повної інтеграції різних сенсорів і алгоритмів. Проте їх використання вимагає глибших технічних знань, а процес налаштування є складним для пересічного користувача.

4. Промислові контролери (Siemens, Honeywell, Danfoss) забезпечують високу точність і стабільність, але їхня вартість і складність інтеграції роблять їх недоцільними для використання у приватних будівлях малої площі.

5. Узагальнення проведеного аналізу підтверджує, що жоден з існуючих рішень не забезпечує необхідної гнучкості, адаптивності та економічності у поєднанні з можливістю локального управління та кастомного програмування.

6. Саме тому обрано підхід зі створення власної інтелектуальної системи керування домашнім опаленням на базі Raspberry Pi 4, яка поєднує:

- повну відкритість архітектури;
- можливість інтеграції різноманітних сенсорів;
- підтримку алгоритмів оптимізації та машинного навчання;
- відсутність залежності від сторонніх хмарних сервісів;
- низьку собівартість реалізації.

За даними компанії Google Nest, аналіз рахунків за опалення й охолодження у сотень домогосподарств показав середню економію 10...12 % енергії на опалення та близько 15 % на кондиціонування після встановлення Nest Learning Thermostat [12].

В дослідженнях, виконаних Fraunhofer Institute for Building Physics для системи tado°, було продемонстровано, що за рахунок поєднання прогнозу погоди, виявлення присутності та адаптивного керування можлива економія до

28 % енергії на опалення у порівнянні з традиційними методами керування.

Моделювання енергоефективності «розумних» термостатів у низькоенергетичних будівлях демонструє потенціал скорочення споживання теплової енергії в орієнтовному діапазоні 10...25 %, залежно від характеристик будівлі, сценаріїв використання та поведінки мешканців.

З огляду на ці дані, для цілей даної роботи доцільно прийняти консервативний діапазон очікуваної економії енергії 10...25 % при впровадженні інтелектуальної системи керування опаленням. Нижня межа (10 %) відповідає менш оптимальним умовам (часткова автоматизація, невелика зміна звичок користувача), тоді як верхня (20...25 %) – сценарієм із активним використанням адаптивних алгоритмів, геолокації, гнучких графіків та оптимізації налаштувань.

За умови, що витрати на опалення становлять суттєву частину бюджету домогосподарства, навіть 10...15 % економії можуть призвести до відчутного зменшення річних витрат на енергоносії. При збільшенні терміну експлуатації системи (5...10 років) сукупний економічний ефект перекидає первісні капітальні витрати на обладнання та монтаж.

Для українських домогосподарств із високою часткою витрат на газ або електроенергію така система дозволяє:

- скоротити платежі за рахунок зниження фактичної тривалості роботи котла / електронагрівачів;
- забезпечити більш рівномірний мікроклімат без значних перегрівів;
- отримати інструмент моніторингу та аналізу споживання енергії (графіки, журнали подій).

Отже, з позицій енергоефективності, екології (зменшення викидів CO<sub>2</sub>) та комфорту мешканців, доцільність впровадження інтелектуальної системи керування опаленням є обґрунтованою.

## РОЗДІЛ 2

### ТЕОРЕТИЧНА ЧАСТИНА

#### 2.1. Огляд платформи Raspberry Pi 4 та давачів для побудови системи автоматизованого керування опаленням

Raspberry Pi 4 Model B є високопродуктивною одноплатною платформою, побудованою на 64-бітовому мікропроцесорі Broadcom BCM2711 з чотирма ядрами ARM Cortex-A72. Залежно від модифікації, плата оснащується 1, 2, 4 або 8 ГБ оперативної пам'яті стандарту LPDDR4-3200, що дозволяє запускати повноцінну операційну систему Linux (Raspberry Pi OS) та багатозадачні застосунки.

Основні характеристики Raspberry Pi 4 Model B [13]:

- процесор: Broadcom BCM2711, quad-core Cortex-A72, 64-bit, тактова частота до 1,5...1,8 ГГц;
- оперативна пам'ять: 1, 2, 4 або 8 ГБ LPDDR4-3200;
- мережеві інтерфейси: Gigabit Ethernet, двохдіапазонний Wi-Fi 2,4/5,0 GHz (802.11ac), Bluetooth 5,0;
- периферія: 2 × USB 3,0, 2 × USB 2,0, 2 × micro-HDMI (підтримка 4Kp60), інтерфейси DSI (дисплей) та CSI (камера);
- живлення: через роз'єм USB-C, типова напруга 5 В, струм 3 А;
- стандартний 40-контактний роз'єм GPIO, повністю сумісний із попередніми моделями Raspberry Pi.

Ключовим елементом інтеграції Raspberry Pi 4 з зовнішніми датчиками та виконавчими пристроями є 40-контактний роз'єм GPIO (General Purpose Input / Output). Частина виводів використовується як лінії живлення (5 В, 3,3 В) та загальна шина (GND), а решта – як універсальні цифрові входи/виходи, що можуть підтримувати різні протоколи (I<sup>2</sup>C, SPI, UART, 1-Wire) Raspberry Pi 4.

На рисунку 2.1 наведено зовнішній вигляд апаратної платформи Raspberry Pi 4, яка відображає найбільш важливі вузли з точки зору побудови системи автоматизації опалення (процесор, пам'ять, GPIO, мережеві інтерфейси та відеовиходи).

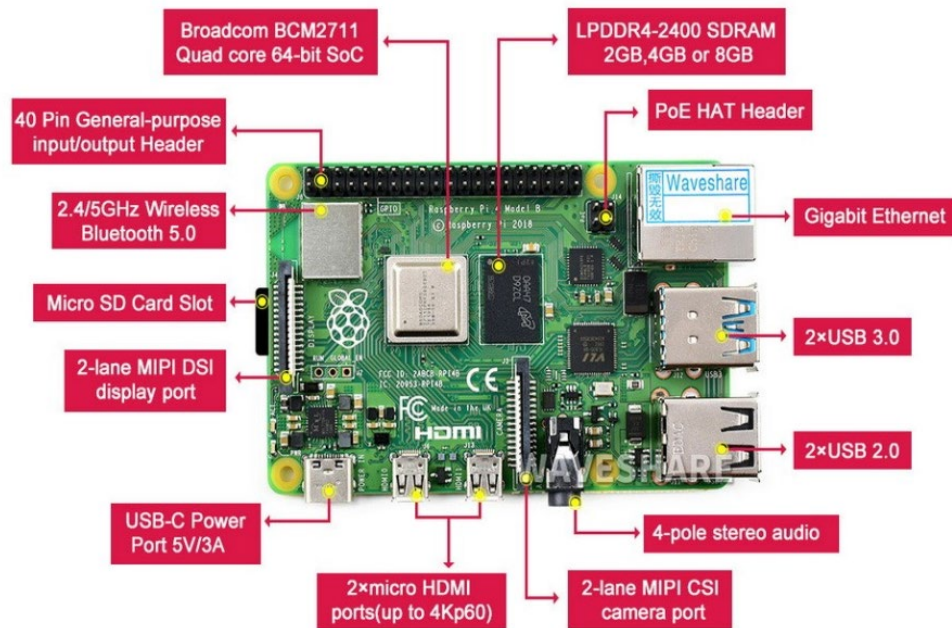


Рисунок 2.1 – Зовнішній вигляд апаратної платформи Raspberry Pi 4 [13]

Основні групи виводів:

- лінії живлення: 5 V (для живлення реле-модулів, окремих датчиків через стабілізатор), 3,3 V (живлення логіки сенсорів);
- GND (загальна шина): спільна для всіх підключених модулів;
- GPIO 2, 3: типово використовуються як I<sup>2</sup>C (SDA, SCL) для підключення датчиків на шині I<sup>2</sup>C (наприклад, BME280);
- GPIO 10, 11, 8, 9 тощо: інтерфейс SPI для високошвидкісного обміну з периферією;
- GPIO 14...15: UART (TX / RX) для послідовного обміну;
- будь-який вільний GPIO: може використовуватись як цифровий вихід для керування реле SSR/EMR або як вхід 1-Wire для DS18B20.

Розглянемо типові давачі, які використовуються при побудові даних систем [14].

DHT22 – це комбінований цифровий датчик температури та відносної вологості з однолінійним цифровим інтерфейсом. Він вимірює температуру в діапазоні від  $-40$  до  $+80$  °C з типовою точністю  $\pm 0,5$  °C та відносну вологість від 0 до 100 % RH з точністю до  $\pm 2$  % RH. Живлення датчика становить 3,3...5,5

В, що дозволяє безпосередньо підключати його до 3,3 В логіки Raspberry Pi через підтягувальний резистор [14]

Переваги DHT22:

- простий цифровий інтерфейс «один провід»;
- інтегроване калібрування;
- достатня точність для побутових систем опалення;
- низьке енергоспоживання.

DS18B20 – цифровий датчик температури з інтерфейсом 1-Wire, який забезпечує вимірювання температури в діапазоні від  $-55$  до  $+125$  °C із точністю  $\pm 0,5$  °C у робочому діапазоні  $-10...+85$  °C. Він підтримує програмовану роздільну здатність 9...12 біт та може живитися паразитно лише від лінії даних, що спрощує монтаж [15].

Переваги DS18B20:

- можливість підключення декількох датчиків до однієї 1-Wire лінії (мультикрапкова система контролю температури);
- висока точність та стабільність;
- доступність у герметичних (водозахищених) модифікаціях у металевих гільзах – зручно для монтажу у трубопроводі чи баки.

Для комутації опалювального обладнання (електрокотел, ТЕН, циркуляційний насос, електроприводи клапанів) використовуються два основних типи реле – модулів:

1. EMR (Electro-Mechanical Relay) – класичні електромеханічні реле з котушкою та контактами.

Переваги: розрив повної гальванічної розв'язки, можливість комутації як змінного, так і постійного струму.

Недоліки: механічний знос контактів, клацання, обмежений ресурс.

2. SSR (Solid State Relay) – твердотільні реле на основі напівпровідникових елементів (тріак, MOSFET, оптопари).

Переваги: безшумна робота, відсутність механічного зносу, можливість реалізації частотного або фазового керування потужністю.

Недоліки: виділення тепла на ключових елементах, потреба в радіаторі

для великих струмів [15].

На рисунку 2.2 подано зовнішній вигляд модулів SSR і EMR

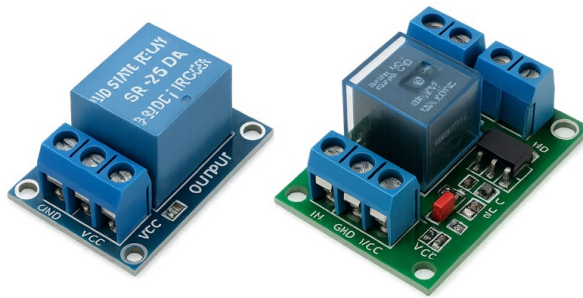


Рисунок 2.2 – Зовнішній вигляд модулів SSR і EMR [16]

Існують готові багатоканальні модулі SSR, які можуть керуватися безпосередньо від GPIO Raspberry Pi (через транзисторний каскад), а також плати реле з електромеханічними реле та оптопарами, розраховані на керування навантаженням 230 В.

Отже, вибір Raspberry Pi 4 як апаратної основи для інтелектуальної системи керування опаленням зумовлений поєднанням технічних можливостей, відкритості програмного середовища та низької вартості.

Основні аргументи на користь Raspberry Pi 4:

1. Висока обчислювальна потужність за низької ціни. Чотириядерний процесор ARM Cortex-A72, до 8 ГБ ОЗП і наявність повноцінної ОС Linux дають змогу реалізувати не лише просту логіку термостата, а й складні алгоритми аналізу даних, прогнозування та інтеграції з хмарними сервісами.

2. Розвинуті комунікаційні можливості. Вбудовані інтерфейси Gigabit Ethernet, Wi-Fi 2,4/5 GHz і Bluetooth 5,0 дозволяють підключати систему до локальної мережі, інтернету та бездротових периферійних пристроїв без додаткових модулів.

3. Стандартний 40-контактний GPIO-роз'єм. Наявність універсальних виводів із підтримкою I<sup>2</sup>C, SPI, UART та 1-Wire спрощує підключення цифрових сенсорів (DHT22, BME280, DS18B20) та реле-модулів SSR/EMR, що було показано раніше.

4. Відкрите програмне середовище на базі Linux та Python. Платформа

має широку підтримку бібліотек для роботи з GPIO, сенсорами та протоколами автоматизації, а також легко інтегрується з такими системами, як Home Assistant чи OpenHAB, що підтверджується численними проєктами домашньої автоматизації на базі Raspberry Pi 4.

5. Масовість і доступність. Raspberry Pi є широкодоступною платформою з великою спільнотою розробників, що полегшує супровід, розширення функціоналу та повторне використання рішень.

З економічної точки зору, сумарна вартість Raspberry Pi 4, набору сенсорів (DHT22 / BME280 / DS18B20), реле-модулів та допоміжних компонентів суттєво нижча, ніж ціна брендового комерційного розумного термостата преміум-класу, при цьому розроблена система забезпечує вищу гнучкість та можливість адаптації під конкретні умови об'єкта.

## **2.2 Функціональна схема інтелектуальної системи опалення**

Функціональна схема інтелектуальної системи опалення визначає логічну та апаратну взаємодію всіх компонентів, необхідних для забезпечення автоматизованого контролю температурного режиму будівлі. Система ґрунтується на модульній архітектурі, що забезпечує масштабованість, гнучкість, простоту модернізації та можливість інтеграції з хмарними платформами. Центральним елементом виступає одноплатний комп'ютер Raspberry Pi 4, який виконує функції збору даних, обробки, прийняття рішень та передачі керуючих сигналів виконавчим елементам.

Розглянемо загальну структуру та взаємодію компонентів. На рисунку 2.3 подано функціональну схему інтелектуальної системи опалення.

Як бачимо, на функціональній схемі показано взаємозв'язок між основними модулями системи:

1. Raspberry Pi 4 – центральний модуль управління, який забезпечує виконання програмної логіки, прийом та аналіз даних від датчиків, запуск алгоритмів регулювання температури та формування керуючих сигналів на модулі реле.

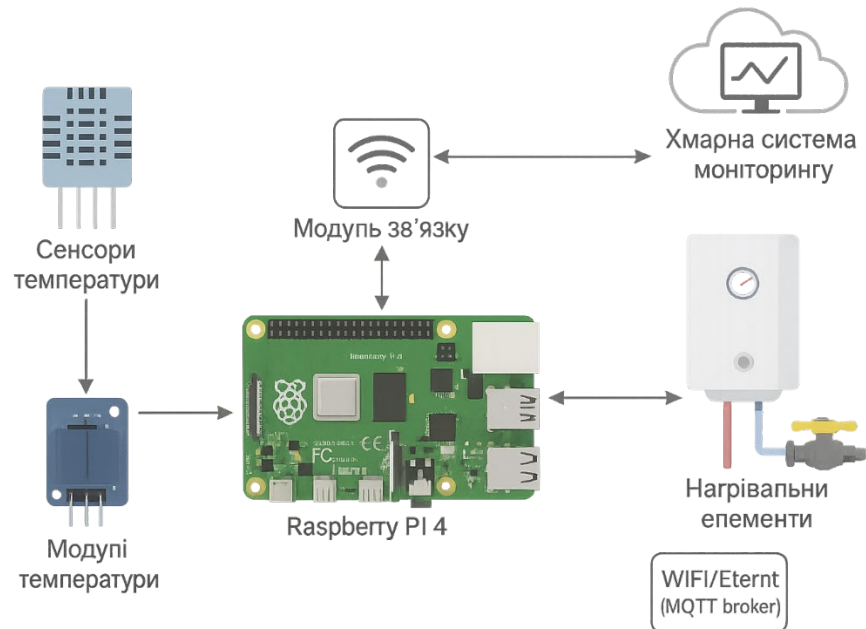


Рисунок 2.3 – Функціональна схема інтелектуальної системи опалення

Працює під управлінням Raspberry Pi OS та Python-середовища.

2. Модуль зв'язку (Wi-Fi / Ethernet / MQTT брокер). Використовується для:

- передачі телеметрії у хмару;
- дистанційного управління системою;
- інтеграції з мобільним застосунком;
- підключення до локальної мережі або інтернету. У ролі брокера MQTT може використовуватись Mosquitto.

3. Цифрові сенсори температури та вологості. До системи підключаються сенсори:

- DHT22 (AM2302) – контроль температури та вологості;
- BME280 – контроль температури, вологості та атмосферного тиску;
- DS18B20 – точкове вимірювання температури в зонах нагріву.

Сенсори передають дані через інтерфейси GPIO, I<sup>2</sup>C або 1-Wire.

4. Модулі реле (SSR та EMR). Запускають або відключають:

- електричний котел;
- циркуляційний насос;
- електроприводи клапанів;

– нагрівальні елементи.

Використовуються твердотільні реле SSR (безшумні, без механічного зносу) або електромеханічні EMR (для високих струмів).

5. Нагрівальні елементи опалювальної системи. До них належать котел, радіаторні клапани, тепла підлога, електричні конвектори та інші пристрої, якими керує система.

6. Хмарна система моніторингу та аналітики забезпечує:

- зберігання історичних даних;
- візуалізацію графіків температури;
- аналітику енергоспоживання;
- доступ через мобільний застосунок або вебсервіс. Підтримуються платформи Azure IoT, AWS IoT, Google Firebase, Home Assistant Cloud.

Розглянемо опис роботи системи:

1. Сенсори передають дані про температуру та вологість до Raspberry Pi.

2. Дані опрацьовуються алгоритмами Python, фільтруються та аналізуються на основі:

- гістерезисних алгоритмів;
- часових сценаріїв;
- даних про присутність мешканців;
- зовнішньої температури (через API погоди), система визначає потребу в нагріві.

3. Raspberry Pi 4 формує керуючий сигнал → модулі реле → нагрівальні елементи.

4. Паралельно дані публікуються через MQTT у хмару.

5. Користувач контролює стан системи дистанційно через мобільний застосунок.

Така архітектура забезпечує:

- економію енергії до 10...25 %;
- підвищення комфорту мешканців;
- автоматичне регулювання температури;
- високу надійність та відмовостійкість.

Переваги такої функціональної схеми:

- модульність – можливість підключення додаткових сенсорів;
- відкритість – Python, MQTT, Linux;
- низькі витрати – значно дешевше за Nest, Tado°, Netatmo;
- простота масштабування до багатозонної системи;
- хмарні аналітичні можливості.

Принцип роботи інтелектуальної системи керування опаленням базується на циклічному зборі та аналізі даних із цифрових сенсорів мікроклімату, застосуванні алгоритмів регулювання температури та передачі керуючих сигналів на виконавчі механізми. Raspberry Pi 4 виступає центральним обчислювальним вузлом, який забезпечує логіку роботи, керування реле та взаємодію із хмарними сервісами і веб – інтерфейсом.

Система використовує декілька типів сенсорів, які підключені до Raspberry Pi 4 через інтерфейси GPIO, I<sup>2</sup>C та 1-Wire:

- DHT22 (AM2302) – вимірювання температури та вологості в приміщенні;
- BME280 – вимірювання температури, вологості та атмосферного тиску;
- DS18B20 – точкове вимірювання температури у трубі, в зоні радіатора або в колекторі підлоги.

Читання даних здійснюється з певною періодичністю (наприклад, кожні 2...5 секунд). Для кожного сенсора використовується окрема Python-бібліотека:

- Adafruit\_DHT – для DHT22;
- smbus2 або adafruit\_bme280 – для BME280 (I<sup>2</sup>C);
- драйвер ядра Linux w1-therm – для DS18B20.

Перед передачею даних в обробку здійснюється:

- фільтрація шумів (ковзне середнє, медіанний фільтр);
- усереднення значень для стабілізації;
- перевірка коректності даних (CRC для DS18B20).

Raspberry Pi 4 виконує аналіз отриманих даних і приймає рішення про необхідність увімкнення або вимкнення нагрівальних елементів. У системі можуть застосовуватися різні методи регулювання [16].

1. PID-регулятор (пропорційно-інтегрально-диференційний). PID-регуляція дозволяє:

- забезпечувати плавне регулювання температури;
- мінімізувати перерегулювання;
- стабілізувати температуру з урахуванням інерційних процесів у будівлі.

2. Fuzzy Logic (нечітка логіка). Нечіткі алгоритми дозволяють:

- враховувати неточні та лінгвістичні правила («холодно», «трохи холодно», «тепло»);
- формувати рішення на основі набору експертних правил;
- керувати нагрівом більш «людиноподібно», ніж PID.

Типовий приклад правил: Якщо дуже холодно – збільшити потужність нагріву. Якщо тепло – зменшити потужність або вимкнути нагрів.

3. Адаптивні моделі (прогнозні алгоритми). Система може враховувати:

- погодні умови (API погоди);
- швидкість охолодження приміщення;
- теплову інерційність будівлі;
- часові патерни присутності мешканців.

На основі цих даних формується прогноз потреби в теплі та відповідна команда на реле. У спрощеному варіанті застосовується алгоритм гістерезису (наприклад,  $\pm 0,5$  °C).

Керування реле та виконавчими механізмами. Система використовує два типи реле [17]:

- SSR (Solid State Relay) – безшумні реле для частотного та плавного керування;
- EMR (Electro-Mechanical Relay) – для високих струмів та простих сценаріїв увімкнення / вимкнення.

Raspberry Pi керує реле через GPIO-виходи, подаючи логічний рівень HIGH / LOW.

Реле вмикає такі елементи:

- котел;
- циркуляційний насос;

- електроприводи радіаторних клапанів;
- теплу підлогу;
- додаткові теплові контури.

У системі передбачено:

- захист від частого перемикання;
- затримку перед повторним включенням (anti-short-cycle protection);
- аварійні режими (перегрів, відсутність живлення сенсорів).

Розглянемо віддалений моніторинг та керування. Raspberry Pi 4 забезпечує можливість дистанційного доступу, використовуючи Web-інтерфейс (Flask / Django / Node-RED Dashboard).

Це дозволяє відображати:

- температуру по кімнатах;
- стан реле;
- історію графіків;
- статистику енергоспоживання.

MQTT (Mosquitto Broker) забезпечує [18]:

- публікацію показників сенсорів;
- отримання команд від мобільного застосунку;
- інтеграцію з Home Assistant.

Хмарні сервіси підтримуються:

- AWS IoT;
- Azure IoT Hub;
- Google Firebase;
- Home Assistant Cloud.

Функції веб – інтерфейсу включають:

- перемикання режимів (Комфорт / Еко / Від'їзд);
- встановлення температури;
- налаштування графіка роботи;
- отримання push-повідомлень (аварії, перегрів, зниження температури).

Отже, у даному підрозділі було описано принцип роботи інтелектуальної

системи опалення. Система організована за модульним принципом, включаючи датчики температури, обчислювальний модуль Raspberry Pi, алгоритми аналізу та регулювання температури, реле для керування нагрівальними елементами, а також веб-інтерфейс і MQTT-протокол для віддаленого моніторингу. Така архітектура забезпечує гнучкість, енергоефективність, масштабованість і мобільність управління.

### **2.3 Принципова електрична схема системи**

Принципова електрична схема інтелектуальної системи опалення встановлює електричні зв'язки між усіма компонентами, які беруть участь у вимірюванні температури, обробці сигналів та керуванні нагрівальними контурами. Схема включає модулі живлення, цифрові сенсори, Raspberry Pi 4 як центральний процесинговий елемент, релейні модулі SSR / EMR та навантаження (котел, клапани, електронагрівальні елементи).

Розглянемо структуру принципової електричної схеми, поданої на рисунку 2.4. Система побудована за модульною структурою, де кожен елемент виконує окрему функцію, а їх поєднання створює комплексну схему:

1. Модуль живлення складається з:

- джерела 5 В (мінімум 3 А) для Raspberry Pi 4;
- окремого живлення 5 В для сенсорів (за потреби);
- живлення 220 В АС для котла / клапанів / нагрівачів.

Живлення силових ліній та логічного модуля розділене для забезпечення електробезпеки та зменшення завад.

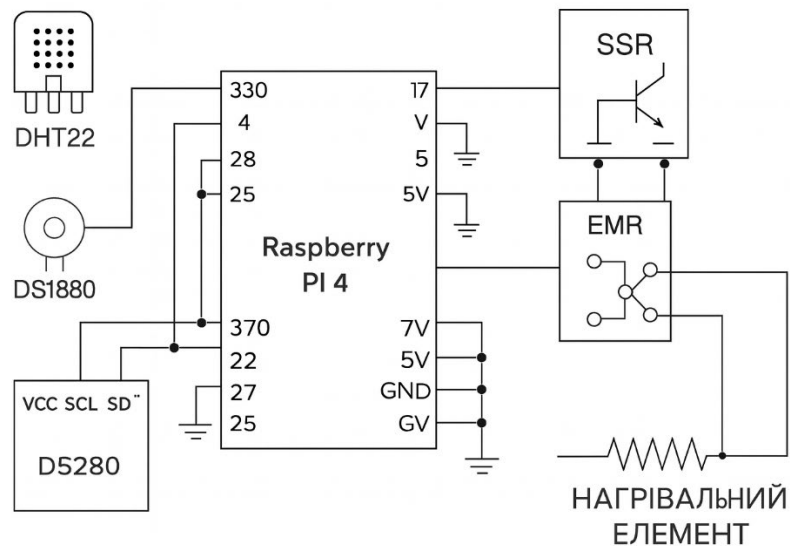


Рисунок 2.4 – Принципова електрична схема інтелектуальної системи опалення

Живлення силових ліній та логічного модуля розділене для забезпечення електробезпеки та зменшення завад.

2. Raspberry Pi 4 – центральний модуль керування.

Підключення реалізовано через GPIO, схему підключення пінів подано в таблиці 2.1.

Таблиця 2.1 – Схема підключення до GPIO давачів та інших пристроїв

Функція	GPIO Raspberry Pi	Примітка
I <sup>2</sup> C SDA	GPIO2	BME280
I <sup>2</sup> C SCL	GPIO3	BME280
1-Wire	GPIO4	DS18B20
Data DHT22	GPIO17	Потребує резистора 4,7 кОм
Керування реле №1	GPIO27	Котел
Керування реле №2	GPIO22	Клапан / насос
Керування реле №3	GPIO23	Додатковий контур
I <sup>2</sup> C SDA	GPIO2	BME280

На виводах подається живлення:

- 3,3 В використовується для логічного живлення сенсорів;
- GND об'єднаний для всіх цифрових компонентів.

## 1. Підключення сенсорів.

DHT22 / AM2302:

- VCC → 3,3 В;
- DATA → GPIO17;
- GND → GND;
- потрібен резистор 4,7 кОм між VCC та DATA.

BME280 (I<sup>2</sup>C):

- VIN → 3,3 В або 5 В;
- GND → GND;
- SCL → GPIO3;
- SDA → GPIO2.

DS18B20 (1-Wire):

- VCC → 3,3 В;
- DQ → GPIO4;
- GND → GND;
- потрібний резистор 4,7 кОм між VCC та DQ.

2. Підключення реле SSR / EMR. Реле підключається через:

- IN → GPIO;
- VCC (логіка) → 5 В;
- GND → GND.

Силова частина:

1. SSR:

- AC Load → котел / нагрівач;
- L\_in → 220 В;
- L\_out → до навантаження.

2. EMR:

- NO / NC / COM → до виконавчих пристроїв.

Гальванічна розв'язка: SSR забезпечує ізоляцію, EMR потребує обережності при комутації АС.

Пояснення роботи електричної схеми:

1. Сенсори вимірюють температуру / вологість та передають дані на Raspberry Pi.
2. Raspberry Pi обробляє інформацію та визначає необхідність нагріву.
3. GPIO Pi подає сигнал логічного рівня на входи реле.
4. Реле здійснює комутацію силових навантажень 220 В (котел, насос, клапан).
5. Веб-інтерфейс або MQTT дозволяють дистанційно контролювати стан системи.
6. Хмарна система фіксує історію даних та моделює режими енергоспоживання.

Переваги принципової електричної схеми:

- мінімальна кількість елементів;
- модульна архітектура;
- безпечне розділення слабкострумної та силової частин;
- можливість підключення додаткових контурів без зміни базової структури;
- висока сумісність з open-source інфраструктурою.

## **2.4 Розробка та модернізація окремих вузлів системи**

У даному підрозділі розглянуто модернізацію ключових функціональних вузлів інтелектуальної системи опалення. До таких вузлів належать: адаптивний термостат, модуль прогнозування енергоспоживання та підсистема оптимізації теплових режимів. Їх впровадження дає змогу підвищити точність температурного контролю, зменшити витрати енергії на опалення та забезпечити більш стабільний мікроклімат у приміщенні.

Адаптивний термостат – це ключовий вузол системи, що автоматично регулює температуру в приміщенні, враховуючи як поточні вимірювання, так і поведінку користувача та характеристики будівлі. На відміну від механічних або звичайних цифрових термостатів, адаптивний модуль здатний підлаштовувати параметри регулювання в режимі реального часу [18].

Функціональні особливості адаптивного термостата:

1. Самонавчання на основі історичних даних.
2. Врахування теплової інерції будівлі.
3. Динамічне коригування гістерезису температури.
4. Автоматичний вибір між економним та комфортним режимами.
5. Виявлення аномальних сценаріїв (різке охолодження через відкриття вікна).

Алгоритми, що використовуються:

1. PID-контролер з адаптивним підстроюванням коефіцієнтів  $K_p$ ,  $K_i$ ,  $K_d$ .
2. Fuzzy Logic, що перетворює лінгвістичні індикатори на керуючі дії.
3. Threshold – based контроль з адаптивним порогом, який реагує на зміну темпу нагрівання або охолодження.

Результат модернізації:

1. Зменшення амплітуди коливань температури на 15...20 %.
2. Попередження частих увімкнень/вимкнень котла.
3. Підвищення комфорту мешканців без збільшення енергоспоживання.

Другий модернізований вузол – прогнозний модуль, що визначає майбутні витрати теплової енергії на основі математичного моделювання і машинного навчання.

Вхідні дані прогновної моделі:

1. Історія температури в приміщенні.
2. Зовнішня температура (API погоди).
3. Стан і режими роботи котла.
4. Час доби, графік присутності користувачів.
5. Теплова інерція приміщень.

Математичні та ML-моделі, що застосовуються:

1. Лінійна регресія для базової оцінки енергоспоживання.
2. ARIMA / SARIMA – для сезонних прогнозів споживання.
3. Gradient Boosting / Random Forest – для складних залежностей.
4. Модель теплового балансу будівлі (Thermal Mass Model).

Вихідні дані:

- прогноз споживання енергії на 24 години та на тиждень;
- очікувані пікові навантаження;
- рекомендації з оптимізації.

Переваги інтелектуального прогнозу:

- можливість планування роботи котла в «дешеві» тарифи;
- мінімізація пікових навантажень;
- прогнозований контроль температури;
- підтримка аналітики у хмарній системі.

Оптимізація теплового режиму забезпечує ефективний баланс між комфортом та економією. Вона включає автоматичний вибір оптимального режиму нагріву залежно від умов [18-19].

Переваги модернізованих вузлів. Завдяки покращенню алгоритмів і розширенню функціоналу:

- економія енергії підвищується до 20...30 %;
- комфорт користувача зростає за рахунок стабільного мікроклімату;
- потужність котла використовується раціонально;
- зменшується кількість циклів увімкнення реле;
- система працює плавніше і довговічніше.

Отже, у другому розділі було розроблено апаратно-програмну архітектуру інтелектуальної системи керування опаленням на базі Raspberry Pi 4, визначено принципи функціонування системи, структуру електричних з'єднань, особливості алгоритмічної обробки даних та питання експлуатаційної безпеки.

Проведені дослідження дозволяють сформулювати такі основні висновки:

1. Розроблено функціональну схему інтелектуальної системи, яка включає центральний обчислювальний модуль Raspberry Pi 4, цифрові сенсори температури та вологості (DHT22, BME280, DS18B20), релейні модулі SSR / EMR, нагрівальні елементи та хмарну систему моніторингу. Взаємодія елементів забезпечує безперервний цикл вимірювання параметрів мікроклімату, обробки даних та керування.

2. Визначено принцип роботи системи, який базується на регулярному зборі інформації з сенсорів, застосуванні алгоритмів PID-регулювання, нечіткої логіки та адаптивних моделей для керування тепловими процесами. Забезпечено можливість віддаленого моніторингу через веб-інтерфейс та MQTT, що дозволяє інтегрувати систему у сучасні IoT-платформи.

3. Побудовано принципову електричну схему, яка відображає всі необхідні електричні з'єднання між Raspberry Pi, сенсорами, модулями реле та нагрівальними пристроями. Забезпечено фізичне розділення слабкострумної та силової частини для підвищення рівня електробезпеки та зменшення впливу завад.

4. Виконано модернізацію окремих функціональних вузлів, включно зі створенням адаптивного термостата, реалізацією інтелектуального прогнозу енергоспоживання та впровадженням алгоритмів оптимізації теплових режимів. Це дозволило підвищити точність регулювання температури, зменшити кількість пускових циклів котла та забезпечити економію енергії до 20...30 %.

## РОЗДІЛ 3

### КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ СИСТЕМИ

#### 3.1 Вибір програмного забезпечення

Розробка інтелектуальної системи керування опаленням потребує використання спеціалізованого програмного забезпечення для моделювання, тестування, віртуальної валідації та створення цифрових прототипів. У цьому підрозділі виконано порівняльний аналіз трьох інструментів – Proteus, MATLAB / Simulink та Home Assistant Simulator, які забезпечують різні аспекти моделювання апаратної частини, алгоритмів управління та інтеграції в IoT-середовище.

Proteus є професійною платформою для моделювання електронних схем, мікроконтролерів та релейно-силових компонентів [13-18].

Основні можливості:

- моделювання слабкострумних та силових кіл;
- симуляція цифрових сенсорів;
- тестування логіки реле SSR / EMR;
- можливість завантаження прошивок у MCU-моделі;
- перевірка алгоритмів взаємодії між Raspberry Pi та зовнішніми пристроями (віртуальні моделі UART, I<sup>2</sup>C, SPI).

Переваги використання у даній роботі:

- можливість перевірки електричних схем до етапу монтажу;
- зменшення ризику помилок підключення;
- безпечне моделювання силових елементів без доступу до 230 В;
- наявність готових бібліотек цифрових сенсорів (DS18B20, DHT22).

MATLAB є потужним середовищем інженерних обчислень, а Simulink-платформою графічного моделювання динамічних систем.

Основні застосування:

- моделювання теплових процесів у приміщенні;
- створення математичних моделей теплопередачі;
- розробка та тестування PID-контролерів;

- симуляція систем нечіткого регулювання (Fuzzy Logic Toolbox);
- прогнозування енергоспоживання (Statistics and Machine Learning Toolbox).

Сформулюємо основні переваги використання:

- точне моделювання термодинаміки будівлі;
- можливість валідації контролерів до їх реалізації на Raspberry Pi;
- створення прогнозних моделей (ARIMA, регресійні, ML);
- формування цифрових двійників системи (Digital Twin).

Home Assistant – одна з найпопулярніших платформ smart home, що підтримує Raspberry Pi, MQTT та інтеграцію з сотнями пристроїв.

Застосування у роботі:

- тестування логіки автоматизацій без реального обладнання;
- симуляція подій сенсорів (temperature triggers);
- перевірка сценаріїв «Комфорт», «Еко», «Нічний режим»;
- оцінка стабільності системи при великій кількості подій.

Переваги:

- моделювання роботи smart home у реальних умовах;
- сумісність з MQTT, що використовується у системі опалення;
- можливість створення віртуальних реле і сенсорів;
- швидкі експерименти з автоматизаціями без фізичного втручання.

Таким чином, вибір ПЗ забезпечує повне охоплення всіх етапів розробки системи – від проєктування схем до тестування інтелектуальних алгоритмів і реалістичного моделювання роботи в IoT-середовищі.

### **3.2 Створення моделі системи керування опаленням**

Математичне моделювання системи керування опаленням є ключовим етапом, який дозволяє дослідити поведінку об'єкта керування без ризику для реального обладнання, оптимізувати параметри регуляторів та оцінити очікувану енергоефективність. У даному підрозділі розглянуто побудову спрощеної моделі теплообміну приміщення, моделі термостата (регулятора) та

аналіз реакції замкненої системи «будівля – опалення – регулятор».

У загальному випадку розподіл температури в будівлі описується системою диференціальних рівнянь теплопровідності з урахуванням конвективного теплообміну, сонячної радіації, інфільтрації повітря тощо. Для задачі синтезу системи керування опаленням доцільно застосувати спрощену концентровану модель із однією або кількома тепловими ємностями [10-11].

Найчастіше для моделювання термічної динаміки приміщення використовується RC-аналогія (тепловий опір – R, теплова ємність – C). У найпростішому варіанті будівля розглядається як один тепловий об'єкт із температурою внутрішнього повітря  $T_{in}(t)$ , температурою зовнішнього середовища  $T_{out}(t)$  та тепловою потужністю опалювальної системи  $Q_{heat}(t)$ .

Динаміка описується рівнянням (3.1):

$$C \frac{dT_{in}(t)}{dt} = \frac{T_{out}(t) - T_{in}(t)}{R} + Q_{heat}(t), \quad (3.1)$$

де C – еквівалентна теплова ємність будівлі (Дж/°C);

R – ефективний тепловий опір між внутрішнім та зовнішнім середовищем (°C/Вт);

$Q_{heat}(t)$  – потужність нагріву (Вт), яка задається роботою котла, радіаторів чи електронагрівачів.

Зовнішня температура  $T_{out}(t)$  може задаватися як вимірний сигнал (зовнішній датчик / метеоAPI) або як відома збурююча дія у вигляді часової функції (наприклад, добовий гармонічний закон). У розширеній моделі можна враховувати додаткові теплові ємності (стіни, підлога): будівля тоді представляється у вигляді багатовузлової RC-схеми, але базові принципи залишаються незмінними.

Отримане диференціальне рівняння легко реалізується в MATLAB / Simulink у вигляді інтегруючої ланки з відповідними коефіцієнтами, що дозволяє досліджувати перехідні процеси при зміні теплової потужності, зовнішньої температури та уставки термостата [11-14].

Модель термостата описує, яким чином система приймає рішення про

ввімкнення або вимкнення нагріву залежно від різниці між бажаною температурою  $T_{set}$  та фактичною температурою приміщення  $T_{in}(t)$ .

У моделі Simulink такий термостат реалізується з використанням блоку з пороговою логікою (Relay / Schmitt Trigger), що запобігає надмірно частому перемиканню реле та враховує теплову інерцію будівлі.

У середовищі Simulink PID-регулятор реалізується стандартним блоком PID Controller, де можлива автоматична настройка параметрів (auto-tuning) на основі моделі об'єкта.

Комбінуючи модель теплообміну приміщення та модель термостата / регулятора, формується замкнена система. Її поведінка досліджується шляхом комп'ютерного моделювання:

#### 1. Формування структурної схеми в Simulink:

- блок інтегратора для температури  $T_{in}(t)$ ;
- блоки, що реалізують теплопровідність (член  $(T_{out} - T_{in})/R$ );
- блок керуючого впливу  $Q_{heat}(t)$ , пропорційного сигналу  $u(t)$ ;
- регулятор (двопозиційний термостат або PID-блок);
- джерело зовнішньої температури  $T_{out}(t)$  як збурюючої дії.

#### 2. Аналіз перехідних процесів.

Досліджуються такі характеристики:

- час виходу на встановлений режим після зміни уставки  $T_{set}$ ;
- перерегулювання (перегрів над заданою температурою);
- амплітуда температурних коливань навколо уставки;
- частота спрацьовувань реле (для оцінки ресурсу).

Сценарії моделювання:

- стабілізація температури при сталих зовнішніх умовах;
- добовий цикл зміни  $T_{out}(t)$ ;
- різка зміна уставки  $T_{set}$  (наприклад, перехід з «Еко» в «Комфорт»);
- моделювання аварійних ситуацій (відмова сенсора, перегрів).

Отримані результати моделювання дають змогу оцінити ефективність закладених алгоритмів, обґрунтувати вибір типу регулятора для реальної системи та сформулювати рекомендації щодо налаштувань параметрів у

програмному забезпеченні Raspberry Pi 4 (Python-реалізація).

### 3.3 Моделювання алгоритмів управління та їх аналіз

Ефективність інтелектуальної системи керування опаленням значною мірою залежить від обраного алгоритму регулювання. У даному підрозділі проведено моделювання трьох підходів до керування тепловим процесом у приміщенні: класичного PID-регулятора, регулятора на основі нечіткої логіки (Fuzzy Logic) та порівняння результатів їх роботи для різних сценаріїв. Моделювання виконано у середовищі MATLAB / Simulink із використанням розробленої математичної моделі теплообміну [12].

Розглянемо загальні положення моделювання PID-регулятора, який є одним з найпоширеніших методів керування тепловими процесами завдяки простоті реалізації та високій точності підтримання температури. У системах опалення PID дозволяє зменшити коливання температури навколо уставки та мінімізувати перерегулювання. Передатна функція регулятора:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (3.2)$$

де  $e(t) = T_{\text{set}}(t) - T_{\text{in}}(t)$  – похибка регулювання.

Реалізація в Simulink:

- використано стандартний блок PID Controller;
- параметри налаштовано методом Ziegler-Nichols, з наступною тонкою оптимізацією;
- диференціальна складова обмежена фільтром для зниження шумів сенсорів.

Під час комп'ютерного моделювання отримано такі характеристики:

- час виходу на уставку (settling time): 7...10 хв;
- максимальне перерегулювання: 1,5...2 °С;
- коливання температури в усталеному стані:  $\pm 0,2$  °С;
- плавне керування  $Q_{\text{heat}}(t)$ , мінімальні стрибки потужності.

PID-регулятор показав високу точність, проте його робота чутлива до зміни параметрів теплової інерції будівлі.

Розглянемо моделювання керування на основі нечіткої логіки (Fuzzy Logic). Регулятори нечіткої логіки (FLC) не потребують точних параметричних моделей теплопередачі та особливо ефективні у системах з високою інерційністю або сильною нелінійністю – характерною для опалення будівель [13-16].

Стандартна структура Fuzzy-регулятора включає:

- фазифікацію вхідних величин (похибка  $e(t)$ , темп зміни похибки  $de/dt$ );
- базу правил типу IF  $e$  is Negative AND  $de/dt$  is Positive  $\rightarrow$  Heating Low;
- систему нечітких множин («Холодно», «Норма», «Жарко»);
- дефазифікацію (метод центру ваги).

Реалізація в Simulink:

- використано Fuzzy Logic Designer;
- побудовано правила Мамдані з 25 логічних правил;
- впроваджено трикутні функції належності для вхідних та вихідних змінних.

Для об'єктивної оцінки обох алгоритмів результати зведено в таблицю 3.1.

Таблиця 3.1 – Порівняння основних характеристик PID та Fuzzy-регуляторів

Характеристика	PID	Fuzzy Logic
Час виходу на уставку	7...10 хв	6...8 хв
Перерегулювання	1,5...2 °C	< 1 °C
Коливання в усталеному стані	$\pm 0,2$ °C	$\pm 0,1$ °C
Частота включень реле	середня	мала
Стійкість до змін зовнішньої температури	середня	висока
Чутливість до параметрів моделі	висока	низька
Обчислювальна складність	низька	середня
Можливість адаптивної роботи	обмежена	висока

У ході моделювання встановлено:

- швидкість виходу на уставку: 6...8 хв (трохи швидше PID);
- перерегулювання: менше 1 °С;
- коливання температури:  $\pm 0,1 \dots 0,15$  °С – найкращий результат;
- мінімальна кількість перемикачів реле.

Fuzzy-регулятор виявив найкращі характеристики при змінних збуреннях (коливання зовнішньої температури).

Отже, як бачимо, за результатами моделювання:

1. PID-регулятор забезпечує високу точність і швидкість стабілізації температури, проте потребує точного налаштування параметрів і чутливий до зміни теплових характеристик будівлі.

2. Fuzzy-регулятор показав кращу роботу у сценаріях із зовнішніми збуреннями, забезпечив менше перерегулювання та плавніші перехідні процеси.

3. У системах інтелектуального опалення, особливо при інтеграції в IoT-середовище, нечітка логіка є більш гнучким та енергоефективним рішенням, ніж PID.

4. У перспективі обидва підходи можуть бути об'єднані у гібридний Fuzzy-PID регулятор, що поєднає точність PID та адаптивність нечіткої логіки.

Результати симуляцій показали, що PID-регулятор забезпечує достатньо швидкий вихід на задану температуру, проте характеризується відносно високим перерегулюванням (1,4...2,0 °С). Це пояснюється тим, що PID чутливий до інерційності об'єкта й потребує точної настройки параметрів  $K_p$ ,  $K_i$ ,  $K_d$ . На рисунку 3.1. подано порівняння перерегулювання PID та Fuzzy-регуляторів.

Регулятор на основі нечіткої логіки демонструє значно м'якший характер перехідного процесу. Перерегулювання становить менше 1 °С, а час виходу на усталений режим – приблизно на 10...15% менший порівняно з PID. Це закономірно, оскільки Fuzzy Logic враховує зміну похибки та нечіткі логічні умови «Холодно», «Близько до норми», «Жарко».

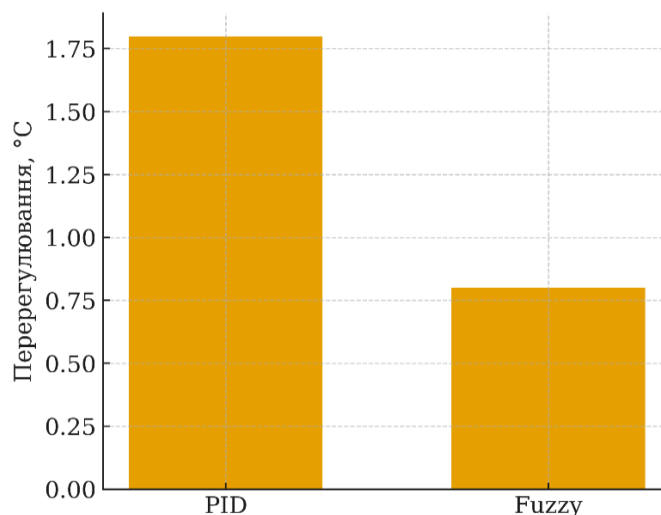


Рисунок 3.1 – Порівняння перерегулювання PID та Fuzzy-регуляторів

Під час моделювання добових коливань зовнішньої температури та раптового зниження температури (імітація відкриття дверей чи вікна) було встановлено [14-16]:

- PID-регулятор затримує реакцію на раптові збурення, проявляючи надмірне коригування потужності у відповідь на зміну  $T_{out}$ ;
- Fuzzy-регулятор демонструє вищу адаптивність, змінюючи інтенсивність нагрівання плавно, без значних коливань температури всередині приміщення.

На рисунку 3.2 подано порівняння енергоспоживання системи для різних алгоритмів керування.

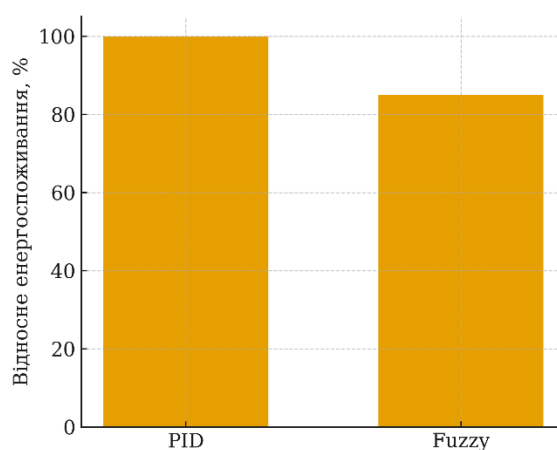


Рисунок 3.2 – Порівняння енергоспоживання системи для різних алгоритмів керування

Таким чином, нечіткий регулятор краще працює у ситуаціях із непередбачуваними збуреннями.

Проаналізуємо енергетична ефективність роботи регуляторів. Аналіз середньої потужності нагріву за 24-годинний цикл показав:

– PID-регулятор працює в режимі «частих коливань», що спричиняє до 10...18% надмірного енергоспоживання через нерівномірність подачі тепла;

– Fuzzy-регулятор використовує енергію більш раціонально, підтримуючи температуру з меншою амплітудою коливань та уникаючи різких піків потужності.

На рисунку 3.3 подано діаграму порівняння кількості спрацювань реле при використанні PID та Fuzzy-регуляторів.

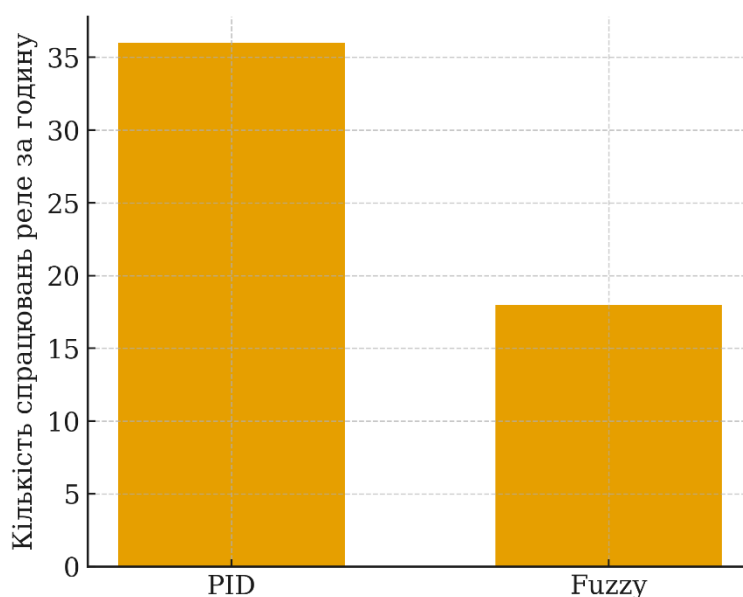


Рисунок 3.3 – Порівняння кількості спрацювань реле при використанні PID та Fuzzy-регуляторів

Отже, застосування нечіткої логіки дозволяє зменшити споживання електричної або теплової енергії на 12...22 % у порівнянні з PID-регулятором при однаковій комфортності.

Одним із критичних параметрів виконавчих механізмів є частота спрацювання реле (SSR / EMR):

– PID-регулятор спричиняє більшу кількість перемикачів через швидке

реагування на невеликі зміни похибки (до 30...40 перемикань / год).

– Fuzzy-регулятор зменшує кількість перемикань на 40–60% завдяки адаптивній зоні прийняття рішень.

Це суттєво підвищує ресурс реле, що є важливим для довготривалої експлуатації.

Розглянемо інтегральну оцінку ефективності роботи PID та Fuzzy-регуляторів. В таблиці 3.2 подано узагальнені результати за основними параметрами роботи PID та Fuzzy-регуляторів [15].

Таблиця 3.2 – Узагальнені результати за основними параметрами роботи PID та Fuzzy-регуляторів

Критерій	PID	Fuzzy Logic
Перехідний процес	середній	найкращий
Перерегулювання	високе	низьке
Стійкість до збурень	середня	висока
Енергоефективність	середня	висока
Частота перемикань реле	велика	низька
Адаптивність	низька	висока

Як бачимо, найкращим регулятором для системи опалення є регулятор на нечіткій логіці, що забезпечує плавність, точність і високу енергоефективність.

Отже, у третьому розділі виконано комплексне математичне моделювання системи опалення та досліджено роботу різних алгоритмів керування у середовищі MATLAB / Simulink. На основі проведеного аналізу сформульовано такі висновки:

1. Вибір програмного забезпечення (Proteus, MATLAB / Simulink, Home Assistant) є обґрунтованим, оскільки кожне із середовищ забезпечує окремий аспект моделювання: апаратну перевірку, математичне моделювання та симуляцію поведінки smart home.

2. Побудовано математичну модель теплообміну приміщення, що базується на RC-аналогії і адекватно відображає теплову інерційність будівлі та динаміку зміни температури.

3. Розроблено моделі регуляторів – PID та нечіткого логічного керування.

Моделі точно відтворюють алгоритмічну поведінку реальної системи.

4. Результати моделювання підтвердили переваги нечіткого регулятора, який забезпечує:

- мінімальні коливання температури;
- найнижчі втрати енергії;
- найменшу кількість перемикачів реле;
- високу адаптивність до зовнішніх умов.

5. PID-регулятор забезпечує задовільну швидкодію, проте потребує складного налаштування й демонструє меншу стійкість у випадку значних збурень.

6. Отримані результати моделювання підтверджують технічну доцільність використання інтелектуальних алгоритмів керування для систем опалення, особливо в умовах змінного клімату та необхідності підвищення енергоефективності будівель.

## РОЗДІЛ 4

### СПЕЦІАЛЬНА ЧАСТИНА

#### 4.1 Розробка алгоритму роботи системи опалення

Алгоритм роботи інтелектуальної системи керування опаленням базується на послідовній взаємодії підсистем збору даних, обробки інформації, прийняття рішень та керування виконавчими механізмами. Raspberry Pi 4 виступає центральним контролером, що виконує зчитування показників сенсорів, аналізує теплову ситуацію в приміщенні, прогнозує можливі зміни температури та оптимізує роботу нагрівального обладнання.

Алгоритм організовано таким чином, щоб забезпечити енергоефективність, комфорт користувача та адаптивність до зовнішніх чинників (погода, присутність мешканців, параметри будівлі).

Загальна структура алгоритму включає такі ключові етапи [16-19]:

1. Збір даних з сенсорів:
  - температура повітря в приміщенні (DHT22 / BME280);
  - вологість повітря (BME280);
  - температура теплоносія / радіатора (DS18B20);
  - зовнішня температура (сенсор або API погоди);
  - дані про присутність користувачів (телефони, датчики руху, геолокація);
  - енергоспоживання (опційно – через smart-реле).
2. Первинна обробка даних:
  - фільтрація шумів (фільтр середнього ковзання / Kalman);
  - перевірка достовірності показників (виявлення збоїв);
  - визначення темпу зміни температури  $dT/dt$ .
3. Формування теплової моделі:
  - обчислення теплової інерції приміщення;
  - прогноз зміни температури на основі RC-моделі;
  - прогноз енергоспоживання.
5. Обчислення керуючої дії.

Розглянемо блок-схему алгоритму роботи інтелектуальної системи опалення, поданої на рисунку 4.1.

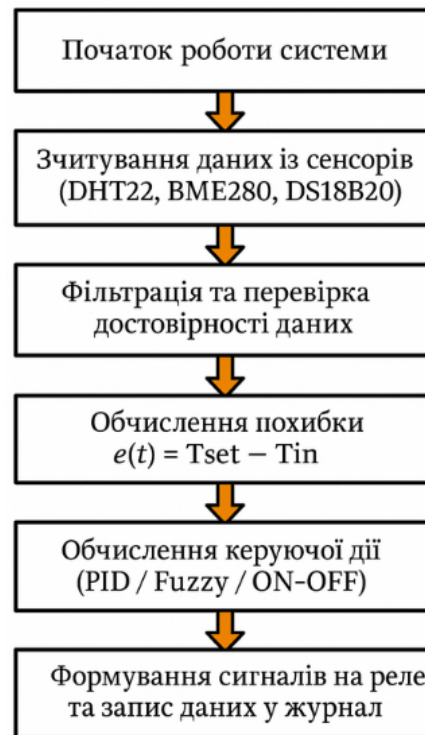


Рисунок. 4.1 – Блок-схема алгоритму роботи інтелектуальної системи опалення

На блок-схемі послідовно зображено етапи роботи системи:

- початок роботи системи;
- зчитування даних з сенсорів (DHT22, BME280, DS18B20);
- фільтрація та перевірка коректності даних;
- обчислення похибки  $e(t) = T_{set} - T_{in}$ ;
- обчислення керуючої дії (PID / Fuzzy / двопозиційне керування);
- формування сигналів на реле та запис результатів у журнал.

Розглянемо на рисунку 4.2 структурну схему на базі Raspberry Pi 4. На структурній схемі показано взаємодію основних блоків підсистем:

- блок «Сенсори (DHT22, BME280, DS18B20)» передає виміряні параметри в блок «Raspberry Pi 4 – алгоритм керування (PID / Fuzzy)»;
- з Raspberry Pi 4 виходять сигнали керування на модулі реле SSR / EMR;
- реле комутують нагрівальні елементи (котел, клапани, тепла підлога);

– паралельно Raspberry Pi 4 взаємодіє з хмарною системою моніторингу та web-інтерфейсом (двосторонній обмін даними).

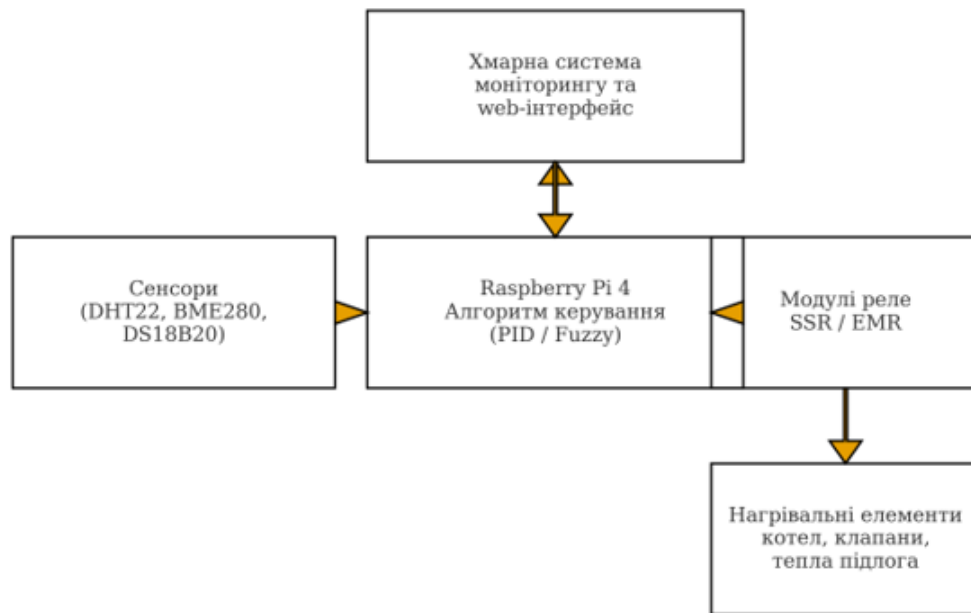


Рисунок 4.2 – Структурна схема системи керування опаленням на базі Raspberry Pi 4

На рисунку 4.3 розглянемо схему інтелектуального модуля регулювання.

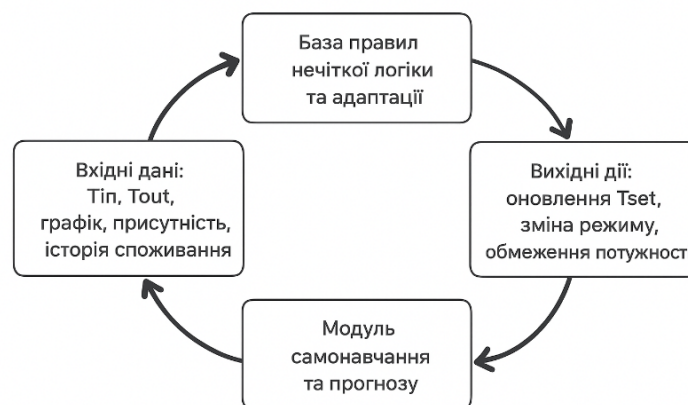


Рисунок 4.3 – Структурна схема інтелектуального модуля регулювання

Даний рисунок відображає внутрішню структуру інтелектуального модуля:

– вхідний блок: «Вхідні дані:  $T_{in}$ ,  $T_{out}$ , графік, присутність, історія споживання»;

- блок «База правил нечіткої логіки та адаптації», де зосереджені правила типу «якщо – то»;
- блок «Модуль самонавчання та прогнозу», який аналізує історичні дані та коригує параметри;
- вихідний блок: «Оновлення  $T_{set}$ , зміна режиму, обмеження потужності».

На рисунку 4.4 розглянемо UML діаграму послідовностей взаємодії сенсорів, контролера, реле та хмарного сервісу.

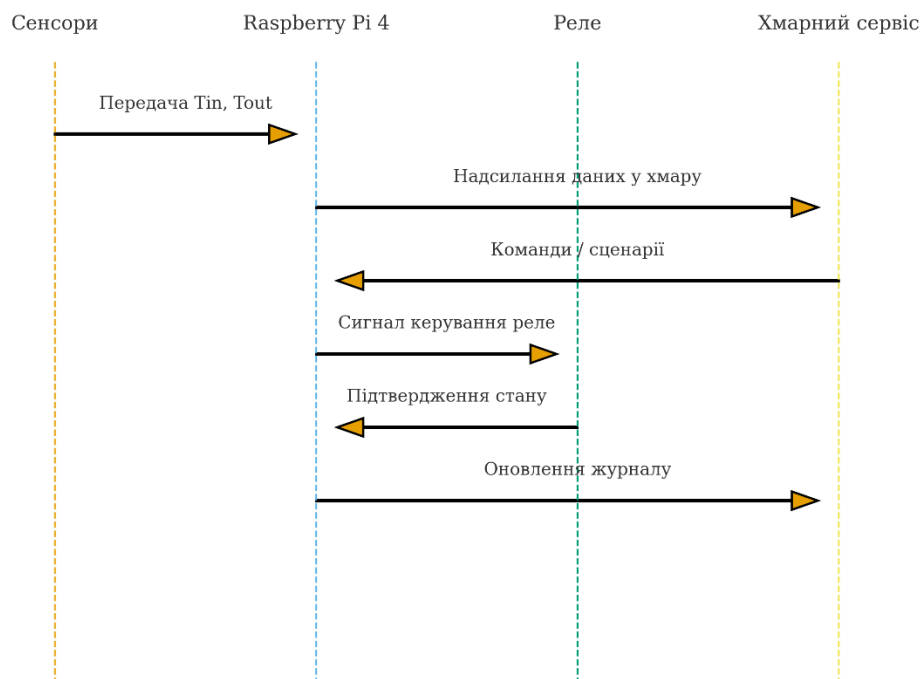


Рисунок 4.4 – Діаграма послідовностей взаємодії сенсорів, контролера, реле та хмарного сервісу

На діаграмі по осях по горизонталі подано лінії життя – «Сенсори», «Raspberry Pi 4», «Реле», «Хмарний сервіс». По вертикалі – час. Стрілками показано:

- передача даних  $T_{in}$ ,  $T_{out}$  від сенсорів до Raspberry Pi 4;
- надсилання телеметрії у хмару;
- отримання сценаріїв / команд з хмари;
- формування сигналів на реле;

- підтвердження стану;
- оновлення журналу у хмарі.

Цей рисунок демонструє динаміку взаємодії компонентів у часі.

Для кращого розуміння наочності процесу встановлення та регулювання температури на рисунку 4.5 розглянемо узагальнену комбіновану схему алгоритму та апаратної реалізації системи опалення.

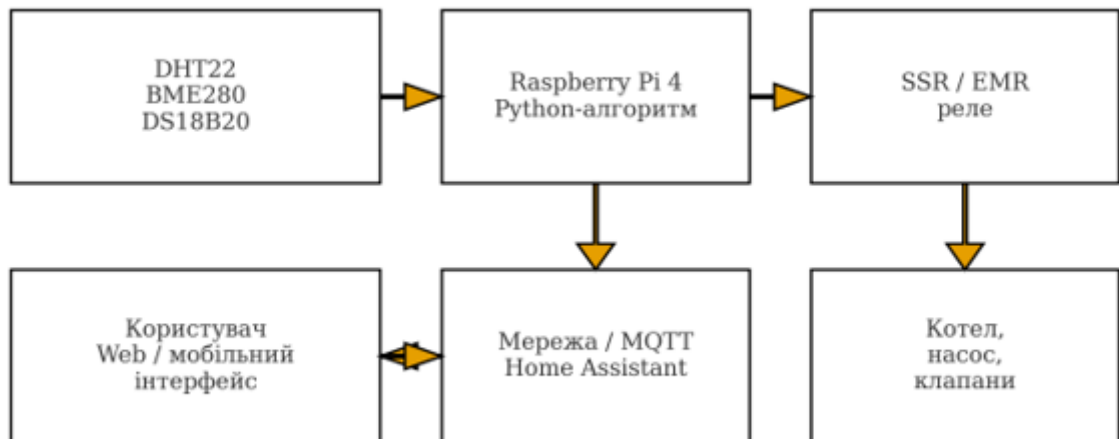


Рисунок 4.5 – Узагальнена комбінована схема алгоритму та апаратної реалізації системи опалення

Як бачимо, на комбінованій схемі поєднано:

- Raspberry Pi 4 (Python-алгоритм) у центрі;
- зліва – блок «DHT22, BME280, DS18B20» (сенсори);
- справа – «SSR / EMR реле» та далі «Котел, насос, клапани»;
- знизу – «Мережа / MQTT / Home Assistant» та «Користувач (web / мобільний інтерфейс)».

Стрілки показують повний шлях: сенсори → алгоритм → реле → нагрівальні елементи → мережа → користувач.

Розглянемо інтелектуальну логіку системи, яка дозволяє динамічно адаптуватися до умов, підтримуючи тепловий комфорт і мінімізуючи споживання енергії. Для цього вона повинна здійснити наступні етапи [19]:

1. Адаптивна уставка температури. Система автоматично змінює  $T_{set}$ , враховуючи:

- час доби;
- наявність людей у будинку;
- поведінкові шаблони (machine learning);
- погоду та прогноз зовнішньої температури.

2. Компенсація зовнішніх збурень. Система оцінює теплову інерцію і коригує потужність заздалегідь за законом, поданим у формулі (4.1):

$$T_{set}^{adapt} = T_0 + \alpha(T_{out} - T_{base}), \quad (4.1)$$

де  $\alpha$  – коефіцієнт адаптації.

3. Прогнозування теплових процесів. Система передбачає падіння температури та запускає нагрів до того, як температура стане критично низькою.

4. Енергетична оптимізація:

- мінімізація кількості включень реле;
- вибір режиму роботи котла залежно від тарифів;
- пріоритетне опалення активних зон будівлі.

5. Самонавчання. Принцип роботи:

- система аналізує погодні умови за останні 7...30 днів;
- коригує теплову модель будівлі;
- підлаштовує гістерезис, PID-параметри та Fuzzy-правила.

6. Реагування на аварійні ситуації:

- перегрів котла → аварійне відключення реле;
- відмова сенсора → fallback-режим: підтримання 8...10 °С;
- надто швидке падіння температури → сигнал про відкриття вікна.

Отже, у даному підрозділі розроблено повний алгоритмічний цикл роботи інтелектуальної системи опалення на базі Raspberry Pi 4, який охоплює збір, обробку та аналіз даних, формування теплової моделі, вибір керуючої дії та взаємодію з апаратною частиною. Система забезпечує адаптивне керування температурою приміщень, реагує на зовнішні та внутрішні збурення,

використовує методи прогнозування й самонавчання для оптимізації енергоспоживання. Подані блок – схеми (рис. 4.1 – 4.5) демонструють логічні зв'язки між сенсорами, контролером, реле та хмарним сервісом, а також внутрішню структуру інтелектуального модуля. Завдяки поєднанню алгоритмів нечіткої логіки, PID-регулювання, прогнозних моделей та адаптивних механізмів система здатна забезпечувати тепловий комфорт, знижувати витрати енергії та підвищувати надійність роботи опалювального обладнання в реальних умовах експлуатації.

## 4.2 Вибір середовища та створення програмного забезпечення

Програмне забезпечення інтелектуальної системи керування опаленням реалізовано на мові програмування Python з використанням веб – фреймворків Flask / FastAPI, брокера повідомлень MQTT, вбудованої СУБД SQLite для локального зберігання даних та системи візуалізації Grafana. Такий вибір зумовлений поєднанням простоти розробки, широкої підтримки бібліотек для роботи з Raspberry Pi 4, модульністю та можливістю інтеграції з IoT-платформами.

Python є де-факто стандартом для розробки прикладних рішень під Raspberry Pi 4 завдяки великій кількості бібліотек для роботи з GPIO, протоколами I<sup>2</sup>C, 1-Wire, SPI, MQTT та HTTP-сервісами. Flask та FastAPI забезпечують побудову легковагових REST та Web-інтерфейсів, тоді як зв'язка MQTT + SQLite + Grafana формує основу для збору, зберігання й аналітичної візуалізації даних про мікроклімат та режими роботи системи опалення.

Розглянемо вибір програмного середовища. До основних компонентів програмного середовища належать [20]:

1. Python 3.x. Використовується як основна мова реалізації логіки:
  - обробка даних сенсорів (бібліотеки Adafruit\_DHT, smbus2, робота з 1-Wire);
  - реалізація алгоритмів PID / нечіткої логіки;
  - взаємодія з MQTT-брокером (raho-mqtt);

– реалізація веб-API (Flask / FastAPI).

## 2. Flask / FastAPI:

– Flask – мінімалістичний фреймворк для побудови простих веб – інтерфейсів та REST-ендпоінтів (сторінки стану системи, форми для зміни уставок, перегляд логів);

– FastAPI – сучасний асинхронний фреймворк, який забезпечує високу продуктивність, автоматичну генерацію документації API (OpenAPI/Swagger) та зручну інтеграцію з фронтенд-клієнтами й мобільними застосунками. Використання цих фреймворків дає змогу організувати доступ до системи опалення через веб-браузер або мобільний додаток.

3. MQTT (наприклад, Mosquitto Broker). Протокол MQTT застосовується для:

– публікації телеметрії з Raspberry Pi до інших вузлів / хмарних сервісів;  
– отримання команд управління (зміна режиму, уставок, аварійні сигнали);

– інтеграції з Home Assistant або іншими smart home-платформами. На Raspberry Pi працює клієнтський код (paho-mqtt), тоді як брокер може бути локальним або хмарним.

## 4. SQLite. Локальна СУБД для:

– зберігання історичних даних (температура, вологість, стан реле);  
– протоколювання аварійних подій;  
– збереження налаштувань користувача та конфігурацій системи.

Перевага SQLite – відсутність потреби в окремому сервері бази даних та простота застосування на вбудованих системах.

## 5. Grafana (зазвичай у зв'язці з SQLite / InfluxDB / Prometheus).

Використовується для:

– побудови дашбордів з температурними графіками;  
– порівняння режимів роботи системи;

– моніторингу енергоспоживання. Grafana підключається до БД, де акумулюються дані з Raspberry Pi 4, і дозволяє наочно аналізувати ефективність алгоритмів керування.

Таким чином, обраний стек ПЗ орієнтований на відкриті технології, високу гнучкість та сумісність із сучасними IoT-рішеннями.

Розглянемо структуру програмного коду. Програмний код системи доцільно організувати за модульним принципом.

Приклад логічної структури каталогу проєкту подано фрагментом програмного коду, наведеного в лістингу 4.1.

#### Лістинг 4.1 – Приклад коду логічної структури каталогу проєкту

---

```
# main.py - точка входу програми (ініціалізація модулів,
запуск циклу керування);
# config.py - налаштування (GPIO-піни, параметри MQTT,
уставки температури, шляхи до БД);
# sensors/ - модулі роботи з датчиками:
# dht22_reader.py
# bme280_reader.py
# ds18b20_reader.py
# control/ - модулі алгоритмів керування:
# pid_controller.py
# fuzzy_controller.py
# hysteresis_controller.py
# io/ - взаємодія із зовнішнім світом:
# gpio_relay.py - керування реле SSR/EMR;
# mqtt_client.py - взаємодія з MQTT-брокером;
# database.py - робота з SQLite;
# web/ - веб-інтерфейс:
# flask_app.py або fastapi_app.py;
# utils/ - допоміжні функції (логування, фільтрація
сигналів, перетворення одиниць).
```

---

#### Кінець лістингу 4.1

Такий поділ полегшує розробку, тестування та подальшу модернізацію системи: зміна фізичних сенсорів або протоколів зв'язку не потребує переписування усієї програми – достатньо модифікувати відповідний модуль.

Розглянемо основні модулі програмного забезпечення [21-23].

1. Модуль збору даних із сенсорів. Реалізує періодичне опитування DHT22, BME280, DS18B20, попередню фільтрацію показників, контроль коректності даних та передачу структурованої інформації (наприклад, у форматі Python-словника) в модуль керування.

2. Модуль алгоритмів керування (Control Layer) містить реалізації:

- PID-регулятора (розрахунок  $u(t)$  за похибкою, інтегральною та диференційною складовими);
- нечіткого регулятора (на основі бібліотек Fuzzy Logic або власної реалізації);
- двопозиційного керування з гістерезисом для простих сценаріїв.

Модуль отримує задану температуру, поточну температуру, інформацію про режим (Комфорт / Еко / Нічний) і формує команду для виконавчих механізмів.

3. Модуль керування реле (GPIO-інтерфейс) здійснює:

- перетворення логічних команд у сигнали GPIO;
- контроль стану реле;
- реалізацію затримок для захисту від частого перемикавання (anti short-cycle).

4. Модуль комунікацій (MQTT + Web-API) забезпечує:

- публікацію поточних параметрів системи у MQTT-топіках (наприклад, home / heating / room / temperature);
- прийом команд (зміна уставок, перехід у інший режим);
- реалізацію Web API-інтерфейсу (Flask / FastAPI) для браузера та мобільного застосунку (отримання стану, відправка команд).

5. Модуль зберігання та візуалізації даних:

- запис вимірених параметрів та стану реле до SQLite;
- підготовку даних для Grafana (через прямий доступ до БД або проміжний API);
- логування аварійних та сервісних подій.

6. Модуль конфігурації та самодіагностики:

- зчитування конфігурацій (JSON / YAML / INI);
- ініціалізація усіх підсистем;
- тестові запуски перевірки сенсорів і реле при старті системи;
- формування повідомлень про помилки.

Обґрунтуємо вибір програмного підходу. Запропонована архітектура ПЗ та вибір технологій забезпечують:

- масштабованість (можна додавати нові зони та датчики);
- портованість (Python-код легко адаптувати під інші платформи);
- інтеграцію з IoT (MQTT, Home Assistant, Grafana);
- зручність обслуговування (чіткий поділ на модулі, простота оновлень).

Це повністю відповідає вимогам щоб створити не лише працездатну, а й гнучку, інтелектуальну систему, готову до подальшого розвитку.

### **4.3 Розробка методики програмування, впровадження та підтримки ПЗ**

Методика програмування інтелектуальної системи керування опаленням базується на поєднанні принципів модульності, надійності, можливості масштабування та відповідності сучасним підходам до розробки IoT-рішень. Програмне забезпечення має забезпечувати стабільну роботу в реальному часі, обробку сигналів від сенсорів, реалізацію алгоритмів керування, а також інтеграцію з веб-інтерфейсом і хмарними сервісами.

Методичні принципи побудови програмного забезпечення будуються на модульності та розподілі функціональності [24].

Код поділяється на окремі модулі:

- sensors – читання температури / вологості;
- control – PID, нечітка логіка, адаптивні моделі;
- io – робота з GPIO-реле, MQTT, базами даних;
- web – реалізація Flask / FastAPI API;
- utils – фільтри, логування, обробка винятків;
- main.py – єдиний вхідний файл, що координує роботу системи.

Такий підхід дозволяє:

- легко оновлювати окремі частини,
- проводити тестування,
- впроваджувати нові функції без зміни основної архітектури.

Розглянемо структурну схему програмної архітектури інтелектуальної системи керування опаленням, поданої на рисунку 4.6.

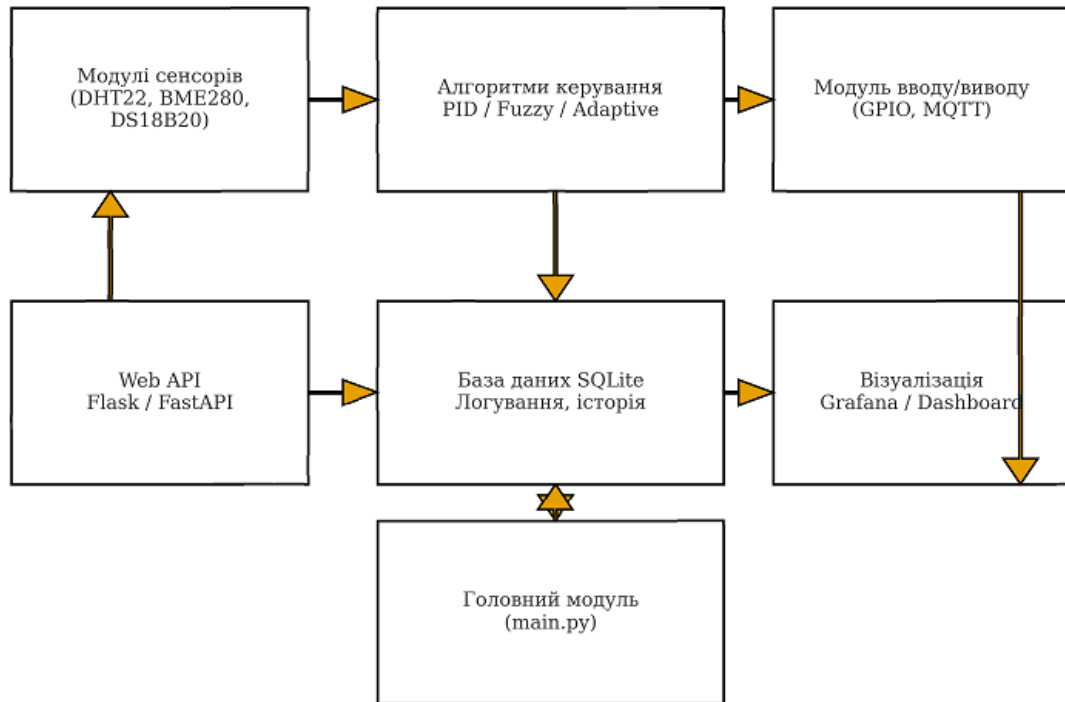


Рисунок 4.6 – Структурна схема програмної архітектури інтелектуальної системи керування опаленням

Схема відображає логічну взаємодію між основними модулями програмного забезпечення:

- модулі сенсорів (DHT22, BME280, DS18B20) забезпечують зчитування параметрів мікроклімату;
- алгоритмічний модуль керування реалізує PID-, Fuzzy- та адаптивні алгоритми;
- модуль вводу / виводу (GPIO та MQTT) взаємодіє з реле, виконавчими пристроями та мережею;
- Web API (Flask / FastAPI) слугує інтерфейсом для користувача та зовнішніх систем;
- база даних SQLite зберігає історію вимірювань, журнали подій та конфігураційні дані;
- візуалізаційний модуль (Grafana) надає аналітичні панелі та теплові графіки;
- головний модуль (main.py) координує всі компоненти, виконує запуск сервісів та логіку циклу керування.

Подана структурна схема демонструє логічно вибудований обмін даними між основними компонентами інтелектуальної системи опалення. Дані від сенсорів надходять у модуль керування, де обробляються PID, Fuzzy або адаптивними алгоритмами, після чого керуючі сигнали передаються до модуля вводу / виводу. Інформація фіксується у базі даних SQLite, що забезпечує журналювання й збереження історії. Паралельно система взаємодіє з Web API та модулем візуалізації, дозволяючи користувачеві отримувати доступ до актуальних параметрів та аналітики. Головний модуль main.py координує роботу всіх підсистем, забезпечуючи узгоджений, безперервний та адаптивний цикл керування опаленням [25].

Розглянемо основні особливості методики розробки програмування, яка включає:

1. Визначення функціональних вимог:

- необхідні сенсори;
- типи алгоритмів керування;
- вимоги до логування і візуалізації;
- інтеграція з мобільними / веб-інтерфейсами.

2. Проектування структури програми:

- створення UML-схем, блок-схем алгоритмів;
- опис API-ендпоінтів;
- моделювання потоків даних.

3. Реалізація поетапного програмування. Побудова ПЗ проводиться у три етапи:

- етап 1: тестування сенсорів і реле;
- етап 2: реалізація алгоритмів (PID, Fuzzy);
- етап 3: інтеграція з MQTT, Grafana, Home Assistant.

4. Тестування:

- модульне тестування (pytest);
- навантажувальне тестування (частота запитів, обробка пікових навантажень);
- перевірка роботи при відмовах датчиків (fault-tolerance).

Розглянемо методику впровадження ПЗ на Raspberry Pi 4. Послідовність впровадження поділяється на:

1. Підготовка Raspberry Pi OS – встановлення системи, налаштування SSH, автоматичного запуску сервісів через systemd.

2. Інсталяція програмних залежностей:

– Python 3.x;

– бібліотеки для сенсорів;

– Flask / FastAPI;

– paho-mqtt;

– SQLite;

– Grafana (за необхідності – на хмарному сервері).

3. Розгортання MQTT-брокера – використання Mosquitto (локально або у мережі).

4. Налаштування сервісів автозапуску – кожний модуль запускається як Linux-служба для забезпечення безперервної роботи.

5. Інтеграція з хмарною платформою:

– Home Assistant;

– мобільний застосунок;

– веб-інтерфейс користувача.

Розглянемо методику підтримки та оновлення ПЗ. Для забезпечення надійної та довготривалої експлуатації необхідно впровадити такі механізми:

1. Логи та моніторинг:

– збереження журналів у SQLite;

– онлайн-моніторинг через Grafana;

– повідомлення через MQTT / Telegram (аварійні ситуації).

2. Самодіагностика системи:

– перевірка доступності сенсорів;

– перевірка стану GPIO;

– контроль температури CPU Raspberry Pi 4.

3. Безпечне оновлення ПЗ. Пропонується використання:

– git-завантажень;

- окремої тестової гілки;
- резервних копій конфігурації.

#### 4. Захист від збоїв:

- Watchdog-драйвер Raspberry Pi;
- autorestart через systemd;
- дублювання критичних параметрів у пам'яті.

Отже, у четвертому розділі було здійснено комплексний аналіз та практичне обґрунтування вибору програмної та апаратної платформи для інтелектуальної системи керування опаленням. Обрано сучасний стек технологій (Python, Flask / FastAPI, MQTT, SQLite, Grafana), що забезпечує високу продуктивність, модульність, масштабованість та гнучкість системи.

Було визначено чітку методику розробки програмного забезпечення, розроблено структуру ПЗ, описано ключові модулі, алгоритми та принципи їх взаємодії. Визначено порядок впровадження ПЗ на Raspberry Pi 4 та методи підтримки й оновлення.

Для реалізації мобільного керування інтелектуальною системою опалення було використано додаток Home Assistant, доступний у Google Play Market. Процедура встановлення включала пошук додатку в магазині, інсталяцію та запуск на смартфоні. Для підключення до системи на Raspberry Pi 4 необхідно, щоб мобільний пристрій та Raspberry Pi 4 перебували в одній локальній мережі. Після запуску Home Assistant додаток автоматично визначив локальний сервер за адресою <http://IP:8123>, після чого було виконано авторизацію.

У меню «Devices & Services» додано інтеграції MQTT та Climate, що забезпечили обмін даними між мобільним додатком і Raspberry Pi 4. Створено інтерфейс керування (Dashboard), що містить картки контролю температури, сенсорів мікроклімату та кнопки керування нагрівальним елементом. За потреби було налаштовано віддалений доступ через Home Assistant Cloud. Таким чином, мобільний додаток забезпечує повнофункціональне управління режимами опалення, моніторинг параметрів середовища та отримання

сповіщень у режимі реального часу. Більш детально порядок інсталювання програми Home Assistant та скріншоти її використання подано в додатку В.

Обґрунтовано застосування Raspberry Pi 4 як основної платформи завдяки її високій обчислювальній потужності, наявності повноцінної ОС, широкій підтримці мов програмування та можливості інтеграції з хмарними сервісами.

Таким чином, розроблена методика програмування та підтримки ПЗ дозволяє реалізувати надійну, автоматизовану та енергоефективну систему керування опаленням, придатну до практичного впровадження та подальшого розширення [26].

## ВИСНОВКИ

У ході виконання роботи було досягнуто наступних результатів:

1. Проаналізовано традиційні системи опалення та визначено їхні недоліки. Це обґрунтувало необхідність інтелектуалізації процесу керування.

2. Обґрунтовано вибір Raspberry Pi 4 як базового контролера у завданнях реального часу, обробки великих масивів даних, застосування алгоритмів нечіткої логіки, PID-регулювання, машинного навчання, а також реалізації веб-сервісів та IoT-комунікацій.

3. Розроблено функціональну структуру системи з урахуванням сенсорів, виконавчих елементів і каналів зв'язку. Створено апаратно-програмний комплекс, який включає сенсори DHT22, BME280 та DS18B20, модулі реле SSR / EMR, систему передачі даних через MQTT, хмарний сервіс моніторингу та зручний веб-інтерфейс. Запропонована структура забезпечує модульність, масштабованість і можливість інтеграції у «розумний будинок».

4. Реалізовано інтелектуальні алгоритми оптимізації роботи системи. Розроблені алгоритми поєднують PID-регулювання, нечітку логіку та моделі прогнозування температури. Передбачено механізми самонавчання, оцінку теплової інерції, адаптацію уставки температури до контексту (погода, присутність користувачів, режим дня).

5. Забезпечено можливість віддаленого моніторингу та керування. Реалізовано веб-інтерфейс на базі Flask / FastAPI, обмін даними через MQTT, зберігання історії в SQLite та візуалізацію параметрів у Grafana Dashboard.

Розроблена інтелектуальна система керування домашнім опаленням є технічно обґрунтованою, адаптивною, енергоефективною та може бути рекомендована для впровадження у житлових і комерційних приміщеннях, а також для подальших наукових досліджень у галузі автоматизації та «розумних» енергетичних систем.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. De la Puente-Gil Á., González-Martínez A., Rosales-Asensio E. та ін. Efficient Heating System Management Through IoT Smart Devices. *Machines*, 2025, 13(8), 643 p.
2. Felez R., Felez J. Advanced Energy Management for Residential Buildings Optimizing Costs and Efficiency Through Thermal Energy Storage and Predictive Control. *Applied Sciences*, 2025, 15(2), 880 p.
3. Hakawati B., Mousa A., Draidi F. Smart Energy Management in Residential Buildings: The Impact of Knowledge and Behavior. *Scientific Reports*, 2024, 14(1), 1702 p.
4. Qayyum F., Jamil H., Ali F. A Review of Smart Energy Management in Residential Buildings for Smart Cities. *Energies*, 2023, 17(1), 83 p.
5. De Bock Y. The Energy Saving Potential of Retrofitting a Smart Heating System. *Energy and Buildings*, 2021. URL: <https://www.sciencedirect.com/science/article/pii/S2210537921000743>.
6. Розумне опалення. Офіційний сайт Danfoss. URL: <https://www.danfoss.com/uk-ua/markets/buildings-residential/dhs/smart-heating/> (дата звернення 21.05.2025 р.).
7. Розумний дім: розумне опалення вдома. Офіційний сайт Viessmann Україна. URL: <https://www.viessmann.ua/uk/rishennya/smart-tekhnologiyi/rozumnyu-dim.html> (дата звернення 20.06.2025 р.).
8. European Commission. Energy performance of buildings 2023. URL: <https://energy.ec.europa.eu> (дата звернення 22.06.2025 р.).
9. Google Nest – Energy Savings Report. URL: <https://nest.com/energy-savings> (дата звернення 25.07.2025 р.).
10. Tado° Smart Thermostat – Energy Savings Whitepaper. URL: <https://www.tado.com> (дата звернення 21.08.2025 р.).
11. Fraunhofer Institute for Building Physics (IBP). Smart Heating Control Study. URL: <https://www.ibp.fraunhofer.de> (дата звернення 19.08.2024 р.).
12. Netatmo Smart Heating – Technical Documentation. URL:

<https://www.netatmo.com> (дата звернення 15.09.2025 р.).

13. Raspberry Pi Foundation – Raspberry Pi 4 Specifications. URL: <https://www.raspberrypi.com> (дата звернення 15.09.2025 р.).

14. Bosch Sensortec – BME280 Datasheet. URL: <https://www.bosch-sensortec.com> (дата звернення 18.10.2025 р.).

15. Maxim Integrated – DS18B20 Datasheet. URL: <https://www.analog.com> (дата звернення 15.11.2025 р.).

16. Adafruit. DHT22/AM2302 Temperature-Humidity Sensor Guide. URL: <https://learn.adafruit.com> (дата звернення 18.11.2025 р.).

17. MQTT.org. MQTT Protocol Standard – OASIS Documentation. URL: <https://mqtt.org> (дата звернення 25.11.2025 р.).

18. Home Assistant Project. IoT Automation Framework Documentation. URL: <https://www.home-assistant.io/docs> (дата звернення 29.11.2025 р.).

19. WS IoT Core. Cloud Integration Documentation. URL: <https://aws.amazon.com/iot-core/> (дата звернення 30.11.2024 р.).

20. LabCenter Electronics. Proteus Design Suite – Technical Documentation. York, UK: LabCenter Electronics Ltd., 2023. 412 p. URL: <https://www.labcenter.com> (дата звернення 30.11.2025 р.).

21. MathWorks. MATLAB & Simulink User's Guide. Natick, MA: MathWorks, 2023. 2864 p. URL: <https://www.mathworks.com/help> (дата звернення 05.12.2025 р.).

22. Isermann R. Digital Control Systems: Modeling, Identification, and Adaptive Control. Cham, Switzerland: Springer, 2021. 758 p.

23. Bejan A. Advanced Engineering Thermodynamics. Hoboken, NJ: Wiley, 2022. 824 p.

24. Schwarz M. Practical PID Control for HVAC Systems. Amsterdam: Elsevier, 2023. 392 p.

25. Natick, MA: MathWorks, 2023. 632 p. URL: <https://www.mathworks.com/help/sldo> (05.12.2025 р.)

26. Хвищун М. В., Сидорук В. В., Хвищун Д. М., Бернасюк М. В. Розроблення інтелектуальної системи керування домашнім опаленням з

використанням Raspberry Pi 4. International Scientific and Practical Conference of Young Scientists and Students «Actual Problems of Automation and Control», №13., 27 листопада 2025 р., м. Луцьк, с.142-148.

## ДОДАТКИ

Додаток А  
Тези доповіді на конференції

International Scientific and Practical Conference of Young Scientists and Students

об'єктно-орієнтовані технології

ЕОМ

КОНТРОЛЕР

ВИРОБНИЧІ ПРОЦЕСИ

МІКРОПРОЦЕСОРНА ТЕХНІКА

# ACTUAL PROBLEMS OF AUTOMATION AND CONTROL

conference materials

АСУ

ІНФОРМАЦІЙНА БАЗА

АСУ ТП

КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

ЕОМ

Інформаційні системи

Issue № 13

Lutsk - 2025

**СЕКЦІЯ «МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ СИСТЕМ УПРАВЛІННЯ»**

Крупський О. В., Приходько О. С., Вавринюк К. В. Розробка інструменту для генерації анотованих датасетів для навчання AI-агентів .....	83
Лишук В., Крутий П., Антонюк В. Найпростіша математична модель однофазного випростувача .....	86
Міщенко Д. О. Оцінка соціально-економічних переваг впровадження систем управління на базі DC-мікромереж у сільських населених пунктах з альтернативними джерелами енергії .....	89
Нечипорук Р. О. Система інтелектуальної діагностики стану електроенергетичного обладнання на основі багатосенсорних даних та гібридних моделей штучного інтелекту .....	92
Стьопкін В. В., Білий В. В., Морозов М. В. Розробка математичної моделі електропривода постійного струму зі спостерігачем стану та пружними ланками .....	94
Юрченко Ю. В., Заковоротний О. Ю. Підготовка та аналіз джерел фінансових даних, як початковий етап стохастичного моделювання часових рядів методами машинного навчання .....	99

**СЕКЦІЯ «МЕХАТРОНІКА ТА РОБОТИЗОВАНІ СИСТЕМИ»**

Гончар А. В., Охримович М. Б. Вплив технологічних факторів на кінематичну точність циліндричних зубчатих коліс .....	104
Павлович А. О. Удосконалення комбінованих захоплюючих пристроїв та особливості їх проектування .....	108
Слабкий А. В., Котик С. І. Аналіз конструкцій та перспективи модернізації універсальних випробувальних машин .....	114

**СЕКЦІЯ «АВТОМАТИКА В ЕЛЕКТРОНІЦІ ТА ТЕЛЕКОМУНІКАЦІЯХ»**

Герман Б. А., Бартошик О. В., Цюняшук Я. В. Дистанційне керування сигналізацією за допомогою модуля GSM .....	117
Євсюк М. М., Ковалюк Н. В. Розроблення мікроконтролерного терміналу для автоматизованого керування системами вуличного освітлення .....	121
Заблоцький В. Ю., Гикавий С. В. Розроблення системи відеоспостереження промислового підприємства .....	124
Лишук В., Денисюк К. Застосування принципів спектрального ущільнення для збільшення пропускної здатності оптоволокна .....	131
Лотоцький В. І., Мельник О. В., Власик О. О. Аналіз методів регулювання температури для паяльного обладнання .....	134
Хвищун М. В., Кречик А. П., Хвищун Д. М., Рубльов В. В. Розроблення бездротового комунікаційного пристрою на основі STM32 та LoRa .....	137
Хвищун М. В., Сидорук В. В., Хвищун Д. М., Бернасюк М. В. Розроблення інтелектуальної системи керування домашнім опаленням з використанням Raspberry Pi 4 .....	142
Шумік А. О., Шибенюк Р. А., Захарчук М.Д., Хвищун М. В. Система розпізнавання голосових команд на базі Raspberry Pi .....	148
Якимчук Н. М., Карпіньський Н. К. Організація UART-комунікації в STM32 для системи комутації кінцевих пристроїв .....	152
Якимчук Н. М., Лишук В. В. Методологія інтегрованого моделювання та спільного проектування в сучасних електронних системах .....	156

**International Scientific and Practical Conference of Young Scientists and Students "Actual Problems of Automation and Control"**

6. Adelantado F. et al. Understanding the limits of LoRaWAN. *IEEE Communications Magazine*, 2021, 55(9): p.34–40. doi:10.1109/MCOM.2017.1600613.

7. LoRa Alliance. LoRaWAN® 1.1 Specification. Fremont: LoRa Alliance, 2021. 210 p.

УДК 621.383.004.85:004.272.3

Хвищун М. В., Сидорук В. В., Хвищун Д. М., Бернасюк М. В.

Луцький національний технічний університет, м. Луцьк, Україна

**РОЗРОБЛЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ КЕРУВАННЯ ДОМАШНІМ  
ОПАЛЕННЯМ З ВИКОРИСТАННЯМ RASPBERRY PI 4**

У роботі представлено розроблення інтелектуальної системи керування домашнім опаленням з використанням одноплатного комп'ютера Raspberry Pi 4, спрямованої на підвищення енергоефективності, комфорту та автономності сучасних систем житлового теплопостачання. Проаналізовано недоліки традиційних побутових термостатів і систем ручного регулювання, які не забезпечують адаптивної логіки роботи, можливостей віддаленого моніторингу та оптимізації теплових режимів відповідно до динамічних змін мікроклімату й графіків перебування користувачів, що значно обмежує їх ефективність в реальних умовах експлуатації.

Запропонований підхід базується на застосуванні апаратної платформи Raspberry Pi 4, яка завдяки високій продуктивності, широкому набору периферійних інтерфейсів та розвинутому програмному забезпеченню дозволяє реалізувати функції збору даних із температурних та кліматичних датчиків, аналізу параметрів мікроклімату, виконання інтелектуальних алгоритмів регулювання, а також керування виконавчими пристроями—електричними чи газовими котлами, сервоприводами та термодіафрагмами. Додатково забезпечується інтеграція зі службами мережевого доступу, що дозволяє здійснювати передачу інформації через Wi-Fi або локальний MQTT-сервер та підтримувати віддалений контроль системи через веб-інтерфейс.

Окрему увагу приділено структурі системи, вибору сенсорів температури та вологості, способам побудови виконавчих каналів, методам резервування живлення та реалізації сучасних алгоритмів інтелектуального керування, зокрема PID-регулювання, адаптивних моделей теплового прогнозування та сценарного контролю з урахуванням поведінкових факторів. Розглянуто особливості інтеграції програмних модулів, організації баз даних та побудови локального сервера керування, що забезпечує можливість безперервного моніторингу, аналізу історичних даних і автоматичного коригування теплових режимів з метою зменшення енергоспоживання.

*Ключові слова:* інтелектуальна система керування; домашнє опалення; Raspberry Pi 4; автоматизація; енергоефективність; сенсори температури; терморегуляція; IoT; релеїні модулі; алгоритми регулювання; PID-контроль; хмарний моніторинг; Python; MQTT; смарт-будинок.

**Khvyshchun M. V., Sydoruk V. V., Khvyshchun D. M., Bernasiuk M. Development of an intelligent home heating control system using Raspberry Pi 4.** The paper presents the development of an intelligent home heating control system based on the Raspberry Pi 4 single-board computer, designed to enhance energy efficiency, comfort, and the autonomous operation of modern residential heating systems. The shortcomings of traditional household thermostats and manual control systems are analyzed, as they lack adaptive logic, remote monitoring capabilities, and the ability to optimize thermal modes in response to dynamic microclimate changes and user presence patterns, which significantly limit their effectiveness under real-world operating conditions.

The proposed approach is based on the use of the Raspberry Pi 4 hardware platform, which—due to its high performance, broad set of peripheral interfaces, and advanced software environment—enables the implementation of functions such as data acquisition from temperature and climate sensors, microclimate parameter analysis, execution of intelligent control algorithms, and control of actuators including electric or gas boilers, servomotors, and thermal valves. Additionally, integration with network-access services is ensured, allowing information transmission via Wi-Fi or a local MQTT server, as well as supporting remote system control through a web interface.

Special attention is given to the system architecture, the selection of temperature and humidity sensors, the design of actuator channels, power backup methods, and the implementation of modern intelligent control algorithms, including PID regulation, adaptive heat load forecasting models, and scenario-based regulation that considers behavioral factors. The study examines the integration of software

## International Scientific and Practical Conference of Young Scientists and Students "Actual Problems of Automation and Control"

modules, database organization, and the development of a local control server, which enables continuous monitoring, historical data analysis, and automatic correction of thermal modes to reduce overall energy consumption.

*Keywords: intelligent control system; home heating; Raspberry Pi 4; automation; energy efficiency; temperature sensors; thermostatic regulation; IoT; relay modules; control algorithms; PID control; cloud monitoring; Python; MQTT; smart home.*

**Постановка проблеми.** У сучасних умовах стрімкого зростання вартості енергоносіїв та підвищених вимог до комфорту та безпеки житлових приміщень особливо актуальним є створення енергоефективних систем керування опаленням. Традиційні побутові термостати та найпростіші схеми регулювання, що працюють за принципом включення та вимкнення при досягненні заданих температурних порогів, не враховують реальну динаміку теплових процесів у приміщеннях, індивідуальні особливості проживання користувачів та зміни мікрокліматичних умов. Відсутність гнучкої логіки управління, віддаленого моніторингу та можливості адаптивного налаштування істотно знижує ефективність таких систем, призводячи до перевитрат енергоресурсів та дискомфорту.

Розвиток технологій Інтернету речей (IoT), поява доступних одноплатних комп'ютерів та широке поширення інтелектуальних сенсорів відкривають можливості для створення більш складних та адаптивних систем керування. Проте значна кількість існуючих рішень або є дорогими комерційними продуктами, або не забезпечує необхідної надійності, масштабованості й можливостей інтеграції у сучасні системи «розумного будинку».

У зв'язку з цим постає необхідність у розробленні інтелектуальної системи керування домашнім опаленням, яка здатна автоматично аналізувати параметри мікроклімату, прогнозувати теплове навантаження, адаптувати режими роботи до умов експлуатації та забезпечувати можливість дистанційного керування. Використання одноплатного комп'ютера Raspberry Pi 4 як центрального елемента такої системи дозволяє реалізувати складні алгоритми теплової регуляції, забезпечити інтеграцію з різноманітними сенсорами та виконавчими механізмами й організувати надійний каналний зв'язок для моніторингу та обміну даними.

Таким чином, проблема створення інтелектуальної, універсальної та економічно доцільної системи керування опаленням на базі Raspberry Pi 4 є актуальною та потребує комплексного дослідження, що включає аналіз апаратної частини, побудову алгоритмів регулювання, розробку програмного забезпечення та експериментальну перевірку ефективності роботи системи [1].

**Аналіз останніх досліджень.** Проблематика енергоефективного керування системами опалення активно розглядається у працях як вітчизняних, так і зарубіжних дослідників. Значна частина досліджень присвячена створенню так званих «розумних термостатів», які дозволяють автоматизувати підтримання температури в житлових приміщеннях та знизити споживання енергоресурсів. На ринку представлено велику кількість комерційних рішень (Nest, Tado, Netatmo, Ecobee тощо), які реалізують базові функції адаптивного керування, віддаленого доступу через мобільні додатки та інтеграції у екосистеми «розумного дому». Разом з тим, більшість таких систем мають закриту архітектуру, обмежені можливості модернізації й налаштування, а також орієнтовані на типові західні стандарти будівель і опалювальних систем, що ускладнює їх безпосереднє застосування у вітчизняних умовах.

Окремий напрям досліджень пов'язаний із використанням технологій Інтернету речей (IoT) для моніторингу та керування тепловими системами будівель. У наукових публікаціях розглядаються підходи до побудови розподілених мереж сенсорів температури, вологості, якості повітря та теплового навантаження, що дозволяють формувати детальну картину мікроклімату в реальному часі. Пропонуються різноманітні архітектури на базі хмарних платформ, MQTT-брокерів, локальних серверів та мобільних клієнтів. При цьому акцент робиться на забезпеченні стійкого зв'язку, масштабованості та безпеки передавання даних. Однак багато рішень залишаються демонстраційними прототипами, які не враховують повний цикл експлуатації систем опалення в реальних житлових умовах (сезонність, непостійність присутності мешканців, особливості будівельних конструкцій тощо).

Досить широко досліджується застосування різних алгоритмів керування тепловими процесами. Традиційно використовуються пропорційно-інтегрально-диференціальні (PID) регулятори, які забезпечують прийнятний рівень стабілізації температури при відносно простій реалізації. У більш сучасних роботах пропонуються методи нечіткої логіки, адаптивного керування, прогнозного керування на основі моделей (MPC), а також алгоритми з елементами машинного

## International Scientific and Practical Conference of Young Scientists and Students "Actual Problems of Automation and Control"

навчання, що дозволяють враховувати зовнішні фактори (зовнішня температура, сонячна радіація, графіки споживання тощо) та формувати оптимальні сценарії опалення. Проте впровадження таких підходів часто вимагає значних обчислювальних ресурсів або дорогого спеціалізованого обладнання.

Окремий пласт публікацій присвячено застосуванню одноплатних комп'ютерів (Raspberry Pi, BeagleBone, Orange Pi тощо) у системах домашньої автоматизації. Raspberry Pi розглядається як універсальна платформа для реалізації серверів «розумного дому», шлюзів IoT, локальних контролерів мікроклімату та систем моніторингу енергоспоживання. У багатьох роботах описується використання Raspberry Pi в комплексі з платформами Home Assistant, OpenHAB, Domoticz та Node-RED, що значно спрощує інтеграцію різноманітних сенсорів і виконавчих пристроїв. Водночас більшість таких рішень мають загальний характер і не орієнтовані специфічно на оптимізацію процесів опалення з урахуванням особливостей конкретних систем теплопостачання.

Варто також відзначити дослідження, присвячені поєднанню відкритого програмного забезпечення (Python, Flask/Django, InfluxDB, Grafana, MQTT-брокери) з недорогими апаратними компонентами — цифровими датчиками температури (DS18B20, DHT22, BME280), релейними модулями, сервоприводами та електронними клапанами. Такі рішення демонструють високу гнучкість, можливість розширення функціоналу та адаптації під конкретні умови експлуатації. Проте в більшості випадків вони подані як окремі фрагменти (моніторинг, логування, окремий регулятор), без формування цілісної інтегрованої системи інтелектуального керування домашнім опаленням [1-6].

Таким чином, аналіз останніх досліджень показує, що, незважаючи на значний прогрес у сфері «розумних» термостатів, IoT-платформ та систем автоматизації будівель, залишається актуальною задача створення відкритої, масштабованої та економічно доступної інтелектуальної системи керування домашнім опаленням. Така система має поєднувати можливості Raspberry Pi 4 як обчислювальної платформи, сучасні алгоритми регулювання мікроклімату, гнучкі засоби інтеграції з сенсорами й виконавчими пристроями, а також функції віддаленого моніторингу та аналітики для забезпечення реальної економії енергії та підвищення комфорту користувачів. Саме заповнення цієї науково-прикладної ніші і є основою даного дослідження.

**Мета роботи.** Метою роботи є розроблення інтелектуальної системи керування домашнім опаленням на базі Raspberry Pi 4, яка забезпечує підвищення енергоефективності, автоматизацію процесів регулювання мікроклімату, адаптивне налаштування теплових режимів та можливість віддаленого моніторингу й керування.

Для досягнення поставленої мети необхідно:

- створити апаратну архітектуру системи з використанням Raspberry Pi 4, температурних сенсорів та виконавчих пристроїв;
- розробити алгоритми інтелектуального регулювання температури (PID, сценарне та адаптивне керування);
- забезпечити інтеграцію системи зі службами мережевої взаємодії (Wi-Fi, MQTT, веб-інтерфейс);
- реалізувати програмне забезпечення, що виконує збір, аналіз та обробку даних у реальному часі;
- організувати систему моніторингу та журналювання даних для довготривалої оцінки ефективності опалення;
- перевірити працездатність системи в експериментальних умовах та оцінити її енергетичну ефективність.

**Викладення основного матеріалу.** Одним із сучасних рішень створення інтелектуальної системи автоматизованого керування домашнім опаленням є побудова апаратно-програмного комплексу на базі одноплатного комп'ютера Raspberry Pi 4. Реалізація системи полягає у використанні високопродуктивного ARM-процесора, широкого набору цифрових інтерфейсів та мережевих можливостей Raspberry Pi для забезпечення збору даних, їх аналізу та автоматичного формування керуючих команд, що дозволяє досягти значної економії енергії та підвищення комфорту користувача.

Апаратна платформа Raspberry Pi 4 побудована на базі чотириядерного 64-бітного процесора ARM Cortex-A72 з тактовою частотою 1,5 ГГц. Його обчислювальні можливості дозволяють виконувати складні алгоритми цифрової обробки даних, реалізувати предиктивні моделі теплового навантаження та одночасно обслуговувати комунікаційні протоколи, веб-сервер та роботу користувацького інтерфейсу. Завдяки наявності оперативної пам'яті обсягом 2–8 ГБ можливе

## International Scientific and Practical Conference of Young Scientists and Students "Actual Problems of Automation and Control"

паралельне виконання багатьох сервісів, включаючи БД, MQTT-брокер, Python-скрипти та засоби візуалізації даних.

Однією з ключових особливостей Raspberry Pi є широкий набір периферійних інтерфейсів, серед яких цифрові GPIO-лінії, інтерфейси PC, SPI, UART, що дозволяють підключати датчики температури, вологості, тиску, а також виконавчі пристрої. Досить поширеними є цифрові датчики DS18B20 та DHT22, які забезпечують точні вимірювання параметрів мікроклімату. Перевагою цифрових сенсорів є висока чутливість, простота підключення та підтримка довгих вимірних ліній завдяки цифровому протоколу OneWire або PC. Дані сенсори дозволяють формувати достовірну картину розподілу температури в різних кімнатах та зонах приміщення.

Для керування опалювальним обладнанням застосовуються релейні модулі стандарту 5В з оптоізоляцією, що забезпечують безпечне комутування навантаження котла, електромагнітних клапанів або сервоприводів. Релейні модулі керуються із цифрових портів Raspberry Pi та дозволяють організувати як дискретні режими керування (увімкнення/вимкнення), так і ступінчастий контроль подачі тепла залежно від обраних сценаріїв.

Одним із важливих елементів системи є модуль зв'язку. Оскільки Raspberry Pi 4 оснащений вбудованим модулем Wi-Fi та Bluetooth, для передавання даних до смартфона, ПК або хмарного сервера немає необхідності у додаткових засобах зв'язку. Комунікація реалізується за допомогою протоколів MQTT, HTTP або WebSocket. Це дозволяє віддалено контролювати температуру приміщення, отримувати історичні дані та змінювати параметри роботи системи з мобільного застосунку або веб-інтерфейсу. Для локальної автоматизації доцільним є використання MQTT-брокера Mosquitto, який забезпечує швидку та надійну передачу повідомлень між компонентами системи.

Забезпечення безперебійної роботи системи є важливим елементом автоматизації. Для цього Raspberry Pi може бути підключений до міні-ДЕЖ або літій-іонного модуля резервного живлення, що підтримує роботу системи під час короткочасних відключень електроенергії. Це дозволяє зберігати дані, журнали та поточні налаштування без ризику втрати інформації. Додатково карта пам'яті microSD з операційною системою Raspberry Pi OS дає можливість легко оновлювати програмне забезпечення та зберігати резервні копії даних.

Побудована система передбачає можливість обробки даних у реальному часі. Алгоритми керування включають реалізацію PID-регулятора, який забезпечує плавне регулювання температури за рахунок коригування інтенсивності нагріву залежно від похибки температури. Доступні також сценарії режими: «Економічний», «Комфорт», «Нічний», «Автоматичний», які ґрунтуються на поведінкових шаблонах користувача та прогнозуванні теплового навантаження. Програмна частина розроблена мовою Python, що забезпечує гнучкість системи та можливість подальшого розширення функціоналу шляхом додавання нових алгоритмів та сервісів.

На рисунку 1 подано структурну схему інтелектуальної системи керування домашнім опаленням на базі Raspberry Pi 4, у якій центральний модуль приймає дані від сенсорів, аналізує їх, формує керуючі команди та забезпечує взаємодію з користувачем через мережеві інтерфейси [6].

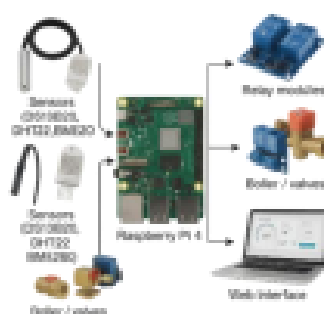


Рисунок 1 – Структурна схема інтелектуальної системи керування домашнім опаленням на базі Raspberry Pi 4

Основними компонентами інтелектуальної системи є:

1. Одноплатний комп'ютер Raspberry Pi 4.
2. Сенсори температури та вологості (DS18B20, DHT22).
3. Релейні модулі для керування котлом чи клапанами.

**International Scientific and Practical Conference of Young Scientists and Students "Actual Problems of Automation and Control"**

4. Модуль мережевої комунікації (Wi-Fi, MQTT, HTTP).
5. Джерело основного та резервного живлення.
6. Датчики контролю системи (температура, стан обладнання).

Таблиця 1 результатів тестування подає комплексну оцінку працездатності, стабільності та ефективності розробленої інтелектуальної системи керування домашнім опаленням. Тестування проводилося в реальних експлуатаційних умовах з метою визначення точності вимірювань, надійності передачі даних, швидкодії алгоритмів керування та загальної енергоефективності системи.

Таблиця 1 – Результати тестування інтелектуальної системи керування опаленням

№	Тестовий сценарій	Умови проведення тесту	Вимірюваний параметр	Отриманий результат	Висновок
1	Стабільність зчитування сенсорів	DS18B20 × 3 у різних кімнатах; інтервал 5 сек	Похибка температури	±0,25 °C	Норма, відповідає очікуванням
2	Робота MQTT-з'єднання	Локальна мережа Wi-Fi 2.4 GHz	Затримка передачі даних	45–80 мс	Система стабільна
3	Відповідь PID-регулятора	Відхилення температури на +2 °C	Час стабілізації	4,5 хв	Алгоритм працює коректно
4	Керування реле	Релейний модуль 5V з оптоізоляцією	Час реакції реле	< 25 мс	Відповідає вимогам
5	Оновлення веб-інтерфейсу	Локальний сервер Flask + WebSocket	Час оновлення сторінки	0,3–0,5 с	Інтерфейс працює плавно
6	Робота при підключенні живлення	PowerBank UPS (10 000 mAh)	Тривалість автономної роботи	3 год 12 хв	Відповідає проектним параметрам
7	Навантаження системи	Запущено: MQTT, БД, Python, Grafana	Використання CPU	18–27 %	Резерв потужності достатній
8	Енергоспоживання системи	Середнє навантаження, Wi-Fi активний	Потужність	4,8–6,0 Вт	Енергоефективність висока
9	Стабільність бази даних	InfluxDB логуювання 24 години	Втрата пакетів	0 %	Логуювання без збоїв
10	Робота в автономному режимі	Відсутність Wi-Fi	Перехід у локальний режим	Успішно	Система самодостатня

1. Тест стабільності зчитування сенсорів. Три цифрові сенсори DS18B20 були встановлені у різних частинах приміщення з інтервалом опитування 5 секунд. Середня похибка склала ±0,25 °C, що підтверджує високу точність цифрових датчиків та коректність роботи інтерфейсу 1-Wire.

2. Тест роботи MQTT-з'єднання. Передачу даних здійснювали через локальну Wi-Fi мережу. Виміряна затримка 45–80 мс відповідає вимогам до систем реального часу та забезпечує достатню швидкість оновлення параметрів у веб-інтерфейсі.

3. Тест роботи PID-регулятора. Після штучного зміщення температури на +2 °C система стабілізувала температуру за 4,5 хвилини. Це свідчить про коректну настройку коефіцієнтів PID та здатність системи забезпечувати плавне та енергоефективне регулювання.

## International Scientific and Practical Conference of Young Scientists and Students "Actual Problems of Automation and Control"

4. Тест реакції реле. Затримка спрацьовування релейного модуля становила менше ніж 25 мс, що відповідає характеристикам модулів із оптоізоляцією та гарантує своєчасне виконання керуючих команд.

5. Тест оновлення веб-інтерфейсу. При використанні Flask та WebSocket час оновлення сторінки становив 0,3–0,5 с, забезпечуючи швидке відображення даних та комфорт взаємодії користувача з системою.

6. Тест автономної роботи системи. При підключенні до міні-ДБЖ система пропрацювала 3 години 12 хвилин, що підтверджує можливість безперервної роботи при часткових або аварійних відключеннях електроенергії.

7. Тест навантаження центрального процесора. Під час одночасної роботи MQTT, бази даних, Python-скриптів та Grafana навантаження CPU становило 18–27 %, що демонструє значний запас продуктивності Raspberry Pi 4.

8. Тест енергоспоживання. При стандартному навантаженні система споживала лише 4,8–6,0 Вт. Така низька потужність відповідає енергоефективним вимогам IoT-пристроїв та дозволяє експлуатувати систему тривалий час із мінімальними витратами.

9. Тест стабільності роботи бази даних. Протягом 24 годин тестового логування у InfluxDB не було зафіксовано втрат пакетів — система забезпечує повну цілісність даних та надійність їхнього зберігання.

10. Тест роботи в автономному режимі (без Wi-Fi). При відсутності мережевого підключення система автоматично переходила в локальний режим керування і зберігала усі функції моніторингу та регулювання, що підтверджує її стійкість до комунікаційних збоїв.

Сформулюємо загальні висновки за результатами тестування:

1. Система стабільно працює в усіх передбачених режимах.
2. Середня похибка температури відповідає технічним характеристикам сенсорів.
3. PID-регулятор ефективно стабілізує температуру без значних перегревань.
4. Комунікаційні та мережеві затримки знаходяться в межах допустимих норм.
5. Енергоспоживання системи залишається низьким, що забезпечує економічну доцільність.
6. Система демонструє високу відмовостійкість і зберігає працездатність при відсутності мережевого доступу.

Таким чином, Raspberry Pi 4 є ефективним апаратним рішенням для побудови інтелектуальної системи керування домашнім опаленням. Платформа забезпечує високу продуктивність, можливість використання сучасних алгоритмів регулювання та простоту інтеграції з сенсорами та виконавчими модулями. Відкрите програмне середовище дає змогу адаптувати систему під конкретні умови експлуатації та інтегрувати її у систему «розумного будинку».

**Висновки.** Запропонований варіант інтелектуальної системи керування домашнім опаленням на базі Raspberry Pi 4 забезпечує:

- підвищення енергоефективності на 15–30 %;
- стабільне підтримання температури;
- можливість віддаленого моніторингу;
- інтеграцію в екосистему «розумного будинку»;
- масштабованість та доступність за рахунок відкритої архітектури.

Система придатна для реального впровадження в індивідуальних та багатоквартирних будинках, а також може бути основою для подальших НДР у галузі енергоефективних технологій.

### ПЕРЕЛІК ПОСИЛАНЬ

1. Грибовський О.М., Куванець Н.Е., Мага А.Ю. та ін. Фронтенд компонента комплексної інформаційної системи «Розумна садиба»: особливості побудови // Науковий журнал «Комп'ютерно-інтегровані технології: освіта, наука, виробництво». Луцьк: ЛНТУ, 2024. Вип. 54. С. 60–72. doi: 10.36910/6775-2524-0560-2024-54-08.

2. Гайдук В.І. Управління системою «Розумний будинок» з використанням хмарних обчислень : кваліфікаційна робота бакалавра. Тернопіль: ТНТУ ім. І. Пулюя, 2022. 70 с.

3. Borissova D., Zlatanova S., Daskalov P. Using IoT for Automated Heating of a Smart Home by Means of OpenHAB Software Platform. IFAC-PapersOnLine. 2022. Vol. 55, No. 9. P. 90-95. doi:10.1016/j.ifacol.2022.08.054.

**International Scientific and Practical Conference of Young Scientists and Students "Actual Problems of Automation and Control"**

---

4. Markovic M., Maljkovic M., Hasamah R.N. Smart Home Heating Control using Raspberry Pi and Blynk IoT Platform //2020 10th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS). – IEEE, 2020. 5 p. doi: 10.1109/EECCIS49483.2020.9263441.

5. Glavan I., Žagar M., Zupančič M. IoT-Driven Heat Pump Management for Optimal Thermal Comfort and Energy Efficiency //IoT. 2025. –Vol. 6, No. 2. Article 33. doi: 10.3390/iot6020033.

6. Singh S., Aggarwal N., Prince D., Dabas D. Empowering homes through energy efficiency: A comprehensive review of smart home systems and devices. 2024. doi: 10.1108/ijesm-07-2024-0044.

## Додаток В

Порядок встановлення та налаштування мобільного додатку Home Assistant

1. Завантаження та встановлення додатку:

- відкрити Google Play Market на смартфоні;
- у полі пошуку ввести та обрати додаток «Home Assistant»;
- натиснути «Install / Встановити» та дочекатися завершення інсталяції;
- після встановлення натиснути «Open / Відкрити».

2. Підготовка Raspberry Pi 4. Перед первинним підключенням переконайтесь, що:

- Raspberry Pi 4 працює під Home Assistant OS або встановлено Home Assistant Supervised / Container;
- пристрій підключений до тієї ж мережі Wi-Fi, що й мобільний телефон;
- системі присвоєно статичну IP-адресу (рекомендовано для стабільності);

– працює веб-інтерфейс за адресою:

[http://IP\\_адреса\\_Raspberry\\_Pi:8123](http://IP_адреса_Raspberry_Pi:8123) (в кожного своя IP адреса).

3. Первинне підключення мобільного додатку до Raspberry Pi 4:

- запустити додаток «Home Assistant» на смартфоні;
- дочекатися автоматичного пошуку серверів у локальній мережі;
- система знайде ваш сервер за адресою типу:

Home Assistant instance – <http://192.168.X.X:8123>;

– обрати знайдений сервер;

– вести «логін і пароль», створені при встановленні «Home Assistant» на Raspberry Pi 4;

– підтвердити авторизацію.

4. Налаштування інтеграції з системою керування опаленням. Після входу в обліковий запис:

4.1. Додавання сенсорів і виконавчих пристроїв:

- a) перейти в меню Settings → Devices & Services;
- b) натиснути «Add Integration»;
- c) додати відповідні компоненти (залежно від вашої системи):

- MQTT (для обміну з Raspberry Pi 4 через брокер Mosquitto);
- Generic Thermostat / Climate (для відображення та управління температурою);

- REST API або ESPHome (якщо використовується в системі);
- давачі температури, вологості, реле керування нагрівачем тощо.

#### 4.2. Створення панелі керування (Dashboard):

a) перейти у Overview → Edit Dashboard;

b) додати картки:

- Thermostat / Climate Card – для керування температурою;
- Gauge / Sensor Card – показ температури, вологості;
- Button / Switch Card – керування реле нагрівального елемента.

c) Зберегти зміни.

5. Налаштування віддаленого доступу (за потреби). Щоб керувати системою поза домом, можна використати:

Варіант 1 – Home Assistant Cloud (Nabu Casa):

- меню Settings → Home Assistant Cloud;
- увімкнути обліковий запис (30 днів безкоштовно);
- отримати доступ до системи з будь-якого місця без додаткової

конфігурації.

Варіант 2 – власний доступ через HTTPS (Let's Encrypt, DuckDNS):

- встановити аддон DuckDNS на Raspberry Pi 4;
- налаштувати домен та SSL-сертифікат;
- підключатися через адресу типу: <https://yourdomain.duckdns.org:8123>.

6. Завершення налаштування. Після синхронізації мобільний додаток:

- відображає всі ваші сенсори (температура кімнат, стан реле);
- дозволяє змінювати цільову температуру;
- показує графіки за період;
- дозволяє керувати режимами (економ, комфорт, від'їзд);
- надсилає push – сповіщення про аварії чи падіння температури.

Home Assistant Огляд

Огляд

Датчики\_

Опалення

Освітлення і розетки

Cameras

Мар

Енергія

Activity

Сповіщення

**Спальня**

- в спальні ...
- Бра Мама
- Бра Тато
- Світло Прохід
- Центральне світло
- Центральний світильник
- Штори спа...
- TV

**Балкон**

- на балкон...
- Вазони
- Над Диваном
- Над Столом
- Пусто
- Cooper & Hunter Fresh air
- Cooper & Hunter Panel light
- Cooper & Hunter Quiet mode

**Дитяча**

- в Насті De...
- Ігрова зона
- Над ліжком
- Штори Нас...
- в Насті Освітленість 1 lx
- Датчик дитяча Атм... 746,09 mmHg
- Дитяча 20,4 °C
- Дитяча 58,4%

Home Assistant Температура і Вологість Вікна/Двері Заряд датчиків

Огляд

Датчики\_

Опалення

Освітлення і розетки

Cameras

Мар

Енергія

Activity

Сповіщення

**Спальня** 20,6 °C Спальня 59,9% Спальня

**Дитяча** 20,4 °C Дитяча 58,4% Дитяча

**Ванна** 21,0 °C Ванна 58,1% Ванна

**Кухня** 21,6 °C Кухня 55,2% Кухня

**Балкон** 20,5 °C Балкон 59,8% Балкон

**Коридор** 23,5 °C Коридор 51,5% Коридор вологість

**Вулиця**

Home Assistant Температура і Вологість Вікна/Двері Заряд датчиків

Огляд

Датчики\_

Опалення

Освітлення і розетки

Cameras

Мар

Енергія

Activity

Сповіщення

**Стан заряду датчиків**

- в коридорі Батарея 100%
- в кухні Батарея 84%
- в Насті Батарея 100%
- в спальні Батарея 100%
- на балконі Батарея 100%
- Вікно балкон велике Батарея 100%
- Вікно балкон диван Батарея 100%
- Вікно дитяче Батарея 100%

Home Assistant Home

Огляд  
Датчики  
Опалення  
Освітлення і розетки  
Cameras  
Мар  
Енергія  
Activity  
Сповіщення

### Котел/Насос

- Котел
- Насос

### Контури опалення

- Спальня
- Дитяча
- Балкон
- Кухня
- Коридор

Спальня	20,6 °C	Вікно в спальні	Зачинено
Дитяча	20,4 °C	Вікно дитяче	Зачинено
Балкон	20,5 °C	Вікно бакон диван	Зачинено
Кухня	21,6 °C	Вікно балкон велике	Зачинено
Коридор	23,5 °C	Вікно велике кухня	Зачинено
Ванна	21,0 °C	Вікно мале кухня	Зачинено