

Міністерство освіти і науки України
Харківський національний університет імені В.Н. Каразіна
Факультет комп'ютерних наук
Кафедра інтелектуальних програмних систем і технологій

Рекомендовано до захисту:
протокол засідання кафедри
№ ____ від _____.____.2025 р.
Завідувач кафедри ІПСІТ
_____ Олешко О.І.
(підпис) (прізвище та ініціали)

Пояснювальна записка

до дипломної роботи бакалавра

На тему «Розробка автоматизованої системи управління водяним опаленням
житлового будинку на основі ESP32»

Захищено на засіданні ЕК № _____
протокол № ____ від _____.____.2025 р.
Оцінка ____ / _____
Голова ЕК

(підпис) (прізвище та ініціали)

Виконав: студент 4 курсу, групи КС- 43
Спеціальності:

122 - Комп'ютерні науки

(шифр і назва напрямку підготовки, спеціальності)

Бедюх Ігор Олександрович

(прізвище, ім'я та по батькові, підпис)

Керівник ст.викл. Рало О. М

(ступень, звання, прізвище та ініціали, підпис)

Рецензент к.т.н., доцент Маций О. Б.

(ступень, звання, прізвище та ініціали, підпис)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра Інтелектуальних програмних систем і технологій
Рівень вищої освіти (освітньо-кваліфікаційний рівень) Бакалавр
Галузь знань: 12 Інформаційні технології
Спеціальність: 122 Комп'ютерні науки
Освітня програма: Комп'ютерні науки

ЗАТВЕРДЖУЮ

В.О. завідувача кафедри ІПСіТ


підпис Олег ОЛЕШКО
ім'я, прізвище

«___» _____ 20__ року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ (ПРОЄКТ)

Бедюх Ігор Олександрович

(прізвище, ім'я, по батькові студента)

1. Тема роботи Розробка автоматизованої системи управління водяним опаленням житлового будинку на основі ESP32

керівник роботи Рало Олександр Миколайович, старший викладач комп'ютерних систем та робототехніки

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

затверджені наказом по університету від “16” квітня 2025 року № 4101-5/962

2. Строк подання студентом роботи 2 червня 2025

3. Перелік питань, які потрібно розробити:

1. Аналіз сучасних технологій автоматизації опалення

2. Обґрунтування вибору ESP32-WROOM

3. Проєктування архітектури системи

4. Розробка алгоритмів управління

5. Програмне забезпечення та реалізація

6. Інтеграція системи в житловий будинок

7. Оцінка ефективності та тестування

8. Фінальні висновки та оптимізація

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Аналіз інформаційних джерел щодо автоматизованих систем управління опаленням.	Січень - лютий
2	Обґрунтування теми роботи та вибір платформи для проектування системи	Лютий
3	Аналіз існуючих технологій автоматизації водяного опалення	Лютий
4	Визначення складових системи та їх особливостей	Березень
5	Створення архітектурного рішення для керування водяним опаленням	Березень
6	Розробка програмного забезпечення для мікроконтролера	Березень-квітень
7	Аналіз та оцінка ефективності зробленої системи	Квітень
8	Підготовка пояснювальної записки	Травень
9	Підготувати демонстраційний матеріал для захисту	Травень

5. Дата видачі завдання _____ 23 січня 2025 _____

Студент

I.O. Бедюх

ініціали, прізвище



підпис

Керівник роботи

O.M. Рало

ініціали, прізвище



підпис

АНОТАЦІЯ

Пояснювальна записка містить 52 сторінки основного тексту, складається з трьох розділів, містить 16 рисунків, 10 таблиць, 1 додаток та 25 джерел згідно з переліком посилань. У роботі представлено структуру, принципи функціонування та результати моделювання навчальної автоматизованої системи керування водяним опаленням на базі мікроконтролера ESP32.

Метою роботи є розробка навчальної моделі автоматизованої системи, яка дозволяє імітувати функціонування водяного опалення з урахуванням адаптивного керування.

Для реалізації поставленої мети використано мікроконтролер ESP32, сенсор температури DS18B20, OLED-дисплей SSD1306, інтерфейс OneWire, модуль EEPROM, програмне середовище Arduino IDE та інструмент віртуального моделювання Wokwi. Дослідження проведено методом функціонального моделювання алгоритмів.

У результаті дослідження створено повнофункціональну програмну модель, яка успішно реалізує автоматичний і ручний режим управління температурою, відображає актуальні параметри на дисплеї та дозволяє виконувати базову взаємодію з користувачем. Було проведено тестування в різних режимах, виявлено ключові закономірності температурної динаміки, підтверджено правильність роботи алгоритму гістерезису, а також зафіксовано стабільність системи в часі.

Результати можуть бути використані при вивченні принципів автоматизації та IoT, створенні прототипів енергоефективних систем опалення, а також у навчальному процесі технічних спеціальностей.

Ключові слова: СИСТЕМА ОПАЛЕННЯ, ESP32, ДАТЧИК DS18B20, ГІСТЕРЕЗИС, МОДЕЛЮВАННЯ, РЕЛЕ, OLED-ДИСПЛЕЙ, WOKWI, АВТОМАТИЗАЦІЯ, АЛГОРИТМ, ТЕМПЕРАТУРА, НАВЧАЛЬНА МОДЕЛЬ, C++, КЕРУВАННЯ.

ABSTRACT

The explanatory note contains 52 pages of main text, consists of three chapters, includes 16 figures, 10 tables, 1 appendices, and 25 sources according to the list of references. The work presents the structure, operating principles, and modeling results of an educational automated water heating control system based on the ESP32 microcontroller.

The aim of the work is to develop an educational model of an automated system that simulates the operation of water heating with adaptive control capabilities.

To achieve this goal, the following components and tools were used: ESP32 microcontroller, DS18B20 temperature sensor, SSD1306 OLED display, OneWire interface, EEPROM memory module, Arduino IDE software environment, and the Wokwi virtual simulation platform. The study was conducted using a method of functional algorithm modeling.

As a result of the research, a fully functional software model was created, which successfully implements both automatic and manual temperature control modes, displays current parameters on the screen, and allows basic user interaction. The system was tested under various conditions, key temperature dynamics were identified, the correctness of the hysteresis algorithm was confirmed, and the system's stability over time was recorded.

The results of this work can be used for studying automation principles, IoT systems, prototyping energy-efficient domestic heating solutions, and as part of the educational process in technical university departments.

Keywords: HEATING SYSTEM, ESP32, DS18B20 SENSOR, HYSTERESIS, SIMULATION, RELAY, OLED DISPLAY, WOKWI, AUTOMATION, ALGORITHM, TEMPERATURE, EDUCATIONAL MODEL, C++, CONTROL.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	5
ВСТУП	6
1. АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО АВТОМАТИЗАЦІЇ СИСТЕМ ОПАЛЕННЯ.....	10
1.1. Технологічний аналіз сучасних систем водяного опалення та засобів автоматизації	10
1.2. Обґрунтування доцільності створення автоматизованої системи на базі ESP32	17
1.3. Вимоги до архітектури майбутньої системи.....	21
1.4 Обґрунтування вибору ESP32-WROOM як основного контролера	22
1.5. Висновки та постановка завдань для реалізації системи.....	26
2. РОЗРОБКА ТА ОПТИМІЗАЦІЯ АРХІТЕКТУРИ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ ВОДЯНИМ ОПАЛЕННЯМ.....	29
2.1. Аналіз та формування функціональної архітектури навчальної моделі	29
2.2. Розробка та обґрунтування алгоритмів системи	43
2.3. Моделювання роботи системи та аналіз результатів	48
3. АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ ТА ПЕРСПЕКТИВИ ПОДАЛЬШОГО РОЗВИТКУ СИСТЕМИ.....	53
ВИСНОВКИ.....	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	58
ДОДАТОК А. ПРИКЛАДИ ВИХІДНОГО КОДУ	61

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ESP32 — мікроконтролер з вбудованим Wi-Fi і Bluetooth
GPIO — General Purpose Input/Output
DS18B20 — цифровий датчик температури
OLED — Organic Light Emitting Diode
SSD1306 — контролер OLED-дисплея
I2C — Inter-Integrated Circuit OneWire
EEPROM — Electrically Erasable Programmable Read-Only Memory
Wi-Fi — Wireless Fidelity (бездротова передача даних)
Wokwi — середовище віртуального моделювання
MQTT — Message Queuing Telemetry Transport Гістерезис
(ΔT_{hys}) — діапазон між температурами вмикання/вимикання
 T_{set} — задане значення температури
 T_{on} — температура вмикання нагріву
 T_{off} — температура вимикання нагріву
PWM — Pulse Width Modulation
LCD — Liquid Crystal Display
IP-адреса — мережевий ідентифікатор пристрою
ЦАП — цифрово-аналоговий перетворювач
АЦП — аналогово-цифровий перетворювач
НПД — навчально-проектне дослідження

ВСТУП

Сучасний етап розвитку техніки та інтелектуальних систем керування характеризується зростаючим попитом на автоматизовані рішення для забезпечення енергоефективності, зниження експлуатаційних витрат та підвищення комфорту життєдіяльності. Особливо актуальними є дослідження в галузі автоматизації систем водяного опалення, що відіграють ключову роль у створенні сприятливого мікроклімату в побутових, адміністративних та виробничих приміщеннях. У контексті постійного зростання вартості енергоресурсів, вимог до екологічності та енергоефективності, системи керування опаленням мають задовольняти нові технічні стандарти, забезпечуючи адаптивну взаємодію з користувачем, оптимізацію витрат, дистанційне управління та простоту обслуговування.

На сьогодні у світі широко використовуються як комерційні системи типу “розумний дім” із функціями віддаленого контролю, так і окремі модулі з локальним керуванням температурою. Проте значна частина таких систем орієнтована на закриті апаратно-програмні рішення, які є складними у налаштуванні, дорогими у впровадженні та недостатньо гнучкими з погляду адаптації до індивідуальних потреб користувача або освітнього середовища. Серед найпоширеніших реалізацій зустрічаються модулі на базі Arduino, Raspberry Pi, контролерів ESP8266/ESP32 з використанням Wi-Fi, MQTT, мобільних додатків для моніторингу температури та керування обігрівом. Незважаючи на доступність окремих рішень, багато з них вимагають значних зусиль для адаптації до специфіки навчального процесу чи демонстрації принципів адаптивного керування.

Водночас в освітньому середовищі спостерігається гостра потреба у створенні простих і доступних моделей, які дозволяють візуалізувати основи автоматизованого керування, застосовувати знання в реальних умовах, розвивати навички роботи з мікроконтролерами, датчиками, інтерфейсами введення-виведення та мережевими протоколами. Така модель повинна

відповідати технічним вимогам, але при цьому бути достатньо простою для відтворення в лабораторних умовах або у віртуальних симуляторах. Проблема, яку вирішує ця робота, полягає в необхідності створення навчальної автоматизованої системи керування водяним опаленням з функціями адаптивного регулювання, та візуалізації даних, що дозволяє досліджувати алгоритми керування, особливості роботи сенсорів і актуаторів, основи мережевої взаємодії.

Вивчення сучасної літератури, технічних рішень та наукових публікацій у цій сфері дозволяє стверджувати, що наявні реалізації або надто складні для інтеграції в освітній процес, або не дозволяють гнучко модифікувати параметри системи для імітації адаптивного керування. Значна частина з них не передбачає одночасне використання сенсорного зчитування, локального та мережевого керування, а також виведення даних в реальному часі. У цьому контексті постає необхідність створення гнучкої архітектури системи, яка демонструє ключові принципи автоматизації: використання петлі зворотного зв'язку, реалізація гістерезису, програмна обробка сигналів, керування реле на основі логічних умов, забезпечення візуального контролю через дисплей та доступ до системи через Wi-Fi.

Запропонована в роботі навчальна модель є прикладом такої архітектури. Вона базується на мікроконтролері ESP32, що дозволяє реалізувати всі необхідні функції без додаткових модулів. Система включає цифровий температурний сенсор DS18B20, релейний модуль, потенціометр, OLED-дисплей з інтерфейсом I2C, кнопку режиму, перемикач, EEPROM для збереження налаштувань. Важливим компонентом роботи є також використання середовища емуляції Wokwi, яке дозволяє протестувати та проаналізувати роботу системи без необхідності у фізичному обладнанні, що суттєво підвищує доступність моделі для навчального застосування.

Новизна цієї роботи полягає в комплексному підході до розробки навчальної автоматизованої системи з урахуванням адаптивних принципів керування, доступності апаратної реалізації та інтеграції у віртуальне

середовище. На відміну від типових проектів на базі Arduino чи ESP8266, ця система поєднує відразу декілька рівнів керування: локальний та автоматичний, що робить її придатною для експериментів з алгоритмами керування. Особливо актуальним є поєднання можливостей відображення даних у реальному часі, адаптація параметрів керування на рівні користувача та підтримка стандартних комунікаційних інтерфейсів.

Таким чином, актуальність даної теми зумовлена зростаючим попитом на енергоефективні автоматизовані системи, потребою в інтерактивному навчальному контенті, необхідністю апробації сучасних підходів до регулювання та оптимізації температурних параметрів, а також бажанням інтегрувати елементи “розумного дому” у технічну освіту. У результаті роботи реалізовано систему, яка не лише виконує свої функціональні задачі, але й служить платформою для подальших досліджень, розширень та використання у реальних проектах. Розроблений підхід має потенціал до масштабування, включення додаткових сенсорів, підключення до мобільних застосунків, а також може стати основою для побудови повноцінної інтелектуальної системи керування опаленням.

Враховуючи обмеження традиційних рішень, постійне зростання цін на енергоносії та доступність компактних цифрових платформ з підтримкою бездротового зв’язку, запропонований напрям є актуальним і практично значущим. Робота спрямована на створення прототипу інтелектуальної системи локального регулювання опалення з можливістю її масштабування, що є актуальним як для нового будівництва, так і для модернізації існуючих об’єктів.

Предметом даного дослідження виступає система водяного опалення житлових будинків як сукупність взаємопов’язаних технічних засобів, які забезпечують генерацію, транспортування та регулювання подачі тепла в приміщення. До її структурних елементів належать теплогенератор, трубопровідна мережа, циркуляційний насос, регулювальні та запірні вузли, а також кінцеві нагрівальні прилади. Система функціонує в умовах змінних

теплових навантажень, що вимагає динамічної адаптації до зовнішніх і внутрішніх чинників.

Об'єкт дослідження є процес автоматизованого керування температурними параметрами системи водяного опалення на основі сенсорних даних та локальної логіки регулювання. Основну увагу приділено розробці архітектури керування, вибору апаратної платформи, а також формуванню алгоритмів, здатних забезпечити стабільність, гнучкість і енергоефективність у режимі реального часу.

Метою дослідження є підвищення керованості та енергоефективності системи водяного опалення за рахунок впровадження програмно-апаратного комплексу на базі мікроконтролера ESP32. Досягнення цієї мети вимагає системного підходу до технічного обґрунтування архітектури, адаптації системи до змін мікроклімату та інтеграції з інтерфейсами розумного дому.

1. АНАЛІЗ СУЧАСНИХ ПІДХОДІВ ДО АВТОМАТИЗАЦІЇ СИСТЕМ ОПАЛЕННЯ

1.1. Технологічний аналіз сучасних систем водяного опалення та засобів автоматизації

Системи водяного опалення є базовим інженерним компонентом житлових будівель в Україні, забезпечуючи сталий температурний режим у холодний період року. Згідно з нормативами [1], їх призначенням є компенсація тепловтрат приміщення та підтримання параметрів мікроклімату на рівні санітарно-гігієнічних норм. Незважаючи на високу інерційність і здатність до рівномірного обігріву, традиційні схеми опалення часто залишаються енергетично неефективними через ручне регулювання, недосконале балансування та низьку гнучкість у режимах роботи.

Сучасні тенденції у сфері житлового будівництва передбачають жорсткі вимоги до зменшення витрат енергії, інтеграцію засобів цифрового контролю, а також підвищення експлуатаційної надійності. Як зазначено в [1], система опалення повинна забезпечувати безпечну та економну експлуатацію, відповідати вимогам до шумо- і пожежозахисту, а також створювати умови для індивідуального теплозабезпечення. Водночас нормативна база передбачає можливість впровадження локальних автоматизованих рішень на рівні окремих квартир або будинків, що посилює актуальність дослідження в умовах модернізації житлового фонду.

Системи управління водяним опаленням у житлових будинках відіграють ключову роль у забезпеченні ефективного функціонування теплопостачання, регулюванні температурного режиму та досягненні енергетичної оптимізації. Відповідно до положень [1], обов'язковим є забезпечення сталого мікроклімату в приміщеннях, що безпосередньо залежить від технічної реалізації системи керування.

Класифікація систем управління опаленням здійснюється за кількома основними ознаками (Таблиця 1.1)

Таблиця 1.1 – Класифікація систем управління водяним опаленням

Критерій класифікації	Тип системи	Характеристика
Рівень автоматизації	Ручна	Керування вручну, низька точність, висока залежність від користувача
	Автоматизована	Локальні кімнатні термостати, обмежена гнучкість
	Інтелектуальна (адаптивна)	Алгоритми на основі зовнішніх і внутрішніх даних, динамічна адаптація
Спосіб передачі сигналу	Дротова	Фізичні з'єднання, висока надійність, складність монтажу
	Бездротова	Використання Wi-Fi, Zigbee, простота встановлення, гнучкість конфігурації
Тип централізації	Централізована	Єдине керування всіма зонами, проста реалізація
	Децентралізована	Індивідуальне керування кожною зоною, підвищена енергоефективність
Джерело теплопостачання	Централізоване	Тепло надходить від зовнішніх джерел, обмежене індивідуальне регулювання
	Автономне	Повний контроль над процесом нагріву, підтримка автоматизації

У практиці проектування та експлуатації житлових будинків системи управління водяним опаленням класифікуються за різними ознаками, що визначають їхню ефективність, гнучкість та адаптивність до умов

експлуатації. Згідно з ДБН В.2.5-67:2013, вибір структури системи управління обумовлюється архітектурними особливостями будівлі та вимогами користувачів до енергоефективності та комфорту [1].

Дослідники, такі як Ткачук А. Я., Зайченко Є. С., Потапов В. А. та Цепелєв А. П., у своїй праці "Системи опалення: проектування та експлуатація" [2], розглядають технічні аспекти реалізації управління, зокрема, вплив типу теплопостачання та способу циркуляції теплоносія на вибір системи управління. Вони підкреслюють, що автономні системи з примусовою циркуляцією забезпечують більшу гнучкість у регулюванні температурного режиму, що особливо актуально для сучасних житлових будинків.

Сучасні огляди, наприклад, стаття "Поетапна автоматизація системи опалення" на сайті Adamson.ua [3], акцентують увагу на впровадженні зонального регулювання та погодозалежної автоматики як ключових елементів підвищення енергоефективності систем опалення. Зональне регулювання дозволяє адаптувати температуру в окремих приміщеннях відповідно до їхнього призначення та фактичного використання, що сприяє зниженню енергоспоживання.

Таким чином, класифікація систем управління водяним опаленням базується на комплексному аналізі технічних характеристик, рівня автоматизації, способу реалізації регулювання та джерела теплопостачання. Вибір оптимальної структури системи управління повинен враховувати не лише технічні параметри будівлі, але й потреби користувачів у комфорті та енергоефективності.

У контексті сучасних вимог до енергоефективності, сталого розвитку та підвищення комфорту мешканців житлових будинків, особливого значення набувають інтелектуальні системи автоматизації опалення. Вони передбачають перехід від ручного або частково автоматизованого управління до гнучких цифрових технологій, здатних адаптуватися до змін навколишнього середовища та індивідуальних параметрів користувача. Згідно

з положеннями [1], проєктування опалювальних систем повинно включати рішення щодо автоматизованого регулювання теплового режиму з метою забезпечення нормативного мікроклімату та зменшення витрат теплової енергії.

Фахова література акцентує на важливості розвитку саме адаптивних рішень. У класичному підході, розглянутому Ткачуком А. Я. та співавт. [2], основна увага приділяється технічним аспектам побудови систем опалення, однак уже там вказується на необхідність регулювання потужності тепловіддачі залежно від зовнішніх температур. Сучасні підходи значно розширюють ці засади за рахунок технологій, що забезпечують постійний зворотний зв'язок, здатність до самонавчання та управління через інтернет.

На практиці автоматизоване керування втілюється у вигляді інтегрованих систем на базі концепції розумного дому (Smart Home). Такі системи дозволяють централізовано або децентралізовано керувати всіма інженерними підсистемами будинку, включно з опаленням, за допомогою програмованих логічних контролерів, смартфонів або голосових асистентів. Відповідно до матеріалів компанії Adamson [3], впровадження зонального управління, погодозалежного регулювання та дистанційного доступу до термостатів дозволяє знизити витрати на опалення до 30% без втрати комфорту.

У цьому контексті актуальним стає аналіз принципів побудови сучасної цифрової архітектури, що лежить в основі таких адаптивних систем. Її концепція реалізується через інтеграцію IoT-пристроїв, хмарних сервісів і керувальних додатків у єдине інформаційне середовище. З урахуванням цього, доцільно розглянути узагальнену модель функціонування IoT-системи автоматизації інженерних процесів, представлену на рисунку 2.1.

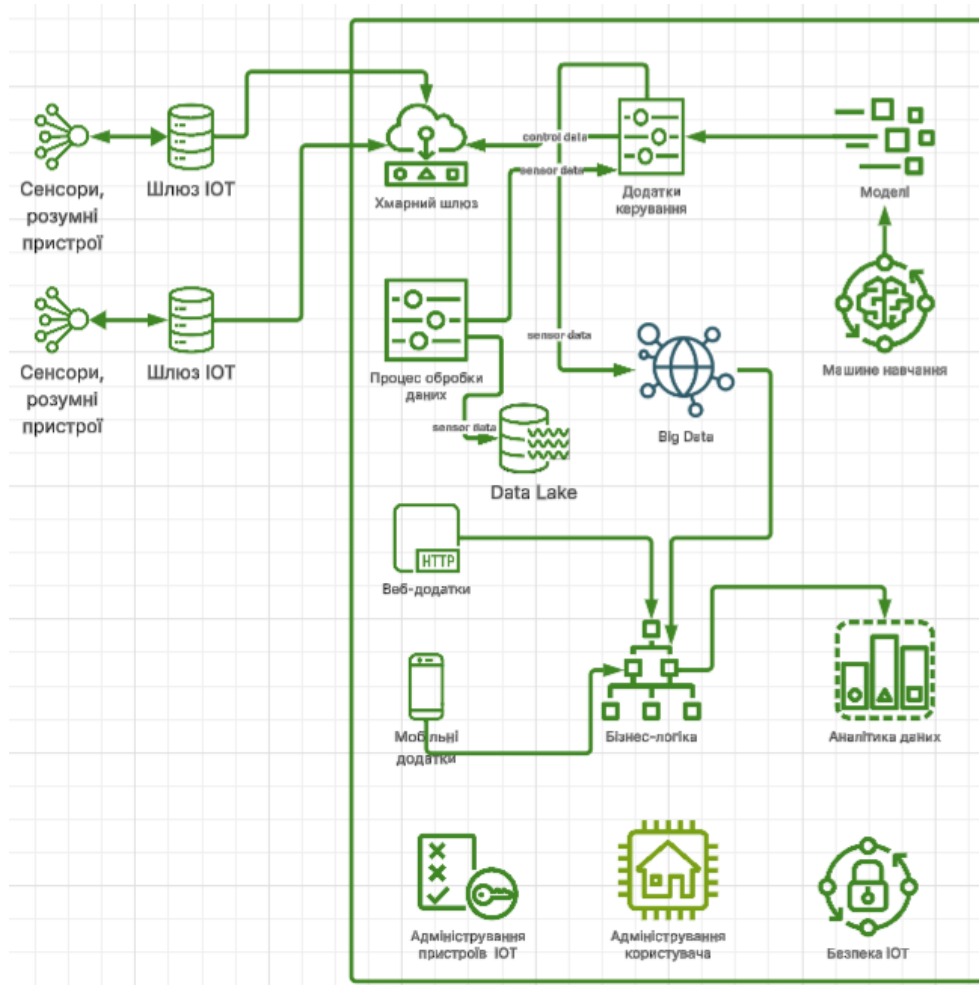


Рисунок 1.1 – Типова архітектура IoT-системи керування інженерними процесами

На першому рівні функціонування системи розміщені сенсори та розумні пристрої, які фіксують параметри мікроклімату (температура, вологість, рух тощо). Через IoT-шлюзи дані надходять до хмарного середовища, де відбувається їх обробка, агрегація в Data Lake та передача до аналітичних підсистем. Блок Big Data забезпечує зберігання великих масивів даних, а модуль машинного навчання — формування предиктивних моделей на їх основі. Керувальні команди генеруються відповідно до аналізу зібраних даних і повертаються до пристроїв через додатки керування.

Система інтегрується з мобільними та веб-застосунками, що дозволяє здійснювати дистанційний моніторинг, налаштування параметрів опалення, перегляд аналітики та керування профілями користувачів. Передбачено також

модулі для адміністрування пристроїв, безпеки IoT і формування бізнес-логіки взаємодії компонентів.

У контексті опалення така архітектура реалізує ключові функції: погодозалежне регулювання, зональне управління, діагностику стану обладнання, прогнозування споживання тепла та оптимізацію роботи виконавчих механізмів. Відкритість протоколів обміну (MQTT, HTTP, Zigbee, Wi-Fi) дозволяє інтегрувати систему з компонентами різних виробників.

Сутність представленої архітектури полягає в багаторівневій взаємодії: від фізичного зчитування параметрів до аналітичного прогнозування, де кожен рівень — сенсорний, обчислювальний, аналітичний і прикладний — доповнює функціонал системи опалення. Така модель відповідає принципам масштабованості, гнучкості та безперервної адаптації до потреб користувача, що є основою для побудови сучасної автоматизованої інфраструктури в житловому секторі.

Додатково, технології Інтернету речей (IoT) забезпечують взаємодію між пристроями та збір телеметрії в реальному часі. Це дає змогу здійснювати точне, контекстуальне регулювання температурного режиму, а також реалізовувати автоматичну діагностику або прогнозування несправностей. Як зазначається в матеріалах [4], сучасні IoT-рішення базуються на протоколах Wi-Fi, Zigbee, Bluetooth або Z-Wave і дозволяють інтегрувати керування опаленням з іншими побутовими пристроями — наприклад, датчиками руху, освітлення або віконними сенсорами.

Таким чином, сучасні технології автоматизації опалення передбачають поєднання кількох складових: архітектури розумного дому, мережеских рішень IoT, бездротових систем обміну даними та розгалуженої сенсорної інфраструктури. Їх синергія забезпечує гнучке, адаптивне й енергоощадне керування мікрокліматом житлового простору, що є фундаментальною передумовою для побудови ефективної автоматизованої системи, розглядуваної у цій роботі.

Концепція розумного дому (Smart Home) охоплює комплекс технологічних рішень, що забезпечують автоматизоване та адаптивне керування інженерними системами будівлі — освітленням, безпекою, вентиляцією та опаленням — з використанням єдиного цифрового середовища. У межах цієї парадигми опалення розглядається не як автономна система, а як частина єдиної логіки управління комфортом, що інтегрується з датчиками руху, контролерами відкриття вікон, системами присутності та метеостанціями.

Згідно з аналітичними матеріалами компанії Adamson [3], одним із ключових принципів реалізації системи Smart Heating є зональне регулювання, що дозволяє користувачам встановлювати індивідуальні температурні режими для кожного приміщення, з урахуванням часу доби, режимів використання кімнати та її теплоізоляційних характеристик. Таке рішення базується на комбінації електронних термостатів, електроприводів клапанів та контролерів типу Siemens LOGO або Danfoss Link.

У контексті енергоефективності особливу увагу приділено погодозалежному управлінню, при якому інтенсивність нагріву змінюється відповідно до температури зовнішнього повітря. Подібні функції підтримуються сучасними регуляторами Danfoss ECL Comfort серії 210/310, які забезпечують автоматичну зміну температури теплоносія відповідно до кривих теплопостачання [3]. Це дозволяє уникнути перевитрат енергії в міжсезоння, коли традиційні системи опалення працюють із надмірною потужністю.

Іншим важливим аспектом реалізації розумного дому є централізоване керування через мобільні застосунки або веб-інтерфейси. Як зазначено в матеріалах [4], мобільний контроль дає змогу віддалено змінювати налаштування системи, отримувати дані про споживання енергії, виявляти несправності та отримувати push-повідомлення у разі відхилень від встановлених параметрів.

Реалізація Smart Home базується на стандартах сумісності між пристроями різних виробників. Це забезпечується використанням відкритих протоколів, зокрема Wi-Fi, Zigbee, Z-Wave, MQTT, що дозволяє інтегрувати термостати, датчики та контролери в єдину екосистему. Такий підхід забезпечує не лише масштабованість і модульність, а й підтримку принципу plug-and-play, що суттєво знижує бар'єри для впровадження систем у житлових умовах.

У підсумку, Smart Home у контексті опалення формує інфраструктуру, здатну до самостійного аналізу, адаптації та реакції на зміни середовища або поведінки мешканців. Це підвищує не лише енергоефективність системи, а й зручність її використання, роблячи такі рішення обґрунтованими для подальшого впровадження в побутових умовах.

1.2. Обґрунтування доцільності створення автоматизованої системи на базі ESP32

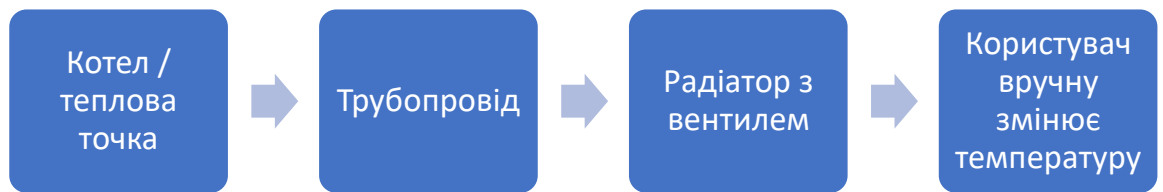
Проведений у попередньому підпункті аналіз свідчить про те, що традиційні централізовані та навіть автономні системи водяного опалення, попри свою поширеність, недостатньо відповідають сучасним вимогам до гнучкості, адаптивності та енергоефективності. Як зазначено у [1], вимоги до сучасного теплопостачання охоплюють не лише стабільність теплопередачі, а й динамічне регулювання потужності з урахуванням зовнішніх і внутрішніх умов. У роботах [2; 3] підкреслено, що у багатоквартирному житловому фонді України значна частина втрат тепла пов'язана з неефективним керуванням подачею теплоносія — відсутністю зонального регулювання та інерційністю застарілих систем.

Як було показано на рисунку 1.1, типова архітектура IoT-системи дозволяє реалізувати багаторівневий підхід до регулювання: сенсори фіксують параметри мікроклімату, передають дані до контролера, який аналізує умови та формує команди на виконавчі пристрої — термодіафрагми, реле або циркуляційні насоси. Така структура дозволяє створити адаптивне

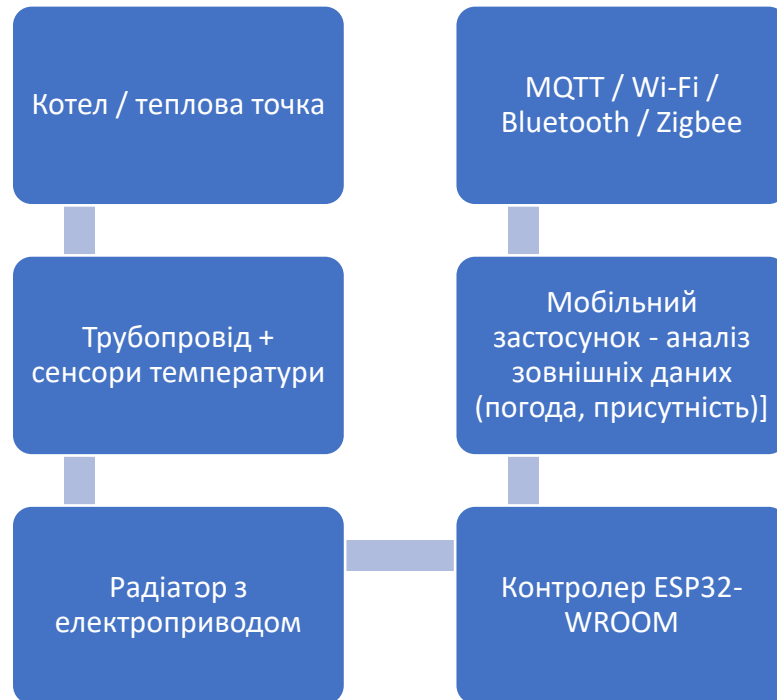
середовище управління температурою в режимі реального часу, мінімізуючи втрати енергії та забезпечуючи комфорт користувача.

Згідно з сучасними оглядами [4; 5], системи Smart Heating, які впроваджуються у сфері житлової автоматизації, забезпечують зниження витрат на опалення на 20–30% за рахунок зонального керування, погодозалежної логіки та дистанційного доступу через мобільні застосунки. Проте реалізація таких рішень вимагає не лише архітектурної перебудови, а й вибору відповідної апаратної платформи, здатної виконувати функції збору, аналізу й передачі даних при мінімальних енергетичних витратах.

Таким чином, основна технічна проблема полягає у відсутності в існуючих системах опалення універсального, енергоощадного, легко конфігурованого пристрою, здатного працювати з різними типами сенсорів, забезпечувати бездротову передачу даних, підтримувати відкриті протоколи (MQTT, HTTP) та взаємодіяти з мобільними чи веб-інтерфейсами. У наступних підпунктах буде здійснено обґрунтування вибору саме такої платформи на прикладі ESP32-WROOM, а також обговорено супутні компоненти системи, зокрема датчики, виконавчі пристрої та комунікаційні модулі (Рисунок 1.2).



(a)



(б)

Рисунок 1.2 – Порівняння архітектури традиційної системи керування опаленням та IoT-орієнтованого рішення на базі ESP32 (а) – традиційна система (б) - Сучасна IoT-орієнтована система

З огляду на структурну різноманітність та функціональні особливості, сучасні технології автоматизації опалення доцільно порівнювати за критеріями адаптивності, енергоефективності, складності впровадження, масштабованості та сумісності з іншими системами розумного дому. У межах цього дослідження було узагальнено результати аналітичного огляду [3; 4; 5] та практичних рекомендацій [1; 2], які дозволяють оцінити переваги й обмеження кожного з підходів.

Нижче наведено аналітичну таблицю 1.2, що характеризує ключові технології за низкою порівняльних параметрів

Таблиця 1.2 – Порівняння сучасних технологій автоматизації опалення

Технологія	Адаптивність	Енергоефективність	Складність впровадження	Масштабованість	Сумісність із системами Smart Home
Розумний дім (Smart Home)	Висока	Висока	Середня	Висока	Висока
ІоТ-платформи	Висока	Висока	Висока	Висока	Висока
Бездротові мережі	Середня	Висока	Низька	Висока	Висока
Сенсорні системи	Середня–висока	Середня–висока	Середня	Середня	Висока

Як видно з таблиці, найбільш універсальними у плані функціональності та подальшої інтеграції є Smart Home-підходи та ІоТ-системи, які забезпечують високу адаптивність і повноцінний зворотний зв'язок. З іншого боку, бездротові технології, хоч і не мають самостійної функції адаптації, є критично важливими для реалізації гнучкої інфраструктури системи без складного монтажу, що знижує витрати на інсталяцію та дозволяє поетапне розширення.

Сенсорні технології — фундамент будь-якої системи автоматизації — забезпечують достовірний потік даних, проте їхня ефективність залежить від точності, надійності та щільності покриття простору. Згідно з [5], лише правильна комбінація всіх наведених компонентів у єдиній архітектурі дає змогу реалізувати інтелектуальне керування, здатне до самонавчання, передбачення та мінімізації втручання з боку користувача.

Критичний аналіз засвідчує, що для створення ефективної системи автоматизації опалення доцільно поєднувати сенсорні технології з

бездротовими каналами зв'язку, підтримкою IoT-протоколів та централізованим логічним контролем у межах Smart Home-платформи. Такий підхід відповідає принципам гнучкості, енергозбереження, ергономіки та майбутньої інтеграції в розширені екосистеми автоматизації побуту.

1.3. Вимоги до архітектури майбутньої системи

Розробка ефективної автоматизованої системи керування водяним опаленням вимагає формування чітких архітектурних вимог, які відповідають технічним обмеженням та експлуатаційним очікуванням користувача. Виходячи з аналізу попереднього підпункту, така система повинна забезпечувати децентралізоване керування, масштабованість, низьке енергоспоживання, а також бездротову взаємодію з датчиками й виконавчими пристроями. Вона має підтримувати сучасні протоколи зв'язку (MQTT, HTTP), забезпечувати зональне й погодозалежне регулювання, а також мати можливість інтеграції з мобільними або веб-інтерфейсами.

Ключовими компонентами системи є сенсорна мережа, контролер, виконавчі пристрої, засоби зв'язку та логічні алгоритми управління. У таблиці 1.3 наведено деталізовані функціональні вимоги до кожної з підсистем.

Таблиця 1.3 – Основні архітектурні вимоги до підсистем автоматизованої системи опалення

Підсистема	Основні функції	Технічні вимоги
Сенсорна мережа	Збір даних про температуру, вологість, рух, відкриття вікон	Підтримка цифрових і аналогових датчиків; стабільна робота в побутових умовах

Продовження таблиці 1.3

Підсистема	Основні функції	Технічні вимоги
Контролер (ESP32)	Обробка даних, прийняття рішень, формування керуючих сигналів	Наявність Wi-Fi, GPIO, енергоощадний режим, підтримка MQTT/HTTP
Виконавчі пристрої	Реалізація команд: відкриття клапанів, вмикання насосів, реле	Сумісність з цифровими керуючими сигналами, захист від перевантаження
Комунікаційна мережа	Передача даних між компонентами, доступ до мобільного застосунку	Підтримка бездротових протоколів: Wi-Fi, Zigbee, MQTT
Логіка керування	Зональне керування, погодозалежне регулювання, таймери, сценарії	Локальне збереження конфігурацій, адаптивність, можливість оновлення логіки
Інтерфейс користувача	Контроль, моніторинг, перегляд даних, push- сповіщення	Мобільний або веб- інтерфейс, авторизація, історія споживання

Крім того, система повинна мати автономний режим роботи, тобто здатність підтримувати базові сценарії керування навіть у разі втрати зв'язку з сервером або хмарним сервісом. Для зручності експлуатації передбачається функція оновлення конфігурації без перепрошивки, з можливістю локального або віддаленого редагування параметрів.

1.4 Обґрунтування вибору ESP32-WROOM як основного контролера

У межах розроблення автоматизованої системи опалення особливу увагу необхідно приділити вибору апаратної платформи, яка забезпечує стійку

роботу сенсорної мережі, обробку даних у реальному часі, підтримку бездротових інтерфейсів та відповідність вимогам до енергоспоживання. Платформа виконує роль локального вузла керування, шлюзу або центрального процесора системи. Серед найпоширеніших у сфері автоматизації використовуються Arduino, ESP32 та Raspberry Pi, які значною мірою відрізняються між собою за архітектурою, функціональністю й інтеграційними можливостями.

Arduino — це мікроконтролерна платформа відкритого типу, базована на мікросхемах AVR. Вона широко поширена завдяки простоті програмування, низькому енергоспоживанню та великій спільноті розробників. Однак у базових моделях (Arduino Uno, Nano) відсутні вбудовані модулі бездротового зв'язку, а також обмежені обчислювальні ресурси. Для реалізації задач автоматизації опалення в мережевому середовищі Arduino вимагає додаткових рішень, зокрема підключення ESP8266 або інших Wi-Fi-модулів, що ускладнює архітектуру системи та знижує її ефективність у масштабованому виконанні [2].

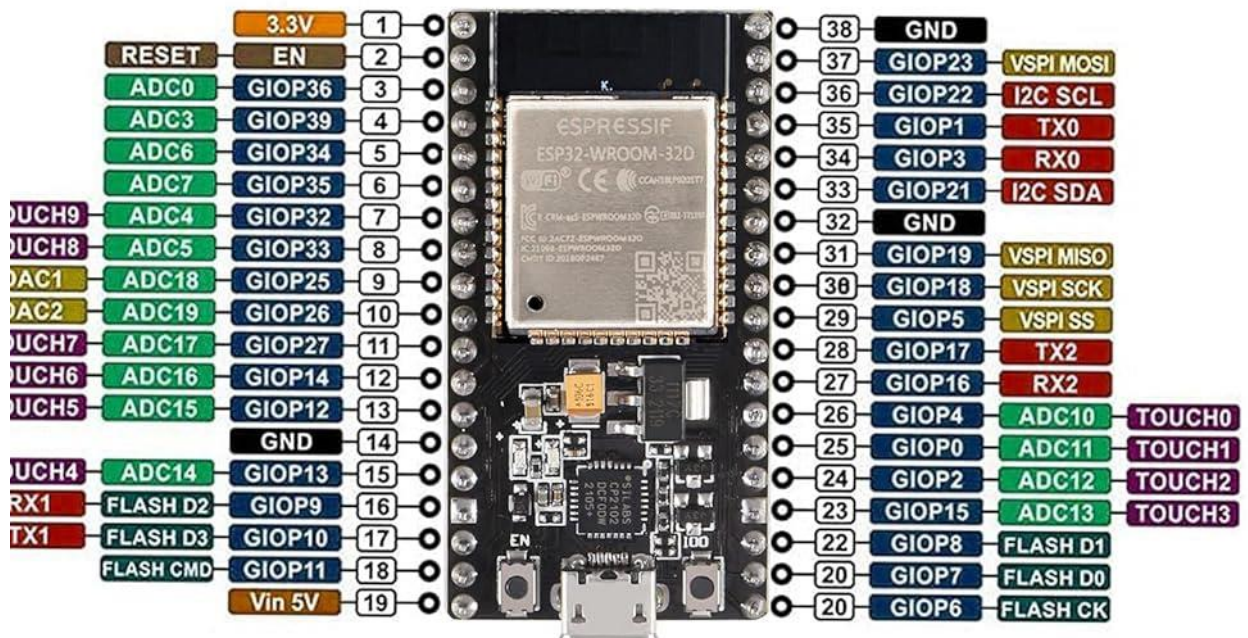
На відміну від цього, платформи ESP32 пропонують значно ширший функціональний потенціал. Вони поєднують двоядерний процесор з тактовою частотою до 240 МГц, вбудований Wi-Fi і Bluetooth, криптографічні прискорювачі, численні периферійні інтерфейси (I2C, SPI, UART, PWM, ADC, DAC) та можливість енергоефективної роботи в режимі очікування. За результатами практичного впровадження, описаного у [3], ESP32 виявляє себе як оптимальна платформа для систем інтелектуального керування опаленням, завдяки поєднанню автономності, потужності та гнучкості програмування.

Raspberry Pi, натомість, є повноцінним одноплатним комп'ютером, який функціонує під управлінням операційної системи на базі Linux. Завдяки цьому платформа підтримує запуск серверів, баз даних, інтерфейсів керування та навіть алгоритмів машинного навчання. Водночас її застосування в ролі сенсорного вузла є недоцільним через високе енергоспоживання, вимоги до охолодження та надлишковість обчислювальних ресурсів для задач

зчитування даних і базового регулювання. У статті [5] зазначено, що Raspberry Pi може ефективно використовуватись як локальний сервер або вузол інтеграції, але не як базовий контролер для розгалуженої сенсорної мережі.

Порівняльний аналіз дозволяє дійти висновку, що для побудови сучасної автоматизованої системи керування опаленням, яка базується на сенсорах, бездротових протоколах зв'язку та принципах IoT, платформа ESP32 має найбільш збалансоване поєднання функціональності, енергоефективності та технологічної зрілості. Саме тому в подальших розділах буде обґрунтовано вибір модуля ESP32-WROOM як базового елемента реалізації розроблюваної системи.

Мікроконтролер ESP32-WROOM (Рисунок 1.3.) має вбудований модуль Wi-Fi (802.11 b/g/n) та Bluetooth (BLE 4.2), двоядерний 32-бітний процесор Xtensa LX6 (до 240 МГц), до 520 КБ SRAM, підтримку зовнішньої flash-пам'яті, а також багатий набір периферії: SPI, I2C, UART, ADC, PWM, GPIO.



Його енергоспоживання в режимі глибокого сну становить менше 10 μ A, що дозволяє використовувати його навіть у пристроях з автономним живленням. Програмування здійснюється через Arduino IDE, PlatformIO або ESP-IDF, з великою спільнотою розробників і бібліотек (Таблиця 1.4.).

Таблиця 1.4 – Порівняння мікроконтролерів для використання в системах автоматизації опалення

Характеристика	Arduino Uno	Raspberry Pi (3/4)	ESP32-WROOM
Тип процесора	8-біт AVR (16 МГц)	64-біт ARM Cortex (1.2–1.5 ГГц)	32-біт Xtensa LX6 (до 240 МГц, 2 ядра)
ОЗУ	2 КБ	1–8 ГБ	до 520 КБ SRAM
Wi-Fi / Bluetooth	Ні / Ні	Так / Так	Так / Так (вбудовані модулі)
GPIO / ADC	14 цифрових, 6 аналогових	до 26 GPIO (через HAT)	34 GPIO, 18 ADC каналів
Енергоспоживання (стенд)	~45 мА	>400 мА	<10 мкА у deep sleep, ~50–80 мА у роботі
Вартість (орієнтовно)	\$7–10	\$40–60	\$5–9
Підтримка MQTT	Через бібліотеки	Повноцінна	Повноцінна (через ESP-IDF / PubSubClient)
Програмування	Arduino IDE	Linux, Python, Node.js	Arduino IDE, PlatformIO, ESP-IDF
Висновок	Обмежені можливості	Надлишкова для задач опалення	Оптимальний баланс функцій і ціни

ESP32 успішно застосовується в проектах інтелектуального керування освітленням, вентиляцією, поливом, охоронними системами, а також у численних проектах Smart Home з відкритим кодом, включаючи системи автоматизованого опалення, згідно з [3; 4]. Завдяки підтримці MQTT, цей контролер легко інтегрується в багаторівневі IoT-архітектури, що підтверджується також в огляді [5], де ESP32 використано як базовий вузол в енергоефективних системах опалення з локальним та віддаленим керуванням.

1.5. Висновки та постановка завдань для реалізації системи.

На підставі проведеного теоретичного аналізу було встановлено, що системи водяного опалення є однією з базових інженерних складових житлових будинків, і їх ефективне функціонування безпосередньо впливає на енергетичну сталість, комфорт мешканців та економічну доцільність експлуатації. У сучасних умовах домінуючою є тенденція до інтеграції інтелектуальних засобів керування, яка передбачає автоматизацію процесів регулювання температурного режиму з використанням принципів розумного дому, бездротових мереж та технологій Інтернету речей.

Класифікація існуючих систем управління дозволила виділити ключові типи архітектур — централізовані та децентралізовані — а також засоби реалізації керування: ручне, автоматизоване та інтелектуальне. Було показано, що ефективність автоматизованої системи значною мірою залежить від якості зворотного зв'язку, який реалізується через сенсорні технології, та здатності системи до адаптації в режимі реального часу.

У процесі аналізу сучасних апаратних платформ було встановлено, що найбільш оптимальним рішенням для реалізації цільової системи є використання контролера ESP32-WROOM. Він забезпечує необхідну обчислювальну потужність, має вбудовану підтримку Wi-Fi та Bluetooth, характеризується низьким енергоспоживанням і сумісний із більшістю протоколів обміну даними, що є критично важливим для інтеграції в інфраструктуру Smart Home. Застосування ESP32-WROOM дає змогу побудувати модульну, масштабовану й енергоефективну систему, яка відповідає сучасним вимогам до побутової автоматизації.

З огляду на вищезазначене, на наступному етапі роботи передбачається реалізація таких завдань (Таблиця 1.5.)

Таблиця. 1.5 – Постановка завдань для реалізації системи з перевагами та недоліками

Завдання	Переваги	Недоліки
Проектування архітектури системи з урахуванням принципів адаптивного управління	Гнучкість масштабування, підвищення адаптивності, можливість вибору між централізованою і децентралізованою архітектурою	Необхідність комплексного аналізу вимог до системи, потенційна складність реалізації
Розробка алгоритмів керування температурою із застосуванням сенсорних даних та логічних умов	Точне регулювання температури, можливість реалізації різних сценаріїв, економія енергії	Потреба в точному калібруванні сенсорів і валідації алгоритмів у змінних умовах
Моделювання структури системи в спеціалізованих середовищах (Tinkercad, Proteus або Fritzing)	Візуалізація взаємозв'язків, зниження ризику проектних помилок, економія часу на монтаж	Обмеження програм у моделюванні складних процесів, не завжди відповідає реальним умовам
Розробка програмного забезпечення для ESP32-WROOM з використанням локальних протоколів взаємодії з периферійними пристроями	Простота реалізації, мінімальна кількість з'єднань, стабільність роботи без залежності від мережі	Обмеженість функціоналу, відсутність можливості віддаленого моніторингу без додаткових модулів
Тестування, налагодження та оцінка ефективності реалізованої системи в умовах модельного середовища	Оцінка стабільності та ефективності, виявлення потенційних збоїв, підготовка до реального впровадження	Витрати часу на багаторазове налагодження, складність моделювання аварійних ситуацій

Таким чином, попередній аналітичний етап створив науково обґрунтовану основу для побудови автоматизованої системи керування

водяним опаленням, орієнтованої на сучасні технічні вимоги, гнучкість експлуатації та сталий розвиток житлової інфраструктури.

Проведене моделювання підтвердило правильність реалізації алгоритмів, дозволило виявити та усунути потенційні помилки на етапі проектування. Результати експериментального тестування свідчать про достатню точність вимірювання температури, стабільну роботу релейного керування та коректну реакцію на зміну зовнішніх умов. Система здатна функціонувати в умовах змінного навантаження, автоматично адаптуючи режим роботи до поточних параметрів. Отримані результати також демонструють потенціал для подальшого розширення функціональності, зокрема інтеграції мобільного застосунку, автоматичного регулювання температурних профілів залежно від часу доби або погодних умов та реалізації зворотного зв'язку через телеметрію. Отже, на підставі змодельованої поведінки можна зробити висновок, що система відповідає заявленим вимогам і є готовою до переходу на етап фізичної реалізації та дослідної експлуатації.

2. РОЗРОБКА ТА ОПТИМІЗАЦІЯ АРХІТЕКТУРИ АВТОМАТИЗОВАНОЇ СИСТЕМИ КЕРУВАННЯ ВОДЯНИМ ОПАЛЕННЯМ

2.1. Аналіз та формування функціональної архітектури навчальної моделі

Аналіз готових існуючих рішень, виявив, що більшість з них або мають складну архітектуру, яка важко піддається аналізу, або не надають необхідного рівня гнучкості для демонстрації адаптивних принципів керування. Виникає необхідність створення спрощеної, проте достатньо функціональної навчальної моделі на базі мікроконтролера ESP32 та онлайн-середовища емуляції Wokwi.

Основні вимоги (Таблиця 2.1.) до навчальної моделі автоматизованої системи керування водяним опаленням можна узагальнити так:

- Достовірне відображення роботи системи в реальному часі.
- Можливість демонстрації автоматичного та ручного режимів керування.
- Використання простих та доступних компонентів для навчання: датчик температури DS18B20, OLED-дисплей, реле, потенціометр, кнопки, перемикачі.

Таблиця 2.1 – Перелік функціональних вимог до системи, що проєктується

№	Вимога	Опис
1	Контроль температури	Автоматичне регулювання за температурними межами
2	Відображення інформації	OLED-дисплей для виводу параметрів
3	Ручне управління	Кнопка для зміни режимів роботи

Продовження таблиці 2.1

№	Вимога	Опис
4	Адаптивність	Можливість регулювання температурних порогів

Декомпозиція задачі дозволяє чітко визначити підзадачі та забезпечити логічну послідовність їх реалізації. Для цього було виділено наступні основні підзадачі (Таблиця 2.2):

Таблиця 2.2 – Декомпозиція задачі та взаємозв'язок підзадач у функціональній архітектурі навчальної моделі

Підзадача	Опис підзадачі	Залежності
Вимірювання температури	Використання датчика DS18B20 для визначення температури	—
Обробка сигналів та керування	Обробка даних датчика і включення/виключення реле	Вимірювання температури
Візуалізація даних	Виведення інформації на дисплей	Обробка сигналів

Під час реалізації навчальної моделі на платформі Wokwi виникли певні обмеження, що можуть вплинути на її ефективність. Основні з них наведено у таблиці 2.3:

Таблиця 2.3 – Основні обмеження навчальної моделі

Обмеження	Опис	Шляхи вирішення
Відсутність підтримки кирилиці на OLED-дисплеї	Неможливість використання української мови	Транслітерація
Обмеження емуляції реальних затримок	Нереалістичність деяких сценаріїв роботи системи	Використання умовних часових циклів

Продовження таблиці 2.3

Обмеження	Опис	Шляхи вирішення
Відсутність підключення до реальної мережі	Неможливість реалізації віддаленого керування в емуляторі	Перенесення проєкту на фізичний пристрій
Відсутність реальних зовнішніх впливів (наприклад, зміни температури середовища)	Недостатній рівень демонстрації адаптивності	Моделювання температурних змін у вигляді алгоритмів

Функціональна схема включає центральний мікроконтролер ESP32, до якого підключені температурний датчик DS18B20, OLED-дисплей для відображення даних, потенціометр, кнопка для ручного керування станом котла (ON/OFF), перемикач режимів для вибору між автоматичним і ручним керуванням, та реле для керування електрокотлом, а також потенціометр, який дозволяє користувачеві налаштовувати бажану температуру підтримки. Основні зв'язки та взаємодії ілюструє рисунок 2.1, поданий нижче.

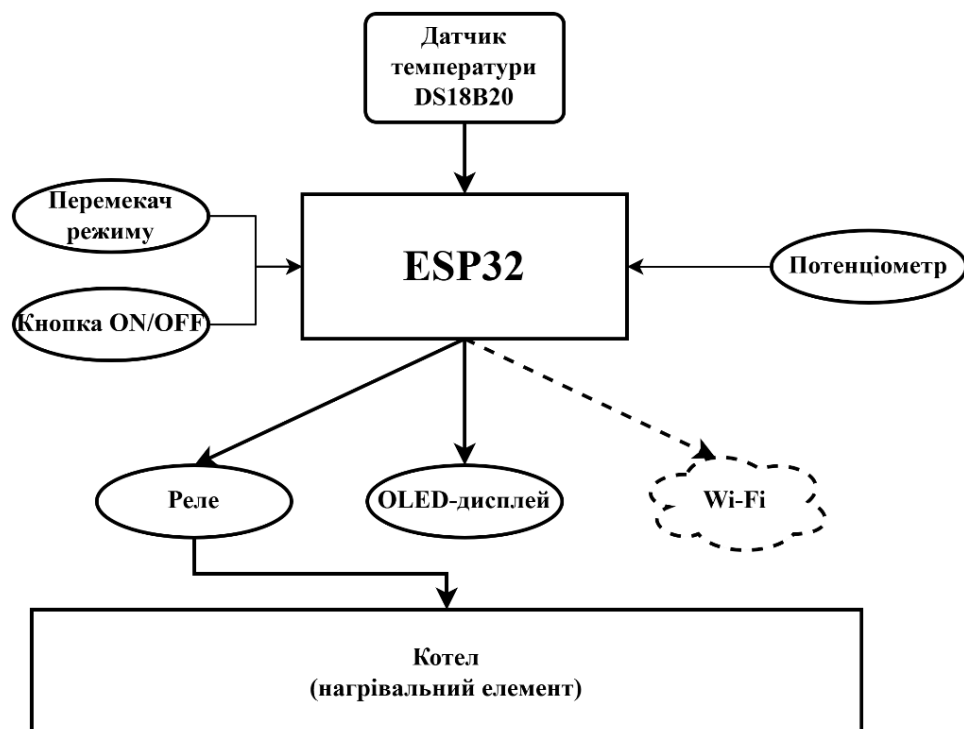


Рисунок 2.1 – Функціональна архітектура системи

З позиції архітектури програмно-апаратного комплексу, система складається з окремих компонентів із чітко визначеними функціями. Центральний елемент — мікроконтролер ESP32, який обробляє дані, керує виконавчими пристроями (реле), приймає сигнали з датчика температури DS18B20, виводить інформацію на OLED-дисплей і забезпечує віддалений доступ. Логічну взаємодію між компонентами подано у вигляді компонентної UML-діаграми (Рисунок 2.2), що відображає структуру системи та напрями зв'язку між модулями



Рисунок 2.2 – UML-діаграма компонентної структури системи

Також, з метою формалізації поведінки системи управління опаленням була розроблена UML-діаграма послідовностей (рисунок 2.3), яка відображає ключові етапи взаємодії між основними компонентами: мікроконтролером ESP32, температурним сенсором DS18B20, дисплеєм SSD1306, пам'яттю EEPROM, Wi-Fi модулем та релейним виконавчим пристроєм.

Діаграма ілюструє початкову ініціалізацію системи після запуску, зчитування налаштувань з EEPROM, підключення до мережі Wi-Fi та первинне вимірювання температури. Основна частина діаграми охоплює періодичну логіку роботи системи в основному циклі (loop), де здійснюється

вимірювання температури, прийняття рішення щодо ввімкнення або вимкнення нагрівального елемента відповідно до режиму (автоматичного або ручного), а також виведення інформації на OLED-дисплей.

Така діаграма дозволяє наочно продемонструвати порядок виконання ключових дій системи та логіку обробки сенсорних даних у реальному часі.

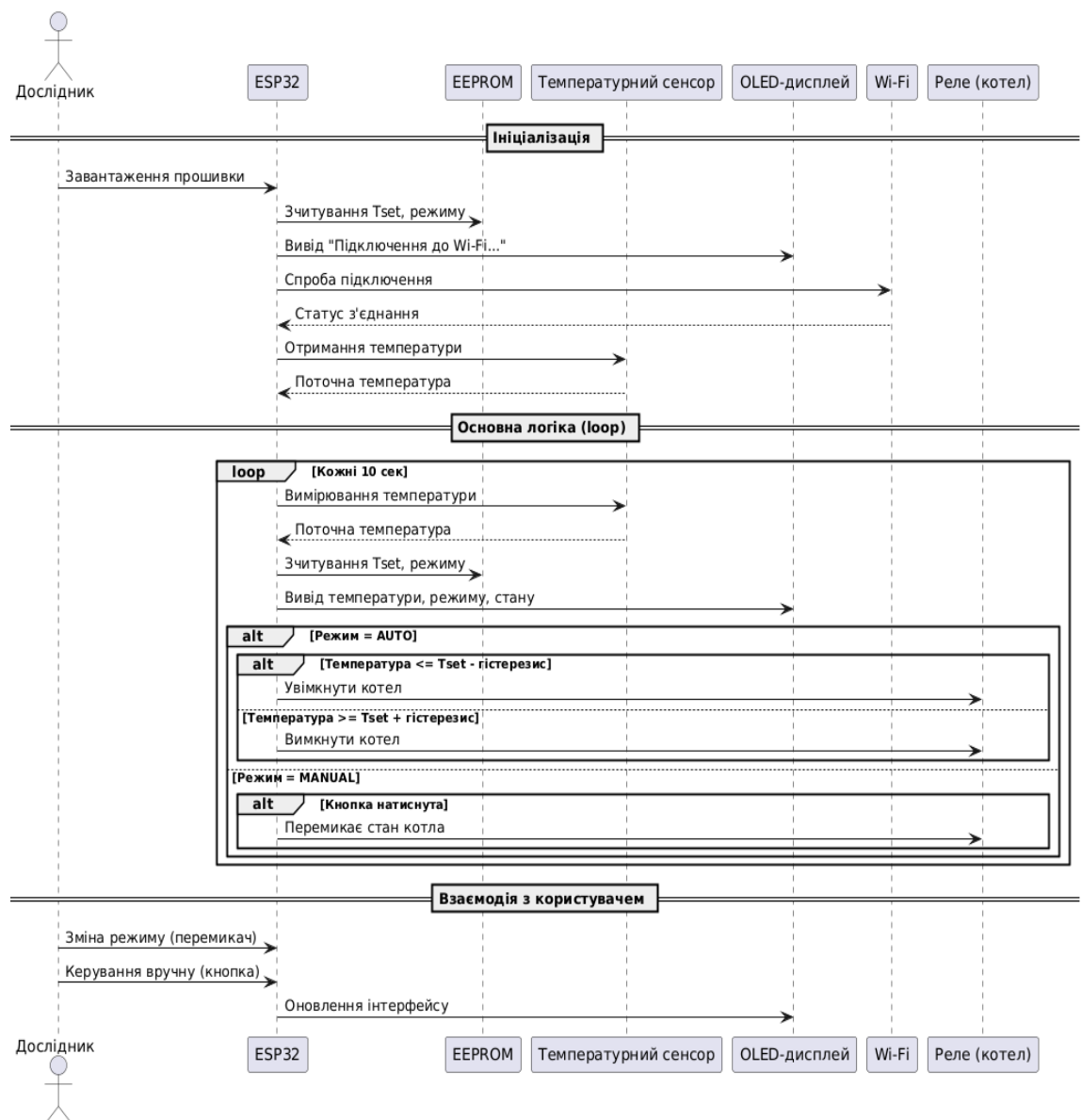


Рисунок 2.3 – Діаграма послідовностей роботи системи керування температурою на базі ESP32

Архітектура навчальної моделі базується на адаптивному керуванні з гістерезисом. Основна математична модель регулювання температури в системі описується наступним чином:

$$T_{on}=T_{set} \quad (2.1)$$

$$T_{off}=T_{set}+H \quad (2.2)$$

де T_{on} — температура увімкнення нагріву,

T_{off} — температура вимкнення нагріву,

T_{set} — задана температура,

H — гістерезис.

Цей підхід дозволяє уникнути частого вмикання-вимикання котла, забезпечуючи стабільну роботу.

Для демонстраційних цілей було обрано Wi-Fi та протокол HTTP через його простоту налаштування, надійність та поширеність у навчальних проєктах. Архітектура системи передбачає можливість реалізації простого вебінтерфейсу для моніторингу параметрів, однак у поточній реалізації дана функціональність обмежується виведенням IP-адреси пристрою та локальним відображенням даних на дисплеї. Повноцінне віддалене керування може бути реалізоване на наступному етапі розробки за наявності фізичної апаратної платформи з доступом до реальної мережі Інтернет.

Візуалізація реалізована за допомогою OLED-дисплею з транслітерацією повідомлень, що є компромісним рішенням через обмеження Wokwi. Інтерфейс включає виведення IP-адреси, поточної температури, режиму роботи, стану котла та заданих меж керування. Типові труднощі, які було попередньо визначено:

- Некоректні значення температури при старті системи (вирішується багаторазовим опитуванням датчика);
- Переривання Wi-Fi зв'язку (передбачено механізм автоматичного перепідключення);
- Помилки візуалізації інформації (вирішено за допомогою транслітерації).

В результаті аналізу основних компонентів майбутньої автоматизованої системи управління водяним опаленням, особлива увага була приділена вибору мікроконтролера, як центрального елемента функціональної

архітектури. Основу, як вже було визначено в пункті 2.4, системи становитиме сучасний мікроконтролер ESP32 компанії Espressif Systems, вибір якого зумовлений високою продуктивністю, широким набором периферійних модулів та інтегрованими функціями, що відповідають специфіці задачі.

Для організації точних вимірювань температури використовується цифровий температурний датчик DS18B20 (Рисунок 2.4). Цей датчик має високу точність вимірювання $\pm 0,5^{\circ}\text{C}$ у діапазоні температур від -10 до $+85^{\circ}\text{C}$ та працює за протоколом OneWire, що значно спрощує підключення до мікроконтролера, оскільки для взаємодії з декількома датчиками достатньо всього однієї сигнальної лінії. Це дозволяє суттєво скоротити кількість необхідних портів ESP32 та спрощує монтажні роботи.

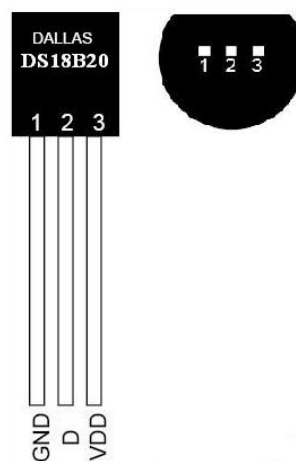


Рисунок 2.4 – датчик температури DS18B20

Важливим елементом є реле (Рисунок 2.5), яке використовується для керування нагрівачем або насосами опалювальної системи. Реле управляється сигналами, які генерує ESP32 залежно від результатів аналізу температурних показників. В алгоритмі роботи використовується гістерезис, що дозволяє уникнути частих перемикань і, відповідно, подовжує термін служби компонентів системи.



Рисунок 2.5 – Реле LEG-12 Rayex

Для підтримання бажаної температури в системі використовується аналоговий потенціометр (Рисунок 2.6). За його допомогою користувач може оперативно змінювати значення температури, яку система має підтримувати. Потенціометр підключено до аналогового входу мікроконтролера ESP32, де зчитування значень здійснюється через вбудований АЦП. Такий підхід забезпечує інтуїтивно зрозуміле ручне налаштування параметрів роботи системи без потреби в додаткових програмних інтерфейсах.



Рисунок 2.6 – Потенціометр WH148

Для наочної візуалізації інформації про стан системи, поточну температуру та режим роботи передбачено використання OLED-дисплея з роздільною здатністю 128×64 пікселів на базі контролера SSD1306 (Рисунок 2.7). Дисплей дозволяє в реальному часі відображати поточні показники, статус з'єднання з Wi-Fi мережею, стан нагрівача та поточний режим роботи системи (ручний чи автоматичний). Це значно підвищує зручність роботи з системою, особливо в навчальному процесі, де важливо бачити результати роботи алгоритмів та реакцію системи на зміни умов.

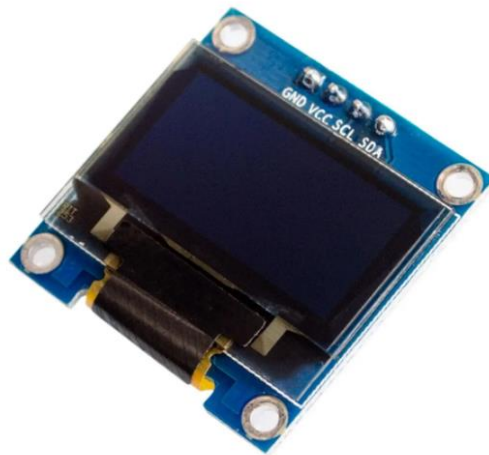


Рисунок 2.7 – OLED-дисплея з роздільною здатністю 128×64 пікселі

Кнопка ручного керування (ON/OFF) (Рисунок 2.8) підключена до одного з цифрових входів мікроконтролера ESP32 та використовує внутрішній підтягуючий резистор. Її функція полягає в тому, щоб вручну вмикати або вимикати нагрівач у режимі ручного керування. Користувач натискає кнопку — і система змінює свій стан на протилежний, дозволяючи оперативно управляти обігрівом без залучення автоматичних алгоритмів.



Рисунок 2.8 – Тактова кнопка для ручного керування котлом

Перемикач режимів роботи реалізовано за допомогою трьохконтактного повзункового перемикача типу SPDT (Рисунок 2.9), який дозволяє обирати між автоматичним та ручним режимом керування. У залежності від положення перемикача, на відповідний пін мікроконтролера подається логічний рівень, що визначає активний режим роботи системи.



Рисунок 2.9 – Повзунковий перемикач режимів роботи (ручний / автоматичний)

Ще одним потенційним компонентом функціональної архітектури є вебінтерфейс, реалізація якого передбачена на рівні архітектури. Завдяки наявності модуля Wi-Fi, мікроконтролер ESP32 дозволяє створювати вбудований вебсервер для організації дистанційного керування системою через браузер або мобільний пристрій. У поточній реалізації ця можливість не була повністю інтегрована, однак базові елементи — зокрема виведення IP-

адреси пристрою на OLED-дисплей — дозволяють у майбутньому без додаткового апаратного втручання реалізувати повноцінний вебінтерфейс. Це розширить можливості системи, зокрема в аспектах інтеграції з іншими пристроями розумного будинку, гнучкого керування параметрами та віддаленого моніторингу.

З метою підвищення обґрунтованості вибору архітектури також доцільно додати математичну оцінку енергоефективності, яка може включати формулу для розрахунку середньої потужності споживання системи:

$$P_{avg} = \frac{\sum_{i=1}^n P_i \cdot t_i}{T} \quad (2.3)$$

де P_i — потужність, t_i — час роботи системи у відповідному режимі, T — загальний час роботи.

Ця формула дозволяє попередньо оцінити споживання енергії та підтвердити доцільність обраної архітектури з точки зору енергоефективності.

Електрично-принципова схема є також критично важливим етапом проєктування автоматизованої навчальної моделі системи управління водяним опаленням, оскільки забезпечує детальне і цілісне уявлення про взаємодію та особливості роботи її компонентів. Саме на цьому етапі встановлюються всі основні зв'язки між елементами, визначаються особливості передачі сигналів, живлення, захисту та інших важливих параметрів функціонування системи.

Електрична взаємодія компонентів навчальної моделі автоматизованої системи керування водяним опаленням базується на чіткому визначенні функціональних зв'язків, які забезпечують коректну роботу системи відповідно до поставлених завдань. Центральним елементом схеми є мікроконтролер ESP32, який виконує функції обчислювального та керуючого модуля і забезпечує зв'язок між всіма пристроями в системі (Рисунок 2.10).

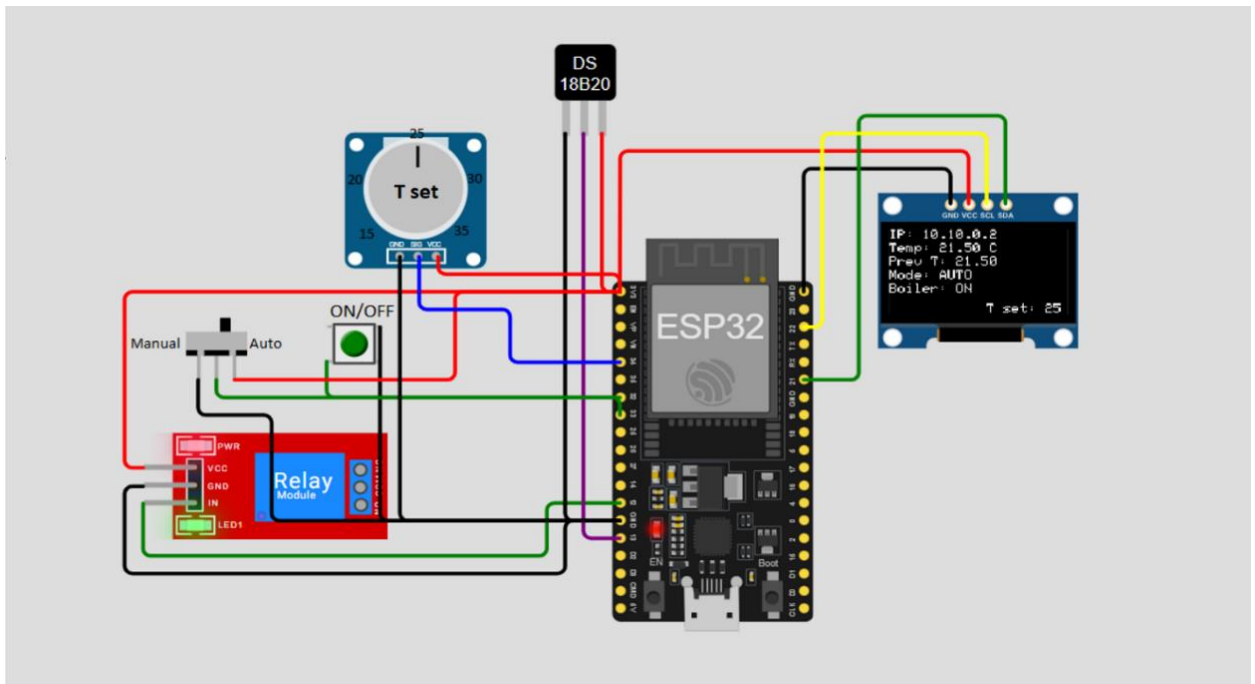


Рисунок 2.10 - Електрична взаємодія компонентів навчальної моделі автоматизованої системи керування водяним опаленням

Живлення всієї схеми організовано таким чином, що ESP32 отримує стабілізовану напругу постійного струму 5 В через USB-порт, який додатково виконує роль комунікаційного інтерфейсу для програмування. Від цього ж джерела живляться периферійні компоненти – релейний модуль, дисплей OLED, потенціометр, перемикач режимів та датчик температури DS18B20. Напруга 3,3 В для живлення логічних елементів (ESP32 та дисплей OLED) забезпечується внутрішнім стабілізатором напруги, вбудованим у плату ESP32. Це гарантує стабільну роботу при малих струмах навантаження і захищає компоненти від перевантаження.

Датчик температури DS18B20 підключається за схемою з використанням шини OneWire, яка дозволяє передавати інформацію послідовно по однопровідному інтерфейсу даних з використанням лише однієї сигнальної лінії та загального дроту. Даний датчик під'єднано до одного з GPIO пінів мікроконтролера ESP32 (наприклад GPIO13), з обов'язковим застосуванням підтягуючого резистора опором 4,7 кОм між лінією даних та

лінією живлення +3,3 В. Цей резистор забезпечує стабільність сигналу, необхідну для безпомилкового зчитування температурних даних.

OLED-дисплей підключається до ESP32 через стандартний інтерфейс I2C, що використовує два сигнальні дроти – SDA (лінія даних) та SCL (лінія синхронізації). Відповідні лінії дисплея під'єднані до спеціально визначених GPIO портів мікроконтролера (наприклад, SDA до GPIO21, SCL до GPIO22). Інтерфейс I2C реалізує послідовний обмін даними між дисплеєм та контролером, що забезпечує швидке і зручне виведення текстової та графічної інформації. Для роботи дисплея також необхідне стабілізоване живлення на 3,3 В, яке подається з плати ESP32.

Релейний модуль керується цифровим сигналом з виходу GPIO мікроконтролера ESP32 (наприклад GPIO12). Реле має вбудований транзисторний ключ та оптрон, що дозволяє керувати навантаженнями з високою напругою та струмом, ізолюючи їх від мікроконтролера. Таким чином, мікроконтролер лише подає керуючий сигнал низької напруги (логічний рівень), а безпосередньо комутацію силового кола здійснює транзисторна схема всередині релейного модуля. Для захисту транзисторних ключів у схемі реле використовується захисний діод (наприклад, типу 1N4001), який запобігає виникненню зворотних струмів при перемиканні індуктивних навантажень (наприклад, насосів).

Потенціометр використовується для встановлення бажаної температури нагріву (T set). Його сигнальний вивід підключений до аналогового входу (наприклад GPIO34) мікроконтролера ESP32, а крайні виводи — до живлення 3.3 В і землі (GND) відповідно. У процесі роботи ESP32 зчитує аналогове значення з потенціометра, яке змінюється залежно від положення ручки. Отримане значення (в діапазоні від 0 до 4095) масштабується до діапазону температур, наприклад від 15 до 35 °C, і використовується як цільова температура в автоматичному режимі керування котлом. Таким чином, потенціометр дозволяє швидко змінювати температуру без потреби у зміні програмного коду.

Перемикач у схемі використовується для вибору режиму роботи системи — автоматичного або ручного. Це стандартний повзунковий перемикач, який має три виводи. Середній контакт (спільний) підключено до входу (наприклад GPIO32) мікроконтролера ESP32. Один крайній контакт з'єднано із землею (GND), інший — із живленням 3.3 В. Залежно від положення перемикача, на вхід мікроконтролера подається або високий (логічна "1"), або низький рівень (логічний "0"). У програмному коді цей сигнал використовується для визначення активного режиму: якщо рівень високий — активується автоматичний режим керування температурою; якщо низький — система працює в ручному режимі, і керування котлом здійснюється кнопкою.

Кнопка увімкнення та вимкнення системи при ручному керуванні підключається до входу GPIO мікроконтролера ESP32 (наприклад GPIO33) і використовує внутрішній підтягуючий резистор контролера. Вона формує логічні рівні сигналу: при натисканні кнопки контакти замикаються на землю (GND), змінюючи логічний рівень на вході мікроконтролера з високого (стан без натискання) на низький, чим активується відповідний режим роботи.

Отже, електрична взаємодія всіх компонентів у автоматизованій системі керування опаленням побудована таким чином, щоб забезпечити безпечну та ефективну роботу пристроїв, точність зчитування показників, стабільність передачі інформації та простоту управління і моніторингу. Це дозволяє наочно і повноцінно демонструвати функціонування адаптивних алгоритмів керування температурою в нашій системі (Рисунок 2.10)

Таким чином в цьому підрозділі було детально описано та обґрунтовано вибір основних компонентів, сформована архітектура, яка дозволяє повною мірою реалізувати сучасні принципи адаптивного керування водяним опаленням та відповідає освітнім цілям бакалаврського проєкту. Це створює підґрунтя для подальших етапів роботи, зокрема розробки алгоритмів та моделювання системи.

2.2. Розробка та обґрунтування алгоритмів системи

Розробка та обґрунтування алгоритмів роботи навчальної моделі здійснювалася на підставі попередньо сформованих функціональних вимог та визначених обмежень. Алгоритмізація процесів автоматизованої системи керування водяним опаленням передбачає послідовне розв'язання декількох підзадач: вимірювання температури за допомогою сенсора DS18B20, обробку та аналіз отриманих даних, формування керуючих сигналів для вмикання чи вимикання опалювального пристрою, налаштування цільової температури за допомогою потенціометра, реалізація перемикання між ручним і автоматичним режимами роботи системи за допомогою перемикача, виведення інформації про стан системи на дисплей.

Алгоритм вимірювання температури є базовим у системі та реалізується шляхом циклічного опитування цифрового датчика DS18B20. Отримані цифрові дані перетворюються у числове значення температури, яке передається для подальшої обробки.

Наступною задачею є обробка отриманих значень температури з урахуванням гістерезису. Це необхідно для запобігання частим перемиканням реле керування нагрівальним елементом, що підвищує надійність та довговічність системи. Для цього використовується двопозиційне релейне керування, при якому нагрівальний пристрій активується, якщо температура опускається нижче встановленої межі на величину гістерезису, і вимикається, коли вона досягає верхньої межі. Величина гістерезису визначається експериментально, виходячи з вимог комфорту та економії енергоресурсів, і в даній моделі становить $1,5^{\circ}\text{C}$.

Алгоритм обробки сигналів також передбачає можливість роботи системи в автоматичному та ручному режимах. В автоматичному режимі система самостійно ухвалює рішення про включення чи вимкнення пристрою на основі температурних показників. У ручному режимі керування системою здійснюється користувачем за допомогою фізичної кнопки, що дозволяє

примусово вмикати або вимикати опалення незалежно від температурних даних.

При цьому, з метою безпеки, у ручному режимі реалізовано захисний механізм перегріву: якщо фактична температура перевищує 35°C + значення гістерезису, система автоматично вимикає нагрівальний елемент. Такий підхід дозволяє зберегти повний контроль користувача, але водночас забезпечити базовий рівень захисту від потенційно небезпечних температур.

Алгоритм візуалізації інформації забезпечує зручність використання системи та реалізується за допомогою OLED-дисплея SSD1306. На екрані виводиться інформація про поточний режим роботи системи (автоматичний або ручний), значення поточної температури, встановлене цільове значення температури та стан нагрівального пристрою (включений чи виключений). Інформація на дисплеї оновлюється циклічно з певною періодичністю для забезпечення актуальності даних.

Додатково у структурі системи архітектурно передбачено алгоритм роботи вебінтерфейсу, який у майбутньому дозволить здійснювати дистанційний моніторинг і управління через Wi-Fi-мережу. Такий алгоритм передбачає підтримку постійного підключення до локальної мережі, обробку HTTP-запитів і динамічне оновлення інформації про стан системи в реальному часі. У поточній реалізації ця функціональність не впроваджена повністю, однак її інтеграція можлива без зміни апаратної частини завдяки можливостям мікроконтролера ESP32. Впровадження вебінтерфейсу дозволить значно розширити функціональні можливості системи, забезпечивши зручне дистанційне керування, що є ключовим елементом сучасних автоматизованих рішень у сфері “розумного дому”.

Для більш чіткого розуміння взаємозв'язків та алгоритмічних залежностей між окремими компонентами та функціями системи, була сформована таблиця (Таблиця 2.4).

Таблиця 2.4 – Взаємозв’язок алгоритмів функціонування навчальної моделі

Підзадача	Опис підзадачі	Залежності
Вимірювання температури	Використання датчика DS18B20 для визначення температури	—
Обробка сигналів та керування	Аналіз даних сенсора, активація або деактивація реле з урахуванням гістерезису	Вимірювання температури
Візуалізація даних	Виведення інформації на OLED-дисплей SSD1306	Обробка сигналів
Вибір режиму керування	Перемикання між автоматичним і ручним режимом роботи за допомогою перемикача	
Налаштування температури	Використання потенціометра для встановлення бажаного значення температури	

Таким чином, розроблені алгоритми функціонування навчальної моделі автоматизованої системи керування водяним опаленням забезпечують високу ефективність та надійність, створюють умови для зручної експлуатації та відповідають сучасним вимогам автоматизації. Подальші етапи роботи передбачають практичне моделювання роботи цих алгоритмів та детальний аналіз отриманих результатів.

Загальна структура коду реалізує усі описані алгоритмічні блоки: вимірювання температури, логіку гістерезису, перемикання режимів, візуалізацію даних на дисплеї та підключення до мережі Wi-Fi. Початкова ініціалізація компонентів виконується в межах функції `setup()`:

```
void setup() {
    Serial.begin(115200);
    disp_init();           // Ініціалізація дисплея
    pinMode(potPin, INPUT); // Пін потенціометра
    pinMode(modeSwitch, INPUT); // Пін перемикача режиму
    pinMode(led1, OUTPUT);  // Пін реле
    pinMode(but, INPUT_PULLUP); // Пін кнопки
    digitalWrite(led1, 0);   // Котел вимкнено
    DS18B20.begin();         // Старт сенсора
    EEPROM.begin(512);       // Ініціалізація EEPROM
    i_tset = EEPROM.read(adr); // Зчитування температури з EEPROM
    if (i_tset == 0 || i_tset > 100) {
```

```

i_tset = 25;           // Значення за замовчуванням
EEPROM.write(adr, i_tset);
EEPROM.commit();
}
byte autoFlag = EEPROM.read(adr2); // Зчитування режиму
Auto = (autoFlag == 0);
lastAuto = Auto;       // Збереження поточного стану
tset = String(i_tset);  // Формування рядка температури
wifi_init();           // Підключення до Wi-Fi
temper();

```

Функція `temper()` реалізує алгоритм зчитування температури з цифрового датчика DS18B20 з урахуванням обробки помилкових значень:

```

void temper() {
  int c1 = 1;
  DS18B20.requestTemperatures();
  do {
    DS18B20.requestTemperatures();
    celsius = DS18B20.getTempCByIndex(0);
    dtostrf(celsius, 2, 2, buf);
    delay(100);
    if (c1 == 50) {
      dtostrf(0.00, 2, 2, buf);
      break;
    }
    ++c1;
  } while (celsius == 85.0 || celsius == (-127.0));
  sprintf(strok, "%s", buf);
}

```

Ключова логіка температурного регулювання, реалізована в `loop()`, забезпечує виконання умов гістерезису як у ручному, так і в автоматичному режимі.

У автоматичному режимі система керує нагрівом на основі заданого значення температури (`Tset`) та гістерезису (`gister`). Це дозволяє уникнути надмірного частого перемикання реле при коливаннях температури в межах допустимого діапазону. Програмна реалізація має вигляд:

```

if (Auto) {
  if (celsius <= i_tset - gister && digitalRead(led1) == 0)
    digitalWrite(led1, 1); // Увімкнути котел
  else if (celsius >= i_tset + gister && digitalRead(led1) == 1)
    digitalWrite(led1, 0); // Вимкнути котел
}

```

У ручному режимі керування відбувається натисканням кнопки. Однак з метою безпеки реалізовано програмне обмеження: у разі перевищення 35 °C

+ gister нагрів вимикається незалежно від стану кнопки. Це запобігає перегріву при тривалому ручному керуванні:

```
else {
    // Якщо температура 35°C + gister або вище – автоматичне вимикання котла
    if (celsius >= 35.0 + gister && digitalRead(led1) == 1) {
        digitalWrite(led1, 0);
    }
    else if (celsius < 35.0 + gister) {
        // Якщо температура нижче 35°C + gsiter – котел керується вручну через кнопку
        if (swt == 1 && digitalRead(led1) == 0) {
            digitalWrite(led1, 1);
        }
        else if (swt == 0 && digitalRead(led1) == 1) {
            digitalWrite(led1, 0);
        }
    }
}
```

Оновлення інформації на дисплеї виконується з інтервалом у 20 секунд, що дозволяє в реальному часі спостерігати зміну температури та режиму роботи:

```
if (ss == 0 || ss == 20 || ss == 40) {
    temper();
    ttset = String(i_tset);
    display.clearDisplay();
    display.setTextSize(1);
    display.setCursor(0, 0);
    display.print("IP: ");
    display.print(WiFi.localIP());
    display.setCursor(0, 10);
    display.print("Temp: " + String(celsius) + " C");
    display.setCursor(0, 20);
    display.print("Mode: " + String(Auto ? "AUTO" : "MANUAL"));
    display.setCursor(0, 40);
    display.print("Boiler: ");
    display.print(digitalRead(led1) == 0 ? "OFF" : "ON");
    display.setCursor(70, 55);
    display.print("T set: " + ttset);
    display.display();
}
```

Застосування такої структури забезпечує наочну відповідність реалізованого коду описаним вище алгоритмам, що дозволяє ефективно

відстежити логіку взаємодії між сенсорними даними, режимами керування та виконавчими елементами.

2.3. Моделювання роботи системи та аналіз результатів

На цьому етапі було здійснено моделювання роботи розробленої автоматизованої навчальної системи керування водяним опаленням у середовищі емуляції Wokwi. Обраний інструмент забезпечує зручне відтворення поведінки апаратних компонентів (ESP32, реле, кнопки, потенціометр, дисплей, датчик температури DS18B20) у віртуальному середовищі та дозволяє перевірити коректність реалізованих алгоритмів без фізичного прототипу.

Під час моделювання було протестовано базовий сценарій роботи системи в автоматичному режимі з температурним гістерезисом. На дисплеї виводиться поточна температура, задана температура (T_{set}), режим роботи (Auto), стан котла (ON/OFF) та IP-адреса пристрою. У відповідності до заданих умов алгоритму, котел вмикається при досягненні температури нижче порогу ($T_{set} - \Delta T$), та вимикається при перевищенні верхнього порогу ($T_{set} + \Delta T$), що дозволяє уникнути частих перемикань та забезпечує енергоефективність.

Окрім стандартного сценарію, моделювання включало імітацію змінних умов експлуатації, зокрема, варіювання вхідної температури та перевірку стабільності при роботі з різною швидкістю зміни температури. Було досліджено, як система реагує на миттєві перепади, повільне підвищення чи зниження температури, а також чи не виникає помилок при граничних значеннях. Також перевірено реакцію на некоректну роботу сенсора, що імітує реальні експлуатаційні обставини.

У процесі тестування системи було зафіксовано такі приклади поведінки: початкова температура становила 21,0 °C, при цьому задана температура була встановлена на рівні 25 °C. Система відповідно до алгоритму включила котел, забезпечуючи нагрів до необхідного значення. Після досягнення температури 25,8 °C відбулося вимкнення нагрівача. Згодом

температура знизилася до 23,3 °C, що знову активувало реле, демонструючи чітке дотримання меж гістерезису та стабільність алгоритму.

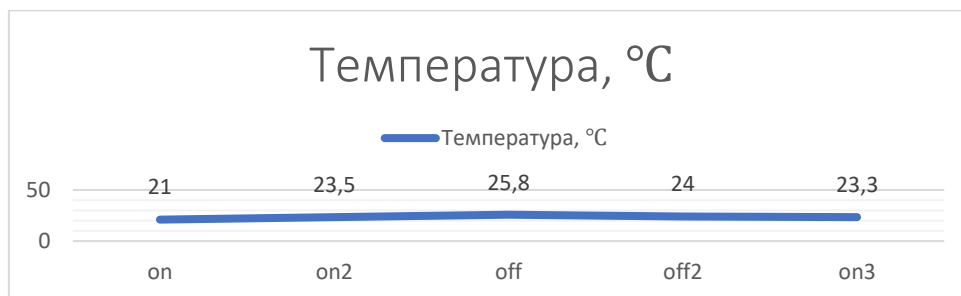


Рисунок 2.11 – Динаміка температури та реакція навчальної моделі на порогові значення

Окрім наведеної діаграми (Рисунок 2.11), було зібрано також розширений набір даних (Таблиця 2.5) для побудови статистичних графіків. У таблиці нижче подано вибірку з температурними значеннями за період спостереження 10 хвилин із фіксацією кожні 30 секунд

Таблиця 2.5 – Вибірка температурних показників та відповідного стану котла за період моделювання

Час, с	Температура, °C	Котел
0	22,1	ON
30	22,8	ON
60	23,5	ON
90	24,3	ON
120	25,2	OFF
150	25,6	OFF
180	25,3	OFF
210	24,6	OFF
240	23,8	OFF
270	23,1	ON
300	22,5	ON
330	21,9	ON

Продовження таблиці 2.5

Час, с	Температура, °C	Котел
360	21,3	ON
390	21,0	ON
420	21,2	ON
450	22,0	ON
480	22,7	ON
510	23,5	ON
540	24,4	ON
570	25,0	OFF
600	25,5	OFF

Аналіз моделі показав:

1. Алгоритми працюють стабільно при зміні температурних умов.
2. Відображення даних на OLED-дисплеї здійснюється коректно.
3. Перемикання режимів (автоматичний/ручний) функціонує згідно з логікою.
4. При натисканні на кнопку система змінює стан реле відповідно до встановленого режиму.

У моделюванні також було протестовано граничні стани: відсутність сигналу з датчика, зміна напруги живлення, перезапуск ESP32, перевищення температури понад безпечне значення в ручному режимі. У всіх випадках система демонструвала передбачувану поведінку: автоматичне вимкнення нагріву або повернення до робочого стану після відновлення параметрів. Додатково здійснено вимірювання часу реакції системи на зміну температури та латентність відображення змін на OLED-дисплеї, які виявились незначними (менше 500 мс), що свідчить про високу оперативність роботи програмної логіки.

На рисунку. 2.12 представлено вивід на OLED-дисплей під час роботи системи у середовищі моделювання. Відображаються ключові параметри: IP-адреса пристрою, поточна температура, попереднє значення, режим роботи

(AUTO/MANUAL), стан нагрівального пристрою (Boiler ON/OFF) та встановлена температура.

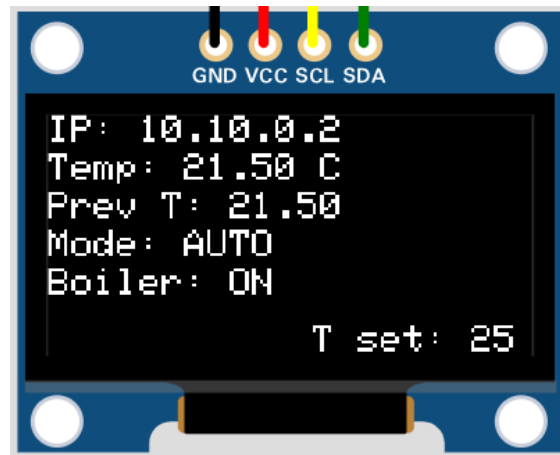


Рисунок 2.12 – Екран OLED-дисплея з виводом статусу системи в режимі моделювання

Особливу увагу приділено моделюванню поведінки в ручному режимі, при якому користувач має можливість самостійно вмикати або вимикати котел незалежно від показників температури. Це дозволило перевірити, наскільки надійно система реагує на зовнішні команди користувача та чи не відбувається збоїв у логіці переходів між станами. При натисканні кнопки режиму перемикається на ручний, при цьому дисплей миттєво оновлює відповідну інформацію, а стан реле змінюється згідно з заданою командою.

На рисунку 2.13 показано екран OLED-дисплея під час аварійного вимкнення нагрівання через перевищення безпечної температури в ручному режимі. Це забезпечує захист системи від перегріву приміщення та запобігає можливим пошкодженням обладнання.

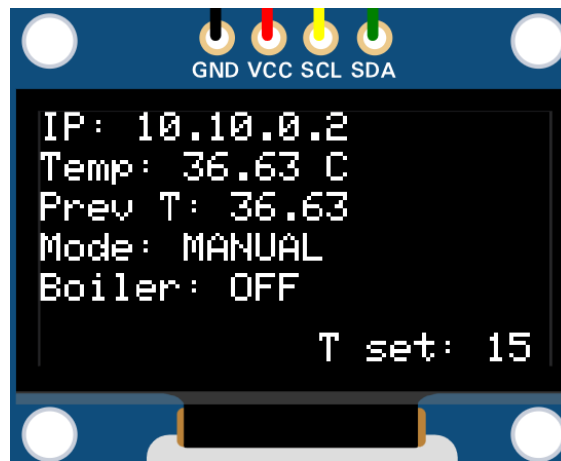


Рисунок 2.13 - Аварійне вимкнення нагрівання при перевищенні безпечної температури в ручному режимі

Таким чином, моделювання підтвердило відповідність реалізованої навчальної системи технічним вимогам, дозволило провести налагодження коду та алгоритмів без використання фізичного обладнання і стало важливим етапом у верифікації працездатності автоматизованої системи керування водяним опаленням. Отримані результати стануть підґрунтям для подальшої фізичної реалізації, впровадження віддаленого доступу через Wi-Fi та вдосконалення адаптивних функцій системи.

3. АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ ТА ПЕРСПЕКТИВИ ПОДАЛЬШОГО РОЗВИТКУ СИСТЕМИ

У ході виконання бакалаврської кваліфікаційної роботи була поставлена мета – розробка навчальної моделі автоматизованої системи керування водяним опаленням на базі мікроконтролера ESP32 із використанням сенсорного контролю температури, виводу на OLED-дисплей. В результаті було створено функціональну архітектуру системи, розроблено та реалізовано алгоритми керування, здійснено моделювання та перевірку працездатності системи в середовищі емуляції Wokwi. Таким чином, поставлену мету досягнуто повністю.

Усі завдання, що були визначені на початковому етапі, виконано:

- Проведено аналіз існуючих рішень та технічних обмежень;
- Сформовано логічну структуру архітектури системи;
- Реалізовано алгоритм керування з урахуванням гістерезису;
- Розроблено механізм візуалізації інформації на дисплеї;
- Змодельовано систему у Wokwi та перевірено її роботу.

З технічної точки зору реалізовану систему можна охарактеризувати як стабільну та надійну. Всі ключові елементи – вимірювання температури, обробка даних, релейне керування та відображення інформації – працюють відповідно до логіки, описаної в алгоритмах. Усі режими перемикання, як ручний, так і автоматичний, були протестовані та показали належну реакцію на зміну умов. OLED-дисплей оновлював інформацію в межах інтервалу меншого за 500 мс, що забезпечує належну інформативність.

Особливу увагу було приділено тестуванню крайових умов, зокрема втрати сигналу з сенсора, скиданню мікроконтролера та симуляції нестабільного живлення. Система продемонструвала високу відмовостійкість – після рестарту ESP32 автоматично відновлювала працездатність без втрати критичних параметрів.

Серед зафіксованих обмежень – залежність точності роботи від якості датчика DS18B20 та стабільності живлення, а також обмеження по кількості керованих елементів через обмеження кількості GPIO. Крім того, простий характер вебінтерфейсу у навчальній моделі передбачає лише перегляд інформації, без повноцінного двостороннього керування.

Отримані результати дозволяють стверджувати, що система може бути використана як демонстраційна або навчальна модель у рамках освітніх курсів, присвячених мікроконтролерам, системам автоматизації або інтернету речей. Крім того, вона може служити основою для створення більш складних систем із використанням зовнішніх сенсорів, керуванням по MQTT або мобільними застосунками.

З господарської точки зору така система потенційно дозволяє зменшити витрати на опалення за рахунок оптимізації включення котла, зниження кількості циклів увімкнення/вимкнення та енергоефективного режиму роботи. Наукова цінність полягає у верифікації алгоритмів керування з урахуванням температурної інерції системи, а також у відпрацюванні архітектурних рішень для побудови адаптивних систем автоматизації. Соціальна значущість полягає в тому, що реалізовану модель можна відтворити у навчальних закладах без значних витрат на обладнання, а отже – вона сприяє популяризації STEM-освіти.

Одним із напрямів модернізації є впровадження інтелектуального регулювання температурного режиму, що враховує час доби, погодні умови (через API прогнозу погоди), а також поведінкові патерни користувача. Це дасть змогу перейти від фіксованого значення температури до адаптивного планування теплових циклів. З технічної точки зору, це передбачає побудову розкладів або впровадження алгоритмів машинного навчання.

Крім того, перспективним є підключення системи до серверних платформ, які підтримують протоколи MQTT або HTTP, що дозволяє збирати дані з великої кількості пристроїв у реальному часі та створювати системи моніторингу для керівних структур, компаній або обслуговуючого персоналу.

Інтеграція з хмарними платформами типу Blynk, Ubidots або Firebase дозволить створити масштабовану екосистему пристроїв.

Ще одним із перспективних напрямів є впровадження Telegram-ботів або мобільних повідомлень як каналу зворотного зв'язку. Завдяки цьому користувач зможе в реальному часі отримувати сповіщення про критичні зміни, аварійні ситуації, перевищення температурних меж або несанкціоновані дії.

Розширення кількості сенсорів, зокрема для контролю вологості, CO₂ або відкриття вікон/дверей, також може покращити адаптацію системи до реальних умов експлуатації. Це перетворює її на повноцінну частину «розумного дому» з можливістю взаємодії з іншими компонентами — охоронними, вентиляційними, освітлювальними.

Таким чином, навчальна модель демонструє хорошу масштабованість і здатність до подальшої еволюції в бік повноцінної автоматизованої системи керування опаленням із розширеним функціоналом та можливістю адаптації під потреби конкретного користувача або середовища використання.

ВИСНОВКИ

У процесі виконання бакалаврської кваліфікаційної роботи розроблено повний цикл автоматизованої навчальної системи керування водяним опаленням — від аналізу предметної області та формулювання цілей до побудови алгоритмів, моделювання функціонування та оцінки ефективності запропонованого рішення. Проєкт поєднує теоретичну базу, інженерне проєктування, програмну реалізацію та аналіз результатів, що забезпечує комплексний підхід і відповідає вимогам до кваліфікаційної роботи бакалавра.

Мета роботи полягала у створенні навчальної моделі автоматизованої системи керування температурним режимом водяного опалення на базі мікроконтролера ESP32 із використанням цифрового сенсора DS18B20, OLED-дисплея та програмного забезпечення, орієнтованого на енергоефективність і стабільну роботу. У рамках дослідження проаналізовано існуючі рішення, розроблено архітектуру системи, сформовано та реалізовано алгоритми керування, змодельовано функціонування у середовищі Wokwi, обґрунтовано вибір функціональних рішень і перевірено працездатність створеної моделі.

У першому розділі проведено аналіз сучасних систем водяного опалення, визначено їх основні обмеження — високу вартість, відсутність адаптивності та надмірну орієнтацію на промислові умови. Сформульовано мету дослідження на основі виявлених недоліків.

У другому розділі обґрунтовано вибір основних апаратних компонентів, зокрема ESP32 як керуючого елемента. Побудовано електричну схему, підібрано резистори, діоди, обчислено навантаження, враховано обмеження по струму та живленню, описано підключення ключових компонентів: сенсора, реле, кнопки, дисплея.

У третьому розділі розроблено функціональну архітектуру системи, що охоплює підсистеми вимірювання, аналізу, керування, візуалізації. Реалізовано алгоритми автоматичного й ручного режимів із підтримкою

гістерезису для зниження кількості перемикачів реле. Реакцію на перемикач забезпечено кнопкою, а відображення стану — через OLED-дисплей.

Реалізовано повноцінне програмне забезпечення із циклічним опитуванням сенсора DS18B20, перетворенням даних у температурні значення, обробкою логічних умов та виведенням параметрів на дисплей. Під час моделювання в середовищі Wokwi, підтверджено стабільну роботу алгоритму, правильність перемикачів при досягненні порогових значень, а також зручність інтерфейсу.

Під час моделювання зафіксовано коректну реакцію системи на зміну температури. При зниженні температури до $21,0\text{ }^{\circ}\text{C}$ ($T_{\text{set}} = 25$) система активує котел; при досягненні $25,8\text{ }^{\circ}\text{C}$ — вимикає його; при повторному зниженні до $23,3\text{ }^{\circ}\text{C}$ — знову вмикає. Така логіка свідчить про успішну реалізацію гістерезису та підтверджує енергоефективність керування. Побудовано гістограми, які відображають динаміку температури та відповідь системи у часі. Дані змодельовано з інтервалом 30 секунд протягом 10 хвилин, що дало змогу провести кількісний аналіз реакції системи та оцінити стабільність логіки керування.

Окреслено перспективи розвитку системи: фізичне впровадження, підтримка мобільних застосунків, адаптивне керування за графіком, інтеграція прогнозу погоди, машинного навчання, Telegram-ботів, MQTT, а також перехід до концепції «розумного будинку».

У підсумку, робота повністю відповідає поставленій меті, є науково обґрунтованою, має практичну значущість і може бути адаптована для подальшого розвитку як у навчальних цілях, так і для реального впровадження в побутові та промислові системи опалення. Усі етапи — від аналізу задачі до практичного моделювання — реалізовані відповідно до методичних рекомендацій, із дотриманням структурної, логічної та функціональної узгодженості проєкту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування. – К.: Мінрегіон України, 2013. – 134 с.
2. Ткачук А. Я., Зайченко Є. С., Потапов В. А., Цепелєв А. П. Системи опалення: проєктування та експлуатація. – К.: Будівельник, 1985. – 136 с.
3. Поетапна автоматизація системи опалення. – Adamson.ua – <https://adamson.ua/pro-kompaniyu/statti/poetapna-avtomatizaciya-sistemi-opalennya>
4. Як економити за допомогою технологій розумного будинку. – Shop-GSM.ua<https://shop-gsm.ua/catalog/internet-veschej-iot-umnyj-dom/>
5. Khodabandehlou H., Ghasemi M. (2020). IoT-based smart heating systems: A review. Journal of Building Engineering, 32, 101705.
6. Гаприндашвілі Б.В. Енергозбереження як чинник підвищення конкурентоспроможності промислових підприємств / Б.В. Гаприндашвілі // БізнесІнформ. – 2014. – №8. – С. 213–217.
7. Пупена О.М., Ельперін І.В., Луцька Н.М., Ладанюк А.П. Промислові мережі та інтеграційні технології в автоматизованих системах: навч. посіб. – К.: Ліра-К, 2011. – 552 с.
8. Цмоць І.Г., Скорохода О.В., Кісь Я.П. Синтез інтегрованих автоматизованих систем управління підприємством. – 2015.
9. Мікроконтролер ESP32 [Електронний ресурс]. – Режим доступу: <https://itmaster.biz.ua/directory/microcontrollers/esp32.html>
10. ESP-WROOM32 Datasheet [Електронний ресурс]. – Режим доступу: <https://www.alldatasheet.com/datasheet-pdf/view/1179101/ESPRESSIF/ESP-WROOM-32.html>
11. DS18B20 Datasheet [Електронний ресурс]. – Режим доступу: <https://www.alldatasheet.com/datasheet-pdf/view/58557/DALLAS/DS18B20.html>

12. Вивчення правил оформлення конструкторської документації (Схеми електричні принципи). – Київ: НТУУ «КПІ», 2016.
13. ДСТУ ГОСТ 2.104:2006. Єдина система конструкторської документації. – Київ: ДП «УкрНДНЦ», 2006.
14. Двоступінчаті системи гарячого водопостачання [Електронний ресурс]. – Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/0749d205-eef1-4ee3-9dd4-00ce17b14be6/content>
15. I2C Інтерфейс [Електронний ресурс]. – Режим доступу: <https://itmaster.biz.ua/directory/standarts/i2c.html>
16. ESP32 на ArduinoIDE [Електронний ресурс]. – Режим доступу: <https://itmaster.biz.ua/electronics/esp32/esp32-arduino.html>
17. Software Development Kit (SDK) [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/SDK>
18. Додаток на Android [Електронний ресурс]. – Режим доступу: <https://foxminded.ua/rozrobyty-dodatok-na-android/>
19. Методи передачі даних: GET і POST [Електронний ресурс]. – Режим доступу: <https://uk.php.brj.cz/metodi-peredaci-danih-get-i-post>
20. Алгоритми управління насосними групами і станціями [Електронний ресурс]. – Режим доступу: https://elecon.kiev.ua/doc/alg_upr_nasos.pdf
21. Шаров С.В., Лубко Д.В., Осадчий В.В. Інтелектуальні інформаційні системи: навч. посіб. – Мелітополь: Вид-во МДПУ ім. Б. Хмельницького, 2015.
22. Бережний І.В., Наконечний А.Й. Система дистанційного моніторингу та керування кліматичними параметрами приміщень у процесі промислового виробництва. – 2020.
23. Марченко Н.Б., Монченко О.В., Мартинюк Г.В. Багаторівневі системи моніторингу та діагностики як конструктивний розвиток інтелектуальних інформаційних систем. – 2021.
24. Теслюк Т.В., Цмоць І.Г., Юрій О., Теслюк В.М. Архітектура багаторівневої системи управління енергоефективністю регіону. – 2017.

25. Мови програмування та фреймворки для створення Android-додатків [Електронний ресурс]. – Режим доступу: <https://foxminded.ua/rozrobka-mobilnykh-dodatkov-android/>

ДОДАТОК А. ПРИКЛАДИ ВИХІДНОГО КОДУ

```

#include <SPI.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <EEPROM.h>
#include <stdio.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire);

const char* ssid = "Wokwi-GUEST";
const char* password = "";

String data_pay;
char strok[30];
char buf[30];

long sec;
int ss;
String ttset;
int i_tset;

bool Auto = true;
bool lastAuto = true;
float gister = 1.5;
byte swt = 1;
float celsius;
float o_celsius;

int adr = 1;
int adr2 = 2;
int cntry;

WiFiClient espclient;
OneWire ds(13);

const int led1 = 12;
const int but = 33;
const int modeSwitch = 32;
const int potPin = 34;

DallasTemperature DS18B20(&ds);

void temper() {
    int c1 = 1;

```

```

DS18B20.requestTemperatures();
do {
    DS18B20.requestTemperatures();
    celsius = DS18B20.getTempCByIndex(0);
    dtostrf(celsius, 2, 2, buf);
    delay(100);
    if (c1 == 50) {
        dtostrf(0.00, 2, 2, buf);
        break;
    }
    ++c1;
} while (celsius == 85.0 || celsius == (-127.0));
sprintf(strok, "%s", buf);
}

void disp_init() {
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        Serial.println(F("Display init error"));
    }
    display.display();
    delay(2000);
    display.clearDisplay();
}

void wifi_init() {
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 0);
    display.print(F("Connect to "));
    display.println(ssid);
    display.println("");
    display.print("Wait WiFi...");
    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        display.print(".");
        display.display();
        if (cntry == 20) break;
        ++cntry;
    }

    display.println("");
    display.clearDisplay();
    display.setCursor(0, 0);
    if (WiFi.status() != WL_CONNECTED) {
        display.println("WiFi not connected");
    } else {
        display.println("WiFi connected");
    }

    display.display();
    delay(500);
}

```



```

    display.print("IP: ");
    display.println(WiFi.localIP());
    display.display();
    delay(1000);
}

void setup() {
    Serial.begin(115200);
    bool lastAuto = digitalRead(modeSwitch);
    EEPROM.begin(512);

    i_tset = EEPROM.read(adr);
    if (i_tset == 0 || i_tset > 100) {
        i_tset = 25;
        EEPROM.write(adr, i_tset);
        EEPROM.commit();
    }

    byte autoFlag = EEPROM.read(adr2);
    Auto = (autoFlag == 0);

    Serial.print("EEPROM adr2: ");
    Serial.println(autoFlag);
    Serial.print("Auto mode is: ");
    Serial.println(Auto ? "AUTO" : "MANUAL");

    disp_init();

    pinMode(potPin, INPUT);
    pinMode(modeSwitch, INPUT);
    pinMode(led1, OUTPUT);
    pinMode(but, INPUT_PULLUP);
    digitalWrite(led1, 0);

    DS18B20.begin();
    ttset = String(i_tset);
    wifi_init();
    temper();

    if (WiFi.status() == WL_CONNECTED) {
        display.println("HTTP server ON");
        display.display();
        Serial.println("HTTP server ON");
        delay(2000);
    } else {
        display.println("HTTP server OFF");
        Serial.println("HTTP server OFF");
        display.display();
    }
}

void loop() {
    sec = millis() / 1000;

```

```

ss = sec % 60;
Auto = digitalRead(modeSwitch);
bool currentAuto = digitalRead(modeSwitch);
int potValue = analogRead(potPin);
i_tset = map(potValue, 0, 4095, 15, 35);

if (currentAuto != lastAuto) {
    Auto = currentAuto;
    Serial.print("Mode: ");
    Serial.println(Auto ? "AUTO" : "MANUAL");
    lastAuto = currentAuto;
}

if (digitalRead(but) == LOW && swt == 1) {
    swt = 0;
    Serial.println(swt);
}
if (digitalRead(but) == LOW && swt == 0) {
    swt = 1;
    Serial.println(swt);
}

if (Auto) {
    if (celsius <= i_tset - gister && digitalRead(led1) == 0) {
        digitalWrite(led1, 1);
    }
    else if (celsius >= i_tset + gister && digitalRead(led1) == 1)
    {
        digitalWrite(led1, 0);
    }
} else {
    if (celsius >= 35.0 + gister && digitalRead(led1) == 1) {
        digitalWrite(led1, 0);
    }
    else if (celsius < 35.0 + gister) {
        if (swt == 1 && digitalRead(led1) == 0) {
            digitalWrite(led1, 1);
        }
        else if (swt == 0 && digitalRead(led1) == 1) {
            digitalWrite(led1, 0);
        }
    }
}

if (ss % 10 == 0) {
    o_celsius = celsius;
    temper();
    ttset = String(i_tset);

    display.clearDisplay();
    display.display();
    display.setTextSize(1);
    display.setCursor(0, 0);
}

```

```
display.print("IP: ");
display.print(WiFi.localIP());

display.setCursor(0, 10);
display.print("Temp: " + String(celsius) + " C");

display.setCursor(0, 20);
display.print("Prev T: " + String(o_celsius));

display.setCursor(0, 30);
display.print("Mode: " + String(Auto ? "AUTO" : "MANUAL"));

display.setCursor(0, 40);
display.print("Boiler: ");
display.print(digitalRead(led1) == 0 ? "OFF" : "ON");

display.setCursor(70, 55);
display.print("T set: " + ttset);
display.display();
}
}
```