

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки

(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

(рівень вищої освіти)

Пристрій автоматичного керування температурою у приміщенні

на базі мікроконтролера

(тема)

Виконав: здобувач 4 року навчання,
групи КІУКІ-21-7

Тотіков С. Д.

(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерна інженерія

(повна назва освітньої програми)

Керівник проф. Литвинова Є.І.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

Чумаченко С.В.

(підпис)

(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія
(шифр і назва)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп'ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«__» _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Здобувачеві Тотікову Станіславу Денисовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Пристрій автоматичного керування температурою у приміщенні на базі мікроконтролера

затверджена наказом університету від " 21 " 05 2025 р. № 403 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 20.06.2025

3. Вихідні дані до роботи _____

функція – автоматичне керування температурою у приміщенні;
датчик температури DS18B20; мікроконтролер ESP32;

Мова програмування C++;

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз предметної галузі та постановка задачі проектування.

Огляд аналогічних пристроїв. Розробка структури пристрою.

Розробка програмної частини пристрою.

Розробка апаратної частини пристрою.

Тестування роботи пристрою.

Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Презентація, 12 слайдів

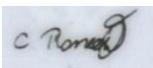
6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Видача теми проекту, узгодження і затвердження теми	02.05.2025-02.05.2025	
2	Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів	03.05.2025 -8.05.2025	
3	Розробка електричної схеми пристрою	9.05.2025 -12.05.2025	
4	Реалізація макету пристрою	13.05.2025 -20.05.2025	
5	Розробка програми для мікроконтролера	25.05.2025 -29.05.2025	
6	Тестування системи	30.05.2025 -04.06.2025	
7	Оформлення пояснювальної записки	05.06.2025 -07.06.2025	
8	Перевірка виконаного проекту керівником, допуск до захисту	08.06.2025 -10.06.2025	
9	Захист проекту	16.06.2025 -25.06.2025	

Здобувач



(підпис)

Тотіков С. Д.

Керівник роботи



(підпис)

проф. Литвинова С. І.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 43 с., 7 рис., 5 джерел.

АВТОМАТИЗАЦІЯ, ТЕМПЕРАТУРНИЙ КОНТРОЛЬ,
МІКРОКОНТРОЛЕР, ESP32, IoT.

Метою кваліфікаційної роботи є підвищення ефективності та зменшення вартості пристрою автоматичного керування температурою у приміщенні за рахунок використання мікроконтролера.

Вирішено задачі розробки структурної та принципової електричної схеми пристрою автоматичного керування температурою у приміщенні, програмного забезпечення для мікроконтролера ESP32, інтерфейсу на базі хмарної платформи Blynk. Реалізовано можливість віддаленого моніторингу та керування за допомогою мобільного додатку або через веб-додаток; проведено верифікацію розробленого пристрою за допомогою симуляції.

ABSTRACT

Bachelor's thesis contains 43 pages format A4, 7 figures and 5 sources in the list of references.

AUTOMATION, TEMPERATURE CONTROL, MICROCONTROLLER, ESP32, IoT.

The purpose of the qualification work is to increase the efficiency and reduce the cost of the automatic temperature control device in the room through the use of a microcontroller.

The tasks of developing a structural and schematic electrical diagram of the automatic temperature control device in the room, software for the ESP32 microcontroller, and an interface based on the Blynk cloud platform have been solved. The possibility of remote monitoring and control using a mobile application or via a web application has been implemented; the developed device has been verified using simulation.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ	
I ТЕРМІНІВ	7
ВСТУП	8
1 ОГЛЯД ЛІТЕРАТУРИ	10
1.1. Мікроконтролер	10
1.2. Аналіз технологій вимірювання температури	14
1.3 Модуль реле	15
1.4. Принципи та алгоритми роботи пристрою	16
1.5. Інтеграція з технологіями Інтернету речей	17
1.6. Переваги підходу на відкритих платформах	18
2 СТРУКТУРА ПРИСТРОЮ	20
2.1 Архітектура та функціональні модулі системи	20
2.2 Модуль сенсорики та збору даних.	21
2.3 Центральний керуючий модуль.	21
2.4 Виконавчий модуль.	23
2.5 Модуль комунікації та взаємодії з користувачем	24
2.6 Постановка завдання на розробку	27
2.7 Вибір програмних засобів.	28
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ	30
3.1 Реалізація програмного забезпечення керуючого модуля	30
3.2 Проектування інтерфейсу користувача	32
3.3 Організація тестового стенду та методика верифікації	35
3.4 Результати експериментальних досліджень	39
3.5 Аналіз тестування	40
ВИСНОВКИ	42
ПЕРЕЛІК ПОСИЛАНЬ	43
ДОДАТКИ	44

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

GPIO – General-Purpose Input/Output, порт вводу/виводу загального призначення

IDE – Integrated Development Environment, інтегроване середовище розробки

IoT – Internet of Things, Інтернет речей

МК – Microcontroller, Мікроконтролер

ВСТУП

Контроль температури в приміщенні є однією з ключових умов створення безпечного, здорового та комфортного мікроклімату як для проживання, так і для роботи. Невідповідна температура — як надмірне охолодження, так і перегрів — може викликати погіршення самопочуття, зниження концентрації, втому, а також збільшення споживання енергоресурсів. Водночас підтримання стабільного температурного режиму вручну часто є неефективним та енергозатратним, особливо у великих або інтенсивно використовуваних приміщеннях.

Значна частка сукупного енергоспоживання припадає на житлово-комунальний сектор, зокрема на системи опалення, вентиляції та кондиціонування. Нераціональне використання енергії в цій галузі зумовлене переважним застосуванням застарілих систем керування, що характеризуються дискретним характером регулювання, низькою точністю та повною відсутністю адаптивності до динамічних змін зовнішніх умов та потреб споживачів.

У цьому контексті актуальним стає використання систем автоматичного керування температурою, які здатні зчитувати поточні параметри повітря, приймати рішення на основі заданих умов і керувати виконавчими пристроями, такими як обігрівачі або кондиціонери, без участі людини.

Імплементация Інтернету речей (IoT) є ключовим інструментом для переходу до систем, здатних до автономного моніторингу, аналізу та керування. Застосування даної концепції для автоматизації будівель дозволяє створювати інтелектуальні системи керування мікрокліматом, що забезпечують підтримання сталості мікрокліматичних параметрів на заданому рівні, та оптимізацію енергоспоживання в реальному часі.

Практична цінність роботи полягає у створенні повністю функціонального прототипу, що може бути імплементований в існуючу інфраструктуру житлових та комерційних об'єктів для підвищення їхньої енергетичної ефективності та рівня комфорту. Розроблений пристрій є економічно вигідною альтернативою існуючим комерційним системам. Результати роботи можуть бути використані як основа для подальших науково-дослідних робіт у галузі IoT та для комерційного застосування

Сутність роботи полягає у розробці компактного, надійного та енергоефективного пристрою на базі мікроконтролера, що забезпечує можливість віддаленого параметричного налаштування та моніторингу через мережу Інтернет.

Мета кваліфікаційної роботи – підвищення ефективності та зменшення вартості пристрою автоматичного керування температурою у приміщенні за рахунок використання мікроконтролера.

Задачі роботи:

- розробка структурної та принципової електричної схеми пристрою;
- розробка програмного забезпечення для мікроконтролера;
- розробка інтерфейсу на базі хмарної платформи Blynk;
- тестування розробленої моделі пристрою.

1 ОГЛЯД ЛІТЕРАТУРИ

1.1 Мікроконтролер

В основі будь-якої сучасної системи автоматичного керування лежить центральний обчислювальний елемент, який виконує логіку всієї системи. Такі пристрої належать до класу вбудованих систем (embedded systems), оскільки вони є спеціалізованими комп'ютерними системами, інтегрованими в об'єкт керування для виконання визначених функцій. У контексті пристроїв для керування температурою, цим ключовим компонентом є мікроконтролер (МК). Його роль полягає в реалізації замкненого циклу керування, що є основою теорії автоматичного керування. Цей цикл є безперервною послідовністю логічно пов'язаних операцій, що забезпечують підтримання заданого стану системи.

Процес керування температурою починається зі збору даних, під час якого мікроконтролер безперервно або з заданою періодичністю опитує датчики температури, отримуючи об'єктивну інформацію про поточний стан керованого середовища. Цей етап є відправною точкою для будь-якого керуючого впливу, оскільки якість та актуальність вхідних даних безпосередньо визначають точність та доцільність подальшого регулювання. Після цього отримана цифрова інформація обробляється шляхом порівняння із заданими користувачем параметрами, такими як бажана температура та величина гістерезису. Для забезпечення надійності та відновлення роботи після збоїв живлення ці налаштування, як правило, зберігаються в енергонезалежній Flash пам'яті мікроконтролера. На основі результатів цього порівняння, мікроконтролер приймає рішення про необхідність корекції температури. Алгоритм програмного забезпечення, є ядром інтелекту системи та визначає, чи потрібно увімкнути обігрів, охолодження, чи залишити систему в режимі очікування. Далі МК формує та подає дискретні електричні сигнали на

виконавчі елементи, котрі, в свою чергу, комутують силові ланцюги, вмикаючи або вимикаючи нагрівачі, кондиціонери чи вентилятори. Одночасно з цими процесами мікроконтролер забезпечує функціонування людино-машинного інтерфейсу, виводячи поточну температуру, стан системи й налаштування на дисплей, а також обробляючи вхідні сигнали від кнопок або інших органів керування для забезпечення взаємодії з користувачем.

Ринок електронних компонентів пропонує широкий спектр мікроконтролерів. Класифікація МК здійснюється за розрядністю архітектури, обсягом пам'яті, набором периферійних модулів та енергоспоживанням.

Широке розповсюдження отримали 8-бітні мікроконтролери, зокрема родини Atmel AVR та Microchip PIC. Контролери AVR, що є основою класичних плат Arduino, таких як Uno, Nano та Mega, побудовані на базі RISC-архітектури, що забезпечує виконання більшості інструкцій за один такт. Вони пропонують чудовий баланс між продуктивністю, енергоспоживанням та простотою використання, що зробило їх ідеальним інструментом для освітніх цілей [9].

Однак сучасні вимоги до систем автоматизації, особливо в контексті концепції Інтернету речей, зумовлюють перехід до більш продуктивних 32-бітних архітектур. Контролери на базі ядер ARM Cortex-M є промисловим стандартом для складних вбудованих систем, пропонуючи значно вищу продуктивність та розширені периферійні можливості.

ESP32 є серією бюджетних систем-на-кристалі (SoC) з низьким енергоспоживанням, розроблених компанією Espressif Systems, та виготовлених на потужностях компанії TSMC за 40-нм технологічним процесом. В основі чіпа лежить двоядерний 32-розрядний мікропроцесор Tensilica Xtensa LX6, що працює на тактовій частоті від 160 до 240 МГц та забезпечує продуктивність до 600 DMIPS. Двоядерна архітектура дозволяє ефективно розподіляти завдання: наприклад, один потік може бути виділений для підтримки мережевого стека, а інший – для виконання основного коду програми в реальному часі [3].

Наявність у ESP32 вбудованих модулів Wi-Fi (стандарту 802.11 b/g/n) та Bluetooth робить його підходящим рішенням для розробки IoT-пристроїв без необхідності використання зовнішніх комунікаційних модулів [1].

Основні технічні характеристики ESP32 наведені нижче.

Процесор:

- Xtensa двоядерний 32-розрядний LX6 мікропроцесор, що працює на 160 або 240 МГц і виконує до 600 DMIPS;

- пам'ять: 520 Кб пам'яті SRAM;

Периферійні інтерфейси:

- 12-розрядний АЦП до 18 каналів;

- 2 × 8-біт ЦАП;

- 4 × СВО;

- 2 × I2S для інтерфейсів;

- 2 × I2C інтерфейси;

- 3 × UART;

- SDIO/SPI підпорядкований контролер.

Бездротовий зв'язок:

- Wi-Fi: 802.11 b/g/n;

- Bluetooth: B4.2 BR EDR.

Головною особливістю цього мікроконтролера є те, що на його базі існують модульні SMT плати. Це дозволяє створювати прототипи, макети майбутнього пристрою на базі цього мікроконтролера без необхідності створення самої плати. Це значно спрощує та покращує розробку та реалізацію. Завдяки цьому ESP-32 набула великої популярності у світі [4].

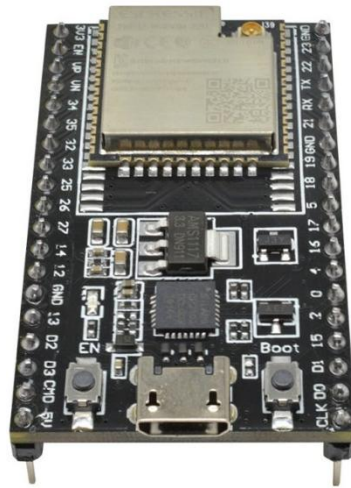


Рисунок 1.1 – Зовнішній вигляд ESP32 - DevKitCV4

Однією з таких SMT плат є ESP32 - DevKitC V4 (рис. 1.1). Ця плата включає у себе безпосередньо сам мікроконтролер ESP32, Micro-USB, кнопку перезавантаження мікроконтролера, світлодіоди, а також 30 портів, ці порти включають живлення 3.3В, заземлення, а також цифрові та аналогові порти (рис. 1.2)[2].

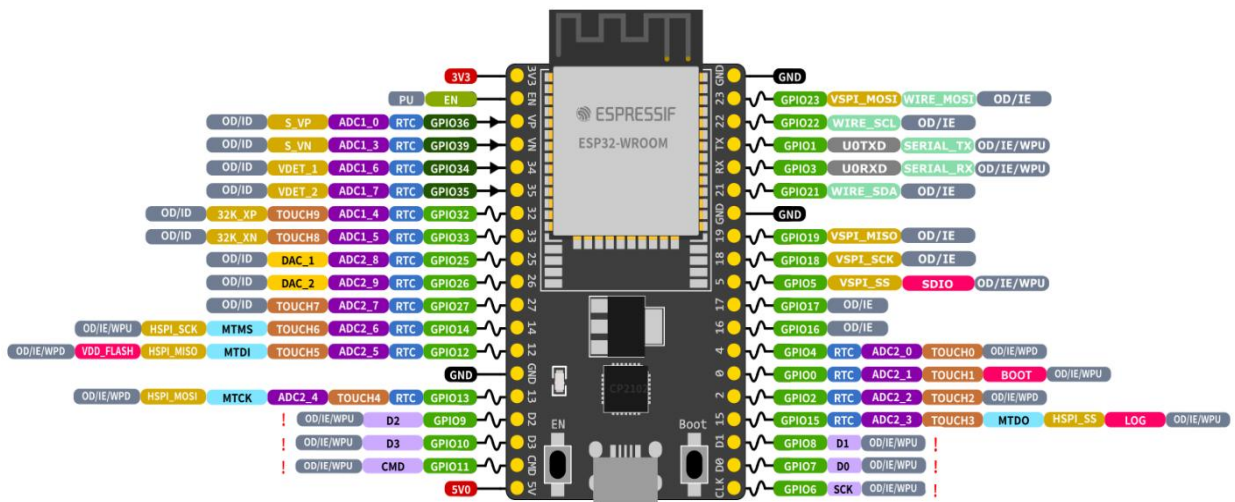


Рисунок 1.2 – Позначення портів на платі ESP32-DevKitCV4

Вибір даної плати був обумовлений наявністю на ній вбудованих елементів, що дозволило уникнути встановлення додаткових компонентів управління. Популярність модуля ESP32 - DevKitC V4 також відіграє важливу роль, оскільки це робить пристрій більш впізнаваним та простим для освоєння.

1.2 Аналіз технологій вимірювання температури

Ключовим елементом будь-якої системи клімат-контролю є датчик температури, оскільки точність та стабільність регулювання безпосередньо залежать від якості первинних вимірювань. Існує велике різноманіття сенсорів, що класифікуються за фізичним принципом дії та типом вихідного сигналу. Їх можна розділити на дві великі групи: аналогові та цифрові.

Аналогові датчики є одними з найпростіших. До них належать інтегральні схеми, як-от LM35. Вони генерують на виході напругу, прямо пропорційну температурі в градусах Цельсія, що спрощує розрахунки. Однак недоліком всіх аналогових датчиків є їхня чутливість до електромагнітних завад та наводок на довгих з'єднувальних проводах, а також необхідність використання якісного та правильно відкаліброваного аналого-цифрового перетворювача (АЦП) мікроконтролера.

Цифрові датчики є більш сучасним та надійним рішенням, оскільки вони виконують аналого-цифрове перетворення безпосередньо всередині свого корпусу і передають вже готове цифрове значення температури. Це робить їх значно стійкішими до завад. Одним з найпопулярніших цифрових датчиків є DS18B20 (рис. 1.3), який був обраний для даної роботи. Він використовує унікальний протокол 1-Wire, що дозволяє підключати декілька датчиків на одну трипровідну шину (дані, живлення, земля), причому кожен датчик має унікальний 64-бітний серійний номер для індивідуальної адресації. DS18B20 забезпечує високу точність вимірювань ($\pm 0.5^{\circ}\text{C}$) в широкому діапазоні температур (від -55°C до $+125^{\circ}\text{C}$) з програмованою роздільною здатністю від 9

до 12 біт. Для роботи з ним існують готові та добре перевірені бібліотеки, що значно спрощує інтеграцію [5].



Рисунок 1.3 – Цифровий датчик температури DS18B20

Інші популярні цифрові датчики, наприклад, DHT11 та DHT22, є комбінованими та вимірюють як температуру, так і вологість. Враховуючи, що завданням даної роботи є виключно керування температурою, вибір спеціалізованого датчика DS18B20 є найбільш обґрунтованим рішенням.

1.3 Модуль реле

Для безпечного керування потужними виконавчими пристроями, що працюють від мережі 220В (наприклад, обігрівачем чи кондиціонером), за допомогою низьковольтного сигналу від мікроконтролера, необхідно використовувати проміжний елемент. Найбільш поширеним, надійним та простим рішенням для таких завдань є електромеханічний релейний модуль. Для даного проекту оптимальним вибором є стандартний 1-канальний релейний модуль з живленням 5В та оптоізоляцією. Це готова плата, що містить всю необхідну схемотехніку для легкого та безпечного підключення до мікроконтролера.

Основні технічні характеристики релейного модуля:

- керуюча напруга (VCC): 5В постійного струму (DC);

- рівень керуючого сигналу (IN): ТТЛ (зазвичай, активний низький рівень);
- максимальне навантаження на контакти: 10А при 250В змінного струму або 10А при 30В постійного струму (DC);
- струм споживання котушки: приблизно 50-70 мА.

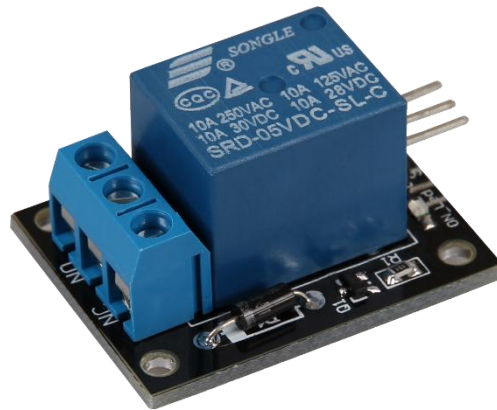


Рисунок 1.4 – Модуль реле KY-019

Одним з найпоширеніших та найдоступніших прикладів модуля, який ідеально підходить для вашого проекту, є 1-канальний модуль, що часто маркується як KY-019 (Рис. 1.4). Вони виробляються безліччю компаній, але зазвичай мають ідентичну або дуже схожу конструкцію та компоненти [8].

1.4 Принципи та алгоритми роботи пристрою

Функціонування автоматизованої системи керування температурою базується на тому, що вона утворює замкнений контур керування. Вона безперервно вимірює поточний стан керованого параметра (температуру в приміщенні), порівнює його з бажаним (заданим) значенням і на основі цієї різниці формує керуючий вплив для мінімізації відхилення.

Найпростішим, але водночас надійним та широко розповсюдженим для інерційних теплових систем є релейний алгоритм керування. Його суть полягає в тому, що виконавчий пристрій має лише два стани: "Увімкнено" або

"Вимкнено". Логіка прийняття рішень базується на двох порогах. Якщо поточна температура опускається нижче нижнього порогу, мікроконтролер подає сигнал на увімкнення нагрівача. Нагрівач працює на повну потужність, доки температура не підніметься вище верхнього порогу, після чого мікроконтролер його вимикає. Різниця між верхнім та нижнім порогами називається петлею гістерезису. Введення гістерезису є критично важливим, оскільки воно запобігає тактуванню – високочастотному перемиканню реле навколо заданої точки, що могло б призвести до швидкого механічного зносу та створення значних навантажень на електричну мережу. Ширина петлі гістерезису є компромісним параметром: вузька петля забезпечує вищу точність підтримки температури, але частіші перемикання, тоді як широка петля зменшує кількість спрацьовувань, але призводить до більших коливань температури.

Для цілей даної роботи, де об'єктом є приміщення з великою тепловою інерцією, надійності та простоти релейного керування з гістерезисом є цілком достатньо.

1.5 Інтеграція з технологіями Інтернету речей

Розвиток мікроелектроніки та бездротових комунікаційних технологій дозволив вийти за межі локальної автоматизації та перетворити прості термостати на інтелектуальні вузли глобальних мереж. Концепція Інтернету речей (IoT) передбачає об'єднання фізичних пристроїв у комп'ютерну мережу, де вони можуть збирати, обробляти та обмінюватися даними, взаємодіючи як між собою, так і з хмарними сервісами. У контексті контролю температури, імплементація IoT-підходу надає системі низку переваг.

Ключовою можливістю є віддалений моніторинг та керування. Завдяки підключенню пристрою до мережі Інтернет через вбудований Wi-Fi модуль мікроконтролера ESP32, користувач може за допомогою смартфона або веб-інтерфейсу з будь-якої точки світу в реальному часі перевіряти поточну

температуру в приміщенні, змінювати налаштування та отримувати сповіщення про стан системи [7].

Технічно це реалізується через взаємодію з хмарними IoT-платформами. У даній роботі для цих цілей було обрано платформу Blynk. Вона є яскравим прикладом PaaS (Platform as a Service) для IoT і надає готову інфраструктуру, що складається з трьох компонентів: хмарного сервера, що виступає посередником (брокером) і зберігає дані; бібліотеки для мікроконтролера, що інкапсулює всю складність мережевої взаємодії; та мобільного додатку з конструктором інтерфейсів. Такий підхід значно прискорює розробку, оскільки звільняє від необхідності створювати власний серверний бекенд та писати мобільний додаток з нуля [6].

1.6 Переваги підходу на відкритих платформах

На відміну від монолітних та дорогих комерційних систем, підхід до автоматизації клімату на базі відкритих апаратних платформ, яскравим представником яких є ESP32, пропонує низку переваг. Ці переваги роблять його особливо привабливим для реалізації освітніх, дослідницьких та індивідуалізованих проєктів, де гнучкість, вартість та можливість повного контролю над системою є пріоритетними.

До ключових переваг, в першу чергу, належить економічна доступність. Собівартість компонентів, таких як плати розробника на базі ESP32 та сумісних з ними датчиків і релейних модулів, є на порядок нижчою порівняно з брендовими "розумними" термостатами. Це суттєво знижує поріг входження для проведення експериментів, валідації нових алгоритмів та створення функціональних прототипів з мінімальними фінансовими вкладеннями. Така доступність робить технології "розумного дому" відкритими для всіх. Якщо раніше цим займалися переважно великі компанії, то тепер це можуть робити також у навчальних закладах, гуртках та просто окремі розробники.

Не менш важливою є виняткова гнучкість та можливість повної кастомізації. Відкрита архітектура надає розробнику абсолютний контроль над апаратною конфігурацією та програмним кодом, усуваючи обмеження "чорної скриньки", притаманні комерційним продуктам. Це дозволяє вільно обирати будь-які датчики, виконавчі механізми та інтерфейси, що найкраще відповідають специфічним потребам проекту, а не обмежуватися асортиментом одного виробника.

Простота освоєння та швидкість прототипування також відіграють значну роль. Використання фреймворку Arduino як високорівневої абстракції над апаратним забезпеченням ESP32 та наявність готових середовищ розробки, як-от Arduino IDE, дозволяють навіть розробникам з базовими навичками програмування швидко досягати результатів [9].

Крім того, системи, побудовані на відкритих компонентах, є легко масштабованими. Функціонал розробленого пристрою можна безперешкодно розширювати в майбутньому. Модульний принцип побудови дозволяє поступово нарощувати складність і можливості системи без необхідності повної її переробки.

Нарешті, однією з найбільших переваг є потужна підтримка з боку величезної світової спільноти розробників та ентузіастів. Велика популярність платформи ESP32 призвела до створення великої кількості форумів, навчальних посібників, відео-уроків, готових проектів та бібліотек з відкритим кодом. Це означає, що при виникненні будь-якої технічної проблеми з високою ймовірністю можна знайти готове рішення або отримати кваліфіковану допомогу від спільноти.

Сукупність цих факторів робить вибір відкритої платформи не просто можливим, а й виправданим для розробки індивідуалізованого, економічно ефективного та розширюваного пристрою, який можна точно оптимізувати під унікальні вимоги конкретного технічного завдання.

2 СТРУКТУРА ПРИСТРОЮ

2.1 Архітектура та функціональні модулі системи

Архітектура системи автоматичного керування температурою є розподіленою та базується на клієнт-серверній моделі взаємодії. Система складається з трьох основних рівнів: фізичного рівня (апаратне забезпечення), рівня керування (вбудоване програмне забезпечення) та рівня взаємодії з користувачем (хмарна платформа та мобільний додаток).

Для детального опису функціональну структуру системи доцільно декомпонувати на чотири логічно завершені модулі, кожен з яких виконує чітко визначені завдання (рис. 2.1).

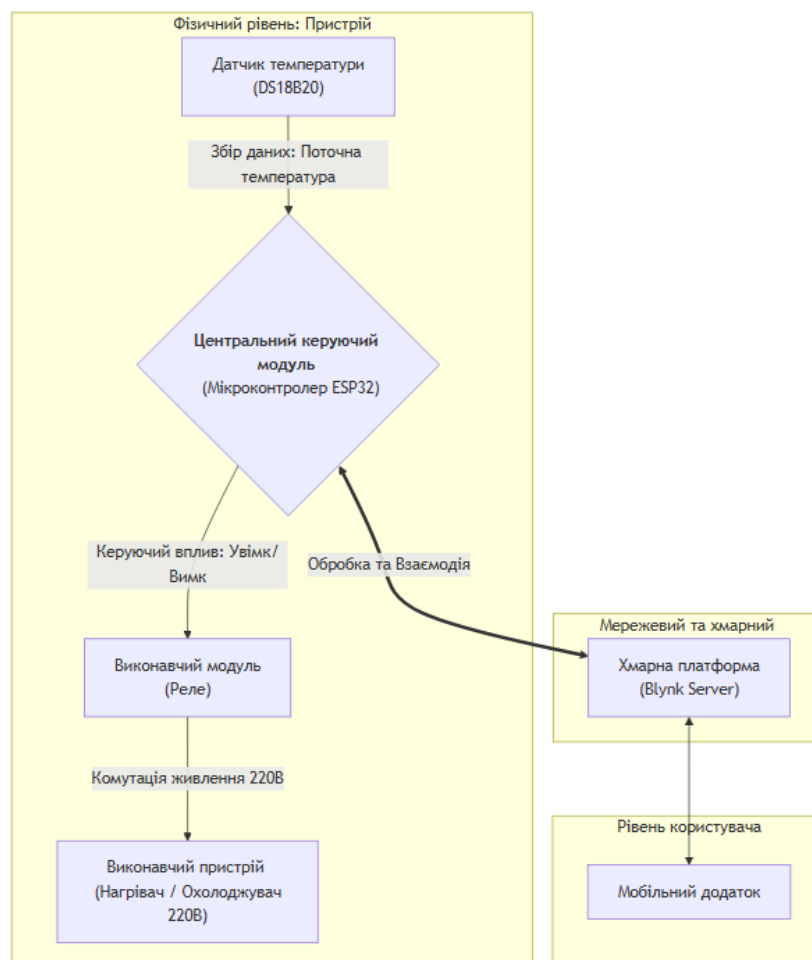


Рисунок 2.1 – Функціонально-структурна схема системи

2.2 Модуль сенсорики та збору даних

Цей модуль є первинною ланкою в ланцюгу отримання інформації про стан об'єкта керування. Його головним завданням є безперервне або дискретне вимірювання ключового фізичного параметра – температури повітря в приміщенні. Виміряна аналогова величина перетворюється на цифровий сигнал, придатний для подальшої обробки центральним керуючим модулем. У даній роботі функцію цього модуля виконує високоточний цифровий датчик температури DS18B20. Важливими вимогами до цього модуля є точність, стабільність показників, низьке енергоспоживання та наявність стандартизованого цифрового інтерфейсу для мінімізації впливу завад при передачі даних.

2.3 Центральний керуючий модуль

Центральний керуючий модуль (ЦКМ) є ядром архітектури розробленої системи, виконуючи функції обчислювального та керуючого центру. Даний модуль є апаратно-програмним комплексом, який відповідає за отримання, обробку та аналіз вхідних даних, реалізацію керуючої логіки, а також за координацію роботи всіх інших модулів системи. Апаратною основою цього модуля є мікроконтролерна плата ESP32 DevKitC V4.

Основні функції, що реалізуються центральним керуючим модулем, є взаємопов'язаним циклом операцій. Процес починається з отримання цифрових даних про температуру від модуля сенсорики. Для цього ЦКМ, використовуючи один зі своїх портів вводу/виводу загального призначення (GPIO), реалізує програмну логіку протоколу 1-Wire для взаємодії з датчиком DS18B20. Процесор мікроконтролера виконує послідовність команд для ініціалізації шини, адресації конкретного датчика та зчитування масиву байтів, що містять інформацію про температуру.

Наступним кроком є зберігання у внутрішній пам'яті заданих користувачем параметрів. Під час активної роботи системи цільова температура та поточний режим роботи зберігаються у швидкій статичній оперативній пам'яті (SRAM) об'ємом 520 Кб, що забезпечує миттєвий доступ до них для керуючого алгоритму. Для збереження налаштувань між сеансами роботи або після збоїв живлення використовується енергонезалежна флеш-пам'ять, яка також наявна в модулях ESP32.

Ключовою функцією центрального керуючого модуля є реалізація алгоритму керування. Ця обчислювальна задача виконується потужним двоядерним 32-розрядним мікропроцесором Xtensa LX6, що працює на частоті до 240 МГц. Процесор порівнює поточну температуру, отриману від сенсора, із цільовою температурою, що зберігається в пам'яті, та на основі логіки гістерезису приймає рішення про необхідність увімкнення чи вимкнення виконавчого модуля. Завдяки високій продуктивності (до 600 DMIPS), виконання цього алгоритму є практично миттєвим.

Крім того, центральний модуль здійснює керування комунікаційним модулем для обміну даними з хмарним сервісом. Процесор ESP32 напряму управляє вбудованим радіомодулем, що підтримує стандарти Wi-Fi 802.11 b/g/n та Bluetooth. Він виконує весь стек мережевих протоколів TCP/IP та логіку бібліотеки Blynk для встановлення з'єднання, передачі телеметрії та прийому команд від користувача.

Архітектура програмного забезпечення ЦКМ забезпечує автономну роботу системи у випадку тимчасової втрати зв'язку з мережею Інтернет. Якщо з'єднання з сервером Blynk втрачено, керуючий алгоритм продовжує виконуватися локально, використовуючи останні збережені налаштування. Це гарантує безперебійне підтримання мікроклімату навіть за умов нестабільного інтернет-з'єднання. Апаратна плата ESP32 DevKitC V4 є вдалим вибором, оскільки вона надає не тільки сам мікроконтролер, але й усі необхідні елементи для його функціонування.

2.4 Виконавчий модуль.

Виконавчий модуль є кінцевою, силовою ланкою системи автоматичного керування. Його призначення полягає у фізичній реалізації керуючого впливу, що генерується центральним модулем, з метою цілеспрямованої зміни стану об'єкта керування. У даному випадку модуль отримує низькорівневі логічні сигнали низької напруги від мікроконтролера та перетворює їх на дію – замикання або розмикання високовольтного силового кола, що безпосередньо змінює температуру в приміщенні..

У даній роботі в якості комутаційного елемента застосовано електромагнітне реле, реалізоване у вигляді готового модуля. Вибір реле обумовлений його здатністю надійно комутувати значні струми та напруги, а також, що є критично важливим, забезпечувати повну гальванічну розв'язку. Це означає, що між низьковольтною керуючою частиною, до якої підключений мікроконтролер, та високовольтними силовими колами відсутній будь-який прямий електричний контакт. Таке розділення досягається за рахунок того, що керуючий сигнал активує електромагнітну котушку, а вже її магнітне поле механічно замикає незалежні силові контакти. Цей принцип гарантує високий рівень безпеки системи, захищаючи чутливу електроніку мікроконтролера та самого користувача від небезпечної напруги.

В ролі виконавчого пристрою може виступати будь-який сумісний за потужністю електричний прилад, призначений для нагрівання або охолодження. До таких приладів належать електричні конвектори, інфрачервоні панелі, системи електричної теплої підлоги, масляні обігрівачі або кондиціонери, що працюють в режимі охолодження. Універсальність системи полягає в тому, що реле функціонує як звичайний вимикач. Прилад підключається до мережі 220В через нормально розімкнені контакти реле, і

доки максимальний струм споживання приладу не перевищує номінальні характеристики реле система може ефективно ним керувати.

2.5 Модуль комунікації та взаємодії з користувачем

Модуль комунікації та взаємодії з користувачем є ключовою ланкою, що перетворює локальний автономний пристрій на повноцінний вузол мережі Інтернету речей (IoT). Його основне призначення полягає у забезпеченні надійного двостороннього каналу зв'язку між системою та зовнішнім світом, а також у наданні користувачеві інтуїтивно зрозумілого інтерфейсу для віддаленого моніторингу та керування. Цей модуль реалізує обмін даними між центральним керуючим модулем (ESP32) та хмарною платформою, передаючи телеметричну інформацію (поточна температура, стан реле) від пристрою до користувача та керуючі команди (зміна цільової температури, увімкнення/вимкнення) у зворотному напрямку. Архітектурно цей модуль складається з двох нерозривно пов'язаних частин: апаратної та програмно-хмарної.

Апаратна складова цього модуля реалізована на базі інтегрованого у мікроконтролер ESP32 Wi-Fi-контролера. Він виконує функцію мережевого інтерфейсу (NIC), забезпечуючи підключення пристрою до точки доступу (Wi-Fi роутера) та подальший вихід в Інтернет. Програмне забезпечення, що працює на ESP32, ініціює процес сканування доступних мереж, проходить автентифікацію за допомогою заданих ідентифікатора мережі (SSID) та пароля, і отримує IP-адресу через протокол DHCP. Інтеграція цього функціоналу безпосередньо в систему-на-кристалі ESP32 є однією з її основних переваг, оскільки це дозволяє відмовитися від зовнішніх комунікаційних модулів, що значно спрощує кінцеву схемотехніку пристрою, зменшує його енергоспоживання та собівартість.

Програмно-хмарна складова представлена комплексною IoT-платформою Blynk. Вона виступає в ролі проміжного програмного забезпечення, що

абстрагує всю складність серверної інфраструктури та мобільної розробки. Платформа Blynk включає в себе хмарний сервер, який є центральним брокером повідомлень. Він відповідає за автентифікацію пристроїв за унікальними токенами, обробку та зберігання даних про стан пристрою, а також за маршрутизацію команд між апаратним забезпеченням та мобільним додатком. Іншою частиною є бібліотека Blynk для мікроконтролера, яка інкапсулює складні процеси встановлення та підтримки захищеного TCP-з'єднання з сервером. Нарешті, третім компонентом є мобільний додаток для операційних систем iOS та Android з графічним конструктором інтерфейсів. Завдяки цьому користувач через додаток може в реальному часі бачити поточну температуру, встановлювати бажане значення та вмикати/вимикати систему.

Таким чином, запропонована архітектура, що поєднує апаратні можливості ESP32 та програмно-хмарну інфраструктуру Blynk, є гнучкою, масштабованою та забезпечує чіткий розподіл функцій між модулями. Це створює надійну основу для функціонування системи та дозволяє легко реалізувати подальшу модернізацію, наприклад, шляхом додавання нових сенсорів або інтеграції з іншими хмарними сервісами.

Ефективність та надійність функціонування розробленої системи значною мірою залежить від обраних протоколів передачі даних та методів реалізації керуючого впливу.

Взаємодія між центральним керуючим модулем та хмарною платформою Blynk здійснюється за стандартним стеком протоколів TCP/IP поверх мережі Wi-Fi. Мікроконтролер ESP32 виступає в ролі клієнта, який встановлює та підтримує захищене з'єднання з сервером Blynk. Обмін даними відбувається за протоколом Blynk, який оптимізовано для передачі невеликих обсягів даних з низькою затримкою, що є типовим для IoT-застосувань.

Взаємодія з користувачем відбувається через мобільний додаток, який також є клієнтом сервера Blynk. Сервер виступає в ролі посередника, що

забезпечує синхронізацію стану між фізичним пристроєм та віртуальними елементами керування в мобільному додатку. Дані прив'язуються до так званих "віртуальних пінів" – абстрактних каналів для обміну інформацією будь-якого типу (числа, рядки, булеві значення). Такий підхід забезпечує гнучкість, оскільки не прив'язує логіку додатку до конкретних фізичних пінів мікроконтролера.

Взаємодія між керуючим модулем та датчиком температури DS18B20 відбувається за допомогою спеціалізованого протоколу 1-Wire. Кожен датчик має унікальний 64-бітний серійний номер, що дозволяє підключати декілька таких сенсорів на одну шину.

В основі системи лежить класичний метод релейного (двопозиційного) керування з затримкою. Це найпростіший і найпоширеніший тип регулятора для систем, де об'єкт керування має значну інерційність, а висока точність підтримки параметра не є критичною.

Нехай $T_{current}$ – поточна температура, T_{set} – задана користувачем цільова температура, а H – значення гістерезису (наприклад, $1\text{ }^{\circ}\text{C}$). Тоді логіка роботи алгоритму для керування нагрівачем буде наступною:

- увімкнення нагрівача: якщо $T_{current} < (T_{set} - H)$, керуючий модуль подає сигнал на увімкнення реле;
- вимкнення нагрівача: якщо $T_{current} > (T_{set} + H)$, керуючий модуль подає сигнал на вимкнення реле;
- зона нечутливості: якщо $(T_{set} - H) < T_{current} < (T_{set} + H)$, стан реле не змінюється.

Використання гістерезису є обов'язковим. Без нього в точці $T_{current} = T_{set}$ відбувалися б безперервні часті перемикання реле що призвело б до швидкого механічного зносу реле та нестабільної роботи виконавчого пристрою. Ширина гістерезису є компромісом між точністю підтримки температури та частотою перемикань.

Для підвищення надійності системи передбачено декілька механізмів. По-перше, у випадку обриву зв'язку з Wi-Fi або сервером Blynk, керуючий модуль продовжує автономно працювати згідно з останніми отриманими налаштуваннями T_set. По-друге, після відновлення живлення або мережевого з'єднання, пристрій автоматично перепідключається до сервера та синхронізує свій стан.

2.6 Постановка завдання на розробку

На основі проведеного аналізу та визначеної архітектури було сформульовано наступний перелік конкретних завдань на апаратно-програмну розробку:

- розробити принципову електричну схему: на основі обраних компонентів створити детальну схему підключення мікроконтролера, датчика температури, релейного модуля та інших елементів, враховуючи розпіновку, номінали резисторів та живлення;

- реалізувати програмне забезпечення. Написати мовою програмування C++ у середовищі Arduino IDE програмний код для мікроконтролера ESP32, який забезпечує: ініціалізацію апаратних компонентів, підключення до мережі Wi-Fi та сервера Blynk, періодичне опитування датчика температури DS18B20, реалізацію логіки двопозиційного керування з гістерезисом, обробку команд, що надходять з мобільного додатку, відправку даних про поточну температуру та стан системи в мобільний додаток;

- спроектувати та налаштувати людино-машинний інтерфейс: створити проект у мобільному додатку Blynk, розмістити та налаштувати необхідні віджети (дисплей, слайдер, перемикач, індикатор) та прив'язати їх до відповідних віртуальних пінів;

- провести верифікацію розробленої системи шляхом моделювання у віртуальному середовищі Wokwi. Розробити тестові сценарії для перевірки

коректності роботи пристрою в різних режимах та підтвердження його відповідності функціональним вимогам;

– проаналізувати отримані результати: оцінити точність підтримки температури, стабільність роботи системи, час реакції на команди користувача.

2.7 Вибір програмних засобів

Для написання прошивки центрального керуючого модуля було обрано мову програмування C++, яка є стандартом для розробки вбудованих систем, оскільки надає високий рівень контролю над апаратними ресурсами та забезпечує необхідну продуктивність. Для спрощення та прискорення процесу розробки було використано фреймворк Arduino для ESP32. Цей фреймворк є високорівневою абстракцією, що надає велику кількість готових бібліотек для роботи з периферією, такою як Wi-Fi, протокол 1-Wire та порти GPIO, що дозволяє розробнику концентруватися на прикладній логіці.

В якості інтегрованого середовища розробки (IDE) для даного проекту було обрано Arduino IDE. Вибір обумовлений простотою даного середовища та низьким порогом входження, що робить його оптимальним для швидкого прототипування та реалізації проектів в рамках навчального процесу.

Arduino IDE є самодостатнім рішенням, що включає текстовий редактор, компілятор та інструменти для завантаження прошивки і моніторингу послідовного порту «з коробки». Це дозволяє розробнику сконцентруватися безпосередньо на написанні програмного коду та логіці роботи пристрою, не витрачаючи час на складне налаштування робочого оточення та залежностей. Вбудовані менеджери плат та бібліотек дозволяють у кілька кліків додати підтримку мікроконтролера ESP32 та встановити всі необхідні бібліотеки.

Для реалізації функціоналу віддаленого керування та моніторингу було прийнято рішення використати спеціалізовану IoT-платформу Blynk. Цей вибір обумовлений тим, що Blynk є комплексним рішенням типу Platform-as-a-Service (PaaS), яке значно спрощує розробку IoT-проектів. Платформа надає готову

хмарну інфраструктуру для обміну даними, спеціалізовані бібліотеки для мікроконтролера, що інкапсулюють всю складність мережевої взаємодії, та мобільний додаток з візуальним конструктором інтерфейсів. Використання Blynk дозволило повністю уникнути необхідності розробки власного серверного бекенду та написання нативного мобільного додатку, що скоротило час розробки на порядок. Можливість створювати користувацький інтерфейс за допомогою віджетів і прив'язки їх до віртуальних пінів забезпечило швидке прототипування та високу гнучкість системи.

Важливим етапом життєвого циклу розробки є верифікація та тестування програмного забезпечення. Для цього було використано сучасний онлайн-симулятор Wokwi. Перевага Wokwi полягає в його спеціалізації на моделюванні мікроконтролерних систем, зокрема повноцінній підтримці архітектури ESP32. Ключовою можливістю симулятора, що визначила його вибір, є унікальна функція симуляції Wi-Fi з'єднання з виходом у реальну мережу Інтернет. Це дозволило на етапі програмної розробки провести повне тестування взаємодії прошивки з хмарним сервером Blynk, перевірити коректність відправки даних з віртуального датчика та отримання команд з віртуального мобільного додатку. Використання Wokwi дало змогу виявити та виправити логічні помилки в алгоритмі керування ще до завантаження коду в фізичний пристрій, що знизило ризики та прискорило процес розробки.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Реалізація програмного забезпечення керуючого модуля

Програмне забезпечення для центрального керуючого модуля, що є ядром всієї системи автоматизації, було розроблено мовою програмування C++ з використанням фреймворку Arduino. В якості інтегрованого середовища розробки (IDE) було обрано PlatformIO, що інтегрується у Visual Studio Code, оскільки воно надає професійні інструменти для керування проектом. Розроблений програмний код має чітку модульну структуру, що значно покращує його читабельність, спрощує налагодження та подальший супровід. Він складається з блоку конфігурації та оголошень, блоку ініціалізації, основного неблокуючого циклу та набору спеціалізованих функцій, що реалізують ключову логіку роботи пристрою.

Програмний код починається з блоку препроцесорних директив та оголошення глобальних змінних. На цьому початковому етапі за допомогою директив `#define` визначаються всі необхідні константи. До цих констант належать піни мікроконтролера, до яких підключені компоненти (наприклад, `ONE_WIRE_BUS` 4), параметри доступу до мережі Wi-Fi, унікальний аутентифікаційний токен для зв'язку з платформою Blynk, а також значення гістерезису. Разом з цим оголошуються глобальні змінні, призначені для зберігання поточного стану системи: `currentTemp`, `setTemp`, `systemMode` та `heatingMode`, `coolingMode`, `heatingState`, `coolingstate`. Використання глобальних змінних у даному випадку є виправданим, оскільки вони забезпечують обмін даними між різними функціями, що викликаються асинхронно. Тут же створюються екземпляри класів бібліотек, необхідні для подальшої роботи з периферією, зокрема об'єкти `OneWire`, `DallasTemperature` та `BlynkTimer`, що є основою для взаємодії з апаратною частиною та реалізації асинхронних операцій.

Наступним ключовим блоком є функція ініціалізації `setup()`, яка виконується автоматично один раз при кожному старті або перезавантаженні мікроконтролера. Вона відповідає за повне початкове налаштування системи та підготовку всіх її компонентів до роботи. Ця функція послідовно виконує ініціалізацію послідовного порту (`Serial.begin(115200)`) для виводу налагоджувальної інформації, значення 115200 означає, що інтерфейс буде налаштовано на передачу та прийом даних зі швидкістю 115 200 біт в секунду, після чого під керування реле конфігурується в режим виходу `pinMode(RELAY_PIN_HEATING, OUTPUT)` та встановлюється у початковий вимкнений стан `digitalWrite(RELAY_PIN_HEATING, LOW)`, що є важливою умовою безпеки. Далі ініціалізується бібліотека для роботи з датчиком температури `sensors.begin()` та запускається процедура підключення до мережі Wi-Fi і сервера Blynk за допомогою функції `Blynk.begin()`. Після успішного встановлення з'єднання відбувається налаштування таймерів для періодичного виклику функцій зчитування температури та застосування логіки керування. Використання таймерів, а не функції `delay()`, дозволяє уникнути блокування виконання основного циклу та забезпечує стабільну роботу з'єднання з сервером Blynk. На завершення процесу ініціалізації виконується синхронізація стану віджетів з сервером за допомогою `Blynk.syncVirtual()`, щоб пристрій одразу отримав актуальні налаштування, встановлені користувачем раніше.

Основний цикл програми `loop()` виконується безперервно і реалізований за неблокуючим принципом, що є критично важливим для стабільності роботи мережевих функцій. Його єдиним завданням є виклик двох функцій: `Blynk.run()` для підтримки з'єднання з сервером та обробки всіх вхідних і вихідних даних, та `timer.run()` для перевірки та запуску запланованих за таймером завдань. Така структура є високоефективною, оскільки основна логіка винесена в асинхронні функції. За таймером, кожні кілька секунд, викликається функція збору даних `readTemperature()`. Вона інкапсулює повний цикл роботи з датчиком: надсилає запит на вимірювання температури датчику DS18B20, зчитує отримане

значення, перевіряє його на коректність (наприклад, на помилку `DEVICE_DISCONNECTED_C`), оновлює відповідну глобальну змінну та надсилає дані на віртуальний пін V0 для відображення в мобільному додатку. Також за таймером працює ядро системи – функція `applyControlLogic()`. Вона реалізує алгоритм релейного керування з гістерезисом. Функція аналізує глобальні змінні `systemMode`, `currentTemp` та `setTemp`, приймає рішення про необхідний стан нагрівача, що зберігається у змінній `heatingState`, керує відповідним GPIO піном для фізичного вмикання/вимикання реле.

Для обробки команд, що надходять від користувача, використовуються спеціальні функції-колбеки (`callbackfunctions`) формату `BLYNK_WRITE(Vx)`. Це потужний механізм бібліотеки `Blynk`, що працює на основі подій. Ці функції автоматично викликаються щоразу, коли користувач взаємодіє з відповідним віджетом у мобільному додатку. Наприклад, функція `BLYNK_WRITE(V1)` спрацьовує при зміні положення слайдера цільової температури, після чого вона отримує нове значення за допомогою `param.asFloat()` та записує його в глобальну змінну `setTemp`. У свою чергу, функція `BLYNK_WRITE(V2)` спрацьовує при натисканні на перемикач ON/OFF, отримуючи значення (0 або 1) через `param.asInt()` та оновлюючи глобальну змінну `systemMode`, що змінює режим роботи всієї системи.

3.2 Проектування інтерфейсу користувача

Інтерфейс користувача, який забезпечує взаємодію людини з розробленим апаратно-програмним комплексом, реалізовано у вигляді мобільного додатку на базі IoT-платформи `Blynk`. Процес проектування зводиться до конфігурації проекту в хмарному середовищі `Blynk` та візуального конструювання дашборду з готових елементів (віджетів) на робочому просторі додатку.

Концепція проекту в `Blynk` базується на двох ключових сутностях: шаблону `Template` та пристрою `Device`. Шаблон виступає в ролі

централізованого креслення або "цифрового двійника" для однотипних пристроїв. Саме в шаблоні визначається назва пристрою, тип апаратного забезпечення, у даному випадку ESP32, та набір каналів для обміну даними. Datastream є абстракцією над віртуальним піном, що визначає не лише його номер, а й тип даних (наприклад, Double для температури, Integer для режиму роботи), одиниці виміру (°C), діапазон допустимих значень та значення за замовчуванням. Такий підхід створює чіткий "контракт даних" між апаратною частиною та хмарою. Після створення шаблону на його основі створюється конкретний екземпляр пристрою, кожен з яких отримує унікальний 64-символьний токен автентифікації BLYNK_AUTH_TOKEN. Цей токен використовується у прошивці мікроконтролера для безпечної ідентифікації та авторизації пристрою на сервері.

Для реалізації необхідного функціоналу та його налаштування було використано наступний набір віджетів, кожен з яких прив'язаний до відповідного, попередньо створеного віртуального піна Datastream:

Першим елементом є віджет LabeledValue V0 (Марковане значення), який слугує основним засобом моніторингу. Його призначення – відображення поточної температури в реальному часі. Віджету присвоєно інформативний заголовок "Температура, °C" для зручності користувача. Він налаштований на отримання даних типу Double з віртуального піна V0 і автоматично оновлює своє значення щоразу, коли мікроконтролер надсилає нові дані за допомогою команди `Blynk.virtualWrite(V0, currentTemp)`. Форматування виводу налаштовано на один десятковий знак, що забезпечує достатню точність без перевантаження інтерфейсу зайвою інформацією.

Другим ключовим елементом є віджет Slider V1 (Повзунок), що виступає в ролі основного органу керування для встановлення бажаної (цільової) температури. Віджету присвоєно заголовок "Бажана температура". Він налаштований на відправку даних типу Double на віртуальний пін V1 при зміні його положення користувачем. Важливою особливістю є можливість

налаштування діапазону значень, який було встановлено від 5 до 30 °C з кроком в 1 °C. Це є елементом валідації вводу на стороні клієнта, що унеможливорює встановлення користувачем некоректних або небезпечних значень температури. Кожна зміна положення повзунка ініціює відправку нового значення на сервер, що, в свою чергу, викликає спрацювання відповідної функції BLYNK_WRITE(V1) на мікроконтролері.

Третім віджетом є Switch V2 (Перемикач), який реалізує функцію увімкнення та вимкнення системи. Цей віджет є елементом цифрового керування, що дозволяє змінювати бінарний стан системи. Він налаштований на відправку цілочисельних значень 0 (що відповідає стану OFF) та 1 (що відповідає стану ON) на віртуальний пін V2. Натискання на перемикач миттєво надсилає відповідне значення на сервер і далі на пристрій, де воно обробляється функцією BLYNK_WRITE(V2) та змінює режим роботи основного алгоритму керування. Це дозволяє користувачеві дистанційно активувати або деактивувати систему клімат-контролю.

Для забезпечення більш гнучкого та роздільного керування виконавчими пристроями, в інтерфейсі було реалізовано два додаткові віджети типу «Перемикач»). Перший перемикач, прив'язаний до віртуального піна V4, відповідає за активацію та деактивацію функції нагрівання. Він є елементом дискретного керування, що дозволяє користувачеві вибірково дозволяти або забороняти роботу нагрівального приладу, не вимикаючи при цьому всю систему. При його перемиканні на сервер надсилаються цілочисельні значення 1 (нагрівання дозволено) або 0 (нагрівання заборонено), які обробляються на мікроконтролері відповідною функцією-колбеком BLYNK_WRITE(V4). Ця функція встановлює відповідний програмний прапорець, що додатково перевіряється в основному алгоритмі керування перед прийняттям рішення про увімкнення реле нагрівача.

Другий перемикач, що працює через віртуальний пін V5, виконує аналогічну функцію, але для системи охолодження. Його активація та

деактивація також надсилає бінарні сигнали на пристрій, де вони обробляються функцією `BLYNK_WRITE(V5)`. Таке розділення керування нагріванням та охолодженням надає користувачеві значно більшу гнучкість у налаштуванні мікроклімату, дозволяючи, наприклад, повністю вимкнути нагрівання в літній період, залишивши активною лише функцію кондиціонування.

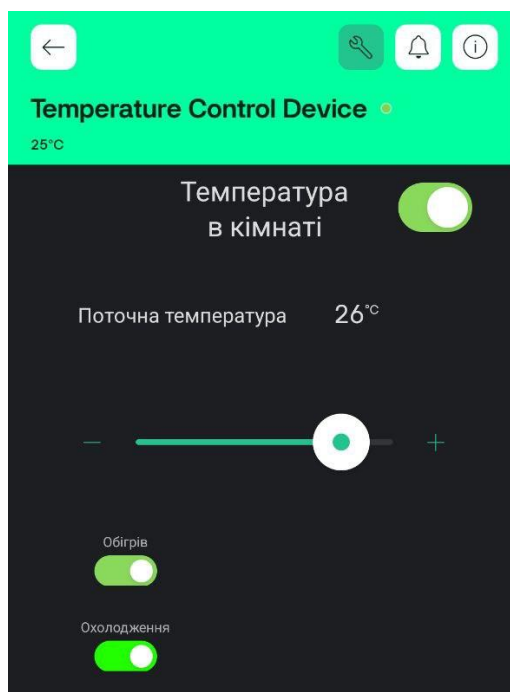


Рисунок 3.1 – Інтерфейс мобільного додатку

Такий інтерфейс (рис. 3.1) є інтуїтивно зрозумілим, не перевантаженим зайвими елементами та надає користувачеві повний контроль над основними функціями системи.

3.3 Організація тестового стенду та методика верифікації

Верифікація та тестування є невід'ємним та критично важливим етапом життєвого циклу розробки. У сучасній інженерній практиці значна частина цього процесу переноситься на етап комп'ютерного моделювання, що дозволяє провести глибоку перевірку відповідності розробленого програмного забезпечення функціональним вимогам ще до створення фізичного прототипу.

Основною метою даного етапу є валідація проектних рішень та алгоритмів, а також раннє виявлення та усунення потенційних логічних помилок у програмному коді.

Для проведення експериментальних досліджень було створено віртуальний тестовий стенд в онлайн-середовищі моделювання Wokwi. Даний симулятор є спеціалізованим інструментом для розробки та тестування вбудованих систем, що дозволяє з високим ступенем точності відтворювати роботу апаратно-програмних комплексів. Склад віртуального стенду був ретельно підібраний для створення цифрової копії, функціонально ідентичної до проектованого фізичного пристрою.

Основою стенду слугував центральний об'єкт тестування – віртуальна модель мікроконтролерної плати ESP32 DevKitC V4. Джерелом вхідних даних для системи виступав віртуальний цифровий датчик температури DS18B20, який виконував функцію зворотного зв'язку. Важливою перевагою симулятора є можливість інтерактивно змінювати показники цього датчика в реальному часі за допомогою повзунка, що дозволило імітувати зміну температури в приміщенні. В якості виконавчого механізму було використано віртуальний релейний модуль. Усі віртуальні компоненти були з'єднані між собою відповідно до принципової електричної схеми шляхом опису зв'язків у конфігураційному файлі `diagram.json`.

Зібраний віртуальний стенд являє собою повноцінну замкнену систему автоматичного керування, що дозволяє проводити об'єктивну оцінку її функціональності.

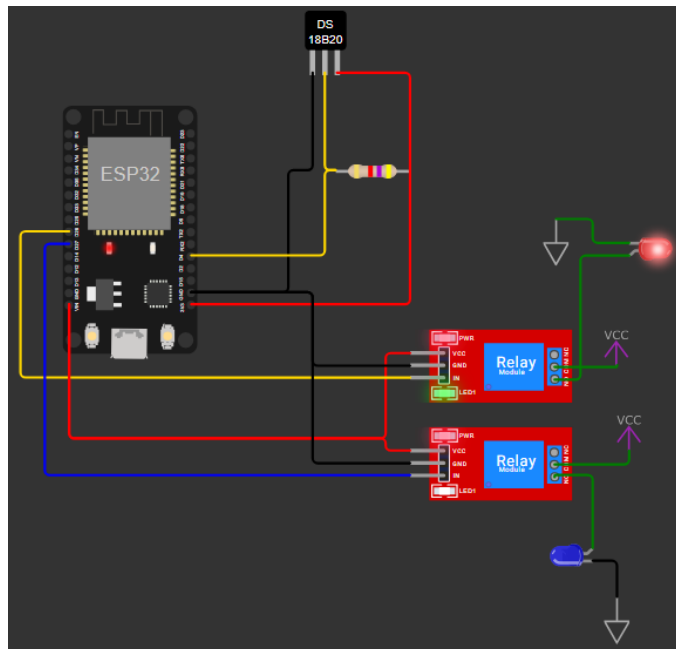


Рисунок 3.2 – Тестовий стенд для верифікації системи

Для комплексної перевірки працездатності системи було розроблено три тестові сценарії, що охоплюють усі ключові режими роботи.

Перший з цих сценаріїв, що полягав у тестуванні логіки підтримки температури, мав на меті перевірку коректності роботи алгоритму релейного керування з гістерезисом.

Хід тестування:

- у мобільному додатку встановлюється цільова температура, наприклад, $T_{set}=25.0^{\circ}\text{C}$ (при значенні гістерезису $H=1^{\circ}\text{C}$);
- система вмикається через додаток. Поточна кімнатна температура (наприклад, 22.0°C) є нижчою за поріг увімкнення ($25.0-1=24^{\circ}\text{C}$), тому реле має увімкнутися;
- за допомогою зовнішнього джерела тепла температура в зоні датчика плавно підвищується;
- фіксується момент, коли реле вмикається. Це має відбутися при досягненні температури $T_{current}26^{\circ}\text{C}$;

- джерело тепла прибирається, і датчик починає охолоджуватися природним шляхом;
- фіксується момент, коли реле знову вмикається. Це має відбутися при зниженні температури $T_{current} < 24$ °С.

За результатами тестування очікується, що розроблена система продемонструє здатність стабільно керувати виконавчим пристроєм. Це означає, що увімкнення та вимкнення навантаження має відбуватися чітко у моменти, коли поточна температура перетинає розраховані порогові значення, визначені цільовою температурою та величиною гістерезису.

В рамках другого тестового сценарію проводилося тестування функціоналу віддаленого керування. Його мета полягала у перевірці коректності обробки системою команд з мобільного додатку, а також в оцінці швидкості її реакції.

Хід виконання:

- послідовно через додаток змінюється цільова температура T_{set} на різні значення (вищі та нижчі за поточну);
- спостерігається реакція реле;
- кілька разів система вмикається та вимикається за допомогою віджета-перемикача.

Очікується, що система продемонструє високу швидкість реакції на команди з мобільного додатку, із затримкою, що не перевищує 1-2 секунди, та забезпечить повну відповідність фізичного стану реле поточному режиму роботи.

Метою третього тестового сценарію є аналіз надійності системи та її здатності повертатися до штатного режиму роботи. В рамках тесту перевіряється здатність пристрою до автоматичного відновлення функціоналу після критичних збоїв, таких як втрата живлення та розрив з'єднання з мережею.

Хід виконання:

- при увімкненій системі тимчасово вимикається живлення мікроконтролера, а потім вмикається знову;
- при увімкненій системі тимчасово вимикається Wi-Fi роутер, а потім вмикається знову.

Очікується, що за результатами тестування система продемонструє здатність до повного автономного відновлення. Після усунення змодельованого збою, як-от переривання живлення чи втрата мережевого зв'язку, пристрій повинен в автоматичному режимі відновити з'єднання з сервером Blynk, синхронізувати свої конфігураційні параметри та повернутися до штатного режиму функціонування без будь-якого втручання з боку користувача.

3.4 Результати експериментальних досліджень

За результатами проведених за вищеописаною методикою тестів було отримано дані, що підтверджують повну працездатність та коректність функціонування розробленого прототипу.

Під час експерименту з цільовою температурою $T_{set}=25.0^{\circ}\text{C}$ та гістерезисом $H=1^{\circ}\text{C}$ було зафіксовано наступну поведінку:

- вимкнення імітатора навантаження відбувалося стабільно при досягненні датчиком температури в діапазоні $26-26.1^{\circ}\text{C}$;
- увімкнення навантаження відбувалося при охолодженні датчика до температури $24-24.1^{\circ}\text{C}$.

Незначні відхилення пояснюються дискретністю вимірювань датчика та інерційністю теплових процесів. Отримані дані повністю підтверджують коректну реалізацію логіки керування.

Всі команди, надіслані з мобільного додатку Blynk, оброблялися системою коректно. Час реакції від моменту натискання кнопки в додатку до фізичного спрацювання реле складав в середньому 1-1.5 секунди, що є

задовільним показником для хмарної системи і є непомітним для користувача при керуванні інерційними процесами.

Після відключення та повторного включення живлення пристрій успішно перезавантажувався, автоматично підключався до мережі Wi-Fi протягом 5-10 секунд та синхронізував свій стан з сервером Blynk, продовжуючи роботу в тому режимі, який був встановлений до збою. Аналогічна поведінка спостерігалася і при тимчасовому зникненні мережі Wi-Fi. Це демонструє надійність програмної реалізації та її стійкість до типових збоїв.

3.5 Аналіз результатів тестування

Проведений комплекс експериментів дозволяє зробити обґрунтований висновок про успішне досягнення цілей розробки.

Розроблений прототип повністю реалізує заявлений функціонал. Він коректно вимірює температуру, передає її в мобільний додаток, отримує команди від користувача та на їх основі керує виконавчими пристроями. Логіка підтримки заданої температури працює стабільно та передбачувано.

Точність підтримки температури визначається двома факторами: точністю самого датчика ($\pm 1^\circ\text{C}$) та шириною петлі гістерезису (в нашому випадку $2H = 2.0^\circ\text{C}$). Таким чином, система забезпечує утримання температури з інструментальною похибкою датчика. Для більшості побутових завдань така точність є більш ніж достатньою.

Система продемонструвала високу стабільність роботи. Використання неблокуючого коду з таймерами забезпечує стабільний зв'язок з хмарним сервісом. Механізми автоматичного перепідключення гарантують відновлення працездатності після збоїв без втручання користувача. Гальванічна розв'язка силових та керуючих кіл забезпечує апаратну безпеку пристрою.

Інтерфейс, реалізований на платформі Blynk, є простим, наочним та інтуїтивно зрозумілим. Низький час реакції на команди забезпечує комфортну взаємодію з системою.

Отже, сукупність результатів експериментальних досліджень дозволяє зробити висновок, що розроблений апаратно-програмний комплекс є повністю працездатним, надійним та відповідає всім початковим вимогам, визначеним у постановці завдання. Прототип успішно вирішує поставлене завдання автоматизації контролю температури, забезпечуючи точне підтримання температури та надаючи зручний інструмент для віддаленого керування. Він є платформою, готовою як для практичного впровадження в реальних умовах експлуатації, так і для подальшого наукового вдосконалення.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи успішно досягнуто поставлену мету, яка полягала в розробці та створенні функціонального прототипу пристрою для автоматичного керування температурою в приміщенні з можливістю віддаленого керування

Для досягнення мети вирішено низку ключових завдань. Проведено аналіз існуючих рішень у сфері «розумного будинку», на основі якого обґрунтовано вибір апаратної платформи. Розроблено архітектуру та принципову електричну схему пристрою на базі мікроконтролера ESP32, цифрового датчика температури DS18B20 та релейного модуля. Створено програмне забезпечення мовою C++, що реалізує алгоритм релейного керування з гістерезисом для підтримки заданої температури. Також спроектовано користувацький інтерфейс у мобільному додатку на платформі Vlnk для забезпечення віддаленого моніторингу та керування.

Основним результатом роботи є повністю функціональний пристрій, що пройшов успішне тестування. Експериментальні дослідження підтвердили, що пристрій стабільно підтримує задану користувачем температуру, коректно реагує на команди з мобільного додатку та здатний до автономної роботи з відновленням функціоналу після збоїв живлення чи зв'язку.

Таким чином, усі поставлені завдання виконано, а розроблений пристрій довів свою працездатність та практичну цінність. Він є гнучким, бюджетним та енергоефективним рішенням для автоматизації клімат-контролю. Подальший розвиток проекту може включати вдосконалення алгоритму керування, інтеграцію з іншими системами розумного будинку через протокол MQTT та розробку власної друкованої плати для створення завершеного комерційного продукту.

ПЕРЕЛІК ПОСИЛАНЬ

1. Espressif Systems. ESP32-WROOM-32E & ESP32-WROOM-32UE Datasheet. Version 1.4. 2023. 42 с.
2. ESP32 від espressif. Технічна документація. Режим доступу до ресурсу: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
3. Espressif Systems. Espressif Technical Reference Manual. Version 5.1. – Espressif Systems, 2024. – 962 с.
4. Сайт компанії espressif. Режим доступу до ресурсу: <https://www.espressif.com/>
5. DS18B20 Programmable Resolution 1-Wire Digital Thermometer. Revision 10/21. 2021. 25 с.
6. Офіційна документація платформи Blynk. – Режим доступу до ресурсу: <https://docs.blynk.io/>.
7. Recommendation ITU-T Y.2060: Overview of the Internet of Things. – International Telecommunication Union (ITU), 2012. – 22 с.
8. Ningbo Songle Relay Co., Ltd. SRD-05VDC-SL-C Relay Datasheet. Version 3.0. Songle Relay, 2018. 2 с.
9. Arduino Language Reference— Режим доступу до ресурсу: <https://www.arduino.cc/reference/en/>.