

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Комп'ютерна система моніторингу температури в приміщенні»
08-54.БКР.027.00.000 ПЗ

Виконав: студент 4 курсу, групи 2СП-21Б
спеціальності 123 — «Комп'ютерна інженерія»
освітня програма — «Системне програмування»

Дяченко В.А. *Дяченко*

Керівник к.т.н., доц. каф. ОТ

Городецька Городецька О. С.

Рецензент доктор філософії, доц. каф. МБІС

Салієва Салієва О. В.

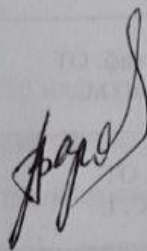
Азаров

Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О. Д.

« 16 » 06 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо—кваліфікаційний рівень бакалвр
Галузь знань — 12 Інформаційні технології
Спеціальність — 123 Комп'ютерна інженерія
Освітня програма — Комп'ютерна інженерія



ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

д.т.н.проф. _____ Азаров О. Д.

«21» березня 2025 року

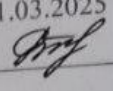
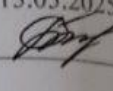
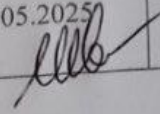
ЗАВДАННЯ **НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Дяченку Вячеславу Андрійовичу

- 1 Тема роботи: Комп'ютерна система моніторингу температури в приміщенні керівник роботи Городецька О.С. к.т.н., доцент кафедри ОТ, затверджено наказом вищого навчального закладу від «20» березня 2025 року №97.
- 2 Строк подання студентом роботи 13.05.2025
- 3 Вихідні дані до роботи: технічний опис мікропроцесорної системи протипожежного сповіщення, технології розробки: мова програмування C для Arduino, бібліотеки для роботи з серійним зв'язком та Ethernet, мобільні сервіси, технології для інтеграції з мобільними застосунками, цільова платформа системи – Arduino, середовище розробки – Arduino IDE.
- 4 Зміст розрахунково-пояснювальної записки: вступ; аналіз сучасних систем моніторингу температури в приміщенні; методи та засоби розробки комп'ютерної системи моніторингу; проектування та реалізація комп'ютерної системи моніторингу температури, висновок.
- 5 Перелік графічного матеріалу: схема електрична структурна, схема електрична функціональна, блок-схема алгоритму обробки даних.

Перелік ілюстративного матеріалу: прототип комп'ютерної системи
6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Консультанти роботи

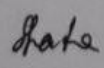
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-3	к.т.н., доц. каф. ОТ Городецька О.С.	21.03.2025 	13.05.2025 
Нормоконтроль	ас. каф. ОТ Швець С. І.	14.05.2025 	30.05.2025

7 Дата видачі завдання 21.03.2025 р.

8 Календарний план приведений в таблиці 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання	Підпис
1	Постановка задачі роботи	21.03.25	Виконав
2	Пошук матеріалів та інформаційних джерел	21.03.25—30.03.25	Виконав
3	Аналіз сучасних систем моніторингу температури в приміщенні	21.03.25—30.03.25	Виконав
4	Методи та засоби розробки комп'ютерної системи моніторингу	31.03.25—13.04.25	Виконав
5	Проектування та реалізація комп'ютерної системи моніторингу температури	14.04.25—27.04.25	Виконав
7	Оформлення пояснювальної записки та ілюстративного матеріалу	28.04.25—11.05.25	Виконав
8	Перевірка якості виконання бакалаврської кваліфікаційної роботи та усунення недоліків	13.05.25	Виконав

Студент 

Керівник роботи

Вячеслав ДЯЧЕНКО

к.т.н., доц. Оксана ГОРОДЕЦЬКА

АНОТАЦІЯ

УДК 621.3

Дяченко В. А. Комп'ютерна система моніторингу температури у приміщенні. Бакалаврська кваліфікаційна робота зі спеціальності 123 — Комп'ютерна інженерія, освітня програма — Системне програмування. Вінниця: ВНТУ, 2025. 85 с.

На укр. Мові. Бібліогр.: 15 назв; рис.: 20; табл.: 1.

У бакалаврській кваліфікаційній роботі розроблено комп'ютерну систему моніторингу температури у приміщенні. Обґрунтовано актуальність створення подібних систем у контексті підвищення енергоефективності будівель і забезпечення комфортних умов. Проведено аналіз існуючих систем моніторингу температури, досліджено методи та засоби реалізації мікропроцесорних систем. Розроблено структурну схему, описано програмну та апаратну частини системи, алгоритми обробки даних, а також результати тестування.

Ключові слова: комп'ютерна система, моніторинг температури, мікроклімат, автоматизація, енергоефективність, ESP32.

ABSTRACT

Diachenko V. A. Computer System for Indoor Temperature Monitoring. Bachelor's qualification work in specialty 123 — Computer Engineering, educational program — System Programming. Vinnytsia: VNTU, 2025. 85 p.

In English. References: 15 sources; figures: 20; tables: 1.

The bachelor's qualification work presents the development of a computer-based system for indoor temperature monitoring. The relevance of implementing such systems is substantiated in the context of improving building energy efficiency and ensuring comfortable living and working conditions. The work includes an analysis of existing temperature monitoring systems, an overview of methods and tools for implementing microprocessor-based systems, as well as the design and practical realization of the system. The structural diagram, hardware and software components, data processing algorithms, and testing results are described.

Keywords: computer system, temperature monitoring, indoor climate, automation, energy efficiency, ESP32.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ СУЧАСНИХ СИСТЕМ МОНІТОРИНГУ ТЕМПЕРАТУРИ В ПРИМІЩЕННЯХ.....	5
1.1 Аналіз існуючих комп'ютерних систем моніторингу температури	5
1.2 Мікропроцесорні рішення для керування температурним режимом	11
1.3 Огляд типів температурних датчиків і принципів їх роботи	16
2 МЕТОДИ ТА ЗАСОБИ РОЗРОБКИ КОМП'ЮТЕРНОЇ СИСТЕМИ МОНІТОРИНГУ.....	26
2.1 Вибір апаратної платформи для реалізації системи	26
2.2 Вибір температурних датчиків та особливості їх інтеграції.....	33
2.3 Вибір мови програмування та середовища розробки.....	41
3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ МОНІТОРИНГУ ТЕМПЕРАТУРИ.....	50
3.1 Розробка структурної схеми комп'ютерної системи.....	50
3.2 Розробка функціональної схеми	62
3.3 Програмування мікропроцесорного засобу.....	66
ВИСНОВКИ	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	70
ДОДАТОК А Технічне завдання	73
ДОДАТОК Б Протокол перевірки кваліфікаційної роботи.....	78
ДОДАТОК В Графічна частина.....	78
ДОДАТОК Г Ілюстративна частина	82
ДОДАТОК Д Лістинг програмного забезпечення.....	84

ВСТУП

У сучасному світі швидкий розвиток комп'ютерних технологій та автоматизації відкриває нові можливості для моніторингу та управління параметрами навколишнього середовища. Однією з ключових задач є забезпечення ефективного контролю температури в приміщеннях, що сприяє підвищенню комфорту користувачів та оптимізації енергоспоживання.

Актуальність роботи полягає в тому, що з розвитком комп'ютерних технологій та засобів автоматизації зростає потреба в точних і доступних системах моніторингу мікроклімату в приміщеннях. Особливе значення має контроль температури як одного з ключових параметрів, що впливають на комфорт, здоров'я людей та ефективність використання енергоресурсів.

Сучасні умови експлуатації вимагають удосконалення існуючих рішень шляхом розширення функціональності, зокрема за рахунок вимірювання не тільки температури, але й вологості, а також впровадження алгоритмів обробки даних, здатних адаптувати роботу системи до змін зовнішніх умов. Інтеграція таких можливостей у мікропроцесорні засоби забезпечує підвищення точності контролю, зниження енергоспоживання та підвищення автономності системи.

Це особливо актуально для житлових, офісних, навчальних та промислових приміщень, де стабільний мікроклімат є запорукою безпеки, комфорту та енергоефективності.

Об'єкт дослідження — процес моніторингу та контролю температури в приміщенні за допомогою комп'ютерних систем.

Предмет дослідження — методи та засоби обробки цифрових сигналів від датчиків температури, спрямовані на підвищення точності та ефективності системи.

Метою роботи є розширення функціональних можливостей комп'ютерної системи для регулювання температури у приміщенні за рахунок додавання комбінованого цифрового датчика, що дозволяє точніше оцінювати мікрокліматичні умови з вищою точністю.

Для досягнення мети необхідно вирішити такі **задачі**:

- провести аналіз існуючих систем моніторингу температури та визначити їх переваги та недоліки;
- вибрати апаратні компоненти та давачі;
- розробити структурну схему комп'ютерної системи;
- розробити функціональну схему засобу вимірювання температури;
- розробити програмне забезпечення для збору, обробки та візуалізації даних.

Практичне значення отриманих результатів полягає в можливості використання розробленого мікропроцесорного засобу для автоматизованого контролю параметрів мікроклімату в різних типах приміщень — житлових, офісних, навчальних або виробничих. Реалізація одночасного вимірювання температури та вологості дозволяє точніше визначати поточний стан середовища та приймати рішення щодо його регулювання.

1 АНАЛІЗ СУЧАСНИХ СИСТЕМ МОНІТОРИНГУ ТЕМПЕРАТУРИ В ПРИМІЩЕННЯХ

1.1 Аналіз існуючих комп'ютерних систем моніторингу температури

Сучасні комп'ютерні системи моніторингу температури відіграють важливу роль у забезпеченні комфортних умов у приміщеннях, ефективного використання енергоресурсів та автоматизації кліматичного контролю. Вони дозволяють у реальному часі відстежувати температурні показники, аналізувати їх зміни та керувати роботою опалювальних і охолоджувальних пристроїв.

Існуючі рішення можна класифікувати за рівнем складності та сферою застосування. Найпростіші локальні автономні системи базуються на мікроконтролерах типу Arduino чи ESP32, до яких підключаються цифрові датчики температури. Такі системи здатні відображати інформацію на вбудованих дисплеях або зберігати дані на зовнішніх носіях. Вони відрізняються простотою та низькою вартістю, проте мають обмежену функціональність і не підтримують дистанційний моніторинг.

Сучасний ринок пропонує різноманітні підходи до організації автономних систем температурного моніторингу, кожен з яких має свої унікальні характеристики та сфери застосування.

Серед найпоширеніших рішень виділяються спеціалізовані терморегулятори, такі як STC-1000 (рис. 1.1) або Inkbird ITC-308 (рис. 1.2). Ці готові модулі відрізняються простотою використання та не вимагають програмування, пропонуючи базове керування температурними режимами через аналогові інтерфейси. Вони особливо підходять для простих завдань контролю опалення чи охолодження, але мають обмежену функціональність. Основним обмеженням таких модулів є їхня жорстко задана логіка роботи, яка не підлягає зміні або гнучкому налаштуванню. Вони не підтримують обробку складних алгоритмів, не мають можливості зберігання даних, віддаленого доступу чи інтеграції з бездротовими мережами. Крім того, більшість таких терморегуляторів не дозволяє вимірювати додаткові параметри, наприклад, вологість або тиск, що обмежує їх

застосування лише найпростішими сценаріями.



Рисунок 1.1 — Терморегулятор STC-1000



Риснок 1.2 — Терморегулятор Inkbird ITC-308

Для складніших промислових завдань часто використовуються програмовані логічні контролери (ПЛК) типу Siemens LOGO! або Omron ZEN (рис. 1.3). Ці пристрої дозволяють реалізовувати складні алгоритми керування за допомогою графічних мов програмування та підтримують роботу з різними типами температурних датчиків. Їх основна перевага - надійність у промислових умовах, хоча вони вимагають спеціальних знань для налаштування.



Риснок 1.3 — ПЛК Omron ZEN

Особливий інтерес представляють системи з автономним логуванням даних на SD-карти. Такі рішення, засновані на мікроконтролерах ATmega або STM32 (рис. 1.4), дозволяють записувати температурні показники протягом тривалого часу без зовнішнього живлення. Вони особливо корисні для наукових досліджень чи моніторингу умов зберігання продуктів, де потрібен тривалий збір даних.

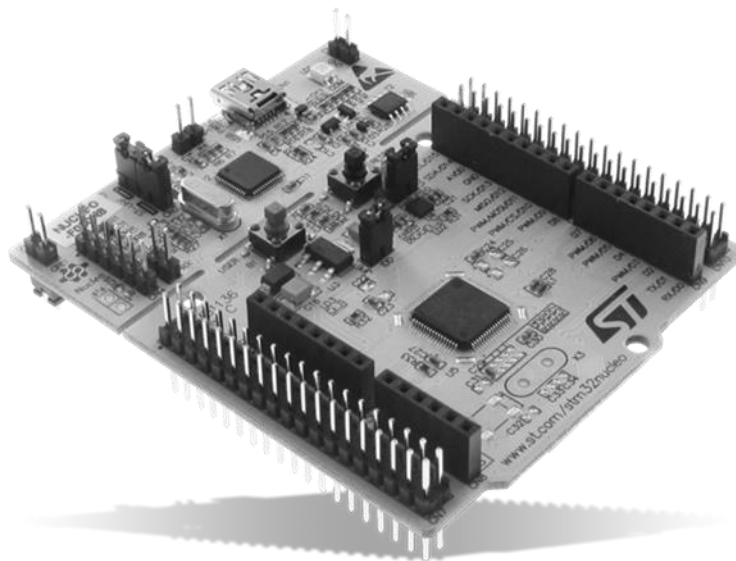


Рисунок 1.4 — Мікроконтролер STM32

Інноваційним напрямком є системи з електронними паперами (E-Ink дисплеями), які поєднують енергоефективність з чітким відображенням інформації. Такі рішення можуть працювати від батарей до року і особливо підходять для медичних застосувань або інформаційних панелей.

Традиційні аналогові системи на базі датчиків типу LM35 або PT100 (рис. 1.5) продовжують використовуватись у випадках, коли потрібна максимальна стійкість до електромагнітних перешкод або робота в екстремальних умовах. Вони відрізняються простотою та надійністю, хоча поступаються цифровим системам у точності.



Рисунок 1.5 — Датчик PT100

Вибір конкретного рішення залежить від комплексу факторів, включаючи необхідну точність вимірювань, умови експлуатації, тривалість автономної роботи та бюджет проекту. Для більшості побутових застосувань оптимальними залишаються системи на базі сучасних мікроконтролерів, тоді як для промислових умов часто вибирають спеціалізовані ПЛК або аналогові схеми. Кожен з цих підходів має свої переваги та обмеження, які необхідно враховувати при проектуванні системи моніторингу.

Більш досконалі мережеві системи, що входять до концепції Інтернету речей, використовують бездротові технології для передачі даних на сервери або хмарні платформи. Наприклад, рішення на базі ESP32 з інтеграцією у системи "розумного дому" дозволяють користувачам віддалено контролювати температурний режим

через мобільні додатки. Такі системи забезпечують більш гнучке керування та аналіз даних, але потребують додаткових витрат на підтримку хмарних сервісів.

Для промислових умов розроблені спеціалізовані SCADA-системи, які використовують високоточні давачі та програмовані логічні контролери. Вони здатні працювати у широкому діапазоні температур і забезпечують надійний моніторинг у складних виробничих умовах. Однак їх впровадження пов'язане з високими витратами та необхідністю кваліфікованого обслуговування.

Порівнюючи різні підходи, можна відзначити, що локальні системи є оптимальним вибором для простих завдань, тоді як IoT-рішення краще підходять для забезпечення дистанційного контролю. Промислові SCADA-системи незамінні у виробничих процесах, де потрібна висока точність та надійність.

Сьогодні спостерігається тенденція до впровадження інтелектуальних алгоритмів, зокрема машинного навчання, для прогнозування температурних змін та оптимізації роботи кліматичного обладнання. Перспективним напрямком є розвиток децентралізованих мереж на основі блокчейн-технологій, які можуть підвищити стабільність та безпеку систем моніторингу.

Таким чином, сучасні комп'ютерні системи контролю температури пропонують різні рівні функціональності – від простих автономних пристроїв до складних промислових комплексів. Вибір конкретного рішення залежить від вимог до точності, масштабованості та бюджету проекту. Майбутній розвиток цієї галузі пов'язаний із інтеграцією штучного інтелекту, автономних джерел живлення та вдосконалених методів аналізу даних.

Сучасні комп'ютерні системи контролю температури еволюціонують у напрямку створення інтелектуальних, адаптивних рішень, здатних ефективно функціонувати в різноманітних умовах експлуатації. Ця еволюція відбувається за рахунок поєднання традиційних підходів з інноваційними технологіями.

На сьогодні спостерігається чітка диференціація ринку за рівнем складності рішень. На одному кінці спектру знаходяться прості, але надійні автономні пристрої, призначені для вирішення вузькоспеціалізованих завдань. Вони зазвичай базуються на мінімалістичних апаратних платформах з обмеженим функціоналом,

але при цьому відрізняються високою енергоефективністю та стабільністю роботи. Типовими прикладами слугують терморегулятори для побутових холодильників або системи контролю температури в теплицях.

На протилежному кінці розташовані складні промислові комплекси, які інтегрують мережі датчиків, потужні засоби аналізу даних та розподілені системи керування. Такі рішення характеризуються:

- високою точністю вимірювань;
- можливістю обробки великих масивів даних у реальному часі;
- складними алгоритмами прогнозування та оптимізації;
- гнучкістю масштабування.

Особливий інтерес представляє проміжна категорія систем, які поєднують доступність автономних рішень з розширеними можливостями промислових комплексів. Це стало можливим завдяки використанню сучасних мікроконтролерних платформ з підтримкою бездротових технологій та хмарних сервісів.

Майбутній розвиток галузі визначається кількома ключовими тенденціями. Перш за все, це інтеграція методів штучного інтелекту, що дозволяє перейти від реактивного до проактивного керування температурними режимами. По-друге, впровадження автономних джерел живлення нового покоління значно розширює сферу застосування систем моніторингу. І по-третє, розвиток передових аналітичних інструментів відкриває нові можливості для виявлення складних залежностей та оптимізації енергоспоживання.

Ці зміни ведуть до формування нового покоління інтелектуальних систем температурного контролю, які поступово трансформуються з ізольованих пристроїв у повноцінні елементи "розумної" інфраструктури. При цьому особлива увага приділяється розробці уніфікованих стандартів та протоколів взаємодії, що є ключовою умовою для створення по-справжньому масштабованих рішень.

Для більш детального вивчення питання можна звернутись до державних стандартів ДСТУ EN 60751, що регулюють використання термоелектричних датчиків, а також до документації популярних платформ Arduino та ESP32. Крім

того, корисним є аналіз ринкових пропозицій від провідних виробників кліматичного обладнання.

1.2 Мікропроцесорні рішення для керування температурним режимом

Сучасні мікропроцесорні системи керування температурою представляють собою високоефективні технічні рішення, які знаходять застосування як у побутових, так і у промислових умовах. В основі цих систем лежать програмовані мікроконтролери, що забезпечують точний контроль температурних параметрів та автоматичне регулювання роботи кліматичного обладнання.

Архітектура сучасних систем ґрунтується на модульному підході, що дозволяє гнучко адаптувати рішення під конкретні завдання. Центральним елементом виступає мікропроцесорний блок обробки даних, до якого підключаються:

- високоточні цифрові давачі температури;
- мережеві інтерфейси для дистанційного моніторингу;
- виконавчі пристрої (реле, симістори, сервоприводи);
- інтерфейси користувача (дисплеї, панелі керування).

Програмне забезпечення цих систем реалізує складні алгоритми регулювання, включаючи:

- ПІД-регулятори для плавного керування;
- адаптивні системи з самонавчанням;
- мережеві протоколи для інтеграції у системи "розумного будинку";
- механізми резервування та відмовостійкості.

Ключові переваги мікропроцесорних систем порівняно з традиційними аналоговими рішеннями:

- висока точність - похибка регулювання до $\pm 0,1^{\circ}\text{C}$;
- гнучкість - можливість зміни алгоритмів без апаратних змін;
- масштабованість - легке додавання нових давачів і пристроїв;
- енергоефективність - оптимізовані алгоритми знижують споживання;
- діагностика - вбудовані системи моніторингу стану.

Перспективні напрями розвитку включають:

- інтеграцію технологій машинного навчання для передбачувального регулювання;
- використання бездротових сенсорних мереж LoRaWAN та NB-IoT;
- розробку енергоавтономних систем на основі energy harvesting;
- впровадження квантових давачів температури нової генерації.

Сфера застосування мікропроцесорних систем регулювання температури стрімко розширюється: від класичних HVAC-рішень до медичних приладів, харчових сховищ та промислового обладнання. Такі системи дозволяють забезпечити стабільність температурного режиму, оптимізувати витрати енергії та підвищити безпеку технологічних процесів. Їхня архітектура зазвичай включає мікроконтролер, цифрові сенсори (наприклад, DHT22, DS18B20), інтерфейси зв'язку (I2C, SPI, UART) та засоби виводу інформації — як-от LCD або OLED дисплеї.

Ключовим етапом є аналіз температурних даних, який виконується мікроконтролером за допомогою фільтрів та адаптивних алгоритмів. Завдяки цьому система не лише реагує на поточні зміни, а й аналізує їхню динаміку. Алгоритми враховують інерційність середовища, затримки виконавчих елементів і навіть можливі зовнішні завади. Управління може здійснюватися як простими реле, так і складними симісторними схемами або ШІМ-контролерами.

Програмна частина таких систем сьогодні набуває все більшої складності: вона включає реалізацію кількох рівнів логіки — від обробки вхідного сигналу до машинного навчання. Деякі сучасні реалізації здатні адаптувати поведінку в реальному часі, навчаючись на основі історичних даних. Це дозволяє прогнозувати зміну температури й вмикати або вимикати пристрої ще до того, як буде зафіксовано відхилення. Саме такі функції роблять систему дійсно інтелектуальною.

Одним з перспективних напрямів розвитку є використання енергоавтономних сенсорів, здатних працювати від сонячних батарей або п'єзоелементів. Такі рішення дозволяють створювати бездротові сенсорні мережі з

необмеженим часом роботи. Розвиваються і квантові сенсори, які забезпечують надвисоку точність, а також біонічні алгоритми, натхненні роботою природних систем терморегуляції — зокрема, шкіри людини або терморецепторів тварин.

Значний інтерес викликає концепція цифрових двійників — віртуальних моделей системи, які дозволяють прогнозувати її поведінку, симулювати зміну конфігурації та обирати оптимальні режими роботи до їх фактичного впровадження. Це особливо актуально у промисловості, де ціна помилки дуже висока. Такий підхід дозволяє мінімізувати ризики, скоротити час розгортання нових рішень і зменшити витрати на обслуговування.

Інтеграція з хмарними платформами типу Blynk, Firebase, ThingSpeak через сучасні протоколи (MQTT, HTTP, WebSocket) дозволяє створити розподілені мережі пристроїв, де кожен вузол взаємодіє з іншими і з центральною системою. Особливу увагу при цьому приділяють кібербезпеці: впроваджуються шифрування, автентифікація, захист від DoS-атак. Системи моніторингу стають частиною більших рішень — від "розумного дому" до промислового IoT.

Нарешті, програмна реалізація всієї логіки — від зчитування датчиків до відображення на дисплеї — виконується зазвичай на базі Arduino IDE або FLProg. Arduino надає широкий спектр бібліотек і прикладів, FLProg дозволяє будувати логіку блок-схемами без глибоких знань у програмуванні, що особливо корисно в навчальних цілях. Обидва інструменти мають активну спільноту, підтримку нових мікроконтролерів і можливість швидкого прототипування.

Таким чином, сучасна комп'ютерна система моніторингу температури — це багаторівнева, адаптивна, інтелектуальна платформа, здатна працювати автономно, взаємодіяти з користувачем, передавати дані в хмару, масштабуватись і залишатися актуальною в умовах стрімкої еволюції цифрових технологій.

Arduino IDE є текстовим середовищем розробки, що базується на спрощеній мові програмування, подібній до C++. Воно забезпечує підтримку великої кількості зовнішніх бібліотек, необхідних для взаємодії з температурними датчиками, дисплеями, модулями зв'язку та іншими компонентами. Розробник отримує змогу реалізувати повноцінну логіку контролю: від збору й фільтрації даних до

формування керуючих сигналів і передачі інформації на зовнішні пристрої. У свою чергу, середовище FLProg орієнтоване на візуальне програмування та дозволяє створювати логіку системи у вигляді блок-схем, що є зручним інструментом для швидкого прототипування та налагодження. Такий підхід не потребує глибоких знань програмування й значно спрощує розробку складних функціональних рішень, зокрема при роботі з панелями керування, таймерами або логічними елементами.

Особливу роль у програмному забезпеченні відіграють алгоритми стабілізації температури. Найефективнішим серед них є ПІД-регулювання, яке базується на аналізі поточної похибки між заданою температурою та фактичним значенням, а також на врахуванні динаміки зміни цієї похибки. Завдяки правильному налаштуванню таких алгоритмів можна досягти стабільної температури без перевищень або запізнь у реакції системи. В Arduino IDE для цього використовують відповідні бібліотеки, які дозволяють швидко реалізувати керування нагрівачами або вентиляторами з високою точністю.

Зібрані температурні дані можуть виводитися на дисплей — наприклад, LCD або OLED, де користувач бачить поточне значення температури, встановлену мету, а також стан роботи керуючих елементів. Крім того, можливе підключення до персонального комп'ютера через послідовний порт для подальшого аналізу даних або побудови графіків у спеціалізованому програмному середовищі, наприклад, у Python або LabVIEW.

У комплексі правильно реалізоване програмне забезпечення забезпечує точне керування кліматичними умовами, надає зручний інтерфейс для моніторингу та обслуговування, а також гарантує гнучкість для майбутнього розширення системи. Від якості програмної реалізації залежить не лише стабільність підтримання температурного режиму, а й загальна енергоефективність та зручність у практичному використанні.

Функціональні можливості сучасних мікропроцесорних систем включають:

- безперервний моніторинг температури в реальному часі;
- програмовані сценарії роботи для різних часів доби;

- можливість дистанційного керування через Wi-Fi або Bluetooth;
- автоматичне виявлення та усунення аварійних ситуацій;
- ведення журналу температурних показників для подальшого аналізу.

Мікропроцесорні системи регулювання температури мають ряд істотних переваг перед традиційними аналоговими рішеннями. Однією з ключових є висока точність, яку забезпечують цифрові сенсори в поєднанні з алгоритмічною обробкою. Це дозволяє системі реагувати на мінімальні відхилення температури, підтримуючи стабільний мікроклімат. У більшості випадків відхилення не перевищують частки градуса, що критично важливо для прецизійних застосувань.

Високий рівень гнучкості таких систем полягає в можливості програмної зміни логіки роботи без втручання в електроніку — це дозволяє швидко адаптувати систему до нових вимог, змін середовища або підключення додаткових модулів. Мікроконтролерна архітектура підтримує широкі комунікаційні інтерфейси (I2C, SPI, UART), що відкриває можливості інтеграції з іншими пристроями або мережею. Це робить систему масштабованою та придатною для використання в комплексних автоматизованих середовищах, включно з IoT.

Завдяки програмним засобам енергозбереження (як-от sleep-режими), мікропроцесорні системи також мають нижче енергоспоживання в порівнянні з аналоговими пристроями. Це робить їх ідеальними для застосування в автономних рішеннях, особливо у віддалених точках або мобільних системах. Крім того, цифрові системи мають більший ресурс надійності: вони не містять механічних елементів, що зношуються, і менш чутливі до температурних коливань, вібрацій чи старіння компонентів.

Ще однією сильною стороною мікропроцесорних рішень є програмна гнучкість, що дозволяє розробникам впроваджувати нові функції без фізичного втручання в пристрій. Наприклад, оновлення прошивки може додати нові режими керування, підтримку додаткових давачів або зміну протоколів зв'язку — усе це без змін у схемотехніці.

Перспективи розвитку систем полягають у вдосконаленні програмної логіки та впровадженні методів штучного інтелекту. Алгоритми машинного навчання вже

зараз дозволяють здійснювати адаптивне керування, прогнозувати зміни температури та автоматично підлаштовувати поведінку системи до змін зовнішніх умов. Такі функції підвищують ефективність і стабільність пристрою, знижують втрати енергії та підвищують комфорт користувача.

Зниження вартості компонентів завдяки масовому виробництву робить такі системи доступними для широкого кола застосувань — від простих побутових пристроїв до промислових і наукових установок. Сьогодні мікропроцесорні контролери можна знайти навіть у бюджетних терморегуляторах, вентиляторах, обігрівачах і розумних термостатах.

Вони знайшли широке застосування в найрізноманітніших сферах: HVAC, агросектор, медицина, зберігання продуктів, промислова автоматика, "розумний дім" та енергетика. Їх інтеграція в такі системи дозволяє не лише підтримувати потрібні параметри мікроклімату, а й дистанційно керувати обладнанням, отримувати аналітику та формувати звіти, що раніше було неможливим або економічно невиправданим із застосуванням аналогових рішень.

У рамках створення комп'ютерної системи моніторингу температури в приміщенні використання мікропроцесорної архітектури забезпечує найкращий баланс між функціональністю, вартістю, масштабованістю й зручністю впровадження. Це дає змогу реалізувати інтелектуальну систему, здатну адаптуватись до змінних умов, працювати автономно або у складі великої мережі, передавати дані у хмару, реагувати на зовнішні події, і при цьому залишатися доступною для реалізації навіть у невеликих проєктах. Усі ці характеристики роблять мікропроцесорні системи найбільш доцільним вибором у проєктах, де важливі надійність, енергоефективність і можливість подальшої модернізації.

1.3 Огляд типів температурних датчиків і принципів їх роботи

У сучасних системах моніторингу температури термістори займають важливе місце завдяки своїй простоті, компактності та здатності точно реагувати на незначні зміни температури. Ці датчики побудовані на основі напівпровідникових матеріалів, опір яких змінюється при зміні температури. Найбільш поширеними є

термістори з негативним температурним коефіцієнтом (NTC) (рис. 1.6), у яких електричний опір зменшується зі зростанням температури. Вони часто використовуються у побутовій техніці, електронних приладах, комп'ютерних системах охолодження, а також у медичних пристроях, де важливо швидко та точно виявляти зміни температурного середовища.

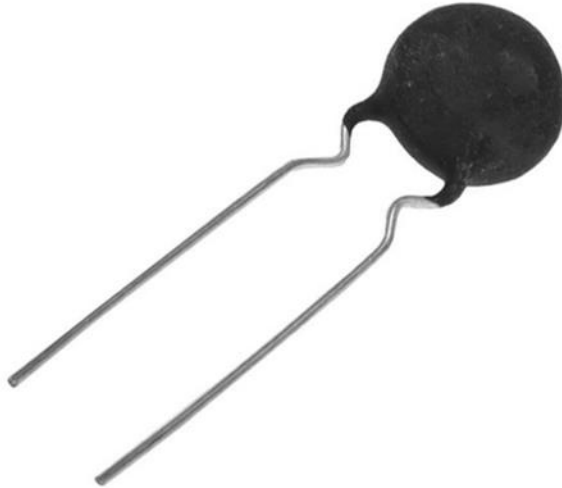


Рисунок 1.6 — Термістор з негативним температурним коефіцієнтом (NTC)

Менш поширеними, але також корисними є термістори з позитивним температурним коефіцієнтом (PTC), які демонструють протилежну поведінку — їхній опір зростає при підвищенні температури. Завдяки цій властивості вони здатні автоматично обмежувати струм, що робить їх корисними в захисті електроніки від перегріву або перевантаження.

Серед технічних переваг термісторів варто відзначити їхню високу чутливість у вузькому діапазоні температур, що дозволяє ефективно використовувати їх у застосуваннях, де потрібно фіксувати малі температурні коливання. Вони мають швидкий час відгуку та не потребують складних схем підключення, що спрощує їх інтеграцію у різноманітні електронні системи. Завдяки малим розмірам і низькій вартості термістори є оптимальним вибором для побудови компактних і бюджетних систем моніторингу.

Проте важливо враховувати й обмеження, притаманні термісторам.

Насамперед, це їхня нелінійна вольт-амперна характеристика, яка ускладнює обробку сигналу та вимагає застосування математичних моделей або використання табличних даних для точного перетворення опору в температурне значення. Також їхній робочий діапазон зазвичай обмежується межами від -50°C до $+150^{\circ}\text{C}$, що не завжди задовольняє потреби промислових або наукових застосувань із екстремальними умовами експлуатації.

Для покращення точності у практиці часто використовують калібрування термісторів під конкретні умови роботи. Це дозволяє зменшити похибку вимірювань і забезпечити стабільну роботу давача в довготривалому режимі. Крім того, у сучасних мікропроцесорних системах обробку даних з термістора може виконувати мікроконтролер із вбудованим АЦП, що дає змогу будувати прості та ефективні температурні вимірювальні канали.

З урахуванням поєднання доступної вартості, простоти використання та задовільної точності, термістори залишаються одним із найпрактичніших рішень для побудови недорогих систем моніторингу температури в приміщеннях, зокрема в межах бакалаврських або навчально-дослідницьких проєктів.

У системах моніторингу температури, де потрібна висока точність або розширений діапазон вимірювання, термопари та термометри опору (RTD) є ключовими типами давачів, що широко застосовуються завдяки своїм унікальним властивостям.

Термопари є простими, але надзвичайно надійними сенсорними елементами, принцип роботи яких базується на термоелектричному ефекті (рис. 1.7). Коли два різнорідних метали з'єднані між собою в двох точках і між цими точками існує різниця температур, у контурі виникає термоелектрорушійна сила (ЕРС), пропорційна різниці температур. Така конструкція дозволяє використовувати термопари у найсуворіших умовах — від кріогенних до високотемпературних середовищ, включаючи металургійні цехи, печі та реактори, де температура може перевищувати 2000°C .

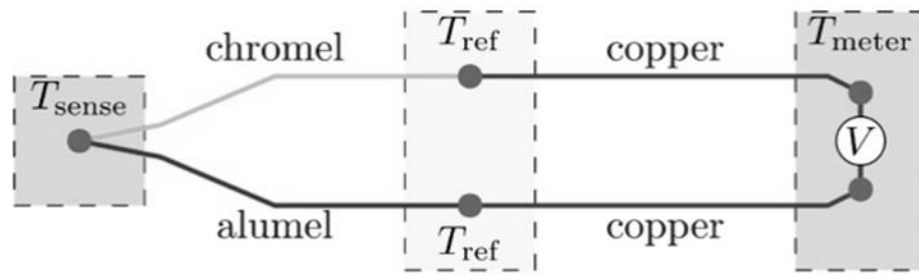


Рисунок 1.7 — Схематичне зображення принципу дії термопар

Основна перевага термопар — надзвичайно широкий температурний діапазон, висока механічна стійкість і придатність до роботи в агресивних середовищах. Вони також мають просту конструкцію, швидкий відгук, доступність та зручність інтеграції з мікроконтролерами. Однак для точних вимірів потрібна компенсація холодного спаю, часто необхідні підсилювачі чи спеціалізовані модулі. Крім того, точність термопар нижча за платинові RTD-давачі, особливо в межах вузького температурного діапазону.

Платинові термометри опору (наприклад, PT100) засновані на зміні електричного опору провідника залежно від температури. Платина забезпечує стабільну характеристику, корозійну стійкість і широку область вимірювання. RTD-давачі вирізняються високою точністю (до $\pm 0,1$ °C), стабільністю показників та лінійністю, що спрощує обробку сигналу. Це робить їх придатними для точних промислових систем, лабораторій і критичних технологічних процесів.

Попри переваги, RTD мають вищу вартість, вимагають прецизійного вимірювання опору та часто реалізуються у чотирипровідних схемах для компенсації довжини провідників. Це ускладнює фізичне впровадження, особливо за умов обмеженого простору або великої кількості каналів вимірювання.

Таким чином, вибір між термопарами й RTD залежить від умов експлуатації: термопари краще підходять для агресивного середовища або надвисоких температур, тоді як RTD — для точних і стабільних вимірювань у вузькому діапазоні. Обидва типи сумісні з мікропроцесорними системами, хоча потребують

різного рівня електронної підтримки. Сучасні мікроконтролери підтримують відповідні інтерфейси, а за потреби застосовуються спеціальні мости, перетворювачі або модулі компенсації.

Паралельно з цими класичними давачами активно використовуються цифрові напівпровідникові сенсори, що поєднують сенсорний елемент, АЦП та інтерфейс передачі даних в одному корпусі. Це значно спрощує інтеграцію, мінімізує апаратні витрати та скорочує час розробки систем. DS18B20, наприклад, підтримує шину 1-Wire, дозволяючи підключати кілька сенсорів до одного виводу мікроконтролера, що зручно в багатозонних системах. DHT22, у свою чергу, передає як температуру, так і вологість через єдиний цифровий вихід, має високу точність і простий протокол обміну даними.

Перевага таких сенсорів у тому, що вони не потребують зовнішніх АЦП чи підсилювачів, легко підтримуються бібліотеками Arduino та інших платформ, мають вбудовані механізми перевірки цілісності даних і стабільно працюють у побутових та напівпромислових умовах. Недоліком є обмеження за температурним діапазоном, порівняно повільна частота оновлення даних і зниження точності при нестабільному живленні або довгих лініях.

У контексті мікропроцесорних систем моніторингу температури саме цифрові сенсори, такі як DS18B20 і DHT22, стають оптимальним рішенням для більшості непрофесійних і напівпрофесійних застосувань, де ключову роль відіграє співвідношення точності, простоти впровадження та вартості. Їхня сумісність із Arduino, ESP32, STM32, Raspberry Pi дозволяє легко адаптувати систему до будь-якої апаратної платформи, реалізувати локальний та віддалений моніторинг, а також масштабувати архітектуру під конкретні потреби користувача.

Загалом сучасні системи дають змогу комбінувати різні типи сенсорів — термопари для високотемпературних ділянок, RTD для точних зон і цифрові сенсори для загального моніторингу — в межах єдиної платформи. Це дозволяє досягти балансу між вартістю, точністю й надійністю, створивши гнучке рішення, здатне адаптуватися до різних умов експлуатації.



Рисунок 1.8 — Напівпровідниковий цифровий давач DHT22

DS18B20 є температурним давачем з цифровим виходом, який забезпечує точність вимірювань до $\pm 0,5\text{ }^{\circ}\text{C}$ у діапазоні від $-55\text{ }^{\circ}\text{C}$ до $+125\text{ }^{\circ}\text{C}$. Завдяки унікальному серійному номеру кожного пристрою він дозволяє будувати багатоточкові системи вимірювання на одній шині, що є надзвичайно зручним для моніторингу великих приміщень або багатозонних систем. Пристрій не потребує зовнішніх калібрувальних процедур і може працювати з живленням від однієї лінії даних, що додатково знижує складність апаратної частини.

DHT22, на відміну від DS18B20, крім температури також вимірює відносну вологість повітря, що робить його привабливим для побутових систем мікроклімату, розумних будинків та портативних пристроїв. Його типова точність вимірювання температури становить $\pm 0,5\text{ }^{\circ}\text{C}$, а вологості — близько від ± 2 до $\pm 5\%$. Обидва ці давачі мають низьке енергоспоживання та компактні розміри, що дозволяє легко інтегрувати їх у корпуси пристроїв або безпосередньо в меблі та декоративні елементи інтер'єру.

Особливої уваги заслуговують пірометри — давачі, що здійснюють безконтактне вимірювання температури на основі інфрачервоного (ІЧ) випромінювання, що випромінюється об'єктом (рис. 1.9). Це робить їх незамінними в тих випадках, коли доступ до об'єкта ускладнений або фізичний контакт є

небажаним чи небезпечним. Наприклад, у виробничих процесах, де спостерігається висока температура або швидке переміщення об'єктів, пірометри дозволяють оперативно й точно контролювати стан обладнання або матеріалів без порушення технологічного циклу.

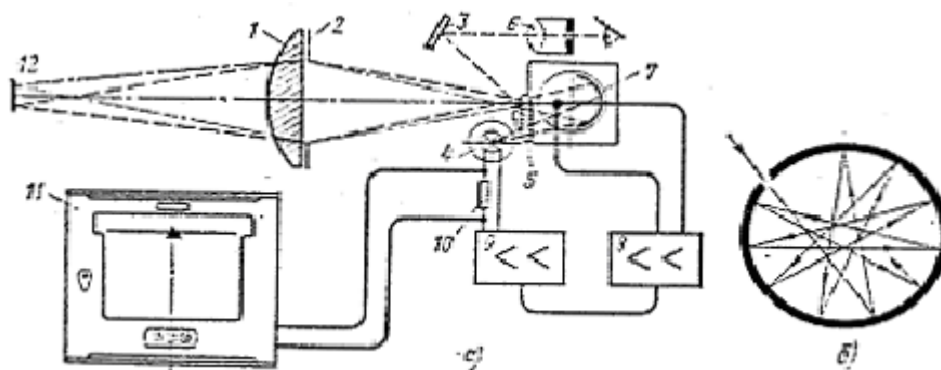


Рисунок 1.9 — Схема дії ІЧ-пірометра

Сучасні пірометри здатні вимірювати температуру в широкому діапазоні — від $-50\text{ }^{\circ}\text{C}$ до $+3000\text{ }^{\circ}\text{C}$, з типовою точністю до $\pm 1\text{ }^{\circ}\text{C}$, що робить їх придатними як для промислових установок, так і для спеціалізованих лабораторних вимірювань. Вони також можуть використовуватись у системах безпеки, наприклад, для виявлення перегріву обладнання, а також у медицині — для безконтактного вимірювання температури тіла. У поєднанні з мікропроцесорними контролерами пірометри здатні стати основою ефективних систем термографії та тепловізійного аналізу.

Завдяки тому, що як цифрові напівпровідникові давачі, так і пірометри мають стандартні інтерфейси передачі даних, вони легко інтегруються в комп'ютеризовані системи моніторингу температури приміщень. Це дає змогу будувати комплексні, гнучкі та інтелектуальні рішення для кліматичного контролю з можливістю дистанційного керування, архівування даних та прогнозування температурних змін. Таким чином, використання зазначених типів давачів у поєднанні з сучасними програмно-апаратними платформами дозволяє реалізувати високоефективні системи, які повністю відповідають вимогам сучасної техніки.

Інфрачервоні тепловізійні камери є високотехнологічними приладами, які

здійснюють вимірювання температури поверхонь на основі інтенсивності інфрачервоного випромінювання, що виходить від об'єкта. На відміну від точкових безконтактних пірометрів, тепловізори формують повноцінне зображення температурного розподілу, створюючи так звану теплову карту — кольорову візуалізацію температурних градієнтів на об'єкті спостереження. Це дозволяє виявляти теплові аномалії, витоки енергії, приховані дефекти в будівельних конструкціях, проблеми з ізоляцією або локалізувати перегрів елементів промислового обладнання без необхідності зупинки процесу. Побачити приклад тепловізійного обстеження промислового обладнання можна на рисунку 1.10



Рисунок 1.10 — Приклад тепловізійного обстеження промислового обладнання

У будівництві тепловізійна діагностика широко застосовується для контролю якості теплоізоляції будівель, виявлення місць втрати тепла, перевірки герметичності вікон та дверей. В медичній сфері тепловізори використовуються для діагностики запальних процесів, судинних порушень, а також як допоміжний інструмент у скринінгу стану організму, зокрема в неінвазивному моніторингу температури тіла. У промисловості інфрачервоні камери активно застосовуються для контролю електричних щитів, трансформаторів, двигунів, підшипників, де нагрівання може свідчити про несправність або зношення.

Попри високу вартість професійних тепловізорів, з розвитком технологій на

ринку з'явилися доступні компактні моделі, які можуть підключатися до смартфонів або мікроконтролерів через стандартні інтерфейси, як-от USB, Bluetooth або Wi-Fi. Це значно розширює можливості інтеграції тепловізійних модулів у комп'ютерні системи моніторингу температури, особливо в автоматизованих або розподілених системах.

Загалом кожен тип температурного давача має свої оптимальні області застосування, і ефективність конкретного рішення залежить від відповідності параметрів давача до вимог конкретного завдання. Визначальними критеріями вибору залишаються точність вимірювань, температурний діапазон, стійкість до впливу агресивних або складних середовищ, а також технічні й економічні аспекти — вартість, доступність, енергоспоживання та складність інтеграції в обчислювальну систему.

Сучасні тенденції розвитку температурних давачів демонструють прагнення до мініатюризації пристроїв без втрати точності, зниження енергоспоживання, що особливо важливо для автономних систем та пристроїв на базі технологій енергозбирання. Крім того, все частіше зустрічаються багатофункціональні давачі, які крім температури можуть вимірювати вологість, тиск, освітленість або рівень вібрацій. Такі сенсорні модулі істотно підвищують інформативність та функціональність системи в цілому.

Ще одним важливим напрямом розвитку є впровадження бездротових інтерфейсів (рис. 1.11), таких як Bluetooth Low Energy, Zigbee, LoRaWAN або NB-IoT, що дозволяє формувати розподілені мережі з великою кількістю температурних вузлів без необхідності прокладання комунікаційних кабелів. Це відкриває широкі можливості для створення масштабованих та гнучких систем моніторингу в умовах розгалужених або важкодоступних об'єктів.



Рисунок 1.11 — Фото сучасного багатофункціонального датчика

Таким чином, розвиток сенсорних технологій у поєднанні з мікропроцесорними засобами обробки інформації формує потужну основу для побудови інтелектуальних, адаптивних та енергоефективних систем кліматичного контролю, здатних працювати в найрізноманітніших умовах та відповідати зростаючим вимогам сучасної техніки.

2 МЕТОДИ ТА ЗАСОБИ РОЗРОБКИ КОМП'ЮТЕРНОЇ СИСТЕМИ МОНІТОРИНГУ

2.1 Вибір апаратної платформи для реалізації системи

Вибір оптимальної апаратної платформи є одним із ключових і критично важливих етапів при розробці комп'ютерної системи моніторингу температури, оскільки саме на цьому етапі закладаються технічні основи функціонування всієї системи. Від обраної платформи залежить не лише набір доступних функцій, таких як можливість підключення датчиків, дисплеїв або модулів зв'язку, а й загальна продуктивність, стабільність роботи, енергоспоживання, гнучкість конфігурації, масштабованість та навіть довговічність готового пристрою.

Успішне вирішення завдання створення надійної й ефективної системи контролю мікроклімату вимагає комплексного підходу до вибору мікроконтролерної бази, що враховує технічні характеристики, апаратну та програмну підтримку, доступність на ринку, а також відповідність бюджету проекту. Зокрема, необхідно враховувати такі параметри, як тактова частота, обсяг оперативної та флеш-пам'яті, кількість і тип доступних входів/виходів, наявність цифрових інтерфейсів (I²C, SPI, UART), підтримка мережевих модулів (Wi-Fi, Bluetooth), а також сумісність із середовищами розробки.

Сучасний ринок пропонує широкий спектр мікропроцесорних платформ, серед яких найчастіше розглядаються такі популярні варіанти, як Arduino Uno, ESP32, STM32 та Raspberry Pi. Кожна з цих платформ має власні сильні сторони та певні обмеження. Наприклад, Arduino Uno є надзвичайно зручною для навчальних цілей і швидкого прототипування, але має обмежену обчислювальну потужність та малий обсяг пам'яті, що ускладнює реалізацію складніших алгоритмів або підключення великої кількості пристроїв. STM32, навпаки, забезпечує високу продуктивність і розширену функціональність, але вимагає глибших знань електроніки та досвіду програмування в середовищі STM32CubeIDE або аналогічних. Raspberry Pi надає можливості повноцінного комп'ютера, проте часто є надлишковим рішенням для задач простого моніторингу температури й потребує

складнішої інфраструктури.

У цьому контексті платформа ESP32 виглядає як збалансоване рішення, що поєднує переваги простоти Arduino з розширеними можливостями сучасних мікроконтролерів. Вона має вбудовані модулі Wi-Fi і Bluetooth, великий обсяг пам'яті, багатофункціональні порти введення/виведення, а також підтримує популярні середовища розробки, зокрема Arduino IDE, PlatformIO та FLProg. Завдяки цьому ESP32 дозволяє реалізовувати складні інтегровані системи контролю з мінімальними витратами часу та ресурсів, що особливо важливо для проєктів обмеженого бюджету та в умовах навчального процесу.

Таким чином, вибір апаратної платформи — це не лише технічне рішення, а стратегічний крок, що визначає потенціал розвитку, надійність та довготривалу ефективність усієї комп'ютерної системи моніторингу температури. Раціональний підхід до цього вибору дозволяє уникнути перевитрат, скоротити терміни розробки, забезпечити стабільну роботу пристрою та створити гнучку основу для майбутніх удосконалень або масштабування системи.

Ключовими критеріями вибору апаратної платформи є:

- обчислювальна потужність та продуктивність — здатність системи швидко та точно обробляти дані з датчиків у реальному часі;
- наявність необхідних інтерфейсів для підключення периферії — підтримка таких стандартів, як I²C, SPI, UART, USB тощо, є обов'язковою умовою для інтеграції температурних датчиків, дисплеїв, модулів зв'язку та інших компонентів;
- споживання енергії — критично важливо для автономних або мобільних рішень, де тривалість роботи від акумулятора є визначальною;
- вартість рішення — впливає на загальний бюджет проєкту, особливо у випадках масового виробництва;
- можливість масштабування системи — важливо для проєктів, які можуть надалі розширюватися, наприклад, шляхом додавання нових датчиків або функціональних модулів;
- наявність інструментів розробки та активної спільноти — спрощує процес

проектування, налагодження та підтримки системи.

Популярні апаратні платформи для систем моніторингу температури можна умовно поділити на три категорії:

а) мікроконтролери (MCU) — компактні та енергоефективні рішення, які зазвичай використовуються в простих або автономних системах моніторингу. Серед найпоширеніших варіантів;

1) ESP8266 / ESP32 (рис. 2.1) — мікроконтролери з вбудованою підтримкою Wi-Fi та Bluetooth, що робить їх ідеальними для IoT-рішень з бездротовою передачею даних;

2) STM32 — серія високопродуктивних мікроконтролерів з широким набором периферійних інтерфейсів, підходять для більш складних та надійних вбудованих систем;

3) Arduino — доступна та зручна для початківців платформа, що підтримується великою спільнотою та має широкий асортимент модулів і бібліотек;

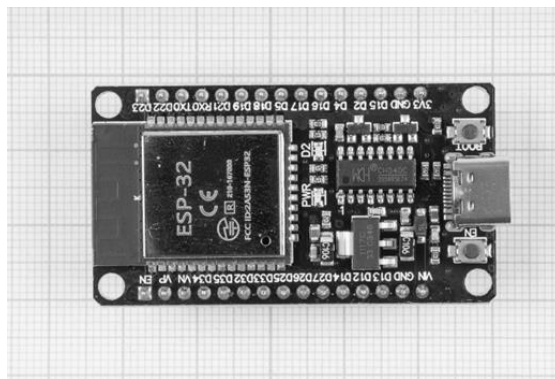


Рисунок 2.1 — Приклад схеми ESP32

б) одноплатні комп'ютери (SBC) — забезпечують вищу обчислювальну потужність і розширені можливості для систем, які потребують роботи з повноцінною операційною системою, обробки зображень, хмарної синхронізації тощо. Найпопулярніші;

1) Raspberry Pi (рис. 2.2) — найвідоміший представник SBC-платформ, підтримує Linux, має широкий спектр застосувань у навчанні, прототипуванні та

реальних системах;

2) Orange Pi / Banana Pi — бюджетні альтернативи з хорошим співвідношенням ціна/функціональність, сумісні з багатьма ОС;

3) BeagleBone — одноплатні комп'ютери промислового класу з розширеною підтримкою введення/виведення, ідеальні для вбудованих систем керування;

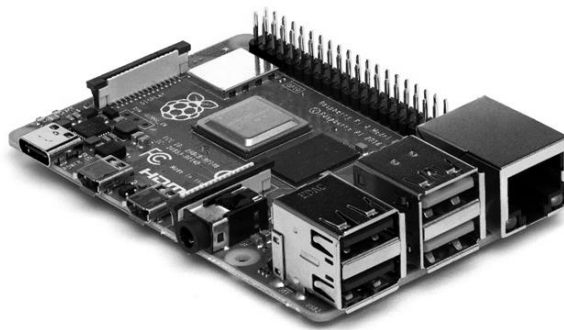


Рисунок 2.2 — Приклад плати Raspberry Pi

в) спеціалізовані промислові контролери застосовуються у складних або критично важливих системах, де висуваються підвищені вимоги до надійності, стабільності та довготривалої експлуатації в жорстких умовах. До цієї категорії входять;

1) PLC (програмовані логічні контролери) — стандартизовані рішення для автоматизованого керування технологічними процесами. PLC забезпечують високу надійність, підтримують реальний час, а також мають широкий спектр входів/виходів для підключення датчиків і виконавчих механізмів. Приклад PLC комп'ютера можна побачити на рисунку 2.3;

2) промислові комп'ютери — повноцінні комп'ютерні системи, захищені від впливу пилу, вологи, вібрацій та температурних коливань. Вони підтримують роботу зі SCADA-системами та здатні обробляти великі обсяги даних у реальному часі;

3) спеціалізовані IoT-шлюзи — пристрої, що виконують роль посередника між польовими пристроями (датчиками, контролерами) та хмарними сервісами. Підтримують різні протоколи передачі даних (Modbus, MQTT, OPC UA тощо) і часто мають функції попередньої обробки даних.



Рисунок 2.3 — Типові PLC та промислові комп'ютери в шафі управління

Для реалізації запропонованої системи моніторингу температури було обрано апаратну платформу Wemos D1 Mini на базі ESP32 (рис. 2.4), що обґрунтовується наступними факторами:

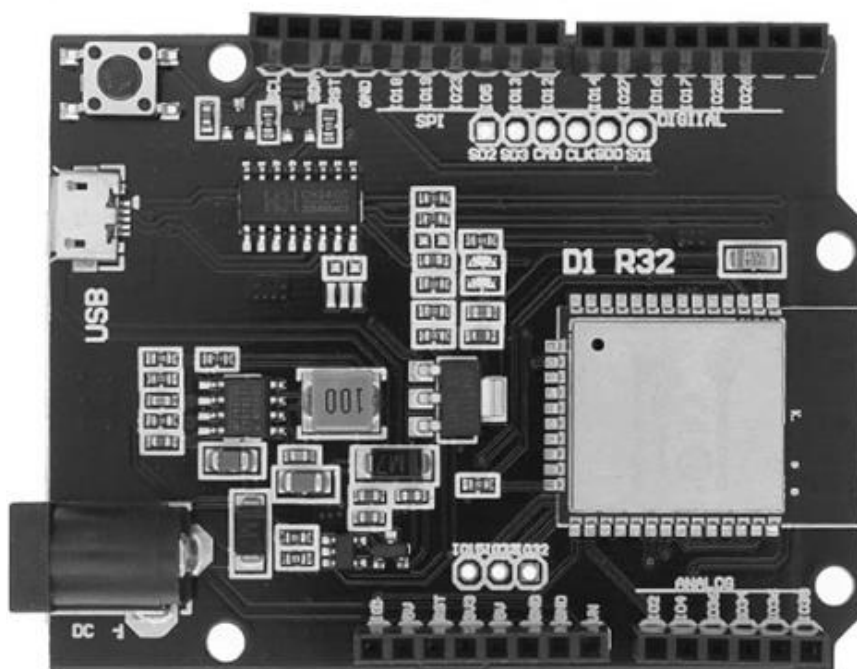


Рисунок 2.4 — Фото Wemos D1 Mini ESP32

- висока продуктивність двоядерного процесора ESP32 при збереженні енергоефективності;
- вбудована підтримка Wi-Fi та Bluetooth, що дозволяє реалізувати як локальний, так і дистанційний моніторинг;

- достатня кількість GPIO-портів для підключення датчиків, дисплеїв, реле та інших елементів;
- підтримка апаратного PWM, ADC, I2C, SPI та інших інтерфейсів, необхідних для розширення функціональності;
- активна спільнота розробників, широке поширення прикладів, бібліотек та документації;
- сумісність з популярними середовищами розробки, зокрема Arduino IDE та PlatformIO.

Архітектура апаратної частини системи включає:

- центральний процесорний модуль — Wemos D1 Mini ESP32 (рис. 2.5);
- датчик температури та вологості — DHT22;
- LCD дисплей з I2C-інтерфейсом — 1602 для локального відображення даних;
- блок живлення — адаптер на 5 В з пониженням до 3.3 В для живлення ESP32;
- додаткові компоненти — кнопки керування режимами, світлодіоди індикації, пасивні елементи.

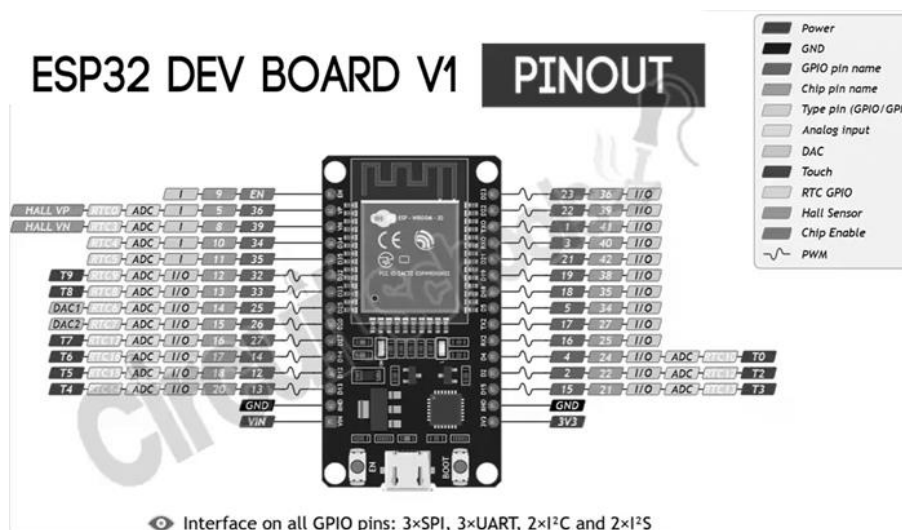


Рисунок 2.5 — Схема підключення компонентів до ESP32

Вибір саме ESP32 у виконанні Wemos D1 Mini є технічно й практично виваженим рішенням, яке дозволяє реалізувати компактну, функціонально

насичену та економічно доцільну комп'ютерну систему моніторингу температури з потенціалом подальшого розвитку. Цей варіант апаратної платформи поєднує в собі низку важливих переваг, які суттєво впливають на зручність реалізації проекту, його гнучкість, стабільність роботи та розширюваність у майбутньому.

Насамперед, форм-фактор Wemos D1 Mini забезпечує мінімальні габарити, що особливо важливо для побудови пристроїв, призначених для встановлення у обмежених просторах — на стінах приміщень, у корпусах щитового обладнання, розетках тощо. Попри свої розміри, плата має повноцінний набір функцій, притаманних повнорозмірній ESP32, включаючи цифрові та аналогові входи/виходи, шини I²C, SPI, UART, а також вбудовані модулі Wi-Fi та Bluetooth, що дозволяє легко додавати бездротові функції без потреби в додаткових компонентах.

З погляду енергоспоживання, ESP32 підтримує кілька режимів енергозбереження, зокрема light sleep та deep sleep, що дає можливість створювати автономні пристрої з живленням від акумулятора або батарей, значно подовжуючи час їх роботи. Ця властивість є важливою перевагою у порівнянні з деякими іншими мікроконтролерами, які не мають апаратної підтримки глибокого сну або вимагають складних рішень для його реалізації.

Ще однією важливою перевагою вибору ESP32 Wemos D1 Mini є її широка підтримка середовищ розробки, таких як Arduino IDE, PlatformIO, Espressif IDF, а також FLProg — середовища візуального програмування, яке активно використовується у навчальному процесі. Завдяки цьому створення програмного забезпечення не вимагає великих зусиль або глибоких знань низькорівневого коду, а отже, значно спрощується вхід у проєкт як для студентів, так і для початківців-розробників.

Щодо вартості, Wemos D1 Mini на базі ESP32 є надзвичайно доступним рішенням — вартість модуля в роздрібному продажу часто в десятки разів нижча за повноцінні одноплатні комп'ютери на кшталт Raspberry Pi, при цьому для реалізації більшості задач моніторингу й автоматизації його функціоналу більш ніж достатньо. Це дозволяє використовувати платформу в масштабованих проєктах або

у випадках, де критичною є вартість кожного пристрою (наприклад, при масовому розгортанні в будівлях чи класах).

Також слід зазначити, що ESP32 у версії Wemos D1 Mini легко піддається модернізації. До неї можна підключити нові сенсори, дисплеї, реле, модулі пам'яті або зв'язку без суттєвої зміни основного коду. Завдяки великій спільноті розробників і відкритим бібліотекам, модернізація та розширення функціоналу не вимагає значних витрат часу чи ресурсів.

У підсумку, вибір ESP32 у форм-факторі Wemos D1 Mini дозволяє створити компактну, ефективну, недорогу та зручну в обслуговуванні систему, яка не лише повністю задовольняє вимоги поточного проєкту, а й має великий потенціал для подальшого вдосконалення, включаючи бездротову інтеграцію, дистанційне керування, автоматичне логування даних та розширення до повноцінної IoT-архітектури. Такий підхід є раціональним і сучасним рішенням як з технічної, так і з економічної точки зору.

2.2 Вибір температурних датчиків та особливості їх інтеграції

При виборі сенсора для реалізації системи моніторингу температури було проведено детальний аналіз низки поширених цифрових датчиків, серед яких розглядалися DHT11 v2, DHT22, DS18B20 та BME280. Кожен із цих сенсорів має свої особливості, що проявляються у технічних характеристиках, складності інтеграції, функціональних можливостях та вартості. Оскільки датчик є одним з ключових елементів у структурі системи, його вибір суттєво впливає як на точність вимірювання, так і на стабільність загальної роботи пристрою. Порівняння розглянутих сенсорів можна побачити у таблиці 2.1.

Під час аналізу враховувались загальні вимоги до системи, яка призначена для контролю мікроклімату в приміщеннях. Зокрема, розглядалися ті технічні параметри, які дозволяють забезпечити прийнятний рівень точності вимірювань у межах побутових температур. Крім того, бралися до уваги спрощення процесу підключення сенсора до мікроконтролера, програмна підтримка та адаптація до обраного середовища розробки. Економічна доцільність також була важливим

чинником, оскільки проєкт орієнтований на недорогі рішення з можливістю масштабування.

Таблиця 2.1 — Порівняння з альтернативними давачами

Давач	Діапазон (°C)	Точність	Інтерфейс	Особливості
DHT11 v2	від 0 до +50	± 2 °C	цифровий	Простий, дешевий, оновлена версія
DHT22	від -40 до +80	± 0.5 °C	цифровий	Вища точність, але дорожчий
DS18B20	від -55 до +125	± 0.5 °C	1-Wire	Висока точність, температурний давач
BME280	від -40 до +85	± 1 °C	I2C/SPI	Вимірює також тиск і вологість

На основі проведеного аналізу найкращим варіантом для впровадження у проєкт виявився сенсор DHT11 v2 (рис. 2.6). Це компактний цифровий давач, який дозволяє одночасно вимірювати температуру та вологість, що значно розширює функціональність системи, не ускладнюючи її апаратну реалізацію. Його робочий діапазон температур охоплює типові значення для житлових і офісних приміщень, а точність вимірювань, хоча й нижча порівняно з більш дорогими аналогами, цілком достатня для задач базового моніторингу. Простий однопровідний цифровий інтерфейс значно полегшує підключення до мікроконтролера та дозволяє ефективно використовувати обмежені апаратні ресурси.

DHT11 v2 також відзначається широкою підтримкою в спільноті розробників, що забезпечує наявність готових бібліотек, документації та прикладів коду для різних платформ, включаючи Arduino IDE та FLProg. Це суттєво знижує поріг входу в проєкт, дозволяє зосередитися на реалізації основної логіки системи та спрощує подальшу модернізацію або переналаштування пристрою. Його стабільність роботи, низька вартість і достатній набір функцій роблять його найбільш раціональним вибором серед аналізованих варіантів.

Отже, вибір сенсора DHT11 v2 був зумовлений його відповідністю технічним вимогам проєкту, простотою апаратної реалізації, підтримкою з боку програмного

забезпечення та оптимальним співвідношенням між ціною і функціональністю. Це дозволило реалізувати систему, яка водночас є простою в розробці й ефективною в роботі, а також залишає можливості для її подальшого удосконалення та масштабування.

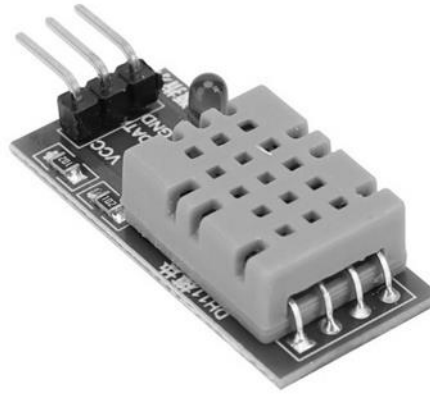


Рисунок 2.6 — Фото давача DHT11 v2

Важливим аргументом на користь використання сенсора DHT11 v2 стала відповідність його технічних характеристик до умов, у яких передбачається експлуатація розробленої системи. Зокрема, діапазон робочих температур цього сенсора становить від 0 до +50 °C, що цілком охоплює типові значення температур у житлових, навчальних, офісних та адміністративних приміщеннях, де найчастіше реалізуються подібні системи контролю мікроклімату. У більшості випадків температура у таких середовищах не виходить за межі цього інтервалу, тому застосування давача з ширшим або промисловим діапазоном вимірювання було б економічно невиправданим.

Хоча DHT11 v2 має дещо нижчу точність у порівнянні з такими сенсорами, як DHT22 або DS18B20, його характеристика похибки в межах ± 2 °C є повністю прийнятною для задач базового моніторингу температури, де не потрібне високоточне керування. У контексті повсякденного застосування, особливо коли йдеться про контроль комфорту в приміщенні, така точність є достатньою для прийняття обґрунтованих рішень щодо увімкнення або вимкнення опалення, вентиляції чи кондиціонування. При цьому більш точні сенсори хоча й

забезпечують меншу похибку, потребують значно вищих витрат на закупівлю та інтеграцію, а також мають складніші протоколи обміну або вищі апаратні вимоги. DHT11 v2 вирізняється ще й своєю винятковою простотою в експлуатації. Для його підключення до мікроконтролера достатньо одного цифрового входу, а передача даних відбувається за допомогою простого однопровідного інтерфейсу. Це зменшує кількість необхідних з'єднань, спрощує трасування друкованої плати й дозволяє уникнути помилок, які часто виникають при роботі з багатожильними чи аналоговими сенсорами. Крім того, цифрова природа вихідного сигналу виключає вплив зовнішніх перешкод на точність вимірювання, що особливо важливо в умовах потенційно нестабільного електромагнітного середовища або при прокладенні довгих з'єднувальних провідників.

Ще однією суттєвою перевагою є широка підтримка DHT11 v2 у найпопулярніших середовищах розробки. У середовищі Arduino IDE існують численні бібліотеки, які дозволяють реалізувати зчитування температури та вологості буквально кількома рядками коду, що робить цей сенсор надзвичайно привабливим для новачків, студентів та розробників із базовими знаннями програмування. У FLProg, який використовує середовище візуального програмування, інтеграція DHT11 v2 є ще простішою — користувачеві достатньо перетягнути блок на схему та налаштувати порт. Це значно скорочує час розробки, зменшує кількість можливих помилок і дозволяє сфокусуватись на логіці функціонування системи, а не на низькорівневому програмуванні.

Завдяки своїй доступності на ринку, стабільній роботі, достатній точності, а також надзвичайній простоті як у технічному, так і в програмному плані, давач DHT11 v2 став оптимальним вибором у межах цього проєкту. Він дозволив реалізувати всі поставлені функціональні задачі без перевищення бюджету та без необхідності ускладнення як апаратної, так і програмної частини системи. Такий підхід забезпечив не лише ефективне виконання поточного завдання — моніторинг мікроклімату — а й створив гнучку платформу для подальшого розвитку, включно з можливістю інтеграції з бездротовими мережами або автоматичного керування виконавчими пристроями.

Отже, вибір DHT11 v2 був зумовлений його цілковитою відповідністю поставленим вимогам до системи, раціональним балансом між функціональністю, точністю та вартістю, а також широкими можливостями інтеграції в умовах обмежених ресурсів і часу розробки. Його використання дало змогу побудувати надійне, стабільне та водночас просте в реалізації рішення, придатне для застосування як у навчальних проєктах, так і в побутових умовах.

Технічні характеристики DHT11 v2:

- діапазон температури: від 0 °C до +50 °C;
- точність температури: ± 2 °C;
- діапазон вологості: від 20% до 90% RH;
- точність вологості: $\pm 5\%$ RH;
- напруга живлення: від 3.0 В до 5.5 В;
- інтерфейс: цифровий, однопровідний (Single-Bus);
- частота опитування: 1 Гц (одне вимірювання за секунду);
- розміри: $15.5 \times 12 \times 5.5$ мм.

Основні переваги використання DHT11 v2:

- простота інтеграції — цифровий інтерфейс дозволяє підключення без необхідності використання АЦП;
- подвійний функціонал — одночасне вимірювання температури та вологості;
- низька вартість — вигідна ціна при збереженні базової точності;
- сумісність — повна апаратна та програмна сумісність з базовим DHT11;
- покращення у версії v2 — вища стабільність, краще захищений від електромагнітних перешкод.

Особливості підключення DHT11 v2 (рис. 2.7):

- VCC — живлення від 3.3 В до 5 В;
- DATA — цифровий сигнал (підключається до GPIO ESP32);
- GND — загальний провід;
- примітка: для коректної роботи необхідно підключити підтягуючий резистор від 4.7 до 10 кОм між лінією DATA і VCC.

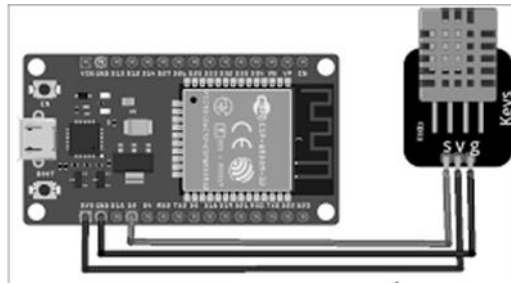


Рисунок 2.7 — Схема підключення DHT11 v2 до ESP32

Реалізація моніторингу температури та вологості на базі давача DHT11 v2 передбачає використання перевірених інструментів і методів для досягнення стабільної та точнішої роботи.

Основні програмні аспекти реалізації:

- підключення бібліотек — використовується офіційна бібліотека DHT sensor library від Adafruit або її аналоги, сумісні з Arduino IDE або PlatformIO;
- ініціалізація давача — давач ініціалізується під час запуску мікроконтролера, викликаючи метод `dht.begin()`;
- періодичне опитування — зчитування значень температури й вологості здійснюється з інтервалом не менше 1 секунди згідно з технічними обмеженнями давача;
- обробка помилок — передбачено перевірку валідності отриманих даних (`isnan()`), що дозволяє уникати виведення некоректної інформації у випадку помилок зв'язку.

Оптимізація точності та надійності вимірювань:

- калібрування давача — за потреби можливе калібрування за допомогою еталонного термометра, з внесенням програмних поправок;
 - фільтрація викидів — використовуються програмні алгоритми, що відкидають аномальні значення, які відрізняються від попередніх показників більш ніж на допустимий поріг;
 - усереднення — система зберігає кілька останніх вимірів і обчислює середнє значення, згладжуючи коливання, спричинені шумами або перешкодами.
- Виявлені обмеження та способи їх подолання:

- невисока точність: компенсується програмним усередненням та фільтрацією значень.;
- необхідність дотримання інтервалу між вимірами: рішення: таймер на основі `millis()` або `delay()` забезпечує мінімальний інтервал у 1 секунду;
- чутливість до перешкод: рекомендовано мінімізувати довжину провідників та використовувати екранування або відповідне трасування на друкованій платі (рис. 2.9).

Good practices

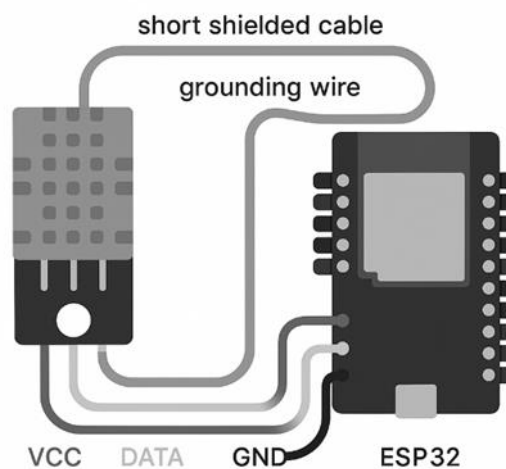


Рисунок 2.9 — Схема good practices для монтажу датчика

Застосування сенсора DHT11 v2 у складі комп'ютерної системи моніторингу температури на основі мікроконтролера ESP32 є технічно доцільним і практично виправданим вибором, який забезпечує необхідну функціональність для більшості побутових і навчальних застосувань. Система, побудована на такому поєднанні компонентів, має достатній рівень точності вимірювань, простоту інтеграції, низький рівень споживання електроенергії та чудову підтримку на рівні програмного забезпечення, що дає змогу реалізувати її навіть у проєктах з обмеженим бюджетом або часовими ресурсами.

З точки зору функціональних можливостей, DHT11 v2 забезпечує вимірювання температури з похибкою $\pm 2^{\circ}\text{C}$, що є прийнятним показником у більшості непрофесійних задач, де не потрібна висока точність — наприклад, у

системах контролю температури повітря в кімнатах, офісах, навчальних аудиторіях, теплицях, інкубаторах, системах вентиляції тощо. Для таких сфер застосування критичним є саме факт зміни температури та тенденція її коливання, а не абсолютні значення з точністю до десятих часток градуса.

Сенсор має цифровий вихід, що є значною перевагою з погляду інженерної реалізації. На відміну від аналогових сенсорів, які вимагають використання АЦП (аналогово-цифрового перетворювача), налаштування опорної напруги та боротьби з перешкодами, DHT11 v2 передає вже оброблене значення температури у цифровому форматі, використовуючи простий однопровідний протокол. Це зменшує складність схеми, підвищує її стійкість до електричних завад, а також значно спрощує програмну реалізацію процесу зчитування. Простота підключення дозволяє легко інтегрувати сенсор до практично будь-якого мікроконтролера, зокрема й до ESP32, який підтримує декілька цифрових портів і має достатній обсяг пам'яті для обробки даних.

Перевагою поєднання DHT11 v2 із платформою ESP32 є те, що обидва компоненти мають широку підтримку в популярних середовищах розробки — таких як Arduino IDE та FLProg. У першому випадку розробники можуть скористатися численними бібліотеками, прикладами коду та документацією, які дозволяють реалізувати функціональну систему буквально за кілька годин. У другому — середовищі FLProg — реалізація можлива навіть без знань програмування, оскільки логіка роботи будується за допомогою блоків, що імітують принципи релейної автоматики. Такий підхід не лише скорочує час розробки, а й знижує ймовірність помилок, підвищує надійність коду та дає змогу швидко адаптувати систему під нові потреби користувача.

Фізичні параметри сенсора також сприяють його успішному використанню. Завдяки компактним розмірам, DHT11 v2 можна без проблем розмістити в корпусах невеликих пристроїв, настінних блоках, розеткових адаптерах або переносних модулях. Крім того, низьке енергоспоживання робить цей сенсор особливо зручним для проєктів, які мають працювати автономно — наприклад, у системах, що живляться від батареї або акумулятора. У поєднанні з

енергозберігаючими режимами ESP32, таким як deep sleep, це дає змогу створювати пристрої, які можуть працювати без підзарядки впродовж багатьох днів або навіть тижнів, залежно від частоти зчитування даних.

Інтеграція DHT11 v2 з ESP32 дозволяє реалізувати не лише локальне вимірювання температури та вологості, а й виведення інформації на дисплей, передачу даних по Wi-Fi або Bluetooth, ведення логів, дистанційний моніторинг через мобільний застосунок або вебінтерфейс. При цьому програмна реалізація залишається простою та зрозумілою навіть для студентів або користувачів-початківців. Завдяки цьому реалізована система може виступати як завершений приклад навчального або побутового пристрою, так і як основа для подальшої розробки більш складних автоматизованих систем.

Таким чином, поєднання сенсора DHT11 v2 із мікроконтролером ESP32 створює технічно збалансовану, надійну й економічно обґрунтовану платформу для розробки систем моніторингу мікроклімату. Такий вибір компонентів дозволяє забезпечити не лише ефективну роботу пристрою, а й зручність у розробці, масштабуванні, супроводі й адаптації до нових завдань.

2.3 Вибір мови програмування та середовища розробки

Для реалізації програмного забезпечення системи моніторингу температури було здійснено комплексний аналіз сучасних інструментів розробки, з урахуванням як технічних обмежень апаратної платформи, так і функціональних потреб, сформульованих на етапі проєктування. Програмне забезпечення відіграє не менш важливу роль, ніж апаратна частина, оскільки саме воно забезпечує коректну взаємодію між сенсорами, логікою керування, засобами індикації, модулями зв'язку та іншими елементами системи. Ефективність, стабільність і зручність експлуатації всього пристрою безпосередньо залежать від якості програмної реалізації. Тому вибір середовища розробки був ключовим етапом у процесі побудови всієї системи.

Основними критеріями при виборі інструментів програмування стали кілька важливих технічних і практичних аспектів. Перш за все, визначальною була повна

сумісність середовища з обраною апаратною платформою ESP32, яка, з одного боку, має значні функціональні можливості, а з іншого — вимагає підтримки специфічних інтерфейсів, таких як Wi-Fi, Bluetooth, а також цифрових шин UART, I²C і SPI. Неможливість повноцінної інтеграції мікроконтролера зі стороннім середовищем автоматично виключала його з розгляду.

Наступним вагомим фактором була наявність стабільних і перевірених бібліотек для роботи з цифровим сенсором DHT11 v2, що застосовується у системі як базовий засіб зчитування параметрів мікроклімату. Оскільки дані з давача надходять у вигляді цифрових сигналів з використанням спеціального протоколу обміну, для ефективного та стабільного зчитування необхідне використання адаптованих бібліотек, які враховують затримки, контроль помилок та часові обмеження. У разі відсутності таких бібліотек виникає потреба у ручній реалізації низькорівневої логіки, що значно ускладнює процес розробки й підвищує ризик виникнення помилок.

Особливу увагу було приділено зручності налагодження проєкту. Важливо, щоб обране середовище дозволяло швидко компілювати програмний код, зручно завантажувати його на мікроконтролер, переглядати серіальний монітор, працювати з точками зупину та вести візуальний моніторинг змінних у процесі виконання. Середовище має також забезпечувати інформативну систему повідомлень про помилки компіляції, що є критично важливим для швидкого виявлення й виправлення програмних недоліків.

Не менш важливою була можливість реалізації функцій Інтернету речей (IoT), що передбачає не лише локальну обробку та відображення даних, але й потенціал для передачі інформації на віддалений сервер, хмарну платформу або мобільний застосунок. З огляду на те, що ESP32 має вбудовані Wi-Fi та Bluetooth-модулі, середовище розробки повинно підтримувати відповідні бібліотеки, інструменти та протоколи, такі як MQTT, HTTP, WebSocket тощо, а також дозволяти реалізацію серверних функцій, зберігання даних, оновлення прошивки "по повітрю" (OTA) й інші сучасні можливості.

Окремо розглядалася активність та розмір спільноти розробників, які

підтримують те чи інше середовище розробки. Важливою перевагою для вибраного інструменту є наявність великої кількості форумів, документації, навчальних відео, прикладів реалізацій та відкритого вихідного коду. Це не лише зменшує час, необхідний для вивчення середовища, а й спрощує вирішення проблем, які можуть виникнути в процесі розробки. Середовище, що активно розвивається та має підтримку спільноти, також швидше адаптується до нових бібліотек, сенсорів та апаратних оновлень, що позитивно впливає на життєвий цикл усього проєкту.

Таким чином, вибір інструментів програмної реалізації здійснювався не випадково, а на основі об'єктивних вимог до сумісності з апаратною частиною, стабільності програмної підтримки, функціональності, зручності роботи розробника та перспективи подальшого розвитку системи. Обраний інструментарій дозволив реалізувати повноцінне, адаптивне та надійне програмне забезпечення, здатне ефективно управляти всіма модулями системи, обробляти дані в реальному часі та забезпечувати користувача достовірною інформацією про поточний стан мікроклімату в приміщенні..

Після визначення оптимального середовища розробки наступним кроком стало прийняття рішення щодо вибору мови програмування, яка дозволила б ефективно реалізувати логіку роботи комп'ютерної системи моніторингу температури. Зважаючи на специфіку обраної апаратної платформи, особливості структури програмного коду, доступність бібліотек, швидкість компіляції, продуктивність під час виконання та можливість масштабування, було вирішено використовувати мову програмування C++ у межах Arduino Framework як базовий інструмент для створення прошивки мікроконтролера.

Такий вибір є цілком виправданим як з технічної, так і з методологічної точки зору. Arduino Framework — це високорівнева абстракція над платформами на основі C/C++, яка суттєво спрощує роботу з апаратною частиною. Мова C++ у цьому контексті дозволяє розробникам ефективно поєднувати зручність доступу до периферійних пристроїв із можливістю тонкого налаштування низькорівневих параметрів системи, зокрема роботи портів введення/виведення, таймерів, шини I²C, UART, SPI та інших апаратних інтерфейсів, які широко використовуються в

системах автоматизації. Це поєднання високої продуктивності, гнучкості та читабельності коду стало визначальним у контексті розробки системи, що має працювати стабільно, швидко реагувати на зміни параметрів середовища та зберігати працездатність навіть у режимах зниженого енергоспоживання.

Крім того, Arduino екосистема надає розробнику доступ до величезної кількості відкритих та добре документованих бібліотек, які значно полегшують роботу з усіма ключовими компонентами системи — давачами температури та вологості, рідкокристалічними дисплеями, реле, Wi-Fi модулями, кнопками, енергонезалежною пам'яттю тощо. Зокрема, для сенсора DHT11 v2 існують бібліотеки, які реалізують оптимізовану обробку сигналу, автоматичне управління затримками, перевірку контрольних сум та обробку винятків, що дозволяє уникнути помилок, зменшити кількість ручного коду та підвищити надійність усієї системи. Те саме стосується модулів індикації: LCD-дисплеї 1602 з інтерфейсом I2C також мають підтримку у вигляді бібліотек, які дозволяють без зайвих зусиль керувати виведенням даних, курсором, прокруткою тощо.

Ще одним вагомим аргументом на користь C++ у середовищі Arduino є ефективність виконання програмного коду, що має критичне значення при роботі з мікроконтролерами. Завдяки компіляції в машинний код і оптимізації за розміром та швидкістю виконання, C++ дозволяє досягти мінімального часу реакції системи на події, що є особливо важливим у задачах реального часу, наприклад, при контролі температури та увімкненні виконавчих пристроїв. Економне використання оперативної пам'яті та флеш-пам'яті забезпечує можливість реалізації досить складних алгоритмів навіть на ресурсно обмежених мікроконтролерах.

Важливо також, що програми, написані в середовищі Arduino з використанням C++, є кросплатформеними у межах всієї родини сумісних мікроконтролерів. Це означає, що логіка, реалізована для ESP32, може бути з незначними змінами перенесена, наприклад, на Arduino Uno, STM32, ESP8266 або навіть на інші 32-бітні платформи. Така переносимість забезпечує додаткову гнучкість у майбутньому та створює передумови для масштабування або

перепроєктування системи під нові апаратні умови без потреби переписування програмного коду з нуля.

Отже, використання мови C++ у поєднанні з Arduino Framework дало змогу поєднати потужність і швидкодію низькорівневого коду з простотою розробки та широкими можливостями апаратної інтеграції. Це рішення стало не лише технічно ефективним, а й зручним у реалізації, підтримці та подальшому розширенні системи, а також дозволило зосередитися на реалізації прикладних задач, не витрачаючи зайві ресурси на розв'язання другорядних технічних складнощів.

Розробка програмного забезпечення у середовищі Arduino Framework на мові C++ отримала ще один суттєвий імпульс завдяки доступності великої кількості попередньо створених бібліотек, які охоплюють широкий спектр функціональності. Зокрема, мова йде про бібліотеки, що реалізують механізми обміну даними через бездротові інтерфейси, зокрема Wi-Fi, які особливо актуальні в контексті реалізації системи моніторингу як частини концепції Інтернету речей (IoT). Завдяки цим бібліотекам розробникам не потрібно створювати протоколи передачі даних «з нуля», що дозволяє зосередитися безпосередньо на логіці обробки інформації та взаємодії з користувачем.

Зокрема, підтримка таких мережевих протоколів, як MQTT (Message Queuing Telemetry Transport) і HTTP (HyperText Transfer Protocol), реалізована у вигляді готових рішень, які легко інтегруються в проєкт за допомогою кількох рядків коду. Це значно спрощує налаштування обміну даними з зовнішніми хмарними сервісами, локальними серверами або мобільними застосунками. Наприклад, за допомогою MQTT можливо організувати постійний канал передачі телеметричних даних на брокер, що дозволяє централізовано збирати показники з багатьох сенсорних вузлів. Протокол HTTP, у свою чергу, забезпечує гнучку підтримку REST-архітектури, що відкриває можливість створення вебінтерфейсів для відображення даних, налаштування параметрів та керування виконавчими пристроями дистанційно.

Реалізація всіх цих функцій стала можливою завдяки використанню потужних та гнучких середовищ розробки, зокрема Arduino IDE та PlatformIO.

Arduino IDE було обрано на початкових етапах розробки як просте, інтуїтивно зрозуміле та швидке у налаштуванні середовище, що дозволяє зосередитись на архітектурі системи без зайвих технічних ускладнень. Його зручний інтерфейс, автоматичне підключення плат та бібліотек, а також інтегрована підтримка завантаження прошивки через USB зробили його ефективним інструментом для побудови базової функціональності проєкту.

Однак для реалізації більш складних задач, зокрема відлагодження мережеских функцій, керування зовнішніми залежностями, інтеграції контролю версій через Git та підтримки автодоповнення коду, було також використано PlatformIO — сучасне середовище розробки, побудоване на основі Visual Studio Code. Його перевагою є гнучке управління проєктною структурою, зручна інтеграція з репозиторіями бібліотек, а також підтримка великої кількості мікроконтролерних платформ. Завдяки цьому розробка на ESP32 стала не лише ефективною, але й максимально масштабованою — з можливістю швидкого додавання нових функцій, підключення додаткових компонентів і забезпечення багатоплатформності.

Крім того, PlatformIO дозволяє створювати ізольовані середовища для кожного проєкту, що забезпечує незалежність від версій бібліотек і мінімізує конфлікти при розгортанні на нових пристроях. Це особливо важливо в умовах, коли система має бути гнучкою до змін конфігурації або потребує масштабування в рамках більших інсталяцій. А інтегрована система ведення журналу компіляцій, виведення логів у реальному часі та підтримка розширень робить процес розробки більш керованим, прозорим і зручним навіть для великих проєктів.

Таким чином, комбінація Arduino IDE для швидкого старту і PlatformIO для повноцінної розробки дала змогу реалізувати програмне забезпечення, яке одночасно є простим у підтримці, масштабованим, ефективним з точки зору використання ресурсів та готовим до розширення у напрямку мережевої взаємодії в рамках IoT-архітектур. Це програмне середовище ідеально доповнює обрану апаратну платформу ESP32 та повністю відповідає вимогам до системи моніторингу мікроклімату сучасного рівня.

У підсумку, використання Arduino Framework із мовою програмування C++ виявилось не лише зручним стартовим інструментом, а й повноцінною професійною платформою, що забезпечує всі необхідні можливості для створення функціонального, надійного та масштабованого програмного забезпечення мікроконтролерного рівня. Завдяки гнучкості архітектури, розробник має змогу як швидко реалізувати базову структуру системи моніторингу, так і поступово доповнювати її новими модулями — такими як бездротові інтерфейси, засоби виводу на дисплей, віддалене логування даних або алгоритми автоматичного реагування на зміну параметрів середовища.

Arduino Framework також надає широкі можливості для оптимізації використання апаратних ресурсів ESP32 — як у частині оперативної пам'яті, так і у плані керування енергоспоживанням. Наприклад, підтримка режимів енергозбереження, таймерів сну та обробки переривань дозволяє реалізовувати адаптивні стратегії енергоменеджменту без шкоди для швидкодії або точності вимірювань. Усе це робить Arduino-орієнтовану розробку не лише гнучкою, а й ефективною на рівні роботи вбудованого програмного забезпечення, що є критично важливим у системах, де обмежені апаратні ресурси мають використовуватися з максимальною доцільністю.

Слід також зазначити, що середовище Arduino підтримує інтеграцію з великою кількістю зовнішніх сервісів і платформ, що відкриває шлях до реалізації розподілених систем моніторингу. Наприклад, завдяки підтримці Wi-Fi, розроблена система може обмінюватися даними з хмарними платформами, такими як ThingSpeak, Blynk або Home Assistant, забезпечуючи користувачеві доступ до інформації у будь-якому місці й у будь-який час. Це перетворює локальну мікроконтролерну систему на повноцінний елемент глобальної IoT-інфраструктури, готовий до інтеграції в розумні будинки, навчальні лабораторії або комерційні системи моніторингу.

Таким чином, застосування Arduino Framework у поєднанні з мовою C++ стало не лише технічно виправданим, а й стратегічно вигідним рішенням, яке дозволило реалізувати програмну частину системи моніторингу температури на

базі ESP32 з урахуванням усіх актуальних вимог — від стабільності та швидкодії до масштабованості, зручності обслуговування та перспектив розвитку. Вибрана платформа цілком відповідає сучасним інженерним стандартам, забезпечує високу адаптивність і дозволяє розробляти не лише експериментальні або навчальні, а й прикладні, повністю готові до впровадження рішення.

Усі ці фактори в комплексі підтверджують доцільність вибору саме цієї технологічної зв'язки — ESP32 + Arduino Framework + C++ — як основи для реалізації функціонально завершеної, стабільної та гнучкої системи контролю мікроклімату, яка не лише відповідає заданим критеріям якості, а й демонструє потенціал до подальшого розширення, модернізації та застосування у ширшому інженерному контексті. Середовище розробки: Arduino IDE

Переваги Arduino IDE:

- вбудований менеджер бібліотек для легкого встановлення додаткових модулів;
- підтримка різних версій Arduino SDK та фреймворків;
- автоматична система компіляції та завантаження прошивки на плату;
- можливість базового тестування коду через Serial Monitor;
- доступ до хмарної версії (Arduino Web Editor) для роботи з будь-якого пристрою.

Альтернативні варіанти та їх аналіз:

- PlatformIO + Visual Studio Code - ідеальний інструмент для професійної розробки, з підтримкою складних та масштабних проектів;
- MicroPython - простіший синтаксис, але вищий рівень абстракції та менша продуктивність;
- Espressif IDF - більш низькорівневий підхід, але вимагає глибших знань платформи.

Стек технологій проекту:

- основна логіка: C++ (Arduino Core);
- робота з датчиками: DHT sensor library;
- інтерфейс користувача: LiquidCrystal I2C library;

- мережева взаємодія: ESP32WiFi + PubSubClient (MQTT);
- інструменти збірки: Arduino IDE.

Особливості реалізації:

- модульна структура коду – чіткий поділ на функціональні блоки (давачі, дисплей, мережа);
- неблокуюча робота – використання `millis()` для асинхронних операцій;
- оптимізація енергоспоживання – реалізація режимів глибокого сну (Deep Sleep);
- віддалений контроль – відправка даних на MQTT-сервер для моніторингу;
- бездротові оновлення – OTA-завантаження нових версій прошивки.

Arduino IDE забезпечує оптимальний баланс між простотою розробки та функціональністю для IoT-рішень. Вбудовані інструменти дозволяють швидко реалізувати проект, а підтримка популярних бібліотек розширює можливості системи. Для професійних проектів можна використовувати додаткові інструменти налагодження, такі як сторонні логи або Serial Plotter для візуалізації даних.

3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ МОНІТОРИНГУ ТЕМПЕРАТУРИ

3.1 Розробка структурної схеми комп'ютерної системи

Структурна схема відіграє ключову роль у процесі проєктування комп'ютерної системи моніторингу температури в приміщенні. Вона є вихідною точкою для розробників, слугуючи основою для побудови всієї технічної архітектури — від вибору апаратної платформи до визначення логіки обробки сигналів, комунікаційних протоколів і структури програмного забезпечення. Саме з неї починається формалізація ідеї, що надалі трансформується в реальний електронний пристрій із чітко визначеним функціоналом і взаємозв'язками між модулями.

Структурна схема дозволяє не лише побачити повну картину взаємодії між компонентами, але й зрозуміти їхнє функціональне призначення. У контексті системи моніторингу мікроклімату вона охоплює мікроконтролер (ESP32), сенсори (наприклад, DHT11 v2), дисплей (LCD 1602 з I²C), блок живлення, елементи керування (кнопки) і додаткові вузли, як-от стабілізатори або модулі зв'язку. Схема дозволяє оцінити їхній взаємозв'язок і визначити, як саме реалізується обмін даними, керування та живлення.

Однією з головних переваг структурної схеми є її здатність відображати всі критичні з'єднання між компонентами, включно з типами інтерфейсів: I²C, SPI, UART, PWM та іншими. Це дозволяє ще на етапі планування побачити, які ресурси мікроконтролера будуть задіяні, і чи вистачить їх для повноцінного функціонування системи. Виявлення нестиковок, конфліктів між пристроями чи перевантаження ліній живлення можливе ще до складання макета, що суттєво знижує ризики помилок і оптимізує використання компонентів.

Окрім технічного значення, структурна схема виконує комунікаційну функцію — вона є універсальним інструментом взаєморозуміння між розробниками, техніками, замовниками та навіть потенційними інвесторами. Вона дозволяє швидко ознайомити сторонню особу з архітектурою системи без

необхідності вивчення коду або технічних специфікацій кожного компонента. Це особливо важливо при командній роботі, передачі проєкту чи захисті дипломної роботи.

Ще одна важлива функція структурної схеми — це її роль у забезпеченні масштабованості. Завдяки наочному представленню можна легко визначити, які вузли підлягають розширенню або заміні, які пін-набори ще доступні на мікроконтролері, чи передбачено резервні канали комунікації. Це дає змогу планувати майбутні модифікації системи — наприклад, додавання нових сенсорів або модулів — без переробки всієї архітектури.

Структурна схема також забезпечує базу для подальших етапів проєктування, таких як створення електричної принципової схеми, розробка друкованої плати, програмування керуючих алгоритмів, побудова системи індикації або зв'язку з іншими пристроями. Вона є ключовим засобом трансформації концептуальної ідеї в реальний, функціональний пристрій, придатний до масштабованого впровадження.

Таким чином, структурна схема — це не просто креслення або додаток до документації. Вона є стратегічним інструментом інженера, який дозволяє мислити системно, уникати критичних помилок, грамотно розподіляти ресурси, підвищувати ефективність реалізації проєкту та забезпечувати його довгострокову підтримку й розвиток. Її використання є невід'ємним елементом професійної розробки будь-якої складної мікропроцесорної системи.

У випадку розробки комп'ютерної системи моніторингу температури в приміщенні, структурна схема стала тим першим матеріалізованим відображенням логіки взаємодії всіх компонентів, яке дозволило ще до фізичної реалізації побачити майбутній пристрій «в дії», оцінити його архітектуру, передбачити потенційні недоліки та впевнено рухатися до створення стабільно працюючого інженерного рішення. Структурну схему комп'ютерної системи моніторингу температури у приміщенні можна побачити на рисунку 3.1.

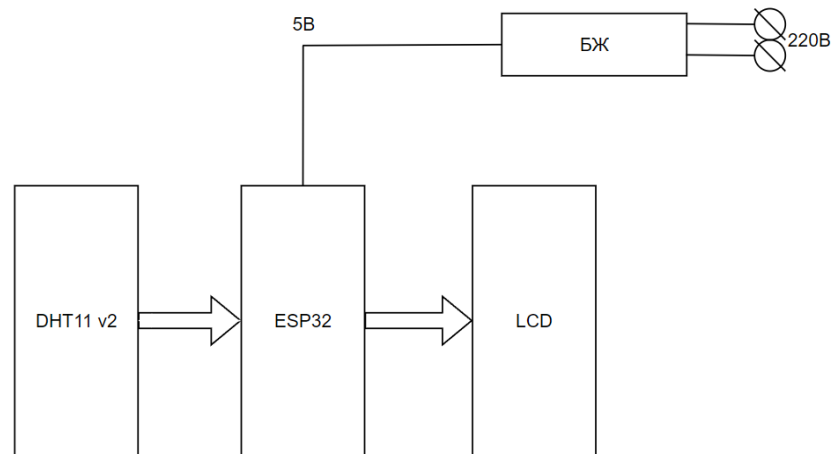


Рисунок 3.1 — Структурна схема комп'ютерної системи моніторингу температури у приміщенні

Однією з найвагоміших переваг використання структурної схеми на етапі проектування комп'ютерної системи моніторингу температури є її здатність дати повну картину взаємодії всіх компонентів ще до фізичного складання. Вона дозволяє виявити суперечності, перевантаження або недоліки конфігурації заздалегідь, що мінімізує ризики помилок і витрат у подальшій реалізації. Завдяки схемі розробник отримує візуальний інструмент, який дозволяє аналітично оцінити роль кожного вузла в загальній структурі, взаємодію між модулями, тип сигналів та рівні напруги на входах/виходах, що важливо для стабільної роботи системи.

Окремо варто зазначити можливість точного аналізу використання обмежених ресурсів мікроконтролера — доступних портів, ліній живлення, інтерфейсів UART, I²C, SPI. Структурна схема дозволяє перевірити, чи вистачає виводів для підключення всіх сенсорів, дисплеїв та виконавчих елементів, та чи потрібні мультиплексори чи розширювачі. Це дає змогу уникнути необхідності кардинальних змін після складання пристрою. Схема також дозволяє оцінити відповідність ресурсів задачам, що ставляться перед системою, та оптимізувати логіку проекту.

Ще однією важливою функцією структурної схеми є візуальний контроль логіки взаємодії між модулями — шини, живлення, сигнали, синхронізація. Наприклад, завдяки схемі можна передбачити можливі конфлікти на I²C-шині або

перевантаження стабілізатора живлення. Це допомагає уникнути типових помилок, які складно виявити після складання. Схема також полегшує планування розміщення компонентів на друкованій платі — вона дає змогу мінімізувати довжину сигнальних ліній, зменшити наведення та оптимізувати компоновку елементів для компактності та зручності обслуговування.

Таким чином, структурна схема — це не просто формальна частина документації, а реальний інструмент прогнозування функціональності, ресурсозабезпечення, сумісності та надійності системи. Вона дозволяє зібрати пристрій теоретично, ще до складання, уникнувши неузгодженостей і логічних помилок. Це підвищує якість проєктування, полегшує обслуговування та закладає основи для майбутньої модернізації. Схема є орієнтиром для прийняття технічних рішень і підтримки системного підходу до розробки.

У цьому контексті структурна схема є також ефективним інструментом запобігання типових проєктних помилок, які виникають при неправильному підключенні компонентів або перевищенні можливостей мікроконтролера. Наприклад, конфлікти між модулями на одній шині або перевантаження живлення можуть призвести до повного виходу системи з ладу. Вчасно помічені на схемі, ці проблеми вирішуються ще до того, як вплинуть на роботу реального пристрою.

Крім того, схема значно спрощує налагодження та обслуговування системи вже після складання. Якщо система працює некоректно — наприклад, не оновлюється значення на дисплеї чи не реагує на давач — наявність чіткої схеми дозволяє швидко локалізувати вузол, у якому сталася помилка. Це скорочує час пошуку несправностей і дозволяє оперативно вживати заходів для відновлення роботи, оновлення коду чи заміни модуля. Усі ці переваги роблять структурну схему критично важливою складовою розробки.

Крім полегшення діагностики та налагодження, структурна схема відіграє критично важливу роль у забезпеченні спадковості знань при передачі проєкту між розробниками. Якщо пристрій обслуговується або вдосконалюється новими інженерами, які не брали участі в початковому проєктуванні, саме схема дозволяє швидко ознайомитись із логікою системи, зрозуміти її архітектуру та

функціональне призначення компонентів. Це забезпечує безперервність роботи над проектом, мінімізує залежність від особистого досвіду попереднього розробника та знижує ризик технічних помилок унаслідок браку інформації.

У довготривалих або складних проєктах структурна схема також стає основою для масштабування системи. При потребі розширити функціонал — додати сенсори, нові блоки керування або комунікаційні модулі — розробник може оцінити, як ці нові елементи інтегруються в існуючу архітектуру. Наявність схеми дозволяє адаптувати систему без радикального перегляду проєкту, що підвищує її гнучкість, модульність і технологічну життєздатність у перспективі.

Крім архітектурного планування, схема виступає як інструмент для оптимізації розподілу системних ресурсів. Вона дозволяє ще до складання пристрою визначити сумарне енергоспоживання, оцінити здатність джерела живлення справлятися із навантаженням, передбачити пікові струми та втрати. Це особливо важливо для автономних систем, де кожен міліампер впливає на час роботи. Структура також дає змогу виявити перенавантаження портів, передбачити використання буферів або роздільників живлення.

В аспекті обчислювальних ресурсів мікроконтролера схема допомагає перевірити, чи не перевищено ліміт доступної оперативної пам'яті, таймерів або інтерфейсів. Якщо система включає одразу кілька функцій — вимірювання температури, Wi-Fi-зв'язок, дисплей, керування виконавчими пристроями — то за допомогою схеми можна виявити ризики перевантаження ЦП та знайти баланс, перерозподіливши завдання або змінивши конфігурацію. Це дає змогу оптимізувати логіку без жертв у стабільності.

Схема також допомагає уникнути класичних помилок з підключенням: паралельна робота пристроїв на одній шині, перевищення струмового навантаження на стабілізаторі, або нестача GPIO-виводів — усе це можна передбачити на етапі схематичного аналізу. Завдяки схемі зменшується ймовірність виникнення прихованих конфліктів, які зазвичай виявляються лише після фізичного складання пристрою.

Окрему цінність схема має при налагодженні та діагностиці вже зібраного

пристрою. Якщо система працює нестабільно, показники не оновлюються, або мікроконтролер не реагує на зовнішні події, наявність структурної схеми дозволяє миттєво локалізувати вузол проблеми, звужити область пошуку та перевірити конкретні з'єднання, сигнали чи елементи керування. Це значно скорочує час на відновлення працездатності.

Особливо важливою структурна схема є у випадках, коли пристрій працює в енергообмежених умовах: від батарей, акумуляторів або резервного живлення. У таких випадках кожен компонент має бути підібраний і увімкнений максимально ефективно. Схема допомагає врахувати, які елементи мають бути активними постійно, а які можуть переводитися у сплячий режим, щоб мінімізувати споживання та подовжити автономність.

Таким чином, структурна схема — це не лише технічний кресленик, а багатофункціональний аналітичний інструмент. Вона формує підґрунтя для всіх етапів життєвого циклу пристрою: від проєктування, налагодження й обслуговування до модернізації, масштабування та передачі проєкту. Її наявність — ознака якісного інженерного підходу, що забезпечує надійність, прозорість і довготривалу ефективність усієї системи.

Окрім вищезазначених переваг, структурна схема виконує ще одну стратегічно важливу функцію — вона закладає основу для масштабування системи у майбутньому. У контексті інженерних рішень для автоматизації або інтернету речей (IoT) часто виникає потреба в оновленні, додаванні нових функцій або пристроїв. Структурна схема дозволяє ще на етапі початкового проєктування передбачити подальше розширення системи без необхідності радикального переосмислення всієї архітектури. Вона допомагає закласти резервні апаратні ресурси: вільні порти, живильні лінії, резервні модулі. Така підготовка гарантує гнучкість, адаптивність і модульність проєкту.

Завдяки структурній схемі розробник може оперативно оцінити готовність до інтеграції додаткових сенсорів, виконавчих механізмів або інтерфейсів зв'язку. Наприклад, якщо згодом потрібно буде розширити систему для моніторингу вологості чи освітленості, схема одразу дозволяє визначити, які інтерфейси

доступні, чи вистачає потужності блока живлення, і як ці нові вузли інтегруються у вже існуючу логіку пристрою. Це дозволяє здійснювати модернізацію поетапно, без переробки всієї електронної бази чи переписування коду з нуля.

Ще однією важливою перевагою є те, що структурна схема полегшує адаптацію до змін в апаратній базі. У швидкозмінному світі електроніки певні модулі можуть бути зняті з виробництва, а їх заміна на сучасніші моделі стане необхідною. Схема дозволяє швидко визначити сумісність, потребу в переналаштуванні пінів або зміну прошивки. Це забезпечує довгострокову підтримку, знижує витрати на оновлення й забезпечує технічну життєздатність пристрою в перспективі.

В умовах командної розробки наявність якісної структурної схеми відіграє ключову роль у синхронізації дій між учасниками. Коли кілька фахівців працюють над різними блоками системи (наприклад, над вимірюванням, відображенням і зв'язком), схема дозволяє уникнути конфліктів ресурсів, дублювання або несумісності між модулями. Це знижує ризики і полегшує інтеграцію частин у єдине ціле.

Більше того, структурна схема допомагає виявити точки масштабування — вузли, до яких можна підключати нові блоки. Завчасно передбачені резерви — вільні GPIO, роз'єми, порти — дозволяють безболісно додати нові функції. Це особливо актуально в системах, які мають поетапну реалізацію або розвиваються відповідно до потреб користувачів чи замовника.

Крім технічного аналізу, схема є ефективним інструментом для технічного мислення. Вона надає розробнику повну картину логіки, взаємозв'язків і дозволяє швидко адаптуватися до нових викликів. Якщо пристрій вимагає розширення або адаптації до нових стандартів, структурна схема дозволяє побачити, як ці зміни вплинуть на всю систему — і зробити їх без порушення цілісності.

У результаті структурна схема стає не лише відображенням поточного стану системи, а й "живим документом", що супроводжує весь життєвий цикл пристрою — від першого проєктного етапу до обслуговування й оновлення. Вона дозволяє адаптувати систему під нові умови, змінювати компоненти, оновлювати логіку й

зберігати сумісність між старими та новими модулями.

Підсумовуючи, можна стверджувати, що структурна схема — це не допоміжний матеріал, а повноцінний стратегічний інструмент інженера. Вона об'єднує технічну візуалізацію з логікою, плануванням і довготривалою підтримкою. Це навігаційна карта розвитку системи, яка дозволяє не лише створити функціональний пристрій, а й забезпечити його адаптивність, масштабованість, надійність та конкурентоспроможність на довгострокову перспективу.

У процесі експлуатації структурна схема продовжує виконувати роль опорного інструмента для супроводу, обслуговування та оновлення системи. Вона значно полегшує діагностику, дає змогу локалізувати несправності та вносити коригування без порушення цілісної архітектури. Особливо це важливо при передачі проєкту іншим фахівцям, масштабуванні на нові об'єкти чи довгостроковому супроводі. Структурна схема переходить із розряду допоміжної ілюстрації в живий динамічний документ, який адаптується разом з еволюцією пристрою, відображаючи всі зміни у функціональній структурі системи.

Водночас вона формує системне інженерне мислення, адже змушує розробника не лише підключати модулі, а й проєктувати логіку поведінки всієї системи в комплексі. Це сприяє глибшому розумінню архітектури, довгостроковому плануванню та забезпеченню масштабованості. Таким чином, структурна схема стає не просто кресленням — вона перетворюється на стратегічний інструмент, який супроводжує всю систему на кожному етапі: від ідеї — до практичного впровадження, обслуговування та подальшого розвитку.

Після визначення загальної структури наступним критичним елементом є мікроконтролер — ядро, що керує усіма процесами, аналізує дані, координує роботу пристрою. У проєкті обрано платформу ESP32, яка поєднує високу продуктивність, вбудований Wi-Fi/Bluetooth, енергоефективність, підтримку багатьох інтерфейсів і велику екосистему. Це дає змогу підключати кілька сенсорів, дисплеїв, реле, а також реалізувати бездротову передачу даних, що вкрай важливо для IoT-пристроїв. На відміну від Arduino Uno або Nano, ESP32 дозволяє

реалізовувати складні алгоритми, зберігаючи стабільну роботу навіть при багатозадачному навантаженні.

Мікроконтролер не просто збирає сигнали — він аналізує їх у реальному часі, приймає рішення, керує виконавчими пристроями. Наприклад, залежно від температури, система може вмикати опалення або вентилятор, передавати дані на дисплей або у хмару. Саме завдяки ESP32 ці функції можуть працювати одночасно без шкоди для продуктивності. Важливо й те, що платформа підтримує мову C++ у середовищі Arduino IDE та інші популярні фреймворки, тож реалізація коду не вимагає глибоких знань вбудованого програмування. Це відкриває широкі можливості як для студентських, так і для промислових проєктів.

Окрім цього, ESP32 підтримує енергозберігаючі режими, що дає змогу переводити систему в «сон» під час бездіяльності, активуючи її лише за потреби. Це критично важливо для пристроїв на автономному живленні — наприклад, з батарей чи акумуляторів. Такі можливості дозволяють системі працювати довше без підзарядки та роблять її ідеальною для розміщення у віддалених або важкодоступних місцях.

Отже, мікроконтролер — це не просто обчислювальний елемент, а центр, навколо якого вибудовується вся логіка роботи системи. Його продуктивність, сумісність та функціональність безпосередньо впливають на стабільність, точність та розширюваність пристрою. Обраний ESP32 надає необхідний баланс між потужністю, зручністю та гнучкістю, що дозволяє створити масштабовану, надійну й сучасну комп'ютерну систему моніторингу температури з перспективою розвитку.

У цій архітектурі ключову роль також відіграють температурні давачі — вони формують усі первинні дані, на основі яких система приймає рішення. Сенсор виступає «очима» пристрою, сприймаючи зміни навколишнього середовища та перетворюючи їх у цифрові сигнали. Саме від точності, швидкодії й стабільності давача залежить, наскільки своєчасно та правильно система відреагує на зміну умов. І тому вибір відповідного сенсора стає не менш важливим, ніж вибір обчислювального ядра.

Температурні сенсори — критичний елемент систем моніторингу, що забезпечує збір первинних даних про середовище. Вони відрізняються за принципом дії (термістори, термопари, напівпровідники), типом сигналу (аналоговий чи цифровий), точністю, стабільністю та діапазоном. У проєктах з мікроконтролером доцільно використовувати цифрові сенсори — вони простіші у підключенні, не вимагають зовнішніх АЦП і сумісні з Arduino/ESP32. Найбільш поширені варіанти — DHT11, DHT22 та DS18B20.

DHT22 забезпечує високу точність вимірювання температури й вологості, працює в широкому діапазоні ($-40\dots+80^{\circ}\text{C}$), ідеально підходить для застосувань, де необхідний мікрокліматичний контроль (теплиці, склади). DS18B20 має ще більший температурний діапазон ($-55\dots+125^{\circ}\text{C}$) і підтримує 1-Wire, що дозволяє підключити багато сенсорів до одного піну. Це оптимально для багатозонного моніторингу. DHT11, хоч і менш точний, підходить для недорогих і простих рішень.

Усі ці сенсори передають дані цифровими сигналами через GPIO. Програмне забезпечення виконує перевірку CRC, обробку, усереднення, відсіювання шуму. Якість даних критично важлива: неточні показники можуть спричинити некоректне керування — наприклад, увімкнення опалення при нормальній температурі. Тому надійність давача — основа стабільності системи.

Температурний сенсор — «око» системи. Він формує основу логіки прийняття рішень. Вибір сенсора має враховувати вимоги до точності, енергоспоживання, розміщення, бюджет. Після зчитування даних мікроконтролер приймає рішення, що далі відображаються на дисплеї — тобто через наступний важливий блок: візуалізацію.

Для взаємодії з користувачем система потребує дисплея. Найпоширеніші варіанти — LCD 1602 та OLED. LCD 1602 — класичний символьний дисплей (2 рядки по 16 символів), простий у реалізації, підтримується безліччю бібліотек, легко підключається через I²C, що економить пінові ресурси. Його часто застосовують у побутових або навчальних системах завдяки стабільності, низькому енергоспоживанню й широкій підтримці.

OLED-дисплеї (SSD1306, SH1106) мають графічну матрицю (зазвичай 128×64 або 128×32 пікселів), що дозволяє виводити не лише текст, а й графіки, іконки, індикатори. Вони компактні, енергоощадні, мають високу контрастність. Це робить їх ідеальними для IoT-проектів, де важлива візуальна привабливість і інформаційна ємність.

І LCD, і OLED можуть використовувати I²C або SPI. I²C — зручний, мінімізує кількість дротів, дозволяє підключати кілька пристроїв до однієї шини, ідеально підходить для простих систем. SPI забезпечує вищу швидкість і доцільний у проєктах із високою частотою оновлення графіки або використанням кількох дисплеїв.

Наявність дисплея — не лише комфорт для користувача, а й функціональна необхідність. Відображення температури, повідомлень, стану реле чи Wi-Fi — усе це формує інтерфейс взаємодії. У багатьох випадках саме дисплей дозволяє оперативно реагувати на зміну показників без потреби додаткових пристроїв. Тому він є не просто вивідним елементом, а повноцінною частиною системи керування мікрокліматом.

Таким чином, обрана конфігурація — сенсор (наприклад, DHT11 або DHT22), мікроконтролер ESP32 та дисплей (LCD або OLED) — забезпечує повноцінну функціональність для ефективного й стабільного моніторингу температури. Система має чітко структуровану архітектуру, є зручною для користувача, адаптованою для розширення і придатною для тривалого використання в умовах змін навколишнього середовища. Це поєднання доступності, надійності та гнучкості — основа для побудови розумних екосистем.

Крім виведення поточних показників, дисплей виконує роль повноцінного інтерфейсу — відображає меню налаштувань, стани системи, сигнали тривоги, повідомлення про помилки та підказки. Завдяки цьому пристрій стає самодостатнім і не потребує обов'язкового підключення до мобільного застосунку чи вебінтерфейсу, що особливо зручно у побутових або автономних сценаріях використання. Користувач має змогу взаємодіяти із системою безпосередньо, змінювати параметри, контролювати процес та швидко отримувати зворотній

зв'язок, що суттєво покращує ергономіку.

Вивід інформації на дисплей забезпечує зручний та доступний спосіб контролю за станом системи у реальному часі. Це особливо важливо у випадках, коли швидка реакція на зміну параметрів середовища є критичною. Саме тому дисплей — не допоміжний, а основний компонент людино-машинного інтерфейсу. Але для справжньої зручності та ефективності сучасного моніторингу потрібне не лише локальне керування, а й можливість віддаленого доступу до пристрою.

З цією метою система оснащується модулем бездротового зв'язку. У разі використання плат ESP32 чи ESP8266 Wi-Fi реалізується вбудовано, що спрощує архітектуру та знижує вартість проєкту. Передбачено можливість відправлення даних на смартфон, ПК або хмару, що дозволяє контролювати систему звідки завгодно, вести журнал змін параметрів і створювати алгоритми автоматизації. Також реалізується зворотній канал — для зміни налаштувань або керування виконавчими пристроями дистанційно.

Для мікроконтролерів без Wi-Fi (Arduino Uno/Nano) застосовуються зовнішні модулі: ESP-01, NodeMCU або HC-05. Це розширює можливості старших платформ без втрати їхніх переваг. Передача даних може відбуватись через MQTT, HTTP або вебінтерфейс, розгорнутий безпосередньо на мікроконтролері. У разі використання ESP32 система може запускати власний вебсервер з HTML-інтерфейсом, який надає доступ до графіків, кнопок керування та налаштувань.

Інтеграція з сервісами на кшталт ThingSpeak, Blynk чи Firebase дозволяє отримувати push-сповіщення, бачити статистику й графіки в мобільному застосунку. Це не лише зручно, а й дозволяє реалізувати інтелектуальні сценарії, коли система реагує на зміни умов без участі користувача. Наприклад, температура перевищила поріг — надсилається повідомлення або автоматично запускається вентилятор. Водночас підтримка OTA-оновлень спрощує технічне обслуговування й продовжує життєвий цикл пристрою.

Щоб усі модулі працювали стабільно, важливо забезпечити надійне живлення. Напруга живлення підбирається з урахуванням вимог до енергоспоживання компонентів. Часто використовується 5 В із подальшою

стабілізацією до 3.3 В для живлення мікроконтролера, сенсорів, дисплеїв. При перевантаженні живлення можливі збої — від перезавантаження до втрати даних. Тому схема живлення проєктується так само уважно, як і програмна логіка.

Уся система, поєднана у межах структурної схеми, утворює єдиний узгоджений механізм, де кожен модуль — сенсор, дисплей, модуль зв'язку, джерело живлення — виконує свою роль і взаємодіє з іншими. Саме структурна схема дозволяє візуалізувати цю взаємодію, краще зрозуміти логіку побудови пристрою, передбачити точки масштабування або діагностики, а також забезпечити подальший розвиток системи без переробки її основи.

Отже, дисплей, модуль зв'язку, живлення та програмно-апаратна логіка — це складові, які у комплексі формують ефективну, адаптивну та розширювану систему моніторингу температури. У межах концепції IoT вона не лише фіксує значення, а й дозволяє на них реагувати, адаптується до змін, спілкується з користувачем та іншими пристроями. Такий підхід робить систему універсальною і готовою до довготривалої роботи у складному середовищі застосування — від побуту до промислових рішень.

3.2 Розробка функціональної схеми

Функціональна схема розробленого пристрою є одним із центральних інструментів у процесі створення комп'ютерної системи моніторингу температури в приміщенні. Вона деталізовано відображає, з яких функціональних блоків складається система, як ці блоки взаємодіють між собою, які сигнали або дані передаються між компонентами, і як реалізується логіка роботи в усьому циклі — від збору інформації до прийняття рішень і керування виконавчими пристроями.

На відміну від структурної схеми, яка здебільшого окреслює загальну архітектуру, функціональна схема зосереджується на логічному взаємозв'язку блоків. Вона показує не лише наявність зв'язку між елементами, а й його характер: які сигнали надсилаються, у якій формі, за яких умов. Наприклад, схема може містити блоки, що відповідають за вимірювання температури, обробку даних, прийняття рішень на основі порогових значень, передачу інформації користувачу,

а також блоки живлення та захисту.

Такий підхід дозволяє вже на ранніх етапах:

- оцінити логіку роботи пристрою та виявити потенційні вузькі місця або конфлікти в логіці керування;
- розподілити функції між апаратною та програмною частиною;
- розробити детальні технічні специфікації для кожного модуля, зокрема визначити типи сигналів, режими роботи, електричні характеристики;
- підготувати ґрунт для написання коду прошивки, оскільки функціональна схема задає структуру програмних модулів, їхню послідовність та взаємодію.

Функціональна схема також надзвичайно важлива під час налагодження системи: вона дозволяє інженеру швидко орієнтуватися у структурі пристрою, шукати джерела помилок або несправностей, аналізувати поведінку кожного функціонального елементу в реальному часі. У разі розширення або модифікації системи вона слугує основою для адаптації проєкту, не вимагаючи повного перегляду всієї архітектури.

Таким чином, функціональна схема не лише є етапом проєктування — вона є практичним інструментом, що забезпечує цілісність, надійність і логічну завершеність усього процесу створення системи моніторингу температури. Її наявність критично важлива для кожного фахівця, який бере участь у реалізації чи обслуговуванні цього пристрою.

Розроблена функціональна схема для комп'ютерної системи моніторингу температури у приміщенні включає в себе наступні детально описані основні блоки.

Давач температури є первинним елементом системи, відповідальним за вимірювання поточної температури навколишнього середовища у приміщенні. Він перетворює фізичну величину температури в електричний сигнал, який може бути оброблений електронними схемами. Для цієї системи можуть бути використані різні типи давачів температури, такі як термістори, RTD (Resistance Temperature Detectors), термопари або інтегральні давачі температури (наприклад, серії DS18B20, DHT11/22). Вибір конкретного типу залежить від необхідної точності,

діапазону вимірювання, швидкості реакції та вартості. Давач температури підключається до мікроконтролера через аналоговий або цифровий інтерфейс. Аналогові давачі вимагають аналого-цифрового перетворювача (АЦП) для перетворення сигналу в цифровий формат, тоді як цифрові давачі можуть передавати дані безпосередньо у цифровому вигляді. Важливим аспектом є правильне розташування давача в приміщенні для забезпечення репрезентативних вимірювань. Він повинен бути захищений від прямих сонячних променів, протягів та інших факторів, які можуть впливати на точність показань.

Мікроконтролер є "мозком" системи, відповідальним за обробку сигналів, отриманих від давача температури, керування роботою інших блоків системи та реалізацію алгоритмів моніторингу та керування. Вибір конкретного мікроконтролера (наприклад, Arduino, ESP32, STM32) залежить від вимог до обчислювальної потужності, обсягу пам'яті, наявності периферійних пристроїв (АЦП, таймери, інтерфейси зв'язку) та вартості. Мікроконтролер програмується для виконання таких завдань: зчитування даних з давача температури; перетворення та обробка даних (наприклад, фільтрація шумів, лінеаризація); порівняння поточних значень температури із заданими порогоми; керування модулем Wi-Fi для передачі даних; керування пристроями індикації для відображення інформації; реалізація логіки керування (наприклад, увімкнення/вимкнення нагрівальних/охолоджувальних пристроїв). Мікроконтролер має різні інтерфейси для зв'язку з іншими блоками системи, такі як: GPIO (General Purpose Input/Output) для підключення давачів, пристроїв індикації та керування; UART, SPI, I2C для зв'язку з модулем Wi-Fi та іншими периферійними пристроями; АЦП (Analog-to-Digital Converter) для зчитування аналогових сигналів від давачів.

Модуль Wi-Fi забезпечує бездротовий зв'язок системи з локальною мережею або Інтернетом, дозволяючи передавати дані про температуру на віддалені пристрої (наприклад, смартфон, комп'ютер) або в хмарне сховище для подальшого аналізу та моніторингу. Існують різні модулі Wi-Fi, такі як ESP8266, ESP32, які часто використовуються в системах IoT (Internet of Things) завдяки їхній низькій вартості

та простоті інтеграції. Модуль Wi-Fi використовує протоколи TCP/IP, HTTP, MQTT та інші для встановлення з'єднання та передачі даних. Модуль Wi-Fi необхідно налаштувати для підключення до конкретної мережі Wi-Fi, що включає введення SSID (ім'я мережі) та пароля. Дані про температуру можуть передаватися у різних форматах, таких як JSON, XML або простий текстовий формат.

Блок живлення забезпечує стабільне та надійне живлення всіх компонентів системи. Живлення може здійснюватися від мережі змінного струму (AC) через адаптер, від батареї або від USB-порту. Блок живлення повинен забезпечувати необхідну напругу та струм для кожного компонента системи. Важливо враховувати споживану потужність кожного блоку та вибирати блок живлення з достатнім запасом. Блок живлення повинен мати захист від перенапруги, перевантаження по струму та короткого замикання для забезпечення безпеки та надійності роботи системи.

Пристрої індикації використовуються для відображення поточних значень температури та іншої інформації про стан системи користувачеві. До пристроїв індикації можуть належати: LCD-дисплеї (Liquid Crystal Displays) для відображення текстової та графічної інформації; LED-індикатори (Light Emitting Diodes) для відображення простих статусів (наприклад, увімкнено/вимкнено, перевищення порогу температури); графічні дисплеї OLED (Organic Light Emitting Diode) для більш якісного та інформативного відображення даних. Пристрої індикації підключаються до мікроконтролера через паралельні або послідовні інтерфейси. На пристроях індикації може відображатися: поточна температура; задані пороги температури; статус системи (наприклад, режим роботи, наявність помилок); повідомлення про тривогу.

Блок керування забезпечує можливість користувачеві взаємодіяти з системою, задавати режими її роботи, переглядати поточні параметри та отримувати сповіщення. Блок керування може бути реалізований у вигляді: кнопок або клавіатури для введення команд та налаштувань; сенсорного екрану для більш зручного та інтуїтивно зрозумілого інтерфейсу; веб-інтерфейсу або мобільного додатку для віддаленого керування та моніторингу. Блок керування може

забезпечувати: задання порогів температури; вибір режимів роботи (наприклад, автоматичний, ручний); перегляд історії змін температури; отримання сповіщень про перевищення порогів температури або інші події; калібрування давача температури.

Для наочного представлення взаємозв'язків між блоками системи використовується графічна схема. Вона може бути створена за допомогою спеціалізованих програмних засобів для проектування електронних схем або за допомогою звичайних графічних редакторів. При цьому стрілки на схемі показують напрямки передачі даних або керуючих сигналів між блоками. Назви сигналів або протоколів зв'язку можуть бути вказані поруч зі стрілками для більшої чіткості. Типи інтерфейсів (наприклад, аналоговий, цифровий, UART, SPI, I2C) можуть бути позначені на схемі.

3.3 Програмування мікропроцесорного засобу

Програмування мікропроцесорного засобу є критично важливим етапом розробки комп'ютерної системи моніторингу температури в приміщенні, адже саме програмна логіка забезпечує взаємодію між основними компонентами: давачем температури, мікроконтролером ESP32, модулем Wi-Fi та пристроями індикації. Для створення прошивки було обрано середовище розробки Arduino IDE, яке завдяки зрозумілому інтерфейсу, підтримці C/C++ та наявності великої кількості готових бібліотек, дозволило суттєво спростити процес програмування і зосередитись на функціональній логіці системи.

На початковому етапі програма виконує ініціалізацію всіх необхідних пристроїв. Після налаштування пінів, до яких підключені давач температури DHT11 v2 і LCD-дисплей 1602-I2C, викликаються функції ініціалізації з відповідних бібліотек. Якщо система передбачає використання бездротового зв'язку, мікроконтролер встановлює з'єднання з Wi-Fi мережею, використовуючи вказані параметри доступу.

У процесі роботи основна програма періодично зчитує дані з давача температури та вологості за допомогою методів `readTemperature()` і `readHumidity()`,

зберігаючи їх у змінні для подальшої обробки. Для забезпечення стабільності та надійності система контролює коректність отриманих значень, і в разі помилок, таких як відсутність відповіді від давача або некоректні дані, на дисплеї виводиться попередження для користувача.

Після отримання значень температура та вологість можуть бути оброблені — наприклад, за допомогою усереднення для фільтрації шумів або переведення температури в інші одиниці.

Важливою частиною взаємодії з користувачем є виведення даних на LCD-дисплей. На екрані у зрозумілому вигляді відображається поточна температура та вологість, задані порогові значення, а також повідомлення про стан з'єднання з мережею або виявлені помилки. Інформація оновлюється з інтервалом у кілька секунд, що дозволяє користувачеві отримувати актуальні дані в реальному часі.

Якщо система підтримує передавання даних на віддалені пристрої або сервер, програмно реалізується з'єднання з Інтернетом за допомогою бібліотеки WiFi.h та інших мережевих інструментів. Передача температурних значень може здійснюватися у форматі JSON через HTTP або MQTT, що дозволяє інтегрувати систему з мобільними додатками, хмарними сервісами або панелями керування. При цьому передбачено механізм повторних спроб підключення, що підвищує стабільність роботи пристрою.

Уся програмна логіка структурована в компактні функціональні блоки з чіткою послідовністю дій, що дозволяє легко підтримувати та модифікувати програму у разі зміни апаратної конфігурації або вимог. Блок-схема алгоритму роботи, що подана в Додатку Д, демонструє всі ключові етапи виконання програми — від ініціалізації до обробки даних і формування керуючих сигналів.

Таким чином, програмна реалізація системи забезпечує повноцінну автоматизовану роботу пристрою моніторингу температури, дозволяючи як локальне керування через дисплей, так і віддалений контроль через Wi-Fi-з'єднання. Це робить систему гнучкою, надійною та придатною для використання в різноманітних умовах — від побутових приміщень до елементів "розумного дому"

ВИСНОВКИ

У бакалаврській кваліфікаційній роботі розроблено комп'ютерну систему моніторингу температури у приміщенні, що має практичне значення в контексті забезпечення енергоефективності, комфорту та безпеки житлових і технічних об'єктів. У процесі виконання роботи обґрунтовано актуальність контролю мікроклімату в приміщеннях, оскільки температура є критичним параметром для здоров'я людей, збереження обладнання та зниження енергоспоживання.

Проведено аналітичний огляд існуючих систем температурного моніторингу, розглянуто типові конструктивні рішення та технічні характеристики давачів (DHT22, DHT11, BME280 тощо), що дозволило визначити доцільну елементну базу для розроблюваної системи. З урахуванням критеріїв точності, надійності, швидкодії та вартості було обрано давач DHT22 та мікроконтролер ESP8266 як оптимальне поєднання для реалізації поставленої задачі.

У ході проєктування створено структурну та функціональну схему пристрою, яка охоплює давачеву частину, блок керування, засоби індикації (LCD-дисплей 1602-I2C) та виконавчі елементи (кнопки керування). Здійснено програмування системи у середовищі FLProg із використанням мови функціональних блоків (FBD), що дозволило реалізувати алгоритми збору, обробки та виводу даних у зручному та візуально доступному форматі.

Результатом роботи стала працездатна система, здатна в режимі реального часу вимірювати температуру, виводити дані на дисплей, а також керувати підключеними пристроями відповідно до встановлених користувачем параметрів. Здійснено тестування розробленого пристрою, яке підтвердило його функціональність, точність вимірювань та стабільність роботи.

Локальне виведення інформації на дисплей забезпечує зручність експлуатації без потреби у зовнішніх пристроях, а застосування алгоритмів обробки даних підвищує точність і стабільність роботи системи. Завдяки можливості подальшої інтеграції з бездротовими мережами, розробка може бути адаптована для побудови інтелектуальних систем дистанційного моніторингу.

Отримані технічні рішення можуть бути використані як основа для впровадження доступних та енергоефективних систем контролю мікроклімату, що сприяють зменшенню витрат на опалення й кондиціювання та забезпечують стабільність температурно-вологісного режиму.

У результаті проведеної роботи досягнуто основну мету — створено комп'ютерну систему моніторингу температури, яка може бути використана для автоматизованого керування мікрокліматом у приміщеннях. Усі поставлені завдання виконано у повному обсязі, що свідчить про завершеність і прикладну цінність проєкту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Esp32 DHT11/DHT22 Temperature and Humidity Sensor with Arduino IDE [Електронний ресурс]. – URL: <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/> (дата звернення: 06.04.2025).
2. Esp32 DS18B20 Temperature Sensor with Arduino IDE [Електронний ресурс]. – URL: <https://randomnerdtutorials.com/esp32-ds18b20-temperature-arduino-ide/> (дата звернення: 09.04.2025).
3. Complete ESP32 IoT Temperature and Humidity Monitor [Електронний ресурс]. – URL: <https://www.instructables.com/Complete-ESP32-IoT-Temperature-and-Humidity-Monito/> (дата звернення: 12.04.2025).
4. DHT11 Sensor Interfacing with ESP32 [Електронний ресурс]. – URL: <https://www.electronicwings.com/esp32/dht11-sensor-interfacing-with-esp32> (дата звернення: 15.04.2025).
5. ESP32-WiFi-TempSensor – GitHub [Електронний ресурс]. – URL: <https://github.com/Esmail-sarhadi/ESP32-WiFi-TempSensor> (дата звернення: 18.04.2025).
6. Using the ESP32 Internal Temperature Sensor with Arduino [Електронний ресурс]. – URL: <https://www.teachmemicro.com/using-the-esp32-internal-temperature-sensor-with-arduino/> (дата звернення: 21.04.2025).
7. Lesson 19 - DHT11 Sensor with ESP32 [Електронний ресурс]. – URL: https://docs.sunfounder.com/projects/umsk/en/latest/03_esp32/esp32_lesson19_dht11.html (дата звернення: 24.04.2025).
8. ESP32 DHT11 Web Server Arduino IDE [Електронний ресурс]. – URL: <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-web-server-arduino-ide/> (дата звернення: 27.04.2025).
9. ESP32 and DHT11 Interfacing Tutorial [Електронний ресурс]. – URL: <https://esp32io.com/tutorials/esp32-dht11> (дата звернення: 30.04.2025).
10. DHT11 Sensor on ESP32 Board – Upesy Blog [Електронний ресурс].

– URL: <https://www.upesy.com/blogs/tutorials/dht11-humidity-temperature-sensor-with-arduino-code-on-esp32-board> (дата звернення: 01.05.2025).

11. How to Build Your Own ESP32 Temperature Monitor [Електронний ресурс]. – URL: <https://medium.com/initial-state/how-to-build-your-own-esp32-temperature-monitor-6967b797b913> (дата звернення: 02.05.2025).

12. ESP32 WiFi Weather Station With BME280 [Електронний ресурс]. – URL: <https://www.instructables.com/ESP32-WiFi-Weather-Station-With-a-BME280-Sensor/> (дата звернення: 03.05.2025).

13. ESP32 TMP117 Fridge Monitor Tutorial [Електронний ресурс]. – URL: <https://www.fathym.com/iot/docs/tutorials/esp32/esp32-tmp117-fridge-monitor> (дата звернення: 04.05.2025).

14. Weather Monitoring on ESP32 via Blynk [Електронний ресурс]. – URL: <https://blynk.io/blueprints/weather-monitoring-with-esp32-with-wifi-provisioning-ota> (дата звернення: 05.05.2025).

15. ESP Easy – Wikipedia [Електронний ресурс]. – URL: https://en.wikipedia.org/wiki/ESP_Easy (дата звернення: 06.05.2025).

ДОДАТКИ

ДОДАТОК А**Технічне завдання****Міністерство освіти та науки України****Вінницький національний технічний університет****Факультет інформаційних технологій та комп'ютерної інженерії****ЗАТВЕРДЖУЮ****Завідувач кафедри ОТ****проф., д.т.н.. Азаров О.Д.****21.03. 2025 р.****ТЕХНІЧНЕ ЗАВДАННЯ****на виконання бакалаврського кваліфікаційної роботи****“Мікропроцесорний засіб****для регулювання температури у приміщенні”****08-54.БКР.027.00.000 ТЗ****Керівник роботи к.т.н. доц. каф. ОТ****Городецька О. С.****Студент групи 2СП–216****Дяченко В. А.**

1. Підстава для виконання бакалаврської кваліфікаційної роботи (БКР)

1.1 Актуальність роботи зумовлена стрімким розвитком технологій енергоефективності та автоматизації, що потребує впровадження інтелектуальних комп'ютерних систем моніторингу температури. Така система дозволяє забезпечити комфортний мікроклімат у приміщеннях, знижуючи витрати на енергоресурси. Особливої актуальності тема набуває в контексті переходу до «розумних» будинків і цифрової інфраструктури.

1.2 Підстава виконання БКР — затвердження теми згідно з наказом по ВНТУ (№ та дата відповідно до індивідуального завдання).

2. Мета БКР і призначення розробки

2.1 Мета роботи — розширення функціональних можливостей комп'ютерної системи для регулювання температури у приміщенні за рахунок додавання комбінованого цифрового датчика, що дозволяє точніше оцінювати мікрокліматичні умови з вищою точністю.

2.2 Призначення розробки — створення доступного мікропроцесорного засобу на базі ESP32 для відстеження кліматичних параметрів у приміщенні, їхнього аналізу та виведення результатів на LCD-дисплей, а також можливість подальшого масштабування та інтеграції з системами «розумного дому».

3. Вихідні дані для виконання БКР

3.1 Призначення — автоматизоване підтримання комфортного мікроклімату в приміщенні;

3.2 Діапазон контрольованих температур — від 0°C до +50°C;

3.3 Тип виходу — дисплейний (візуальне відображення);

3.4 Комутована напруга — ~220 В;

3.5 Комутований струм — не менше 1 А;

3.6 Живлення — зовнішнє джерело постійної напруги 5 В;

3.7 Тип датчика — цифровий датчик температури та вологості DHT11 v2.

4. Вимоги до виконання БКР

4.1 Провести аналіз сучасних систем моніторингу температури, включаючи IoT-рішення;

4.2 Обґрунтувати вибір апаратної платформи та датчиків;

4.3 Розробити структурну та функціональну схеми системи;

4.4 Реалізувати програмне забезпечення для зчитування, обробки та візуалізації даних з датчика;

4.5 Забезпечити точність і стабільність роботи системи у реальному часі;

4.6 Підготувати графічні матеріали: структурна схема, блок-схема алгоритму, схема підключення компонентів, лістинг коду.

5. Етапи БКР та очікувані результати

Етапи виконання кваліфікаційної роботи наведено у таблиці А.1 пояснювальної записки.

Таблиця А.1 — Етапи БКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасних систем моніторингу температури в приміщенні	15.04.2025	19.04.2025	Вступ, Розділ 1
2	Аналіз методів та засобів розробки комп'ютерної системи моніторингу	20.04.2025	24.04.2025	Розділ 2
3	Проектування та реалізація комп'ютерної системи моніторингу температури	25.04.2025	29.04.2025	Розділ 3
4	Розробка програми для мікроконтролера	30.04.2025	06.05.2025	Розділ 3
5	Оформлення пояснювальної записки, графічного матеріалу і презентації	06.05.2025	09.05.2025	ПЗ, графічний матеріал і презентація
6	Підготовка і підпис супроводжуючих документів, нормоконтроль та тест на плагіат	10.05.2025	13.05.2025	Оформленні документи

6. Матеріали, що подаються до захисту БКР

До захисту подаються:

- пояснювальна записка БКР;
- Перелік графічного матеріалу: структурна схема системи, Лістинг коду програми, функціональна схема системи, розпіновка ESP32 nano, блок-схема алгоритму обробки даних.
- протокол попереднього захисту БКР на кафедрі;
- відгук наукового керівника;
- відгук рецензента;
- анотації до БКР українською та англійською мовами.

7. Порядок контролю виконання та захисту БКР

Виконання етапів графічної та розрахункової документації БКР контролюється науковим керівником згідно зі встановленими термінами, передбаченими календарним планом.

Захист БКР проводиться на засіданні Екзаменаційної комісії, затвердженої наказом ректора університету.

8. Вимоги до оформлення та порядок виконання БКР

8.1 При оформленні БКР необхідно використовувати такі нормативні документи:

- ДСТУ 3008:2015 «Звіти у сфері науки і техніки. Структура та правила оформлення»;
- ДСТУ 8302:2015 «Бібліографічні посилання. Загальні положення та правила складання»;
- ГОСТ 2.104–2006 «Єдина система конструкторської документації. Основні написи»;
- методичні вказівки кафедри до виконання кваліфікаційних робіт за

спеціальністю 123 — Комп'ютерна інженерія;

— інші нормативні документи, на які є посилання в зазначених вище джерелах.

8.2 Порядок виконання БКР регламентується «Положенням про кваліфікаційні роботи на першому (бакалаврському) рівні вищої освіти» (СУЯ ВНТУ-03.02.02-П.001.01:21).

Графічна частина

КОМП'ЮТЕРНА СИСТЕМА МОНІТОРИНГУ ТЕМПЕРАТУРИ В
ПРИМІЩЕННІ

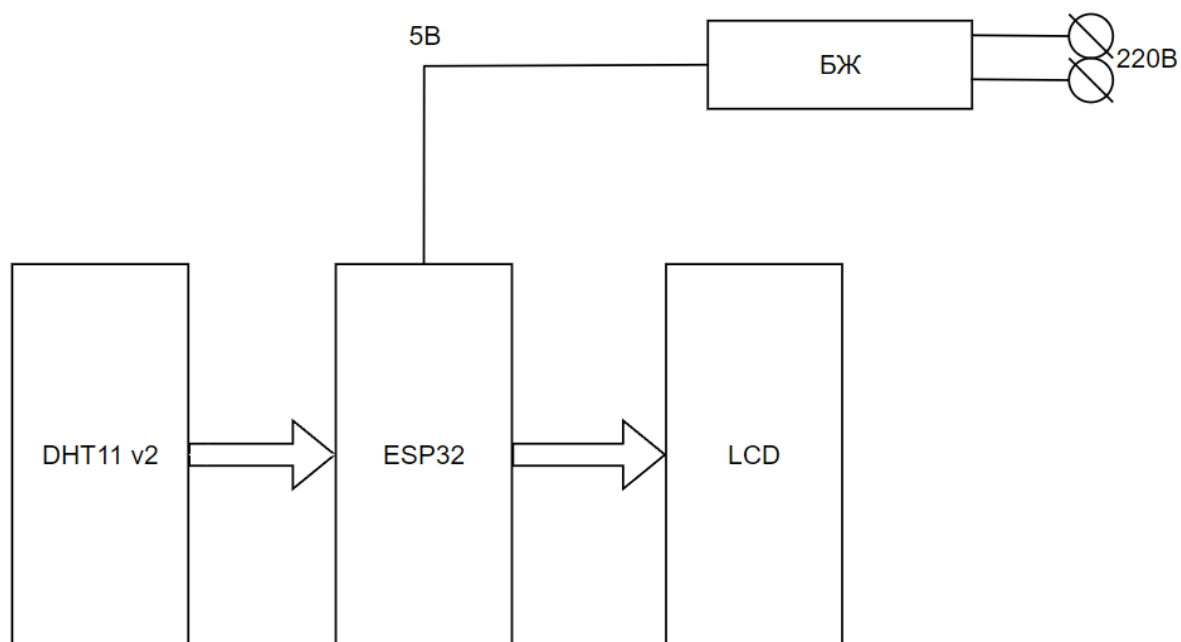


Рисунок В.1 — Структурна схема системи

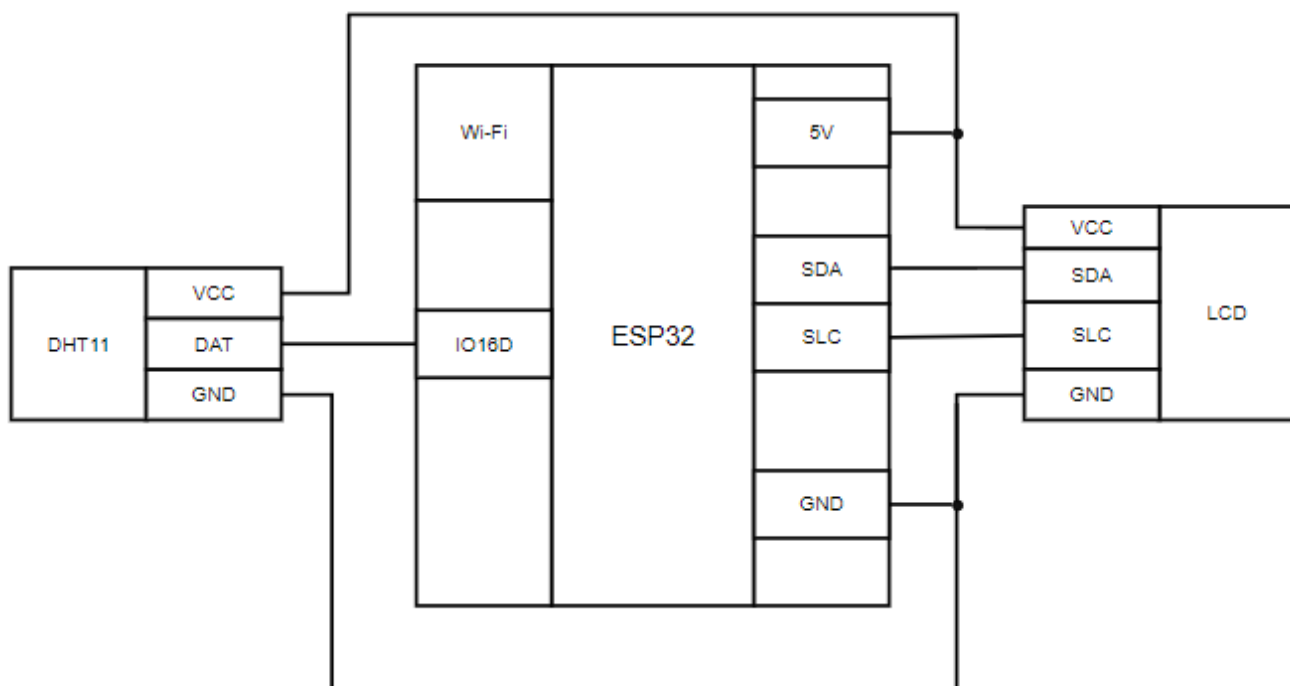


Рисунок В.2 — Функціональна схема системи

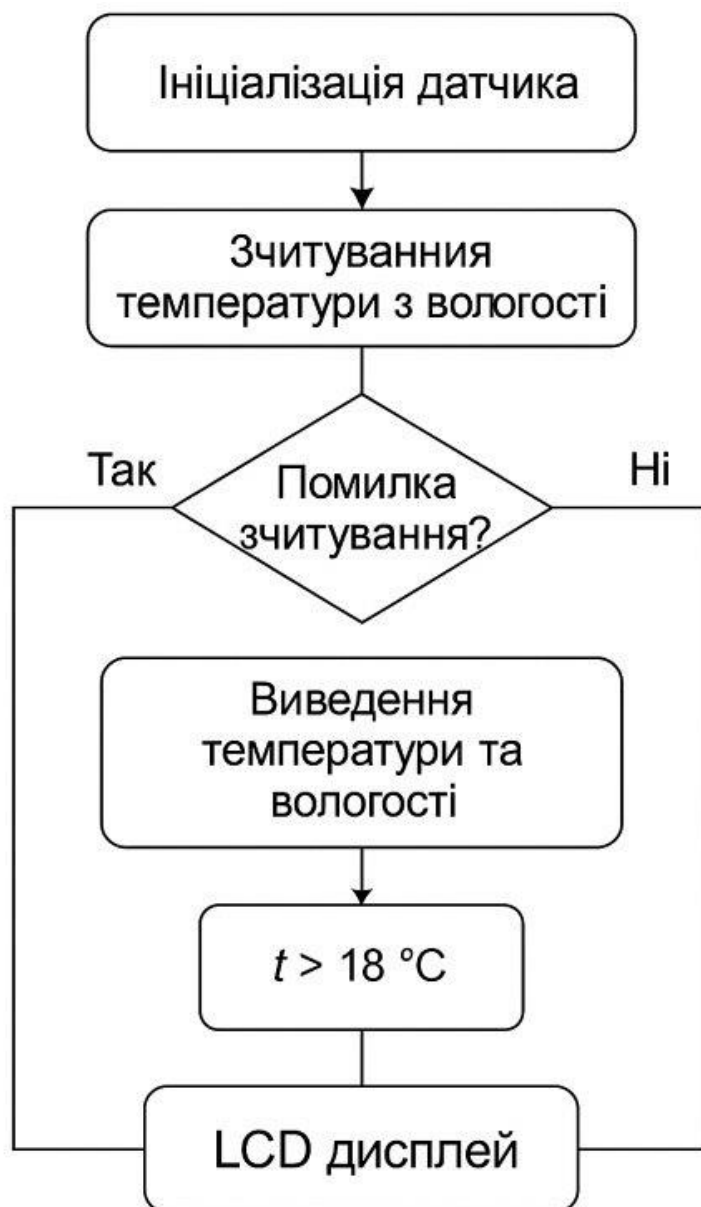


Рисунок В.3 — Блок-схема алгоритму обробки даних

ДОДАТОК Г
Ілюстративна частина

**КОМП'ЮТЕРНА СИСТЕМА МОНІТОРИНГУ ТЕМПЕРАТУРИ В
ПРИМІЩЕННІ**

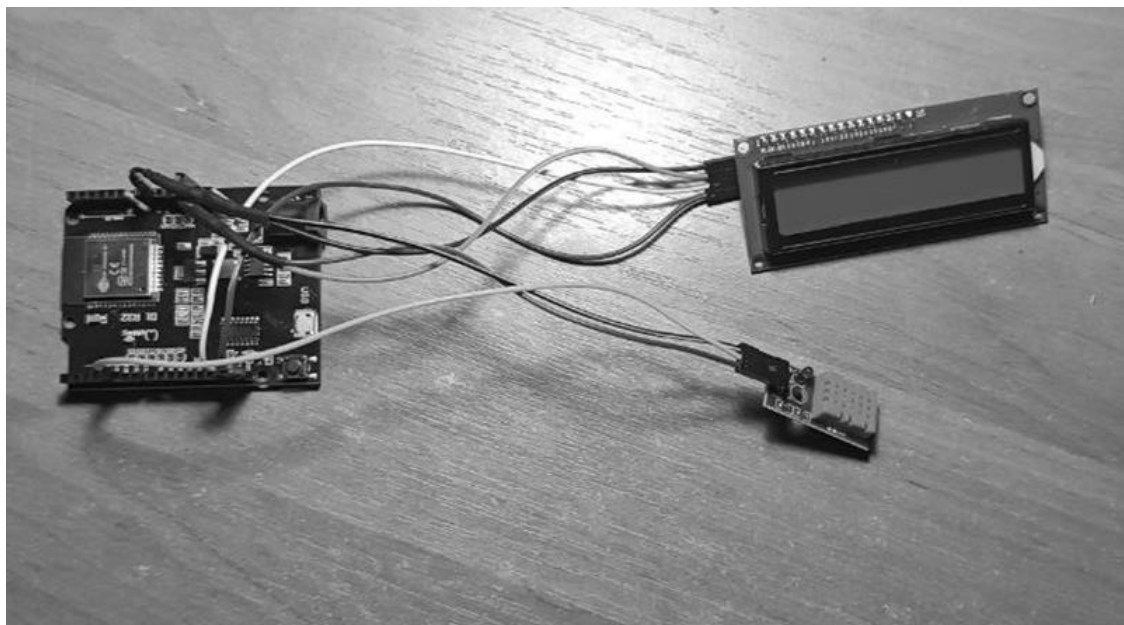


Рисунок Г.1 — Прототип комп'ютерної системи

ДОДАТОК Д

Лістинг програмного забезпечення

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN 16
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  dht.begin();
  lcd.setCursor(0, 0);
  lcd.print("Temp & Humidity");
  delay(2000);
  lcd.clear();
}
void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("Помилка зчитування DHT!");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("DHT Error");
    delay(2000);
    return;
  }
  Serial.print("Температура: ");
  Serial.print(t);
  Serial.print(" °C | Вологість: ");
  Serial.print(h);
  Serial.println(" %");
  lcd.setCursor(0, 0);
  lcd.print("T:");
  lcd.print(t);
  lcd.print((char)223);
  lcd.print("C");
  lcd.setCursor(0, 1);

```

```
lcd.print("H:");  
lcd.print(h);  
lcd.print("%");  
delay(2000);  
}
```