

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра комп'ютерної інженерії

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувачка кафедри,
д-р техн. наук, проф.

_____Ірина ЖУРАВСЬКА

«__» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
Пристрій для вимірювання температури повітря
складського приміщення та контролю умов для
зберігання продукції на базі Arduino

Спеціальність 123 Комп'ютерна інженерія

Освітня програма «Комп'ютерна інженерія»

Здобувач

_____Микола Барда
підпис

«__» _____ 2025 р.

Керівник доцент каф. КІ

_____Ярослав Крайник
підпис

«__» _____ 2025 р.

Миколаїв – 2025

Факультет	Комп'ютерних наук
Кафедра	Комп'ютерної інженерії
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступень	Бакалавр
Спеціальність	123 Комп'ютерна інженерія
Освітня програма	Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерної інженерії
_____ Ірина ЖУРАВСЬКА
« ____ » _____ 2025 р.

ЗАВДАННЯ на кваліфікаційну роботу здобувача

_____ Барді Миколі Олександровичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи

Пристрій для вимірювання температури повітря складського приміщення та контролю умов для зберігання продукції на базі Arduino.

Затверджена наказом по ЧНУ ім. Петра Могили від 30.10.2024 № 297.

2. Строк представлення кваліфікаційної роботи « ____ » _____ 2025 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Розробка повноцінного прототипу пристрою на базі Arduino Pro Mini для вимірювання температури в складському приміщенні. Система повинна забезпечити безперервний моніторинг температури, виведення даних на OLED-дисплей та GSM-оповіщення у разі перевищення критичних значень. Очікується також реалізація можливості запиту актуальних показників через вхідний дзвінок. Початкові дані включають вимоги до температурного режиму зберігання продукції, технічні характеристики компонентів системи та дані з тестових випробувань.

4. Перелік питань, що підлягають розробці:

- Аналіз технічних рішень у сфері моніторингу умов зберігання;
- Вибір апаратної платформи та сенсорів;
- Розробка архітектури апаратно-програмного комплексу;
- Реалізація програмного забезпечення для збору та обробки температурних даних;

- Тестування системи в реальних умовах;
- Оцінка енергоефективності та автономності роботи пристрою.

5. Перелік графічних матеріалів

Блок-схема логіки роботи системи, схема підключення компонентів пристрою, структурна схема системи в корпусі, скріншоти інтерфейсу користувача (OLED-дисплей, SMS-оповіщення), діаграми з результатами тестування, фото зібраного пристрою.

6. Консультанти:

Консультант	Кафедра (організація)	Частина роботи

Керівник роботи

Особистий підпис

Ярослав Крайник
Власне ім'я ПРИЗВИЩЕ

Здобувач



Особистий підпис

Микола Барда
Власне ім'я ПРИЗВИЩЕ

Дата видачі завдання «_____» _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної бакалаврської роботи

Тема: Пристрій для вимірювання температури повітря складського приміщення та контролю умов для зберігання продукції на базі Arduino.

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на виконання КБР	01.05.2025	1.05.2025	Виконано
2	Огляд літератури за темою роботи	02.05.2025	03.05.2025	Виконано
3	Складання календарного плану КБР	03.05.2025	03.05.2025	Виконано
4	Аналіз предметної області	04.05.2025	05.05.2025	Виконано
5	Розробка проєктних рішень	05.05.2025	06.06.2025	Виконано
6	Моделювання та конструювання АПЗ	06.06.2025	07.05.2025	Виконано
7	Перевірка працездатності, тестування та апробація розробленого АПЗ, аналіз результатів тестування	08.05.2025	10.05.2025	Виконано
8	Відгук керівника КБР	10.05.2025	12.05.2025	Виконано
9	Оформлення КБР та презентації	15.05.2025	20.05.2025	Виконано
10	Попередній захист	21.05.2025	4.06.2025	Виконано
11	Рецензування	10.06.2025	11.06.2025	Виконано
12	Завершення оформлення КБР та презентації	12.06.2025	13.06.2025	Виконано
13	Захист кваліфікаційної бакалаврської роботи	24.06.2025	25.06.2025	

Керівник роботи

Особистий підпис

_____ Ярослав Крайник

Власне ім'я ПРІЗВИЩЕ

Здобувач



Особистий підпис

_____ Микола Барда

Власне ім'я ПРІЗВИЩЕ

« ____ » _____ 2025 р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи

«Пристрій для вимірювання температури повітря складського приміщення та контролю умов для зберігання продукції на базі Arduino»

Студент 405 гр.: Барда Микола Олександрович

Керівник: к.т.н., доцент каф. КІ Ярослав Михайлович Крайник

Кваліфікаційна бакалаврська робота присвячена розробці автономного пристрою для вимірювання температури повітря у складському приміщенні та контролю умов зберігання продукції на базі Arduino Pro Mini.

Актуальність теми зумовлена зростаючою потребою в автоматизованому моніторингу мікроклімату в умовах логістичних центрів, агропромислових підприємств, фармацевтичних складів тощо. Незабезпечення належного температурного режиму призводить до втрат продукції та фінансових збитків, що обґрунтовує необхідність використання доступних та енергоефективних рішень для моніторингу.

Об'єкт дослідження – процес збереження продукції в складських приміщеннях.

Предмет дослідження – апаратно-програмний комплекс для моніторингу температури повітря з використанням мікроконтролерів Arduino.

Мета роботи – створити недорогий, автономний та ефективний пристрій на базі Arduino Pro Mini, здатний в реальному часі вимірювати температуру, передавати інформацію на дисплей і через GSM-канал.

Практична значимість полягає в застосуванні пристрою на невеликих складах, у фермерських господарствах, харчовій та фармацевтичній логістиці, а також у навчальному процесі.

Апробація роботи відбулася в межах зустрічі FOSS з представниками Харківського національного економічного університету імені Семена Кузнеця. Тези виступу були схвалені керівником роботи та надані у додатку Б.

Основні результати:

- у розділі 1 проведено аналіз технічних рішень у сфері моніторингу умов зберігання продукції, здійснено порівняння існуючих рішень на основі Arduino та промислових систем;
- у розділі 2 обґрунтовано вибір апаратної бази, описано структуру, логіку та компоненти пристрою: DS18B20, SIM800L, TP4056, MT3608 тощо;
- у розділі 3 реалізовано програмну логіку, описано архітектуру коду, тестування в реальних умовах та оцінено енергоефективність системи.

Система протестована на стабільність, точність зчитування даних та здатність до автономної роботи. Реалізовано також можливість отримання температурних даних за допомогою вхідного дзвінка, що суттєво підвищує зручність використання.

Робота містить 75 сторінки (разом з додатками), 8 таблиць, 27 рисунків, 18 джерел посилання, 2 додатки.

Ключові слова: моніторинг температури, Arduino, GSM, складське зберігання, DS18B20, автономне живлення, енергоефективність.

ABSTRACT

of the Bachelor's Thesis

"Device for Measuring Air Temperature in a Warehouse and Monitoring Storage Conditions Based on Arduino"

Student 405 group: Barda Mykola

Supervisor: Candidate of Technical Sciences (Ph. D.), Assoc. Prof. Yaroslav Kraynyk

The bachelor's thesis is devoted to the development of an autonomous device for measuring air temperature in a warehouse and monitoring storage conditions based on the Arduino Pro Mini microcontroller.

Relevance of the topic is due to the growing need for automated microclimate monitoring in logistics centers, agro-industrial enterprises, pharmaceutical warehouses, and other storage facilities. Failure to maintain the appropriate temperature regime leads to product spoilage and financial losses, which justifies the need for accessible and energy-efficient monitoring solutions.

Object of the study – the process of preserving products in warehouse environments.

Subject of the study – a hardware-software complex for monitoring air temperature using Arduino microcontrollers.

The aim of the work is to create a low-cost, autonomous, and effective device based on Arduino Pro Mini that can measure temperature in real time and transmit the data to a display and via a GSM module.

Practical significance lies in the application of the device in small warehouses, farming facilities, the food and pharmaceutical logistics sectors, and in the educational process.

The research was presented at a FOSS meeting with representatives of Simon Kuznets Kharkiv National University of Economics. The presentation thesis was approved by the project supervisor and included in Appendix B.

Main results:

- chapter 1 analyzes technical solutions in the field of storage condition monitoring and compares existing Arduino-based and industrial systems;
- chapter 2 justifies the hardware selection and describes the structure, logic, and components of the device: DS18B20, SIM800L, TP4056, MT3608, etc;
- chapter 3 describes the implementation of the software logic, code architecture, real-world testing, and evaluation of the system's energy efficiency.

The system was tested for stability, measurement accuracy, and autonomous operation. The implementation of temperature data retrieval via an incoming call significantly increases usability.

The thesis consists of 73 pages (with appendices), 8 tables, 27 figures, 18 references, and 2 appendices.

Keywords: *temperature monitoring, Arduino, GSM, warehouse storage, DS18B20, autonomous power supply, energy efficiency.*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП	5
1 АНАЛІЗ ТЕХНІЧНИХ РІШЕНЬ ДЛЯ МОНІТОРИНГУ УМОВ ЗБЕРІГАННЯ ПРОДУКЦІЇ	7
1.1 Актуальність моніторингу умов зберігання.....	7
1.2 Існуючі технічні рішення	8
1.3 Критерії вибору систем моніторингу, технології зв'язку та хмарні рішення.....	19
2 ПРОЄКТУВАННЯ СИСТЕМИ, ПРИСТРОЯ ДЛЯ ВИМІРЮВАННЯ ТЕМПЕРАТУРИ ПОВІТРЯ СКЛАДСЬКОГО ПРИМІЩЕННЯ ТА КОНТРОЛЮ УМОВ ДЛЯ ЗБЕРІГАННЯ ПРОДУКЦІЇ НА БАЗІ ARDUINO	24
2.1 Методологія комп'ютерного моделювання та підхід до керування..	24
2.2 Покрокове проєктування алгоритму	26
2.3 Деталізований алгоритм роботи	34
3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ	38
3.1 Архітектура апаратно-програмного забезпечення	38
3.2 Технології та мови програмування	41
3.3 Компоненти АПЗ.....	43
3.4 Інтерфейс користувача	44
3.5 Основні компоненти коду	46
3.6 Методика тестування.....	49
3.7 Результати експериментів	52
3.8 Керівництво користувача	53
ВИСНОВКИ.....	58
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	59
ДОДАТОК А Код програми.....	61

ДОДАТОК Б Матеріали апробації роботи	63
--	----

ПЕРЕЛІК СКОРОЧЕНЬ

АПЗ	– апаратно-програмне забезпечення
КБР	– кваліфікаційна бакалаврська робота
AC	– Alternating Current
API	– Application Programming Interface
DC	– Direct Current
ESP	– Espressif Systems Platform
GSM	– Global System for Mobile Communications
HTTP	– HyperText Transfer Protocol
HTTPS	– HyperText Transfer Protocol Secure
IAE	– Integral of Absolute Error
IoT	– Internet of Things
LAN	– Local Area Network
LCD	– Liquid Crystal Display
LoRa	– Long Range
MATLAB	– Matrix Laboratory
MQTT	– Message Queue Telemetry Transport
PID	– Proportional–Integral–Derivative controller
SIM	– Subscriber Identity Module
USB	– Universal Serial Bus
Wi-Fi	– Wireless Fidelity

ВСТУП

У сучасному світі питання збереження якості продукції під час транспортування та зберігання набуває особливої актуальності. Це стосується як харчових продуктів, так і медикаментів, хімічних речовин, електроніки, текстилю тощо. Одним із найважливіших факторів, що впливають на якість зберігання, є температура повітря в складському приміщенні. Нерегульовані або неконтрольовані температурні умови можуть призвести до псування продукції, порушення санітарних норм, а в деяких випадках – до суттєвих економічних втрат. У зв'язку з цим виникає об'єктивна потреба у розробці ефективних систем моніторингу мікроклімату, які дозволяють відстежувати зміну температури та оперативно реагувати на відхилення від заданих параметрів.

Завдяки розвитку цифрових технологій, зокрема у сфері мікроконтролерів і сенсорної електроніки, з'явилася можливість створювати недорогі, надійні та зручні у використанні пристрої для контролю умов зберігання. Однією з найбільш доступних платформ для реалізації таких рішень є Arduino – апаратно-програмне середовище з відкритим кодом [1], яке дає змогу швидко створювати прототипи електронних пристроїв.

Об'єктом дослідження є процес забезпечення оптимальних умов зберігання продукції шляхом контролю мікроклімату складських приміщень.

Предметом дослідження є технічні засоби та методи моніторингу температури повітря в складських умовах із використанням мікроконтролерів Arduino та відповідних сенсорів.

Мета роботи полягає у розробці та тестуванні прототипу пристрою на базі Arduino для безперервного вимірювання температури повітря у складському приміщенні з метою підвищення ефективності зберігання продукції та запобігання її псуванню внаслідок несприятливих температурних умов.

Основними завданнями проєкту є:

- обґрунтування актуальності теми роботи;
- аналіз технічних рішень для моніторингу умов зберігання продукції;
- огляд існуючих пристроїв і технічних рішень;
- аналіз аналогічних розробок;
- розробка нового технічного рішення з урахуванням чинників, пов'язаних із темою роботи.

Практичне значення пристрою для вимірювання температури повітря у складському приміщенні полягає у його широкому застосуванні в галузях, де важливо контролювати температурний режим. Система на базі Arduino є доступною за вартістю, простою у впровадженні та може використовуватись:

- на малих і середніх складах для моніторингу температури без потреби в дорогому обладнанні;
- в аграрному секторі: для контролю умов зберігання овочів, фруктів, зерна;
- у харчовій промисловості: для забезпечення відповідності санітарним нормам при зберіганні продуктів харчування;
- на фармацевтичних складах: для дотримання температурного режиму зберігання медикаментів;
- у навчальних закладах: як приклад практичного застосування знань з мікроелектроніки, автоматизації та програмування;
- у невеликих підприємствах електроніки та текстилю: для базового контролю кліматичних умов зберігання сировини або продукції.

Таким чином, запропоноване рішення може слугувати як повноцінним інструментом для практичного застосування, так і навчальним прикладом у технічній освіті.

Апробація розробки відбувалася під час зустрічі FOSS з представниками Харківського національного економічного університету імені Семена Кузнеця шляхом подання тез за тематикою КБР [14].

1 АНАЛІЗ ТЕХНІЧНИХ РІШЕНЬ ДЛЯ МОНІТОРИНГУ УМОВ ЗБЕРІГАННЯ ПРОДУКЦІЇ

1.1 Актуальність моніторингу умов зберігання

Проблема забезпечення оптимальних умов зберігання продукції на складах є надзвичайно актуальною як для промислових підприємств, так і для малого та середнього бізнесу. Належна організація складських приміщень, зокрема контроль температури і вологості, є обов'язковою умовою для забезпечення якості товарів, дотримання нормативних вимог і збереження інвестиційних коштів фізичних осіб.

Особливо чутливими до температурного режиму є продукти харчування, фармацевтичні препарати, косметика, електроніка та хімічні матеріали. Для прикладу, при зберіганні молочних продуктів необхідна температура від +2 °C до +6 °C. Відхилення від цього діапазону навіть на кілька градусів може призвести до розвитку мікроорганізмів, зменшення строку придатності або повної втрати товарної якості. Те саме стосується вакцин, які потребують особливих умов зберігання в рамках холодового ланцюга.

Окрім загроз безпосередньому стану продукції, порушення умов зберігання може призвести до фінансових збитків, логістичних ускладнень, втрати репутації та навіть до адміністративної відповідальності, якщо мова йде про порушення санітарних норм. У зв'язку з цим все більше компаній впроваджують системи моніторингу мікроклімату [16] в складських приміщеннях.

Сучасні технології дають змогу автоматизувати процес контролю, уникнути людського фактору та вчасно отримувати сповіщення про критичні зміни в мікрокліматі. Для цього використовуються як комерційні, так і саморобні системи з мікроконтролерами, які вимірюють температуру, вологість, рівень освітлення, концентрацію газів та інші параметри середовища. Завдяки доступності апаратних платформ, таких як Arduino,

Raspberry Pi та ESP, стало можливим створювати компактні та недорогі системи моніторингу навіть у побутових або навчальних умовах.

Отже, впровадження систем моніторингу умов зберігання є необхідним кроком для забезпечення стабільності логістичних процесів, підвищення якості обслуговування клієнтів та оптимізації витрат.

1.2 Існуючі технічні рішення

Станом на сьогодні існує широкий спектр готових рішень для моніторингу температури та вологості в приміщеннях, які відрізняються за складністю, ціною, функціональністю та точністю. Найбільш популярними є промислові системи на базі контролерів з підтримкою передачі даних через інтернет (IoT-рішення) [15], а також більш прості та доступні проекти на основі мікроконтролерів, як-от Arduino.

У дослідженні, опублікованому на порталі Neliti [13], розглядається система моніторингу температури в умовах тваринництва – зокрема в козлятнику. В основі системи – мікроконтролер Arduino Uno, датчик DHT22 [1] для зчитування температури та вологості, а також модуль ESP8266 [2], який передає дані на хмарний сервер або смартфон. Це рішення дозволяє власнику господарства постійно отримувати актуальну інформацію про мікроклімат у приміщенні, навіть перебуваючи на відстані. Система побудована з акцентом на автономність та низьке енергоспоживання.

Інше дослідження, представлене на платформі ResearchGate [7], описує систему для контролю умов на складі. В її основі також Arduino-платформа, але з використанням сенсора DHT11, який є дешевшим, хоча й менш точним [9] у порівнянні з DHT22. Інформація з сенсора відображається на LCD-дисплеї у реальному часі, без можливості віддаленого моніторингу. Це просте, але дієве рішення, що може бути використане для малих складів або підсобних приміщень [7].

Крім того, існують і комерційні рішення, такі як системи Testo Saveris, Elitech, Tempmate, які забезпечують точне вимірювання, архівування даних, GPS-локалізацію та можливість інтеграції з ERP-системами підприємства. Однак такі пристрої є досить дорогими і не завжди виправдані для малого бізнесу чи побутового використання.

1.2.1 Аналіз аналогічних пристроїв, їх переваг та недоліків.

Для дослідження було обрано два пристрої ECOFAN та автоматична система моніторингу температури з керуванням вентилятором на базі Arduino [7].

Запропонований прилад для контролю температури приміщення побудовано на базі мікроконтролера Arduino Uno. Його основна мета – вимірювати температуру повітря, відображати її на екрані, а також надсилати повідомлення у випадку перевищення встановленої межі.

Блок-схему логіки представлено на рис. 1.1.

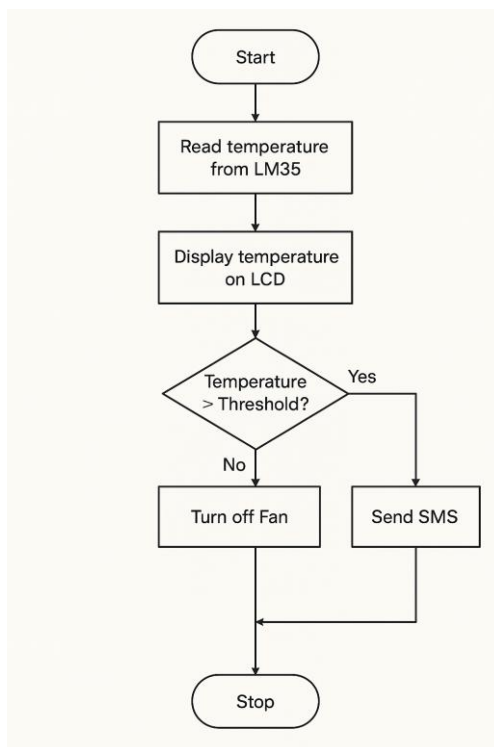


Рисунок 1.1 – Блок-схема логіки пристрою ECOFAN

Основні компоненти системи:

- Arduino Uno: головний контролер, який керує усіма підключеними модулями.
- Датчик температури LM35: вимірює температуру з високою точністю і передає ці дані на Arduino.
- LCD-дисплей 16×2 : виводить температуру та інші повідомлення для користувача.
- Вентилятор 12 В: автоматично вмикається, якщо температура перевищує норму.
- GSM-модуль: надсилає SMS-повідомлення на телефон у разі перегріву.
- Світлодіод: сигналізує про увімкнення або перегрів.
- Транзистор 2N2222 і діод 1N4007: для керування вентилятором.
- Резистори (1К Ом), конденсатор (10 мкФ) : для стабільної роботи схеми.
- Макетна плата: збірка всіх компонентів без пайки.
- Кабелі-перемички (тато-тато, тато-мама): для з'єднання деталей на макетці.
- Живлення: батарея 12 В або блок живлення через USB.

Усі перелічені компоненти формують функціональну систему автоматизованого контролю температури повітря в приміщенні. Центральним елементом є мікроконтролер Arduino Uno, який обробляє дані з температурного датчика LM35 та приймає рішення про активацію вентилятора або надсилання SMS-повідомлень через GSM-модуль у разі перевищення критичних температурних значень. Інформація про температуру виводиться на LCD-дисплей, а світлодіод подає візуальні сигнали про стан системи. Уся схема реалізована на макетній платі, що дозволяє швидко протестувати і змінювати конфігурацію без потреби в пайці. Такий підхід робить систему

зручною для прототипування, адаптації до різних умов експлуатації та подальшого вдосконалення.

Схему підключення показано на рис. 1.2 [7].

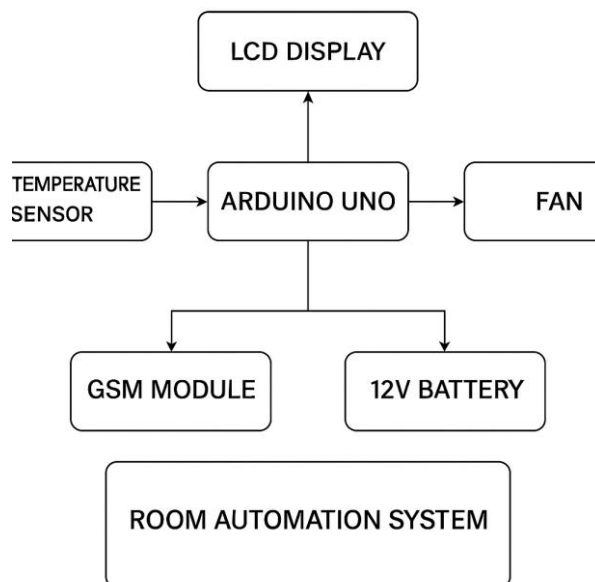


Рисунок 1.2 – Схема підключення пристрою ECOFAN

Програмне забезпечення:

Програма для керування пристроєм написана на мові C++ у середовищі Arduino IDE. Це безкоштовне середовище розробки, яке дозволяє швидко створювати та завантажувати програми на плату Arduino. Воно підтримується на Windows, Mac і Linux.

Принцип роботи системи:

1. LM35 постійно вимірює температуру;
2. значення температури виводяться на LCD-дисплей;
3. якщо температура перевищує заданий поріг:
 - увімкнеться вентилятор;
 - засвітиться світлодіод;
 - через GSM-модуль надійде SMS-повідомлення користувачу.

Користувач може бачити температуру в реальному часі та оперативно реагувати на зміни.

Прилад ECOFAN показано на рис. 1.3 [7].

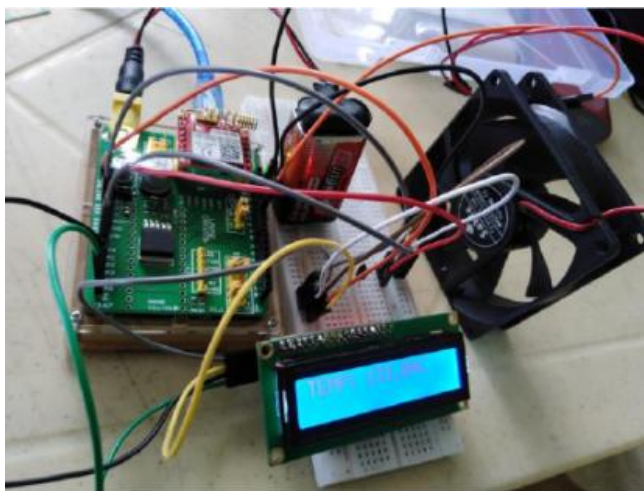


Рисунок 1.3 – Прилад ECOFAN у зборі

Преваги та недоліки описаного пристрою ECOFAN надано у таблиці 1.1.

Таблиця 1.1 – Переваги та недоліки пристрою ECOFAN.

Преваги	Недоліки
Автоматичне реагування на зміну температури: пристрій самостійно вмикає вентилятор та надсилає сповіщення при перегріві, що підвищує безпеку зберігання продукції або комфорт у приміщенні.	Відсутність резервного зберігання даних: пристрій не має флеш-пам'яті або SD-карти для збереження історії вимірювань.
Низька вартість реалізації: усі компоненти (Arduino Uno, LM35, дисплей тощо) – доступні й недорогі, що робить систему економічно вигідною для широкого використання.	Залежність від живлення: пристрій може працювати нестабільно при коливаннях напруги або розряді батареї (особливо при використанні 12В живлення).
Простота збірки та налаштування: схема не потребує пайки (збирається на макетній платі),	Одноканальна система контролю: ECOFAN вимірює лише температуру одного об'єкта/зони.

а програмування здійснюється у зручному середовищі Arduino IDE.	Для великих приміщень потрібно кілька таких приладів або розширення системи.
Можливість віддаленого моніторингу: завдяки GSM-модулю користувач може отримувати SMS-повідомлення й контролювати ситуацію на відстані.	Затримка в GSM-зв'язку: надсилення SMS може мати невелику затримку залежно від покриття мобільної мережі.
Модульність і розширюваність: до системи можна легко додати інші датчики (вологості, освітлення, CO ₂ тощо) або виконавчі елементи.	Низький рівень безпеки зв'язку: GSM-повідомлення не шифруються, тому система не підходить для критично важливих об'єктів, де потрібен захист даних.
Захист електронних компонентів: діод та транзистор забезпечують надійне керування вентилятором та захист від перенапруги.	

Пристрій **ECOFAN** є ефективним рішенням для моніторингу та автоматичного регулювання температури в приміщенні. Завдяки простій конструкції, низькій вартості та можливості віддаленого сповіщення, система підходить для побутового та напівпромислового використання. Попри деякі обмеження, такі як залежність від живлення та відсутність збереження даних, ECOFAN демонструє надійну та стабільну роботу при виконанні основних функцій кліматичного контролю.

Наступним для дослідження було обрано **пристрій: Arduino-Based PID Контролер Температури [8]**.

Запропонований пристрій – термостатична система на базі Arduino Uno, розроблена для дослідження та реалізації різних методів керування

температурою (ON/OFF, гістерезис, PID). Його головна мета – підтримувати задану температуру за допомогою нагрівального елемента та забезпечити ефективне регулювання за критерієм ІАЕ.

Блок-схему логіки представлено на рис. 1.4 [8].

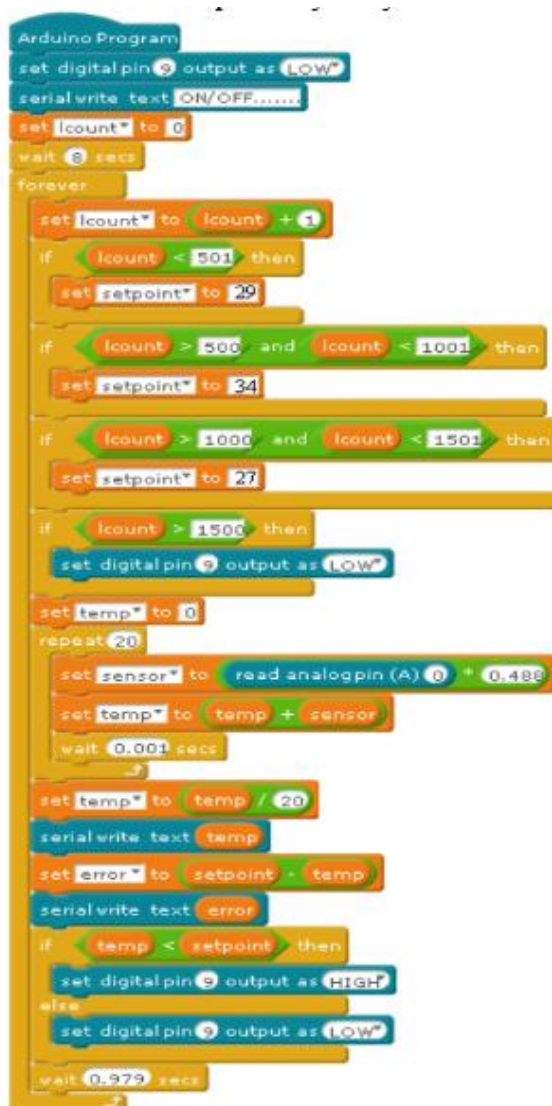


Рисунок 1.4 – Блок-схема логіки приладу Arduino-Based PID Контролер
Температури

Основні компоненти системи:

– Arduino Uno: головний контролер, що реалізує алгоритм керування (PID, ON/OFF, гістерезис).

- Температурний датчик LM35DZ: аналоговий сенсор з вихідною напругою $10 \text{ мВ}/^{\circ}\text{C}$, підключений до A0.
- Нагрівальний резистор: виконавчий елемент, керований через транзистор.
- Оптоізолятор 4N25: для гальванічної розв'язки сигналу керування.
- Транзистор BD137: комує живлення до резистора.
- LCD 2x16: відображення поточної температури та заданого значення.
- Серійний монітор Arduino IDE: для виводу графіків температури та помилок у реальному часі.
- Живлення: трансформатор з 230V у 9V AC + стабілізатор 7808 – 8V DC.

Ця система реалізує точне керування температурою за допомогою мікроконтролера Arduino Uno, який виконує заданий алгоритм регулювання, зокрема PID, ON/OFF або гістерезис. Температурний датчик LM35DZ, підключений до аналогового входу A0, забезпечує точне вимірювання температури, перетворюючи її у відповідну напругу. Для керування нагрівальним елементом використовується транзистор BD137 у зв'язці з оптоізолятором 4N25, що дозволяє забезпечити безпечну гальванічну розв'язку керуючого сигналу. Всі значення температури та задані параметри відображаються на LCD-дисплеї 2x16, а додатковий вивід у серійному моніторі Arduino IDE дає змогу спостерігати графіки в режимі реального часу. Живлення реалізоване через трансформатор з пониженням до 9V AC та стабілізатором 7808 для отримання стабільної напруги 8V DC, необхідної для роботи системи. Такий підхід дозволяє реалізувати надійну та адаптивну систему керування температурою з можливістю візуалізації та налаштування.

Схему підключень основних компонентів представлено на рис. 1.5 [8].

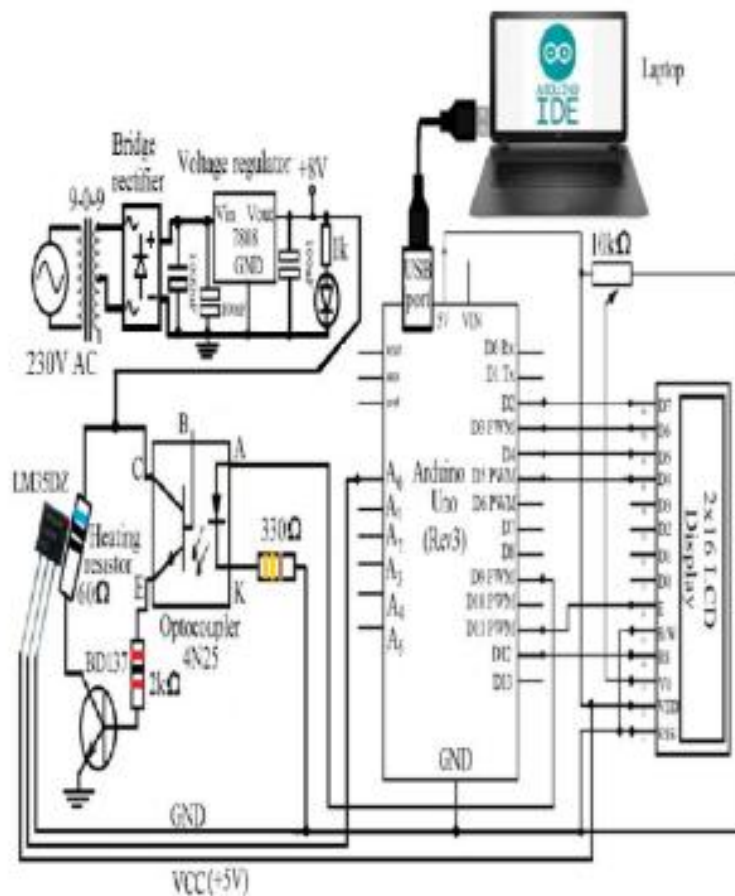


Рисунок 1.5 – Схема підключень приладу Arduino-Based PID Контролер Температури.

Програмне забезпечення:

- Arduino IDE: основне середовище розробки, що використовується для написання, компіляції та завантаження коду.
- Flowcode: графічне середовище на основі блок-схем.
- mBlock (Scratch-based): блокове програмування, зручне для візуального конструювання логіки.
- MATLAB Arduino I/O: для прямої взаємодії Arduino з MATLAB без компіляції.

Для програмування та налагодження систем на базі Arduino використовується низка програмних середовищ, кожне з яких має свої переваги. Arduino IDE є класичним інструментом, що дозволяє розробляти та

завантажувати код мовою C/C++ безпосередньо на мікроконтролер. Для користувачів, які надають перевагу візуальному програмуванню, Flowcode пропонує можливість створення логіки у вигляді блок-схем, що особливо зручно на етапі моделювання. mBlock, побудований на основі Scratch, забезпечує інтуїтивне блокове програмування, що ідеально підходить для новачків і швидкого створення навчальних проєктів. MATLAB Arduino I/O дозволяє взаємодіяти з Arduino в режимі реального часу без потреби компіляції, що відкриває широкі можливості для математичного моделювання та аналізу експериментальних даних. Сукупність цих інструментів робить розробку під Arduino гнучкою та доступною для різних рівнів підготовки.

Принцип роботи системи:

1. Зчитування температури з LM35;
2. обчислення похибки між реальною температурою та заданим значенням;
3. використання одного з трьох алгоритмів (ON/OFF, гістерезис або PID) для обчислення керуючого впливу;
4. зміна ШІМ-сигналу на пін D9 або вмикання/вимикання нагрівача (залежно від методу);
5. відображення даних на дисплеї або серійному моніторі.

За таким принципом працює прилад Arduino-Based PID Контролер Температури.

Експерементальна збірка приладу представлена на рис. 1.6 [8].

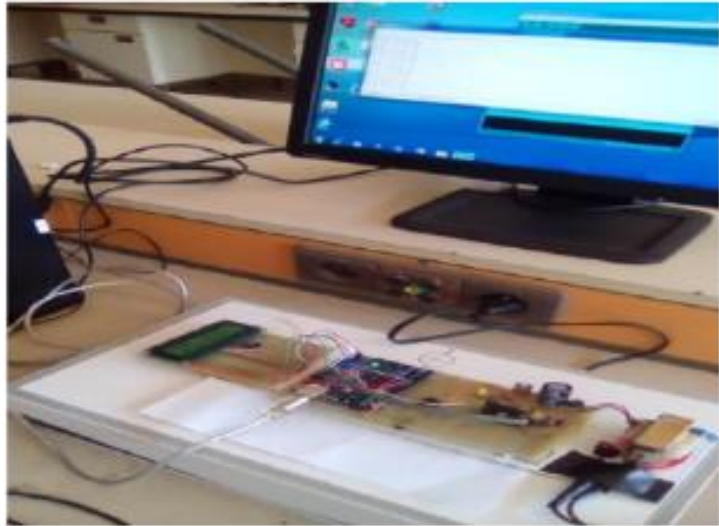


Рисунок 1.6 – Експериментальна збірка приладу

Преваги та недоліки описаного пристрою Arduino-Based PID Контролер Температури надано у таблиці 1.2.

Таблиця 1.2 – Переваги та недоліки пристрою Arduino-Based PID Контролер Температури.

Преваги	Недоліки
Підтримка кількох алгоритмів керування: ON/OFF, гістерезис, PID.	Відсутність GSM-зв'язку чи віддаленого сповіщення.
Можливість програмування в різних середовищах (IDE, Flowcode, mBlock, MATLAB).	Система не має збереження історії вимірювань.
Низька вартість компонентів.	Ручне налаштування PID-параметрів (K_p , K_i , K_d) потребує досвіду або методу Зіглера-Ніколса.
Висока точність з внутрішньою опорою АЦП 1.1В.	Підходить тільки для локального контролю температури.
Демонструє мінімальну похибку в режимі PID.	PID реалізація займає більше пам'яті (до 21% скетчу).

Система, розроблена на базі Arduino Uno, дозволяє реалізувати різні стратегії керування температурою з використанням простих та доступних компонентів. Особливо ефективною є реалізація PID-регулятора, що забезпечує мінімальну статичну похибку та кращу стабільність порівняно з ON/OFF і гістерезисними методами. Завдяки можливості програмування в різних середовищах, пристрій підходить як для навчальних, так і прикладних задач автоматизованого температурного контролю.

1.3 Критерії вибору систем моніторингу, технології зв'язку та хмарні рішення

1.3.1 Основні критерії вибору систем моніторингу

Вибір системи моніторингу мікроклімату залежить від багатьох факторів: особливостей об'єкта, вимог до точності, бюджету, масштабу приміщення, необхідності віддаленого контролю тощо. Щоб забезпечити ефективність контролю умов зберігання, слід враховувати комплекс технічних та експлуатаційних параметрів, які описано нижче.

1. Тип продукції: харчові товари, ліки, хімікати, електроніка мають різні вимоги до зберігання. Наприклад:

- Продукти харчування вимагають безперервного температурного контролю з точністю до $\pm 0,5$ °C.
- Фармацевтика потребує архівації даних та відповідності стандартам GxP.

2. Точність і стабільність вимірювань: вибір датчика (DHT11, DHT22, LM35, VME280 тощо) має залежати від допустимого відхилення температури/вологості. Чим вища точність – тим дорожчий сенсор.

3. Можливість віддаленого моніторингу: системи з GSM, Wi-Fi або LoRa дозволяють контролювати мікроклімат у режимі реального часу з будь-якої точки світу.

4. Масштабованість: якщо приміщення велике або має багато зон з різними мікрокліматичними умовами – важлива підтримка кількох датчиків або мережевого підключення.

5. Енергозалежність і резервування: автономні системи з батарейним живленням або резервними джерелами енергії є критично важливими у разі перебоїв з живленням.

6. Зручність інтеграції: сумісність з ERP/SCADA-системами або можливість API-підключення до існуючої цифрової інфраструктури підприємства.

Таким чином, для досягнення максимальної ефективності система має бути адаптована до специфіки об'єкта, а також забезпечувати гнучкість у налаштуванні та розширенні функціоналу.

1.3.2 Технології зв'язку в системах моніторингу

Сучасні пристрої можуть передавати дані за допомогою різних технологій. У таблиці 1.3 наведено порівняння найбільш поширених типів зв'язку.

Таблиця 1.3 – Порівняння найпоширеніших типів зв'язку.

Технологія	Радіус дії	Переваги	Недоліки
Wi-Fi	до 50 м	Висока швидкість, інтеграція з мережами	Високе енергоспоживання, обмежена дальність
LoRa	до 10 км	Дуже низьке споживання енергії, великий радіус	Низька швидкість передачі, потрібна LoRa-шлюз

Продовження таблиці 1.3.

Bluetooth	до 10 м	Простота реалізації, низьке енергоспоживання	Мала дальність, нестабільність
GSM/3G/4G/5G	глобально	Можливість роботи без Wi-Fi мереж	Витрати на мобільний зв'язок
Ethernet (LAN)	необмежено	Висока стабільність та швидкість	Потреба в прокладанні кабелю

Також важливо враховувати протоколи передачі даних, зокрема:

- HTTP/HTTPS – для інтеграції з веб-сервісами.
- MQTT – легкий протокол для IoT, ідеальний для передачі телеметрії з пристроїв.
- WebSocket – забезпечує двостороннє з'єднання для реального часу.

Розуміння переваг і обмежень кожної технології зв'язку дозволяє оптимально підібрати інфраструктуру для конкретних умов експлуатації системи моніторингу.

1.3.3 Хмарні сервіси та програмні рішення

Для візуалізації даних, зберігання історії вимірювань і віддаленого моніторингу часто використовують хмарні сервіси. Найпопулярніші з них:

1. ThingSpeak – відкритий сервіс для збору та аналізу даних з сенсорів. Дозволяє будувати графіки, налаштовувати тригери та відправляти email або SMS при критичних значеннях.
2. Blynk IoT – потужна мобільна платформа для керування пристроями на Arduino, ESP32 та ін. Можна створювати інтерфейси прямо зі смартфона.

3. Firebase (від Google) – платформа для зберігання даних у режимі реального часу. Підходить для розробки мобільних або web-додатків з функцією моніторингу.

4. Ubidots – хмарна платформа для промислових IoT, підтримує MQTT, HTTP API, дозволяє будувати дашборди, графіки, логіку тригерів.

5. Node-RED + InfluxDB + Grafana (локальне рішення) – надає можливість створити повністю автономний сервер моніторингу у локальній мережі з власним графічним інтерфейсом.

Застосування хмарних платформ суттєво спрощує обробку та інтерпретацію даних, забезпечуючи гнучкість доступу та масштабованість системи моніторингу.

Висновок до розділу 1

У розділі 1 було проведено детальний аналіз актуальності моніторингу умов зберігання продукції та існуючих технічних рішень для реалізації таких систем. Зокрема, розглянуто різноманітні підходи до контролю температури та вологості в складських приміщеннях, використовуючи як комерційні, так і саморобні рішення на базі мікроконтролерів.

Одним із прикладів є система моніторингу температури та вологості на базі Arduino Uno з використанням датчика DHT11 [7], яка дозволяє автоматично контролювати мікроклімат у складських приміщеннях та вчасно реагувати на відхилення від заданих параметрів. Інше рішення передбачає використання датчика DHT22 та модуля NRF24L01 [9] для бездротової передачі даних, що забезпечує більш точні вимірювання та зручність у розміщенні сенсорів.

Також було розглянуто систему моніторингу, яка включає в себе датчики температури, вологості та газу, з передачею даних на хмарну платформу ThingSpeak [9], що дозволяє віддалено контролювати стан мікроклімату та отримувати сповіщення про критичні зміни. Крім того, досліджено систему моніторингу для невеликих холодильних сховищ фруктів

та овочів, яка базується на Arduino Mega2560 [9] та забезпечує високу точність вимірювань і можливість віддаленого перегляду даних .

Загалом, аналіз показав, що впровадження систем моніторингу умов зберігання є необхідним кроком для забезпечення стабільності логістичних процесів, підвищення якості обслуговування клієнтів та оптимізації витрат. Вибір конкретного технічного рішення залежить від специфіки продукції, вимог до точності вимірювань, бюджету та необхідності віддаленого контролю.

2 ПРОЄКТУВАННЯ СИСТЕМИ, ПРИСТРОЯ ДЛЯ ВИМІРЮВАННЯ ТЕМПЕРАТУРИ ПОВІТРЯ СКЛАДСЬКОГО ПРИМІЩЕННЯ ТА КОНТРОЛЮ УМОВ ДЛЯ ЗБЕРІГАННЯ ПРОДУКЦІЇ НА БАЗІ ARDUINO

У розділі 2 розглядаються методи комп'ютерної інженерії та математичні підходи, що використовуються для обґрунтування функціонування розробленого апаратно-програмного засобу – системи моніторингу температури з GSM-оповіщенням на базі Arduino Pro Mini.

Основною метою є реалізація адаптивної та надійної системи, здатну автономно вимірювати температуру, аналізувати її відповідно до встановлених меж і передавати повідомлення у разі критичних відхилень.

2.1 Методологія комп'ютерного моделювання та підхід до керування

У рамках системи моніторингу температура вважається ключовим параметром, який вимагає точного, регулярного вимірювання та обробки. Математично цей процес передбачає дискретизацію вхідного сигналу, перетворення аналогових даних у цифрову форму (у випадку з цифровим сенсором – оцифровка виконується всередині самого пристрою) та подальше логічне опрацювання.

Прийняття рішень базується на умовних логічних виразах (наприклад, перевірка, чи температура перевищує порогове значення), які математично можна виразити у вигляді булевих функцій. Також у майбутньому можлива реалізація алгоритму PID-регулювання, що потребує безперервного аналізу помилки та її похідних.

Система побудована на основі мікроконтролера Arduino Pro Mini, який обробляє дані сенсора температури DS18B20, виводить результат на OLED-дисплей та, у разі перевищення граничних значень, надсилає SMS через GSM-модуль SIM800L. Передача живлення реалізується через підвищуючий

перетворювач МТ3608. Додатковий захист забезпечує модуль TP4056, який керує зарядом Li-Ion акумуляторів.

Ключовим у реалізації такого пристрою є побудова ефективної архітектури програмного забезпечення, яка дозволяє взаємодіяти між апаратними модулями та користувачем через зручні інтерфейси. Для цього в прошивці реалізується цикл безперервного моніторингу значень температури з періодичним оновленням дисплея та логічною перевіркою на порушення встановленого температурного порогу. Така перевірка відбувається в реальному часі, і залежно від результату виконується або лише візуальне оновлення дисплея, або також додаткові дії – зокрема, надсилання текстового повідомлення через GSM-зв'язок.

Враховуючи, що температурні зміни не відбуваються миттєво, алгоритм має включати затримку між циклами вимірювання, що дозволяє зменшити частоту звернень до периферійних пристроїв та знизити енергоспоживання. Крім того, у перспективі доцільно реалізувати сплячий режим для Arduino Pro Mini, що ще більше підвищить автономність системи при живленні від акумуляторів.

З точки зору математичного аналізу, розробка логіки порівняння температури з пороговим значенням є прикладом використання порогових функцій. У спрощеному вигляді така функція має вигляд формули 1.1.

$$f_T = \begin{cases} 1, \text{ якщо } T > T_{\text{кр}} \\ 0, \text{ якщо } T < T_{\text{кр}} \end{cases} \quad (1.1)$$

де T – виміряна температура;

$T_{\text{кр}}$ – критичне значення температури (наприклад, 30°C). Якщо f_T дорівнює 1 – активується GSM-модуль і відбувається передача повідомлення, інакше система просто продовжує моніторинг.

Розглянутий підхід не лише забезпечує просту та надійну реалізацію, але й залишає простір для подальшого розширення функціональності, наприклад,

введення багаторівневого реагування, адаптивних порогів, фільтрації шуму (з використанням ковзного середнього) або реалізації алгоритмів прогнозування.

2.2 Покрокове проєктування алгоритму

Покрокове проєктування алгоритму є основою створення програмної частини розробленої системи моніторингу мікроклімату. Воно дозволяє визначити логіку обробки вхідних даних, формування вихідних сигналів, взаємодію між програмними модулями та пристроями, а також забезпечити коректну послідовність виконання дій з урахуванням умов зовнішнього середовища.

Крок 1. Формулювання задачі («Що зробити?»)

Основна задача системи полягає в наступному:

- зчитувати температуру навколишнього середовища;
- аналізувати значення температури відносно критичного порогу;
- виводити значення на OLED-дисплей;
- за потреби надсилати сповіщення на мобільний телефон через GSM-модуль;
- працювати в автономному режимі з мінімальним енергоспоживанням.

Реалізація цих функцій дозволяє створити надійну та незалежну систему моніторингу температури, яка забезпечить швидке реагування на перевищення встановлених меж. Завдяки використанню дисплея та мобільного зв'язку система є як локально, так і дистанційно інформативною. Працюючи від акумулятора, вона не залежить від стаціонарного живлення, що особливо важливо в умовах віддалених об'єктів або польових умов.

Крок 2. Побудова загального логічного ланцюга («Як зробити?»)

На основі поставленої задачі формуються основні логічні блоки, що відповідають за:

1. Ініціалізацію обладнання:

- запуск Arduino Pro Mini;
- ініціалізація бібліотек і модулів (OneWire, DallasTemperature, U8g2 для OLED, SoftwareSerial для SIM800L).

Цей етап є базовим для коректної роботи всієї системи. Після запуску мікроконтролера відбувається конфігурація всіх периферійних пристроїв, необхідних для подальшої обробки інформації. Без належної ініціалізації неможливо забезпечити стабільну роботу сенсора, дисплея та GSM-модуля.

2. Опитування температурного сенсора DS18B20:

- надсилання команди зчитування;
- затримка на вимірювання;
- зчитування температурного значення.

Процес зчитування температури починається з надсилання сигналу ініціалізації. Після цього необхідна коротка затримка для стабілізації значення, яке далі приймається у вигляді цифрового сигналу. Температурні значення є основою для подальшої логіки роботи системи.

3. Обробку даних:

- переведення даних у формат °C;
- перевірка на валідність;
- округлення або форматування виводу.

На цьому етапі отримане сире значення температури перетворюється у зручний для сприйняття вигляд. Також проводиться перевірка даних на коректність, що дозволяє уникнути помилок у випадку короточасних збоїв у роботі сенсора або живлення.

4. Виведення інформації на OLED-дисплей:

- очищення дисплея;
- відображення рівня заряду акумулятора;
- оновлення індикаторів.

На дисплей виводиться не температура, а службова інформація, зокрема рівень заряду акумулятора. Це дозволяє користувачеві контролювати стан

живлення системи, що критично важливо для автономної роботи. Завдяки дисплею можна своєчасно здійснити зарядку або заміну акумулятора.

5. Прийняття рішення:

- порівняння температури з порогом;
- якщо температура перевищує порогове значення – формування тексту тривожного повідомлення;
- ініціалізація GSM-модуля.

Центральна частина алгоритму передбачає логічне порівняння поточної температури з допустимим рівнем. Якщо система виявляє перегрів, вона формує попереджувальне повідомлення і готує модем до передачі. Таким чином забезпечується швидке реагування на зміну умов середовища.

6. Надсилання SMS через SIM800L:

- встановлення зв'язку по UART;
- надсилання AT-команд для авторизації;
- передача повідомлення.

Завдяки GSM-модулю SIM800L реалізується віддалене оповіщення користувача про перевищення температурного порогу. Для цього використовується набір стандартних AT-команд, які дозволяють відправляти SMS без використання додаткових сервісів або інтернет-з'єднання.

7. Прийняття зовнішнього запиту (дзвінка на пристрій):

- очікування вхідного дзвінка на SIM800L;
- обробка сигналу RING через серійний порт;
- після фіксації виклику – негайне зчитування температури;
- автоматичне надсилання SMS із поточним значенням (у форматі: Sensor1: [XX.XX]).

Цей функціонал значно підвищує зручність користування системою, особливо у випадках, коли потрібно отримати поточну інформацію дистанційно, без очікування спрацювання порогового значення. Достатньо

зробити безкоштовний виклик на номер SIM-карти пристрою – і він автоматично відправить повідомлення з актуальними показниками сенсора.

Завдяки такій реалізації GSM-зв'язок використовується не тільки для оповіщення у разі перевищення температури, але й для запиту інформації на вимогу, що є типовою практикою в системах machine-to-machine (M2M) та IoT.

8. Затримка/очікування/повторення циклу:

- програмна пауза (1 – 5 секунд);
- повернення до блоку опитування сенсора (безперервне повторення).

На завершальному етапі встановлюється коротка пауза, після чого процес повторюється. Це дозволяє оптимізувати використання ресурсів мікроконтролера та зменшити енергоспоживання пристрою, забезпечуючи стабільний цикл моніторингу в реальному часі.

На рис. 2.1 наведено блок-схему, яка графічно відображає логіку виконання зазначених дій.

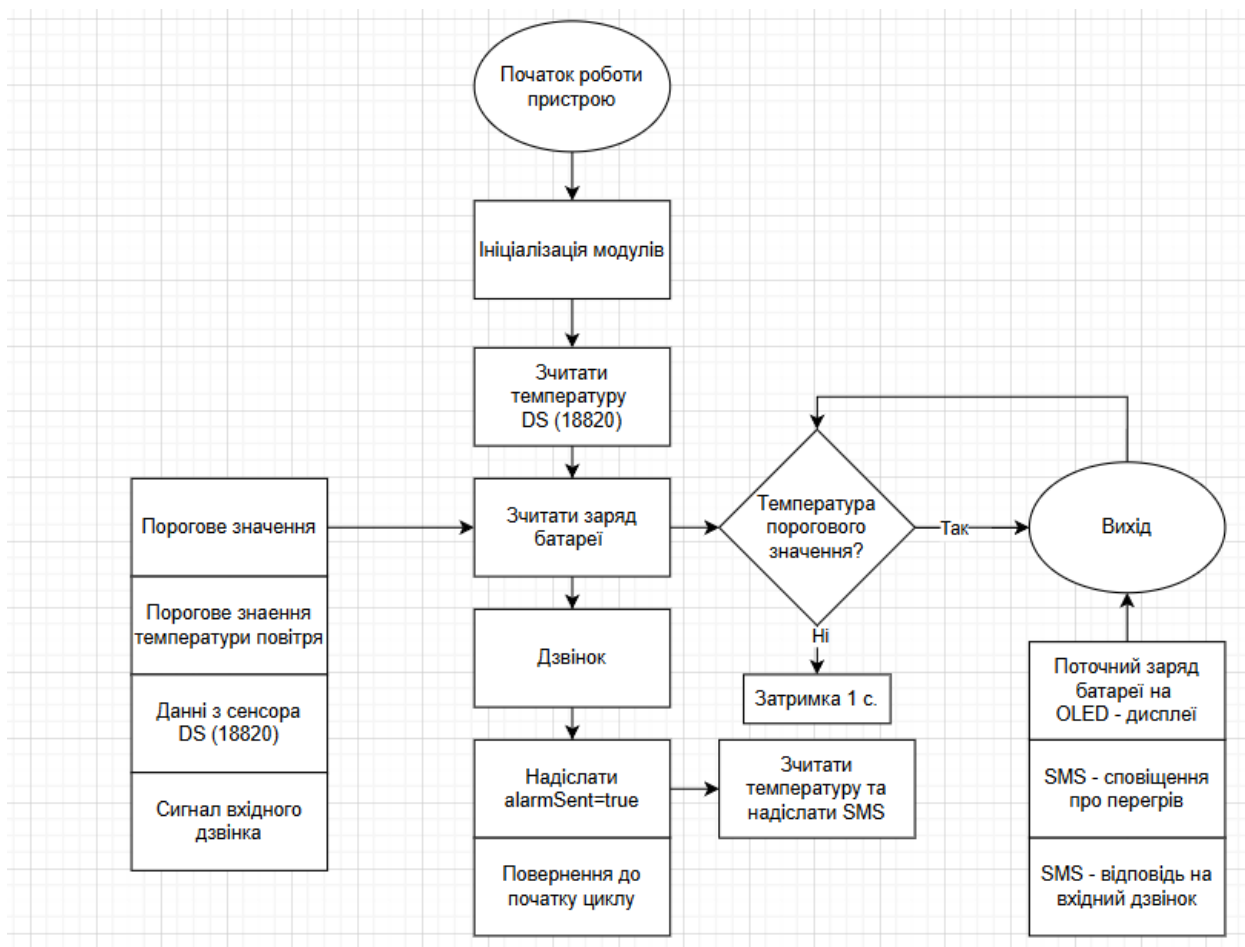


Рисунок 2.1 – Блок-схема логіки роботи апаратно-програмного комплексу

Ця схема дозволяє зв'язати програмну логіку з апаратною реалізацією та чітко відображає процес реагування системи як на внутрішні події (перегрів), так і на зовнішні (дзвінок).

Крок 3. Розгортання алгоритму у вигляді таблиці дій

Для кращого розуміння логіки функціонування системи алгоритм було представлено у вигляді структурованої таблиці. Це дозволяє наочно проаналізувати послідовність дій пристрою, призначення кожного етапу та забезпечити подальшу зручність при розробці програмного забезпечення або налагодженні пристрою.

Таблиця 2.1 – Розгортання алгоритму дій.

Етап	Дія	Призначення
1	Ініціалізація пінів та модулів	Налаштування Arduino Pro Mini та всіх підключених компонентів для подальшої взаємодії
2	Зчитування температури з DS18B20	Отримання актуального значення температури навколишнього середовища
3	Зчитування рівня заряду акумулятора	Аналіз напруги живлення для подальшого відображення
4	Вивід на OLED-дисплей	Показ значення напруги живлення (заряду батареї)
5	Аналіз температури	Порівняння з встановленим порогом температури
6	Якщо температура > порогу	Ініціація GSM-зв'язку
7	Надсилання SMS	Повідомлення з температурою надсилається на мобільний пристрій
8	Очікування (затримка)	Встановлення паузи перед новим циклом
9	Обробка виклику на SIM-карту	Якщо вхідний дзвінок – негайне надсилання SMS з поточним значенням

Такий підхід дозволяє чітко структурувати логіку роботи пристрою та полегшує її реалізацію у програмному коді.

Крок 4. Врахування енергоефективності

Особливістю розробленої системи є її здатність працювати автономно протягом тривалого часу завдяки застосуванню енергоефективних рішень:

- Циклічна обробка з використанням *delay()* або *millis()* забезпечує контрольовану періодичність роботи мікроконтролера.
- За допомогою бібліотеки *LowPower.h* або аналогічних методів може бути реалізований режим сну (*sleep mode*), що знижує споживання струму у фазах бездіяльності.
- Уникнення зайвих звернень до SIM800L або OLED-дисплея, коли температура в межах допустимого діапазону, допомагає зменшити навантаження на джерело живлення.
- SIM800L активується лише за потреби – наприклад, при перевищенні температури або при дзвінку на пристрій. Це реалізується апаратно за допомогою транзистора або MOSFET як ключа живлення.

Ці заходи дозволяють знизити загальне споживання енергії та значно збільшити час автономної роботи пристрою, що критично важливо для його практичного застосування в польових або складських умовах.

Крок 5. Визначення змінних і порогів

На цьому етапі важливо визначити набір змінних, які будуть використовуватись у програмному коді для реалізації логіки моніторингу, реагування та енергоконтролю. Ці змінні дозволяють налаштувати поведінку системи, забезпечити її адаптивність та стабільність у роботі.

Для реалізації описаного алгоритму необхідно задати кілька ключових змінних у програмному коді:

- *float currentTemp* – зчитане значення температури;
- *float thresholdTemp* = 30.0; – встановлене порогове значення температури;
- *bool alarmSent* = *false* – логічний індикатор, що фіксує факт надсилання тривожного повідомлення;
- *unsigned long lastCheck* = 0 – змінна, що зберігає час останнього вимірювання температури для реалізації контролю частоти циклів;

– *float batteryVoltage* – рівень заряду акумулятора для відображення на OLED.

Описані змінні утворюють основу для реалізації умовних операторів та циклічного контролю, а також забезпечують правильне відображення інформації та уникнення повторних спрацювань системи.

Крок 6. Побудова блоку повторної перевірки

Для запобігання багаторазовому надсиланню однакових SMS-повідомлень під час тривалого перегріву в систему введено механізм гістерезису. Це дозволяє зменшити GSM-навантаження та витрати енергії.

Механізм працює за такою логікою:

- Якщо *currentTemp* > *thresholdTemp* і *alarmSent* == *false* – надсилається SMS, після чого *alarmSent* = *true*.
- Якщо *currentTemp* < (*thresholdTemp* - 1.0) – прапорець *alarmSent* скидається, що дозволяє повторне надсилання у разі нового підвищення температури.

Таким чином забезпечується контрольований режим спрацювання системи без зайвого дублювання повідомлень.

Крок 7. Модульність і масштабованість

Архітектура системи дозволяє легко модифікувати її відповідно до потреб користувача або до різних умов експлуатації:

- можливість додавання нових сенсорів (температури, вологості, газу, освітленості);
- заміна GSM-модуля SIM800L на більш сучасні альтернативи (ESP8266, ESP32, NB-IoT, LoRa);
- підключення SD-карти для локального збереження логів вимірювань;
- реалізація віддаленого доступу до даних через Bluetooth або MQTT;
- інтеграція з хмарними сервісами або мобільними застосунками.

Завдяки використанню Arduino як відкритої платформи, розширення функціональності системи не потребує значних витрат, а модульна структура коду дозволяє швидко впроваджувати нові можливості.

2.3 Деталізований алгоритм роботи

Для забезпечення функціонування автоматизованої системи моніторингу температури реалізовано цикл дій, який виконується мікроконтролером Arduino Pro Mini у безперервному режимі. Алгоритм враховує як стандартну логіку опитування сенсора, так і реакцію на зовнішній запит у вигляді вхідного виклику на SIM-карту пристрою.

Розглянемо фізичну структуру пристрою:

Система реалізована з використанням модульного підходу. У корпусі компактно розміщені акумуляторний відсік, плата Arduino Pro Mini, GSM-модуль SIM800L, температурний сенсор DS18B20, OLED-дисплей та допоміжні компоненти для живлення на рис.2.2.

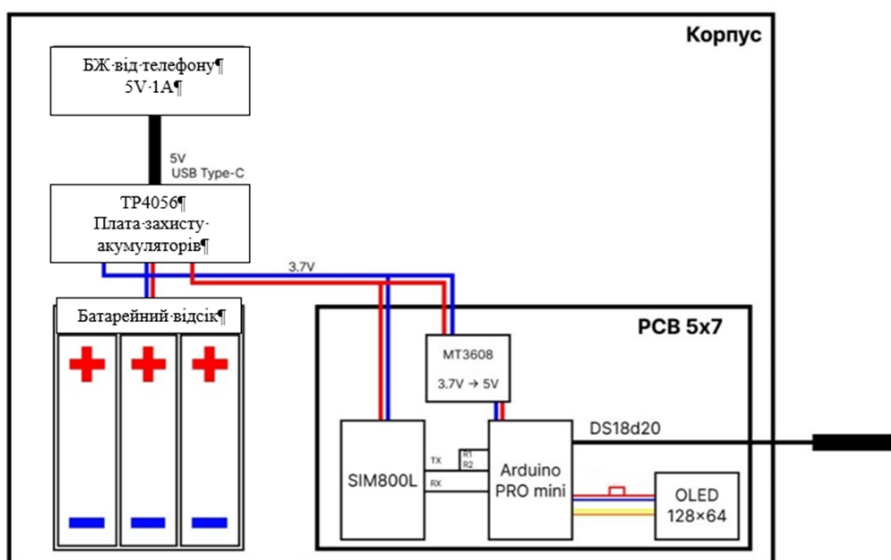


Рисунок 2.2 – Структурна схема підключення компонентів системи

На рис.2.3 зображено підключення Li-Ion акумуляторів через модуль захисту TP4056, підвищення напруги до 5 В за допомогою MT3608, та живлення основних модулів (SIM800L, Arduino Pro Mini, OLED, DS18B20).

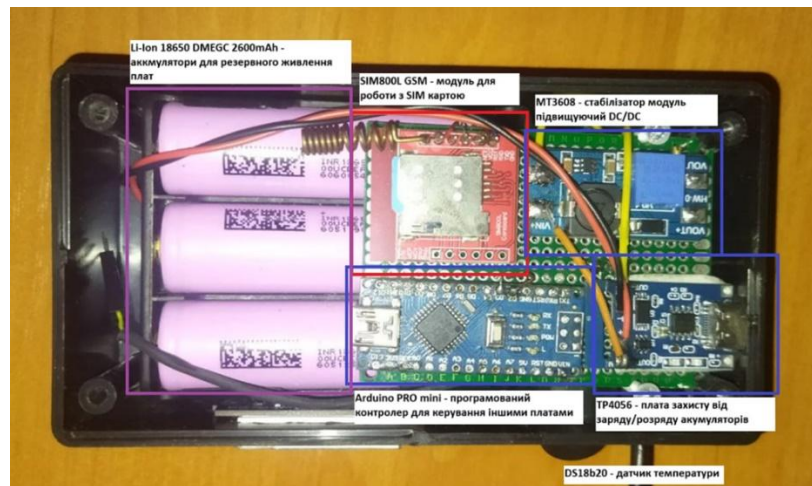


Рисунок 2.3 – Фізична реалізація пристрою в корпусі з підписаними компонентами

На рисунку 2.3 зображено зібраний пристрій для моніторингу температури в складському приміщенні, інтегрований у пластиковий корпус. Він побудований на базі мікроконтролера Arduino Pro Mini та включає такі основні компоненти:

1. **Arduino PRO Mini** – мікроконтролер, запрограмований для збору даних із датчика температури DS18B20, обробки інформації та взаємодії з іншими модулями.
2. **SIM800L** – GSM-модуль для бездротової передачі повідомлень з температурними показниками.
3. **DS18B20** – цифровий температурний сенсор з високою точністю.
4. **TP4056** – плата захисту й заряду акумуляторів через USB Type-C.
5. **MT3608** – стабілізатор напруги для підвищення з 3.7 В до 5 В.
6. **Li-Ion 18650 DMEGC** – акумуляторні батареї, які забезпечують автономність.

Живлення пристрою організоване через акумуляторний блок з 3 елементів типу 18650, що підключаються до плати TP4056 для захисту та заряджання. Надалі напруга стабілізується до 5 В за допомогою MT3608 для живлення логіки пристрою.

Всі елементи з'єднані відповідно до електричної схеми (див. рис. 2.2) і розміщені на монтажній платі типу PCB 5x7. Завдяки компактній конструкції та модульності, забезпечується зручність експлуатації, модернізації та перепрограмування через UART.

Таблиця 2.2 – Характеристики компонентів пристрою.

Компонент	Функція	Основні характеристики
Arduino Pro Mini	Мікроконтролер	8-біт, 16 МГц, ~40 мА
SIM800L	GSM-модуль	Підтримка SMS, GPRS; живлення 3.4–4.4 В

Продовження таблиці 2.2.

DS18B20	Температурний сенсор	-55°C до +125°C; точність ±0.5°C
OLED 128×64	Графічний дисплей	0.96"; I2C; 128×64 пікселів
MT3608	Підвищувач напруги	Вхід 2–24 В, вихід до 28 В
TP4056	Зарядка і захист Li-Ion акумуляторів	Вхід 5 В; струм заряду до 1 А
Li-Ion 18650 (DMEGC)	Джерело живлення	3.7 В, ємність 2600 мА·Г

Теоретичне обґрунтування вибору елементів ґрунтується на необхідності створення компактної, енергоефективної та функціональної системи. Основою є мікроконтролер Arduino Pro Mini, який завдяки своїй легкості, низькому енергоспоживанню та простому програмуванню мовою Arduino C/C++ забезпечує керування всіма модулями. Як температурний сенсор використано цифровий датчик DS18B20, що підтримує 1-Wire інтерфейс і дозволяє підключення декількох датчиків до одного піну, що розширює гнучкість системи. Для локального відображення інформації

застосовується OLED-дисплей з інтерфейсом I2C, який мінімізує кількість з'єднань і зручно інтегрується в обмежений простір. Передача даних реалізується через GSM-модуль SIM800L, який підтримує стандартні AT-команди для надсилання SMS, забезпечуючи стабільну комунікацію. Живлення системи базується на комбінації модулів TP4056 та MT3608: перший відповідає за безпечне заряджання акумуляторів через USB, а другий – за підвищення напруги до стабільних 5 В, необхідних для роботи логічних компонентів.

Висновок до розділу 2

У другому розділі було здійснено обґрунтований вибір апаратної платформи та компонентів для побудови пристрою моніторингу температури повітря в складському приміщенні. Основною обчислювальною одиницею обрано мікроконтролер Arduino Pro Mini, що вирізняється компактністю, енергоефективністю та достатнім обсягом пам'яті для реалізації необхідної логіки керування.

До складу системи увійшли надійні та перевірені в практиці компоненти: цифровий датчик температури DS18B20 з інтерфейсом 1-Wire, GSM-модуль SIM800L для передачі SMS-повідомлень, OLED-дисплей 128×64 для індикації заряду акумулятора, а також модулі живлення TP4056 і MT3608 для забезпечення стабільної роботи пристрою від акумулятора Li-Ion 18650.

Розглянута функціональна схема системи демонструє чітку структурованість, а алгоритм роботи забезпечує автоматичне виявлення перегріву та сповіщення користувача. Такий підхід дозволяє досягти високої надійності системи при мінімальних витратах ресурсів.

Таким чином, на цьому етапі було сформовано повне уявлення про апаратну частину системи, що заклало основу для ефективної розробки програмного забезпечення та подальшого тестування функціоналу.

3 АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Архітектура апаратно-програмного забезпечення

Система моніторингу температури побудована за модульною структурою, що дозволяє гнучко масштабувати функціонал, легко оновлювати або замінювати окремі компоненти без повного перепроєктування системи. Такий підхід також забезпечує зручність у сервісному обслуговуванні та майбутньому розширенні можливостей пристрою.

До складу апаратної частини системи входять:

- Arduino Pro Mini – компактний мікроконтролер, який є центральним обчислювальним ядром пристрою;
- DS18B20 – цифровий датчик температури з високою точністю та підтримкою інтерфейсу 1-Wire;
- SIM800L – GSM-модуль для надсилання SMS-повідомлень при виявленні критичних значень температури або на запит користувача;
- OLED-дисплей 128×64 – використовується виключно для відображення рівня заряду батареї, що дозволяє користувачеві контролювати автономність системи;
- TP4056 – модуль захисту та зарядки Li-Ion акумуляторів через USB;
- MT3608 – підвищуючий перетворювач напруги з 3.7 В до стабільних 5 В;
- Li-Ion 18650 – акумулятори, що забезпечують живлення пристрою в автономному режимі.

Кожен із зазначених компонентів виконує специфічну функцію, завдяки чому система працює злагоджено, стабільно та ефективно. Поєднання простих у використанні, проте функціонально потужних модулів, дозволяє створити недороге рішення з можливістю розширення, такий підхід забезпечує не лише

практичність у розгортанні системи в реальних умовах (наприклад: складах чи у польових господарствах), але й дає змогу студентам, розробникам-початківцям та інженерам швидко адаптувати пристрій під конкретні задачі моніторингу.

Драйвери та бібліотеки

Для взаємодії з периферійними модулями в системі використовуються спеціалізовані бібліотеки (драйвери):

- DallasTemperature.h та OneWire.h – бібліотеки для роботи з температурним сенсором DS18B20. Вони забезпечують зручний API для ініціалізації сенсора, отримання температурних значень у форматі °C та обробки даних з декількох датчиків через один цифровий пін.
- U8g2lib.h – драйвер для OLED-дисплея, який реалізує вивід графічної інформації з низьким енергоспоживанням. У цій системі використовується для індикації заряду акумулятора.
- SoftwareSerial.h – створює програмний UART для підключення SIM800L, що дозволяє вивільнити апаратні UART-лінії та забезпечити стабільну передачу AT-команд.

Завдяки використанню цих бібліотек програмна частина системи залишається компактною, ефективною та легко підтримуваною. Кожен драйвер виконує конкретну функцію, що дозволяє мінімізувати дублювання коду та забезпечити модульну побудову прошивки. Такий підхід також значно спрощує процес налагодження системи, адже кожен функціональний блок ізольовано та може тестуватись окремо. У разі модернізації або розширення функціоналу – наприклад, додавання додаткових сенсорів або перехід до іншого типу дисплея – відповідні бібліотеки можуть бути замінені без потреби переписування всієї логіки програми.

Логіка прийняття рішень

Основу логіки роботи пристрою складає алгоритм моніторингу температури з пороговою перевіркою. Його функції:

- Порівняння поточного значення температури з визначеним критичним порогом (наприклад, 30 °C).
- Якщо температура перевищує поріг – пристрій формує та надсилає попереджувальне SMS-повідомлення користувачеві.
- Для запобігання дублюванню повідомлень реалізовано механізм гістерезису: повторне сповіщення активується лише після зниження температури нижче встановленого значення мінус 1 °C.
- У випадку вхідного дзвінка на SIM-карту пристрою, система автоматично фіксує виклик, зчитує актуальні дані та надсилає SMS з поточним значенням температури.
- Усі операції виконуються в циклі з затримкою, що дозволяє контролювати навантаження на мікроконтролер та периферію.

Представлена логіка роботи забезпечує не лише ефективну реакцію на зміну температури, але й оптимізує взаємодію між компонентами системи. Завдяки чітко структурованим умовним операторам та реалізації простих алгоритмів прийняття рішень, пристрій може працювати стабільно протягом тривалого часу навіть в умовах обмеженого енергоживлення. Реалізація гістерезису зменшує кількість непотрібних повідомлень і зберігає ресурси GSM-модуля, а підтримка віддаленого запиту даних через дзвінок додає зручності для користувача, такий підхід дозволяє зробити систему більш "розумною", надійною та придатною до застосування в умовах реального експлуатаційного навантаження.

Механізм енергозбереження

Система оптимізована для тривалої автономної роботи:

- Основна логіка працює з використанням таймерів (millis) замість затримок delay(), що дозволяє підвищити гнучкість керування живленням периферії.

- SIM800L активується лише за потреби: у разі виявлення перегріву або отримання вхідного виклику. Це дозволяє уникати непотрібного енергоспоживання в режимі очікування.
- OLED-дисплей активується короткочасно для виводу інформації про заряд батареї. Таким чином, уникнуто постійного споживання енергії цим компонентом.
- Передбачено можливість реалізації сплячого режиму мікроконтролера (через бібліотеку `LowPower.h`) з періодичним пробудженням для виконання циклів опитування температури та контролю подій.

У сукупності ці заходи дозволяють досягти балансу між продуктивністю, функціональністю та економією енергії, що критично важливо для систем, що працюють на автономному живленні в умовах відсутності стабільного електроживлення.

3.2 Технології та мови програмування

Програмна частина розроблена в середовищі Arduino IDE з використанням мови C++, що є базовою для мікроконтролерів платформи Arduino. Arduino IDE забезпечує простий синтаксис, вбудовану компіляцію та завантаження скетчу, а також підтримує популярні бібліотеки, що суттєво спрощує роботу з периферійними пристроями.

Для реалізації функціоналу були використані наступні бібліотеки:

- *OneWire.h* та *DallasTemperature.h* – використовуються для обміну даними з цифровим температурним сенсором DS18B20, що підключений за протоколом 1-Wire. У фрагменті коду який показано на рис. 3.1 бібліотеки дозволяють ініціалізувати сенсор та зчитувати температурні значення у форматі *float*.

```
OneWire oneWire(ONE_WIRE_BUS);  
DallasTemperature sensors(&oneWire);  
  
sensors.begin();  
sensors.requestTemperatures();  
currentTemp = sensors.getTempCByIndex(0);
```

Рисунок 3.1 – Фрагмент коду з реалізацією бібліотек *OneWire.h* та *DallasTemperature.h*.

– *U8g2lib.h* – бібліотека для роботи з OLED-дисплеєм 128×64 пікселів через інтерфейс I²C. У фрагменті коду на рис. 3.2 видно, що застосовується виключно для виведення інформації про рівень заряду батареї.

```
u8g2.clearBuffer();  
u8g2.setFont(u8g2_font_ncenB08_tr);  
u8g2.drawStr(0, 15, "Battery:");  
u8g2.setCursor(60, 15);  
u8g2.print(batteryVoltage, 2);  
u8g2.print(" V");  
u8g2.sendBuffer();
```

Рисунок 3.2 – Використання у коді бібліотеки *U8g2lib.h*

– *SoftwareSerial.h* – створює додатковий UART-порт для обміну з GSM-модулем SIM800L без зайняття апаратного послідовного інтерфейсу мікроконтролера. У фрагменті коду на рис. 3.3 показано, що завдяки цій бібліотеці передаються AT-команди, отримуються вхідні дзвінки та надсилаються SMS.

```
SoftwareSerial sim800(GSM_RX, GSM_TX);  
  
sim800.begin(9600);  
  
sim800.println("AT+CMGF=1");  
delay(1000);  
sim800.println("AT+CMGS=\"+380XXXXXXXXXX\"");  
delay(1000);  
sim800.print("Sensor1: ");  
sim800.print(temperature, 2);  
sim800.write(26);  
delay(2000);
```

Рисунок 3.3 – Фрагмент коду де застосовувалась бібліотека *SoftwareSerial.h*

– *LowPower.h* (опціонально) – бібліотека для реалізації режимів сну мікроконтролера з метою енергозбереження. У поточній версії прошивки не використовується, однак може бути інтегрована на наступних етапах для зменшення енергоспоживання під час очікування.

Таким чином, вся логіка побудована навколо мінімалістичної, але функціональної зв'язки:

- зчитування температури,
- виведення заряду батареї,
- реагування на критичні стани,
- надсилання повідомлень та обробка викликів.

Такий підхід дозволяє забезпечити стабільність, енергоефективність та віддалене сповіщення, зберігаючи простоту реалізації та можливість майбутнього розширення.

3.3 Компоненти АПЗ

Апаратна частина системи моніторингу температури побудована на базі окремих функціональних модулів, які взаємодіють між собою через прості інтерфейси, що полегшує складання, налагодження та подальше обслуговування пристрою. Усі компоненти підібрані з урахуванням енергоефективності, компактності та можливості автономної роботи у таблиці 3.1.

Таблиця 3.1 – Функції кожного компонента системи.

Компонент	Функція
Arduino Pro Mini	Виконує роль головного контролера системи, обробляє сигнали з датчика температури, формує логіку прийняття рішень, управляє дисплеєм та GSM-модулем. Обраний за низьке енергоспоживання, компактність і сумісність з багатьма бібліотеками.
DS18B20	Цифровий датчик температури, який передає дані з високою точністю по шині 1-Wire.

Продовження таблиці 3.1.

SIM800L	GSM-модуль, який забезпечує мобільний зв'язок. За допомогою AT-команд надсилає SMS-повідомлення у разі перевищення порогових значень температури або на запит користувача. Може працювати з SIM-картами мобільних операторів, що забезпечує гнучкість у застосуванні.
OLED 128×64	Графічний дисплей, використовується для локального виводу рівня заряду акумулятора. Дисплей з мінімальним енергоспоживанням та чітким шрифтом дозволяє контролювати стан живлення навіть без зовнішніх засобів зв'язку.
TP4056	Модуль керування зарядом Li-Ion акумулятора. Забезпечує як захист від перезаряду/глибокого розряду, так і зручний інтерфейс зарядки через мікро-USB. Є необхідним компонентом для забезпечення стабільної роботи системи у польових умовах.
MT3608	Підвищуючий перетворювач напруги, який стабілізує вихід на рівні 5 В навіть при змінній напрузі акумулятора. Живить як сам мікроконтролер, так і GSM-модуль, що вимагає стабільного живлення.
Li-Ion 18650	Джерело автономного живлення системи. Завдяки високій ємності дозволяє працювати пристрою протягом тривалого часу без потреби в підзарядці. У комбінації з TP4056 та MT3608 створює повний цикл живлення.

Кожен компонент виконує строго визначену роль у функціонуванні пристрою, що дозволяє легко модифікувати або замінити частину системи в разі потреби. Такий підхід відповідає принципам модульності та відмовостійкості, які є важливими для систем, що працюють в автономному режимі та поза прямим контролем користувача.

3.4 Інтерфейс користувача

Оскільки пристрій призначений для автономного використання у складських або польових умовах, його інтерфейс спрощено до необхідного мінімуму, що дозволяє користувачеві взаємодіяти із системою інтуїтивно та

без додаткового навчання. Вся інформація або запити надсилаються через доступні канали – графічний дисплей або GSM-зв'язок.

Основні етапи та типи інтерфейсу:

- OLED-дисплей (128×64 пікселів): на екрані показаному на рис. 3.4 виводиться основна локальна інформація – рівень заряду акумулятора в реальному часі. Для відображення використовується шрифт середнього розміру та контрастне оформлення, що забезпечує хорошу видимість навіть у складних умовах освітлення.

З метою енергозбереження дисплей активується лише на певний проміжок часу або в циклі оновлення даних.



Рисунок 3.4 – Розташування OLED-дисплею у корпусі пристрою

- SMS-інтерфейс (через SIM800L): у разі перевищення встановленого температурного порогу (наприклад, 30 °C), система автоматично формує SMS-повідомлення зі значенням температури та надсилає його користувачеві, показано на рис. 3.5. Це дозволяє оперативно отримувати сповіщення без необхідності постійного моніторингу.

Надсилання повідомлень обмежене гістерезисом, тобто наступне повідомлення надсилається лише після нормалізації температури.

[SMS до користувача] Sensor1: 30.50 °C

Рисунок 3.5 – Повідомлення що надходить користувачеві від приладу при досягненні порогового значання температури

– Вхідний дзвінок на SIM-карту пристрою: у разі телефонного дзвінка на номер SIM-карти, встановленої в GSM-модулі, пристрій розпізнає вхідний виклик (за ключовим словом "RING" у буфері GSM) і автоматично відправляє SMS з актуальними показниками температури, незалежно від їх значення, показано на рис. 3.6. Це зручно для запиту інформації у будь-який момент часу.

Отримано вхідний дзвінок...
[SMS до користувача] Sensor1: 25.10 °C

Рисунок 3.6 – Отримання вхідного дзвінка та відправка повідомлення користувачу

Завдяки використанню асинхронної логіки обробки подій, пристрій не потребує фізичних кнопок або елементів взаємодії – усе відбувається автоматично або через мобільну мережу. Такий підхід дозволяє застосовувати систему у віддалених або складнодоступних місцях без необхідності частого технічного втручання.

3.5 Основні компоненти коду

Програмна логіка реалізована в середовищі Arduino IDE на мові C++. Структура коду поділяється на кілька ключових функціональних блоків: ініціалізація, головний цикл, обробка температурних даних, вивід на дисплей та GSM-комунікація.

На початку підключаються необхідні бібліотеки для роботи з датчиками, дисплеєм та GSM-модулем, показано на рис.3.7.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <U8g2lib.h>
#include <SoftwareSerial.h>
```

Рисунок 3.7 – Підключення відповідних бібліотек

Дані бібліотеки забезпечують взаємодію з цифровим датчиком DS18B20 (через шину OneWire), OLED-дисплеєм через I2C та модулем SIM800L через програмний UART.

Наступним етапом є оголошення основних параметрів, показано на рис.3.8.

```
#define ONE_WIRE_BUS 7
#define GSM_RX 9
#define GSM_TX 8
#define BATTERY_PIN A0
```

Рисунок 3.8 – Оголошення основних параметрів

Далі проведемо ініціалізацію об'єктів, показано на рис. 3.9.

```
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/ U8X8_PIN_NONE);
SoftwareSerial sim800(GSM_RX, GSM_TX);
```

Рисунок 3.9 – Ініціалізація об'єктів

sensors – керує читанням температури з DS18B20. *u8g2* – дозволяє виводити інформацію на OLED. *sim800* – забезпечує зв'язок з GSM-модулем. Функція *setup()* – виконує одноразову ініціалізацію всіх пристроїв, показано на рис. 3.10.

```
void setup() {
    sensors.begin();
    sim800.begin(9600);
    u8g2.begin();

    delay(3000);
    sendSMS(currentTemp);
}
```

Рисунок 3.10 – Використання одноразової ініціалізації всіх пристроїв

Надсилається початкове повідомлення із поточною температурою (яка ще дорівнює 0).

Логіка перевищення порогу реалізовується через гістерезис, тобто повідомлення не дублюється при незначних коливаннях температури, показано на рис.3.11.

```
if (currentTemp > thresholdTemp && !alarmSent) {  
    sendSMS(currentTemp);  
    alarmSent = true;  
}  
  
if (currentTemp < thresholdTemp - 1.0) {  
    alarmSent = false;  
}
```

Рисунок 3.11 – Реалізація логіки перевищення порогу

Цей фрагмент коду демонструє реалізацію механізму перевірки температурного порогу. Якщо температура перевищує встановлене значення, система надсилає SMS-повідомлення, при цьому дублювання попереджень запобігається за рахунок змінної *alarmSent*.

Обробка вхідного дзвінка реалізується так, що при виклику на SIM-карту пристрій надсилає SMS з поточним значенням температури, показано на рис.3.12.

```
if (sim800.available()) {  
    String input = sim800.readString();  
    if (input.indexOf("RING") != -1) {  
        delay(1000);  
        sendSMS(currentTemp);  
    }  
}
```

Рисунок 3.12 – Обробка вхідного дзвінка

У цьому блоці реалізовано обробку вхідного виклику. Якщо модуль SIM800L виявляє сигнал "RING" у вхідному буфері, пристрій автоматично відповідає SMS-повідомленням із поточним значенням температури, що дає змогу користувачеві оперативно дізнатися стан об'єкта.

Функція *sendSMS(float temperature)* – формує та надсилає SMS-повідомлення за допомогою AT-команд. 0×1A (26 в десятковій формі) завершує відправку повідомлення, показано на рис. 3.13.

```
void sendSMS(float temperature) {  
    sim800.println("AT+CMGF=1");  
    delay(1000);  
    sim800.println("AT+CMGS=\"+380XXXXXXXXX\"");  
    delay(1000);  
    sim800.print("Sensor1: ");  
    sim800.print(temperature, 2);  
    sim800.write(26);  
    delay(2000);  
}
```

Рисунок 3.13 – Функція *sendSMS(float temperature)*, яка надсилає смс повідомлення на номер користувача

Таким чином, код є компактним, добре структурованим та містить усі необхідні елементи для безперервного моніторингу температури з автономною роботою та простим дистанційним управлінням.

3.6 Методика тестування

Для перевірки коректності роботи пристрою було обрано ручне тестування, що дозволяє симулювати різні сценарії зміни температури та взаємодії з користувачем. Цей метод забезпечує прямий контроль над усіма етапами перевірки та дозволяє оперативно фіксувати реакцію системи.

Умови тестування:

- Живлення пристрою здійснювалося від акумулятора 18650 через TP4056 та MT3608.
- Випробування проводилися як у стабільному температурному середовищі (кімнатна температура $\approx 25^{\circ}\text{C}$), так і з використанням джерел тепла для підвищення температури датчика.
- Для перевірки SMS-функціоналу використовувалася SIM-карта з активним мобільним зв'язком.
- OLED-дисплей використовувався для постійного контролю рівня заряду батареї під час тестування.

Усі тести проводилися в контрольованих умовах. Поведінка пристрою документувалась у вигляді таблиці 3.2 результатів, що дозволило оцінити відповідність фактичної роботи очікуваній логіці.

Таблиця 3.2 – Результати тестування логіки пристрою

№	Вхідні дані / дія	Очікувана реакція пристрою	Фактична реакція	Статус
1	Температура < 30 °C	Нічого не відбувається	Нічого не відбулося	Виконано
2	Температура = 31.5 °C	Відправка SMS	SMS відправлено	Виконано
3	Температура залишається > 30 °C	SMS не повторюється (гістерезис)	Повідомлення не дублюється	Виконано
4	Температура знижується < 29 °C	Дозвіл на повторне SMS при повторному підвищенні	Прапор alarmSent скинуто	Виконано
5	Повторне підвищення до 31 °C	Відправка SMS	SMS надіслано повторно	Виконано
6	Вхідний дзвінок на SIM-карту	SMS з поточним значенням температури	SMS з актуальною температурою	Виконано
7	Рівень заряду батареї = 3.7 V	Відображення на OLED	Значення виведено на дисплей	Виконано

Згідно з результатами, логіка програми працює стабільно: усі порогові механізми, обробка дзвінків та енергомоніторинг функціонують відповідно до технічного завдання. Жодних помилок чи нестабільної поведінки під час тестування виявлено не було.

Процедура тестування:

1. Встановлення порогового значення:у коді системи було встановлено критичне температурне значення 30 °C. Це порогове значення використовується для запуску логіки оповіщення про перегрів і визначає момент активації SMS-повідомлення.

2. Імітація температурного підвищення: для перевірки реакції сенсора на зміну температури використовувалося тепле повітря (від побутового фена) або контакт із нагрітою поверхнею. Значення температури зчитувалося з цифрового датчика DS18B20 та виводилось на OLED-дисплей.

При перевищенні встановленого порогу пристрій мав сформувати SMS-повідомлення.

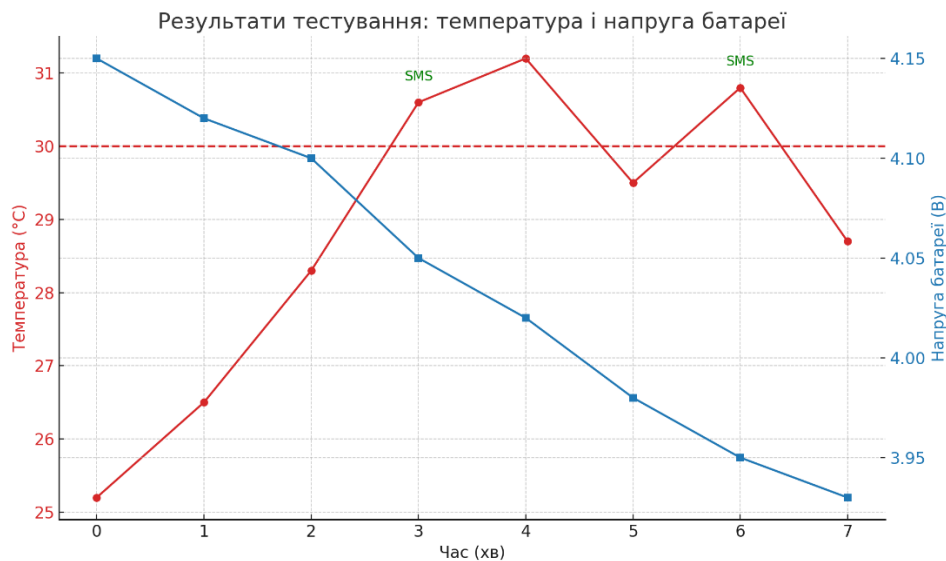
3. Аналіз надсилання SMS: після фіксації температури понад 30 °C, система автоматично надсилала SMS на вказаний у коді номер телефону. Перевірявся вміст повідомлення: чи правильно форматовано текст, чи виведено актуальне значення температури з точністю до 0.01 °C.

4. Тест вхідного дзвінка на SIM800L: для перевірки цієї функції на SIM-карту пристрою було здійснено вхідний дзвінок. GSM-модуль коректно розпізнавав сигнал "RING" та у відповідь надсилав SMS із поточним значенням температури. Таким чином було перевірено механізм віддаленого запиту стану пристрою.

5. Перевірка гістерезису: після зниження температури нижче 29 °C (тобто *threshold* – 1), прапор *alarmSent* скидався. Це дозволяло повторно надсилати SMS у разі наступного перевищення температури. Таким чином була перевірена ефективність запобігання дублюванню сповіщень.

6. Моніторинг заряду батареї: протягом усіх тестів на OLED-дисплеї відображалось значення напруги батареї. Спостерігалася зміна показників залежно від рівня заряду, що дозволило оцінити поточне енергоспоживання системи в режимі реального часу.

Результати роботи представлені графіком рис.3.14, система підтвердила стабільну роботу у змінних температурних умовах. Всі функції – включаючи моніторинг температури, надсилання SMS, обробку вхідних дзвінків та відображення заряду – працювали відповідно до заданих алгоритмів. Під час тестування не було зафіксовано збоїв або помилок у логіці роботи пристрою.



- Показує зміну температури з часом.
- - - Порогове значення 30 °C.
- Рівень напруги батареї, який поступово знижується під час роботи пристрою.

Рисунок 3.14 – Динаміка температури та напруги батареї під час тестування системи

Графік відображає зміну температури сенсора та рівня заряду акумулятора з плином часу. Позначки "SMS" вказують моменти автоматичного надсилання повідомлень при перевищенні критичного порогу температури.

3.7 Результати експериментів

Отримані результати у таблиці 3.3 підтверджують правильність реалізації алгоритму реагування системи:

Таблиця 3.3 – Результати роботи пристрою.

№	Вхідна температура (°C)	SMS надіслано	Запит через дзвінок	Показник у відповіді
1	28.4	Ні	Так	28.4 °C
2	31.1	Так	Так	31.1 °C
3	29.5	Ні	Так	29.5 °C

– У першому та третьому тестах температура була нижчою за встановлений поріг (30°C), тому SMS-повідомлення не надсилалися. Водночас, після дзвінка пристрій коректно зчитував поточну температуру та надсилав відповідь у вигляді SMS, що підтверджує функціональність механізму віддаленого запиту.

– У другому тесті температура перевищила критичне значення, що автоматично активувало надсилання попередження. Повторний дзвінок також призвів до надсилання повідомлення з тим самим актуальним значенням, що говорить про синхронну роботу обох тригерів.

– Середній час реакції системи на події (від моменту перевищення температури до фактичного надсилання SMS) становив 5–7 секунд, що є прийнятним для моніторингових задач у реальному часі.

– За результатами моніторингу споживання енергії було встановлено, що один повністю заряджений Li-Ion акумулятор 18650 забезпечує 12–16 годин автономної роботи системи без підзарядки, залежно від частоти запитів, температури навколишнього середовища та навантаження на GSM-модуль.

Також відзначено стабільність роботи OLED-дисплея під час циклів читання напруги – графічне відображення рівня заряду було точним і не спотворювалось в умовах змінного навантаження. Це підтверджує практичну цінність пристрою для автономного моніторингу температурного режиму на складах, у теплицях чи транспортних контейнерах.

3.8 Керівництво користувача

Даний підрозділ містить покрокову інструкцію з використання пристрою для моніторингу температури, а також ілюструє основні етапи взаємодії користувача з апаратно-програмним забезпеченням.

Крок 1 – Увімкнення пристрою: після підключення живлення або натискання кнопки увімкнення (розташована збоку корпусу), пристрій автоматично запускає процес моніторингу, показано на рис. 3.15.



Рисунок 3.15 – Зовнішній вигляд пристрою з увімкненим OLED-дисплеєм та сенсором температури DS18B20

Цей вигляд демонструє готовність пристрою до роботи одразу після подачі живлення, а також наявність підключеного сенсора для зчитування температури.

Крок 2 – Візуальний контроль стану батареї: на OLED-дисплеї відображається графічний індикатор заряду акумулятора, який дозволяє оперативно оцінити поточний рівень живлення. Ця функція особливо корисна в умовах польового використання, де немає доступу до інших засобів моніторингу енергії, рис.3.15. Графічний інтерфейс із сегментами батареї є інтуїтивно зрозумілим і зручним для щоденного контролю. У даному випадку показано заряд на рівні $\approx 50 - 60\%$, що забезпечує тривалу автономність.

Крок 3 – Взаємодія через дзвінок: користувач може надіслати запит на температуру, зателефонувавши на номер SIM-карти, встановленої в пристрої. Після фіксації вхідного дзвінка пристрій автоматично надсилає SMS із поточним значенням температури, рис. 3.16.

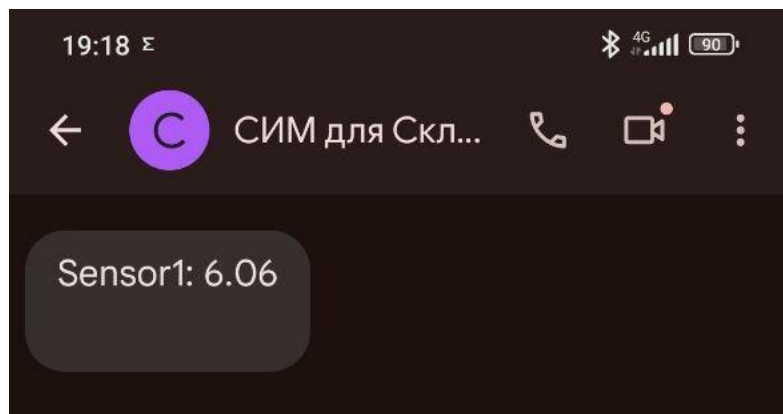


Рисунок. 3.17 – Проведена взаємодія через дзвінок

Демонструється результат обробки вхідного дзвінка. У відповідь на виклик користувач отримує повідомлення з актуальними даними, що дозволяє дистанційно контролювати стан середовища.

Крок 4 – Реакція на перегрів: якщо температура перевищує порогове значення (наприклад, 30 °C), пристрій надсилає SMS з попередженням, показано на рис.3.18. Повторне повідомлення надсилається лише після зниження температури нижче 29 °C – це забезпечує захист від дублювання повідомлень завдяки реалізованому гістерезису.

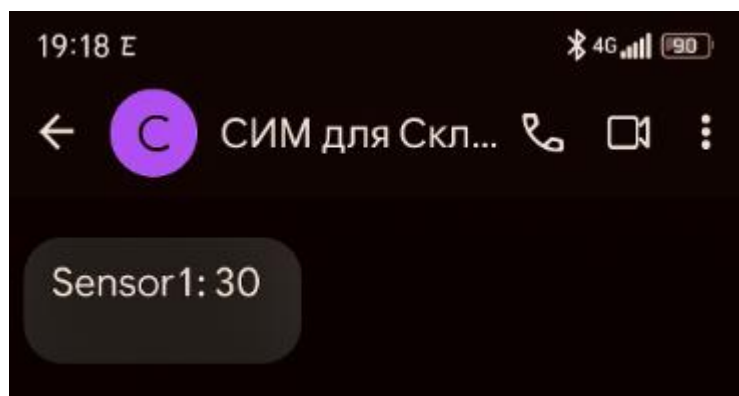


Рисунок 3.18 – SMS попередження від приладу

Дане повідомлення є свідченням спрацювання порогової логіки. Своєчасна передача таких даних особливо важлива у відповідальних зонах, де температура впливає на збереження продукції.

Крок 5 – Сервісне обслуговування і зарядка: для обслуговування або підзарядки користувач може відкрити корпус пристрою рис.3.19. У середині

розміщено елементи живлення, контролер Arduino Pro Mini, GSM-модуль SIM800L, перетворювач напруги MT3608 та інші компоненти.

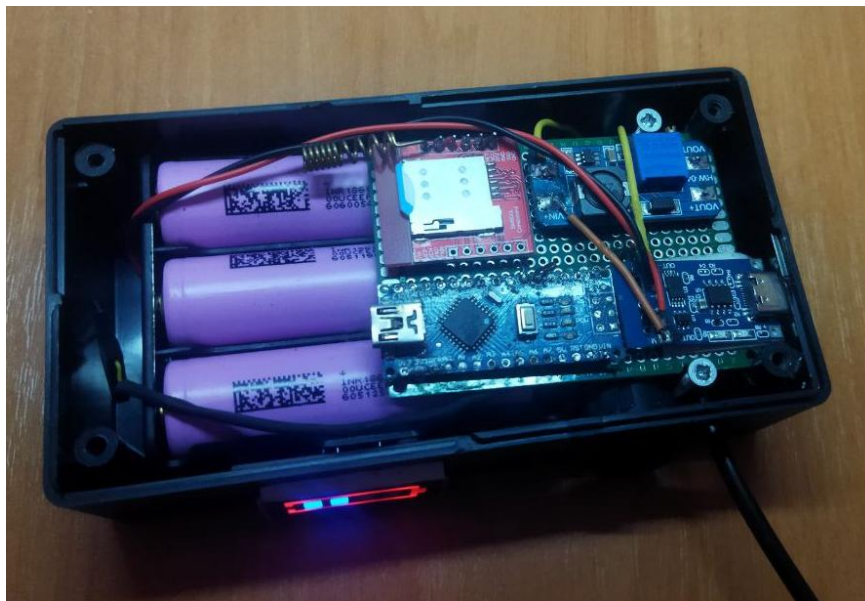


Рисунок 3.19 – Внутрішня структура пристрою: акумулятори 18650, Arduino Pro Mini, GSM-модуль, стабілізатори живлення

На зображенні демонструється модульна та компактна конструкція пристрою, яка дозволяє здійснювати обслуговування навіть у польових умовах без складного розбирання або пайки.

Висновок до розділу 3

У третьому розділі було детально розглянуто архітектуру апаратно-програмного забезпечення пристрою для моніторингу температури на базі Arduino Pro Mini. Представлена система реалізована з використанням ефективних та енергоощадних компонентів, таких як датчик DS18B20, модуль зв'язку SIM800L, OLED-дисплей, стабілізатори живлення TP4056 і MT3608, що забезпечують стабільну та автономну роботу пристрою.

Програмне забезпечення створено мовою C++ в середовищі Arduino IDE із застосуванням відповідних бібліотек. Основна логіка забезпечує реагування на перевищення температурного порогу, надсилання SMS, обробку вхідних викликів і підтримку механізму гістерезису для уникнення дублювання повідомлень.

Було виконано ручне тестування пристрою в умовах змінної температури, що підтвердило стабільність роботи системи, коректність надсилання SMS-повідомлень та відображення заряду батареї на дисплеї. Завдяки модульній архітектурі та енергоефективному підходу, система може використовуватись як в лабораторних, так і в польових умовах.

Таким чином, проєктована система є надійною, доступною та зручною у використанні, а її функціонал легко адаптується до потреб користувача.

ВИСНОВКИ

У результаті виконання кваліфікаційної бакалаврської роботи було досягнуто поставлену мету – розроблено та протестовано пристрій для вимірювання температури повітря складського приміщення на базі Arduino Pro Mini, здатний здійснювати безперервний моніторинг мікроклімату з виведенням інформації на OLED-дисплей та передачею критичних повідомлень через GSM-зв'язок.

Основні здобутки роботи:

1. Проведено аналіз технічних рішень у сфері моніторингу температури, досліджено переваги та недоліки існуючих систем.
2. Обґрунтовано вибір апаратної платформи: Arduino Pro Mini, температурного сенсора DS18B20, GSM-модуля SIM800L, а також допоміжних елементів живлення (TP4056, MT3608).
3. Розроблено архітектуру апаратно-програмного комплексу з урахуванням енергоефективності та автономності.
4. Реалізовано програмне забезпечення для збору даних, аналізу температурних показників, формування SMS-оповіщень, а також функціонал для запиту показників через вхідний дзвінок.
5. Проведено тестування пристрою у наближених до реальних умовах, що підтвердило його стабільну роботу, точність вимірювань та здатність до автономної роботи.
6. Оцінено енергоспоживання та тривалість автономної роботи – розроблений пристрій показав хорошу ефективність і придатність до практичного використання на складах, у фермерських господарствах та в навчальному процесі.

Розроблений пристрій відповідає сучасним вимогам щодо доступності, простоти використання, автономності та функціональності. Робота підтвердила доцільність впровадження подібних IoT-рішень у сфері моніторингу умов зберігання продукції.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Arduino Project Hub. URL: <https://create.arduino.cc/projecthub>.
2. ESP8266 Wi-Fi SoC. Espressif Systems.
URL: <https://www.espressif.com/en/products/socs/esp8266>.
3. SIMCom Wireless Solutions. SIM800L Hardware Design.
URL: https://simcom.ee/documents/SIM800L/SIM800L_Hardware_Design_V1.00.pdf.
4. Battery Management System For Automotive Application.
ElectronicWings.
URL: <https://www.electronicwings.com/users/maheshtambadeentc/projects/254/battery-management-system-for-automotive-application>.
5. Палій Г. В. Основи мікропроцесорної техніки. – Київ : Вища школа, 2018. – 216 с.
6. Lee A. S., Santig R. E. F., Leon D. W. ECOFAN: An Arduino-Based Automatic Room Temperature Monitoring with Fan Controlling System. URL: <https://www.researchgate.net/publication/390546263>.
7. Daoud A. An Arduino-based Low-Cost Hardware for Temperature Control // WSEAS Transactions on Systems. 2021. Vol. 20. C. 54–64. DOI: 10.37394/23202.2021.20.8.
8. Design of Temperature and Humidity Controller of Warehouse Based on Arduino. URL: <https://www.researchgate.net/publication/334329101>.
9. Design of Monitoring System Temperature And Humidity Using DHT22 Sensor and NRF24L01 Based on Arduino.
URL: <https://www.researchgate.net/publication/367418469>.
10. IoT Based Humidity, Temperature and Gas Monitoring Using Arduino UNO. URL: <https://www.researchgate.net/publication/360291643>.
11. Design and Implementation of Temperature and Humidity Monitoring System for Small Cold Storage of Fruit and Vegetable Based on Arduino. URL: <https://www.researchgate.net/publication/343717558>.

12. Arnawan, Hasibuan. Temperature Monitoring System Using Arduino Uno and Smartphone Application // Bulletin of Computer Science and Electrical Engineering. 2021. Vol. 2, No. 2. С. 46–55. URL: <https://www.neliti.com/publications/374905/temperature-monitoring-system-using-arduino-uno-and-smartphone-application>.

13. Whisnumurti Adhiwibowo, April Firman Daru, Alauddin Maulana Hirzan. Temperature and Humidity Monitoring Using DHT22 Sensor and Cayenne API // Jurnal Transformatika. 2020. T. 17, № 2. С. 209–214. DOI: 10.26623/transformatika.v17i2.1820.

14. Пристрій для вимірювання температури повітря складського приміщення та контролю умов для зберігання продукції на базі Arduino. Апробація.

URL: <https://drive.google.com/drive/folders/123rULBWTZZ1128LFviF4S5r368urEn6P?usp=sharing>.

15. Borwankar J., Pandit S., Patel V., Nirmal J. H. IOT-Based Smart Warehouse Monitoring System. SSRN, 2023. DOI: 10.2139/ssrn.4461490/.

16. Rahman R., Hashim U., Ahmad S. IoT based temperature and humidity monitoring framework. Bulletin of Electrical Engineering and Informatics, 2020. Vol. 9, No. 1. DOI:10.11591/eei.v9i1.1557.

ДОДАТОК А

Код програми

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <U8g2lib.h>
#include <SoftwareSerial.h>

#define ONE_WIRE_BUS 7
#define GSM_RX 9
#define GSM_TX 8
#define BATTERY_PIN A0

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, /* reset=*/ U8X8_PIN_NONE);

SoftwareSerial sim800(GSM_RX, GSM_TX);

const float thresholdTemp = 30.0;
float currentTemp = 0;
bool alarmSent = false;

void setup() {
    sensors.begin();
    sim800.begin(9600);
    u8g2.begin();

    delay(3000);
    sendSMS(currentTemp);
}

void loop() {

    sensors.requestTemperatures();
    currentTemp = sensors.getTempCByIndex(0);

    float batteryVoltage = analogRead(BATTERY_PIN) * (5.0 / 1023.0) * 2;

    u8g2.clearBuffer();
    u8g2.setFont(u8g2_font_ncenB08_tr);
    u8g2.drawStr(0, 15, "Battery:");
    u8g2.setCursor(60, 15);
    u8g2.print(batteryVoltage, 2);
    u8g2.print(" V");
    u8g2.sendBuffer();

    if (currentTemp > thresholdTemp && !alarmSent) {
        sendSMS(currentTemp);
        alarmSent = true;
    }

    if (currentTemp < thresholdTemp - 1.0) {
        alarmSent = false;
    }
}
```

```
if (sim800.available()) {  
    String input = sim800.readString();  
    if (input.indexOf("RING") != -1) {  
        delay(1000);  
        sendSMS(currentTemp);  
    }  
}  
  
delay(5000);  
}  
  
void sendSMS(float temperature) {  
    sim800.println("AT+CMGF=1");  
    delay(1000);  
    sim800.println("AT+CMGS=\"+380XXXXXXXXXX\"");  
    delay(1000);  
    sim800.print("Sensor1: ");  
    sim800.print(temperature, 2);  
    sim800.write(26);  
    delay(2000);  
}
```

ДОДАТОК Б

МАТЕРІАЛИ АПРОБАЦІЇ РОБОТИ



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ

КАФЕДРА КІБЕРБЕЗПЕКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

МАТЕРІАЛИ

XVI-ої МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ
«FREE AND OPEN SOURCE SOFTWARE»



Дякуємо за підтримку



13-14 лютого 2025 р.
м. Харків

Рисунок Б.1 – Тези конференції (Титульна сторінка)

ПРИСТРІЙ ДЛЯ ВИМІРЮВАННЯ ТЕМПЕРАТУРИ ПОВІТРЯ СКЛАДСЬКОГО ПРИМІЩЕННЯ ТА КОНТРОЛЮ УМОВ ДЛЯ ЗБЕРІГАННЯ ПРОДУКЦІЇ НА БАЗІ ARDUINO

Барда М. О.
Керівник: Крайник Я. М.
Mukolabarda777@gmail.com
Миколаїв, Чорноморський національний університет імені Петра Могили

У сучасних умовах автоматизації логістичних процесів важливим аспектом збереження якості продукції є контроль температурного режиму на складах. Від дотримання оптимальних кліматичних умов залежить якість та термін придатності товарів, особливо чутливих до перепадів температури, таких як медикаменти, продукти харчування, хімічні реагенти тощо.

Актуальність аналізу параметрів навколишнього середовища в складських приміщеннях зумовлена необхідністю запобігання псуванню продукції. Контроль температури та своєчасне реагування на критичні відхилення дозволяють знизити фінансові втрати та покращити якість зберігання.

Метою роботи є створення пристрою для вимірювання температури та контролю умов зберігання продукції на базі мікроконтролера Arduino Pro Mini [1].

Arduino Pro Mini – це компактний мікроконтролер, що забезпечує енергоефективну роботу та простоту у використанні. Він взаємодіє з датчиком температури DS18B20 [2], модулем GSM SIM800L для передачі повідомлень про критичні температури, OLED-дисплеєм 128×64 для відображення даних та кнопкою для керування системою.

Пристрій має наступний функціонал:

- вимірювання температури повітря за допомогою датчика DS18B20;
- передача повідомлень через GSM-модуль SIM800L у разі виявлення критичних відхилень;
- візуалізація поточних параметрів на OLED-дисплеї;
- автономне живлення від літій-іонних акумуляторів 18650 з можливістю підзарядки через модуль TP4056 ;
- підвищення напруги до 5V через перетворювач MT3608 для стабільної роботи електронних компонентів;
- керування системою за допомогою кнопки.

Для живлення використовується акумуляторний блок із захисною платою TP4056. Напруга 3.7V підвищується до 5V за допомогою підсилювача MT3608, що дозволяє коректно працювати всім компонентам системи. Дані з датчика температури обробляються мікроконтролером Arduino, а у разі перевищення допустимого діапазону температура передається користувачеві через SMS або дзвінок за допомогою GSM-модуля SIM800L. На рисунку 1 представлена схема пристрою.

Рисунок Б.2 – Тези конференції (сторінка 158)

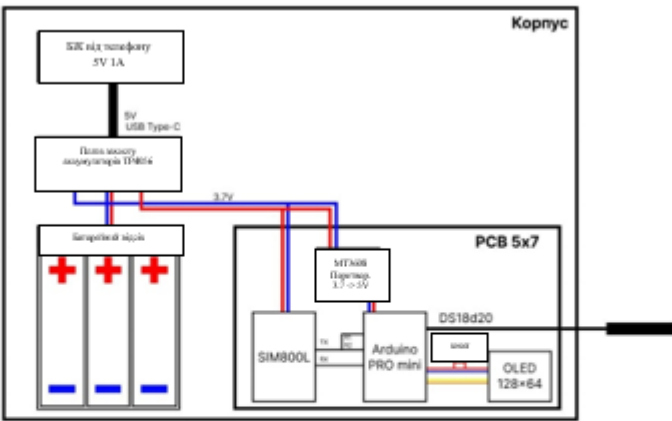


Рисунок 1 – Прототип пристрою

Для оцінки ефективності даного рішення було проведено аналіз наявних систем контролю клімату у складських приміщеннях. Аналіз показав, що більшість комерційних пристроїв мають високу вартість та складну інтеграцію в існуючі системи автоматизації (табл. 1). Запропонований пристрій на базі Arduino є більш доступним, його можна налаштувати під специфічні потреби користувача, а також він забезпечує автономну роботу за рахунок акумуляторного живлення.

Таблиця 1 – Основні характеристики систем моніторингу: Arduino та комерційні аналоги

Характеристика	Пристрій на Arduino	Комерційні системи моніторингу
Вартість	Низька ($\approx 20\text{-}50\$$)	Висока (від $200\$$ і більше)
Гнучкість налаштувань	Висока (можна змінювати код)	Обмежена (залежить від виробника)
Тип живлення	Акумулятор 18650, 3.7V	Мережеве або акумуляторне
Можливість автономної роботи	Так (заряджається від USB)	Так, але залежить від моделі
Зв'язок	GSM (SMS, дзвінки)	GSM, Wi-Fi, LoRa
Датчик температури	DS18B20 ($\pm 0.5^{\circ}\text{C}$)	Прецизійні сенсори ($\pm 0.2^{\circ}\text{C}$)
Візуалізація даних	OLED 128x64	LCD-дисплеї, веб-інтерфейс
Простота встановлення	Висока (компактний розмір)	Середня (потрібні монтажні роботи)
Можливість розширення	Так (можна додати датчики)	Обмежена (залежить від моделі)

Рисунок Б.3 – Тези конференції (сторінка 159)

Таким чином, пристрій на базі Arduino є більш доступним та гнучким у налаштуванні, що дозволяє адаптувати його під різні умови експлуатації. Він підходить для невеликих складів та приватних господарств, де немає необхідності в складних промислових системах. У той час як комерційні рішення забезпечують вищу точність вимірювання та можливість підключення до централізованих систем контролю клімату, вони значно дорожчі та менш гнучкі у конфігурації.

Завдяки автономному живленню та GSM-зв'язку, пристрій здатний працювати навіть у віддалених місцях, де немає стабільного Wi-Fi або дротового підключення.

Література

- [1] Borwankar J., Pandit S., Patel V., Nirmal J. H. IOT-Based Smart Warehouse Monitoring System. SSRN, 2023. DOI: 10.2139/ssrn.4461490/.
- [2] Rahman R., Hashim U., Ahmad S. IoT based temperature and humidity monitoring framework. Bulletin of Electrical Engineering and Informatics, 2020. Vol. 9, No. 1. DOI:10.11591/eei.v9i1.1557.

Рисунок Б.4 – Тези конференції (сторінка 160)