Міністерство освіти і науки України

Луцький національний технічний університет
(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій
(повне найменування факультету)

Кафедра комп'ютерної інженерії та безпеки
(повне найменування кафедри)

КВАЛІФІКАЦІЙНА РОБОТА ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

КОНТРОЛЕР ШВИДКОСТІ ВЕНТИЛЯТОРА В ЗАЛЕЖНОСТІ ВІД ТЕМПЕРАТУРИ З ВИКОРИСТАННЯМ ARDUINO UNO

TEMPERATURE-DEPENDENT FAN SPEED CONTROLLER USING ARDUINO UNO

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІс-21
Кринчик Віталій Олександрович

(підпис)

Керівник:
к.т.н., доцент
Поліщук Микола Миколайович

(підпис)

Кваліфікаційну роботу
допущено до захисту
« 10 » червня 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *комп'ютерної інженерії та безпеки*
Ступінь вищої освіти: *бакалавр*
Галузь знань: *12 Інформаційні технології*
Спеціальність: *123 Комп'ютерна інженерія*
Освітня програма: *«Комп'ютерна інженерія»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ доц. Т. ТЕРЛЕЦЬКИЙ
« 10 » 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Кринчику Віталію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Контролер швидкості вентилятора в залежності від температури з використанням Arduino UNO*

Керівник роботи *к.т.н., доц. Поліщук Микола Миколайович*

затверджені наказом закладу вищої освіти від «20» березня 2025 року № 170/01-02
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 10.06.2025р.

3. Вихідні дані до роботи *джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування.*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Вступ
Теоретичні основи автоматичного керування температурозалежними системами
Апаратна реалізація контролера температурозалежного керування вентилятором
Практична реалізація контролера швидкості вентилятора з використанням Arduino Uno
Висновки

5. Перелік графічного (ілюстративного) матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Теоретичні основи автоматичного керування температурозалежними системами	*Поліщук М.М., доцент*		
Апаратна реалізація контролера температурозалежного керування вентилятором	*Поліщук М.М., доцент*		
Практична реалізація контролера швидкості вентилятора з використанням Arduino Uno	*Поліщук М.М., доцент*		
Нормоконтроль	*Багнюк Н.В., доцент*		
Гарант ОП	*Лавренчук С.В., доцент*		
Показник запозичень тексту	___%		
Академічна доброчесність	*Міскевич О.І.,ст.викладач*		

7. Дата видачі завдання 22.03.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	*Теоретичні основи автоматичного керування температурозалежними системами. Огляд літератури із досліджуваної проблеми*	до 30.03.2025 р.	Виконано
2.	*Апаратна реалізація контролера температурозалежного керування вентилятором*	до 12.04.2025 р.	Виконано
3.	*Практична реалізація контролера швидкості вентилятора з використанням Arduino Uno*	до 28.04.2025 р.	Виконано
4.	*Висновки та пропозиції*	до 01.05.2025 р.	Виконано
5.	*Формування списку використаних джерел*	до 09.05.2025 р.	Виконано
6.	*Формування додатків*	до 11.05.2025 р.	Виконано
7.	*Оформлення ілюстративного матеріалу*	до 13.05.2025 р.	Виконано
8.	*Представлення остаточного варіанту кваліфікаційної роботи керівникові*	до 15.05.2025 р.	Виконано
9.	*Нормоконтроль*	до 30.05.2025 р.	Виконано
10	*Інструментальна перевірка на академічний плагіат*	до 06.06.2025 р	Виконано
11.	*Здача кваліфікаційної роботи та всіх супровідних документів на кафедру*	до 10.06.2025 р.	Виконано

Здобувач вищої освіти _____ (підпис) Кринчик В.О. (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ (підпис) Поліщук М.М. (прізвище, ініціали)

АНОТАЦІЯ

Кринчик В. О. Контролер швидкості вентилятора в залежності від температури з використанням Arduino UNO. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, списку використаних джерел, додатку.

Перший розділ присвячено огляду предметної області, тут обґрунтовано актуальність обраної теми. Представлено призначення та сфери застосування систем керування температурою, принципи побудови системи автоматичного регулювання температури та вимоги до контролера швидкості вентилятора. Представлено огляд омпонентів та методів керування в системах контролю температури

В другому розділі здійснено вибір апаратних компонентів для розробки контролеру швидкості вентилятора на основі Arduino UNO. Представлена програмна реалізація температурозалежного керування та можливості інтеграції системи в реальні умови експлуатації.

Третій розділ присвячено практичній реалізація контролера швидкості вентилятора з використанням Arduino UNO. Зроблено огляд комплектуючих системи. Представлено складання контролера швидкості вентилятора з використанням Arduino Uno та програмуванга мікроконтролерара в середовищі розробки Arduino IDE.

Ключові слова: контролер швидкості, температура, мікроконтролер, Arduino IDE, керування, вентилятор, датчик

ANNOTATION

Krynchyk V. Temperature-dependent fan speed controller using Arduino UNO. Manuscript.

Qualification work of the bachelor of the OP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

Qualification work consists of an introduction, three sections, conclusions, a list of sources used, an appendix.

The first section is devoted to an overview of the subject area, the relevance of the chosen topic is substantiated here. The purpose and scope of application of temperature control systems, the principles of building an automatic temperature control system and the requirements for a fan speed controller are presented. An overview of components and control methods in temperature control systems is presented

The second section selects hardware components for the development of a fan speed controller based on Arduino UNO. The software implementation of temperature-dependent control and the possibilities of integrating the system into real operating conditions are presented.

The third section is devoted to the practical implementation of a fan speed controller using Arduino UNO. An overview of the system components is made. The assembly of a fan speed controller using Arduino Uno and programming of the microcontroller in the Arduino IDE development environment are presented.

Keywords: speed controller, temperature, microcontroller, Arduino IDE, control, fan, sensor

ЗМІСТ

ВСТУП

У сучасних умовах інтенсивного розвитку електроніки, автоматизації та мікроконтролерних систем особливої актуальності набуває створення інтелектуальних систем керування температурним режимом. Однією з таких систем є автоматичний контролер швидкості обертання вентилятора, який забезпечує оптимальне охолодження електронного або технічного обладнання залежно від температури навколишнього середовища чи конкретного об’єкта. Нестабільність температурного режиму, перегрів компонентів та недостатня вентиляція можуть суттєво впливати на довговічність і надійність технічних систем, що зумовлює потребу у впровадженні засобів динамічного контролю за охолодженням.

Застосування мікроконтролера Arduino Uno у ролі центрального керуючого модуля дає змогу реалізувати просте, доступне та гнучке рішення для побудови автоматизованих систем керування. Завдяки доступності апаратних модулів, відкритому програмному забезпеченню та активній спільноті розробників, Arduino стає ідеальним вибором для реалізації проєктів у галузі температурного моніторингу та керування.

Об’єкт дослідження – процес автоматичного керування швидкістю вентилятора залежно від температури середовища.

Предмет дослідження – апаратно-програмна реалізація контролера на базі мікроконтролера Arduino Uno для управління вентилятором за температурними показниками.

Мета роботи полягає у розробці та практичній реалізації мікропроцесорного пристрою на базі Arduino Uno, який забезпечує автоматичне регулювання швидкості обертання вентилятора залежно від поточної температури

Для досягнення поставленої мети було поставлено ряд завдань:

– визначити ключові сфери застосування систем автоматичного керування температурою;

– сформувати вимоги до контролера швидкості вентилятора з урахуванням необхідності підтримки PWM та цифрових інтерфейсів;

– провести огляд компонентів і методів їх застосування у системах керування температурою;

– обґрунтувати вибір апаратних компонентів системи;

– реалізувати програмну частину системи, яка здійснює зчитування температури.

Така система має забезпечити ефективне охолодження, знизити рівень шуму та покращити енергоспоживання в порівнянні з традиційними вентиляційними системами постійної швидкості.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ АВТОМАТИЧНОГО КЕРУВАННЯ ТЕМПЕРАТУРОЗАЛЕЖНИМИ СИСТЕМАМИ

1.1 Призначення та сфери застосування систем керування температурою

Системи керування температурою (СКТ) є ключовим компонентом сучасної техніки та промислових процесів, що забезпечують підтримку заданих температурних режимів для безпечної та ефективної роботи пристроїв. Зростання вимог до точності, енергоефективності та надійності систем спонукає до впровадження інтелектуальних регуляторів, зокрема, з використанням мікроконтролерів типу Arduino для локального моніторингу й керування.

Основною функцією будь-якої системи керування температурою є підтримка температури об’єкта або середовища в межах допустимого діапазону. Це досягається за допомогою сенсорів, контролерів і виконавчих механізмів, що взаємодіють у режимі зворотного зв’язку. Залежно від умов і сфери застосування, такі системи можуть бути реалізовані з використанням простого порогового або складного адаптивного керування.

Системи керування температурою можна класифікувати за різними ознаками, серед яких найпоширенішими є тип керування, складність алгоритму та принцип реалізації.

Залежно від принципів роботи, наявності зворотного зв’язку, алгоритму керування та типу реалізації, системи регулювання температури поділяються на кілька основних типів. Кожен із них має свої переваги, обмеження та характерні сфери застосування. У таблиці 1.1 наведено узагальнену класифікацію систем керування температурою, яка дозволяє порівняти їхні особливості та вибрати оптимальне рішення для конкретного завдання, зокрема в умовах побудови регулятора на основі Arduino UNO.

Таблиця 1.1 – Приклад класифікації систем керування температурою [1-4]

Тип системи	Характеристика	Приклад реалізації
Open-loop	без зворотного зв’язку	вентилятор на фіксованій швидкості
On/Off-керування	вмикається/вимикається при порозі	Arduino + реле
пропорційне (P)	ШІМ-регулювання за відхиленням	Arduino + PWM + вентилятор
ПІД-регулювання	точне, стабільне, багатопараметричне	Arduino + програмна PID-бібліотека
цифрова система	гнучка, програмована, модульна	Arduino UNO + DHT22 + MOSFET

Системи з відкритим контуром (open-loop systems) працюють без урахування фактичного значення температури. Наприклад, вентилятор постійно обертається з фіксованою швидкістю, незалежно від зміни температури. Такі системи прості, але ненадійні в умовах змінного середовища [1].

Порогові (on-off) системи включають або вимикають виконавчий механізм (наприклад, вентилятор), коли температура перевищує певний поріг. Простий приклад – включення вентилятора при >30° C [2].

У пропорційних системах (P) вихідна дія (швидкість вентилятора) змінюється пропорційно відхиленню температури від заданої. Зазвичай реалізується через ШІМ-сигнал, змінюючи скважність.

ПІ/ПІД-регулятори додатково враховують інтегральну та диференційну складову відхилення температури, забезпечуючи більш плавну стабілізацію без коливань. Реалізація таких регуляторів можлива навіть на Arduino, але потребує точного налаштування.

Цифрові системи реалізовані на базі мікроконтролерів, таких як Arduino, STM32, ESP32. Дозволяють зчитувати покази з датчиків, реалізовувати складну логіку керування, надсилати дані по мережі.

Особливої актуальності набувають системи, які автоматично регулюють швидкість вентилятора залежно від зміни температури середовища. Вони знаходять застосування в охолодженні електронних компонентів, серверів, блоків живлення, акумуляторних станцій, а також у побутових вентиляційних пристроях.

У побутових умовах СКТ використовуються в системах опалення, вентиляції та кондиціонування повітря (HVAC). Вони забезпечують комфортні температурні умови в приміщеннях та сприяють енергоефективності. Наприклад, у дослідженні [2] представлено систему автоматичного регулювання швидкості вентилятора на основі температури з використанням Arduino, яка дозволяє зменшити споживання енергії та підвищити комфорт користувачів (рис. 1.1).

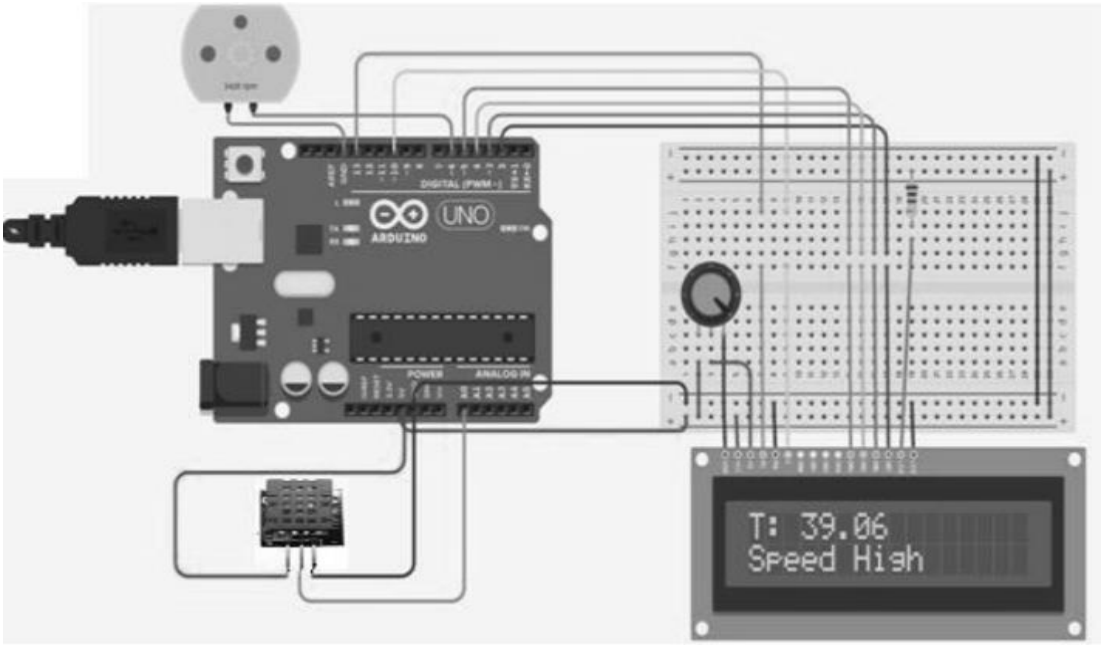

Рисунок 1.1 – Приклад структури системи з датчиком температури та вентилятором, керованим Arduino [2]

Сучасні кондиціонери, обігрівачі, осушувачі повітря і холодильники реалізують складні алгоритми регулювання температури, що адаптуються до навколишніх умов. У проєктах з «розумним» керуванням вентиляцією все частіше використовуються мікроконтролери з вбудованими температурними сенсорами або підключеними через шину I2C чи 1-Wire.

У рамках концепції «розумного будинку» СКТ інтегруються в мережу інтернету речей (IoT), дозволяючи автоматично регулювати клімат у приміщеннях, знижуючи витрати на електроенергію. Arduino, ESP32 та подібні платформи застосовуються в якості вузлів збору даних і виконавчих модулів.

У комп’ютерних системах та електронних пристроях СКТ забезпечують охолодження процесорів, графічних карт та інших компонентів. Наприклад, у дослідженні [3] описано систему автоматичного регулювання швидкості вентилятора на основі температури з використанням Arduino UNO, яка дозволяє підтримувати стабільну температуру в приміщенні та знижує енергоспоживання.

Більшість сучасних материнських плат та мікропроцесорів оснащені датчиками температури і мають можливість керувати обертами вентиляторів за допомогою PWM (імпульсно-широтною модуляцією), часто на базі вбудованих мікроконтролерів або спеціалізованих чипів.

У промислових процесах, таких як виробництво, зберігання та транспортування продукції, СКТ забезпечують підтримку необхідних температурних умов. Наприклад, у роботі [4] розглядається система контролю температури для ізотермічної коробки з використанням мікроконтролера, що дозволяє стабілізувати внутрішню температуру та забезпечити якість продукції.

На виробництві точне регулювання температури необхідне, наприклад, у процесах термічної обробки металів, сушіння, лиття пластмас, 3D-друку, виготовлення продуктів харчування. Тут використовуються як класичні ПІД-регулятори, так і гнучкі програмовані логічні контролери (PLC), а також відкриті платформи на зразок Arduino в дослідних і малих автоматизованих системах.

Отже, системи керування температурою є невід’ємною частиною сучасних технологій, забезпечуючи стабільну та ефективну роботу різноманітних пристроїв і систем.

1.2 Принципи побудови системи автоматичного регулювання температури

Основною метою системи автоматичного регулювання температури (САРТ) є підтримка заданої температури об’єкта або середовища шляхом

автоматичного регулювання параметрів, що впливають на тепловий баланс системи.

Типова САРТ складається з таких основних компонентів:

– датчик температури – вимірює поточну температуру об’єкта або середовища. У проєктах на базі Arduino часто використовуються датчики типу LM35, DHT11 або DHT22, які забезпечують аналоговий або цифровий вихідний сигнал, пропорційний температурі [5];

– контролер (регулятор) – обробляє сигнал від датчика та визначає необхідну дію для підтримки заданої температури. У нашому випадку, функцію контролера виконує мікроконтролер Arduino UNO, який реалізує алгоритм керування на основі отриманих даних;

– виконавчий механізм – елемент, який змінює параметри системи для досягнення бажаної температури. Це може бути вентилятор, нагрівач або інший пристрій. Керування виконавчим механізмом здійснюється через транзистор або реле, які дозволяють змінювати швидкість обертання вентилятора або вмикати/вимикати його [6];

– зворотний зв’язок – механізм, який забезпечує порівняння поточної температури з заданою та коригування дій системи відповідно до відхилення.

Зворотний зв’язок є ключовим елементом САРТ, що забезпечує адаптацію системи до змін зовнішніх умов. У контексті регулятора швидкості вентилятора на основі температури, зворотний зв’язок реалізується шляхом постійного моніторингу температури та коригування швидкості вентилятора відповідно до відхилення від заданого значення. Це дозволяє системі підтримувати стабільну температуру, компенсуючи вплив зовнішніх факторів [7].

На рисунку 1.2 зображено типову блок-схему регулятора швидкості вентилятора на основі температури, реалізованого з використанням мікроконтролера Arduino UNO та цифрового температурного сенсора DHT11. Система включає в себе основні функціональні модулі: сенсор температури, мікроконтролер, виконавчий елемент (транзистор або реле) та вентилятор.

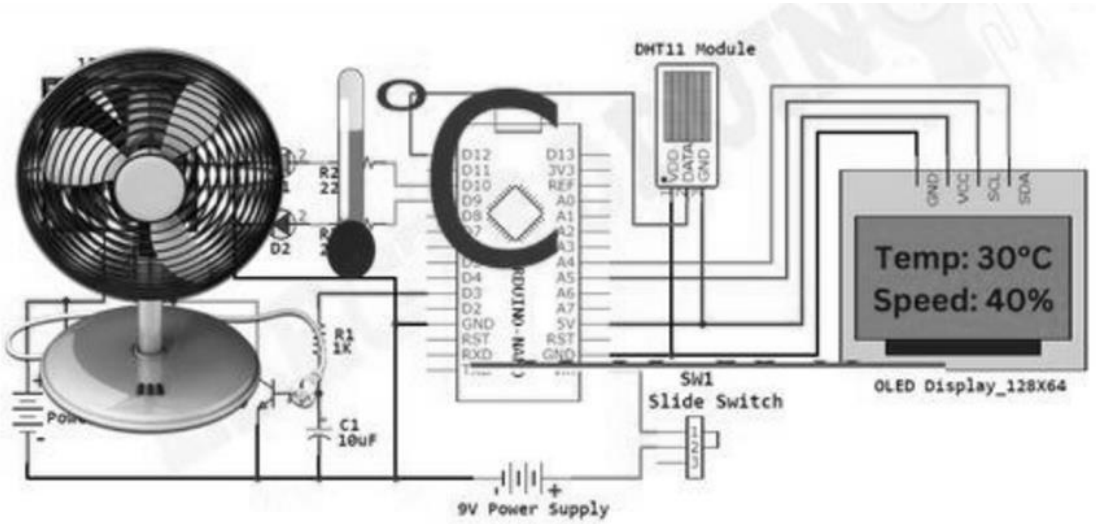

Рисунок 1.2 – Регулятор швидкості вентилятора на основі температури за допомогою Arduino та DHT11 [7]

Застосування мікроконтролера Arduino спрощує реалізацію керуючого алгоритму, а сенсор DHT11 забезпечує необхідну точність вимірювання температури для базових задач. Така схема є гнучкою в налаштуванні та може бути легко масштабована або адаптована під інші типи датчиків і виконавчих механізмів.

Алгоритм керування є основним інструментом, що визначає реакцію системи на зміну температури. У практиці розробки регуляторів швидкості вентилятора на основі Arduino UNO найчастіше використовуються порогове, пропорційне та ПІД-регулювання. Вибір конкретного алгоритму залежить від вимог до точності, динамічності та стабільності системи.

ПІД-алгоритм забезпечує найточніше регулювання без значного перерегулювання або коливань. На Arduino UNO реалізація ПІД-алгоритму можлива завдяки вбудованим бібліотекам, таким як PID_v1, але вимагає налаштування коефіцієнтів та більшої обчислювальної уваги [8]. ПІД-регуляція є ідеальним варіантом для систем з інерційними процесами або високими вимогами до стабільності.

Пропорційно-інтегрально-диференціальне керування (ПІД) поєднує в собі три компоненти: пропорційна складова (P) забезпечує реакцію на поточне

відхилення, інтегральна (I) враховує накопичене відхилення в часі та диференціальна (D) реагує на швидкість зміни температури.

Порогове керування (On/Off control) може викликати часті цикли вмикання/вимикання, що знижує енергоефективність і ресурс компонентів [9]. Метод P-contro рекомендований для систем, де потрібна гнучкість без складних обчислень.

На рисунку 1.3 наведено порівняння трьох основних алгоритмів керування, які використовуються в системах автоматичного регулювання температури, зокрема у проектах з Arduino UNO.

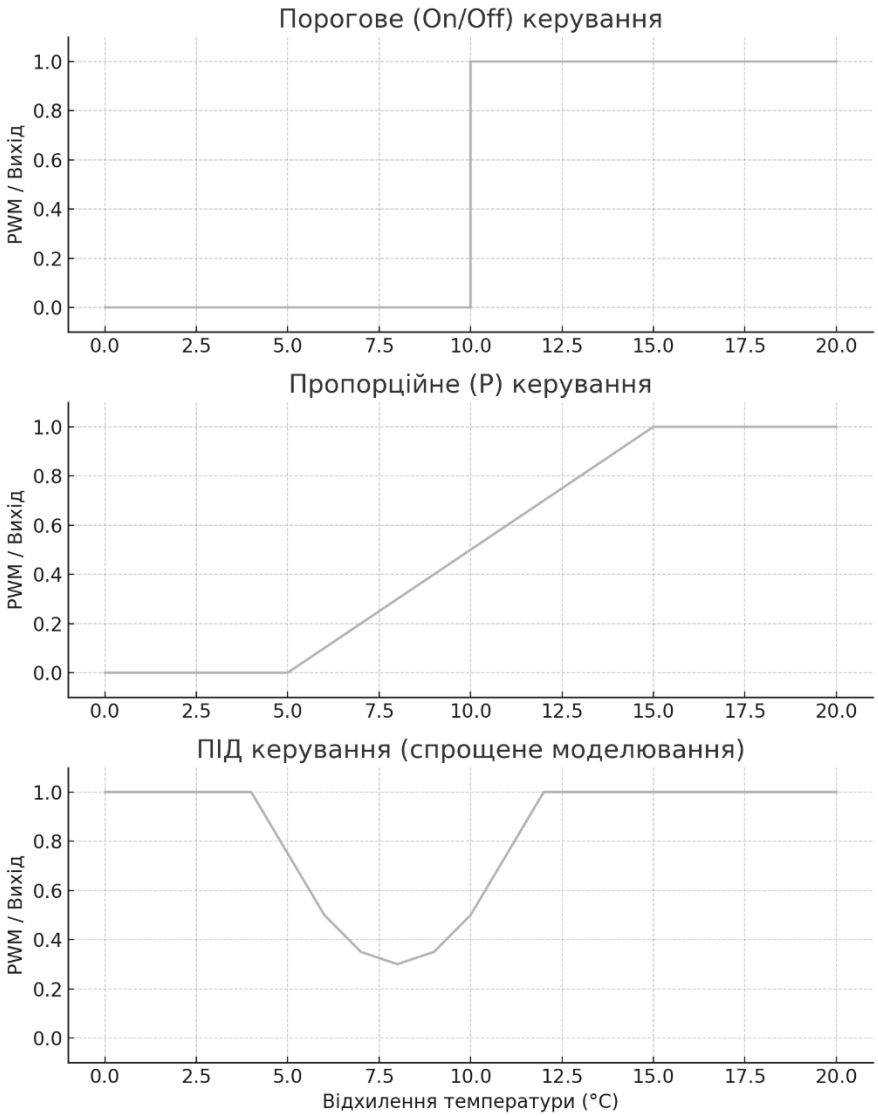

Рисунок 1.3 – Залежність керуючого сигналу від температурного відхилення [6, 8, 9]

Графіки ілюструють залежність керуючого сигналу (наприклад, ширини PWM-імпульсів для вентилятора) від температурного відхилення. Порогове керування представлено як двостанова логіка з фіксованим порогом, пропорційне – як лінійна реакція на зміну температури, а ПІД-регулювання – як адаптивна реакція, що враховує поточне, інтегральне та диференціальне відхилення. Такий візуальний аналіз дозволяє краще зрозуміти вплив кожного з алгоритмів на плавність та ефективність температурного регулювання.

Аналіз графіків дає змогу зробити висновок, що вибір алгоритму керування суттєво впливає на поведінку системи регуляції температури. Порогове керування є найпростішим, однак може спричиняти надмірне навантаження через часті перемикання. Пропорційне керування забезпечує більш плавну реакцію системи, але може залишати залишкове відхилення. Найвищу точність та стабільність демонструє ПІД-регулятор, хоча його реалізація вимагає складніших розрахунків та налаштування.

Отже, системи автоматичного регулювання температури є важливими елементами сучасних електронних пристроїв, забезпечуючи стабільну та ефективну роботу шляхом підтримки оптимальних температурних умов. Реалізація таких систем на базі Arduino UNO дозволяє створювати доступні та гнучкі рішення для різноманітних застосувань, включаючи регулятори швидкості вентилятора на основі температури.

1.3 Вимоги до контролера швидкості вентилятора

Контролер у системі регулювання температури відіграє центральну роль – він забезпечує обробку вхідних даних з сенсора, виконання алгоритму керування та формування відповідного сигналу для приводу вентилятора. Залежно від складності системи, вимоги до контролера можуть варіюватися від мінімального керування в режимі вмикання/вимикання до складних адаптивних моделей із застосуванням ПІД-регуляції та інтеграції з іншими модулями.

Контролер має бути здатним виконувати обчислення у реальному часі, зокрема, опрацювання значень температури з певною частотою дискретизації та реалізація відповідного алгоритму керування. Arduino UNO на базі мікроконтролера ATmega328P забезпечує тактову частоту 16 МГц, що є достатнім для реалізації простих порогових, пропорційних або ПІД-алгоритмів для керування одним або кількома вентиляторами [10].

Для підключення сенсорів та виконавчих елементів необхідно мати щонайменше один аналоговий вхід (наприклад, A0) для зчитування температурного сенсора (типу LM35), або цифровий пін – для DHT11/DHT22, а також один цифровий вихід з підтримкою ШІМ (наприклад, D3, D5, D6) – для передачі сигналу керування на вентилятор через транзистор або драйвер.

Оскільки зміна швидкості вентилятора в більшості реалізацій здійснюється через ШІМ-сигнал, контролер повинен мати відповідні пін-конектори з апаратною підтримкою генерації PWM-сигналу. Arduino UNO підтримує до 6 таких виходів, що є достатнім для кількох незалежних каналів керування [6].

Arduino IDE та відповідна екосистема бібліотек (зокрема DHT, PID_v1, Servo) забезпечують швидку розробку і тестування програмного коду. Крім того, плата легко інтегрується з модулями Wi-Fi (ESP8266), дисплеями, Bluetooth-модулями та іншими периферійними пристроями, що розширює функціональні можливості системи [3].

У таблиці 1.2 наведено узагальнені технічні вимоги до контролера, які забезпечують стабільну роботу системи, точне регулювання температури, а також можливість подальшої модифікації та масштабування.

Таблиця 1.2 – Додаткові технічні вимоги [7]

Параметр	Вимоги
тактова частота	≥ 16 МГц
підтримка ШІМ	≥ 1 апаратний PWM-вихід
аналогові/цифрові входи	≥ 1 цифровий (для DHT11/DHT22) або аналоговий (LM35)
ОЗП/ПЗУ	≥ 2 КБ SRAM / ≥ 32 КБ Flash
простота прошивки	підтримка USB-завантаження та Arduino IDE
вартість	бюджетне рішення (< 10 USD)
розширюваність	підтримка модулів UART/I²C/SPI

Отже, контролер у системі автоматичного регулювання температури повинен відповідати ряду технічних та функціональних вимог, зокрема мати достатню обчислювальну потужність, підтримувати генерацію ШІМ-сигналу, забезпечувати сумісність із сенсорами температури та мати розширювану апаратну архітектуру. Плата Arduino UNO повністю задовольняє ці вимоги, що робить її доцільним вибором для реалізації регулятора швидкості вентилятора. Завдяки відкритій екосистемі, зручному середовищу програмування та широкій підтримці модулів Arduino UNO забезпечує надійну основу для побудови гнучкої й ефективної системи керування температурою.

1.4 Компоненти та методи керування в системах контролю температури

1.4.1 Порівняльний аналіз мікроконтролерів для побудови систем керування

У виборі апаратної платформи для побудови системи автоматичного регулювання температури важливо враховувати низку технічних характеристик мікроконтролера.

Arduino UNO – класична платформа на базі 8-бітного мікроконтролера ATmega328P. Підтримує 6 аналогових входів, 6 PWM-виходів і стандартні інтерфейси UART, I2C, SPI. Arduino UNO має низький поріг входу, що робить її ідеальною для навчальних проєктів. Обмеженням є невеликий обсяг оперативної пам’яті 2 КБ та обмежена продуктивність [10].

ESP32 – потужний 32-бітний мікроконтролер з двома ядрами, частотою до 240 МГц, великим обсягом пам’яті (до 520 КБ SRAM) та вбудованою підтримкою Wi-Fi і Bluetooth. Підтримує велику кількість PWM-виходів і аналогових входів. Підходить для реалізації розподілених і віддалено керованих температурних систем [11].

STM32F103C8T6 – представник мікроконтролерів сімейства STM32 на базі ядра ARM Cortex-M3. Має більшу обчислювальну здатність, ніж Arduino UNO,

та ширші можливості керування, проте вимагає більше зусиль для програмування і налаштування середовища розробки (наприклад, STM32CubeIDE, PlatformIO) [12].

Raspberry Pi Pico – мікроконтролер від Raspberry Pi Foundation на базі власного чипа RP2040. Підтримує 16 апаратних PWM-виходів і працює на частоті до 133 МГц. Підтримує мову програмування C++ та MicroPython. Є перспективною платформою для складніших реалізацій температурних регуляторів [13].

У таблиці 1.3 узагальнено основні характеристики кожного з розглянутих мікроконтролерів, що дозволяє порівняти їх з точки зору застосування в температурно-чутливих автоматизованих системах.

Таблиця 1.3 – Порівняння мікроконтролерів для систем керування температурою [10-13]

Характеристика	Arduino UNO	ESP32	STM32F103C8T6	Raspberry Pi Pico
мікроконтролер	ATmega328P	Tensilica Xtensa LX6	ARM Cortex-M3	RP2040
процесор	8-біт AVR	32-біт	32-біт	2× ARM Cortex-M0+
тактова частота	16 МГц	240 МГц	72 МГц	133 МГц
ОЗП (SRAM)	2 КБ	520 КБ	20 КБ	264 КБ
ПЗП (Flash)	32 КБ	4 МБ	64 КБ	2 МБ
аналогові входи	6	18	10	3
PWM-виходи	6	16	12	16
інтерфейси зв’язку	UART, I2C, SPI	UART, I2C, SPI, Wi-Fi	UART, I2C, SPI	UART, I2C, SPI
підтримка Wi-Fi/Bluetooth	ні	так	ні	ні
складність програмування	низька	середня	висока	середня
вартість (USD)	~7	~5-8	~3-5	~4-6

Порівняння характеристик показує, що всі чотири платформи можуть бути використані для побудови систем керування температурою. Проте саме Arduino UNO є найбільш доцільним вибором завдяки простоті використання, доступності, добре документованій екосистемі та підтримці базових функцій (аналоговий ввід, PWM-вивід, серійна передача даних).

1.4.2 Температурні сенсори для автоматизованих систем

Температурні сенсори є невід'ємною частиною систем моніторингу та керування, зокрема у вентиляційних, охолоджувальних, кліматичних і безпекових автоматизованих системах. Вони забезпечують збирання ключових даних для прийняття рішень у реальному часі. Для мікроконтролерних систем на базі Arduino найбільш поширеними є аналогові та цифрові сенсори, які відрізняються точністю, температурним діапазоном, інтерфейсом з'єднання та вартістю.

Розглянемо чотири поширені температурні сенсори, які активно використовуються в проєктах:

– DHT11 – бюджетний цифровий сенсор температури і вологості (рис. 1.4). Має простий 1-провідний інтерфейс, проте обмежений діапазон температур (0…50° С) і відносно низька точність (±2° С) [14];

– DHT22 (AM2302) – більш точний аналог DHT11, з ширшим діапазоном вимірювання (-40…80° С) і точністю ±0,5° С (рис. 1.5). Також має цифровий вихід і простий протокол взаємодії, сумісний з Arduino [15] ;

– LM35 – аналоговий сенсор, який видає напругу пропорційно температурі 10 мВ/° С (рис. 1.6). Має хорошу точність (±0,5° С), але потребує стабільного живлення і наявності АЦП на контролері [16];

– DS18B20 – цифровий сенсор з високою точністю та широким діапазоном температур (-55…125° С), підтримує протокол 1-Wire (рис. 1.7), що дозволяє підключення кількох сенсорів до одного піну Arduino [17].

Рисунок 1.4 – Цифровий сенсор температури і вологості DHT11 [14]

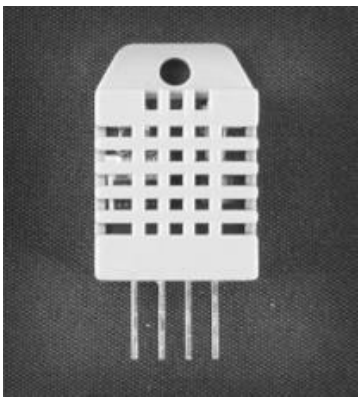

Рисунок 1.5 – Цифровий вихідний датчик відносної вологості та температури / модуль AM2303 [15]

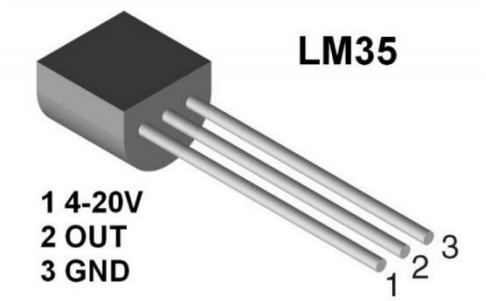

Рисунок 1.6 – Датчик температури LM35 [16]

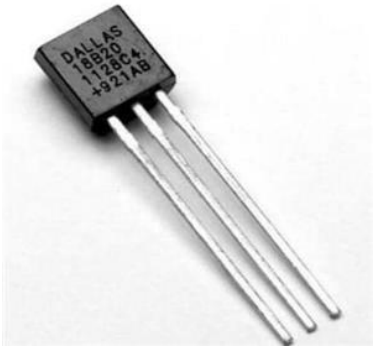

Рисунок 1.7 – Цифровий сенсор DS18B20 [17]

Для порівняння наведених температурних сенсорів складено таблицю 1.4, яка дозволяє оцінити їхню доцільність використання в конкретних умовах автоматизованої системи керування температурою.

Таблиця 1.4 – Основні параметри цифрових і аналогових температурних сенсорів [14-17]

Характеристика	DHT11	DHT22 (AM2302)	LM35	DS18B20
тип сенсора	цифровий	цифровий	аналоговий	цифровий
діапазон температур	0…50° C	-40…80° C	0…100° C	-55…125° C
точність	±2° C	±0,5° C	±0,5° C	±0,5° C
інтерфейс	1-провідний	1-провідний	Аналоговий	1-Wire
тип виходу	цифровий	цифровий	аналоговий	цифровий
живлення	3,3-5 В	3,3-6 В	4-30 В	3,0-5,5 В
середня вартість (USD)	~1,5	~3-5	~1-2	~2-3

Аналіз таблиці 1.4 дозволяє зробити висновок, що найбільш універсальним сенсором для системи на базі Arduino є DHT22 завдяки хорошому балансу між точністю, діапазоном та простотою використання. Для задач, де важливий широкий температурний діапазон і можливість мережевого з’єднання кількох сенсорів, доцільно застосовувати DS18B20. У свою чергу, LM35 буде корисним у випадках, коли використовується аналоговий вхід контролера та необхідна мінімальна вартість. DHT11 варто застосовувати лише у навчальних або демонстраційних проєктах, де висока точність не є критичною.

РОЗДІЛ 2
АПАРАТНА РЕАЛІЗАЦІЯ КОНТРОЛЕРА ТЕМПЕРАТУРОЗАЛЕЖНОГО КЕРУВАННЯ ВЕНТИЛЯТОРОМ

2.1 Вибір апаратних компонентів

2.1.1 Arduino UNO: технічні характеристики

Arduino UNO (рис. 2.1) є базовим елементом обраної апаратної платформи для реалізації системи температурозалежного керування вентилятором. У межах цієї підсистеми плата виконує роль центрального керувального блоку, який забезпечує прийом даних із сенсора температури, обробку сигналу відповідно до заданого алгоритму та генерацію керувального ШІМ-сигналу для регулювання обертів вентилятора.

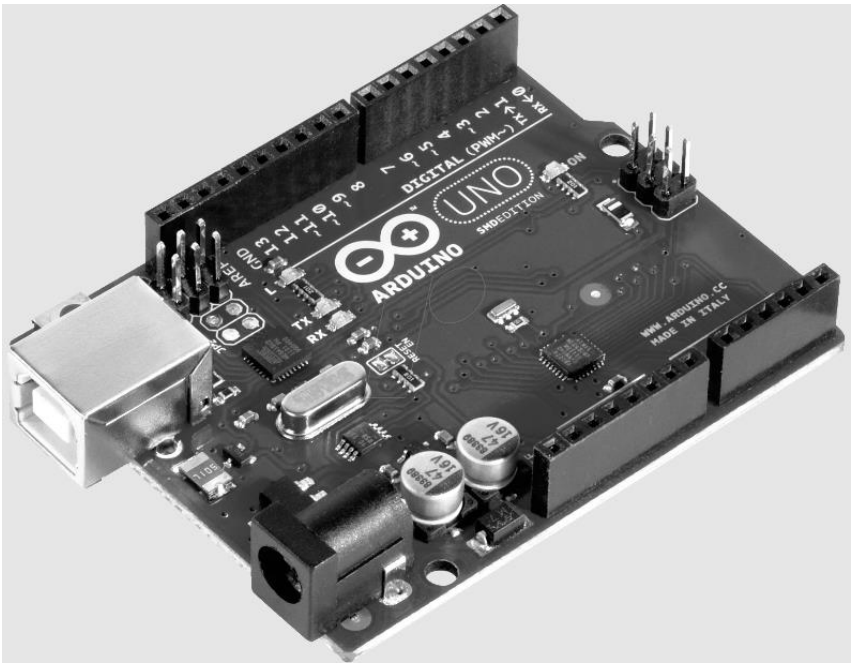

Рисунок 2.1 – Плата мікроконтролера на базі ATmega328P [18]

Простота інтеграції в макетну або друковану плату, наявність стандартного USB-інтерфейсу для прошивки та живлення системи додатково спрощує процес апаратної реалізації.

У дослідженні використано Arduino UNO як стабільну, перевірену платформу, яка підтримує безперебійну роботу за умови живлення 5 В через

зовнішнє джерело або через порт USB. Крім того, плата має компактні габарити, що дозволяє легко інтегрувати її в корпус системи охолодження чи керування мікрокліматом.

Під час практичної реалізації особливу увагу було приділено розміщенню елементів на макетній платі (breadboard), розведенню проводів живлення та керуючих сигналів, а також надійності з'єднань із зовнішніми компонентами, такими як транзисторний ключ, сенсор температури та вентилятор постійного струму. Використання Arduino UNO дозволило зосередитися на функціональності програмної логіки, мінімізуючи ризики апаратних збоїв.

2.1.2 Температурний датчик: особливості та точність

У системі температурозалежного керування вентилятором основним джерелом вхідних даних є температурний сенсор. Для нашого дослідження обрано цифровий датчик DHT22 (також відомий як AM2302) – компактний модуль, який забезпечує надійні вимірювання температури й вологості навколишнього середовища в широкому діапазоні [15]. Його застосування у поєднанні з Arduino UNO є виправданим завдяки простому протоколу обміну, стабільній роботі та достатній точності.

У таблиці 2.1 наведено ключові параметри сенсора DHT22, що визначають його придатність для використання в автоматизованих мікроконтролерних системах.

Таблиця 2.1 – Характеристики температурного сенсора DHT22 [15]

Характеристика	Значення
діапазон вимірювання температури	від –40° С до +80° С
точність температурного вимірювання	±0,5° С при 25° С
роздільна здатність	0,1° С
інтерфейс	1-провідний цифровий
інтервал опитування	не менше 2 секунд
живлення	3,3-6,0 В
тип вихідного сигналу	цифровий імпульсний з фіксованим форматом даних

Сенсор має вбудований АЦП і термістор, який вимірює температуру, а результат передається у вигляді цифрової послідовності. Це зменшує похибки,

пов’язані з аналоговими перешкодами та спрощує схему підключення – достатньо одного цифрового піну Arduino.

Порівняно з простішим сенсором DHT11, DHT22 має кращу точність, стабільність і ширший діапазон вимірювання. Саме це стало вирішальним фактором при виборі в рамках даного дослідження. У прикладній системі точність ±0,5° С є достатньою для забезпечення ефективного температурного керування вентилятором і запобігання перегріву компонентів.

DHT22 підключається до Arduino UNO через один цифровий пін (наприклад, D2), а також потребує резистора підтягування (10 кОм) між лінією даних і Vcc для забезпечення стабільного сигналу. На рисунку 2.1 представлено фізичне розташування компонентів і правильне з’єднання контактів живлення.

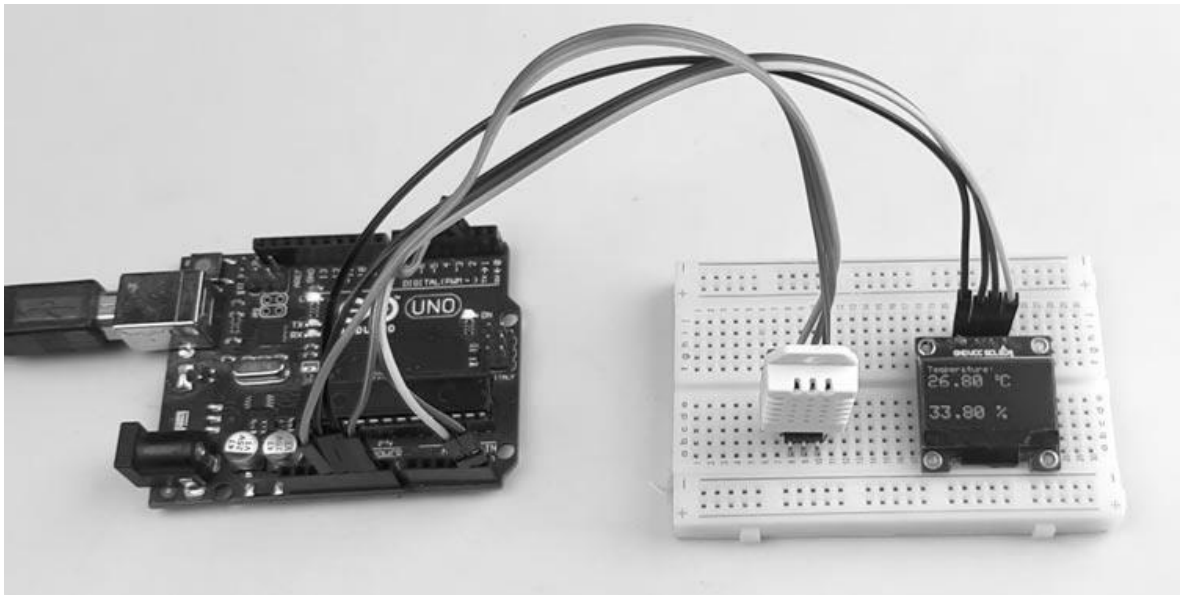

Рисунок 2.1 – Підключення температурного сенсора DHT22
до плати Arduino UNO [19]

Температурний сенсор DHT22 є оптимальним вибором для невисокої вартості систем керування температурою на базі Arduino. Його точність і цифровий інтерфейс роблять інтеграцію з контролером простою та надійною. У поєднанні з ШІМ-регулюванням швидкості вентилятора сенсор дозволяє ефективно реалізувати автоматизовану систему мікроклімату для невеликих електронних або побутових пристроїв.

2.1.3 Вентилятор постійного струму: параметри, тип живлення

Вентилятор у системі температурозалежного керування виконує функцію виконавчого елемента, завданням якого є відведення надлишкового тепла для стабілізації температурного режиму. В межах даного дослідження використано вентилятор постійного струму DC fan (рис. 2.2), що характеризується простотою підключення, низькою напругою живлення, компактними розмірами та стабільною роботою в умовах керування за допомогою широтно-імпульсної модуляції (PWM).

Рисунок 2.2 – Вентилятор охолодження DC fan [20]

У таблиці 2.2 наведено основні параметри вентилятора постійного струму, які визначають його сумісність із мікроконтролером Arduino UNO, можливість керування за допомогою ШІМ-сигналу, а також придатність до використання в умовах обмеженого енергоспоживання.

Таблиця 2.2 – Основні параметри вентилятора DC fan [20]

Параметр	Значення
тип	вентилятор постійного струму (DC 5V)
напруга живлення	5 В (підходить для живлення від Arduino)
сила струму	~100-200 мА (залежно від моделі)
швидкість обертання	3000-6000 об/хв
діаметр крильчатки	40-60 мм
тип керування	PWM або через комутацію (On/Off)
кількість контактів	2 або 3 (VCC, GND, PWM- або Tach)

Вентилятори на 5 В можуть бути живлені безпосередньо від лінії 5 V на платі Arduino, проте з міркувань енергоефективності та уникнення

перевантаження стабілізатора доцільно використовувати окреме зовнішнє джерело живлення з транзисторним ключем. При використанні вентилятора з номінальною напругою 12 В потрібно обов’язково застосовувати окремий стабілізатор або живлення з окремого джерела.

Для зміни швидкості вентилятора використовується або апаратна підтримка PWM (якщо вентилятор підтримує керування ШІМ), або програмна модуляція на Arduino через транзистор, що комутує живлення вентилятора. Зокрема, в роботі реалізовано схему, в якій вентилятор керується сигналом з PWM-виводу Arduino, який через n-канальний транзистор подає живлення на двигун.

Вентилятор постійного струму є надійним і економічно ефективним рішенням для реалізації виконавчого елемента в системі температурного регулювання. Його сумісність з Arduino, можливість PWM-керування та низьке споживання струму роблять його ідеальним вибором для використання в компактних автоматизованих системах охолодження.

2.1.4 Транзистор або MOSFET для керування вентилятором та додаткові компоненти

У мікроконтролерних системах типу Arduino UNO цифрові виходи не можуть забезпечити достатню силу струму для прямого живлення виконавчих пристроїв, таких як вентилятори. Тому для комутації живлення та реалізації керування за допомогою ШІМ використовується електронний ключ, зазвичай у вигляді біполярного транзистора або MOSFET. Окрім цього, система включає низку допоміжних компонентів – резистори, стабілізатори напруги, макетні плати, з’єднувальні дроти.

У рамках дослідження обрано n-канальний MOSFET IRF540N, який має низький опір каналу та може комутувати струми до 3 А при напрузі 5-12 В. Його перевага полягає у високій швидкодії, енергоефективності та можливості прямого керування від логічного рівня Arduino (5 В).

Для простіших застосувань також можливе використання біполярного транзистора NPN типу TIP120, що має вбудований захисний діод і дозволяє

комутувати навантаження з струмом до 5 А. Проте він має вищу насичувану напругу колектора, що знижує ефективність порівняно з MOSFET.

Схема підключення (рис. 2.3) включає:

– затвор (G) MOSFET підключається через обмежувальний резистор (наприклад, 220 Ом) до PWM-виходу Arduino;

– сток (D) з'єднаний із «–» лінією вентилятора;

– стік (S) підключений до GND;

– резистор 10 кОм між затвором і GND для запобігання «зависанню» MOSFET у відкритому стані при вимкненому Arduino.

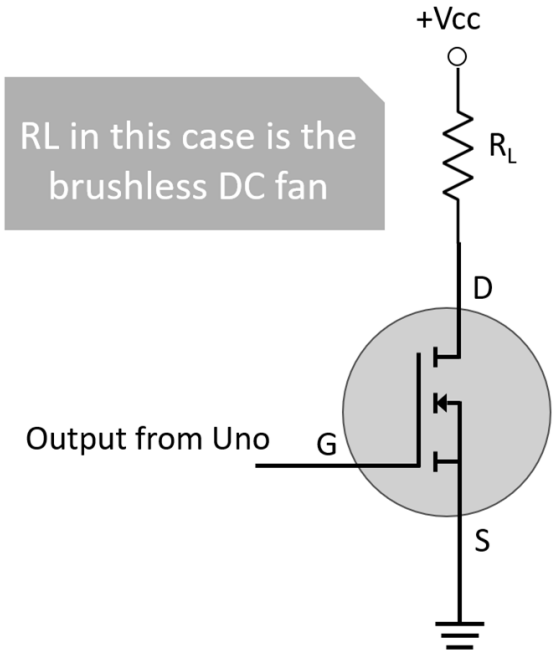

Рисунок 2.3 – Схематичне підключення N-канального MOSFET [21]

Використання MOSFET-транзистора як ключового елементу у керуванні вентилятором забезпечує надійність, енергоефективність і високу швидкодію схеми. У поєднанні з допоміжними компонентами, такими як стабілізатори, резистори та діоди, система стає стабільною й безпечною для тривалої роботи. Такий підхід є стандартною практикою в автоматизованих вбудованих системах охолодження.

2.2 Програмна реалізація температурозалежного керування

Програмна частина системи температурозалежного керування вентилятором реалізована на платформі Arduino UNO з використанням середовища розробки Arduino IDE. Основне завдання коду – періодичне зчитування температури з цифрового сенсора DHT22 та формування керуючого PWM-сигналу, що регулює швидкість вентилятора залежно від отриманих даних.

У першій фазі роботи програми виконується ініціалізація температурного сенсора та PWM-виходу мікроконтролера. Для цього використовується вбудована бібліотека DHT.h, яка забезпечує інтерфейс для зчитування температурних значень з сенсора DHT22. В межах функції setup() відбувається виклик методу dht.begin(), що активує датчик, а також конфігурація піну, призначеного для широтно-імпульсної модуляції (FAN_PIN), як виходу. Паралельно, для відлагодження та моніторингу вмикається послідовний порт (Serial), що дозволяє виводити дані на комп’ютер у реальному часі.

Після запуску ініціалізації, в головному циклі loop() реалізується процедура зчитування температурного значення з сенсора. Метод readTemperature() повертає числове значення у градусах Цельсія, яке зберігається у змінній типу float. При збої або недоступності сенсора, функція повертає значення NaN, що додатково перевіряється умовним оператором isnan(). Цей етап критично важливий для стабільної роботи системи, оскільки некоректні або відсутні дані можуть призвести до небажаної активації вентилятора або його повного вимкнення.

На основі отриманої температури система виконує перетворення значення у відповідний рівень PWM-сигналу, який визначатиме швидкість обертання вентилятора. Тут використовується функція map(), що лінійно масштабує значення температури у межах, наприклад, 25-45° С до стандартного діапазону PWM (0-255). Таким чином, при температурі нижче 25° С вентилятор практично не активується, а при перевищенні 45° С працює на максимальній швидкості.

Функція constrain() додатково запобігає виходу за межі допустимого діапазону ШІМ.

Сформоване значення PWM передається на цифровий вихід Arduino за допомогою функції analogWrite(), що безпосередньо формує широтно-імпульсний сигнал заданої потужності. Цей сигнал подається на затвор транзистора або MOSFET, який, у свою чергу, комутує вентилятор. В результаті, змінюється ефективна напруга на вентиляторі, а отже – і швидкість його обертання. Такий підхід дозволяє забезпечити не тільки плавне регулювання, але й суттєве зниження енергоспоживання та зменшення механічного зношення вентилятора.

Завершальним етапом програми є затримка між циклами зчитування температури, реалізована за допомогою функції delay(2000). Це дозволяє уникнути надмірного опитування сенсора, який фізично не підтримує інтервали зчитування менші за 2 секунди. Така затримка не впливає на загальну швидкодію системи, оскільки термоінерційні властивості об’єкта дозволяють працювати з інтервалами в кілька секунд. Завдяки цьому система працює стабільно, із заданою періодичністю оновлює значення температури та відповідно коригує швидкість вентилятора в режимі реального часу.

Програмна реалізація системи базується на лаконічному та ефективному алгоритмі зчитування температури та керування швидкістю вентилятора. Застосування функцій map() і analogWrite() дозволяє створити плавну ШІМ-модуляцію, яка забезпечує адаптивне охолодження в умовах змінного теплового навантаження.

2.3 Можливості інтеграції системи в реальні умови експлуатації

Розроблена система температурозалежного керування вентилятором на базі Arduino UNO має потенціал широкого практичного застосування завдяки своїй модульності, гнучкості та енергоефективності. Завдяки відкритому програмному забезпеченню, сумісності з різними типами сенсорів та простоті

підключення, така система легко адаптується до потреб конкретних об’єктів експлуатації.

Щоб продемонструвати практичну цінність розробленої системи, у таблиці 2.3 узагальнено основні напрями її можливого застосування в реальних умовах. Вказано як типові сфери експлуатації, так і ключові особливості інтеграції, що враховують технічні й функціональні характеристики системи. Такий аналіз дозволяє оцінити універсальність рішення та визначити перспективи його масштабування залежно від завдань користувача.

Таблиця 2.3 – Можливості інтеграції температурозалежної системи керування вентилятором [22, 23]

Сфера застосування	Особливості інтеграції
охолодження комп’ютерних систем	регулювання обертів вентилятора відповідно до температури процесора чи блоку живлення
розумні системи вентиляції	автоматичне керування мікрокліматом з можливістю хмарного моніторингу
автономні енергетичні пристрої	живлення від акумулятора, компактність, економія енергії
серверні шафи та телеком-обладнання	багатозональне керування для рівномірного розподілу повітряного потоку
побутові витяжки та теплиці	температурне керування витяжним вентилятором на основі внутрішньої температури

Однією з основних сфер інтеграції є охолодження мікроконтролерних або комп’ютерних систем, де необхідно уникнути перегріву процесора, стабілізаторів напруги або силових транзисторів. Система може бути встановлена у корпус пристрою або серверного вузла, з можливістю регулювання швидкості вентилятора відповідно до зміни температури внутрішнього середовища. Завдяки підтримці PWM-регулювання та температурних сенсорів, забезпечується низький рівень шуму і споживання енергії в умовах невисокого тепловиділення.

Іншою перспективною галуззю застосування є розумні системи кліматконтролю, зокрема у побутових умовах: у вентиляційних модулях, витяжках або невеликих теплицях. У таких випадках система може адаптуватися до умов зовнішнього середовища, а з використанням додаткових модулів (наприклад,

ESP8266 або ESP32) бути підключеною до хмарної інфраструктури для віддаленого моніторингу та керування.

Система також може бути інтегрована у переносні або автономні пристрої, такі як акумуляторні кейси, сонячні інвертори, пристрої з обмеженим доступом до енергомереж. У таких випадках перевагою є низьке енергоспоживання та можливість адаптації до живлення від батарей. Крім того, Arduino UNO може бути замінено на менш енерговитратні версії (наприклад, Arduino Pro Mini або STM32), що дозволяє зменшити габарити та підвищити автономність системи.

Завдяки відкритому коду й універсальності апаратних компонентів, систему легко масштабувати. Наприклад, при встановленні кількох температурних датчиків у різних точках корпусу можна реалізувати багатозональне керування вентиляторами. Це особливо корисно в серверних приміщеннях, шафах з обладнанням або промислових електронних системах, де різні зони потребують різної інтенсивності охолодження.

Інтеграція розробленої системи в реальні умови експлуатації є цілком здійсненною без значних доопрацювань. Модульний підхід, використання стандартних компонентів, підтримка розширення функціональності та стабільна робота в умовах змінного температурного навантаження забезпечують широкі перспективи для впровадження у промислових, побутових та освітніх системах керування.

РОЗДІЛ 3
ПРАКТИЧНА РЕАЛІЗАЦІЯ КОНТРОЛЕРА ШВИДКОСТІ ВЕНТИЛЯТОРА З ВИКОРИСТАННЯМ ARDUINO UNO

3.1 Комплектуючі системи контролю швидкості вентилятора

Для реалізації системи контролю швидкості вентилятора було вибрано ряд комплектуючих:

– мікроконтролер (рис. 3.1), що є «мозком» ророблюваної системи;

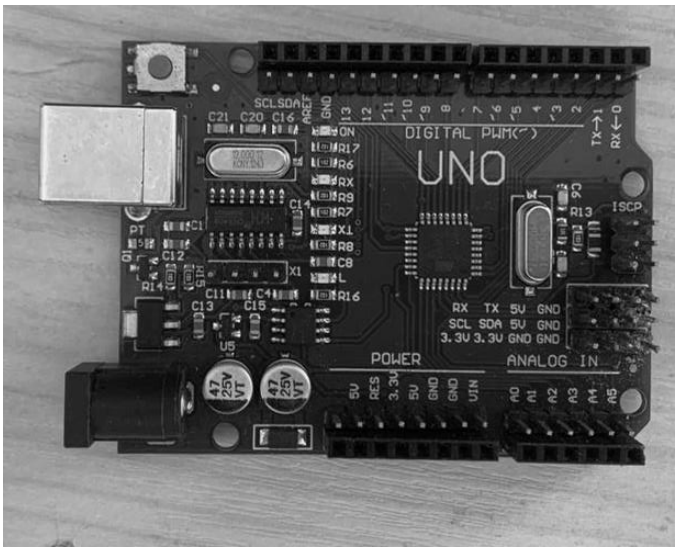

Рисунок 3.1 – Мікроконтролер Arduino UNO

– 16-канальний РК-дисплей (рис. 3.2), для відображення температури;

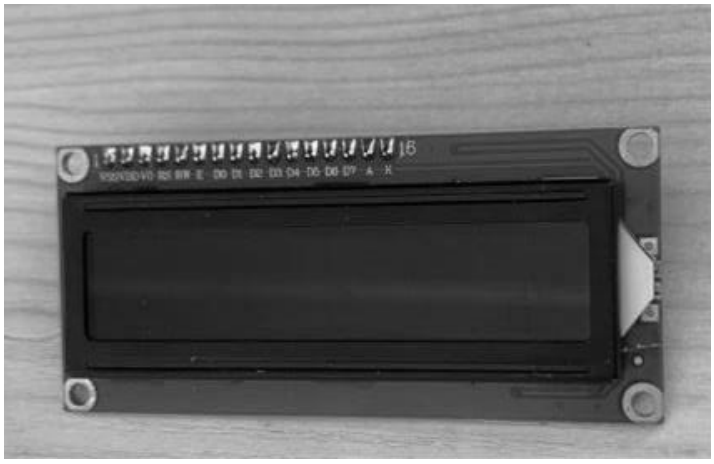

Рисунок 3.2 – 16-канальний РК-дисплей

– драйвер двигуна моделі L298N (рис. 3.3), для підключення вентилятора;

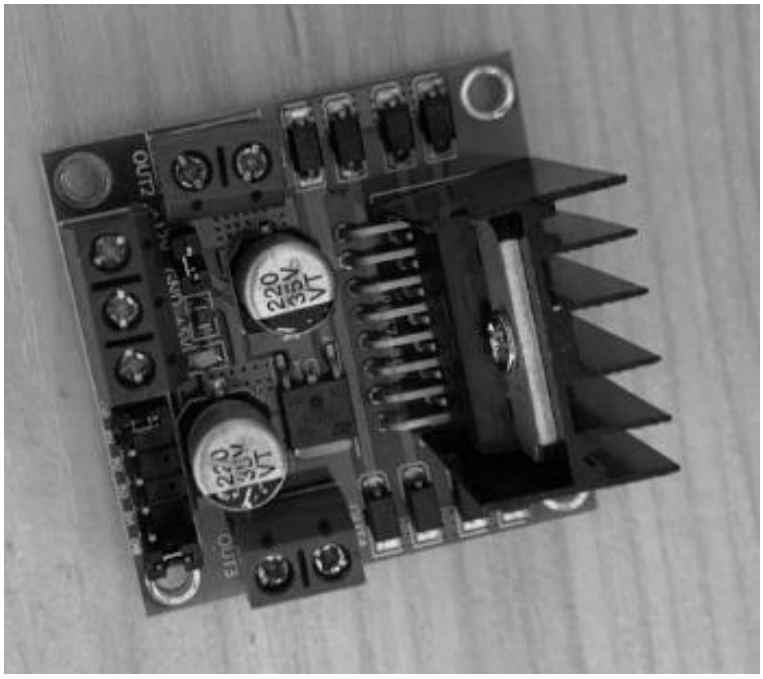

Рисунок 3.3 – Драйвер двигуна L298N

– вентилятор (рис. 3.4);

Рисунок 3.4 – Вентилятор

– датчиком температури (рис. 3.5)

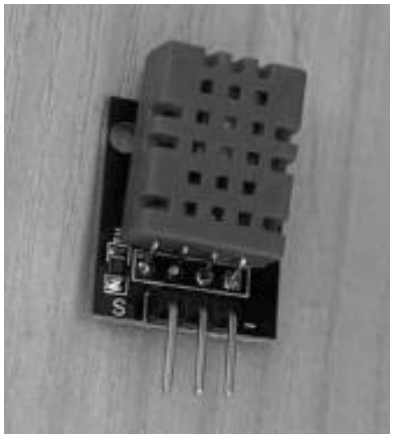

Рисунок 3.5 – Датчик температури

– макетна плата (рис. 3.6) для з’єднання всіх комплектуючих;

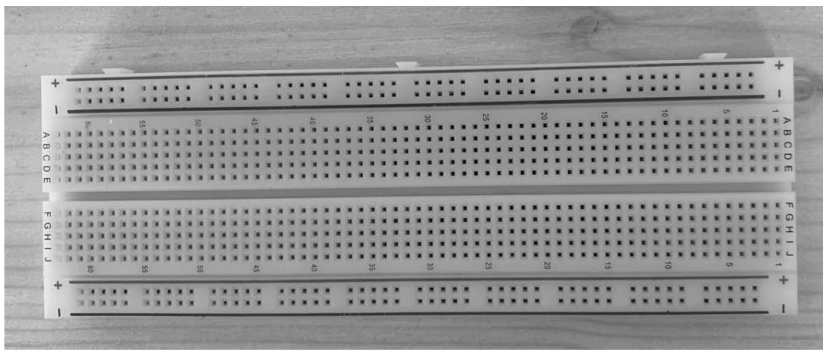

Рисунок 3.6 – Макетна плата

– перемички дюпон (рис. 3.7).

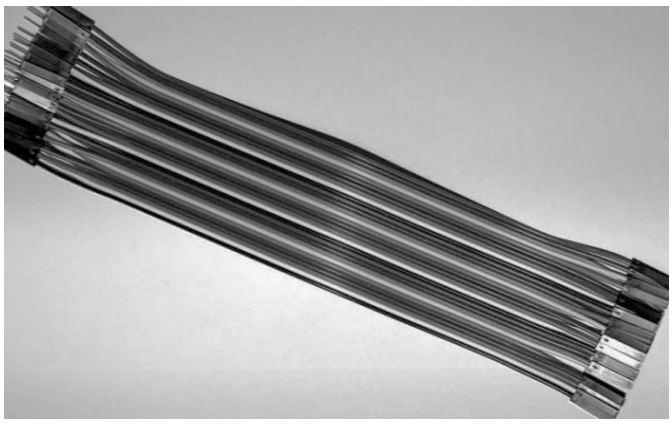

Рисунок 3.7 – Перемички

3.2 Складання контролера швидкості вентилятора з використанням Arduino Uno

Для працездатності контролера швидкості вентилятора потрібно правильно підключити всі комплектуючі. Почиється з підключання 16-канального РК-дисплея до мікроконтролера, як представлено на рисунку 3.8. А саме крім «живлення» та зазамлення підключаються виходи SDA та SCL дисплея до пінів A4 та A5 мікроконтролера.

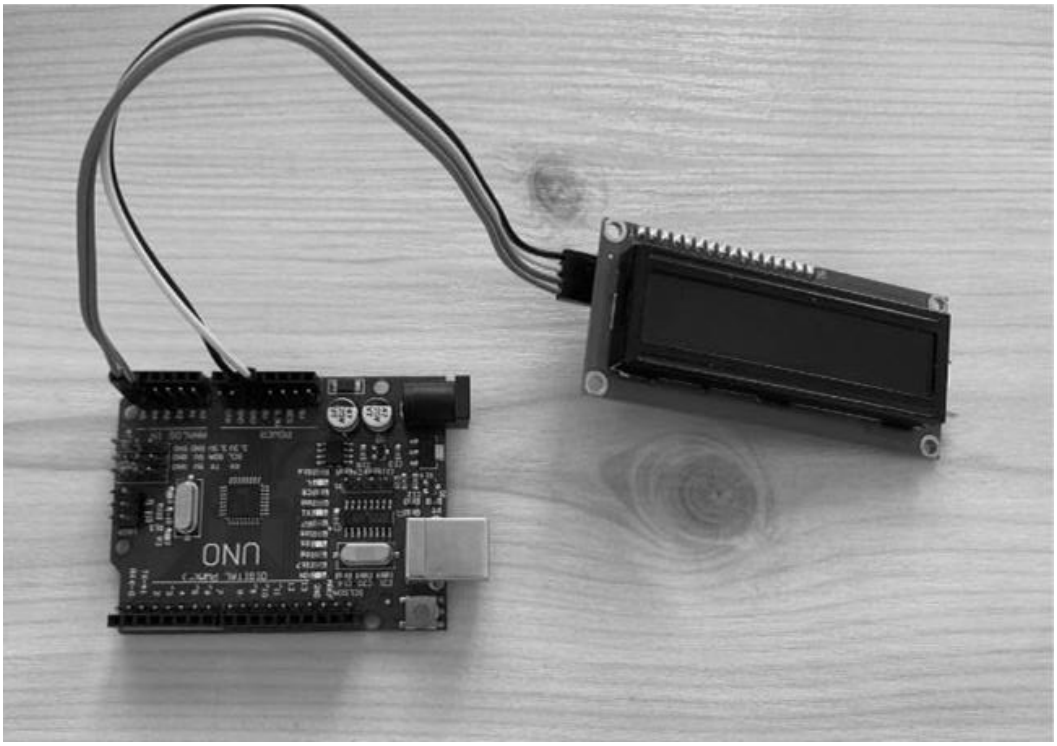

Рисунок 3.8 – Підключення РК-дисплея до мікроконтролера

Далі потрібно взяти макетну плату та підключити датчик температури та підключаємо до мікроконтролера. Датчик температури має три контакти: S для сигналу яки підключається до пін 6, середній контакт до 5 В, а останній – для заземлення, як зображено на рисунку 3.9.

Після підключення температурного датчика потрібно приєднати драйвер двигуна L298N (рис. 3.10) для керування двигуном вентилятора.

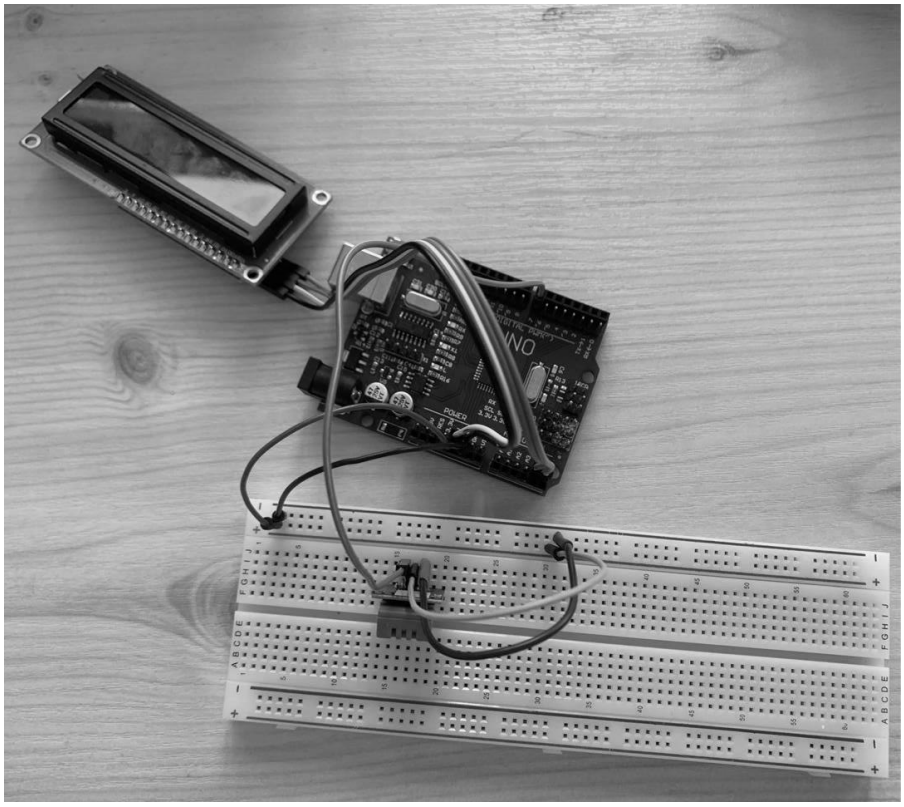

Рисунок 3.9 – Підключення датчика температури

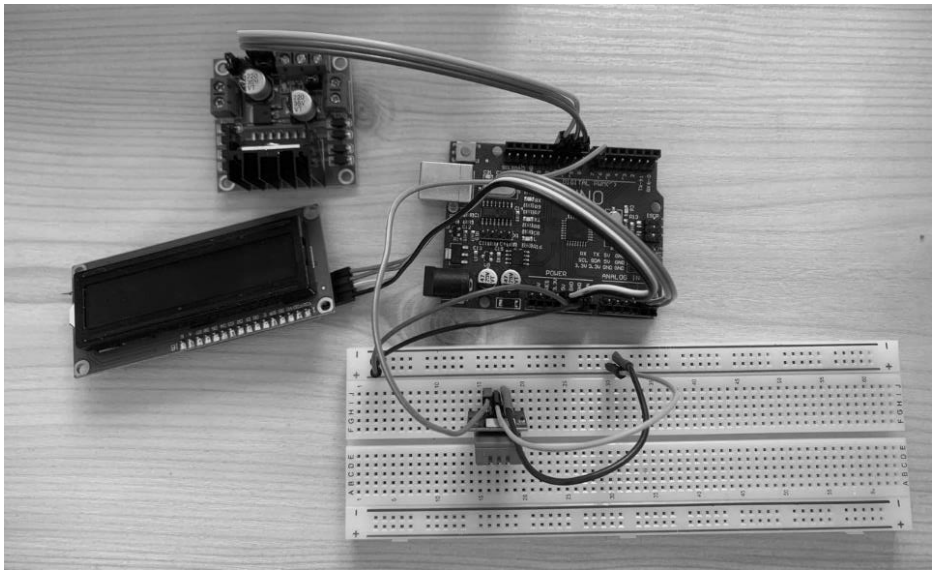

Рисунок 3.10 – Приєднання драйвера двигуна L298N

Після цього потрібно підключити двигун з вентилятором та драйвера двигуна L298N до USB-кабеля, як на рисунку 3.11. Потрібно взяти перемички та USB-кабель для підключання позитивного проводу USB-кабелю до контакту приводу двигуна, а потім підключються дроти до негативного контакту

негативного дроту USB-кабелю. Далі цей провід до контакту заземлення драйвера, після чого підключається один провід до 5-вольтового контакту драйвера двигуна, як представлено на рисунку 3.11.

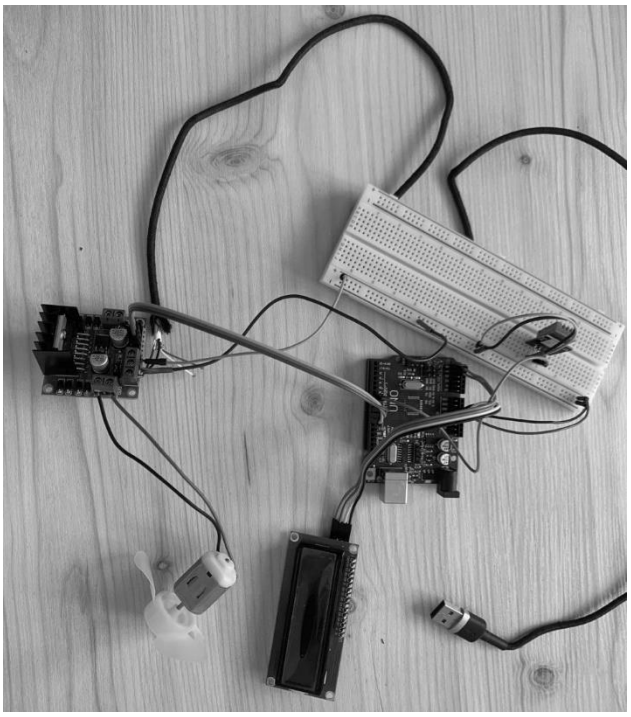

Рисунок 3.11 – Підключення вентилятора та

Далі потрібно запрограмувати мікроконтролер для роботи системи контролю швидкості вентилятора.

3.3 Конфігурація середовища розробки Arduino IDE

Для початку роботи над програмуванням нашого мікроконтролера Arduino Uno необхідно запустити інтегроване середовище розробки Arduino IDE, яке забезпечує зручний інтерфейс для написання, компіляції та завантаження коду на плату (рис. 3.12).

Наступним кроком є визначення моделі використовуваної плати. Для коректної роботи потрібно перейти до меню Інструменти → Плата, де із запропонованого списку необхідно обрати Arduino Uno або іншу плату, що

відповідає апаратній реалізації (рис. 3.13). Це дозволить компілятору правильно адаптувати код під вибрану архітектуру мікроконтролера.

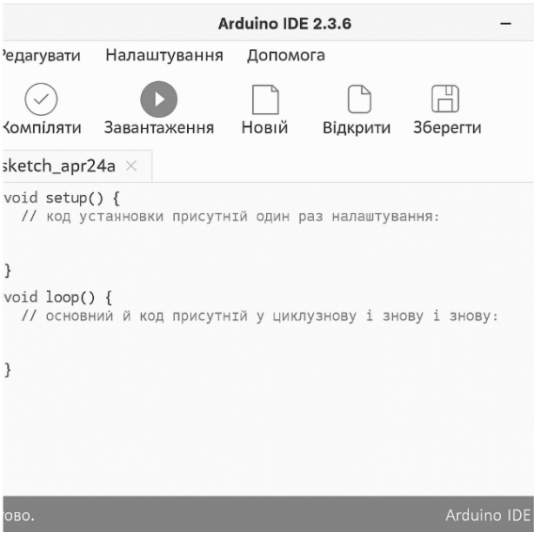

Рисунок 3.12 – Головне вікно Arduino IDE після запуску

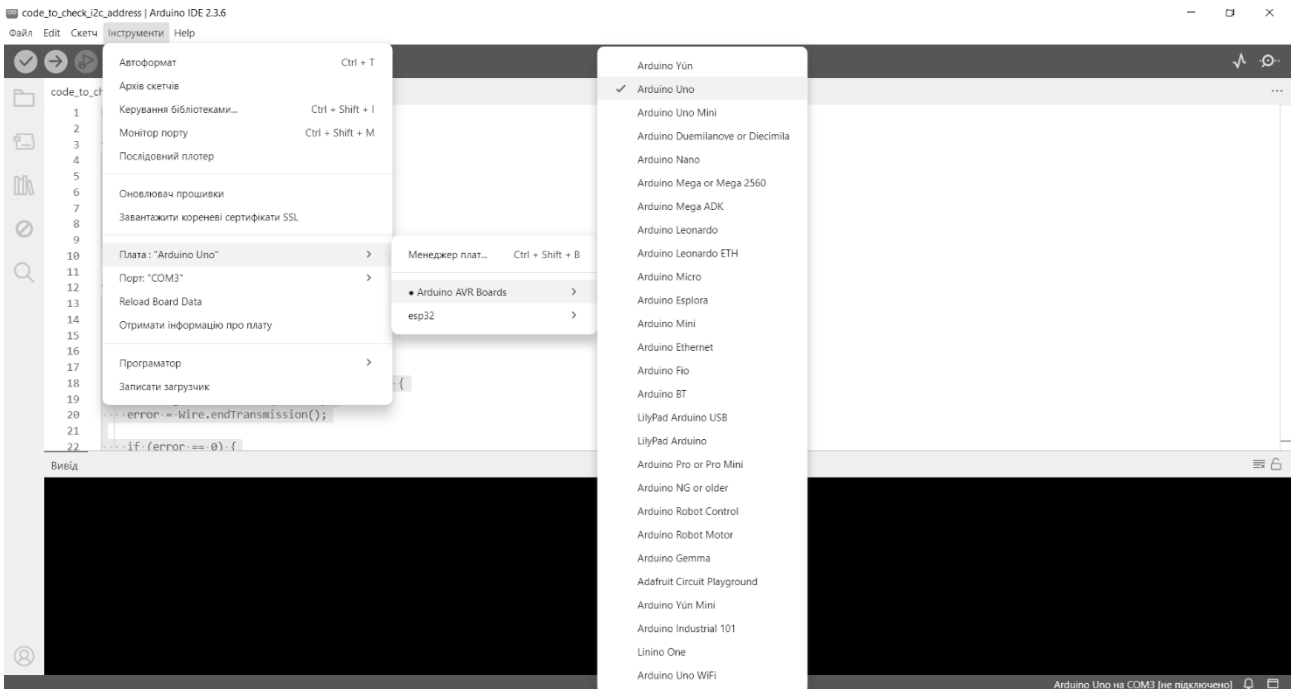

Рисунок 3.13 – Вибір моделі плати Arduino у меню інструментів

Після цього потрібно встановити правильний порт для з’єднання з мікроконтролером. COM-порт з’явиться у списку після успішного встановлення драйвера пристрою. Його вибір здійснюється у тому ж меню – Інструменти → Порт (рис. 3.14).

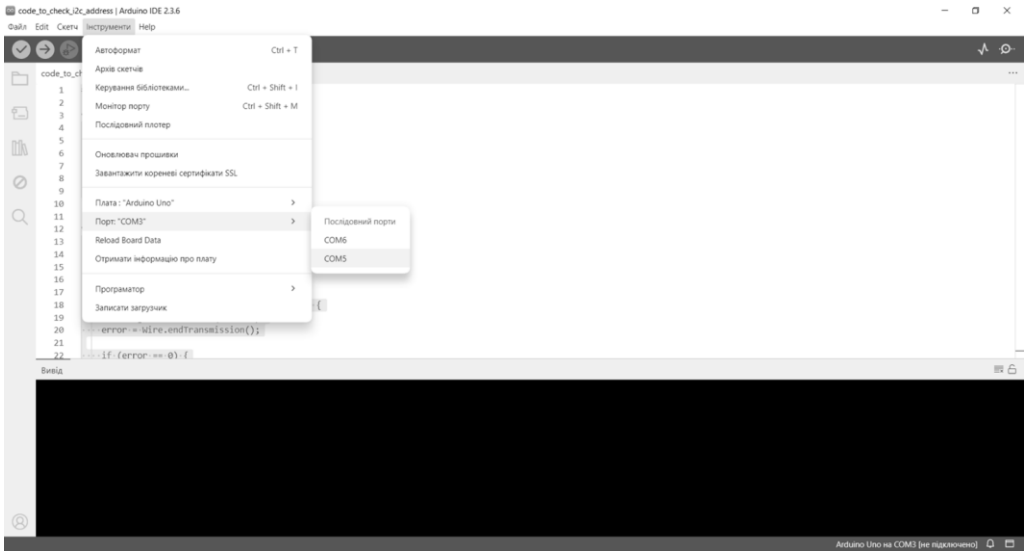

Рисунок 3.14 – Визначення відповідного COM-порту у налаштуваннях Arduino IDE

На панелі інструментів середовища розробки знаходяться основні елементи керування, які дозволяють здійснювати перевірку коду, завантаження ескізу (sketch), створення нових проектів, відкриття існуючих, збереження файлів та перегляд даних у послідовному моніторі.

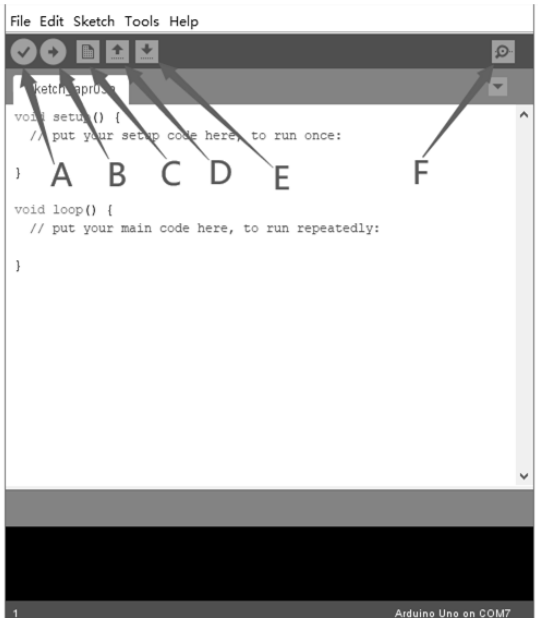

Рисунок 3.15 – Основні елементи керування Arduino IDE: A – перевірка коду на наявність помилок; B – завантаження коду на плату; C – створення нового проекту; D – відкриття прикладів; E – збереження файлу; F – запуск послідовного монітора для перегляду даних із плати

Для перевірки функціонування мікроконтролера рекомендується скористатись стандартним прикладом – програмою «Button». Вона демонструє принцип роботи цифрового виходу мікроконтролера шляхом періодичного натисканя кнопки Button (рис. 3.16).

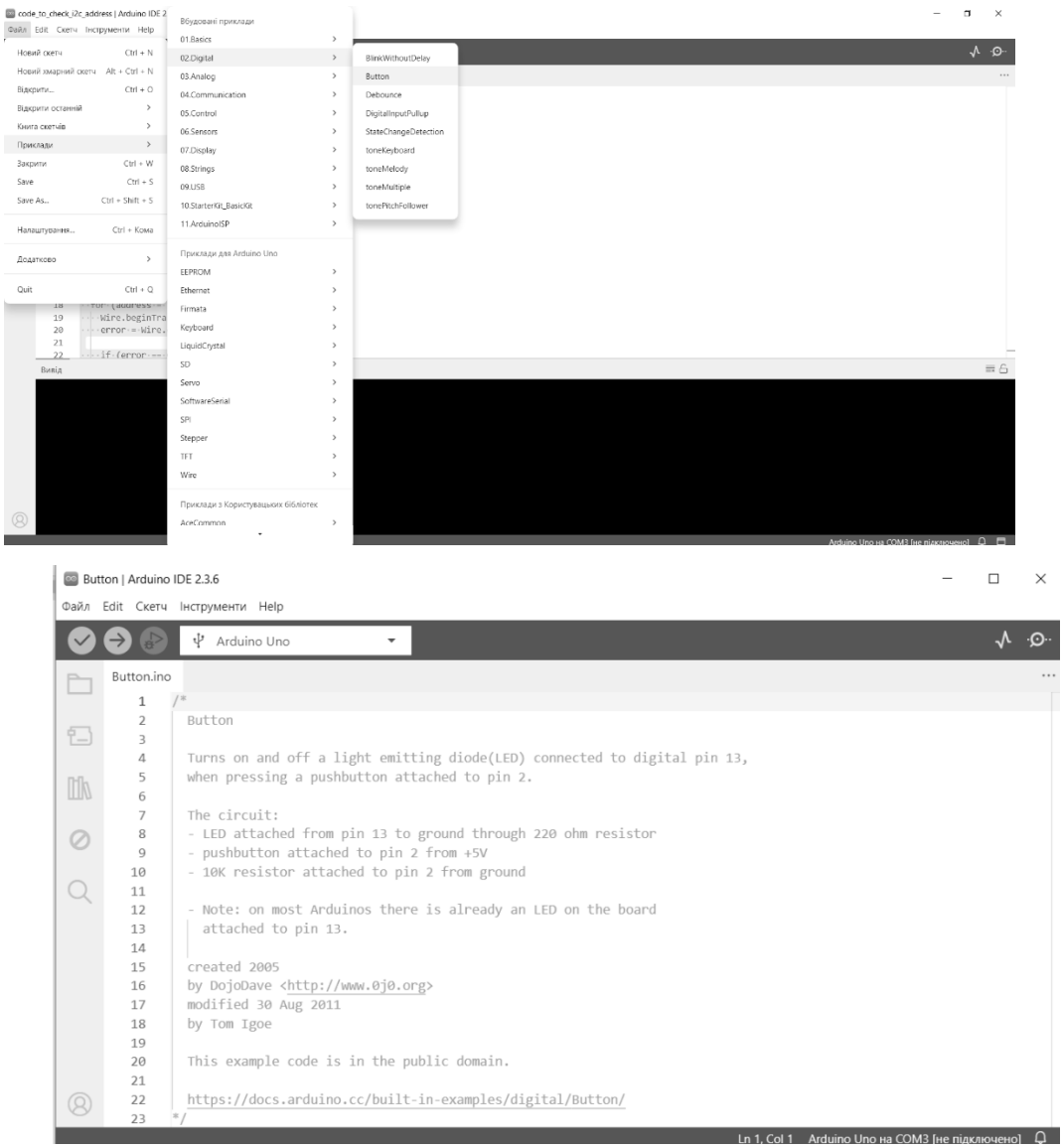

Рисунок 3.16 – Завантаження прикладу «Digital» із категорії «Button»

Перед компіляцією важливо переконатись, що обрана модель плати та COM-порт співпадають з реальним з’єднанням. Вони відображаються у нижній частині вікна IDE (рис. 3.17).

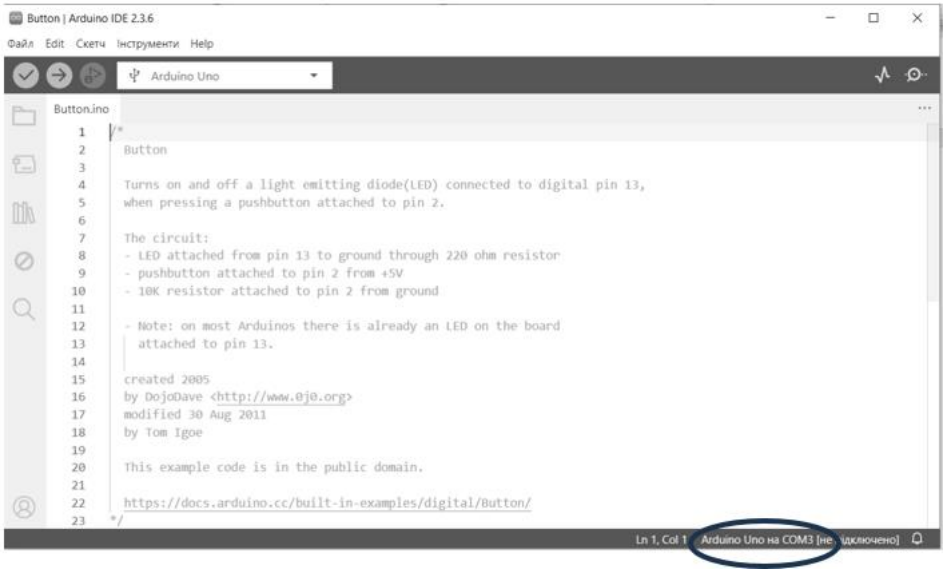

Рисунок 3.17 – Відображення активної плати та порту в інтерфейсі Arduino IDE

Для компіляції проєкту натискається кнопка перевірки, яка запускає аналіз коду на синтаксичні помилки (рис. 3.18).

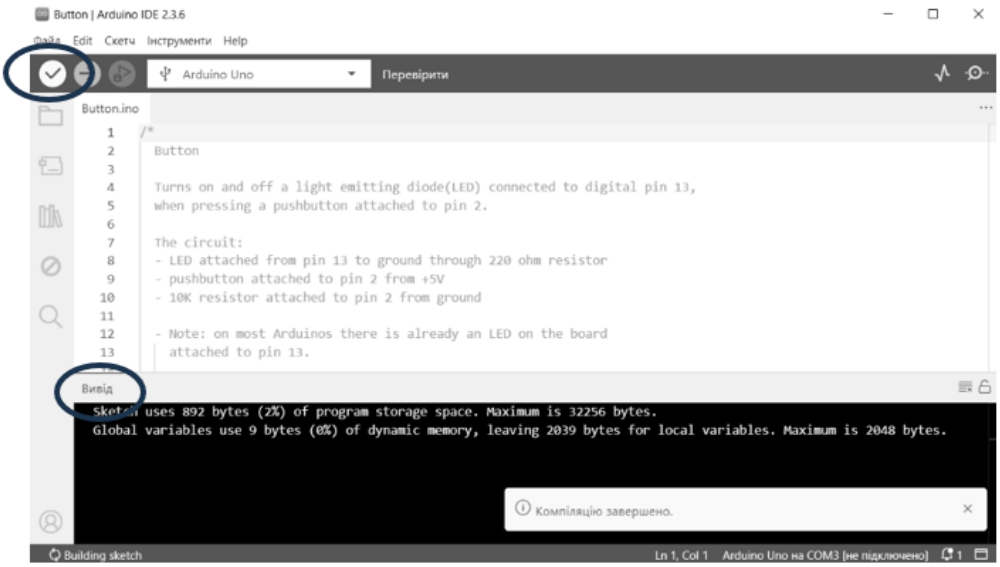

Рисунок 3.18 – Процес компіляції та виявлення помилок у коді

Після успішної перевірки необхідно натиснути кнопку завантаження, щоб передати прошивку на мікроконтролер, як на рисунку 3.19.

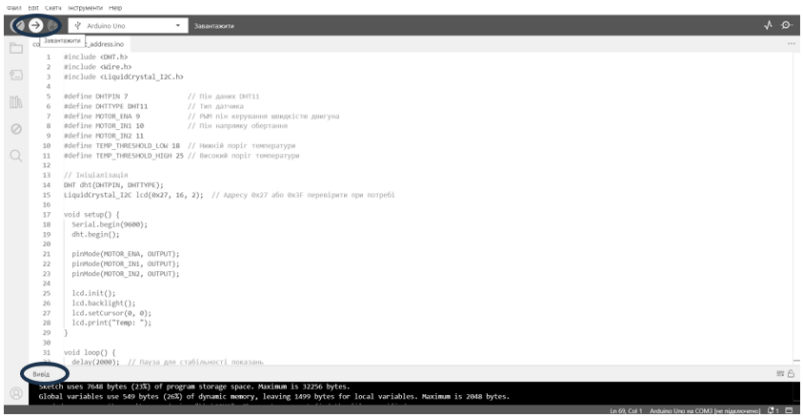

Рисунок 3.19 – Завантаження прошивки на плату Arduino

Після завершення процесу на платі Arduino Uno починає очинає працювати вбудований світлодіод, який мигає з інтервалом в одну секунду, підтверджуючи правильність прошивки та функціональність плати. Скетч програми наведено у додатку А.

Мікроконтролер підключють до ноутбука через USB-кабель, або до зарядного пристрою, або ж до портативного акумулятора, оскільки, при роботі можна спосткрігати, що зі збільшенням температури швидкість двигуна також збільшується. Робота пристороюю контролю швидкості вентилятора з використанням Arduino Uno представлено на рисунку 3.20.

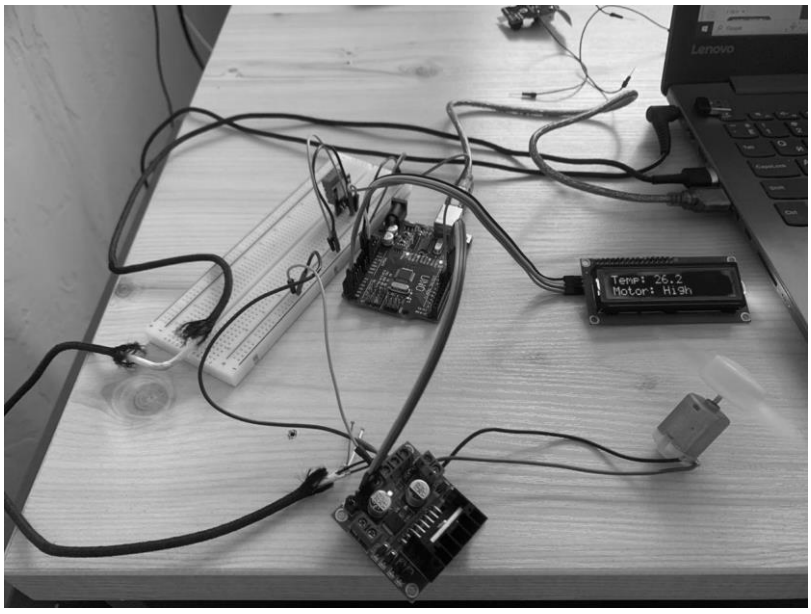

Рисунок 3.20 – Робота пристороюю контролю швидкості вентилятора з використанням Arduino Uno

За допомогою контролера можна автоматично вимикати швидкість вентилятора залежно від температури у вашій кімнаті чи на робочому столі.

ВИСНОВКИ

Системи температурного керування є критично важливими для захисту електронних пристроїв від перегріву, забезпечення комфортного мікроклімату та оптимізації енергоспоживання в побутових, промислових і вбудованих системах.

Було охарактеризовано ключові сфери застосування систем автоматичного керування температурою, а також базові структури та типові підходи до реалізації автоматичного керування, включно з пороговим, пропорційним і ПІД-алгоритмами, що дозволило сформувати загальне уявлення про логіку регулювання на практиці.

На основі аналізу сформовано перелік технічних і функціональних вимог до мікроконтролера, серед яких ключовими є підтримка PWM, доступність цифрових входів/виходів та енергоефективність – це стало основою для обґрунтування вибору Arduino.

Було систематизовано методи керування вентилятором і охарактеризовано основні електронні компоненти, що беруть участь у побудові системи, зокрема сенсори, мікроконтролери та силові ключі.

Здійснено аналіз можливостей сучасних плат Arduino та сумісних платформ, що дозволило виявити переваги Arduino UNO як базового рішення завдяки простоті, доступності та великій спільноті підтримки. Порівняння популярних температурних сенсорів підтвердило доцільність використання DHT22 через його точність, цифровий інтерфейс і відповідність вимогам до побудови надійної системи контролю температури. Було підібрано оптимальні компоненти для реалізації системи, зокрема Arduino UNO, DHT22, MOSFET та вентилятор DC, які забезпечують узгоджену та надійну апаратну основу для реалізації заданих функцій.

Arduino UNO виправдав вибір завдяки сумісності з цифровими сенсорами, підтримці PWM та широкій бібліотечній базі, що спрощує реалізацію логіки керування та інтеграцію із периферією. DHT22 забезпечив точне зчитування температури, а його стабільна робота в заданому діапазоні дозволила побудувати

ефективну систему моніторингу з мінімальною похибкою. Аналіз вентилятора показав відповідність вимогам до низьковольтного живлення, керованості та компактності, що є ключовими для інтеграції у портативні або стаціонарні системи охолодження.

Реалізований алгоритм забезпечує адаптивне керування вентиляцією на основі температури в реальному часі, яка дозволяє досягти плавного регулювання швидкості обертання.

Система може бути ефективно застосована у побутових, навчальних та промислових умовах завдяки простій архітектурі, можливості автономної роботи та масштабованості.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ghaffari R. Basic Control Systems for Electronics. *Electronics Tutorials*. 2023. URL: https://www.electronics-tutorials.ws/systems (дата звернення: 22.03.2025).

2. Ashima Jain et al. Temperature Based Automatic Fan Speed Control System using Arduino. *SSRN Electronic Journal*. 2022. URL: https://ssrn.com/abstract=4159188 (дата звернення: 22.03.2025).

3. Rachelle Ann Buco et al. Smart room temperature regulation with responsive Arduino-driven temperature-controlled fan system. *International Journal of Research in Science & Engineering*. Vol. 5(1). 2025. URL: https://journal.hmjournals.com/index.php/IJRISE/article/view/5316 (дата звернення: 24.03.2025).

4. A. Elasha & R. Binns. Design and Implementation of a Microcontroller-Based Temperature Control System for a Cosmetics Insulation Box. *ACM Digital Library*. 2023. URL: https://dl.acm.org/doi/10.1145/3640115.3640119 (дата звернення: 29.03.2025).

5. Temperature Based Fan Speed Controller using Arduino. *How To Electronics*. URL: https://how2electronics.com/temperature-based-fan-speed-controller-using-arduino/ (дата звернення: 31.03.2025).

6. Temperature Controlled Fan with Arduino. *GitHub*. URL: https://github.com/Circuit-Digest/temperature-controlled-fan (дата звернення: 31.03.2025).

7. Nikhil Asware et al. Temperature Based Fan Speed Controller and Monitoring with Arduino. *International Research Journal of Modernization in Engineering Technology and Science*. Vol. 05. Issue:10. 2023. URL: https://www.irjmets.com/uploadedfiles/paper//issue_10_october_2023/45788/final/fin_irjmets1698948143.pdf?utm_source=chatgpt.com (дата звернення: 31.03.2025 р.)

8. Arduino PID. *GitHub*. URL: https://github.com/lily-osp/arduino-pid (дата звернення: 31.03.2025).

9. Open Loop System Vs Closed Loop System. *Electronicsinfos*. URL: https://www.electronicsinfos.com/2023/02/open-loop-system-vs-closed-loop-system.html (дата звернення: 31.03.2025).

10. ATMEGA328P Datasheet. *Microchip Technology*. URL: https://www.alldatasheet.com/datasheet-pdf/download/1425005/MICROCHIP/ATMEGA328P.html (дата звернення: 01.04.2025).

11. ESP32 – Technical Reference Manual. *Espressif Systems*. URL: https://www.espressif.com/en/products/socs/esp32/resources (дата звернення: 08.04.2025).

12. STM32F103C8. Mainstream Performance line. *STMicroelectronics*. URL: https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html (дата звернення: 09.04.2025).

13. A Microcontroller by Raspberry Pi. *RP2040 Datasheet*. URL: https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf (дата звернення: 15.04.2025).

14. DHT11 Humidity & Temperature Sensor. *Mouser Electronics*. URL: https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf (дата звернення: 17.04.2025).

15. Digital-output relative humidity & temperature sensor/module AM2303. *Aosong(Guangzhou) Electronics Co.,Ltd*. URL: https://cdn-shop.adafruit.com/datasheets/DHT22.pdf (дата звернення: 23.04.2025).

16. LM35 Precision Centigrade Temperature Sensors. *Texas Instruments Incorporated*. URL: https://www.ti.com/product/LM35?qgpn=lm35 (дата звернення: 23.04.2025).

17. DS18B20 Programmable Resolution 1-Wire Digital Thermometer. *Maxim Integrated Products*. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/ds18b20.pdf (дата звернення: 26.04.2025).

18. Arduino UNO. *Arduino*. URL: https://docs.arduino.cc/hardware/uno-rev3/ (дата звернення: 01.05.2025).

19. Interfacing DHT22 Humidity & Temperature Sensor with Arduino. *Circuit Digest.* URL: https://circuitdigest.com/microcontroller-projects/interface-dht22-sensor-module-with-arduino?utm_source=chatgpt.com (дата звернення: 05.05.2025).

20. Arduino – Control Fan. *Arduinogetstarted.* URL: https://arduinogetstarted.com/tutorials/arduino-controls-fan (дата звернення: 05.05.2025).

21. Connecting an N-Channel MOSFET. *Arduino.* URL: https://projecthub.arduino.cc/ejshea/connecting-an-n-channel-mosfet-6a7325 (дата звернення: 05.05.2025).

22. Ovi Saha et al. Automatic Fan Speed Control using Temperature and Humidity Sensor using Arduino. *ResearchGate GmbH.* 2023. URL: https://www.researchgate.net/publication/376828589_Automatic_Fan_Speed_Control_using_Temperature_and_Humidity_Sensor_using_Arduino (дата звернення: 05.05.2025).

23. Snehashis Das, Sayak Pal, Tithi Mukhopadhyay, Sukalyan Nath. Automatic temperature controlled fan. *International Journal of Recent Research in Electrical and Electronics Engineering (IJRREEE).* Vol. 11. Issue 3. 2024. URL: https://zenodo.org/records/13353749?utm_source (дата звернення: 10.05.2025).

ДОДАТКИ

Додаток А

Код програми

```
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define DHTPIN 7               // Пін даних DHT11
#define DHTTYPE DHT11          // Тип датчика
#define MOTOR_ENA 9            // PWM пін керування швидкістю двигуна
#define MOTOR_IN1 10           // Пін напрямку обертання
#define MOTOR_IN2 11
#define TEMP_THRESHOLD_LOW 18  // Нижній поріг температури
#define TEMP_THRESHOLD_HIGH 25 // Високий поріг температури

// Ініціалізація
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Адресу 0x27 або 0x3F перевірити при потребі

void setup() {
  Serial.begin(9600);
  dht.begin();

  pinMode(MOTOR_ENA, OUTPUT);
  pinMode(MOTOR_IN1, OUTPUT);
  pinMode(MOTOR_IN2, OUTPUT);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
}

void loop() {
  delay(2000);  // Пауза для стабільності показань

  float tempC = dht.readTemperature();
  if (isnan(tempC)) {
    Serial.println("Error reading temperature");
    return;
  }

  Serial.print("Temp: ");
  Serial.print(tempC);
  Serial.println(" °C");

  lcd.setCursor(6, 0);
  lcd.print("     ");     // Очищення попереднього значення
  lcd.setCursor(6, 0);
  lcd.print(tempC, 1);    // Один знак після коми

  lcd.setCursor(0, 1);

  // Логіка керування двигуном
```

```
  if (tempC > TEMP_THRESHOLD_HIGH) {
    analogWrite(MOTOR_ENA, 150);
    digitalWrite(MOTOR_IN1, HIGH);
    digitalWrite(MOTOR_IN2, LOW);
    lcd.print("Motor: High ");
  } else if (tempC > TEMP_THRESHOLD_LOW) {
    analogWrite(MOTOR_ENA, 80);
    digitalWrite(MOTOR_IN1, HIGH);
    digitalWrite(MOTOR_IN2, LOW);
    lcd.print("Motor: Med  ");
  } else {
    analogWrite(MOTOR_ENA, 45);
    digitalWrite(MOTOR_IN1, HIGH);
    digitalWrite(MOTOR_IN2, LOW);
    lcd.print("Motor: Low  ");
  }
}
```