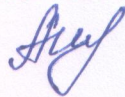


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут автоматики та електротехніки

Кафедра: Морського приладобудування

«Допущений до захисту»

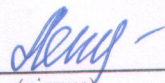
В.о. завідувача кафедри

 Гайдай Г.Ю.
«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти «магістр»

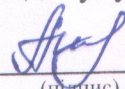
на тему: «Моніторинг апаратного забезпечення комп'ютеру
на базі Arduino з LCD дисплеєм»

Виконав: студент групи 6651м

 Летунов К.І.
(підпис)

Керівник роботи:

В.о завідувача кафедри морського
приладобудування, к.т.н., доц.

 Гайдай Г.Ю.
(підпис)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут автоматики та електротехніки

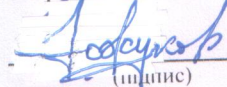
Кафедра морського приладобудування

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма «Комп'ютерні системи та мережі»

«ЗАТВЕРДЖУЮ»

Гарант освітньої програми


(підпис)

Жуков Ю.Д.

«___» _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти «магістр»

Студенту _____ Летунову Кирилу Ігоровичу
(Прізвище, ім'я, по батькові)

1. Тема роботи: «Моніторинг апаратного забезпечення комп'ютеру на базі Arduino з LCD дисплеєм»

Керівник роботи: Гайдай Ганна Юріївна, кандидат технічних наук, доцент

Затверджені наказом ректора № 121742 від «14» 10 2025 року

2. Термін подання роботи: 17.12.25

3. Вихідні дані по роботі: тематика 49 мв/А; час виконув. 3.ххх.

4. Перелік питань, що належать до розробки (найменування розділів): _____

1. Аналіз призначення пристрою моніторингу персонального комп'ютеру; 2. Обґрунтування та розробка апаратної структури системи моніторингу; 3. Розробка апаратно-програмного забезпечення на основі платформи Arduino; 4. Охорона праці;

5. Перелік презентаційних матеріалів: 1. Актуальність теми; 2. Мета

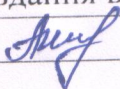
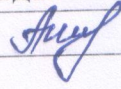
та завдання дослідження; 3. Вибір апаратної платформи;

4. Вибір датчиків та методів вимірювання; 5. Схемотехнічна

схема розв'язку; 6. Схема з'єднання; 7. Схема

підключення; 8. Загальні висновки

6. Консультанти розділів роботи

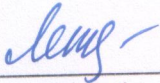
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Чоп	Тубалєв А.М.		

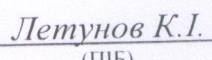
7. Дата видачі завдання « 3 » 09 2025 року

КАЛЕНДАРНИЙ ПЛАН

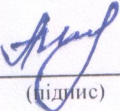
№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз призначення пристрою моніторингу персонального комп'ютеру	3.09 – 3.10.25	век.
2.	Обґрунтування та розробка апаратної структури системи моніторингу	3.10 – 27.10.25	век.
3.	Розробка апаратно-програмного забезпечення на основі платформи Arduino	27.10 – 17.11.25	век.
4.	Охорона праці	17.11 – 17.12.25	век.

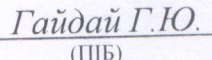
Студент


(підпис)


(ПІБ)

Керівник роботи


(підпис)


(ПІБ)

АНОТАЦІЯ

Летунов К.І. «Моніторинг апаратного забезпечення комп'ютеру на базі Arduino з LCD дисплеєм». Кваліфікаційна робота зі спеціальності 123 «Комп'ютерна інженерія», освітньої програми «Комп'ютерні системи та мережі». Миколаїв: НУК ім. адм. Макарова. 2025 рік. 81 сторінок.

У кваліфікаційній роботі розроблено автономний апаратно-програмний комплекс для моніторингу температурних та електричних параметрів персонального комп'ютера на базі платформи Arduino. Проведено аналіз існуючих методів діагностики несправностей ПК, обґрунтовано вибір елементної бази та сенсорів. Розроблено структурну та принципову електричну схеми пристрою, реалізовано алгоритми обробки сигналів датчиків і логіку автоматичного керування системою охолодження (Smart PWM). Створено клієнтське програмне забезпечення для візуалізації телеметрії та налаштування параметрів системи. Отримані результати забезпечують підвищення надійності та відмовостійкості комп'ютерної техніки завдяки незалежному контролю критичних вузлів навіть у випадку програмних збоїв операційної системи.

Ключові слова: Апаратний моніторинг; платформа Arduino; датчик температури; система охолодження; ШІМ-керування; інтерфейс 1-Wire; гальванічна розв'язка; діагностика ПК.

ABSTRACT

Letunov K.I. «Arduino-based computer hardware monitoring with LCD display». Qualification work in specialty 123 «Computer Engineering», educational program «Computer Systems and Networks». – Mykolaiv: Admiral Makarov National University of Shipbuilding, 2025, 81 pages.

An autonomous hardware-software complex for monitoring the temperature and electrical parameters of a personal computer based on the Arduino platform has been developed in this qualification paper. An analysis of existing PC fault diagnosis methods has been conducted, and the selection of the component base and sensors has been justified. Structural and schematic diagrams of the device have been developed, and algorithms for sensor signal processing and logic for automatic cooling system control (Smart PWM) have been implemented. Client software for telemetry visualization and system configuration has been created. The obtained results ensure increased reliability and fault tolerance of computer equipment due to independent monitoring of critical components, even in the event of operating system software failures.

Keywords: Hardware monitoring; Arduino platform; temperature sensor; cooling system; PWM control; 1-Wire interface; galvanic isolation; PC diagnostics.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	9
ВСТУП	10
РОЗДІЛ 1. АНАЛІЗ ПРИЗНАЧЕННЯ ПРИСТРОЮ МОНІТОРИНГУ ПЕРСОНАЛЬНОГО КОМП'ЮТЕРУ	13
1.1 Вплив систем охолодження та електроживлення на стабільність роботи ПК	13
1.1.1 Наслідки та діагностика перегріву	13
1.2 Обґрунтування необхідності апаратного моніторингу комп'ютерних систем	15
1.3 Види та технічна реалізація моніторингу комп'ютерних систем	16
1.3.1 Особливості температурного моніторингу компонентів ПК	19
1.3.2 Моніторинг електричних параметрів	21
1.3.3 Моніторинг швидкості обертання вентиляторів	23
1.3.4 Програмні засоби діагностики та моніторингу	24
1.4 Переваги та практичне значення моніторингу стану комп'ютерної системи	25
1.5 Рекомендації щодо контролю ключових технічних параметрів комп'ютерної системи	27
1.6 Аналіз існуючих апаратних аналогів	28
1.6.1 Апаратний комплекс Lampton HC070	28
1.6.2 Апаратний комплекс LCDSysInfo 3.5	30
1.7 Аналіз існуючих програмних аналогів	31
1.7.1 Утиліта Piriform Speccy	32
1.7.2 Програмний комплекс AIDA64 Extreme	33
Висновки за розділом 1.....	34
РОЗДІЛ 2. ОБґРУНТУВАННЯ ТА РОЗРОБКА АПАРАТНОЇ СТРУКТУРИ СИСТЕМИ МОНІТОРИНГУ	36
2.1 Аналіз функціональних вимог до системи	36
2.2 Аналіз методів вимірювання фізичних величин	38

2.2.1 Методи вимірювання електричних величин	39
2.2.2 Вплив вимірювальних приладів на об'єкт дослідження	40
2.3 Обґрунтування вибору та опис апаратної платформи Arduino	41
2.3.1 Архітектура та апаратна частина	42
2.3.2 Інтерфейси введення-виведення	42
2.4 Обґрунтування вибору та опис модулю дисплея	43
2.5 Обґрунтування вибору та опис датчика температури DS18B20.....	47
2.6 Обґрунтування вибору та опис датчика струму CSLA1CD	48
2.7 Схемотехніка гальванічної розв'язки інтерфейсу 1-Wire	50
Висновки за розділом 2.....	52
РОЗДІЛ 3. РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ПЛАТФОРМИ ARDUINO	54
3.1 Життєвий цикл розробки програмного забезпечення	54
3.2 Проектування та макетування апаратної частини	56
3.2.1 Розробка принципової схеми підключення	56
3.2.2 Тестування та налагодження апаратної частини прототипу	58
3.3 Програмна реалізація системи	59
3.3.1 Програмна реалізація алгоритму вимірювання струму	60
3.3.2 Програмування інтерфейсу користувача (LCD I2C)	62
3.3.3 Програмна реалізація алгоритму термометрії	63
3.4 Візуалізація монтажною схемою у середовищі Fritzing	65
3.5 Опис інтерфейсу користувача та функціональних можливостей програмного комплексу	66
Висновки за розділом 3.....	69
РОЗДІЛ 4. ОХОРОНА ПРАЦІ	70
4.1 Аналіз шкідливих та небезпечних виробничих факторів	70
4.2 Вимоги до мікроклімату робочої зони.....	71
4.3 Виробниче освітлення	71
4.4 Електробезпека.....	72
4.5 Пожежна безпека.....	72

Висновки за розділом 4.....	73
ВИСНОВКИ.....	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	76
ДОДАТОК А	78
ДОДАТОК Б	81

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

RGB – (Red Green Blue) один із основних способів кодування кольорів;
IDE – Integrated Development Environment;
LCD – Liquid Crystal Display;
USB – Universal Serial Bus;
MVC – Model–View–Controller;
IT – Information Technologies;
CPU – Central Processing Unit;
GPU – Graphics Processing Unit;
SATA – Serial ATA;
БЖ – блок живлення;
РК – рідкокристалічний дисплей;
ПК – персональний комп'ютер;
АПЗ – Апаратно-програмне забезпечення;
ОС – операційна система.

ВСТУП

Функціонування сучасних електронних та електромеханічних систем нерозривно пов'язане з процесом виділення теплової енергії. Рівень тепловиділення варіюється від незначного (у мобільних пристроях) до критично високого (у високопродуктивних обчислювальних системах). Надмірне тепловиділення без належного відведення здатне порушити працездатність як самих обчислювальних ядер, так і периферійних компонентів, розташованих поруч.

На надійність системи, окрім номінальних технічних характеристик, суттєво впливають умови експлуатації та фактор старіння матеріалів. З часом ефективність систем охолодження знижується через накопичення пилу, а термоінтерфейси втрачають свої фізико-хімічні властивості (висихання, затвердіння), що призводить до погіршення теплопровідності. Емпіричні дані свідчать, що відсутність технічного обслуговування протягом року може підвищити робочу температуру компонентів на 5–15 °C, що значно прискорює деградацію апаратної частини.

Якщо центральні процесори (CPU) та графічні прискорювачі (GPU) зазвичай оснащені вбудованими сенсорами та системами термозахисту (тротлінг, аварійне вимкнення), то інші критично важливі вузли, зокрема блоки живлення (БЖ) та силові елементи материнських плат, часто залишаються без контролю. У більшості споживчих моделей БЖ відсутня телеметрія, що унеможливорює програмний моніторинг їхнього стану.

З огляду на це, актуальним завданням є розробка автономної системи моніторингу на базі зовнішніх сенсорів. Використання накладних датчиків температури та безконтактних датчиків струму дозволить контролювати тепловий режим та енергоспоживання вузлів, не охоплених штатною діагностикою, що значно підвищить загальну надійність та відмовостійкість обчислювальної техніки.

Мета роботи: Розробка та реалізація автономної системи моніторингу температурних та електричних параметрів компонентів персонального комп'ютера (ПК) на базі мікроконтролера Arduino NANO з виведенням діагностичної інформації на LCD-дисплей. Система має забезпечувати підвищення надійності роботи комп'ютерної техніки шляхом своєчасного виявлення перегріву та попередження аварійних ситуацій.

Об'єкт дослідження: Процес діагностики та моніторингу технічного стану апаратних компонентів комп'ютерних систем.

Предмет дослідження: Апаратно-програмні засоби зовнішнього моніторингу параметрів ПК на основі мікроконтролерних систем.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Провести огляд існуючих методів та засобів моніторингу комп'ютерних систем, проаналізувати їхні переваги та недоліки.
2. Розробити структурну та принципову схему пристрою на базі платформи Arduino NANO (ATmega328).
3. Створити прототип системи, що складається з внутрішнього блоку (датчики, контролер) та зовнішнього блоку управління (LCD-дисплей, органи керування).
4. Розробити прошивку для мікроконтролера у середовищі Arduino IDE (мова C/C++).
5. Створити клієнтське програмне забезпечення для ПК (portable-версія) для зчитування системних даних та передачі їх на зовнішній пристрій через USB-інтерфейс, а також для візуалізації графіків температур.
6. Забезпечити простоту підключення (Plug-and-Play) та зручність інтерфейсу користувача.

Практичне значення одержаних результатів полягає у створенні універсального, бюджетного пристрою для модернізації («модингу») застарілого комп'ютерного парку, який не має вбудованих сучасних засобів діагностики.

Запропонований пристрій дозволяє:

- Отримувати дані про стан системи незалежно від виводу зображення на монітор.
- Використовувати зовнішні датчики там, де відсутні вбудовані сенсори.
- Здійснювати моніторинг без складного налаштування ПЗ.

Перспективи подальшого розвитку проекту включають інтеграцію з мережевими технологіями (IoT) для віддаленого моніторингу груп комп'ютерів (наприклад, у навчальних класах або офісах) через мобільний додаток або локальний сервер.

Актуальність створення зовнішнього пристрою моніторингу зумовлена необхідністю обслуговування великого парку комп'ютерної техніки в освітніх закладах, офісах та на підприємствах. З часом ефективність систем охолодження знижується через забруднення пилом та висихання термоінтерфейсів, що призводить до перегріву, виходу з ладу конденсаторів, процесорів та інших критичних вузлів.

Більшість існуючих програмних рішень вимагають активного втручання користувача для перевірки температур. Розроблюваний апаратний комплекс дозволяє автоматизувати цей процес та зробити його наочним. Це дає змогу обслуговуючому персоналу швидко ідентифікувати проблемні машини в аудиторії чи офісі, запобігаючи дорогому ремонту та втраті даних. Таким чином, розробка є економічно доцільною як для приватного використання, так і для корпоративного сектору.

РОЗДІЛ 1. АНАЛІЗ ПРИЗНАЧЕННЯ ПРИСТРОЮ МОНІТОРИНГУ ПЕРСОНАЛЬНОГО КОМП'ЮТЕРУ

1.1 Вплив систем охолодження та електроживлення на стабільність роботи ПК

Ефективність роботи активної системи охолодження є критичним фактором для забезпечення стабільного температурного режиму обчислювальної техніки. У процесі експлуатації відбувається природний знос механічних частин вентиляторів (підшипників) та накопичення пилу на ребрах радіаторів. Це призводить до погіршення аеродинамічних властивостей системи та зниження коефіцієнта тепловіддачі, що неминуче спричиняє перегрів критичних вузлів.

Окремим фактором ризику є підсистема електроживлення. Номінальна потужність блоків живлення (БЖ) не завжди відповідає піковим навантаженням. Розподіл енергії в ПК відбувається по лініях напруги +3.3 В, +5 В та +12 В. Сумарна сила струму, що споживається компонентами, динамічно змінюється залежно від навантаження на систему. Якщо споживана потужність (яка визначається як добуток сили струму на напругу) перевищує можливості БЖ, виникає явище падіння напруги («просідання») на лініях живлення. Аналогічні проблеми виникають внаслідок деградації електролітичних конденсаторів у ланцюгах БЖ через їх старіння.

Особливої актуальності питання термоконтролю набуває в режимах підвищеної продуктивності («оверклокінг»). Експлуатація обладнання на частотах, що перевищують заводські, вимагає суворого дотримання температурних норм для запобігання деградації кристалів напівпровідників.

1.1.1 Наслідки та діагностика перегріву

Перегрів визначається як стан, при якому тепловиділення компонентів перевищує можливості системи тепловідведення. Це може призвести до:

- Тротлінгу (примусового зниження тактової частоти для охолодження).

- Аварійного вимкнення системи для запобігання фізичному руйнуванню процесора (CPU).
- Виходу з ладу суміжних компонентів: оперативної пам'яті, накопичувачів (HDD/SSD), графічного адаптера (GPU) та елементів VRM (модуля регулювання напруги).

Основними причинами порушення теплового режиму є: відмова активних елементів охолодження, порушення циркуляції повітря всередині корпусу, забруднення пилом, висихання термоінтерфейсів та нестабільне електроживлення. Для запобігання аварійним ситуаціям необхідно проводити регулярне технічне обслуговування та модернізацію системи охолодження.

Контроль температури на апаратному рівні здійснюється за допомогою вбудованих терморезистивних датчиків (термисторів), інтегрованих безпосередньо в структуру мікросхем (рис. 1.1). Однак для компонентів, позбавлених вбудованих сенсорів, необхідне використання зовнішніх засобів моніторингу.

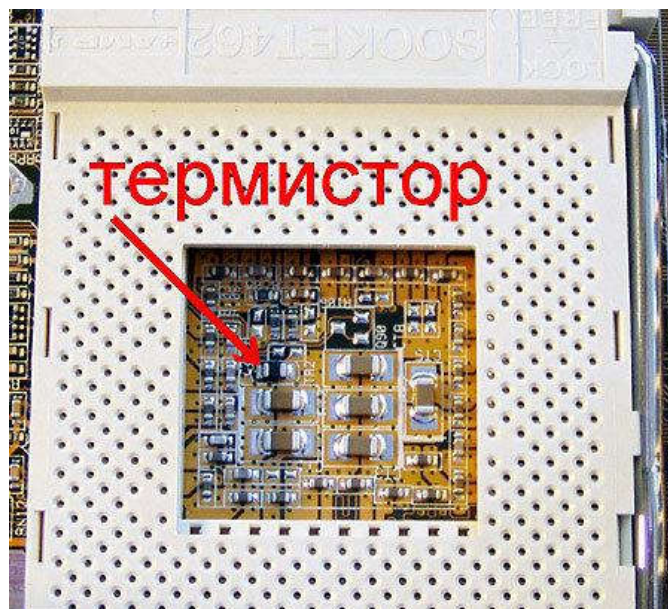


Рис. 1.1 – Фізичне виконання термистора

1.2 Обґрунтування необхідності апаратного моніторингу комп'ютерних систем

Надійність та функціональні можливості будь-якого електронного пристрою безпосередньо залежать від його фізичного стану та режимів експлуатації. Це стосується як комп'ютера в цілому, так і окремих його комплектуючих. Діагностувати перед аварійний стан деталі, спираючись лише на зовнішні ознаки (звук роботи, візуальний огляд), вкрай складно, оскільки критичні зміни параметрів часто не мають явних зовнішніх проявів.

Система апаратного моніторингу значно прискорює пошук причин нестабільної роботи ПК та спрощує його обслуговування. За відсутності автоматизованих засобів контролю, для перевірки параметрів живлення довелося б щоразу розбирати корпус системного блоку та проводити ручні вимірювання за допомогою мультиметра. При цьому такий метод дозволяє виміряти напругу лише в конкретний момент часу, тоді як температура та швидкість обертання вентиляторів вимагають інших методів фіксації.

Критично важливим параметром є стабільність напруги. Згідно зі стандартом ATX, якщо напруги блоку живлення виходять за межі $\pm 5\%$ від номіналу, такий блок вважається несправним і підлягає ремонту або заміні. Особливо небезпечним є підвищення напруги по лінії +12 В, яка використовується, зокрема, для живлення двигунів жорстких дисків (HDD). Драйвер управління шпиндельним двигуном може перегрітися і вийти з ладу, що загрожує незворотною втратою даних.

Допустимі діапазони відхилень напруги (стандарт ATX):

- Лінія +3,3 В: у межах +3,14 ... +3,47 В;
- Лінія +5 В: у межах +4,75 ... +5,25 В;
- Лінія +5 VSB (чергова напруга): у межах +4,75 ... +5,25 В;
- Лінія +12 В: у межах +11,4 ... +12,6 В.

Відхилення напруги чергового джерела живлення (+5 VSB) нижче норми унеможливорює запуск основного інвертора блоку живлення, і комп'ютер не увімкнеться. Якщо ж напруга буде вищою за норму, це призведе до нестабільної

роботи («зависання») системи або пошкодження компонентів материнської плати.

Розглянемо необхідність моніторингу на прикладі жорсткого диску. Хоча механічні пошкодження іноді можна ідентифікувати за сторонніми звуками (тріск, свист, стукіт), у багатьох випадках деградація поверхні дисків або електроніки відбувається безшумно. Накопичувач може вийти з ладу раптово, без попередніх акустичних симптомів. Саме для запобігання таким ситуаціям та завчасного виявлення відхилень у роботі необхідне використання спеціалізованих апаратних та програмних засобів моніторингу (рис. 1.2).



Рис. 1.2 – Програмний засіб моніторингу ПК

1.3 Види та технічна реалізація моніторингу комп'ютерних систем

Сучасні системи моніторингу дозволяють у реальному часі відстежувати, протоколювати (через лог-файли) та візуалізувати наступні критичні параметри:

- Температурні режими: центрального процесора (CPU), графічного процесора (GPU), чіпсета материнської плати, жорстких дисків (HDD/SSD) та силових елементів блоку живлення.
- Електричні параметри: напруги на основних лініях живлення (+3.3V, +5V, +12V), напругу ядра процесора (Vcore) та модулів пам'яті (Vdim).

- Параметри систем охолодження: швидкість обертання вентиляторів (CPU, Chassis, GPU, Aux).

Хоча моніторинг прийнято поділяти на апаратний та програмний, фактично будь-який програмний засіб базується на зчитуванні даних з апаратних регістрів. Програмне забезпечення (ПЗ) візуалізує стани величин, звертаючись до регістрів спеціалізованих контролерів. У свою чергу, ці контролери отримують первинну інформацію від мережі датчиків. Таким чином, існує нерозривний зв'язок: ПЗ не може працювати без даних від контролера, а контролер не має власних засобів відображення інформації користувачеві без програмного інтерфейсу (або зовнішнього дисплея).

Загальна схема моніторингу виглядає наступним чином: аналоговий сигнал знімається з первинних перетворювачів (датчиків) і подається на входи контролера моніторингу. Основними типами датчиків є:

1. Термодатчики: терморезистори, термодіоди (часто інтегровані в кристал CPU/GPU) або транзистори в діодному включенні. Точність вимірювання суттєво залежить від типу датчика та його розміщення (наприклад, підсокетний датчик менш точний, ніж внутрішньокристальний).
2. Тахометричні датчики: зчитують імпульси від датчика Холла у вентиляторі.
3. Дільники напруги: для контролю рівнів напруг блоку живлення.

Вбудований у мікросхему моніторингу аналого-цифровий перетворювач (АЦП, зазвичай 8-бітний або вище) перетворює входні аналогові сигнали у двійковий код. Після оцифрування дані записуються у відповідні регістри, звідки їх зчитує BIOS або спеціалізоване ПЗ операційної системи.

Мікросхеми Super I/O (Multi I/O) У сучасних материнських платах функції моніторингу покладено на мультиконтролери, відомі як Super I/O. Це комплексні мікросхеми, які, окрім моніторингу, забезпечують роботу портів введення-виведення (COM, LPT, PS/2), керування вентиляторами та роботу з BIOS.

Історично використовувалися окремі чіпи (наприклад, серії LM78/79 від National Semiconductor), проте зараз стандартним рішенням є інтеграція функцій

у Super I/O від таких виробників, як Nuvoton (Winbond), ITE Tech, Fintek та SMSC.

В якості приклада розглянемо поширений мультиконтролер ITE IT8712F (рис. 1.3).

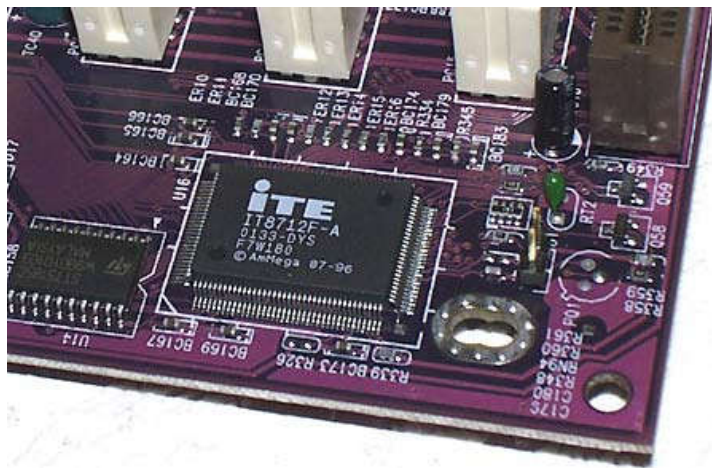


Рис. 1.3 – Мультіконтролер материнської плати ITE8712F

Ця мікросхема підключається до системи через шину LPC (Low Pin Count). Її функціональні можливості моніторингу включають:

- 8 входів для вимірювання напруги (VIN0–VIN7), що дозволяє контролювати основні лінії живлення та напругу батареї CMOS (Vbat).
- 3 входи для температурних датчиків (термісторів або термодіодів).
- 5 входів для тахометрів вентиляторів.
- SmartFan контролер: вбудований ШІМ-контролер (PWM) для автоматичного регулювання швидкості трьох вентиляторів залежно від температури.

Мікросхема (рис. 1.4) має апаратну логіку захисту: вона здатна автоматично фіксувати аварійні ситуації (наприклад, зупинку вентилятора або перевищення температурного порогу) та генерувати сигнал переривання або подавати звуковий сигнал через системний динамік (Beeper), навіть якщо операційна система зависла.

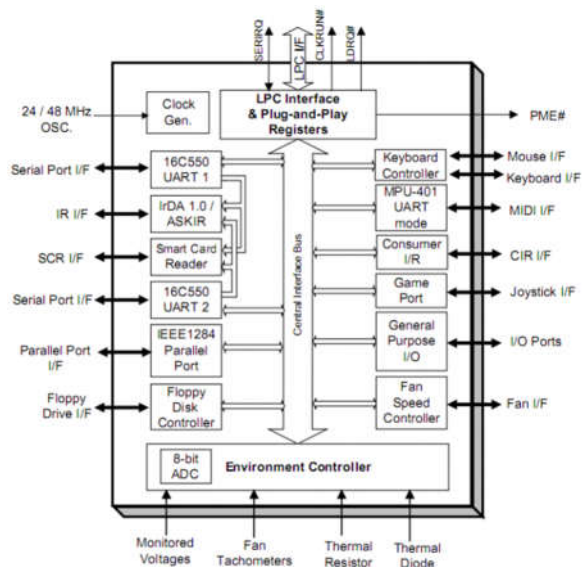


Рис. 1.4 – Схема мультіконтроллера ITE8712F

Окрім моніторингу, IT8712F забезпечує роботу застарілих, але важливих інтерфейсів: двох послідовних портів (UART), паралельного порту, контролерів клавіатури/миші (PS/2) та інтерфейсу флоппі-дисководу.

1.3.1 Особливості температурного моніторингу компонентів ПК

Ключовим завданням системи моніторингу є зняття точних показників температури з центрального процесора (CPU) та графічного прискорювача (GPU), оскільки саме ці компоненти мають найбільше тепловиділення. Перегрів кристала або помилкове спрацювання термозахисту є найпоширенішими причинами нестабільної роботи ПК, що призводить до аварійного вимкнення або перезавантаження системи. Особливо гостро ця проблема стоїть у портативних комп'ютерах (ноутбуках), де висока щільність компонування ускладнює тепловідведення, що з часом призводить до деградації BGA-пайки графічних чіпів та необхідності дорогого ремонту.

У сучасних мікропроцесорах термодатчики інтегровані безпосередньо в кристал (DTS — Digital Thermal Sensor). У багатоядерних архітектурах кожне ядро має власний датчик, що дозволяє отримувати дані з високою дискретністю без затримок.

Для захисту від термічних пошкоджень у процесорах Intel та AMD реалізовано багаторівневу систему безпеки:

1. Thermal Monitor (Тротлінг): Базується на сигналі PROCHOT#. Коли температура кристала досягає критичної межі ($T_{junction}$), активується механізм модуляції тактової частоти. Процесор починає пропускати робочі такти (вставляє «холості» цикли), що знижує його продуктивність та тепловиділення. Це дозволяє охолодити чіп без вимкнення системи навіть при пікових навантаженнях.
2. Аварійне вимкнення: Якщо тротлінг не допомагає і температура продовжує зростати (перевищує TCASEMAX приблизно на 20°C), активується сигнал THERMTRIP#. Це призводить до миттєвого зняття напруги живлення (VCC) з ядра та повного вимкнення комп'ютера для порятунку обладнання.

Сучасні графічні процесори (GPU) також оснащені розвиненою системою термоконтролю та захисту, аналогічною до CPU (динамічне зниження частот та напруги).

Не менш важливим є контроль температури жорстких дисків (HDD). Дослідження показують пряму кореляцію між робочою температурою та терміном служби накопичувача: перегрів прискорює деградацію механічних частин та магнітної поверхні. Отримати дані про температуру HDD можна, зчитавши відповідні атрибути системи самодіагностики S.M.A.R.T. (Self-Monitoring, Analysis and Reporting Technology).

Також слід враховувати нагрів системної логіки (чіпсета) та силових елементів ланцюгів живлення (VRM). Зростання пропускної здатності шин пам'яті та PCIe призводить до значного тепловиділення на "мостах" материнської плати. Якщо для рішень від Intel часто достатньо пасивних радіаторів, то високопродуктивні чіпсети (а також VRM при розгоні) можуть вимагати активного охолодження. Ігнорування температурних показників материнської плати може призвести до відвалу чіпів або виходу з ладу конденсаторів.

Рекомендовані температурні діапазони:

- CPU: $30\text{--}70^{\circ}\text{C}$ (залежно від архітектури та навантаження);

- Чіпсет/VRM: 30–60 °C;
- GPU: 40–80 °C (допускається короткочасне підвищення до 90°C для потужних моделей);
- HDD: 30–45 °C (температура вище 50°C вважається критичною).

1.3.2 Моніторинг електричних параметрів

Запорукою стабільної роботи обчислювальної системи є суворе дотримання номіналів напруг живлення всіх вузлів. Використання неякісних блоків живлення часто призводить до того, що під навантаженням напруга падає нижче допустимих меж («просідає»), що викликає збої в роботі програмного забезпечення, зависання ПК або мимовільні перезавантаження.

Для апаратного контролю цих параметрів лінії живлення підключаються до аналогових входів мультиконтролера (Super I/O). Оцифрування сигналу здійснюється за допомогою вбудованого аналого-цифрового перетворювача (АЦП).

Як правило, в бюджетних контролерах використовуються 8-розрядні АЦП. Діапазон вхідних напруг для них зазвичай обмежений опорною напругою (V_{ref}), яка часто становить 4,096 В. При 8-бітній розрядності ($2^8 = 256$ значень) крок квантування (точність вимірювання) становить:

$$\Delta U = \frac{4,096}{256} = 16 \text{ мВ}. \quad (1.1)$$

Оскільки номінали основних ліній живлення ПК (+5 В та +12 В) перевищують максимальну вхідну напругу АЦП (4,096 В), пряме підключення неможливе. Для вимірювання цих ліній використовуються резистивні дільники напруги (рис. 1.5), які пропорційно зменшують вхідний сигнал до безпечного рівня.

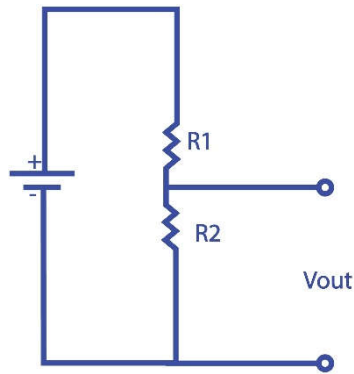


Рис. 1.5 – Резистивний дільник напруги

Для отримання достовірних даних критично важливим є узгодження вихідного опору джерела сигналу (датчика або дільника) з вхідним опором АЦП мікросхеми моніторингу. Для цього можуть застосовуватися додаткові узгоджувальні резистори або буферні каскади (схеми-повторювачі на операційних підсилювачах). Це дозволяє максимізувати співвідношення сигнал/шум та уникнути спотворення даних.

Похибка вимірювання часто виникає не через дефект мікросхеми, а через неточний підбір номіналів резисторів у дільниках або некоректну схемотехнічну реалізацію розведення плати. Саме тому користувачі іноді спостерігають у програмах моніторингу значення, що суттєво відрізняються від реальних показників мультиметра (рис. 1.6).

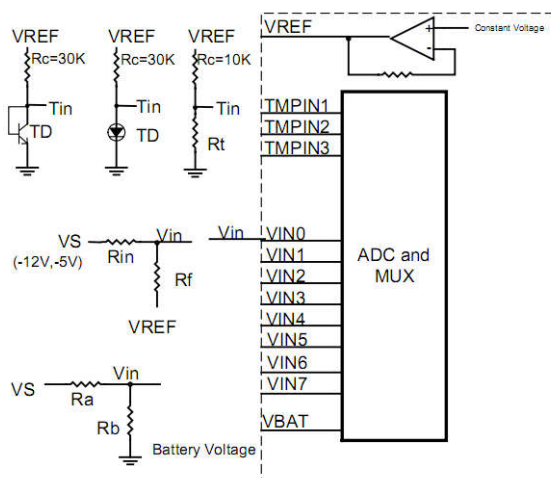


Рис. 1.6 – Схема АЦП

1.3.3 Моніторинг швидкості обертання вентиляторів

Моніторинг роботи активної системи охолодження є критично важливим елементом діагностики. Раптова зупинка вентилятора (внаслідок заклинювання підшипника або електричної несправності) призводить до лавиноподібного зростання температури, що загрожує термічним пошкодженням мікросхем.

Слід зазначити, що програмний моніторинг можливий лише для тих вентиляторів, які підключені безпосередньо до материнської плати через відповідні роз'єми (fan headers). Вентилятори, підключені безпосередньо до блоку живлення (через роз'єм Molex PATA), системою не визначаються.

Стандартна схема підключення, що дозволяє здійснювати моніторинг, вимагає наявності щонайменше трьох контактів (Рис. 1.7).

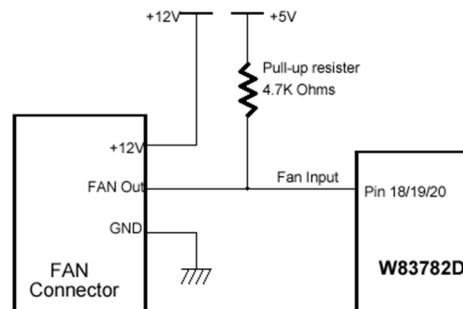


Рис. 1.7 – Принцип підключення вентиляторів

Типова розпіновка (pinout) та колірне маркування проводів наступні:

1. GND (Земля): чорний провід.
2. +12V (Живлення): червоний провід. Напруга на ньому може змінюватися материнською платою для регулювання обертів (DC-управління).
3. Sense/Tach (Тахометр): часто жовтий, білий або зелений провід. Через цей канал на мультиконтролер надходять імпульси від датчика Холла (зазвичай два імпульси на один повний оберт крильчатки).

Двопроводні вентилятори (тільки «+» та «-») не мають сигнальної лінії, тому відстежити їхню швидкість неможливо.

Сучасним стандартом є 4-контактні роз'єми. Доданий четвертий провід (Control/PWM, зазвичай синього кольору) дозволяє керувати швидкістю обертання методом широтно-імпульсної модуляції (ШІМ). При цьому напруга

живлення (+12В) залишається постійною, що забезпечує стабільніший крутний момент на низьких обертах.

Роз'єми 3-pin та 4-pin є механічно та електрично сумісними (завдяки ключу на конекторі), що дозволяє підключати нові системи охолодження до старих материнських плат і навпаки, хоча функціонал управління при цьому може бути обмежений.

1.3.4 Програмні засоби діагностики та моніторингу

Для візуалізації даних, отриманих від апаратних датчиків, використовується спеціалізоване прикладне програмне забезпечення. Його основним завданням є зчитування вмісту регістрів мультиконтролера (Super I/O), інтерпретація отриманих шістнадцяткових значень та виведення їх у зручному для користувача форматі (графіки, цифрові панелі, віджети).

Програмні засоби моніторингу можна класифікувати на дві групи:

1. Фірмові утиліти виробників (Vendor-specific) Постачаються в комплекті з материнськими платами або відеокартами (наприклад, ASUS AI Suite, MSI Center).

- Переваги: Найточніше відображення даних, оскільки ПЗ калібрується під конкретну схемотехніку плати; можливість керування системою охолодження з-під Windows.
- Функціонал: Дозволяють встановити порогові значення (Thresholds), при досягненні яких система сповіщає користувача про небезпеку.

2. Універсальне програмне забезпечення До цієї категорії належать такі популярні пакети, як AIDA64 (раніше Everest), HWiNFO та SpeedFan. Ці програми працюють з широким спектром контролерів, використовуючи власні бази даних обладнання.

Окремої уваги заслуговує утиліта SpeedFan. Попри те, що її розробка була призупинена, вона залишається актуальною для старіших систем та ентузіастів завдяки гнучким налаштуванням. Основні можливості SpeedFan:

- Робота з технологією S.M.A.R.T. для оцінки стану жорстких дисків.

- Висока точність та швидкість оновлення даних порівняно з громіздкими діагностичними пакетами.
- Система подій (Events): Реалізовано можливість програмування реакції системи на досягнення порогових значень. Користувач може налаштувати не лише звукове сповіщення, але й автоматичний запуск зовнішніх додатків, відправку повідомлень електронною поштою (e-mail) або примусове вимкнення ПК.

Варто зазначити, що будь-який програмний моніторинг має суттєві обмеження:

1. Залежність від операційної системи (ОС): моніторинг неможливий при зависанні ОС або на етапі завантаження BIOS/UEFI.
2. Споживання системних ресурсів: постійне опитування датчиків навантажує шину та процесор.
3. Помилки інтерпретації: універсальне ПЗ може некоректно визначати датчики на специфічних материнських платах.

Саме ці недоліки обґрунтовують доцільність створення зовнішнього апаратного комплексу моніторингу, який працює автономно та не залежить від стану програмного середовища комп'ютера.

1.4 Переваги та практичне значення моніторингу стану комп'ютерної системи

Впровадження систем постійного моніторингу дозволяє користувачеві в режимі реального часу отримувати об'єктивні дані про функціонування апаратного комплексу. Своєчасний аналіз цієї інформації дозволяє виявляти аномалії на ранніх стадіях та попереджати вихід обладнання з ладу, значно знижуючи ризики втрати даних та простою системи.

Основними підсистемами, що підлягають моніторингу, є:

- Центральний процесор (CPU): Ключовий елемент системи. Моніторингу підлягають: поточна температура кожного ядра, тактова частота, напруга

живлення (Vcore) та відсоток завантаження. Це дозволяє оцінити ефективність системи охолодження та наявність тротлінгу.

- Графічний прискорювач (GPU): Відстежуються температура графічного чіпа та пам'яті, швидкість обертання вентиляторів, частоти ядра/пам'яті, а також поточна роздільна здатність та частота оновлення дисплея.
- Оперативна пам'ять (RAM): Відображається інформація про тип (наприклад, DDR4/DDR5), робочу частоту, таймінги, загальний обсяг та поточний рівень заповнення пам'яті.
- Материнська плата: Крім ідентифікаційних даних (модель, версія BIOS), критично важливим є моніторинг температури системної логіки (чіпсета) та зони VRM (модуля регулювання напруги), а також напруг на основних лініях.
- Дискова підсистема (Storage): Розділ включає всі накопичувачі (HDD, SSD, NVMe). Крім температури та обсягу вільного місця, ключову роль відіграє технологія самодіагностики S.M.A.R.T., яка дозволяє виявити пошкоджені сектори (bad blocks), помилки читання/запису та оцінити залишковий ресурс SSD.
- Аудіо та Периферія: Відображення статусу підключених пристроїв вводу-виводу (клавіатури, миші, аудіоінтерфейси) та перевірка коректності роботи їх драйверів.

Незважаючи на велику кількість параметрів, найбільш критичними для надійності системи є показники центрального процесора, відеокарти та накопичувачів даних. CPU та GPU є найбільш термонавантаженими компонентами, схильними до перегріву. Накопичувачі є критичними з точки зору збереження інформації.

Інші компоненти (периферія, звукові карти) виходять з ладу значно рідше і їх несправність зазвичай не призводить до фатальних наслідків для всієї системи або втрати даних (рис. 1.8).

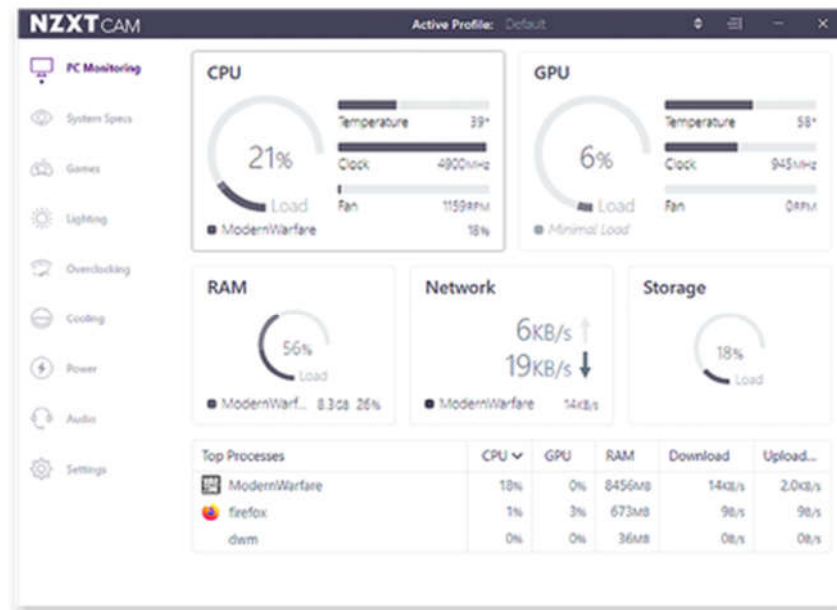


Рис. 1.8 – Програмний моніторинг ПК «NZXT CAM»

1.5 Рекомендації щодо контролю ключових технічних параметрів комп'ютерної системи

Для забезпечення довговічності та стабільності роботи комп'ютерної техніки необхідно дотримуватися регламенту моніторингу основних підсистем:

- **Центральний процесор (CPU):** Вимагає постійного контролю теплового режиму. Тривала робота при критичних температурах прискорює електроміграцію в кристалі, що призводить до незворотної деградації процесора. Вихід цього компонента з ладу тягне за собою значні фінансові витрати на ремонт або заміну платформи.
- **Графічний адаптер (GPU):** Аналогічно до центрального процесора, відеокарта є вразливою до перегріву. Рекомендується налаштувати профіль роботи вентиляторів так, щоб температура під навантаженням не перевищувала 75–80 °C. Також необхідно проводити регулярне технічне обслуговування (очищення від пилу, заміна термоінтерфейсу) не рідше одного разу на 1–2 роки.
- **Дискова підсистема (Storage):** Головним інструментом діагностики є моніторинг атрибутів S.M.A.R.T.. Сучасне програмне забезпечення автоматично аналізує «сирі» (raw) значення та інтерпретує їх у зрозумілий статус стану

накопичувача («Good», «Warning», «Critical»/«Bad»). Це дозволяє спрогнозувати відмову диска та вчасно зробити резервну копію даних.

Нехтування нормами експлуатації та регламентом технічного обслуговування (відсутність чистки, стара термопаста) неминуче призводить до апаратних збоїв.

Слід чітко розрізняти наслідки виходу з ладу різних компонентів:

- Поломка відеокарти або процесора — це фінансові збитки, які вирішуються заміною комплектуючих.
- Вихід з ладу накопичувача інформації (HDD/SSD) є критичним інцидентом, оскільки несе загрозу незворотної втрати даних. Відновлення інформації з пошкодженого носія є технічно складним, дорогим, а іноді й неможливим процесом.

Саме тому наявність незалежної системи моніторингу, яка попередить про небезпечні режими роботи ще до моменту аварії, є критично необхідною.

1.6 Аналіз існуючих апаратних аналогів

У даному підрозділі проведено огляд та порівняльний аналіз існуючих на ринку апаратно-програмних комплексів для моніторингу стану ПК. Метою аналізу є виявлення технічних особливостей, функціональних можливостей та недоліків комерційних рішень для врахування цього досвіду при розробці власної системи на базі Arduino.

Більшість сучасних рішень базуються на взаємодії зовнішнього дисплейного модуля з програмним забезпеченням в середовищі ОС Windows.

1.6.1 Апаратний комплекс Lamptron HC070

Lamptron HC070 — це класичний приклад моніторингу, реалізованого за принципом «другого монітора». Пристрій являє собою 7-дюймову IPS-матрицю, яка інтегрується в корпус комп'ютера.

Технічні особливості:

1. Інтерфейс підключення: Вимагає підключення двох кабелів — HDMI (для передачі відеосигналу) та USB (для живлення).
2. Програмна частина: Пристрій не має власного логічного контролера для обробки даних. Він працює у тандемі з популярним діагностичним ПЗ AIDA64 (функція SensorPanel).
3. Монтаж: Завдяки товщині корпусу 6 мм, дисплей може бути встановлений як всередині системного блоку (наприклад, на кожух блоку живлення або бічну стінку), так і розміщуватися на столі як зовнішній пристрій.

Недоліки рішення:

1. Займає відеовихід відеокарти (HDMI), що змушує графічний адаптер витрачати ресурси на рендеринг зображення для другого екрана.
2. Залежність від платного стороннього ПЗ (AIDA64).
3. Відсутність автономності (без завантаження ОС і драйверів відеокарти екран не працюватиме).



Рис. 1.9 – Зовнішній вигляд Lamptron HC070

1.6.2 Апаратний комплекс LCDSysInfo 3.5

На відміну від Lampton, LCDSysInfo 3.5 є більш спеціалізованим рішенням, що працює через USB-інтерфейс і не вимагає підключення до відеокарти. Система базується на програмному забезпеченні GOverlay. Пристрій оснащено 3.5-дюймовим дисплеєм з роздільною здатністю 480x320 пікселів. Взаємодія з ПК відбувається через власний драйвер, що дозволяє уникнути навантаження на графічний адаптер комп'ютера.

Ключові особливості ПЗ GOverlay:

- Кастомізація інтерфейсу: Користувач має можливість створювати власні профілі відображення («скіни»), використовуючи графічні примітиви (лінії, кола, гістограми), завантажувати фонові зображення та шрифти.
- Інтеграція даних: Система підтримує імпорт даних з більшості діагностичних утиліт: HWiNFO, AIDA64, OpenHardwareMonitor, SpeedFan тощо.
- Логічна обробка (Visual Scripting): Реалізовано функцію візуальних сценаріїв, що дозволяє налаштовувати поведінку датчиків (наприклад, змінювати колір тексту залежно від температури або конвертувати байти в мегабайти).
- Гейміфікація: Підтримка плагінів для відображення ігрової статистики (ранги, K/D ratio) у реальному часі.

У версії 3.5 було оптимізовано протокол передачі даних, що забезпечило вищу швидкість оновлення екрана (FPS) порівняно з попередніми версіями (2.8"). Підтримується використання альфа-каналу (прозорості) для накладання елементів інтерфейсу.

Недоліки рішення:

- Висока вартість пропрієтарного пристрою.
- Закрита екосистема (обмежена можливостями програми GOverlay).



Рис. 1.10 – Зовнішній вигляд LCDSysInfo 3.5

Проаналізовані аналоги демонструють два різні підходи: використання відеовиходу відеокарти (Lampttron) та передача даних через USB (LCDSysInfo). Для розробки дипломного проекту обрано концепцію, близьку до LCDSysInfo (передача даних через послідовний порт USB), оскільки це є більш ресурсоефективним рішенням. Однак, на відміну від дорогих комерційних аналогів, розроблюваний пристрій на базі Arduino матиме відкритий вихідний код, низьку собівартість та можливість гнучкої модифікації апаратної частини.

1.7 Аналіз існуючих програмних аналогів

Ринок спеціалізованих апаратних пристроїв для моніторингу є досить вузьким, а існуючі рішення часто мають високу вартість та закриту архітектуру. Тому для визначення необхідного функціоналу проектованої системи доцільно проаналізувати провідні програмні продукти, які є стандартом де-факто у сфері діагностики ПК.

1.7.1 Утиліта Piriform Speccy

Speccy – це популярна утиліта від розробників CCleaner, призначена для отримання детальної інформації про апаратне забезпечення та програмне середовище комп'ютера. Головною перевагою програми є лаконічний інтерфейс, що надає зведену інформацію (Summary) про всі ключові вузли системи.

- Operating System: Відображає версію Windows, налаштування безпеки та час роботи системи (Uptime).
- CPU / RAM / Motherboard: Надає дані про тактові частоти, таймінги пам'яті, ревізії плати та поточні температури.
- Storage: Дозволяє переглядати атрибути S.M.A.R.T. жорстких дисків, що дає змогу оцінити їх технічний стан («Good», «Warning», «Bad»).

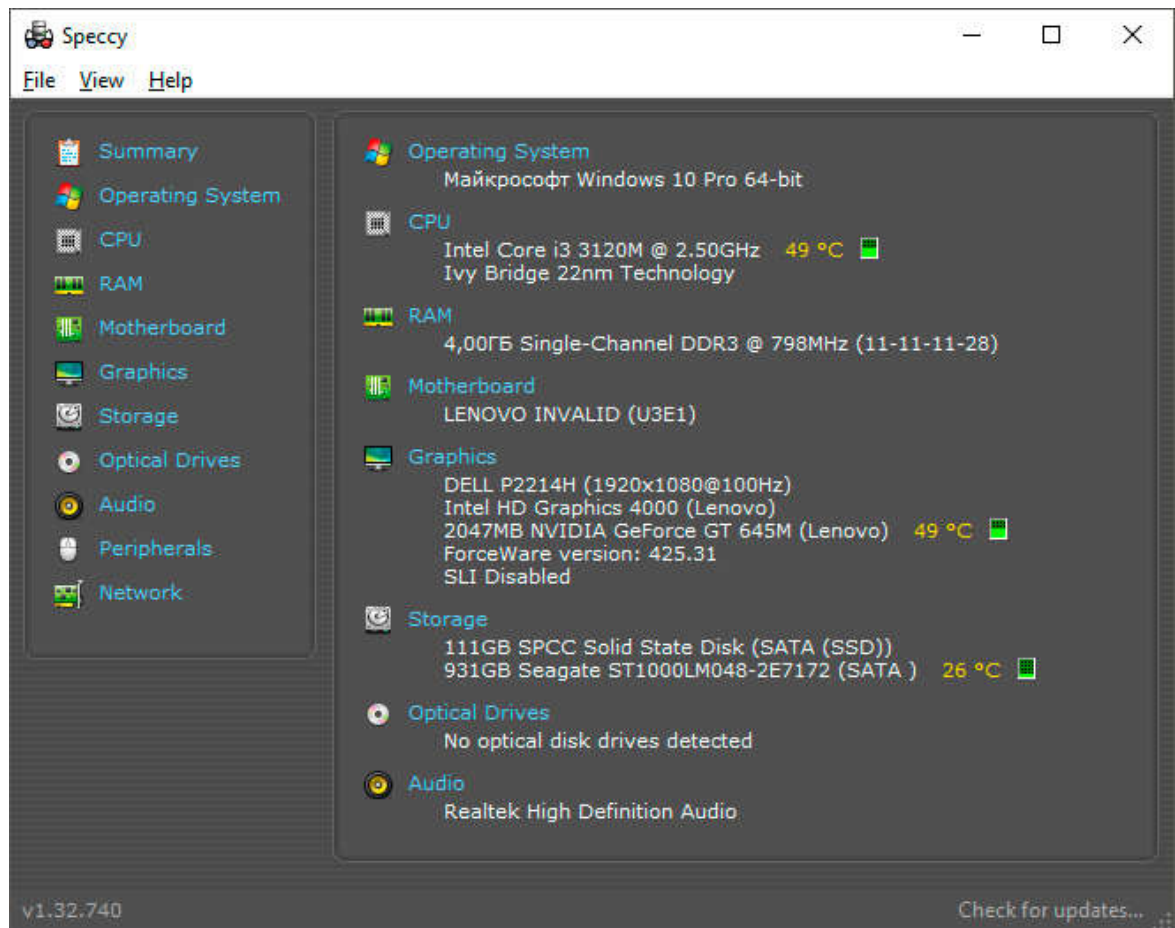


Рис. 1.11 – Інтерфейс програми Speccy

Спрессу орієнтована на отримання миттєвого «зрізу» інформації про систему, але має обмежені можливості щодо побудови графіків у реальному часі та логування даних. Крім того, розвиток утиліти на даний момент сповільнився.

1.7.2 Програмний комплекс AIDA64 Extreme

AIDA64 (спадкоємець Everest) є найбільш потужним інструментом для ідентифікації та тестування компонентів ПК. Програма використовує власну, одну з найбільших у світі, базу даних обладнання, що дозволяє їй розпізнавати новітні комплектуючі без оновлення драйверів

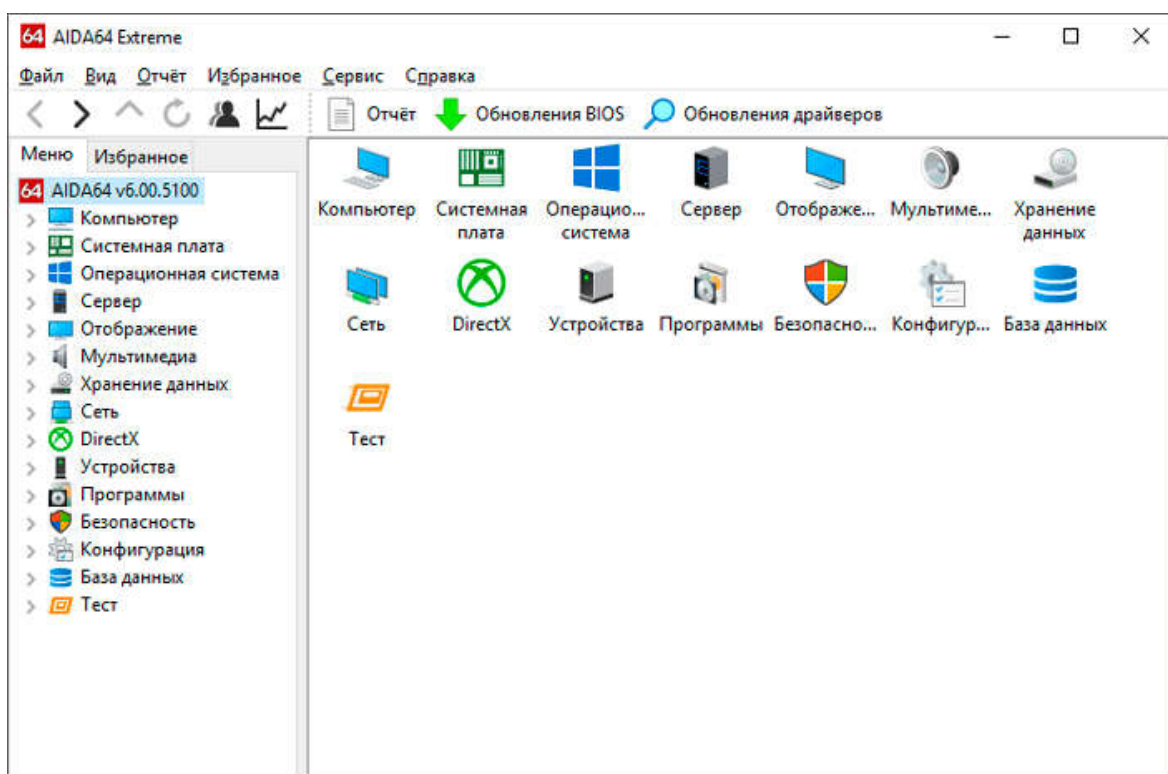


Рис. 1.12 – Інтерфейс AIDA64

Функціональні блоки AIDA64:

1. Датчики (Sensors): Це найбільш релевантний для даної роботи розділ. Він збирає показники з усіх доступних термісторів, тахометрів вентиляторів та датчиків напруги. Саме AIDA64 часто використовується як джерело даних (backend) для зовнішніх дисплеїв завдяки можливості експорту даних у Shared Memory або Registry.

2. Стрес-тестування (System Stability Test): Дозволяє створювати максимальне навантаження на CPU, FPU, кеш та оперативну пам'ять для перевірки стабільності системи охолодження та живлення.

3. Детальна інформація:

- Системна плата / ЦП: Детальні характеристики чіпсета, SPD модулів пам'яті, інструкції процесора.
- Відображення (Display): Інформація про графічний процесор (GPU), підтримку GPGPU/Vulkan/DirectX та характеристики монітора.
- Збереження даних (Storage): Розширена діагностика фізичних накопичувачів.

Проведений аналіз показує, що сучасні програмні засоби (зокрема AIDA64) надають вичерпну інформацію про стан ПК. Однак вони мають критичний недолік, властивий усім програмним рішенням: дані доступні лише тоді, коли операційна система завантажена і працює стабільно.

У випадку зависання ОС під час ігор або ресурсомістких завдань, користувач втрачає доступ до показників температур, і не може оцінити причину збою. Проектований апаратний комплекс на базі Arduino вирішує цю проблему, виводячи дані на зовнішній незалежний екран, що дозволяє контролювати параметри системи навіть у повноекранних додатках або у випадку програмного збою ОС.

Висновки за розділом 1

У першому розділі проведено детальний аналіз проблематики температурного та електричного моніторингу персональних комп'ютерів. За результатами дослідження можна зробити наступні висновки:

Встановлено, що стабільність роботи комп'ютерних систем безпосередньо залежить від ефективності відведення тепла та якості електроживлення. Основними факторами ризику є деградація термоінтерфейсів, запиленість радіаторів та нестабільність напруги блоку живлення. Нехтування цими

факторами призводить до тротлінгу, аварійних вимкнень, а в критичних випадках – до незворотної втрати даних через вихід з ладу накопичувачів.

Визначено, що сучасна архітектура ПК має розвинену мережу сенсорів. Температурні дані зчитуються з внутрішньокристальних датчиків (DTS), а напруги та оберти вентиляторів контролюються мікросхемами Super I/O (наприклад, ITE IT8712F). Розуміння принципів роботи АЦП та інтерфейсів підключення вентиляторів (3-pin/4-pin) є необхідним базисом для проектування власної системи.

Аналіз програмного забезпечення (AIDA64, Spessy) виявив суттєвий недолік: програмний моніторинг повністю залежить від працездатності операційної системи. У випадку зависання ОС («синій екран смерті», повноекранні ігри) користувач втрачає візуальний контакт з показниками датчиків і не може оперативно зреагувати на перегрів.

Огляд апаратних рішень (Lampticon, LCDSysInfo) показав, що існуючі комерційні продукти мають високу вартість, закриту архітектуру або використовують надлишкові ресурси ПК (підключення через HDMI як другий монітор).

На основі вищевикладеного сформульовано технічне завдання на розробку власного пристрою моніторингу. Найбільш доцільним є створення автономного апаратно-програмного комплексу на базі платформи Arduino. Такий підхід забезпечить:

- Незалежність індикації від стану ОС (дані передаються через послідовний порт).
- Низьку собівартість порівняно з промисловими аналогами.
- Можливість гнучкої кастомізації під конкретні потреби користувача.

Таким чином, розробка зовнішнього модуля моніторингу є актуальним науково-технічним завданням, що дозволяє підвищити надійність комп'ютерних систем.

РОЗДІЛ 2. ОБҐРУНТУВАННЯ ТА РОЗРОБКА АПАРАТНОЇ СТРУКТУРИ СИСТЕМИ МОНІТОРИНГУ

Етап проектування апаратно-програмного комплексу є визначальним для успішної реалізації проекту. На цьому етапі необхідно сформулювати загальну концепцію пристрою, обрати елементну базу та визначити логіку взаємодії компонентів.

Головною передумовою розробки власної системи стала економічна доцільність. Аналіз ринку, проведений у попередньому розділі, показав, що існуючі комерційні рішення мають невиправдано високу вартість та часто обмежені закритою архітектурою. Тому в основу розробки покладено принцип використання доступних модульних компонентів (платформа Arduino) та відкритого програмного коду, що дозволяє створити гнучку систему моніторингу, здатну працювати незалежно від основного програмного забезпечення ПК.

2.1 Аналіз функціональних вимог до системи

Функціональні вимоги визначають перелік сервісів та можливостей, які система повинна надавати користувачеві, а також її поведінку в різних режимах експлуатації. Для систематизації завдань проектування доцільно класифікувати вимоги за рівнем пріоритетності:

1. Критично важливі (Must Have): Базовий функціонал, без якого система не має сенсу.
2. Важливі (Should Have): Функції, реалізація яких є бажаною для повноцінної роботи.
3. Бажані (Could Have): Додаткові можливості, що розширюють зручність використання.

1. Критично важливі вимоги (Must Have): Система зобов'язана забезпечувати:

- Відображення системних даних: Отримання від ПК та вивід на дисплей інформації про температуру центрального (CPU) та графічного (GPU) процесорів, завантаженість ядер, обсяг використаної оперативної (RAM) та відеопам'яті.
- Автономний моніторинг: Вимірювання фізичних параметрів за допомогою зовнішніх незалежних сенсорів. Це включає контроль температури у довільних точках корпусу (термодатчики) та моніторинг споживання енергії (за допомогою безконтактних датчиків струму).
- Система захисту: Реалізація логіки автоматичного керування системою охолодження. Якщо температура зовнішнього датчика перевищує встановлений користувачем поріг (Critical Threshold), мікроконтролер повинен автоматично підвищити оберти підключених вентиляторів до максимуму або подати аварійний сигнал.

2. Важливі вимоги (Should Have): З метою підвищення зручності адміністрування, доцільно реалізувати:

- Бездротовий зв'язок: Інтеграція модулів Bluetooth або Wi-Fi (ESP8266/ESP32) для передачі телеметрії на мобільний пристрій (Android/iOS).
- Віддалений моніторинг: Можливість перегляду стану парку комп'ютерів (наприклад, в офісі або комп'ютерному класі) адміністратором мережі через єдиний веб-інтерфейс або мобільний додаток.

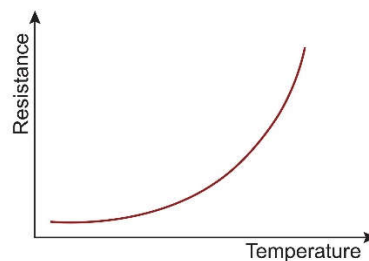
3. Бажані вимоги (Could Have): Для поглибленого аналізу роботи системи пропонується впровадити:

- Логування даних (Data Logging): Запис історії змін температур, напруг та обертів вентиляторів на зовнішній носій (SD-карту) у форматі CSV або TXT.
- Аналітика: Можливість експорту накопичених лог-файлів на ПК для подальшої обробки у табличних процесорах (наприклад, Microsoft Excel) з метою побудови графіків залежності температури від навантаження та часу.

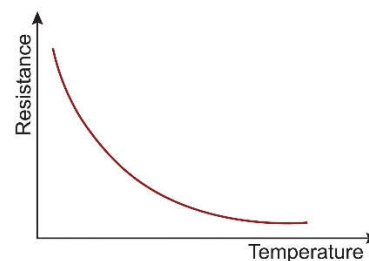
2.2 Аналіз методів вимірювання фізичних величин

Основою системи моніторингу є точне перетворення фізичних величин (температури, напруги, струму) в електричні сигнали, які може обробити мікроконтролер. Для цього необхідно визначити математичні моделі перетворення та оцінити вплив вимірювального кола на об'єкт дослідження.

Для вимірювання температури в комп'ютерних системах найчастіше використовуються NTC-термістори (Negative Temperature Coefficient). Їхня особливість полягає в нелінійній залежності опору від температури (рис. 2.1). Для точних вимірювань у широкому діапазоні проста лінійна апроксимація є неприйнятною.



PTC Thermistor



NTC Thermistor

Рис. 2.1 – Залежність опорів термісторів

Загальноприйнятим стандартом для опису характеристики напівпровідникових термісторів є рівняння Стейнхарта-Харта (Steinhart–Hart equation), яке забезпечує наближення третього порядку:

$$\frac{1}{T} = a + b \ln R + c (\ln R)^3 \quad (2.1)$$

де T – абсолютна температура (у Кельвінах); R – опір при температурі T (Ом); A , B , C – коефіцієнти Стейнхарта–Харта, які варіюються залежно від типу і моделі термістора і від діапазону температур.

Для зворотного розрахунку (знаходження опору за відомою температурою) використовується обернена форма рівняння:

$$R = \exp \left(\sqrt[3]{\left(y - \frac{x}{2}\right)} - \sqrt[3]{\left(y + \frac{x}{t}\right)} \right) \quad (2.2)$$

де параметри x та y визначаються через коефіцієнти A , B , C та температуру T :

$$x = \frac{1}{C} \left(A - \frac{1}{T} \right), \quad (2.3)$$

$$y = \sqrt{\left(\frac{B}{3C}\right)^3 + \left(\frac{x}{2}\right)^2} \quad (2.4)$$

У практичних застосуваннях на базі мікроконтролерів (зокрема Arduino) часто використовують спрощену модель (β -параметр), однак повне рівняння Стейнхарта–Харта забезпечує вищу точність (похибка менше 0.02 °C у робочому діапазоні).

2.2.1 Методи вимірювання електричних величин

У сучасній вимірювальній техніці розрізняють два основні методи:

1. Метод безпосередньої оцінки: Значення величини визначається безпосередньо за показами вимірювального пристрою (у цифрових системах – через АЦП), попередньо відградуйованого у відповідних одиницях. Це найбільш оперативний метод, який і буде використано у розроблюваному пристрої.
2. Метод порівняння: Вимірювана величина (X) порівнюється з еталонною величиною (X_{ref}). Процедура зводиться до отримання різниці сигналів на вході компаратора. Цей метод є точнішим, але складнішим у реалізації.

2.2.2 Вплив вимірювальних приладів на об'єкт дослідження

При підключенні вимірювального пристрою до кола неминує виникає методична похибка взаємодії. Вольтметр підключається паралельно до ділянки кола, а амперметр — послідовно (Рис. 2.2).

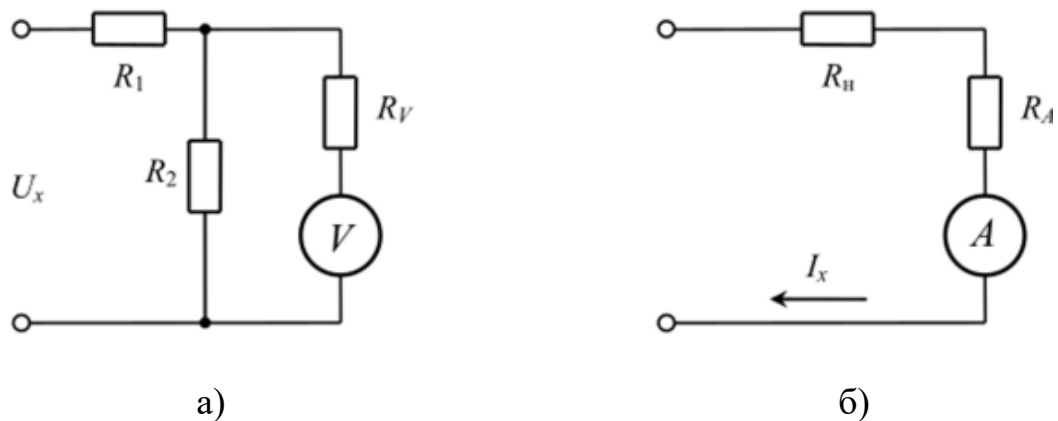


Рис. 2.2 – Схема підключення приладів методом безпосередньої оцінки:

а – вольтметра; б – амперметра

Вимірювання напруг і струму завжди супроводжується похибкою, величина якої залежить від внутрішнього опору вольтметра R_v амперметра R_A . Включення вольтметра і амперметра в досліджувану ланцюг спотворює режим роботи цього ланцюга.

Наприклад, напруга на резисторі R_2 до включення вольтметра

$$U_{R_2} = \frac{UR_2}{R_1 + R_2} \quad (2.5)$$

Напруга на цьому ж резисторі після підключення вольтметра

$$U_{R_2}^* = \frac{UR_2 R_v / (R_2 + R_v)}{R_1 + R_2 R_v / (R_2 + R_v)} \quad (2.6)$$

Абсолютна похибка вимірювання в даному випадку тим більше, чим менше опір вольтметра:

$$\Delta = U_{R_2}^* - U_{R_2} \quad (2.7)$$

$$\delta = \frac{U_{R_2}^* - U_{R_2}}{U_{R_2}} = - \frac{R_2 / R_v}{1 + R_2 / R_v + R_2 / R_1} \quad (2.7)$$

Аналіз показує, що похибка тим більша, чим менший внутрішній опір вольтметра R_v відносно опору ділянки кола R_2 . Аналогічно, при вимірюванні струму падіння напруги на шунті амперметра вносить похибку в роботу навантаження.

Для мінімізації цих похибок у проекті будуть використані високоомні вхідні ділянки напруги та низькоомні шунти (або датчики Холла) для вимірювання струму.

2.3 Обґрунтування вибору та опис апаратної платформи Arduino

Arduino — це відкрита апаратна обчислювальна платформа для прототипування, основними компонентами якої є плата введення-виведення та середовище розробки (Processing/Wiring). Мова програмування платформи є спрощеною підмножиною C/C++, що значно прискорює процес розробки вбудованих систем. Arduino може використовуватися як для створення автономних інтерактивних об'єктів, так і для взаємодії з програмним забезпеченням на ПК.

Для реалізації дипломного проекту обрано плату Arduino Nano V3.0 (рис. 2.3). Це повнофункціональний мініатюрний пристрій на базі мікроконтролера ATmega328P (рис.2.4), адаптований для використання з макетними платами (breadboard). За функціональністю пристрій аналогічний платі Arduino Uno, але відрізняється компактними розмірами, відсутністю винесеного гнізда зовнішнього живлення та використанням інтерфейсу Mini-B (або Type-C у нових ревізіях) USB.

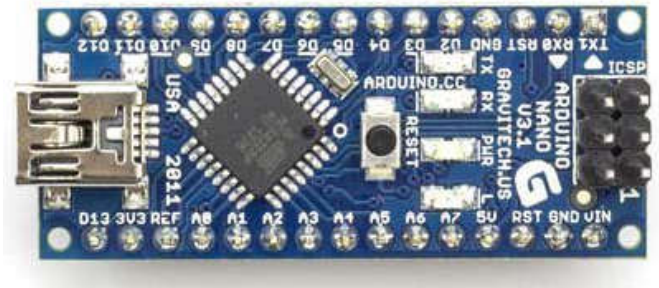
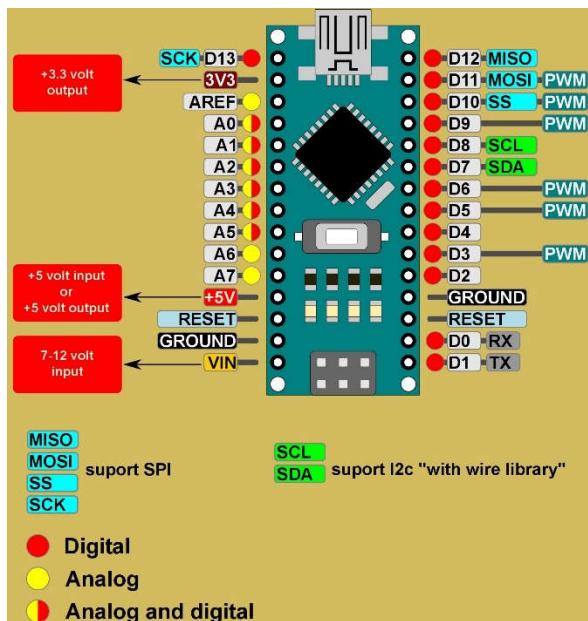
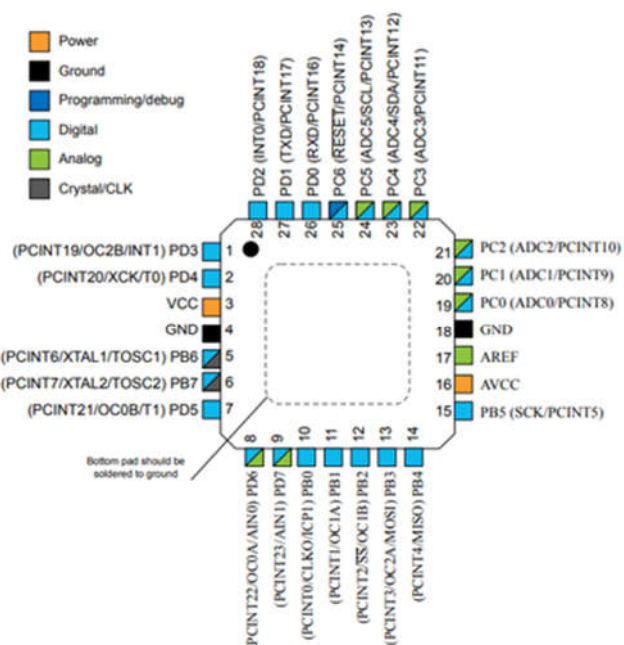


Рис. 2.3 – Зовнішній вигляд Arduino Nano

2.3.1 Архітектура та апаратна частина

Ядром платформи є 8-бітний мікроконтролер архітектури AVR – Atmel ATmega328P (рис. 2.4).



Тактування здійснюється кварцовим резонатором на частоті 16 МГц. У пам'ять мікроконтролера попередньо записаний завантажувач (bootloader), що дозволяє оновлювати прошивку через USB-інтерфейс без використання зовнішніх програматорів.

Зв'язок з комп'ютером здійснюється через перетворювач інтерфейсів USB-UART (на оригінальних платах це FTDI FT232RL, на аналогах — CH340G).

Arduino Nano підтримує три режими живлення:

1. Через USB-кабель (5 В).
2. Від нестабілізованого зовнішнього джерела 6–20 В (через пін VIN або 30).
3. Від стабілізованого джерела 5 В (через пін 5V або 27).

Пристрій автоматично обирає джерело живлення з найвищою напругою. Слід зазначити особливість роботи виходу 3.3V: на платах з чіпом FTDI напруга 3.3 В формується внутрішнім регулятором цієї мікросхеми, тому вона доступна лише при підключенні USB. При живленні від зовнішнього джерела цей вихід може бути неактивним.

Мікроконтролер ATmega328P має таку структуру пам'яті:

- Flash-пам'ять: 32 КБ (з яких 2 КБ відведено під завантажувач), використовується для зберігання коду програми.
- SRAM (ОЗП): 2 КБ — для зберігання змінних під час виконання програми.
- EEPROM: 1 КБ — енергонезалежна пам'ять для зберігання налаштувань.

2.3.2 Інтерфейси введення-виведення

Плати Arduino дозволяють гнучко налаштовувати порти GPIO. Arduino Nano має:

- 14 цифрових входів/виходів (D0–D13): Кожен з них може бути налаштований як вхід або вихід. Виводи D3, D5, D6, D9, D10 та D11 підтримують широтно-імпульсну модуляцію (ШІМ / PWM), що необхідно для керування швидкістю вентиляторів.

- 8 аналогових входів (A0–A7): Використовуються для зчитування сигналів з температурних датчиків та дільників напруги. АЦП має розрядність 10 біт (1024 значення).

Таблиця 2.1

Технічні характеристики Arduino Nano 3.0

Параметр	Значення
Мікроконтролер	Atmel ATmega328P
Робоча напруга (логічний рівень)	5 В
Вхідна напруга (рекомендована)	7–12 В
Вхідна напруга (гранична)	6–20 В
Цифрові Входи/Виходи	14 (з них 6 ШІМ)
Аналогові входи	8
Максимальний струм одного виводу	40 мА
Flash-пам'ять	32 КБ
SRAM	2 КБ
EEPROM	1 КБ
Тактова частота	16 МГц
Розміри плати	18.5 x 43.2 мм

2.4 Обґрунтування вибору та опис модулю дисплея

Для відображення діагностичної інформації (температур, напруг, станів вентиляторів) обрано символний рідкокристалічний дисплей (LCD) формату 1602 (рис. 2.5).

Дисплей побудований на базі контролера Hitachi HD44780 (або його сумісних аналогів). Маркування «1602» вказує на його інформаційну ємність: можливість відображення 16 символів у 2 рядки. Основні переваги даного модуля:

- Низьке енергоспоживання.
- Висока контрастність символів за рахунок вбудованого LED-підсвічування.
- Широка підтримка у середовищі Arduino IDE (бібліотека LiquidCrystal).



Рис. 2.5 – Зовнішній вигляд LCD 1602

Однак стандартне підключення дисплея через паралельний інтерфейс має суттєвий недолік: для керування вимагається задіяння щонайменше 6 цифрових портів мікроконтролера (RS, E, D4, D5, D6, D7), не враховуючи живлення. Для плати Arduino Nano, яка має обмежену кількість виводів, це створює дефіцит портів для підключення інших датчиків.

Для оптимізації використання портів мікроконтролера застосовано додатковий модуль розширення портів (рис. 2.6) на базі мікросхеми PCF8574T (або PCF8574AT). Цей модуль перетворює паралельні сигнали дисплея у послідовний інтерфейс I2C (Inter-Integrated Circuit).

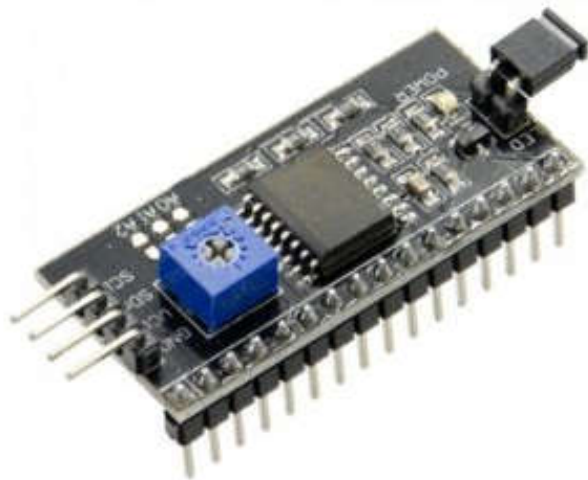


Рис. 2.6 – Модуль I2C на базі PCF8574

Використання I2C-адаптера надає наступні переваги:

1. Економія портів: Замість 6–10 проводів використовуються лише 2 сигнальні лінії (SDA та SCL).
2. Адресація: Можливість підключення декількох пристроїв на одну шину (стандартні адреси модуля: 0x27 або 0x3F).
3. Зручність налаштування: На платі модуля розміщено підлаштувальний резистор (потенціометр) для апаратного регулювання контрастності символів без використання додаткових зовнішніх компонентів.

Комутація дисплея з платою Arduino здійснюється за чотирипровідною схемою (Рис. 2.7)

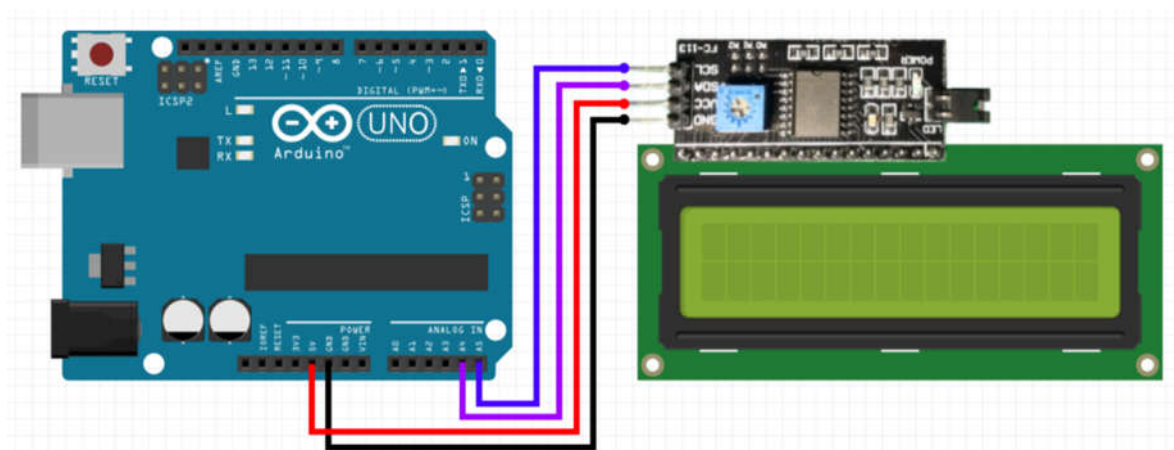


Рис. 2.7 – Підключення LCD-дисплея до Arduino

Таблиця 2.2

Розпіновка підключення по шині I2C

Вивід модуля I2C	Функція	Підключення до Arduino
GND	Загальний (Земля)	GND
VCC	Живлення	+5V
SDA (Serial Data)	Лінія даних	Пін A4
SCL (Serial Clock)	Лінія тактування	Пін A5

Програмна реалізація виводу даних здійснюється за допомогою бібліотеки LiquidCrystal_I2C, яка дозволяє ініціалізувати дисплей за його HEX-адресою та

виводити текстову інформацію, використовуючи стандартні методи обробки рядків.

2.5 Обґрунтування вибору та опис датчика температури DS18B20

Для реалізації функції термоконтролю обрано цифровий датчик DS18B20 від компанії Maxim Integrated (Dallas Semiconductor). Вибір саме цифрового сенсора обумовлений необхідністю передачі даних на значну відстань без втрати точності, що є проблематичним для аналогових терморезисторів через опір довгих проводів.

Технічні характеристики DS18B20 (рис. 2.8) – це високоточний цифровий термометр, що забезпечує вимірювання температури в градусах Цельсія. Основні параметри:

- Діапазон вимірювань: від -55°C до $+125^{\circ}\text{C}$.
- Точність: $\pm 0.5^{\circ}\text{C}$ (у найбільш актуальному для ПК діапазоні від -10°C до $+85^{\circ}\text{C}$).
- Напруга живлення: 3.0 В – 5.5 В.
- Роздільна здатність: Програмується користувачем і становить від 9 до 12 біт.
- Час перетворення: При максимальній роздільній здатності (12 біт) час конвертації становить 750 мс.



Рис. 2.8 – Зовнішній вигляд датчика DS18B20 у герметичному корпусі

Ключовою особливістю DS18B20 є використання протоколу 1-Wire. Це дозволяє підключати необмежену кількість датчиків до одного цифрового піна мікроконтролера, використовуючи лише один сигнальний провід (та "землю").

Кожен датчик має унікальний заводський 64-розрядний серійний код (ROM-код), що дозволяє мікроконтролеру звертатися до конкретного пристрою на шині індивідуально. Це ідеально підходить для архітектури моніторингу ПК, де потрібно одночасно вимірювати температуру в різних зонах (наприклад, "Вдув", "Видув", "Відеокарта", "Радіатор"), використовуючи мінімум портів Arduino.

Датчик може працювати у двох режимах живлення:

1. Зовнішнє живлення: Використовуються 3 провода (VCC, GND, Data). Це найбільш стабільний режим.
2. Паразитне живлення (Parasitic Power): Живлення береться безпосередньо з лінії даних, що дозволяє використовувати лише 2 провода.

Обов'язковою умовою роботи шини 1-Wire є використання підтягуючого резистора (Pull-up) номіналом 4.7 кОм, включеного між лінією даних (DQ) та живленням (VCC). Без цього резистора обмін даними буде неможливим через особливості схемотехніки виходу "відкритий колектор".

Датчик також має вбудовану енергонезалежну пам'ять (EEPROM) для зберігання налаштувань користувача, зокрема порогів спрацьовування аварійної сигналізації (верхня та нижня межі температури).

2.6 Обґрунтування вибору та опис датчика струму CSLA1CD

Для розрахунку споживаної потужності ПК ($P = U \times I$) необхідно вимірювати силу струму, що протікає через провідники блоку живлення (зокрема, лінію +12В, яка є найбільш навантаженою). Для реалізації цього завдання обрано датчик CSLA1CD виробництва Honeywell (рис. 2.9).

На відміну від шунтових методів вимірювання, які вимагають розриву електричного кола, датчик CSLA1CD базується на ефекті Холла. Це безконтактний лінійний датчик відкритого типу (Open-Loop). Конструктивно пристрій складається з:

1. Магнітопроводу (кільця): Концентрує магнітне поле, що створюється провідником зі струмом, пропущеним крізь отвір датчика.
2. Датчика Холла (LOHET): Інтегральна схема (типу 91SS12-2 або SS94A1), розміщена у зазорі магнітопроводу. Вона перетворює магнітну індукцію в електричний сигнал.



Рис. 2.9 – Зовнішній вигляд датчика струму CSLA1CD

Вихідна напруга датчика змінюється лінійно та пропорційно силі струму в провіднику. Датчик є раціометричним: його чутливість та вихідна напруга спокою (Offset) залежать від напруги живлення. Це дозволяє підключати його безпосередньо до АЦП мікроконтролера.

Переваги обраного датчика:

- Гальванічна розв'язка: Вимірювальне коло електрично ізольоване від силового, що гарантує безпеку мікроконтролера при високих навантаженнях.
- Відсутність падіння напруги: Оскільки вимірювання відбувається безконтактно, датчик не вносить опір у силове коло і не гріється, на відміну від шунтів.
- Вимірювання AC/DC: Здатність вимірювати як постійний, так і змінний струм.

Корпус датчика виготовлено з поліестеру (PBT). Монтаж здійснюється на друковану плату (PCB) через 3-контактний роз'єм: Живлення (+), Земля (-), Вихід (Out). Провідник, струм у якому необхідно виміряти, просто пропускається крізь отвір у корпусі.

Таблиця 2.3

Технічні характеристики Honeywell CSLA1CD

Параметр	Значення
Тип вимірювання	Безконтактний, на ефекті Холла
Вихідний сигнал	Лінійний, аналогова напруга
Максимальний вимірюваний струм	57 А
Напруга живлення	8 ... 16 В (постійного струму)
Чутливість	49.6 мВ / А
Час відгуку	3 мкс
Діапазон робочих температур	-25 ... +85 °C
Споживаний струм	макс. 19 мА
Тип корпусу	Наскрізний монтаж (Through-hole)

Оскільки номінальна напруга живлення датчика становить 8–16 В (рекомендовано 12 В від блоку живлення ПК), а вихідний сигнал може перевищувати 5 В при максимальному струмі, для підключення до входу Arduino (який розрахований на макс. 5 В) необхідно використовувати резистивний дільник напруги на виході датчика. Це дозволить узгодити рівні сигналів та захистити вхід АЦП мікроконтролера.

2.7 Схемотехніка гальванічної розв'язки інтерфейсу 1-Wire

Оскільки датчики DS18B20 можуть розміщуватися на значній відстані від мікроконтролера, часто виникає необхідність захисту вхідних портів Arduino від електромагнітних завад, статичної електрики або потрапляння високої напруги при аваріях на лініях. Для цього застосовується схема гальванічної розв'язки.

Основна складність реалізації ізоляції для протоколу 1-Wire полягає в його двонаправленості по одному дроту (Half-Duplex). Більшість стандартних схем на оптронах працюють лише в одному напрямку (Simplex), тому для 1-Wire необхідна спеціалізована симетрична схема (рис. 2.10).

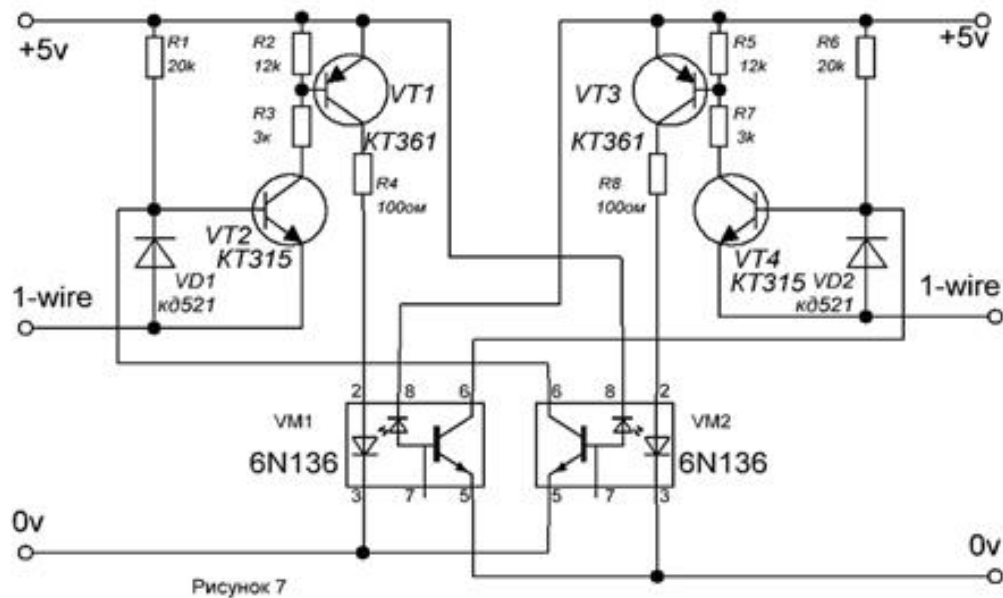


Рис. 2.10 – Принципова схема гальванічної розв'язки інтерфейсу 1-Wire

Вимоги до компонентів схеми:

1. Швидкодія: Стандартні оптопари (типу PC817) мають занадто повільні фронти перемикання для таймінгів 1-Wire (тривалість імпульсу може становити 10–15 мкс). Тому необхідно використовувати високошвидкісні оптрони, наприклад 6N136 або 6N137 (швидкість передачі до 1–10 Мбіт/с).
2. Струми: Схема повинна коректно працювати зі слабкими струмами підтяжки лінії (1–5 мА).

Схема складається з двох симетричних вузлів. Умовно позначимо ліву частину як Master (сторона мікроконтролера), а праву – як Slave (сторона датчиків).

- VT1, VT2 та VT3, VT4 – транзисторні ключі для керування світлодіодами оптронів.
- VM1, VM2 – високошвидкісні оптрони (6N136).

- VD1, VD2 – діоди Шотткі (або швидкодіючі діоди типу 1N4148) для розділення сигналів.

Логіка роботи:

1. Режим спокою (Idle): У цьому стані обидві лінії підтягнуті до високого рівня (+5В) резисторами. Транзистори VT2 та VT4 закриті. Відповідно, струм через світлодіоди оптронів не протікає, фототранзистори оптронів закриті. Система перебуває у стані логічної «1».
2. Передача даних від Master до Slave (Запис): Мікроконтролер притягує лінію до землі (формує логічний «0»).
- Відкривається транзистор VT2.
- Струм протікає через світлодіод оптрона VM1.
- Відкривається вихідний транзистор оптрона VM1, який замикає лінію Slave на землю (через діод VD2).
- Датчик фіксує низький рівень (імпульс Reset або запис біта «0»).
3. Передача даних від Slave до Master (Читання): Датчик (Slave) притягує лінію до землі (наприклад, сигнал Presence Pulse).
- Низький рівень на стороні Slave відкриває транзистор VT4.
- Вмикається оптрон VM2.
- Вихід оптрона VM2 замикає лінію Master на землю.
- Мікроконтролер зчитує логічний «0».

Така схемотехніка забезпечує повну гальванічну ізоляцію до 2.5 кВ (залежно від типу оптрона) та дозволяє безпечно використовувати датчики у промислових умовах або на вулиці.

Висновки за розділом 2

У другому розділі проведено комплексне обґрунтування та проектування апаратної архітектури системи моніторингу ПК. На основі аналізу функціональних вимог та технічних характеристик компонентної бази отримано наступні результати:

В якості керуючого ядра системи обрано платформу Arduino Nano V3.0 на базі мікроконтролера ATmega328P. Цей вибір забезпечує оптимальний баланс між продуктивністю, компактністю та вартістю, а також надає достатню кількість інтерфейсів (UART, I2C, PWM) для підключення периферії.

Для термометрії обрано цифрові датчики DS18B20, що працюють по протоколу 1-Wire. Їх перевагами є висока завадостійкість, точність ($\pm 0.5^{\circ}\text{C}$) та можливість адресації декількох датчиків на одній лінії.

Для контролю енергоспоживання обрано безконтактний датчик струму CSLA1CD на ефекті Холла, який забезпечує гальванічну розв'язку силового кола від вимірювального, що критично важливо для безпеки мікроконтролера.

Для візуалізації даних обрано LCD-дисплей формату 1602 з I2C-адаптером на базі чіпа PCF8574. Таке рішення дозволило скоротити кількість задіяних портів мікроконтролера з 6 до 2, звільнивши ресурси для датчиків.

Розроблено схемотехнічне рішення для гальванічної розв'язки інтерфейсу 1-Wire з використанням високошвидкісних оптронів. Це дозволяє використовувати виносні датчики в умовах електромагнітних завад та захищає порти Arduino від можливих стрибків напруги.

Таким чином, у даному розділі повністю сформовано апаратну структуру пристрою, визначено методи вимірювання фізичних величин та обґрунтовано вибір елементної бази. Спроектowana апаратна частина готова до наступного етапу – розробки електричної принципової схеми та програмного забезпечення.

РОЗДІЛ 3. РОЗРОБКА АПАРАТНО-ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ПЛАТФОРМИ ARDUINO

У третьому розділі розглядається процес проектування та реалізації програмної складової системи моніторингу. Основна увага приділяється вибору алгоритмів обробки даних, опису середовища розробки, а також програмній реалізації взаємодії між апаратною частиною (мікроконтролером) та персональним комп'ютером.

Для розуміння структури проекту необхідно визначити базові поняття:

- Апаратне забезпечення (Hardware): Фізична складова електронного пристрою (мікроконтролер, датчики, дисплей), що відповідає за виконання низькорівневих операцій та взаємодію з фізичним світом.
- Програмне забезпечення (Software): Сукупність програм, процедур та правил, що керують роботою апаратних засобів.

Процес розробки програмного забезпечення (Software Engineering) включає етапи проектування архітектури, написання вихідного коду, тестування, відлагодження та документування.

3.1 Життєвий цикл розробки програмного забезпечення

Життєвий цикл програмного забезпечення (SDLC — Software Development Life Cycle) — це період часу, що починається з моменту прийняття рішення про необхідність створення програмного продукту і закінчується в момент його повного вилучення з експлуатації.

Для реалізації даного дипломного проекту обрано Каскадну модель (Waterfall Model). Це класична методологія, в якій процес розробки виглядає як потік, що послідовно проходить через фази аналізу вимог, проектування, реалізації, тестування, інтеграції та підтримки.

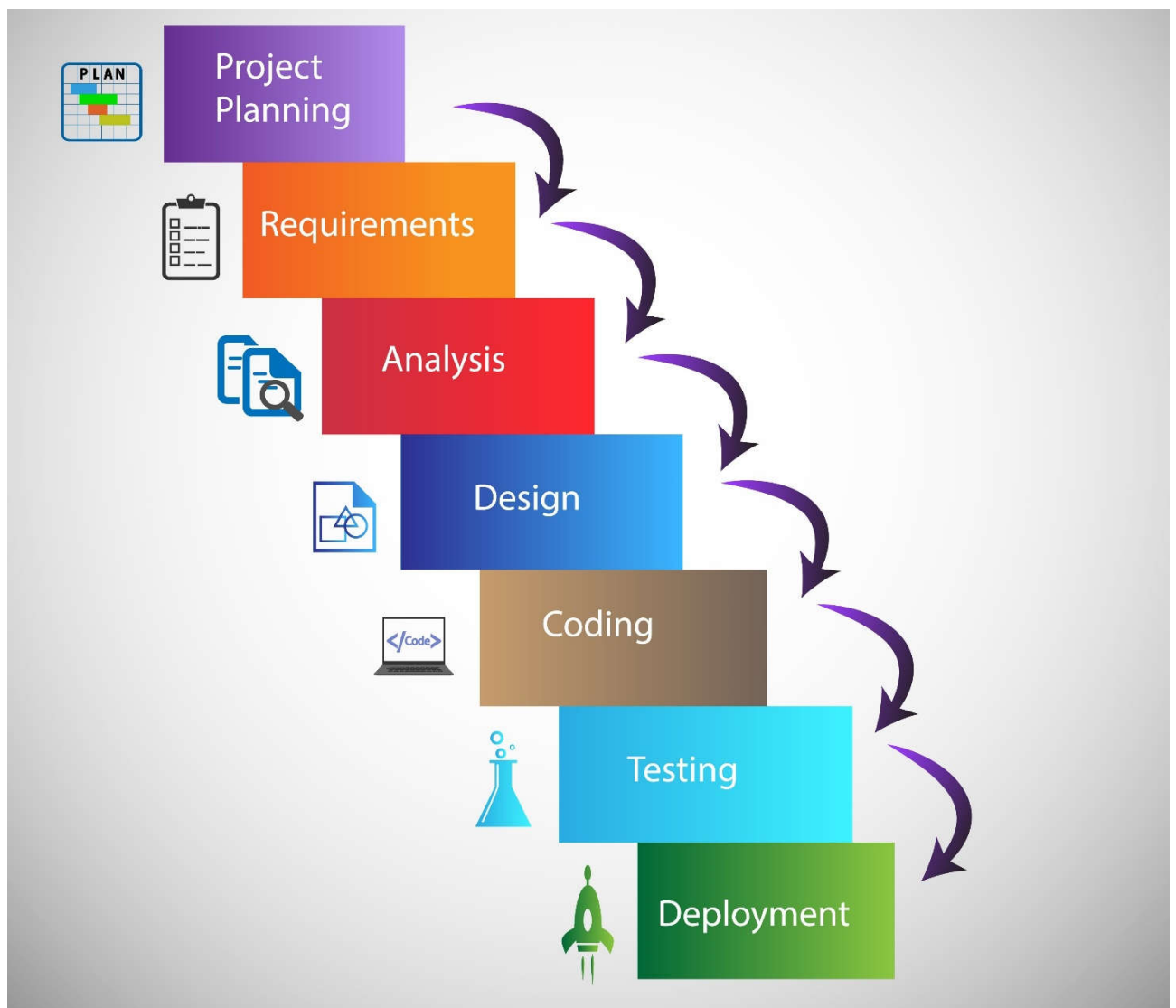


Рис. 3.1 – Каскадна модель життєвого циклу розробки

Основні етапи каскадної моделі в контексті даної роботи:

1. Аналіз вимог: Визначення параметрів, що підлягають моніторингу (температура, напруга), та вибір необхідних датчиків (проведено у Розділі 2).
2. Проектування: Розробка блок-схеми алгоритму роботи пристрою та архітектури програмного коду.
3. Реалізація (Кодування): Написання скетчу для Arduino мовою C++ та розробка інтерфейсу.
4. Тестування: Перевірка коректності зчитування даних із сенсорів та виведення їх на дисплей.

5. Експлуатація та підтримка: Встановлення пристрою в корпус ПК та його використання.

Переваги каскадної моделі:

- Чітка структура: Суворі послідовність етапів дозволяє точно планувати терміни виконання робіт.
- Визначеність вимог: Вимоги до системи фіксуються на початку проекту і залишаються незмінними, що спрощує проектування архітектури.
- Документованість: Кожен етап завершується формуванням звітної документації, що ідеально підходить для написання кваліфікаційної роботи.

Недоліки моделі:

- Низька гнучкість: Внесення змін у проект на пізніх етапах (наприклад, заміна типу датчика після написання коду) є складним і вимагає повернення до попередніх фаз.
- Пізнє тестування: Працездатний прототип з'являється лише на етапі реалізації, що підвищує ризик виявлення критичних помилок у кінці розробки.

Незважаючи на недоліки, для індивідуального проекту з чітко сформульованими технічними завданнями каскадна модель є найбільш оптимальною.

3.2 Проектування та макетування апаратної частини

3.2.1 Розробка принципової схеми підключення

На основі обґрунтованої у другому розділі елементної бази розроблено електричну схему з'єднань компонентів системи. Проектування здійснювалося у середовищі автоматизованого проектування (CAD) EasyEDA, що дозволило візуалізувати комутацію модулів перед фізичним монтажем.

До платформи Arduino Nano підключаються наступні вузли:

1. Датчик температури DS18B20: Підключається до цифрового порту з використанням підтягуючого резистора 4.7 кОм.

2. Датчик струму CSLA1CD: Підключається до аналогового входу через дільник напруги (для узгодження рівнів 12В/5В).
3. LCD-дисплей 1602: Підключається через I2C-конвертер до шини I2C (піни A4, A5).
4. Елементи керування та індикації: Тактові кнопки для навігації по меню та RGB-стрічка для візуалізації статусів (норма/аварія).

Для зручності монтажу та подальшого обслуговування пристрою на схемі використано кольорове маркування провідників відповідно до загальноприйнятих стандартів (Червоний – VCC, Чорний – GND, Жовтий/Зелений – Сигнальні лінії).

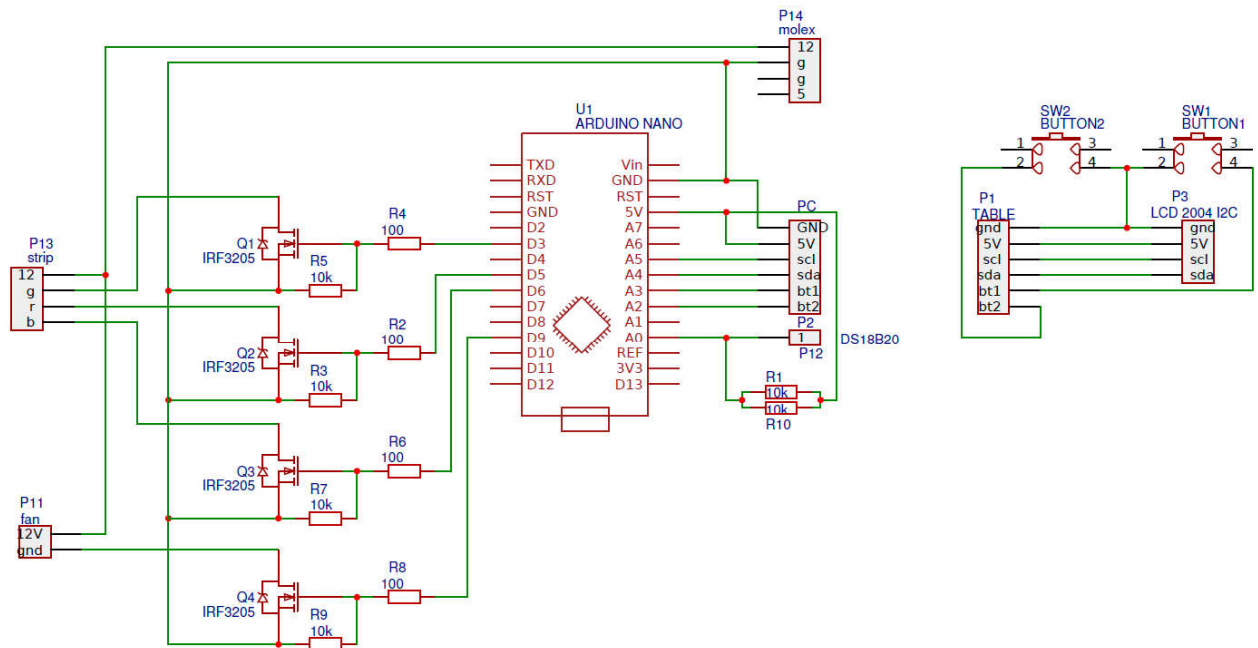


Рис. 3.2 – Схема підключення компонентів у середовищі EasyEDA

Розподіл пінів мікроконтролера (Pinout) для підключення периферичних пристроїв наведено у таблиці 3.1.

Карта підключення портів Arduino Nano

Компонент	Вивід компонента	Порт Arduino Nano	Тип сигналу
LCD 1602 (I2C)	SDA	A4	Шина даних I2C
	SCL	A5	Шина тактування I2C
Датчик DS18B20	DQ (Data)	D2	Цифровий (1-Wire)
Датчик CSLA1CD	OUT	A0	Аналоговий
Кнопка "Menu"	Pin 1	D4	Цифровий вхід
Кнопка "Select"	Pin 1	D5	Цифровий вхід
RGB Стрічка	R (Red)	D9	PWM вихід
	G (Green)	D10	PWM вихід
	B (Blue)	D11	PWM вихід

3.2.2 Тестування та налагодження апаратної частини прототипу

Тестування апаратної частини є невід'ємним етапом розробки, метою якого є перевірка надійності електричних з'єднань та коректності роботи окремих вузлів перед написанням основного програмного коду. Процес налагодження проводився за модульним принципом:

1. Перевірка підсистеми живлення: Перед встановленням мікроконтролера було перевірено напруги на лініях +5В та +12В, щоб уникнути електричного пошкодження компонентів.
2. Сенсорна група:
 - Перевірено стабільність зчитування даних з датчика температури DS18B20. Підключення виконано з використанням підтягуючого резистора 4.7 кОм. Перевірено коректність визначення унікальної адреси пристрою на шині 1-Wire.
 - Протестовано роботу датчика струму CSLA1CD. За допомогою мультиметра проведено калібрування значень на аналоговому вході Arduino відповідно до реального струму навантаження.

3. Виконавчі механізми:

- Перевірено роботу силового блоку керування вентилятором. Ключовим елементом є MOSFET-транзистор, який керується ШІМ-сигналом (PWM) від мікроконтролера. Тестування підтвердило плавність регулювання обертів.
 - Перевірено роботу RGB-стрічки, підключеної через транзисторні ключі.
4. Інтерфейс: Перевірено реакцію на натискання тактових кнопок (усунення ефекту «брязкоту контактів») та виведення тестових символів на LCD-дисплей через шину I2C.

Після інтеграції всіх модулів було проведено навантажувальне тестування прототипу. Система працювала у безперервному режимі моніторингу протягом 24 годин. За цей період збоїв у роботі мікроконтролера, перегріву силових ключів або зависання шини I2C зафіксовано не було. Апаратна частина визнана придатною до експлуатації.

3.3 Програмна реалізація системи

Для написання мікропрограми (прошивки) використано інтегроване середовище розробки Arduino IDE (рис. 3.3). Це кросплатформний додаток, написаний на Java, що включає редактор коду, компілятор (avr-g++) та модуль завантаження прошивки в мікроконтролер.

Мова програмування Arduino базується на C/C++ і адаптована для швидкого прототипування. Структурно код програми (скетч) поділяється на такі логічні блоки:

1. Підключення бібліотек (Libraries): Ініціалізація готових модулів коду для роботи з конкретним обладнанням (наприклад, `LiquidCrystal_I2C.h` для дисплея, `OneWire.h` та `DallasTemperature.h` для термодатчиків).
2. Оголошення змінних та констант (Data): Визначення глобальних змінних для зберігання значень температур, таймерів та статусів системи, а також призначення пінів (Pin Definitions).

3. Функція `void setup()`: Виконується одноразово при подачі живлення. Тут відбувається налаштування режимів роботи портів (INPUT/OUTPUT), ініціалізація шин даних (Serial, I2C, 1-Wire) та запуск дисплея.
4. Функція `void loop()`: Нескінченний цикл, у якому виконується основний алгоритм: опитування датчиків, обробка даних, оновлення інформації на екрані та керування вентилятором.

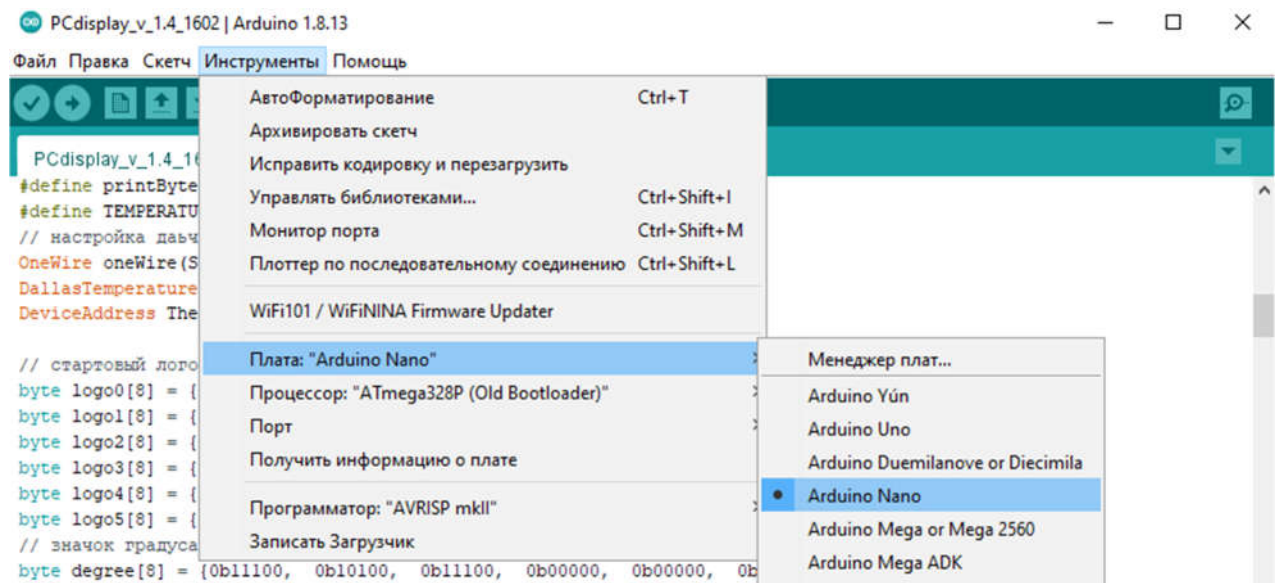


Рис. 3.3 – Інтерфейс середовища розробки Arduino IDE

Для забезпечення сумісності з використовуваними бібліотеками рекомендовано використання версії Arduino IDE не нижче 1.8.13. Перед завантаженням скетчу в меню налаштувань необхідно обрати цільову плату (Tools > Board > Arduino Nano) та відповідний COM-порт, до якого підключено пристрій.

3.3.1 Програмна реалізація алгоритму вимірювання струму

Оскільки обраний датчик струму CSLA1CD є пристроєм з лінійним аналоговим виходом, для роботи з ним не потрібно встановлювати специфічні сторонні бібліотеки. Обробка сигналу здійснюється засобами вбудованого в мікроконтролер Аналого-Цифрового Перетворювача (АЦП).

Алгоритм обробки даних:

1. Ініціалізація: У функції `void setup()` пін, до якого підключено датчик (згідно зі схемою – A0), не потребує спеціальної конфігурації, оскільки аналогові входи налаштовані на вхід за замовчуванням.
2. Зчитування: У головному циклі `void loop()` використовується стандартна функція `analogRead(pin)`, яка повертає ціле число у діапазоні від 0 до 1023 (для 10-бітного АЦП).
3. Конвертація: Отримане значення необхідно перетворити у фізичну величину (Амperi) за формулою:

$$I = \frac{(V_{out} - V_{ref}) \times K_{div}}{Sens} \quad (3.1)$$

де: V_{out} – напруга, розрахована з показань АЦП; V_{ref} – напруга зміщення (нульова точка), зазвичай дорівнює половині напруги живлення (2.5 В); $Sens$ – чутливість датчика (для CSLA1CD ≈ 49.6 мВ/А); K_{div} – коефіцієнт резистивного діляника (якщо використовується для узгодження рівнів 12В/5В).

```
int analogPin = A0; // с аналогового выхода датчика Холла
int digitalPin = 7; // с цифрового выхода датчика Холла

void setup() {
    pinMode(digitalPin, INPUT);
    Serial.begin(9600);
}

void loop() {
    int analogValue = analogRead(analogPin); // считываем аналоговое значение
    int digitalValue = digitalRead(digitalPin); // считываем цифровое значение
    Serial.println((String)digitalValue + "\t" + (String)analogValue);
    delay(100);
}
```

Рис. 3.4 – Фрагмент коду для обробки сигналу аналогового датчика Холла

Для підвищення точності вимірювань у програмному коді рекомендується реалізувати метод «ковзного середнього» (Moving Average), роблячи серію з 10–50 вимірювань та обчислюючи їх середнє арифметичне. Це дозволить нівелювати шум АЦП.

3.3.2 Програмування інтерфейсу користувача (LCD I2C)

Для організації взаємодії мікроконтролера з LCD-дисплеєм формату 1602 через шину I2C необхідно використовувати спеціалізовані бібліотеки, які інкапсують низькорівневий протокол обміну даними.

У середовищі Arduino IDE використовуються:

1. Бібліотека <Wire.h>: Входить до стандартного пакету Arduino IDE. Відповідає за апаратну реалізацію протоколу I2C на пінах SDA/SCL.
2. Бібліотека <LiquidCrystal_I2C.h>: Стороння бібліотека, яка забезпечує керування контролером дисплея HD44780 через розширювач портів PCF8574. Вона дозволяє ініціалізувати дисплей, керувати підсвічуванням та виводити текст простими командами.

Перед початком роботи необхідно створити об'єкт дисплея, вказавши його адресу на шині (зазвичай 0x27 або 0x3F) та розмірність (16 стовпців, 2 рядки).

Основні методи, що використовуються в коді:

- lcd.init() – ініціалізація дисплея;
- lcd.backlight() – увімкнення підсвічування;
- lcd.setCursor(col, row) – встановлення курсору в задану позицію;
- lcd.print("String") — виведення текстової інформації.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // Подключение библиотеки
LiquidCrystal_I2C lcd(0x27,16,2); // Указываем I2C адрес (наиболее распространенный)

void setup()
{
    lcd.init();                // Инициализация дисплея
    lcd.backlight();           // Подключение подсветки
    lcd.setCursor(0,0);        // Установка курсора в начало первой строки
    lcd.print("Hello");        // Набор текста на первой строке
    lcd.setCursor(0,1);        // Установка курсора в начало второй строки
    lcd.print("REFLEX-TH3");    // Набор текста на второй строке
}

void loop()
{
}
```

Рис. 3.5 – Приклад використання бібліотеки LiquidCrystal_I2C

У деяких випадках, якщо дисплей не відображає символи, проблема може полягати у несумісності версії чіпа розширювача портів. Як альтернативу рекомендується використовувати бібліотеку `LiquidCrystal_PCF8574`, яка має розширені алгоритми автовизначення адреси пристрою.

3.3.3 Програмна реалізація алгоритму термометрії

Для організації обміну даними з цифровим термометром DS18B20 використовується протокол 1-Wire. Хоча фізичний рівень протоколу досить складний (вимагає точних часових затримок порядку мікросекунд), у середовищі Arduino IDE для роботи з цими датчиками використовуються дві стандартні бібліотеки:

1. `<OneWire.h>` – реалізує низькорівневий протокол обміну бітами та байтами по одній лінії.
2. `<DallasTemperature.h>` – високорівнева надбудова, яка містить зручні методи для ініціалізації датчиків, відправки команд на вимірювання та перетворення отриманих даних у градуси Цельсія.

Однією з головних переваг DS18B20 є наявність унікального 64-бітного ROM-коду, який прошивається заводом-виробником. Цей код складається з:

- Family Code (8 біт): Для DS18B20 це завжди 0x28. Це дозволяє програмі автоматично ідентифікувати тип підключеного датчика.
- Serial Number (48 біт): Унікальний номер пристрою.
- CRC (8 біт): Контрольна сума для перевірки цілісності даних.

Існує два підходи до роботи з адресами у програмному коді:

1. Динамічний пошук: Використання методу `sensors.getTempCByIndex(i)`, де `i` – порядковий номер датчика на шині. Це спрощує код, але має недолік: якщо один датчик вийде з ладу, нумерація може зміститися, і дані "переплутаються".
2. Статична адресація (Hardcoding): Попереднє зчитування адрес усіх датчиків (за допомогою сканера I2C/1-Wire) і прописування їх у скетчі як констант (`DeviceAddress`). Це найбільш надійний метод для системи

моніторингу, оскільки він гарантує, що температура CPU завжди буде прив'язана до датчика, встановленого на CPU, незалежно від стану інших сенсорів.

У головному циклі програми `void loop()` реалізовано наступну послідовність дій:

1. Запит на вимірювання: Команда `sensors.requestTemperatures()` надсилається всім пристроям на лінії.
2. Конвертація: Датчики виконують вимірювання та записують результат у власну оперативну пам'ять (Scratchpad). Цей процес займає час (до 750 мс для 12-бітної точності). Бібліотека може очікувати завершення автоматично.
3. Зчитування: Команда `sensors.getTempC(sensorAddress)` зчитує готові дані з пам'яті конкретного датчика та повертає значення типу `float`.

```
#include <OneWire.h>
OneWire ds(10);
void setup(void)
{
    Serial.begin(9600);
}
void loop(void)
{
    byte i;
    byte present = 0;
    byte data[12];
    byte addr[8];
    if ( !ds.search(addr)) {
        Serial.print("No more addresses.\n");
        ds.reset_search();
        return;
    }
    Serial.print("R=");
    for( i = 0; i < 8; i++) {
        Serial.print(addr[i], HEX);
        Serial.print(" ");
    }
```

Рис. 3.6 – Приклад програмного коду для роботи з DS18B20

Протокол 1-Wire дозволяє жити датчик безпосередньо від лінії даних, накопичуючи енергію у внутрішньому конденсаторі, поки на лінії логічна "1". Однак у даному проекті, зважаючи на наявність потужних джерел електромагнітних завад (вентилятори, блоки живлення ПК), реалізовано схему з

активним зовнішнім живленням (+5В). Це забезпечує стабільність передачі даних та виключає "зависання" датчиків при частих опитуваннях.

3.4 Візуалізація монтажно́ї схеми у середовищі Fritzing

Для створення наочно́ї монтажно́ї схеми (wiring diagram), яка демонструє фізичне розташування компонентів та з'єднань на макетній платі, було обрано програмне середовище Fritzing.

Fritzing – це система автоматизованого проектування (CAD) початкового рівня з відкритим вихідним кодом. Цільовою аудиторією програмного пакету є спільнота мейкерів (makers), радіоаматорів та розробників вбудованих систем на базі платформи Arduino.

Основні можливості середовища, використані в проекті:

1. Режим «Макетна плата» (Breadboard View): Дозволяє віртуально відтворити процес прототипування, з'єднуючи компоненти провідниками так, як це виглядає у реальному пристрої.
2. Бібліотека компонентів: Fritzing містить широку базу стандартних модулів (Arduino Nano, LCD-дисплеї, датчики, резистори), а також дозволяє імпортувати користувацькі елементи.
3. Експорт документації: Можливість генерації векторних зображень (PDF, SVG) для оформлення графічної частини кваліфікаційної роботи.

Робота над візуалізацією проекту складалася з наступних етапів:

1. З бібліотеки компонентів (права панель інтерфейсу) на робоче поле було додано контролер Arduino Nano v3.
2. Додано периферійні пристрої: модуль LCD 1602 з I2C-адаптером, датчик температури DS18B20 та датчик струму.
3. Виконано трасування з'єднань (Routing) з використанням колірного кодування провідників (червоний – +5V, чорний – GND, жовтий – сигнальні лінії), що відповідає схемі, розробленій у п. 3.2.1.

Головною перевагою Fritzing є висока наочність графічного представлення, що робить його ідеальним інструментом для створення ілюстрацій до технічної документації та інструкцій зі збирання.

Однак слід зазначити обмеження даного ПЗ для професійного застосування:

- Відсутність симуляції роботи схеми (на відміну від Proteus або Tinkercad).
- Спрощений функціонал редактора друкованих плат (PCB), який поступається професійним пакетам типу EasyEDA або Altium Designer.

Саме тому в даній роботі Fritzing використано виключно для створення монтажних ескізів, наведених у Додатку Б, тоді як принципова електрична схема розроблялася у спеціалізованому середовищі EasyEDA.

3.5 Опис інтерфейсу користувача та функціональних можливостей програмного комплексу

Інтерфейс програми:



Рис. 3.7 – Головна сторінка



Рис. 3.8 – Друга сторінка



Рис. 3.9 – Третя сторінка



Рис. 3.10 – Четверта сторінка



Рис. 3.11 – П'ята сторінка

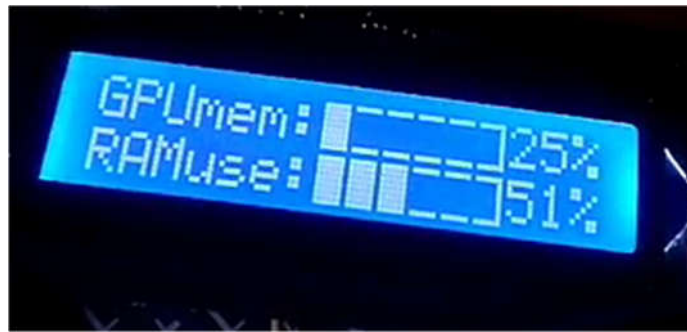


Рис. 3.12 – Шоста сторінка

Прототип та його можливості(Рис.3.13):

- Вивод основних параметрів заліза на зовнішній LCD дисплей
- Температура: CPU, GPU, материнська плата, найгарячіший HDD
- Рівень завантаження: CPU, GPU, RAM, відеопам'ять
- Графіки зміни перерахованих вище параметрів за часом
- Температура з зовнішніх датчиків (DS18B20)
- Поточний рівень швидкості зовнішніх вентиляторів
- Управління великою кількістю 12 вольтів 2, 3, 4 провідних вентиляторів
- Автоматичне управління швидкістю пропорційно температурі
- Ручне управління швидкістю з інтерфейсу програми
- Управління RGB світлодіодною стрічкою
- Управління кольором пропорційно температурі
- Ручне управління кольором з інтерфейсу програми
- Управління яскравістю.



Рис. 3.13 – Прототип

Висновки за розділом 3

У третьому розділі кваліфікаційної роботи виконано повний цикл розробки апаратно-програмного комплексу системи моніторингу ПК. За результатами проведеної роботи можна зробити наступні висновки:

1. Для організації процесу створення програмного забезпечення обрано каскадну модель (Waterfall). Це дозволило чітко структурувати етапи проектування, кодування та тестування, забезпечивши послідовне виконання технічного завдання.
2. За допомогою САПР EasyEDA розроблено принципову електричну схему пристрою, а у середовищі Fritzing створено наочну монтажну схему. На основі платформи Arduino Nano зібрано фізичний прототип, до якого інтегровано цифрові термометри DS18B20, аналоговий датчик струму CSLA1CD, символічний дисплей LCD 1602 та виконавчі механізми (силові ключі керування вентиляторами, RGB-стрічка).
3. У середовищі Arduino IDE розроблено мікропрограму (скетч) мовою C++. Реалізовано алгоритми опитування датчиків по протоколах 1-Wire та I2C, обробку аналогових сигналів з використанням фільтрації шумів, а також логіку автоматичного керування системою охолодження (Smart PWM).
4. Розроблено клієнтське програмне забезпечення для ПК, яке забезпечує зручну візуалізацію параметрів системи (Dashboard), побудову графіків у реальному часі та гнучке налаштування профілів охолодження і підсвічування.
5. Проведено комплексне налагодження та навантажувальне тестування прототипу протягом 24 годин. Результати випробувань підтвердили стабільність роботи апаратної частини, відсутність перегріву силових елементів та коректність передачі даних телеметрії.

Таким чином, розроблений прототип повністю відповідає функціональним вимогам проекту: він забезпечує автономний моніторинг критичних параметрів ПК та активне керування системою охолодження, підвищуючи надійність роботи комп'ютерної техніки.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Метою даного розділу є аналіз умов праці розробника під час створення апаратно-програмного комплексу моніторингу ПК. Робоче місце розробника характеризується поєднанням розумової праці (програмування, проектування схем) та монтажних робіт (паяння, складання прототипу, тестування електричних кіл).

4.1 Аналіз шкідливих та небезпечних виробничих факторів

Згідно з ДСТУ 12.0.003-74, на розробника під час виконання дипломного проекту можуть впливати наступні групи небезпечних та шкідливих факторів:

1. Фізичні фактори:

- Підвищений рівень електромагнітного випромінювання: джерелом є системний блок ПК, монітор та периферійне обладнання.
- Небезпека ураження електричним струмом: робота ведеться з мережею 220В (живлення ПК та паяльника) та лініями 12В (блок живлення ПК). Замикання в колі може призвести до електротравм.
- Термічні опіки: можливі при роботі з паяльним обладнанням (температура жала паяльника досягає 250–350°C).
- Недостатнє освітлення робочої зони: може призвести до швидкої втоми очей, особливо при монтажі дрібних SMD-компонентів.
- Шум: від системи охолодження системного блоку (вентиляторів).

2. Хімічні фактори:

- Загазованість повітря робочої зони продуктами паяння (пари свинцю, каніфолі, флюсів), що можуть викликати алергічні реакції або інтоксикацію при тривалій роботі без вентиляції.

3. Психофізіологічні фактори:

- Нервово-емоційне напруження при відлагодженні коду та схеми.
- Перенапруження зорового аналізатора.
- Монотонність праці та гіподинамія (тривале сидіння у фіксованій позі).

4.2 Вимоги до мікроклімату робочої зони

Метеорологічні умови на робочому місці (температура, відносна вологість, швидкість руху повітря) суттєво впливають на самопочуття та працездатність. Оскільки робота відноситься до категорії легкої фізичної праці (категорія Ia), згідно з ДСН 3.3.6.042-99 встановлюються наступні норми:

- Температура повітря: 22–24 °C (у холодний період року) та 23–25 °C (у теплий період).
- Відносна вологість повітря: 40–60%.
- Швидкість руху повітря: не більше 0.1 м/с.

Для забезпечення цих параметрів приміщення повинно бути обладнане системою опалення та природною або штучною вентиляцією. Під час паяльних робіт обов'язковим є провітрювання або використання витяжки для видалення шкідливих випарів.

4.3 Виробниче освітлення

Рациональне освітлення є запорукою збереження зору при роботі з дрібними радіoeлектронними компонентами (Arduino, резистори, дисплей). Використовується комбіноване освітлення: загальне (стельові світильники) та місцеве (настільна лампа).

Згідно з ДБН В.2.5-28:2018 для робіт високої точності (розряд зорової роботи III) нормуються такі показники:

- Освітленість робочої поверхні столу: не менше 300–500 лк.
- Відсутність відблисків на екрані монітора (розміщення робочого місця боком до вікна).

- Коефіцієнт пульсації освітлення не повинен перевищувати 5% (використання LED-ламп без мерехтіння).

4.4 Електробезпека

Оскільки розробка пристрою передбачає безпосередній контакт з струмопровідними частинами та вимірювальними приладами, електробезпека є пріоритетним напрямком. Приміщення лабораторії/офісу відноситься до приміщень без підвищеної небезпеки (сухе, зі струмонепровідною підлогою).

Основні заходи захисту (згідно з ДСТУ 7237:2011):

1. Заземлення: Корпуси ПК, паяльної станції та вимірювальних приладів (осцилограф, лабораторний БЖ) повинні бути надійно заземлені.
2. Ізоляція: Перед початком роботи необхідно перевіряти цілісність ізоляції дротів живлення та щупів мультиметра.
3. Режим роботи: Забороняється проводити перекомутацію схеми під напругою. Підключення датчиків до Arduino виконується лише при відключеному живленні USB.
4. Монтаж: При роботі з напругою 12В та 220В слід використовувати інструмент з ізольованими ручками.

4.5 Пожежна безпека

Основними причинами пожежі в лабораторії радіоелектроніки можуть бути:

- Коротке замикання в електромережі або в експериментальному пристрої.
- Перегрів паяльного обладнання ("забутий" увімкнений паяльник).
- Загоряння легкозаймистих матеріалів (папір, спирт для промивки плат).

Відповідно до «Кодексу цивільного захисту України», приміщення повинно бути оснащене первинними засобами пожежогасіння. Для гасіння електроустановок під напругою (ПК, прилади) використовуються вуглекислотні (ОУ-2, ОУ-5) або порошкові вогнегасники. Використання води або пінних вогнегасників категорично заборонено через ризик ураження електричним струмом та пошкодження обладнання.

Висновки за розділом 4

У розділі «Охорона праці» проведено аналіз умов праці при розробці системи моніторингу ПК. Встановлено, що основними небезпечними факторами є ураження електричним струмом, термічні опіки при паянні та зорове напруження.

Для забезпечення безпечних умов праці визначено необхідні параметри мікроклімату та освітлення, розроблено заходи з електробезпеки (заземлення, перевірка ізоляції) та пожежної безпеки (наявність вуглекислотних вогнегасників). Дотримання цих вимог гарантує збереження здоров'я розробника та мінімізує ризики виникнення аварійних ситуацій під час роботи над дипломним проектом.

ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне науково-технічне завдання – розробку автономної системи апаратного моніторингу стану персонального комп'ютера на базі мікроконтролерної платформи Arduino. Це дозволило створити ефективний інструмент для підвищення надійності обчислювальної техніки шляхом незалежного контролю критичних параметрів.

Основні наукові та практичні результати роботи полягають у наступному:

1. Проведено комплексний аналіз предметної області. Встановлено, що штатні засоби термоконтролю ПК (внутрішні датчики) та програмні методи моніторингу (AIDA64, HWMonitor) мають суттєвий недолік – залежність від стабільності операційної системи. При програмних збоях користувач втрачає контроль над станом обладнання, що підвищує ризик аварійного перегріву. Доведено доцільність створення зовнішнього апаратного модуля, який працює автономно.

2. Обґрунтовано вибір елементної бази та розроблено апаратну архітектуру. В якості обчислювального ядра обрано платформу Arduino Nano V3.0 (ATmega328P), що забезпечує оптимальне співвідношення ціни та функціональності. Спроектовано сенсорну мережу, яка включає цифрові датчики температури DS18B20 (точність $\pm 0.5^{\circ}\text{C}$) та безконтактний датчик струму CSLA1CD з гальванічною розв'язкою. Для візуалізації даних застосовано LCD-дисплей 1602 з інтерфейсом I2C, що дозволило оптимізувати використання портів мікроконтролера.

3. Розроблено схемотехнічні рішення. Створено принципову електричну схему пристрою в середовищі EasyEDA та монтажну схему у Fritzing. Особливу увагу приділено надійності: розроблено вузол оптичної розв'язки для інтерфейсу 1-Wire, що захищає порти мікроконтролера від електромагнітних завад та стрибків напруги.

4. Реалізовано програмне забезпечення:

- Створено мікропрограму (скетч) для Arduino мовою C++, яка реалізує алгоритми опитування датчиків, фільтрації шумів (метод ковзного середнього) та автоматичного керування системою охолодження (Smart PWM).
- Розроблено клієнтське ПЗ для ПК, що виконує функції панелі моніторингу (Dashboard) та дозволяє користувачеві налаштовувати профілі роботи вентиляторів і RGB-підсвічування.

5. Виготовлено та протестовано фізичний прототип. Зібраний пристрій пройшов успішні навантажувальні випробування протягом 24 годин. Система продемонструвала стабільну роботу, коректне зчитування телеметрії та ефективне керування обертами вентиляторів залежно від температури компонентів.

6. Проаналізовано питання охорони праці. Визначено основні шкідливі фактори при розробці та експлуатації електронного обладнання, розроблено заходи щодо електробезпеки та організації робочого місця.

Практична цінність роботи полягає у створенні бюджетного, відкритого (Open Source) рішення, яке за своїм функціоналом не поступається дорогим комерційним аналогам. Розроблений пристрій може бути використаний як для модернізації («модингу») домашніх ПК, так і для адміністрування парку техніки в навчальних закладах або офісах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Белов В. М. Мікропроцесорна техніка : підручник / В. М. Белов, В. Г. Сліпченко. – К. : Каравела, 2018. – 346 с.
2. Периков Ю. М. Основи програмування мікроконтролерів AVR : навчальний посібник / Ю. М. Периков. – Одеса : ОНАЗ ім. О. С. Попова, 2011. – 136 с.
3. Шілдт Г. С++: Базовий курс / Г. Шілдт. – 3-тє вид. – М. : Вільямс, 2010. – 624 с.
4. Arduino Nano (V3.0) User Manual [Електронний ресурс]. – Режим доступу: <https://store.arduino.cc/arduino-nano>.
5. ATmega328P Datasheet. 8-bit AVR Microcontrollers [Електронний ресурс] / Microchip Technology Inc. – 2020. – Режим доступу: <http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega328P-Datasheet.pdf>.
6. DS18B20 Datasheet. Programmable Resolution 1-Wire Digital Thermometer [Електронний ресурс] / Maxim Integrated. – 2019. – Режим доступу: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
7. CSLA1CD Datasheet. Linear Current Sensors [Електронний ресурс] / Honeywell International Inc. – Режим доступу: <https://sensing.honeywell.com/honeywell-sensing-current-sensors-line-guide-005894-2-en.pdf>.
8. PCF8574 Datasheet. Remote 8-Bit I/O Expander for I2C Bus [Електронний ресурс] / Texas Instruments. – 2015. – Режим доступу: <https://www.ti.com/lit/ds/symlink/pcf8574.pdf>.
9. HD44780U (LCD-II). Dot Matrix Liquid Crystal Display Controller/Driver [Електронний ресурс] / Hitachi, Ltd. – 1999. – Режим доступу: <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>.
10. Офіційна документація середовища Arduino IDE [Електронний ресурс]. – Режим доступу: <https://www.arduino.cc/en/Reference/HomePage>.

11. Бібліотека DallasTemperature. Опис та приклади використання [Електронний ресурс]. – Режим доступу: <https://github.com/milesburton/Arduino-Temperature-Control-Library>.
12. EasyEDA Tutorial. Web-based EDA tool suite [Електронний ресурс]. – Режим доступу: <https://easyeda.com/Doc/Tutorial/>.
13. Кодекс законів про працю України : Закон України від 10.12.71 р. № 322-VIII // Відомості Верховної Ради УРСР. – 1971. – Додаток до № 50. – Ст. 375.
14. ДСТУ 7237:2011. Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту. – К. : Держспоживстандарт України, 2012. – 18 с.
15. ДБН В.2.5-28:2018. Природне і штучне освітлення. – К. : Мінрегіон України, 2018. – 133 с.
16. Правила улаштування електроустановок (ПУЕ). – Харків : Форт, 2017. – 760 с
17. Багатоекранний метод та пристрій відображення персонального комп'ютера:<https://patents.google.com/patent/US6522309B1/en>
18. Система і спосіб для контролю та управління віддаленими пристроями<https://patents.google.com/patent/US7468661B2/en>
19. Метод моніторингу виконуваного обладнання комп'ютера:<https://patents.google.com/patent/CN1808395A/en>
20. Промислова система управління інтелектуальним обладнанням моніторингу: <https://patents.google.com/patent/US20150346706>

КОД ПРОГРАМИ

```

#define DRIVER_VERSION 1
#define COLOR_ALGORITHM 0
#define ERROR_DUTY 90
#define ERROR_BACKLIGHT 0
#define ERROR_UPTIME 0

#define FAN_PIN 9
#define R_PIN 5
#define G_PIN 3
#define B_PIN 6
#define BTN1 A3
#define BTN2 A2
#define SENSOR_PIN 14

#include <OneWire.h>
#include <DallasTemperature.h>
#include <string.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <TimerOne.h>

#define printByte(args) write(args);
#define TEMPERATURE_PRECISION 9
OneWire oneWire(SENSOR_PIN);
DallasTemperature sensors(&oneWire);
DeviceAddress Thermometer1, Thermometer2;

byte logo0[8] = {0b00011, 0b00110, 0b01110, 0b11111, 0b11011, 0b11001, 0b00000, 0b00000};
byte logo1[8] = {0b10000, 0b00001, 0b00001, 0b00001, 0b00000, 0b10001, 0b11011, 0b11111};
byte logo2[8] = {0b11100, 0b11000, 0b10001, 0b11011, 0b11111, 0b11100, 0b00000, 0b00000};
byte logo3[8] = {0b00000, 0b00001, 0b00011, 0b00111, 0b01101, 0b00111, 0b00010, 0b00000};
byte logo4[8] = {0b11111, 0b11111, 0b11011, 0b10001, 0b00000, 0b00000, 0b00000, 0b00000};
byte logo5[8] = {0b00000, 0b10000, 0b11000, 0b11100, 0b11110, 0b11100, 0b01000, 0b00000};

byte degree[8] = {0b11100, 0b10100, 0b11100, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000};

byte right_empty[8] = {0b11111, 0b00001, 0b00001, 0b00001, 0b00001, 0b00001, 0b00001, 0b11111};
byte left_empty[8] = {0b11111, 0b10000, 0b10000, 0b10000, 0b10000, 0b10000, 0b10000, 0b11111};
byte center_empty[8] = {0b11111, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111};

byte row8[8] = {0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111};
byte row7[8] = {0b00000, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111};
byte row6[8] = {0b00000, 0b00000, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111};
byte row5[8] = {0b00000, 0b00000, 0b00000, 0b11111, 0b11111, 0b11111, 0b11111, 0b11111};
byte row4[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b11111, 0b11111, 0b11111, 0b11111};
byte row3[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111, 0b11111, 0b11111};
byte row2[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111, 0b11111};
byte row1[8] = {0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b00000, 0b11111};

char inData[82];
int PCdata[20];
byte PLOTmem[6][16];
byte blocks, halves;
byte index = 0;

```

```

int display_mode = 6;
String string_convert;
unsigned long timeout, uptime_timer, plot_timer;
boolean lightState, reDraw_flag = 1, updateDisplay_flag, updateTemp_flag, timeOut_flag = 1;
int duty, LEDcolor;
int k, b, R, G, B, Rf, Gf, Bf;
byte mainTemp;
byte lines[] = {4, 5, 7, 6};
byte plotLines[] = {0, 1, 4, 5, 6, 7};
String perc;
unsigned long sec, mins, hrs;
byte temp1, temp2;
boolean btn1_sig, btn2_sig, btn1_flag, btn2_flag;

// Названия для легенды графиков
const char plot_0[] = "CPU";
const char plot_1[] = "GPU";
const char plot_2[] = "RAM";

const char plot_3[] = "temp";
const char plot_4[] = "load";
const char plot_5[] = "mem";

static const char *plotNames0[] = {
    plot_0, plot_1, plot_0, plot_1, plot_2, plot_1
};
static const char *plotNames1[] = {
    plot_3, plot_3, plot_4, plot_4, plot_4, plot_5
};

void setup() {
    Serial.begin(9600);
    Timer1.initialize(40);
    pinMode(R_PIN, OUTPUT);
    pinMode(G_PIN, OUTPUT);
    pinMode(B_PIN, OUTPUT);
    digitalWrite(R_PIN, 0);
    digitalWrite(G_PIN, 0);
    digitalWrite(B_PIN, 0);
    pinMode(BTN1, INPUT_PULLUP);
    pinMode(BTN2, INPUT_PULLUP);
    sensors.begin();
    sensors.getAddress(Thermometer1, 0);
    sensors.getAddress(Thermometer2, 1);
    sensors.setResolution(Thermometer1, TEMPERATURE_PRECISION);
    sensors.setResolution(Thermometer2, TEMPERATURE_PRECISION);
    // инициализация дисплея
    lcd.init();
    lcd.backlight();
    lcd.clear();
    show_logo();
    delay(2000);
    lcd.clear();
    show_logo2();
}

```

```

Timer1.pwm(FAN_PIN, 400);
delay(2000);
lcd.clear();
PCdata[8] = speedMAX;
PCdata[9] = speedMIN;
PCdata[10] = tempMAX;
PCdata[11] = tempMIN;
}

void loop() {
  parsing();
  updatePlot();
  getTemperature();
  dutyCalculate();
  Timer1.pwm(FAN_PIN, duty * 10);
  LEDcontrol();
  buttonsTick();
  updateDisplay();
  timeoutTick();
}

void buttonsTick() {
  btn1_sig = !digitalRead(BTN1);
  btn2_sig = !digitalRead(BTN2);
  if (btn1_sig && !btn1_flag) {
    display_mode++;
    reDraw_flag = 1;
    if (display_mode > 9) display_mode = 0;
    btn1_flag = 1;
  }
  if (!btn1_sig && btn1_flag) {
    btn1_flag = 0;
  }
  if (btn2_sig && !btn2_flag) {
    display_mode--;
    reDraw_flag = 1;
    if (display_mode < 0) display_mode = 9;
    btn2_flag = 1;
  }
  if (!btn2_sig && btn2_flag) {
    btn2_flag = 0;
  }
}

void getTemperature() {
  if (updateTemp_flag) {
    sensors.requestTemperatures();
    temp1 = sensors.getTempC(Thermometer1);
    temp2 = sensors.getTempC(Thermometer2);
    updateTemp_flag = 0;
  }
}

void LEDcontrol() {
  b = PCdata[16];
  if (PCdata[13] == 1)
    LEDcolor = PCdata[15];
  else {
    LEDcolor = min(mainTemp, PCdata[11], PCdata[10], 0, 1000);
  }
}

```

СХЕМА ПІДКЛЮЧЕННЯ

