

**КВАЛІФІКАЦІЙНА РОБОТА**

Енергоефективне планування задач на одноплатному комп'ютері

Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КВРКІ.2301101.23.01.24 ПЗ

Виконав здобувач III курсу, група КІ2с-23-1

Керівник

Науковий ступінь, учене звання

Нормоконтролер

Науковий ступінь, учене звання

До захисту допускаю:  
завідувач кафедри КІС  
«01» червня 2026 р.

дата

Точ  
Підпис

Підпис

Яцків  
Підпис

Підпис

Підпис

Валентин ГОМЕНЮК

Ініціали, прізвище

Василь ЯЦКІВ

Ініціали, прізвище

Сергій ЛИСЕНКО

Ініціали, прізвище

Ольга ПАВЛОВА

Ініціали, прізвище

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІС



Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Гоменюку Валентину Сергійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Енергоефективне планування задач на одноплатному комп'ютері  
Керівник проекту (роботи) Яцків Василь Васильович, д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Теоретичні аспекти енергоефективного виконання обчислювальних задач на одноплатних комп'ютерах

Розроблення програмно-апаратного засобу енергоефективного планування задач на одноплатному комп'ютері

Практична реалізація та перевірка роботи програмно-апаратного засобу енергоефективного планування задач

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

Схема електрична функціональна

Схема електрична принципова

Блок-схема алгоритму програмного забезпечення

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 10 » 01 2026 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2026	виконано
4	Робота над розділом 2 – вибір компонентів для енергоефективного планування задач на одноплатному комп'ютері	01.04.2026	виконано
5	Робота над розділом 3 – проектування програмно-апаратного засобу енергоефективного планування задач	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	виконано
7	Попередній захист ВКР	26.05.2026	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач Тош Валентин ГОМЕНЮК  
Підпис Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи Яцків Василь ЯЦКІВ  
Підпис Імя, ПРІЗВИЩЕ

№ р я д к а	Ф о р м а т	Позначення	Найменування	К і л · л и с т і в	№ ек з	П р и м і т к а
			Текстові документи			
1		КвРКІ.301101.23.01.24 ПЗ	Пояснювальна записка	71		
2			<u>Графічні матеріали</u>			
3		КвРКІ.301101.23.01.24 Е8	Схема електрична функціональна	1		
4		КвРКІ. 301101.23.01.24 Е3	Схема електрична принципова	1		
5		КвРКІ. 301101.23.01.24 Е8	Блок-схема алгоритму програмного забезпечення	1		

					КвРКІ.2301101.23.01.24 ВП				
Зм	Арк	№ докум	Підпис	Дата	Енергоефективне планування задач на одноплатному комп'ютері		Літера	Аркуш	Аркушів
Розробив		Гоменюк	<i>Гоу</i>	01.06			У	1	1
Перевір.		Яцків	<i>Яцків</i>	01.06	Відомість проекту		ХНУ, КІ2с-23-1		
Н. контр.		Лисенко	<i>Лисенко</i>	01.06					
Затв.		Павлова	<i>Павлова</i>	01.06					

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Енергоефективне планування задач на одноплатному комп'ютері».

Автор роботи: Валентин ГОМЕНЮК.

Керівник роботи: Василь ЯЦКІВ.

Пояснювальна записка: 71 с., 11 рис., 3 дод., 50 джерел.

Графічна частина: 3 креслення.

АЛГОРИТМ КЕРУВАННЯ, ЕНЕРГОЕФЕКТИВНІСТЬ, ЖУРНАЛЮВАННЯ ТЕЛЕМЕТРІЇ, ОДНОПЛАТНИЙ КОМП'ЮТЕР, ПЛАНУВАННЯ ЗАДАЧ, ТЕМПЕРАТУРА ПРОЦЕСОРА, ЧАСТОТА CPU.

Кваліфікаційна робота бакалавра присвячена розробленню програмно-апаратного засобу енергоефективного планування задач на одноплатному комп'ютері. Актуальність теми зумовлена поширенням компактних обчислювальних платформ у системах автоматизації, моніторингу, периферійних обчислень, навчальних стендах і малих серверних рішеннях, де важливими залишаються стабільна продуктивність, контроль температури та зменшення енергоспоживання.

Метою роботи є проектування, програмна реалізація та перевірка працездатності програмно-апаратного засобу енергоефективного планування задач на одноплатному комп'ютері. У процесі виконання роботи проаналізовано особливості одноплатних комп'ютерів як платформи для прикладних обчислень, розглянуто вплив навантаження та частоти процесора на енергоспоживання і тепловий режим, узагальнено підходи операційних систем до керування продуктивністю, а також сформовано задачу створення засобу керування режимами CPU на основі телеметрії та зворотного зв'язку.

*Тош*

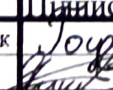
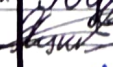
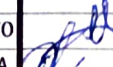

Підпис здобувача

30.05.2026

Дата

## ЗМІСТ

Вступ .....	4
1 Теоретичні аспекти енергоефективного виконання обчислювальних задач на одноплатних комп'ютерах .....	6
1.1 Особливості одноплатних комп'ютерів як платформи для вбудованих та прикладних обчислень .....	6
1.2 Вплив навантаження та частоти процесора на енергоспоживання і тепловий режим.....	10
1.3 Огляд підходів до керування продуктивністю та енергоспоживанням у сучасних операційних системах.....	12
1.4 Методи енергоефективного планування задач і керування ресурсами процесора .....	15
1.5 Постановка задачі створення програмно-апаратного засобу енергоефективного планування задач на одноплатному комп'ютері.....	18
2 Розроблення програмно-апаратного засобу енергоефективного планування задач на одноплатному комп'ютері.....	21
2.1 Загальна архітектура програмно-апаратного засобу енергоефективного планування задач.....	21
2.2 Обґрунтування вибору апаратних і програмних компонентів системи.....	26
2.3 Організація збору телеметричних даних з одноплатного комп'ютера .....	32
2.4 Розроблення логіки аналізу стану системи та визначення режиму навантаження.....	38
2.5 Розроблення алгоритму енергоефективного планування задач і керування частотою процесора .....	44
3 Практична реалізація та перевірка роботи програмно-апаратного засобу енергоефективного планування задач.....	52

<b>КВРКІ.2301101.23.01.24 ПЗ</b>								
Зм.	Арк.	№ док.ум.	Підпис	Дата	Енергоефективне планування задач на одноплатному комп'ютері	Літера	Аркуш	Аркушів
Виконав		Валентин ГОМЕНЮК		01.06		у	2	71
Перевір.		Василь ЯЦКІВ		01.06	<b>ХНУ КІ2с-23-1</b>			
Н.контр.		Сергій ЛИСЕНКО		01.06				
Затвер.		Ольга ПАВЛОВА		01.06				
					Пояснювальна записка			

3.1 Формування експериментального середовища на базі одноплатного комп'ютера .....	52
3.2 Реалізація програмного модуля збору та журналювання телеметричних даних .....	56
3.3 Реалізація алгоритму аналізу стану системи та керування частотою CPU .....	60
3.4 Перевірка роботи засобу в різних режимах навантаження .....	64
3.5 Аналіз отриманих результатів та оцінювання ефективності запропонованого підходу .....	68
Висновки .....	73
Перелік джерел посилань .....	75
Додаток А Копія креслення «Схема електрична функціональна» .....	81
Додаток Б Копія креслення «Схема електрична принципова» .....	82
Додаток В Копія креслення «Блок-схема алгоритму програмного забезпечення» .....	83

## ВСТУП

Одноплатні комп'ютери активно застосовуються в автоматизації, моніторингових комплексах, медіасервісах, периферійних обчисленнях та навчальних стендах, де поєднуються обмежені ресурси охолодження, вимоги до безперервної роботи та потреба у стабільній продуктивності. За таких умов саме процесор стає ключовим елементом, який одночасно формує швидкодію системи, рівень енергоспоживання та тепловий режим.

Сучасні операційні системи вже містять механізми керування частотою та енергетичними станами процесора, однак ці підходи орієнтовані на універсальні сценарії використання і не завжди враховують специфіку конкретних навантажень, тривалу роботу під стабільним навантаженням або обмеження за температурою, характерні для компактних платформ. У реальних умовах експлуатації це призводить до ситуацій, коли система або працює з надмірним запасом по продуктивності та підвищеним енергоспоживанням, або входить у режими обмеження частоти через перегрів, що негативно впливає на стабільність сервісів. Унаслідок цього формується практична потреба у більш гнучкому підході до керування ресурсами процесора, який спирається не лише на миттєве навантаження, а й на поточний тепловий стан і динаміку його зміни.

У межах кваліфікаційної роботи увагу зосереджено на поєднанні програмних і апаратних засобів керування з використанням телеметрії та зворотного зв'язку. Такий підхід дозволяє підтримувати роботу системи в керованому тепловому діапазоні, зменшувати енергетичні витрати та водночас зберігати прийнятний рівень продуктивності для прикладних задач. Практичний інтерес становить не лише сам факт зниження споживання енергії, а й досягнення більш рівномірного та передбачуваного режиму роботи без різких коливань частоти та вимушених спадів швидкодії.

Метою кваліфікаційної роботи є створення програмно-апаратного засобу енергоефективного планування задач на одноплатному комп'ютері, який здійснює збір телеметричних даних, аналізує стан системи та керує режимами

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

роботи процесора на основі принципу зворотного зв'язку. Для досягнення поставленої мети в роботі розглянуто архітектурні особливості одноплатних комп'ютерів, проаналізовано вплив навантаження і частоти процесора на енергоспоживання та тепловий режим, узагальнено існуючі підходи операційних систем до керування продуктивністю, а також сформовано методи енергоефективного планування задач. Окрему увагу приділено побудові та перевірці регулятора, який забезпечує більш стабільний баланс між швидкістю, температурою та витратами енергії.

Об'єктом кваліфікаційної роботи є процес керування ресурсами процесора в одноплатних обчислювальних системах. Предметом кваліфікаційної роботи є методи та засоби енергоефективного планування задач і керування частотою процесора на основі телеметрії та зворотного зв'язку.

Практична цінність отриманих результатів полягає у можливості застосування розробленого підходу в реальних системах, де важливими залишаються стабільність роботи, помірний тепловий режим і зменшене енергоспоживання. Реалізація та експериментальна перевірка запропонованого засобу дозволяє наочно продемонструвати взаємозв'язок між продуктивністю, температурою та енерговитратами й створює основу для подальшого вдосконалення систем керування ресурсами на компактних обчислювальних платформах.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

# 1 ТЕОРЕТИЧНІ АСПЕКТИ ЕНЕРГОЕФЕКТИВНОГО ВИКОНАННЯ ОБЧИСЛЮВАЛЬНИХ ЗАДАЧ НА ОДНОПЛАТНИХ КОМП'ЮТЕРАХ

## 1.1 Особливості одноплатних комп'ютерів як платформи для вбудованих та прикладних обчислень

Кваліфікаційна робота розглядає одноплатні комп'ютери як платформу, що поєднала в одному компактному виробі можливості повноцінної операційної системи й набір апаратних інтерфейсів, характерних для вбудованої техніки. На практиці цей клас пристроїв представлено сімействами Raspberry Pi (зокрема Raspberry Pi 4/5), подано на рисунку 1.1, Orange Pi, Banana Pi та близькими за ідеологією рішеннями на ARM-процесорах. Спільною рисою для них є використання системи-на-кристалі (SoC), у якій інтегровано процесорні ядра, контролери оперативної пам'яті, графічний блок, мережеві та периферійні контролери. У результаті на одній платі вже зібрано майже весь мінімально необхідний набір вузлів для запуску прикладного ПЗ без окремої материнської плати, дискретного чипсета чи додаткових плат розширення.

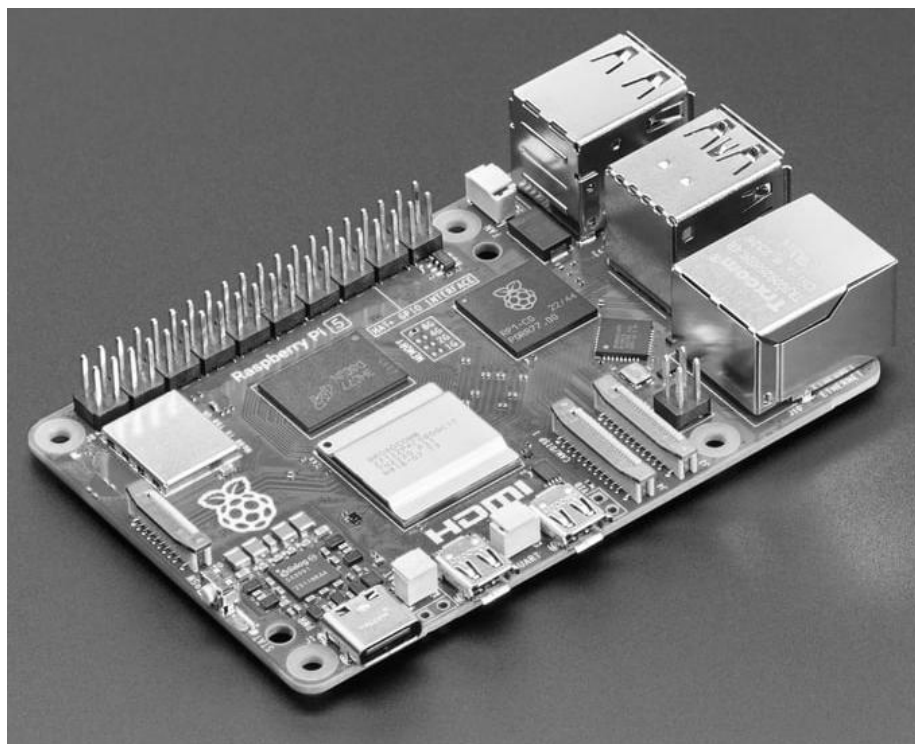


Рисунок 1.1 – Одноплатний комп'ютер Raspberry Pi [48]

						КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата			6

Одноплатний комп'ютер уже забезпечив запуск Linux-подібних систем із багатозадачністю, файловою системою, стандартним мережевим стеком і звичними інструментами адміністрування [1, 2]. Це принципово відрізняє платформу від мікроконтролерів: замість виконання одного циклічного алгоритму в прошивці реалізовано паралельну роботу сервісів і процесів, конкуренцію за CPU-час, планування потоків та керування пріоритетами [3, 4]. У межах теми енергоефективного планування задач саме ця властивість стає ключовою, оскільки регулювання частоти процесора та обмеження навантаження мають виконуватися всередині середовища ОС, не руйнуючи стабільність роботи сервісів і не погіршуючи відгук системи у важливих моментах [5, 6].

Одноплатні комп'ютери вже мають характерні апаратні обмеження живлення, які безпосередньо впливають на режим експлуатації [7, 8]. У більшості моделей живлення подається від 5 В через USB-C або подібний роз'єм, а одне джерело живлення часто одночасно забезпечує роботу периферії: USB-накопичувачів, Wi-Fi-адаптерів, камер, дисплеїв, додаткових контролерів [9, 10]. За високого навантаження процесора та інтенсивної роботи інтерфейсів пікове споживання помітно зростає [11, 12]. При неякісному адаптері або довгому кабелі вже з'являються просідання напруги, що проявляється нестабільністю, перезапусками або раптовими помилками вводу-виводу [13, 14]. Через це енергоефективні методи керування навантаженням розглядаються не лише як спосіб економії, а і як інструмент стабілізації роботи в умовах обмеженого живлення [15, 16].

Окремою технічною особливістю є підсистема зберігання даних. У багатьох конфігураціях системний диск реалізовано через microSD-карту, рідше - через eMMC або зовнішній SSD [17, 18]. microSD спрощує старт і здешевлює систему, але має нижчу стійкість до інтенсивних записів і часто демонструє нестабільну швидкодію при довгих серіях дрібних операцій [19, 20]. Тому в задачах, де постійно збирається телеметрія або ведуться журнали подій, актуальними стають буферизація даних у пам'яті, пакетний запис, контроль

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

частоти логування і відмова від зайвих синхронних операцій диска [21, 22]. Це важливо, бо надмірний I/O створює додаткове навантаження на CPU, збільшує енергоспоживання і підвищує температуру навіть тоді, коли «корисна» обчислювальна робота відносно невелика [23, 24].

Тепловий режим одноплатних комп'ютерів є одним із найпомітніших обмежень у тривалих сценаріях [25, 26]. Компактність плати, невелика площа розсіювання тепла та відсутність потужного охолодження за замовчуванням призводять до швидкого росту температури SoC під навантаженням [27, 28]. У багатьох моделях реалізовано температурний захист: при досягненні порогів система автоматично знижує частоту й переходить у більш «холодні» стани, тобто активує тротлінг [29, 30]. У такому режимі продуктивність падає не тому, що задача стала складнішою, а тому, що платформа захищає себе від перегріву [31]. Це прямо впливає на якість сервісу: час відповіді збільшується, черги задач ростуть, а короткі піки активності можуть провокувати циклічні коливання «прискорення–нагрів–обмеження» [32, 33]. У підсумку контроль температури через керування частотою та навантаженням стає базовою умовою стабільної роботи одноплатного комп'ютера [34, 35].

Важливою перевагою одноплатних комп'ютерів є наявність інтерфейсів, які зближують їх із класичними вбудованими системами [36, 37]. GPIO-виводи, шини I<sup>2</sup>C, SPI, UART, а також PWM-канали дозволяють підключати датчики та виконавчі пристрої [38, 39]. У контексті теми це відкриває шлях до розширеної телеметрії: окрім системних показників, які ОС надає через стандартні файли та утиліти (частота, завантаження CPU, температура), може бути додано вимір струму й напруги на лінії живлення через зовнішній модуль або «розумний» датчик живлення [40, 41]. Така комбінація програмної та апаратної телеметрії формує більш реальну картину, де видно не лише внутрішні лічильники ОС, а й фактичні зміни енергоспоживання [42, 43].

Процесорна частина одноплатних комп'ютерів зазвичай має кілька ядер і підтримує динамічне керування частотою та енергетичними станами [44]. На рівні ОС це забезпечено механізмами DVFS і політиками керування частотою

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

(governor), які підлаштовують частоту під поточне навантаження [45, 46]. Однак універсальні політики не враховують специфіку конкретного сервісу та конкретної метрики якості. Наприклад, занадто «агресивне» зниження частоти може погіршувати час відповіді під короткими сплесками, тоді як завищена частота постійно тримає систему теплою і наближає момент тротлінгу [47]. Через це одноплатний комп'ютер добре підходить для перевірки власного регулятора, який спирається на зворотний зв'язок за температурою й завантаженням та забезпечує більш плавну, керовану зміну режимів. Структурну схему одноплатного комп'ютера зображено на рисунку 1.2.



Рисунок 1.2 – Структурна схема одноплатного комп'ютера

Додатковою перевагою одноплатної платформи є відтворюваність експериментів. На відміну від різномірних настільних ПК, одноплатний комп'ютер має стабільну архітектуру, тип живлення та подібні теплові умови. Це спрощує порівняння режимів керування, оскільки зміни в залежностях «продуктивність - температура - енергія» пояснюються алгоритмом керування та сценарієм навантаження, а не випадковими відмінностями апаратури. У підсумку одноплатний комп'ютер у цій кваліфікаційній роботі сформувався як зручна основа для побудови програмно-апаратного засобу, що дозволяє керувати частотою CPU, контролювати нагрів і зменшувати енергоспоживання без відчутної втрати якості сервісу.

## 1.2 Вплив навантаження та частоти процесора на енергоспоживання і тепловий режим

Робота розглядає навантаження та робочу частоту процесора як два ключові фактори, що безпосередньо формують енергоспоживання й тепловий стан обчислювальної платформи [1, 2]. У сучасних одноплатних комп'ютерах центральний процесор уже виконує роль основного споживача енергії, а всі інші вузли, такі як пам'ять, мережеві інтерфейси чи накопичувачі, зазвичай лише доповнюють загальну картину [3, 4]. Унаслідок цього саме режим роботи CPU визначає, чи перебуває система в «холодному» економному стані, чи переходить у зону підвищеного нагріву та зростаючого споживання [5, 6].

Навантаження процесора характеризує частку часу, протягом якого обчислювальні ядра зайняті виконанням задач [7]. За низького навантаження значна частина тактів простоює, або ж процесор переходить у стани зниженого споживання, де частина блоків тимчасово вимикається чи працює у спрощеному режимі [8, 9]. За високого навантаження ядра постійно виконують інструкції, активно використовуються кеші, блоки обробки чисел з плаваючою комою та векторні розширення, що призводить до суттєвого зростання споживаної потужності [10, 11]. Уже на цьому рівні видно, що одна й та сама частота процесора може давати різний тепловий ефект залежно від характеру задачі: обчислювально інтенсивні алгоритми нагрівають кристал значно швидше, ніж прості операції введення-виведення або очікування подій [12, 13].

Частота процесора визначає швидкість, з якою виконуються інструкції, і водночас впливає на напругу живлення, що подається на ядра [14, 15]. У більшості сучасних систем керування енергоспоживанням підвищення частоти супроводжується підвищенням напруги, оскільки стабільна робота на високих тактових частотах потребує більшого електричного запасу [16, 17]. Через це залежність між частотою та споживаною потужністю набуває нелінійного характеру: збільшення частоти не просто пропорційно підвищує енергоспоживання, а часто призводить до значно швидшого росту

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

тепловиділення [18, 19]. У практичних умовах це означає, що перехід у максимальний режим роботи процесора може різко підняти температуру навіть за відносно короткий проміжок часу [20, 21].

Тепловий режим одноплатного комп'ютера формується як результат балансу між виділеною потужністю та можливістю відведення тепла в навколишнє середовище [22, 23]. Компактність плати, невеликі радіатори або їх відсутність, а також обмежена вентиляція призводять до того, що запас по охолодженню є досить скромним [24, 25]. За тривалого високого навантаження та підвищеної частоти температура кристала досягає порогових значень, після чого апаратні механізми захисту активують обмеження продуктивності [26, 27]. У таких умовах система вже не підтримує заявлену частоту, а примусово знижує її, щоб уникнути перегріву [28]. Це явище відоме як тротлінг і воно безпосередньо впливає на стабільність часу відповіді та загальну передбачуваність роботи прикладних програм [29, 30].

З погляду енергоспоживання важливо враховувати не лише миттєву потужність, а й інтегральні витрати енергії за певний інтервал часу [31, 32]. Висока частота дозволяє виконати задачу швидше, проте при цьому споживана потужність зростає [33]. Низька частота, навпаки, зменшує миттєві витрати, але подовжує час виконання [34]. У підсумку загальні витрати енергії можуть виявитися як меншими, так і більшими залежно від конкретного балансу між швидкодією та потужністю [35, 36]. Саме через це у практичних системах уже недостатньо орієнтуватися лише на максимальну або мінімальну частоту - більш ефективним підходом стає пошук такого режиму, де задача виконується без зайвого нагріву та без надмірних енергетичних втрат [37, 38].

Характер навантаження також має суттєвий вплив на теплову динаміку [39]. Короткі, але інтенсивні сплески обчислень можуть не встигнути суттєво підвищити температуру, оскільки теплова інерція системи згладжує піки [40, 41]. Натомість тривалі рівномірні навантаження поступово прогрівають плату до стаціонарного стану, у якому навіть незначне підвищення частоти може стати критичним [42, 43]. У межах кваліфікаційної роботи це означає, що для коректної

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

оцінки впливу частоти та навантаження необхідно враховувати не лише середні значення, а й часові характеристики процесу: швидкість нагріву, швидкість охолодження та поведінку системи при зміні режимів [44, 45].

Окрему роль відіграє взаємодія між планувальником задач операційної системи та механізмами керування частотою [46]. Коли в системі з'являється кілька конкурентних процесів, розподіл CPU-часу між ними може створювати ситуації, за яких процесор то різко переходить у високопродуктивний стан, то знову повертається до економного режиму. Такі коливання частоти та напруги спричиняють додаткові втрати енергії та можуть провокувати нестабільний тепловий режим [47]. Через це керування навантаженням і частотою доцільно розглядати як єдину задачу, де важливо не лише реагувати на поточний стан, а й забезпечувати плавність переходів між режимами.

У межах кваліфікаційної роботи вплив навантаження та частоти процесора розглянуто як основа для побудови регулятора, що спирається на зворотний зв'язок за температурою та завантаженням CPU. Такий підхід дозволяє підтримувати систему в зоні, де тепловий режим залишається керованим, а енергоспоживання - помірним, без різких стрибків продуктивності. Це створює передумови для більш передбачуваної роботи сервісів і відкриває можливість експериментально дослідити співвідношення між швидкодією, нагрівом та витратами енергії в реальних сценаріях використання одноплатного комп'ютера.

### 1.3 Огляд підходів до керування продуктивністю та енергоспоживанням у сучасних операційних системах

Найбільш поширеною платформою для таких механізмів залишається Linux, оскільки саме вона встановлюється на більшість одноплатних комп'ютерів сімейств Raspberry Pi, Orange Pi та подібних пристроїв [1, 2]. У межах цієї системи сформовано стандартну підсистему керування частотою процесора, відому як `cpufreq`, яка надає набір готових політик керування режимами роботи CPU [3, 4]. Ці політики вже багато років застосовуються в

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

реальних системах і вважаються базовим інструментом для досягнення компромісу між швидкістю та енергетичними витратами [5, 6].

Одним із найпростіших і водночас найпоказовіших рішень є політика, що фіксує частоту процесора на максимальному рівні [7]. У середовищі Linux цей режим широко використовується на серверних системах і робочих станціях, де пріоритетом є стабільна висока продуктивність [8, 9]. На одноплатних комп'ютерах такий підхід також інколи застосовується, наприклад, у мультимедійних або обчислювальних стендах, де важлива передбачувана швидкодія [10]. Проте в практичних умовах уже помічено, що постійна робота на максимальній частоті призводить до підвищеного енергоспоживання та швидкого нагріву, а за відсутності активного охолодження - до переходу в режим тротлінгу [11, 12]. Через це подібне рішення рідко використовується як універсальне для тривалих безперервних сценаріїв роботи на компактних платформах [13, 14].

Протилежним за ідеологією є режим, орієнтований на економію енергії, у якому процесор більшу частину часу працює на знижених частотах [15, 16]. Такий підхід знайшов застосування в мобільних і вбудованих системах, де пріоритетом є мінімізація споживання та зниження тепловиділення [17]. На одноплатних комп'ютерах цей режим часто використовується у фонових сервісах, які не потребують високої швидкодії, наприклад у системах моніторингу або мережевих шлюзах із помірним трафіком [18, 19]. Практика показує, що за такого налаштування температура платформи залишається значно нижчою, проте під час появи пікового навантаження час відповіді системи може помітно збільшуватися, що не завжди прийнятно для інтерактивних або часо-критичних задач [20, 21].

Найбільш поширеним у реальних інсталяціях став адаптивний підхід, який автоматично змінює частоту процесора залежно від поточного завантаження [22, 23]. У дистрибутивах Linux для цього використовується політика, що динамічно підвищує частоту при зростанні навантаження та знижує її в періоди простою [24, 25]. Саме цей режим зазвичай увімкнено за замовчуванням на одноплатних

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

комп'ютерах із заводських образів операційної системи [26]. На практиці таке рішення дозволяє без додаткового налаштування отримати прийнятний баланс між продуктивністю та енергоспоживанням у більшості побутових сценаріїв [27, 28]. Водночас досвід експлуатації показує, що універсальний алгоритм реагує лише на поточне завантаження CPU і не враховує, наприклад, повільний ріст температури або специфіку тривалих стабільних навантажень, через що система може періодично входити в зону тротлінгу [29, 30].

Окрім керування частотою, у сучасних операційних системах активно застосовуються механізми переходу процесора в різні стани енергозбереження під час простою [31, 32]. Ці стани використовуються практично у всіх дистрибутивах Linux і мобільних системах, де значна частина часу припадає на очікування подій [33]. У реальних рішеннях, таких як мережеві маршрутизатори на базі одноплатних комп'ютерів або домашні сервери, завдяки цим механізмам досягається суттєве зниження споживання у фоновому режимі [34, 35]. Проте в сценаріях із частими короткими періодами активності та простою ефект уже не є однозначним, оскільки переходи між станами супроводжуються додатковими витратами часу й енергії [36, 37].

На прикладному рівні також використовуються готові програмні рішення, що намагаються оптимізувати енергоспоживання системи в цілому [38, 39]. У середовищі Linux поширення набули утиліти для централізованого налаштування параметрів живлення, які керують частотою процесора, політиками сну пристроїв введення-виведення та окремими параметрами планувальника [40, 41]. Подібні інструменти активно застосовуються на ноутбуках і серверах для зниження споживання у періоди низького навантаження [42]. На одноплатних комп'ютерах вони також використовуються, зокрема в проєктах домашніх серверів або мультимедійних центрів, де важливо обмежити нагрів у закритому корпусі [43]. Проте такі рішення здебільшого потребують ручного підбору параметрів і не завжди коректно реагують на швидку зміну характеру навантаження [44].

Окрему нішу займають вбудовані платформи та комерційні пристрої, де поверх стандартних механізмів операційної системи реалізовано власні алгоритми керування продуктивністю [45]. У медіаплеєрах, промислових контролерах і мережевому обладнанні часто застосовується підхід, за якого система орієнтується не лише на завантаження CPU, а й на температуру кристала та допустимі межі нагріву корпусу [46]. У таких рішеннях частота процесора знижується завчасно, ще до досягнення критичних температур, що дозволяє уникнути різких стрибків продуктивності та нестабільної поведінки [47]. Подібна логіка вже показала свою ефективність у пристроях, розрахованих на цілодобову роботу без активного охолодження.

Стандартні політики операційної системи, утиліти енергозбереження та спеціалізовані вбудовані алгоритми вже продемонстрували можливість впливу на баланс між продуктивністю, температурою та енергоспоживанням. Водночас їх універсальний або наперед заданий характер обмежує точність налаштування під конкретний сценарій. Це створює передумови для реалізації регулятора, що використовує телеметрію в реальному часі та дозволяє більш гнучко керувати частотою і навантаженням CPU з урахуванням реального теплового стану та вимог до стабільності сервісу.

#### 1.4 Методи енергоефективного планування задач і керування ресурсами процесора

Практика експлуатації показала, що утримання максимальної частоти за будь-яких умов призводить до швидкого нагріву та активації захисних обмежень, тоді як надмірне зниження частоти погіршує час відповіді та зменшує пропускну здатність сервісів [5, 6]. Через це сучасні підходи орієнтуються не на фіксовані режими, а на адаптивне керування, яке враховує поточний стан системи та зміну навантаження в часі [7, 8].

Одним із базових напрямів такого керування є використання динамічного масштабування частоти та напруги процесора [9, 10]. У межах цього підходу

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

процесор уже працює не в одному режимі, а в наборі дискретних станів, між якими система може перемикатися залежно від інтенсивності обчислень [11, 12]. Зниження частоти та напруги у періоди низького навантаження дозволяє помітно зменшити миттєву потужність і, як наслідок, сповільнити нагрів кристала [13, 14]. Підвищення частоти у моменти пікової активності забезпечує необхідну швидкодію без потреби постійно утримувати систему в «гарячому» стані [15, 16]. У реальних умовах ефективність такого підходу визначається не лише самим фактом перемикання частот, а й швидкістю та плавністю цих переходів, оскільки різкі стрибки режимів супроводжуються додатковими енергетичними втратами та коливаннями температури [17, 18].

Важливим доповненням до керування частотою є врахування характеру виконуваних задач на рівні планувальника [19, 20]. У багатозадачних системах різні процеси можуть мати відмінні вимоги до часу відгуку та пропускну здатності [21]. Частина з них орієнтована на інтерактивність і чутлива до затримок, інші ж виконують фонову обробку даних і допускають більш повільний темп роботи [22, 23]. Енергоефективне планування в такому контексті передбачає розподіл процесорного часу таким чином, щоб критичні до затримок задачі отримували ресурси своєчасно, а фонові навантаження виконувалися в більш економних режимах [24, 25]. Це дозволяє зменшити середню частоту роботи процесора без відчутного погіршення якості сервісу для користувача або прикладної системи [26, 27].

Окреме місце займає керування кількістю активних ядер і розподілом навантаження між ними [28, 29]. У багатоядерних одноплатних комп'ютерах можлива ситуація, коли частина ядер залишається майже невикористаною, тоді як інші працюють із високим завантаженням [30]. З точки зору енергоефективності доцільним є або рівномірний розподіл навантаження при помірних частотах, або, навпаки, концентрація обчислень на меншій кількості ядер із переведенням решти в економні стани [31, 32]. Обидва підходи застосовуються на практиці залежно від архітектури процесора та теплових умов [33, 34]. Вибір конкретної стратегії впливає не лише на споживану потужність, а

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

й на розподіл тепла по кристалу, що в компактних платформах може мати принципове значення для стабільності роботи [35, 36].

Ще одним важливим напрямом енергоефективного керування є використання зворотного зв'язку за температурою [37, 38]. На відміну від простих алгоритмів, які реагують лише на завантаження CPU, температурно-орієнтовані методи дозволяють оцінювати фактичний тепловий стан платформи та коригувати режими роботи завчасно, не доводячи систему до порогів захисту [39, 40]. У таких підходах температура розглядається як інтегральний показник, що відображає не лише поточну активність, а й накопичений тепловий ефект від попередніх навантажень [41]. Керування на основі цього сигналу дозволяє згладжувати піки нагріву, підтримувати більш стабільний тепловий режим і уникати циклів різкого зниження продуктивності через тротлінг [42, 43].

На практиці подібні методи часто поєднуються з регуляторами, що мають властивість поступово підлаштовувати керуючий вплив залежно від відхилення параметрів від заданих меж [44]. У контексті процесорних ресурсів це означає, що частота або допустимий рівень навантаження змінюється не стрибком, а плавно, із урахуванням поточного тренду температури та завантаження [45]. Такий підхід дозволяє зменшити коливання режимів роботи, стабілізувати споживання енергії та забезпечити більш передбачувану поведінку системи під змінним навантаженням [46]. Для одноплатних комп'ютерів із обмеженим охолодженням саме плавність і передбачуваність переходів між станами стає критично важливою умовою надійної експлуатації [47].

Значну роль у загальній енергоефективності відіграє і керування допоміжними підсистемами процесора та платформи загалом. Активність підсистем пам'яті, інтерфейсів введення-виведення та мережевих модулів безпосередньо впливає на загальне навантаження і тепловиділення. Тому енергоефективне планування задач уже виходить за межі лише CPU та охоплює оптимізацію доступу до пам'яті, зменшення кількості зайвих операцій введення-виведення, групування обробки даних і зниження частоти переривань. У реальних сценаріях це проявляється у зменшенні фонові активності, скороченні

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

кількості дрібних операцій та переході до більш пакетної обробки, що зменшує як енергетичні витрати, так і теплове навантаження на систему [24, 27, 35].

У межах кваліфікаційної роботи ці методи розглянуто як взаємопов'язані елементи єдиного підходу до керування ресурсами процесора. Динамічне масштабування частоти, адаптивний розподіл задач, використання температурного зворотного зв'язку та згладжування переходів між режимами формують основу для побудови системи, яка не лише реагує на поточне навантаження, а й підтримує стабільний тепловий та енергетичний режим у тривалих сценаріях роботи [37, 42, 46]. Це дозволяє зберігати прийнятну продуктивність без надмірного нагріву та зайвих витрат енергії, що особливо важливо для компактних одноплатних платформ, орієнтованих на безперервну експлуатацію в обмежених умовах охолодження [34, 47].

### 1.5 Постановка задачі створення програмно-апаратного засобу енергоефективного планування задач на одноплатному комп'ютері

У попередніх підрозділах показано, що навантаження і частота процесора безпосередньо формують тепловий режим платформи, а існуючі універсальні політики керування орієнтовані на усереднений сценарій і не враховують специфіку конкретного застосування. Це створило передумови для формування прикладної задачі, спрямованої на розробку власного програмно-апаратного засобу, який забезпечує більш керовану та передбачувану поведінку системи під змінним навантаженням.

У межах цієї кваліфікаційної роботи сформульовано задачу створення засобу, що здійснює безперервний збір телеметрії з одноплатного комп'ютера, аналізує поточний стан системи та на основі цього керує режимами роботи процесора. Під телеметрією в даному контексті розуміються показники завантаження CPU, його робоча частота, температура кристала та, за можливості, оцінка споживаної потужності або непрямі показники енергетичних витрат. Зібрані дані мають використовуватися не лише для візуалізації, а й як вхідні

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

сигнали для регулятора, що формує керуючий вплив на частоту або допустимий рівень навантаження.

Постановка задачі передбачає, що розроблюваний засіб уже працює поверх стандартної операційної системи та не замінює її механізми, а доповнює їх власною логікою прийняття рішень. У межах такої логіки має бути реалізовано принцип зворотного зв'язку, за якого система реагує не лише на миттєве навантаження, а й на накопичений тепловий ефект. Це дозволяє уникати ситуацій, коли частота процесора підвищується до максимуму в умовах, близьких до теплових обмежень, і замість цього підтримувати роботу в стабільнішій зоні, де температура змінюється повільно і прогнозовано. Такий підхід орієнтовано не на досягнення пікової продуктивності за будь-яку ціну, а на забезпечення стійкої та довготривалої роботи без різких провалів швидкодії через тротлінг.

У межах задачі також визначено необхідність реалізації механізму керування, який змінює режими роботи процесора поступово, без різких стрибків між мінімальними та максимальними значеннями. Плавність переходів розглядається як важлива умова зменшення енергетичних втрат і стабілізації теплового режиму. Регулятор має враховувати не лише поточне відхилення температури або завантаження від заданих меж, а й тенденцію їх зміни, що дозволяє завчасно коригувати керуючий вплив і запобігати виходу системи в критичні стани.

Окремим аспектом постановки задачі є забезпечення збереження прийнятної якості сервісу для прикладних задач. Розроблюваний засіб не має призводити до помітних затримок у роботі важливих процесів або до різкого падіння пропускну здатності. Навпаки, передбачається, що за рахунок більш рівномірного теплового та енергетичного режиму система зможе підтримувати стабільні характеристики протягом тривалого часу, без циклів прискорення та вимушеного уповільнення через перегрів. Це означає, що керування частотою і навантаженням має здійснюватися з урахуванням пріоритетів задач і характеру їх виконання, розділяючи фонові та критичні до часу відгуку процеси.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

У рамках цієї кваліфікаційної роботи також поставлено задачу експериментальної перевірки ефективності запропонованого підходу. Для цього передбачено організацію серії випробувань у різних сценаріях навантаження, включно з режимами простою, помірної активності та тривалого інтенсивного обчислення. Результати роботи системи мають бути зафіксовані у вигляді графіків, що відображають взаємозв'язок між продуктивністю, температурою та енергоспоживанням. Саме ці залежності дозволяють оцінити, наскільки запропонований засіб забезпечує більш рівномірний і керований режим роботи порівняно зі стандартними налаштуваннями операційної системи.

У підсумку постановка задачі зводиться до створення програмно-апаратного засобу, який інтегрується в середовище одноплатного комп'ютера, здійснює збір і аналіз телеметрії, керує режимами роботи процесора на основі зворотного зв'язку та забезпечує експериментальну базу для оцінки енергоефективності. Такий засіб має слугувати практичним інструментом для демонстрації можливості зниження теплового навантаження і споживання енергії без суттєвої втрати продуктивності, що відповідає загальній меті цієї кваліфікаційної роботи та логічно завершує теоретичну частину першого розділу.

Крім цього, у постановці задачі враховано необхідність збереження прозорості роботи розроблюваного засобу, оскільки керування частотою процесора має бути зрозумілим і контрольованим. Для цього передбачено фіксацію основних параметрів системи в журналі, де відображаються час вимірювання, температура процесора, рівень завантаження, поточна частота, визначений режим роботи та прийнята керуюча дія. Такий підхід дозволяє не лише спостерігати за поточним станом одноплатного комп'ютера, а й надалі пояснювати причини зміни режимів роботи. Журналювання створює основу для порівняння стандартного керування операційної системи із запропонованим підходом, а також дає можливість перевірити, чи справді алгоритм реагує на зростання температури завчасно, а не лише після наближення до критичних значень.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

## 2 РОЗРОБЛЕННЯ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ ЕНЕРГОЕФЕКТИВНОГО ПЛАНУВАННЯ ЗАДАЧ НА ОДНОПЛАТНОМУ КОМП'ЮТЕРІ

### 2.1 Загальна архітектура програмно-апаратного засобу енергоефективного планування задач

Після розкриття теоретичних аспектів енергоспоживання, теплового режиму та керування частотою процесора логічним продовженням стала побудова архітектури засобу, який поєднує збір телеметрії, аналіз стану системи, прийняття керуючого рішення та зміну режиму роботи процесора. Такий підхід узгоджено із загальною метою кваліфікаційної роботи, де основну увагу приділено не лише зменшенню енергоспоживання, а й підтриманню стабільної поведінки одноплатного комп'ютера під змінним навантаженням.

Загальну архітектуру засобу сформовано та подано на рисунку 2.1 як поєднання апаратної платформи, операційної системи, програмного модуля збору даних, блока аналізу телеметрії, керуючого регулятора та модуля журналювання результатів. У центрі цієї архітектури розміщено одноплатний комп'ютер, який виконує прикладні задачі та одночасно надає системні показники для подальшого аналізу. У ролі базового середовища використано Linux-подібну операційну систему, оскільки саме вона забезпечує доступ до механізмів керування частотою процесора, показників завантаження, температурних даних і стану виконуваних процесів. Завдяки цьому програмно-апаратний засіб не потребує повної заміни штатних механізмів ОС, а доповнює їх власною логікою керування на основі зворотного зв'язку.

На рисунку 2.1 зображено загальну архітектуру програмно-апаратного засобу, у якій показано зв'язок між одноплатним комп'ютером, операційною системою, модулем збору телеметрії, блоком аналізу, регулятором режимів роботи процесора та модулем збереження результатів. У схемі сформовано замкнений контур керування: система зчитує поточні параметри, аналізує їх, визначає потрібний режим роботи CPU, застосовує керуючу дію та знову

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

перевіряє результат. Це дозволяє показати, що запропонована архітектура працює не як одноразове налаштування частоти, а як постійний процес спостереження та коригування режиму роботи платформи.



Рисунок 2.1 – Загальна архітектура програмно-апаратного засобу енергоефективного планування задач на одноплатному комп'ютері

Апаратна частина архітектури представлена одноплатним комп'ютером, джерелом живлення, системою охолодження та, за потреби, додатковим модулем вимірювання споживання енергії. Основним об'єктом керування виступає процесор одноплатного комп'ютера, оскільки саме його частота, завантаження і температура найбільше впливають на тепловий режим та енергетичні витрати. Джерело живлення забезпечує стабільну роботу плати, а система охолодження створює фізичні умови, у межах яких програмний регулятор може підтримувати прийнятну температуру. Додатковий вимірювальний модуль не є обов'язковим для базової роботи засобу, але його наявність дозволяє точніше оцінити різницю

між стандартним режимом роботи ОС і запропонованим режимом енергоефективного керування.

Програмна частина архітектури побудована навколо модуля збору телеметричних даних. Цей модуль періодично зчитує показники завантаження процесора, поточну частоту, температуру кристала, стан виконуваних задач і часові характеристики роботи системи. У межах архітектури передбачено, що телеметрія не обмежується пасивним спостереженням. Зібрані значення передаються до блока аналізу, де визначено поточний стан платформи: спокійний режим, помірно навантаження, інтенсивне навантаження або наближення до небажаного теплового стану. Саме цей перехід від простого зчитування параметрів до їх змістовної оцінки є важливою частиною побудованого засобу.

Блок аналізу телеметрії виконує роль проміжної ланки між вимірюванням і керуванням. У ньому показники не розглядаються окремо один від одного, оскільки для енергоефективного планування важливим є їх спільний вплив. Наприклад, високе завантаження процесора саме по собі ще не завжди означає потребу негайно знижувати частоту, якщо температура залишається в безпечному діапазоні та не має швидкого зростання. Натомість помірно навантаження при високій температурі може вимагати обережнішого режиму, оскільки платформа вже накопичила тепловий запас. Через це в архітектурі передбачено аналіз не лише поточного значення температури, а й тенденції її зміни в часі.

Керуючий регулятор сформовано як окремий програмний блок, який на основі результатів аналізу визначає подальшу дію. До таких дій належить утримання поточної частоти, плавне підвищення частоти при безпечному тепловому стані, зниження частоти при наближенні до температурного обмеження або обмеження інтенсивності фонових задач. Регулятор не працює за принципом різкого перемикавання між мінімальним і максимальним режимом. У його логіці передбачено поступову зміну режимів, що зменшує коливання

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

температури та дозволяє уникнути ситуації, коли система спочатку швидко нагрівається, а потім

Важливе місце в архітектурі займає модуль взаємодії з операційною системою. Через нього програмний засіб отримує доступ до системних файлів, службових утиліт і механізмів керування частотою процесора. На практичному рівні це може бути реалізовано через стандартні інтерфейси Linux, які дозволяють зчитувати поточну частоту CPU, температуру SoC і доступні режими роботи процесора. Також цей модуль відповідає за застосування керуючих дій, сформованих регулятором. Завдяки такому розподілу логіка прийняття рішення відокремлена від низькорівневого доступу до системних параметрів, що робить архітектуру більш зрозумілою і зручною для подальшої реалізації.

Окремо в архітектурі передбачено модуль журналювання та підготовки результатів. Його призначення полягає у збереженні послідовності телеметричних записів, значень частоти, температури, завантаження процесора та обраного режиму керування. Такі дані необхідні для подальшого порівняння різних режимів роботи: стандартного керування операційної системи, режиму фіксованої частоти та запропонованого адаптивного режиму. Збереження журналів також дозволяє будувати графіки, які наочно відображають зміну температури, частоти та навантаження в часі. У третьому розділі ці дані можуть бути використані для перевірки працездатності засобу та оцінювання ефективності запропонованої логіки.

Архітектуру засобу побудовано з урахуванням того, що одноплатний комп'ютер часто працює у тривалих сценаріях без постійного втручання користувача. Через це система має виконувати керування автоматично, без ручного перемикавання режимів. Після запуску програмний модуль переходить до циклічного збирання телеметрії, а регулятор поступово коригує режим роботи процесора відповідно до поточного стану. Такий підхід особливо важливий для компактних платформ, де невеликий запас охолодження швидко призводить до нагріву, а різкі зміни навантаження можуть викликати нестабільну поведінку частоти.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

У запропонованій архітектурі також враховано потребу в розділенні прикладних і фонових задач. Прикладні задачі становлять основне навантаження, заради якого використовується одноплатний комп'ютер, тоді як фонові процеси забезпечують обслуговування системи, запис журналів, обробку допоміжних даних або моніторинг. Для енергоефективного планування важливо, щоб керування частотою не створювало відчутного погіршення роботи основних задач. Через це архітектура передбачає оцінювання стану системи не лише за температурою, а й за характером навантаження. Якщо платформа працює в безпечному режимі, регулятор може залишати продуктивність на достатньому рівні. Якщо температура поступово наближається до небажаної зони, пріоритет отримує стабілізація теплового стану.

Перевагою такої архітектури є її модульність. Кожна частина засобу виконує окрему функцію: модуль збору даних отримує телеметрію, блок аналізу визначає стан системи, регулятор формує рішення, модуль взаємодії з ОС застосовує це рішення, а журналювання фіксує отримані результати. Завдяки цьому окремі компоненти можна змінювати або вдосконалювати без повної перебудови всієї системи. Наприклад, у майбутньому може бути змінено спосіб оцінювання температурного тренду, додано точніше вимірювання енергоспоживання або розширено набір режимів керування процесором. При цьому загальна логіка роботи засобу залишається сталою.

Запропонована архітектура також має практичну цінність для експериментального порівняння режимів. Оскільки всі показники фіксуються в журналі, можна відтворити поведінку системи в різних сценаріях навантаження та оцінити, наскільки плавно змінюється частота, як швидко зростає температура і чи вдається уникнути тротлінгу. У межах кваліфікаційної роботи це дозволяє перейти від теоретичного опису енергоефективного планування до практичного програмно-апаратного засобу, який має чітку структуру, зрозумілий потік даних і керуючий механізм.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

## 2.2 Обґрунтування вибору апаратних і програмних компонентів системи

Вибір апаратних і програмних компонентів програмно-апаратного засобу енергоефективного планування задач виконано з урахуванням мети кваліфікаційної роботи, особливостей одноплатних комп'ютерів та потреби у практичній перевірці запропонованого підходу. Для цієї системи важливо не просто зібрати телеметричні дані, а забезпечити замкнений цикл керування, у якому поточний стан платформи впливає на подальший режим роботи процесора. Через це компоненти підбрано так, щоб вони підтримували зчитування системних параметрів, зміну частоти CPU, збереження результатів і подальше порівняння режимів роботи.

Основою апаратної частини обрано одноплатний комп'ютер, оскільки саме така платформа найкраще відповідає темі кваліфікаційної роботи. Одноплатний комп'ютер поєднує компактність, невисоке енергоспоживання, доступну вартість і можливість запуску повноцінної операційної системи. На відміну від мікроконтролера, така платформа виконує кілька процесів одночасно, підтримує файлову систему, мережеві служби, засоби планування задач і механізми керування частотою процесора. У межах цієї роботи це має принципове значення, оскільки енергоефективне планування задач розглядається не як просте увімкнення або вимкнення навантаження, а як керування роботою CPU в середовищі операційної системи.

Як базову апаратну платформу доцільно використати одноплатний комп'ютер класу Raspberry Pi або подібне ARM-рішення, яке має кілька процесорних ядер, підтримує Linux-подібну операційну систему та надає доступ до системної телеметрії. Такий вибір пояснюється тим, що подібні плати широко застосовуються в навчальних стендах, домашніх серверах, системах моніторингу, периферійних обчисленнях і компактних автоматизованих пристроях. Для них характерні саме ті обмеження, які розглянуто в першому розділі: обмежене охолодження, залежність теплового режиму від навантаження,

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

потреба у стабільному живленні та ризик зниження продуктивності через перегрів.

Важливою перевагою одноплатного комп'ютера є доступність системних показників без використання складних зовнішніх вимірювальних пристроїв. Температура процесора, поточна частота, рівень завантаження CPU та стан окремих процесів можуть зчитуватися засобами операційної системи. Це спрощує побудову програмної частини, оскільки основна телеметрія формується без додаткового апаратного ускладнення. Водночас архітектура системи не обмежує можливість подальшого підключення модуля вимірювання струму та напруги, якщо потрібно отримати точніші дані про фактичне енергоспоживання. У базовому варіанті достатньо системних показників, оскільки головною задачею є перевірка логіки керування частотою і тепловим режимом.

Джерело живлення обрано як окремий важливий компонент системи, адже стабільність напруги безпосередньо впливає на роботу одноплатного комп'ютера. Для практичної реалізації засобу потрібне джерело живлення з достатнім запасом струму, яке здатне підтримувати роботу плати під навантаженням. Якщо живлення нестабільне, результати вимірювань можуть спотворюватися: частота процесора знижується не через роботу регулятора, а через проблеми з живленням; температура змінюється нерівномірно; можливі перезапуски або помилки запису даних. Через це в архітектурі системи передбачено використання якісного адаптера живлення та короткого кабелю з нормальним перерізом провідників.

Система охолодження також врахована як частина апаратної основи. Для одноплатного комп'ютера може використовуватися пасивний радіатор, активний вентилятор або корпус із покращеною вентиляцією. У межах цієї роботи охолодження не розглядається як головний засіб зниження температури, але воно створює стабільні умови для порівняння режимів роботи. Якщо охолодження надто слабе, платформа швидко переходить у тротлінг незалежно від програмної логіки. Якщо охолодження занадто потужне, вплив алгоритму керування на тепловий режим стає менш помітним. Тому для

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

експериментального стенда доцільною є помірна конфігурація охолодження, яка дозволяє побачити різницю між стандартним і запропонованим режимом керування. У таблиці 2.1 наведено склад основних компонентів, які сформували апаратну та програмну основу засобу.

Таблиця 2.1 – Обрані апаратні та програмні компоненти програмно-апаратного засобу

Компонент	Призначення в системі	Причина вибору
Одноплатний комп'ютер	Виконання прикладних задач і надання системної телеметрії	Підтримує Linux, багатозадачність і керування частотою CPU
Джерело живлення	Стабільне живлення плати під навантаженням	Зменшує ризик збоїв і спотворення результатів
Система охолодження	Відведення тепла від процесора	Дає змогу перевірити керування температурним режимом
Linux-подібна ОС	Середовище запуску програмного засобу	Надає доступ до <code>cpufreq</code> , системних файлів і процесів
Системні інтерфейси ОС	Зчитування температури, частоти та навантаження CPU	Дозволяють працювати без складного низькорівневого драйвера
CSV/SQLite-журнал	Збереження телеметрії та результатів роботи	Спрощує подальшу побудову графіків і порівняння режимів
Засоби візуалізації	Побудова графіків температури, частоти й навантаження	Дають змогу оцінити ефективність запропонованого підходу

У ній показано, що кожен компонент виконує окрему роль і безпосередньо пов'язаний із загальною логікою енергоефективного планування. Одноплатний комп'ютер виступає центральною платформою, Linux забезпечує доступ до механізмів керування, Python реалізує логіку роботи, а журналювання та візуалізація створюють основу для подальшого аналізу результатів.

Для програмної частини обрано Linux-подібну операційну систему, оскільки вона надає стандартні засоби роботи з процесами, плануванням задач, частотою процесора та системною телеметрією. Така ОС уже має вбудовані механізми керування продуктивністю, зокрема підсистему `cpufreq`, яка дозволяє переглядати доступні частоти, поточний режим і політики керування процесором. У цій роботі стандартні механізми ОС не відкинута, а використано як основу для побудови власного регулятора. Це означає, що запропонований засіб не втручається в апаратний рівень напряму, а працює через наявні системні інтерфейси.

Вибір Linux також пояснюється відкритістю та гнучкістю цієї платформи. На відміну від закритих систем, Linux дозволяє зчитувати службові параметри через файлову систему, запускати фонові служби, автоматизувати збір даних і керувати процесами за допомогою стандартних команд. Для кваліфікаційної роботи це зручно, оскільки всі етапи роботи засобу можна показати прозоро: звідки надходять дані, як вони обробляються, де зберігаються і яким способом застосовується керуюча дія. Це спрощує не лише реалізацію, а й подальше пояснення роботи системи в третьому розділі.

Як основну мову реалізації програмних модулів обрано Python. Такий вибір пов'язаний із тим, що Python добре підходить для швидкої розробки службових програм, роботи з файлами, обробки числових даних і формування журналів. У межах засобу Python може виконувати кілька функцій: зчитування телеметрії, розрахунок середніх значень, визначення температурного тренду, прийняття рішення щодо режиму CPU та запис результатів у файл. Крім того, Python має достатню кількість бібліотек для подальшої побудови графіків, що важливо для оцінювання ефективності системи.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

Разом із Python можуть застосовуватися окремі системні команди та Bash-скрипти. Їх доцільно використовувати для простих операцій, пов'язаних із перевіркою доступних частот, зміною політик керування CPU або запуском допоміжних тестових навантажень. Такий підхід дозволяє не перевантажувати основну програму зайвими низькорівневими діями. Python виконує роль центральної логіки, а системні команди забезпечують взаємодію з операційною системою. У результаті програмна частина засобу залишається зрозумілою, модульною і достатньо простою для перевірки.

Для збереження результатів обрано журналювання у форматі CSV або SQLite. Формат CSV є простим і зручним для початкової реалізації, оскільки його можна переглядати у звичайному табличному редакторі та швидко використовувати для побудови графіків. SQLite доцільно застосувати, якщо потрібно зберігати більший обсяг даних, виконувати вибірки за часом або порівнювати кілька експериментальних запусків. У межах цієї роботи обидва варіанти є придатними, але базова реалізація може спиратися саме на CSV-журнал, оскільки він достатній для фіксації температури, частоти, завантаження, обраного режиму та часу вимірювання.

Засоби візуалізації обрано як допоміжну програмну частину, необхідну для подальшого аналізу результатів. Для побудови графіків можуть використовуватися Python-бібліотеки або табличний редактор. Основними графіками мають стати зміна температури в часі, зміна частоти CPU, динаміка завантаження процесора та порівняння режимів роботи. Такі графіки дозволяють не лише показати числові результати, а й візуально підтвердити, чи справді запропонований засіб зменшує різкі коливання частоти та підтримує більш стабільний тепловий режим.

На рисунку 2.2 зображено структуру апаратних і програмних компонентів засобу енергоефективного планування задач. У схемі показано, що апаратний рівень сформовано одноплатним комп'ютером, джерелом живлення, системою охолодження та можливим модулем вимірювання енергоспоживання. Програмний рівень представлено операційною системою, модулем збору

телеметрії, блоком аналізу, регулятором частоти, журналом результатів і засобами візуалізації. Така схема дозволяє наочно показати, що обрані компоненти не існують окремо, а працюють як єдина система керування.



Рисунок 2.2 – Структура апаратних і програмних компонентів засобу енергоефективного планування задач

У виборі компонентів особливу увагу приділено простоті реалізації та можливості відтворення експериментів. Система не потребує складної електронної частини, спеціального промислового контролера або власного драйвера ядра. Основні функції реалізовано на рівні користувацького програмного забезпечення, яке працює поверх стандартної ОС. Це робить засіб доступним для практичної перевірки на типовому одноплатному комп'ютері та дозволяє зосередити увагу саме на логіці енергоефективного планування, а не на складному апаратному проектуванні.

Разом із цим обрані компоненти забезпечують достатню гнучкість для розширення системи. Якщо на першому етапі зчитуються лише температура,

частота й завантаження CPU, то в подальшому може бути додано вимірювання споживаної потужності, мережеву передачу телеметрії або вебінтерфейс для віддаленого перегляду результатів. Така можливість важлива, оскільки програмно-апаратний засіб має не лише виконати поставлену задачу в межах кваліфікаційної роботи, а й залишити простір для подальшого вдосконалення.

### 2.3 Організація збору телеметричних даних з одноплатного комп'ютера

Організація збору телеметричних даних є однією з основних частин програмно-апаратного засобу енергоефективного планування задач, оскільки саме зібрані показники формують основу для подальшого аналізу стану платформи та прийняття керуючих рішень. У попередніх підрозділах сформовано загальну архітектуру засобу та обґрунтовано вибір апаратних і програмних компонентів. У цьому підрозділі розглянуто, яким способом одноплатний комп'ютер надає інформацію про власний стан, які параметри потрібно зчитувати та як ці дані перетворюються на придатний для обробки телеметричний запис.

У межах кваліфікаційної роботи телеметрію розглянуто не як допоміжний набір службових значень, а як основний інформаційний потік системи керування. Саме на основі телеметрії регулятор отримує уявлення про те, як поводить себе одноплатний комп'ютер під поточним навантаженням, чи наближається процесор до небажаного температурного режиму, чи є підстави знижувати частоту або, навпаки, залишати продуктивність на достатньому рівні. Через це збір даних організовано так, щоб система отримувала не окремі випадкові значення, а послідовність вимірювань у часі. Така послідовність дозволяє аналізувати не лише поточний стан, а й динаміку зміни температури, завантаження та частоти процесора.

Основним джерелом телеметричних даних виступає сама операційна система одноплатного комп'ютера. Linux-подібне середовище надає доступ до службових файлів, системних інтерфейсів і команд, через які можна отримати

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

інформацію про стан CPU, його частоту, рівень завантаження, температуру кристала та активність виконуваних процесів. Це дозволяє реалізувати збір даних без складного втручання в ядро ОС і без розроблення окремого драйвера. Такий підхід добре відповідає практичній спрямованості роботи, оскільки програмний засіб працює поверх штатної системи та використовує вже наявні можливості платформи.

До базового набору телеметричних параметрів включено завантаження процесора, поточну частоту CPU, температуру процесорного кристала, часову мітку вимірювання, стан обраного режиму керування та, за можливості, непрямую або пряму оцінку енергоспоживання. Завантаження CPU показує, наскільки активно процесорні ядра виконують задачі в конкретний момент часу. Частота процесора відображає поточний режим продуктивності, у якому працює система. Температура кристала характеризує накопичений тепловий ефект, який формується не тільки миттєвим навантаженням, а й попередньою роботою платформи. Часова мітка потрібна для побудови графіків і порівняння режимів, а запис активного режиму керування дозволяє зрозуміти, яке саме рішення прийняв регулятор у певний момент.

У роботі збір телеметрії організовано циклічно. Після запуску програмний модуль переходить до періодичного опитування системних параметрів із заданим інтервалом. Надто короткий інтервал може створити зайве фонове навантаження, оскільки сама система моніторингу почне споживати помітну частину ресурсів. Надто довгий інтервал, навпаки, може приховати короткі піки навантаження або швидкі зміни температури. Через це інтервал збору даних підібрано як компроміс між точністю спостереження та мінімальним впливом засобу на роботу одноплатного комп'ютера. Для базової реалізації достатнім є інтервал у кілька секунд, оскільки теплова динаміка процесора змінюється не миттєво, а з певною інерцією.

На рисунку 2.3 зображено схему збору та попередньої обробки телеметричних даних одноплатного комп'ютера. У схемі показано послідовність отримання параметрів від операційної системи, зчитування значень

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

температури, частоти та завантаження CPU, формування єдиного телеметричного запису, попередню перевірку коректності даних і передавання результату до блока аналізу. Така структура дозволяє показати, що телеметрія проходить кілька етапів підготовки перед тим, як використовується регулятором режимів роботи процесора.

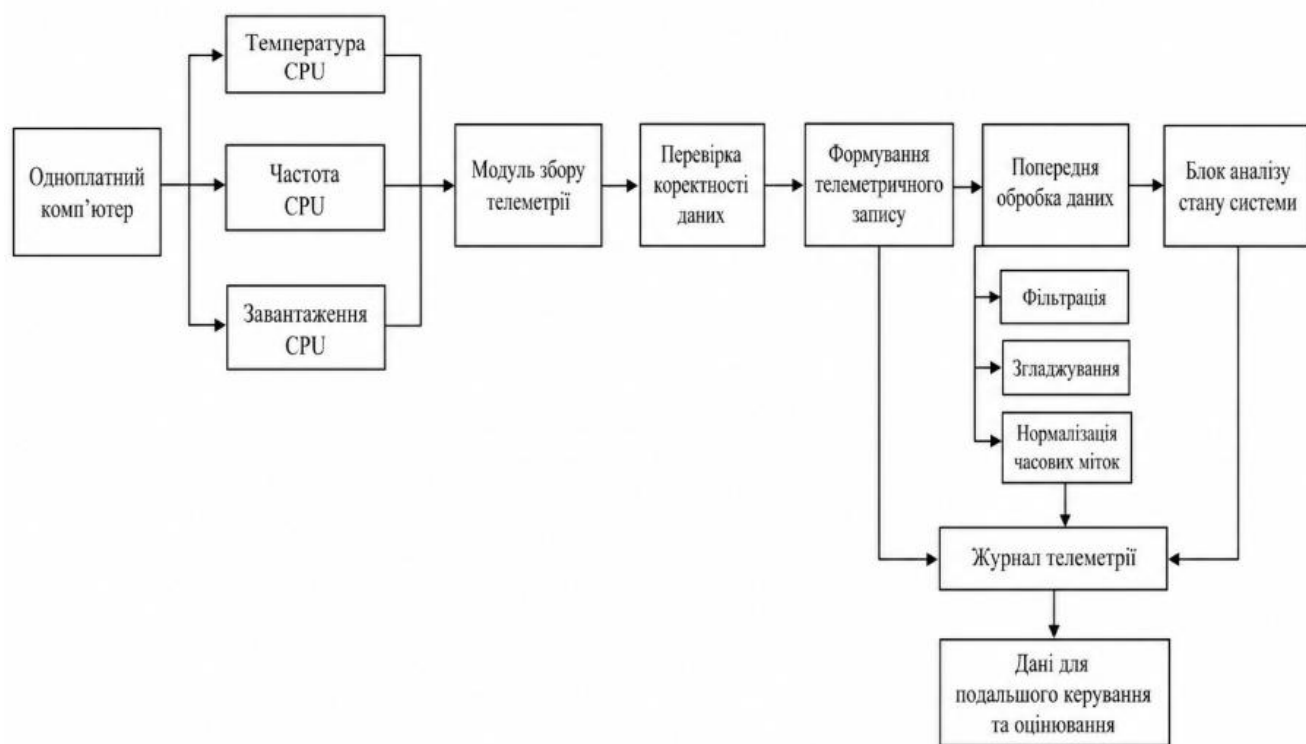


Рисунок 2.3 – Схема збору та попередньої обробки телеметричних даних одноплатного комп'ютера

Процес збору даних починається із запуску програмного модуля моніторингу. На цьому етапі система перевіряє доступність потрібних системних файлів і команд, визначає поточні права доступу та готує структуру журналу. Після цього виконується перше зчитування параметрів, яке використовується як початкова точка для подальшого порівняння. Початковий стан важливий, оскільки температура та частота процесора можуть суттєво відрізнитися залежно від того, чи плата щойно запущена, чи вже працювала під навантаженням. Без

урахування цього моменту подальший аналіз може помилково сприймати нормальне поступове охолодження або нагрівання як результат дії регулятора.

Значення завантаження процесора зчитується на основі статистики використання CPU за певний проміжок часу. Для цього враховується різниця між активними та неактивними періодами роботи процесора. Такий спосіб дає змогу отримати не одиничний випадковий показник, а більш коректну оцінку завантаження за короткий інтервал. У багатоядерній платформі показник може розраховуватися як середнє значення по всіх ядрах або як набір значень для кожного ядра окремо. Для базової реалізації засобу достатнім є узагальнений показник завантаження CPU, оскільки регулятор керує загальним режимом роботи процесора, а не окремими ядрами.

Поточна частота процесора зчитується через системні інтерфейси керування частотою CPU. Це значення важливе, бо саме воно показує, у якому режимі фактично працює процесор у конкретний момент. Частота може відрізнитися від очікуваної через стандартні політики операційної системи, теплові обмеження або механізми захисту. Тому в журналі потрібно фіксувати не тільки бажаний режим, який сформував регулятор, а й фактичну частоту, отриману від системи. Це дозволяє в подальшому порівняти, чи справді керуюча дія застосована коректно і чи не втрутилися апаратні механізми обмеження.

Температура процесорного кристала є одним із найважливіших параметрів для цієї роботи. Вона зчитується через стандартні системні датчики та відображає тепловий стан платформи. На відміну від завантаження CPU, температура змінюється повільніше і має інерційний характер. Саме через це для аналізу важливе не тільки поточне значення, а й напрям зміни температури. Якщо температура поступово зростає навіть при помірному навантаженні, система має врахувати наближення до небажаного режиму заздалегідь. Якщо температура стабільна або знижується, регулятор може залишити поточну частоту без змін або обережно підвищити продуктивність.

Для підготовки даних до подальшого аналізу кожне вимірювання перетворюється на єдиний телеметричний запис. До нього входять час

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

вимірювання, температура, частота CPU, завантаження процесора, поточний режим керування та службові ознаки стану. Такий запис має однакову структуру протягом усієї роботи системи, що спрощує журналювання і подальше порівняння результатів. У разі розширення системи до запису можуть бути додані показники напруги, струму, споживаної потужності або стану охолодження. Проте базова структура залишається незмінною: кожен рядок журналу описує один момент роботи одноплатного комп'ютера.

Після формування телеметричного запису виконується попередня перевірка коректності даних. На цьому етапі система має відкинути або позначити некоректні значення, які можуть з'явитися через тимчасову недоступність датчика, помилку читання системного файлу або недостатні права доступу. Наприклад, якщо температура не зчиталась, регулятор не повинен приймати рішення на основі порожнього значення. У такому випадку може використовуватися останній коректний показник або тимчасове утримання поточного режиму. Це дозволяє уникнути різких і необґрунтованих змін частоти через одиничну помилку вимірювання.

Окрему увагу приділено згладжуванню телеметрії. Миттєві значення завантаження CPU можуть швидко змінюватися навіть під час нормальної роботи системи, тому пряме реагування на кожен короткий стрибок створило б нестабільний режим керування. Для цього в блоці попередньої обробки можуть застосовуватися середні значення за кілька останніх вимірювань. Такий підхід дозволяє відокремити справді стійке зростання навантаження від короткого випадкового піку. Для температури згладжування також має значення, але через її природну інерційність основний акцент зроблено на визначенні тренду, тобто напрямку зміни протягом певного проміжку часу.

Підготовлені телеметричні дані передаються до блока аналізу стану системи. На цьому етапі вони вже мають придатний для обробки вигляд і можуть використовуватися для визначення режиму роботи платформи. Якщо завантаження низьке, температура стабільна, а частота перебуває в середньому або низькому діапазоні, система може залишатися в економному режимі. Якщо

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

навантаження зростає, але температура ще перебуває в безпечній зоні, можливе підвищення продуктивності. Якщо температура наближається до заданої межі, навіть за високого навантаження пріоритет переходить до стабілізації теплового стану. Така логіка стає можливою саме завдяки тому, що телеметрія збирається регулярно та подається в узгодженому форматі.

Журналювання телеметрії виконує подвійну роль. З одного боку, журнал зберігає історію роботи системи для подальшого аналізу. З іншого боку, він є інструментом перевірки коректності самого алгоритму. За записами журналу можна побачити, чи вчасно система реагує на зростання температури, чи не надто часто перемикає режими, чи справді зниження частоти призводить до стабілізації теплового стану. У практичній частині ці записи можуть бути використані для побудови графіків, де відображено залежність температури, частоти та завантаження від часу.

Побудова єдиного формату телеметричного запису має важливе значення для стабільності програмної реалізації. Якщо кожен параметр обробляється окремо, зростає ризик неузгодженості даних, коли температура належить одному моменту часу, частота - іншому, а завантаження CPU - ще іншому. Щоб уникнути цього, усі значення фіксуються в межах одного циклу опитування та записуються як один набір. Такий підхід забезпечує коректне відображення стану системи й дозволяє виконувати порівняння між різними експериментальними режимами.

Під час організації збору телеметрії враховано також вплив самого програмного засобу на навантаження. Модуль моніторингу не повинен створювати значну додаткову активність, оскільки в такому разі він сам спотворює результати вимірювань. Для цього зчитування параметрів реалізується простими операціями, а запис у журнал виконується без надмірної частоти. За потреби дані можуть накопичуватися в пам'яті та зберігатися пакетно, щоб зменшити кількість операцій запису на накопичувач. Це особливо важливо для одноплатних комп'ютерів, які часто використовують microSD-карти з обмеженою швидкістю та ресурсом запису.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

У межах запропонованої системи телеметрія також виконує роль зв'язку між другим і третім розділами кваліфікаційної роботи. У другому розділі визначено, які саме дані зчитуються та як вони організовані, а в третьому розділі ці дані можуть бути використані для демонстрації роботи програмного засобу. На основі журналів можна сформувати графіки температури, частоти CPU та завантаження в різних режимах. Це дозволяє не обмежуватися описом алгоритму, а показати його практичний вплив на поведінку одноплатного комп'ютера.

Організований збір телеметричних даних забезпечує основу для подальшого аналізу стану системи. Завдяки періодичному зчитуванню, перевірці коректності, згладжуванню та журналюванню сформовано інформаційну базу, на якій може працювати регулятор енергоефективного планування задач. У результаті система отримує можливість реагувати не лише на миттєве завантаження CPU, а й на поступову зміну теплового стану. Це створює умови для більш передбачуваного керування частотою процесора та підготовлює основу для розроблення логіки аналізу стану системи в наступному підрозділі.

#### 2.4 Розроблення логіки аналізу стану системи та визначення режиму навантаження

Після організації збору телеметричних даних з одноплатного комп'ютера наступним етапом розроблення програмно-апаратного засобу стало формування логіки аналізу стану системи. Зібрані значення температури, частоти процесора та завантаження CPU самі по собі ще не дають повного уявлення про реальний режим роботи платформи. Для енергоефективного планування задач важливо не тільки отримати набір числових показників, а й правильно їх інтерпретувати. Саме тому в структурі засобу передбачено окремий логічний блок, який перетворює телеметрію на узагальнену оцінку стану одноплатного комп'ютера.

Розроблена логіка аналізу спирається на те, що робота процесора залежить не від одного параметра, а від їх поєднання. Завантаження CPU показує,

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

наскільки активно виконуються обчислювальні задачі. Частота процесора відображає поточний рівень продуктивності, який використовується операційною системою. Температура кристала характеризує тепловий стан платформи та показує, чи має система запас до небажаного перегріву. Окреме значення має також динаміка зміни температури, оскільки однакове числове значення може мати різний зміст залежно від того, зростає температура, знижується чи залишається стабільною.

У межах запропонованого засобу аналіз стану системи побудовано як послідовний процес. Спочатку модуль отримує сформований телеметричний запис, у якому вже містяться значення завантаження CPU, поточної частоти, температури та часу вимірювання. Після цього дані перевіряються на коректність, щоб випадкова помилка зчитування не стала причиною неправильного керування частотою процесора. Якщо отримані значення мають допустимий вигляд, система переходить до оцінювання температурного стану, рівня навантаження та фактичного режиму роботи CPU.

Основну увагу під час аналізу приділено температурі процесора, оскільки саме вона визначає, наскільки безпечним є подальше підвищення продуктивності. У компактних одноплатних комп'ютерах запас охолодження зазвичай обмежений, тому тривала робота на високій частоті може швидко привести до перегріву та тротлінгу. Через це температура розглядається не як другорядний показник, а як один із головних сигналів для регулятора. Якщо температура перебуває в безпечному діапазоні, система може дозволити процесору працювати продуктивніше. Якщо температура наближається до верхньої межі, пріоритет переходить до стабілізації теплового режиму.

Важливо, що в логіці аналізу враховано не лише поточне значення температури, а й напрям її зміни. Наприклад, температура 60 °C може бути прийнятною, якщо вона поступово знижується після завершення інтенсивного навантаження. Та сама температура може бути тривожною, якщо кілька попередніх вимірювань показали швидке зростання. У першому випадку система може залишити поточний режим без змін, а в другому - завчасно обмежити

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

частоту, щоб не допустити переходу до перегріву. Це дозволяє сформувати більш передбачувану поведінку системи та уникнути запізнілої реакції на нагрів.

Завантаження процесора в запропонованій логіці аналізу також не розглядається як самостійна підстава для миттєвої зміни режиму. Одноплатний комп'ютер може мати короткі піки активності через роботу фонових служб, оновлення журналів, мережеві запити або запуск окремих процесів. Якщо реагувати на кожен такий пік, частота CPU буде постійно змінюватися, а це призведе до зайвих коливань енергоспоживання та температури. Тому в системі враховано стійкість навантаження протягом кількох послідовних циклів вимірювання. Короткочасне зростання CPU не сприймається як підстава для різкого переходу в продуктивний режим, а тривале навантаження вже розглядається як значущий стан.

На основі поєднання температури, навантаження та частоти процесора сформовано кілька логічних режимів стану системи. Спокійний режим відповідає ситуації, коли процесор майже не навантажений, температура стабільна, а потреби в підвищенні частоти немає. У такому стані система може підтримувати економний режим, оскільки додаткова продуктивність не використовується. Помірний режим навантаження виникає тоді, коли процесор виконує активні задачі, але температура залишається в безпечній зоні. У цьому випадку допускається утримання середньої частоти або плавне підвищення продуктивності.

Інтенсивний режим визначається тоді, коли процесор має стійке високе навантаження, але тепловий стан ще дозволяє підтримувати продуктивність. У такій ситуації система не повинна передчасно обмежувати CPU, оскільки це може погіршити час виконання задач. Водночас регулятор уже контролює температуру уважніше, щоб не допустити переходу до небажаної зони. Якщо температура починає швидко зростати, інтенсивний режим поступово переходить у температурно напружений. У цьому стані головним завданням стає не максимальна продуктивність, а стримування нагріву та збереження стабільної роботи платформи.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

Температурно напружений режим сформовано як проміжний стан між нормальною роботою та захисним обмеженням. Він потрібний для того, щоб система реагувала на наближення до перегріву раніше, ніж спрацюють апаратні механізми тротлінгу. У цьому режимі частота процесора може бути плавно знижена або утримана від подальшого підвищення. Такий підхід дозволяє уникнути різкої втрати продуктивності, яка виникає тоді, коли платформа вже досягає критичної температури й операційна система або апаратний захист примусово знижують частоту.

Захисний режим використовується в ситуації, коли температура досягає небажаної зони або її подальше зростання створює ризик нестабільної роботи. У цьому стані пріоритетом стає охолодження процесора та зменшення теплового навантаження. Частота CPU знижується, фонові задачі можуть виконуватися менш інтенсивно, а система продовжує збирати телеметрію для контролю повернення до безпечного стану. Вихід із захисного режиму не виконується одразу після невеликого зниження температури. Для цього передбачено стабілізаційний проміжок, протягом якого система переконується, що охолодження справді стало стійким.

Для зменшення частих перемикань між режимами в логіку аналізу введено принцип гістерезису. Його суть полягає в тому, що перехід до обмежувального режиму та повернення до продуктивнішого режиму виконуються за різними умовами. Наприклад, якщо система знизилася частоту через наближення до температурної межі, вона не має одразу підвищувати її після мінімального охолодження. Потрібно, щоб температура стабільно повернулася в безпечний діапазон. Це дозволяє уникнути ситуації, коли процесор постійно переходить між двома сусідніми режимами через незначні коливання температури.

Застосування гістерезису особливо важливе для одноплатних комп'ютерів, оскільки їхній тепловий режим має помітну інерційність. Навіть після зниження частоти температура може ще певний час продовжувати зростати, оскільки тепло вже накопичене в кристалі, радіаторі та корпусі. Якщо система оцінюватиме результат керуючої дії надто швидко, вона може помилково вважати зниження

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

частоти недостатнім і надмірно обмежити продуктивність. Через це аналіз стану передбачає спостереження за кількома послідовними вимірюваннями, а не за одним окремим значенням.

Окремо враховано фактичну частоту процесора. Вона потрібна не тільки для журналювання, а й для перевірки того, як операційна система реально відпрацювала керуючу дію. Бажаний режим, який сформував регулятор, може не завжди повністю збігатися з фактичним станом CPU. Наприклад, апаратний захист може самостійно обмежити частоту через перегрів, або стандартна політика ОС може утримувати процесор у певному діапазоні. Через це аналіз фактичної частоти дозволяє краще зрозуміти поведінку платформи та відокремити дію запропонованого засобу від внутрішніх механізмів операційної системи.

Логіка визначення режиму навантаження має також враховувати характер задач, які виконуються в системі. У загальному вигляді навантаження можна умовно поділити на фонове та прикладне. Фонові процеси не потребують максимальної швидкодії та можуть виконуватися з нижчим пріоритетом. Прикладні задачі мають більшу вагу, оскільки саме вони формують корисну роботу одноплатного комп'ютера. Через це запропонована логіка не зводиться до простого зниження частоти при кожному зростанні температури. Вона має підтримувати баланс між зменшенням енергоспоживання та збереженням прийнятної продуктивності.

На рисунку 2.4 зображено логічну модель аналізу стану системи та визначення режиму навантаження. У схемі показано, що телеметричний запис спочатку проходить перевірку коректності, після чого проаналізовано температуру процесора, тренд її зміни, завантаження CPU та фактичну частоту. Далі сформовано узагальнену оцінку стану платформи, на основі якої визначено режим роботи: спокійний, помірний, інтенсивний, температурно напружений або захисний. Така модель дозволяє показати, що рішення прийнято не за одним окремим показником, а за сукупністю параметрів, які разом описують реальну поведінку одноплатного комп'ютера.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

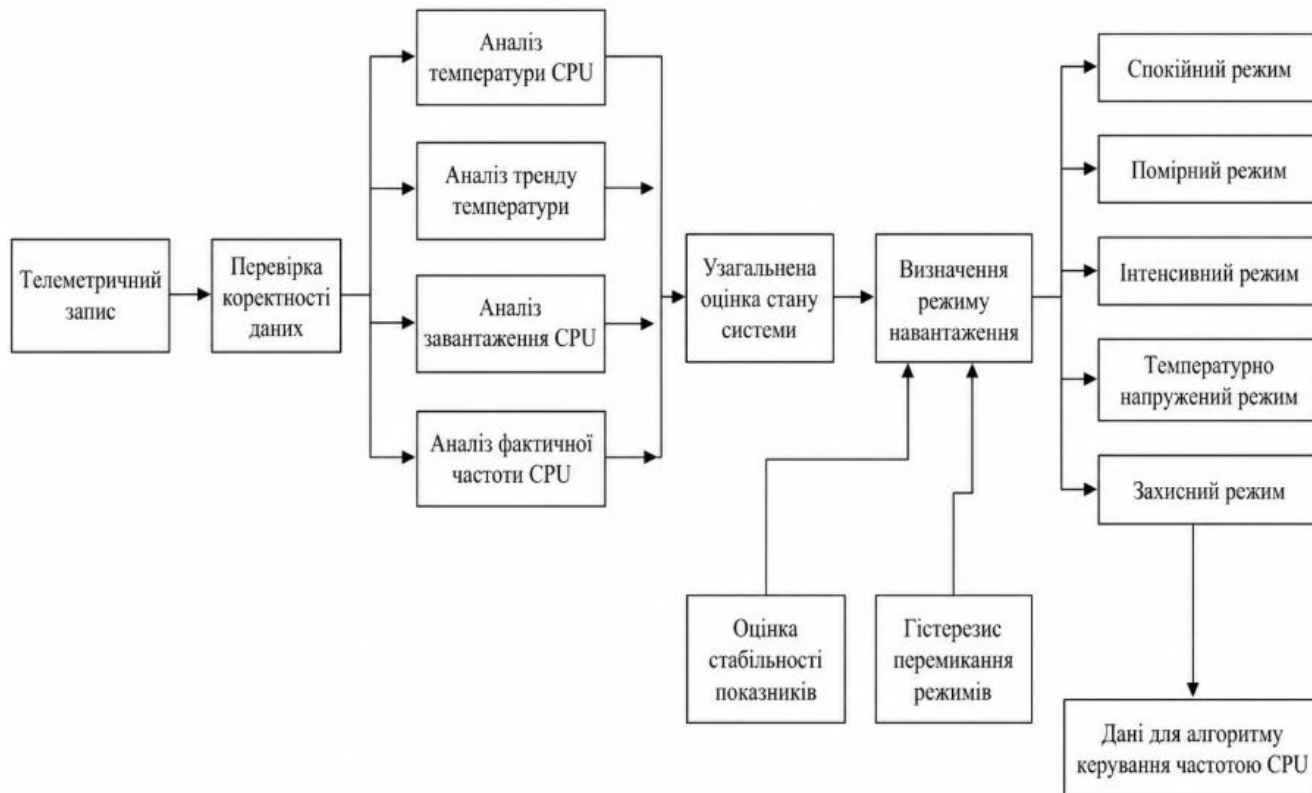


Рисунок 2.4 – Логічна модель аналізу стану системи та визначення режиму навантаження

Після визначення режиму навантаження результат передається до алгоритму енергоефективного планування задач і керування частотою процесора. У цьому місці логічний блок аналізу завершує свою роботу: він не змінює частоту напругу, а лише формує оцінку стану системи. Такий розподіл функцій робить архітектуру більш зрозумілою. Модуль збору телеметрії відповідає за отримання даних, блок аналізу - за їх інтерпретацію, а регулятор - за конкретну керуючу дію. Завдяки цьому окремі частини засобу можна вдосконалювати незалежно одна від одної.

Практична цінність розробленої логіки полягає в тому, що вона дозволяє уникнути надто грубого керування процесором. Без такого аналізу система могла б просто знижувати частоту при досягненні певної температури або підвищувати її при кожному збільшенні навантаження. Такий підхід не враховує ні теплової інерції, ні коротких піків активності, ні фактичного стану платформи. Запропонована логіка працює гнучкіше: вона оцінює стабільність навантаження,

напрям зміни температури, наявність теплового запасу та реальну частоту CPU. Це дозволяє керувати продуктивністю плавніше й не створювати зайвих перемикачів режимів.

Розроблення логіки аналізу стану системи також створює основу для подальшого експериментального порівняння режимів роботи. Якщо в журналі зберігається не тільки температура й частота, а й визначений режим навантаження, стає можливим пояснити, чому в певний момент система знизила або залишила частоту без змін. Це важливо для третього розділу, де результати роботи засобу можуть бути подані у вигляді графіків. На таких графіках можна показати не лише зміну температури чи CPU, а й реакцію регулятора на різні стани платформи.

## 2.5 Розроблення алгоритму енергоефективного планування задач і керування частотою процесора

Після формування логіки аналізу стану системи наступним етапом стало розроблення алгоритму енергоефективного планування задач і керування частотою процесора. Якщо попередній підрозділ пояснює, як телеметричні дані перетворюються на узагальнений режим навантаження, то цей підрозділ розкриває, як саме на основі визначеного режиму система формує керуючу дію. У межах кваліфікаційної роботи алгоритм розглянуто як центральну частину програмно-апаратного засобу, оскільки саме він забезпечує практичний зв'язок між зібраними даними, тепловим станом одноплатного комп'ютера та режимом роботи процесора.

Алгоритм енергоефективного планування побудовано за принципом замкненого керування. Це означає, що система не задає частоту процесора один раз на початку роботи, а постійно перевіряє фактичний стан платформи, аналізує зміну параметрів і за потреби коригує режим CPU. Такий підхід відрізняється від простого ручного вибору політики продуктивності, оскільки враховує не лише

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

бажання зменшити енергоспоживання, а й реальний стан одноплатного комп'ютера в конкретний момент часу. Завдяки цьому частота процесора не знижується без потреби, але й не утримується на завищеному рівні тоді, коли це призводить до зайвого нагріву.

Вхідними даними для алгоритму є результат роботи блока аналізу стану системи. До нього входить визначений режим навантаження, поточна температура процесора, напрям її зміни, фактична частота CPU та рівень завантаження. На основі цих даних алгоритм визначає, яку дію потрібно виконати: залишити поточний режим без змін, плавно підвищити частоту, знизити частоту, обмежити подальше зростання продуктивності або перейти до захисного режиму. У межах цього підходу керування частотою не виконується хаотично. Кожна дія має бути пов'язана з конкретним станом системи та підтверджена кількома послідовними вимірюваннями.

Першим етапом роботи алгоритму стало отримання телеметричних даних і перевірка їх коректності. Якщо дані неповні або мають помилковий вигляд, алгоритм не виконує різких змін частоти. У такому випадку безпечнішим є утримання поточного режиму до моменту, коли система отримає достовірні показники. Це рішення має практичне значення, оскільки випадкова помилка зчитування температури або частоти не повинна призводити до необґрунтованого зниження чи підвищення продуктивності. Такий підхід робить алгоритм стійкішим до одиничних збоїв і забезпечує більш передбачувану поведінку засобу.

Після перевірки даних алгоритм переходить до оцінювання визначеного режиму навантаження. Якщо система перебуває у спокійному режимі, процесор не потребує високої частоти, оскільки активних обчислювальних задач немає або їхня інтенсивність незначна. У такому стані алгоритм переводить CPU у нижчий або економний діапазон частот, якщо температура і навантаження залишаються стабільними. Це дозволяє зменшити фонове енергоспоживання та уникнути зайвого нагріву під час простою або виконання неважких службових процесів.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

У режимі помірною навантаження алгоритм поводить себе обережніше. Якщо навантаження процесора зростає, але температура залишається в безпечному діапазоні, частота може бути залишена на середньому рівні або плавно підвищена. У цьому випадку метою не є максимальне зниження енергоспоживання будь-якою ціною. Важливо зберегти достатній рівень продуктивності, щоб прикладні задачі виконувалися без помітних затримок. Тому алгоритм не обмежує частоту відразу після появи навантаження, а оцінює, чи має система тепловий запас для підтримання продуктивнішого режиму.

У режимі інтенсивного навантаження алгоритм враховує два протилежні завдання. З одного боку, одноплатний комп'ютер має виконувати задачі з достатньою швидкістю. З іншого боку, тривала робота на високій частоті може швидко підняти температуру процесора. Через це алгоритм підтримує підвищену частоту лише тоді, коли температура не наближається до небажаної зони. Якщо температура зростає повільно і залишається в прийнятному діапазоні, система не знижує продуктивність передчасно. Якщо ж за кілька циклів вимірювання видно стійкий нагрів, алгоритм поступово обмежує частоту, щоб уникнути різкого переходу до тротлінгу.

Температурно напружений режим став для алгоритму сигналом до обережного зниження продуктивності. У цьому стані процесор ще може працювати без аварійного обмеження, але запас до небажаної температури вже зменшився. Алгоритм не виконує різкого скидання частоти до мінімального рівня, оскільки це може суттєво погіршити виконання задач. Замість цього передбачено поступове зниження частоти на один або кілька доступних кроків. Після кожної такої зміни система продовжує збір телеметрії та перевіряє, чи стабілізувалася температура. Якщо нагрів сповільнився або температура почала знижуватися, подальше обмеження може не виконуватися.

У захисному режимі пріоритетом алгоритму стає стабілізація теплового стану. Якщо температура досягає небажаної зони або є ознаки наближення до тротлінгу, система знижує частоту до безпечнішого рівня та утримує її протягом певного часу. Такий режим не розглядається як штатний робочий стан, а

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

використовується як механізм запобігання нестабільній поведінці. Вихід із захисного режиму відбувається не одразу після першого зниження температури, а лише після стабільного повернення до безпечного діапазону. Це дозволяє уникнути повторного швидкого нагріву, коли процесор передчасно повертається до високої частоти.

Важливою особливістю розробленого алгоритму є плавність керування. Частота процесора змінюється не стрибком між мінімальним і максимальним значенням, а поступово, з урахуванням попереднього стану. Такий підхід потрібний для того, щоб не створювати різких коливань температури та енергоспоживання. Якщо система часто перемикається між крайніми режимами, вона поводить себе нестабільно: процесор то швидко нагрівається, то різко обмежується, а прикладні задачі отримують нерівномірний час виконання. Поступова зміна частоти дозволяє зробити реакцію засобу м'якшою та більш придатною для тривалої роботи.

Для зменшення зайвих перемикань у алгоритмі використано затримку між керуючими діями. Після зміни частоти система не приймає нове рішення миттєво, а очікує кілька циклів збору телеметрії. Це потрібно тому, що температура не реагує на зміну частоти одразу. Тепло, яке вже накопичилося в процесорі та радіаторі, ще певний час впливає на показники. Якщо оцінювати результат занадто швидко, алгоритм може помилково вирішити, що зниження частоти не допомогло, і надмірно обмежити продуктивність. Затримка між діями дозволяє дати системі час на стабілізацію.

Ще одним важливим елементом алгоритму стало врахування гістерезису. Перехід до нижчої частоти та повернення до продуктивнішого режиму виконуються за різними умовами. Якщо частоту знижено через нагрів, алгоритм не підвищує її після мінімального покращення температури. Для повернення до вищого режиму потрібно, щоб температура не просто зменшилася, а стабільно утримувалася в безпечному діапазоні. Це дозволяє уникнути частого перемикання між сусідніми частотами та робить поведінку процесора менш «нервовою» під час роботи біля температурної межі.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		47

Алгоритм також враховує характер задач, які виконуються на одноплатному комп'ютері. Для фонових процесів допустиме виконання в економнішому режимі, оскільки вони не потребують миттєвої реакції. Для активних прикладних задач потрібно зберігати достатню продуктивність, інакше енергоефективність буде досягнута за рахунок надмірного погіршення швидкодії. Через це керування частотою не зводиться до простого зниження продуктивності при кожному зростанні температури. Алгоритм має підтримувати баланс: зменшувати зайві енергетичні витрати там, де це не шкодить виконанню задач, і залишати ресурс процесора там, де він справді потрібний.

Окремо в алгоритмі передбачено журналювання прийнятих рішень. Кожна зміна режиму супроводжується записом у журнал, де фіксується час, поточна температура, завантаження CPU, фактична частота, визначений режим навантаження та обрана керуюча дія. Такий журнал потрібний не лише для перевірки працездатності засобу, а й для подальшого аналізу ефективності. За цими записами можна побачити, чи вчасно система реагує на нагрів, чи не занадто часто змінює частоту, чи справді після керуючої дії температура стабілізується. У третьому розділі такі дані можуть бути використані для побудови графіків і порівняння роботи стандартного та запропонованого режимів.

На рисунку 2.5 зображено блок-схему алгоритму енергоефективного планування задач і керування частотою процесора. У схемі показано, що робота алгоритму починається з отримання телеметричного запису та перевірки його коректності. Далі виконано аналіз режиму навантаження, температури, тренду нагріву та фактичної частоти CPU. На основі цих даних сформовано керуючу дію: утримання поточного режиму, плавне підвищення частоти, поступове зниження частоти або перехід до захисного режиму. Після застосування дії результат збережено в журналі, а система повертається до наступного циклу збору телеметрії. Така блок-схема показує замкнений характер керування, де

кожне нове рішення приймається з урахуванням фактичного стану платформи після попередньої дії.

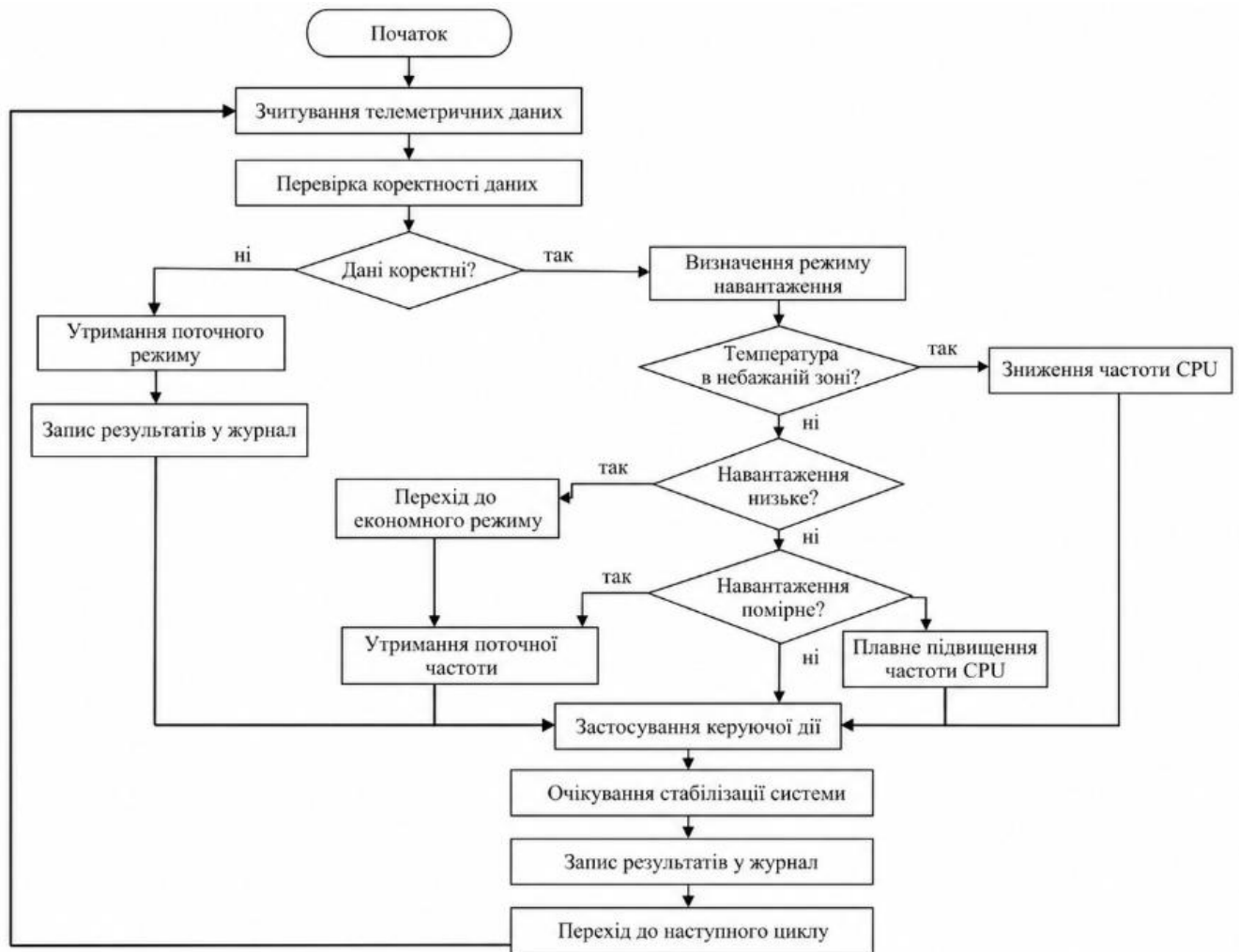


Рисунок 2.5 – Блок-схема алгоритму енергоефективного планування задач і керування частотою процесора

У практичному вигляді алгоритм працює циклічно. Після запуску засіб не завершує роботу після одного проходу, а постійно повторює послідовність «збір даних - аналіз - керуюча дія - перевірка результату». Завдяки цьому система може пристосовуватися до змін навантаження в часі. Якщо одноплатний комп'ютер переходить із простою до активної роботи, алгоритм поступово підвищує продуктивність. Якщо тривале навантаження призводить до нагріву, частота обмежується. Якщо після завершення задач температура знижується, система повертається до економнішого або збалансованого режиму.

Розроблений алгоритм не замінює штатні механізми операційної системи, а працює поверх них. Це важливо з практичної точки зору, оскільки повне втручання в механізми ядра потребувало б складнішої реалізації та могло б знизити стабільність платформи. Запропонований засіб використовує доступні системні інтерфейси для читання параметрів і встановлення режимів CPU. Завдяки цьому алгоритм можна реалізувати у вигляді окремого службового програмного модуля, який запускається разом із системою або вручну під час експериментальної перевірки.

Під час розроблення алгоритму враховано, що сам засіб керування не повинен створювати значне додаткове навантаження. Якщо модуль моніторингу й аналізу споживатиме багато ресурсів, він сам погіршить енергоефективність системи. Через це логіку побудовано на простих перевірках, порогових правилах, аналізі кількох останніх вимірювань і поступовому виборі режиму. Такий підхід не потребує складних обчислень, але дає змогу враховувати головні фактори: завантаження, температуру, тренд нагріву та фактичну частоту процесора.

Сформований алгоритм також забезпечує зрозумілу основу для майбутньої програмної реалізації. Його можна поділити на кілька програмних функцій: зчитування телеметрії, перевірку даних, визначення режиму навантаження, вибір керуючої дії, застосування частоти та запис результату. Такий поділ спрощує реалізацію в третьому розділі, оскільки кожен етап має чітке призначення. При цьому загальна логіка залишається простою: система намагається підтримувати продуктивність там, де це потрібно, і зменшувати частоту там, де це дозволяє знизити нагрів та енергоспоживання.

У другому розділі кваліфікаційної роботи сформовано проектну основу програмно-апаратного засобу енергоефективного планування задач на одноплатному комп'ютері. Розділ охоплює загальну архітектуру засобу, вибір апаратних і програмних компонентів, організацію збору телеметрії, логіку аналізу стану системи та алгоритм керування частотою процесора,

У процесі розроблення архітектури засобу визначено основні функціональні частини системи: одноплатний комп'ютер, Linux-подібну

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

операційну систему, модуль збору телеметричних даних, блок аналізу стану системи, регулятор режимів CPU та модуль журналювання результатів. Таку структуру сформовано за принципом замкненого контуру керування, де система не лише зчитує показники, а й використовує їх для прийняття керуючих рішень. Це дозволяє розглядати запропонований засіб не як звичайний інструмент моніторингу, а як програмно-апаратне рішення для підтримання стабільного режиму роботи процесора.

Обґрунтовано вибір апаратних і програмних компонентів системи. Як апаратну основу використано одноплатний комп'ютер, джерело живлення, систему охолодження та за потреби додаткові засоби вимірювання енергоспоживання. Програмну частину побудовано на основі Linux-подібної операційної системи, системних інтерфейсів доступу до параметрів CPU та засобів програмної обробки телеметрії. Такий вибір забезпечує практичну реалізацію засобу без надмірного ускладнення апаратної частини та дає можливість працювати поверх штатних механізмів операційної системи.

У розділі також організовано збір телеметричних даних з одноплатного комп'ютера. Основними параметрами визначено температуру процесора, поточну частоту CPU, рівень завантаження, часову мітку та службовий стан вимірювання. Зібрані дані використовуються не тільки для подальшого відображення, а й як інформаційна основа для визначення режиму навантаження. Це створює зв'язок між фактичним станом платформи та подальшими керуючими діями.

Розроблено логіку аналізу стану системи, у межах якої враховано не лише поточне завантаження CPU, а й температуру процесора, її динаміку, фактичну частоту та стабільність показників. На основі цих параметрів визначено режими роботи системи: спокійний, помірний, інтенсивний, температурно напружений і захисний. Такий підхід дозволяє уникати різких реакцій на короткі піки навантаження та забезпечує більш точне визначення реального стану одноплатного комп'ютера.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

### 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ПЕРЕВІРКА РОБОТИ ПРОГРАМНО-АПАРАТНОГО ЗАСОБУ ЕНЕРГОЕФЕКТИВНОГО ПЛАНУВАННЯ ЗАДАЧ

3.1 Формування експериментального середовища на базі одноплатного комп'ютера

Практичну реалізацію програмно-апаратного засобу енергоефективного планування задач розпочато з формування експериментального середовища на базі одноплатного комп'ютера. Цей етап став основою для подальшого запуску програмних модулів, збору телеметричних даних, перевірки алгоритму керування частотою процесора та оцінювання поведінки системи під різними режимами навантаження. Експериментальне середовище сформовано так, щоб воно відображало реальні умови використання компактної обчислювальної платформи, у якій одночасно важливими залишаються продуктивність, температура, стабільність живлення та можливість тривалої роботи без перегріву.

Система охолодження також стала важливою частиною експериментального середовища. У роботі не поставлено задачу повністю усунути нагрів процесора за рахунок потужного охолодження, оскільки тоді вплив алгоритму керування частотою був би менш помітним. Натомість використано такий підхід, за якого охолодження забезпечує базову стабільність платформи, але не приховує теплову реакцію процесора на зміну навантаження. Це дозволяє простежити, як температура змінюється під час роботи системи, як вона реагує на підвищення або зниження частоти CPU та чи здатний запропонований засіб підтримувати більш рівномірний тепловий режим.

Програмне середовище сформовано на базі Linux-подібної операційної системи, оскільки вона надає доступ до системних показників процесора та механізмів керування його частотою. У межах експериментального стенда операційна система виконує роль проміжного рівня між апаратною платформою та розробленим програмним засобом. Через системні інтерфейси зчитуються

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

температура CPU, поточна частота, завантаження процесора та інші службові параметри. Через ці ж інтерфейси може застосовуватися керуюча дія, пов'язана зі зміною режиму роботи процесора. Такий підхід дозволяє не втручатися безпосередньо в апаратний рівень, а використовувати штатні можливості ОС.

Для реалізації програмної частини експериментального середовища підготовлено інтерпретатор Python, службові бібліотеки для роботи з файлами та часовими мітками, а також засоби для запису результатів у журнал. Python обрано як зручний інструмент для практичної реалізації модулів збору, перевірки, аналізу та збереження телеметричних даних. У межах цього середовища програмні модулі можуть виконуватися як окремі скрипти або як єдиний службовий застосунок, що працює циклічно. Завдяки цьому структура програмної реалізації залишається зрозумілою, а кожен етап роботи засобу можна окремо перевірити під час запуску.

Під час підготовки середовища враховано потребу в доступі до системних параметрів процесора. Для цього перевірено можливість зчитування температури, частоти та завантаження CPU, а також доступність механізмів керування частотою. Така перевірка потрібна для того, щоб програмний засіб не працював у відриві від реального стану платформи. Якщо певний системний показник недоступний, алгоритм не може коректно оцінити стан одноплатного комп'ютера. Тому перед запуском основної логіки передбачено перевірку доступності потрібних файлів, команд і прав виконання.

У практичній установці також передбачено створення тестового навантаження. Воно потрібне для перевірки роботи засобу в різних умовах: у режимі простою, при помірній активності та під час тривалого інтенсивного навантаження. Без такого навантаження неможливо повноцінно оцінити, як система реагує на зміну температури та завантаження CPU. Тестове навантаження може формуватися за допомогою простих обчислювальних задач, циклічної обробки даних або спеціальних утиліт, які створюють контрольоване використання процесора. У межах кваліфікаційної роботи важливо не саме

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

джерело навантаження, а можливість повторити однакові умови для порівняння різних режимів роботи.

Експериментальне середовище сформовано з урахуванням необхідності подальшого журналювання. Кожен запуск програмного засобу має супроводжуватися фіксацією результатів, оскільки саме журнал телеметрії дозволяє оцінити поведінку системи не тільки в окремий момент, а протягом усього періоду роботи. До журналу мають потрапляти час вимірювання, температура процесора, поточна частота CPU, рівень завантаження, визначений режим навантаження та прийнята керуюча дія. Такі записи створюють основу для побудови графіків і порівняння стандартного режиму роботи з режимом енергоефективного керування.

На рисунку 3.1 зображено структуру експериментального середовища програмно-апаратного засобу енергоефективного планування задач, у якій показано взаємозв'язок між одноплатним комп'ютером, джерелом живлення, системою охолодження, операційною системою, модулем збору телеметрії, модулем керування частотою CPU та журналом результатів.

Подана на рисунку 3.1 структура відображає загальний принцип організації практичного стенда. Одноплатний комп'ютер виступає центральним вузлом, на якому поєднано апаратні та програмні складові. З боку апаратної частини до нього підключено джерело живлення та систему охолодження, які забезпечують стабільну роботу під час виконання навантаження.

З боку програмної частини на платформі функціонує операційна система, через яку організовано доступ до телеметричних показників і механізмів керування частотою процесора.

У результаті сформовано замкнене середовище, у якому можна не лише спостерігати за станом системи, а й впливати на нього за допомогою програмного алгоритму.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

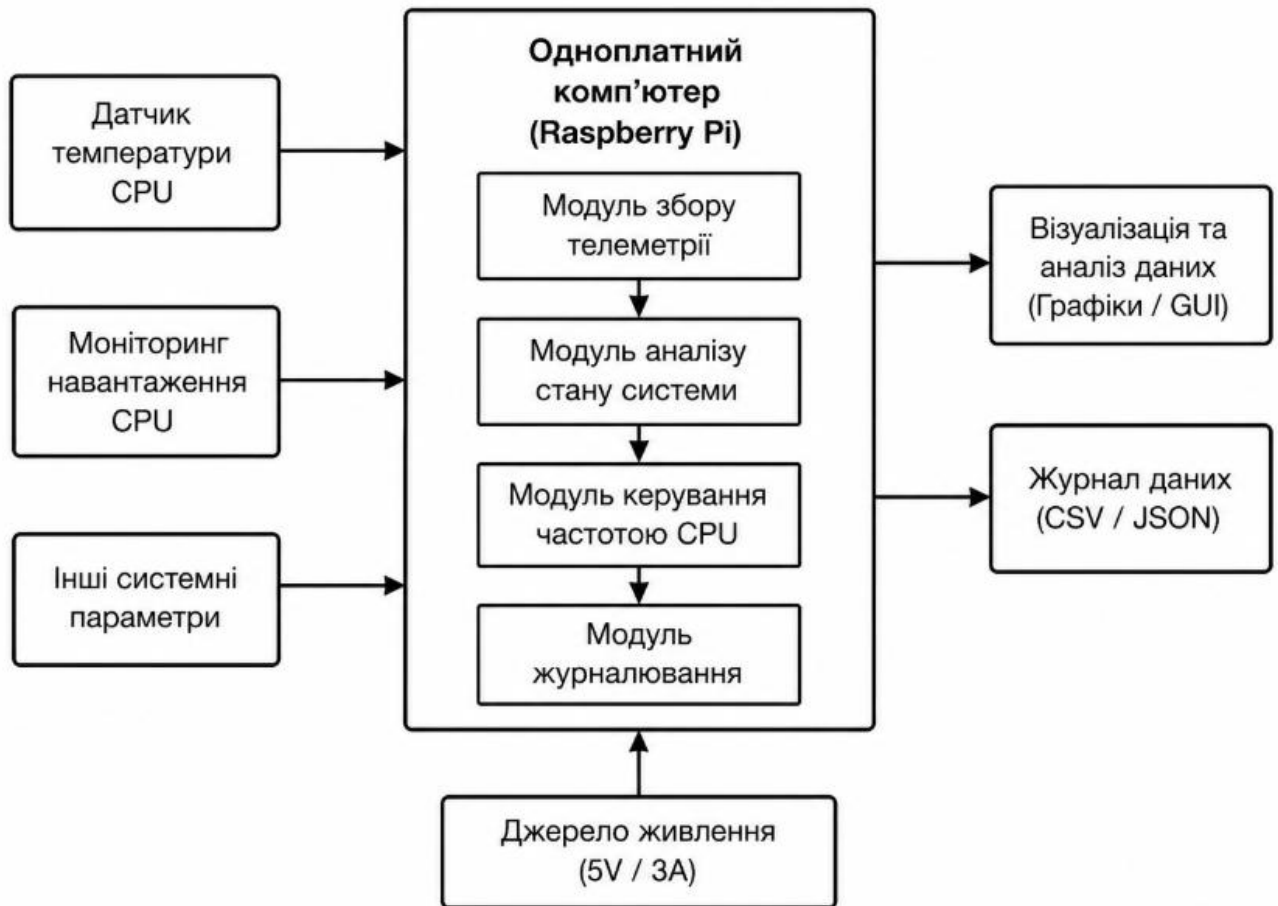


Рисунок 3.1 – Структура експериментального середовища програмно-апаратного засобу енергоефективного планування задач

Важливою особливістю експериментального середовища є те, що воно не потребує складної зовнішньої інфраструктури. Усі основні процеси виконуються безпосередньо на одноплатному комп'ютері. Це спрощує перевірку засобу, оскільки зменшує кількість зовнішніх факторів, які могли б впливати на результати. Водночас така побудова добре відповідає реальним сценаріям використання одноплатних платформ, де пристрій часто працює автономно, самостійно виконує прикладні задачі та самостійно збирає дані про власний стан.

Сформоване середовище також дозволяє відокремити роботу запропонованого засобу від стандартної поведінки операційної системи. Для цього передбачено можливість запуску одних і тих самих сценаріїв навантаження в різних режимах. Спочатку платформа може працювати зі штатними налаштуваннями керування продуктивністю, після чого запускається

запропонований алгоритм енергоефективного планування. Порівняння журналів у цих режимах дає змогу оцінити, як змінилася температура, частота процесора, стабільність роботи та загальна реакція системи на тривале навантаження.

Під час формування експериментального середовища враховано також вплив самого програмного засобу на роботу платформи. Модуль збору телеметрії та аналізу не повинен створювати надмірного додаткового навантаження, інакше він спотворюватиме результати власної перевірки. Через це програмна логіка має виконуватися з помірною періодичністю, а операції запису до журналу мають бути організовані без зайвого навантаження на накопичувач. Такий підхід дозволяє зберегти реалістичність експерименту та не перетворювати засіб моніторингу на окреме джерело підвищеного енергоспоживання.

Сформоване експериментальне середовище стало основою для подальшої реалізації програмних модулів. На його базі можна послідовно перевірити роботу збору телеметрії, журналювання, визначення режиму навантаження та застосування керуючих дій. Завдяки цьому третій розділ переходить від опису загальної структури до конкретної реалізації програмної частини. У наступному підрозділі розкривається робота модуля збору та журналювання телеметричних даних, який забезпечує надходження фактичних показників для подальшого аналізу й керування частотою процесора.

### 3.2 Реалізація програмного модуля збору та журналювання телеметричних даних

Після формування експериментального середовища на базі одноплатного комп'ютера наступним практичним етапом стала реалізація програмного модуля збору та журналювання телеметричних даних. Саме цей модуль забезпечує надходження фактичної інформації про стан системи, тому без нього неможливо виконати аналіз навантаження, визначити тепловий режим платформи та прийняти обґрунтоване рішення щодо зміни частоти процесора. У межах

кваліфікаційної роботи цей модуль став нижнім програмним рівнем усього засобу енергоефективного планування задач, оскільки він безпосередньо взаємодіє з операційною системою та формує вхідні дані для подальшого керування.

Програмний модуль реалізовано як окремий службовий компонент, який працює циклічно та з певним інтервалом зчитує основні параметри одноплатного комп'ютера. До таких параметрів віднесено температуру CPU, поточну частоту процесора, рівень завантаження, часову мітку вимірювання та службову інформацію про стан виконання циклу. Такий набір показників обрано тому, що він безпосередньо відображає зв'язок між продуктивністю, тепловим режимом і фактичним навантаженням платформи. Температура показує накопичений тепловий ефект, частота CPU відображає поточний режим продуктивності, а завантаження процесора дає змогу зрозуміти, наскільки активно система виконує обчислювальні задачі.

У межах реалізації програмний модуль не обмежується простим одноразовим зчитуванням значень. Його роботу побудовано як послідовний процес, у якому кожен цикл складається з отримання даних, перевірки їх коректності, формування телеметричного запису та збереження результату в журналі. Така організація потрібна для того, щоб кожен запис мав однакову структуру та міг бути використаний у подальшому аналізі. Якщо дані зберігати нерівномірно або без єдиного формату, порівняння режимів роботи стане неточним, а побудова графіків температури, частоти та завантаження ускладниться.

Після ініціалізації запускається основний цикл збору телеметрії. У кожному циклі послідовно зчитуються значення температури, частоти та завантаження CPU. Після цього до них додається часова мітка, яка фіксує момент отримання даних. Часова мітка є обов'язковою, оскільки без неї неможливо побудувати часову залежність температури або простежити, як змінювалася частота процесора під час навантаження. Саме прив'язка до часу перетворює

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

окремі вимірювання на корисний журнал, придатний для аналізу поведінки системи протягом усього експерименту.

Журналювання реалізовано як послідовний запис телеметрії у файл. Для базової реалізації зручним є формат CSV, оскільки він простий, відкривається у табличних редакторах і легко обробляється програмно. Кожен рядок такого журналу відповідає одному циклу вимірювання. У результаті формується послідовність записів, за якою можна простежити поведінку одноплатного комп'ютера протягом усього часу роботи програмного засобу. За потреби журнал може бути перенесений у SQLite або інше локальне сховище, однак для перевірки працездатності алгоритму CSV-формату достатньо.

Окрему увагу під час реалізації приділено тому, щоб сам модуль збору телеметрії не створював надмірного навантаження на систему. Якщо програма моніторингу виконується занадто часто або постійно звертається до накопичувача, вона сама починає впливати на температуру та завантаження CPU. Через це інтервал збору даних підбрано так, щоб він забезпечував достатню деталізацію, але не створював зайвого фону.

На рисунку 3.2 показано структуру роботи програмного модуля збору та журналювання телеметрії, у якій відображено послідовність зчитування системних показників, перевірки коректності даних, формування телеметричного запису та збереження результатів у журналі.

На рисунку 3.2 зображено логіку роботи програмного модуля, який є основою інформаційного забезпечення всього засобу енергоефективного планування задач. У схемі показано, що дані надходять від системних інтерфейсів одноплатного комп'ютера, після чого проходять перевірку, об'єднуються в один телеметричний запис і зберігаються у журналі. Така побудова дозволяє відокремити етап первинного збору даних від подальшого аналізу та керування. Це робить програмну реалізацію зрозумілішою, а результати вимірювань - придатними для перевірки.



Рисунок 3.2 – Структура роботи програмного модуля збору та журналювання телеметрії

Після збереження запису в журналі програмний модуль переходить до очікування наступного циклу. У цей період система не виконує зайвих дій, що дозволяє зменшити фонове навантаження. Після завершення інтервалу очікування цикл повторюється: знову зчитуються показники, перевіряються значення, формується новий запис і доповнюється журнал. Такий принцип роботи створює безперервний потік телеметрії, який потрібен для замкненого керування частотою процесора.

Практична реалізація модуля збору та журналювання також передбачає можливість подальшого використання отриманих даних у графічному вигляді. Оскільки кожен запис має часову мітку, температуру, частоту та завантаження, на основі журналу можна побудувати графіки зміни параметрів у часі. Такі графіки дозволяють побачити, як система поводить себе в режимі простою, під помірним навантаженням і під час інтенсивного обчислення. Вони також дають змогу порівняти штатний режим роботи ОС із запропонованим режимом енергоефективного планування.

### 3.3 Реалізація алгоритму аналізу стану системи та керування частотою CPU

Після реалізації програмного модуля збору та журналювання телеметричних даних наступним етапом практичної частини стала реалізація алгоритму аналізу стану системи та керування частотою CPU. Цей алгоритм є основною керуючою частиною програмно-апаратного засобу, оскільки саме він перетворює зібрані телеметричні показники на конкретну дію щодо режиму роботи процесора. Якщо модуль збору даних відповідає за отримання фактичних значень температури, частоти та завантаження, то алгоритм аналізу вже визначає, у якому стані перебуває одноплатний комп'ютер і чи потрібно змінювати режим роботи CPU.

Реалізацію алгоритму побудовано за циклічним принципом. Після кожного зчитування телеметрії система отримує набір актуальних показників, перевіряє їх коректність, визначає поточний режим навантаження та формує керуючу дію. У межах такого підходу алгоритм не працює як одноразове налаштування, а постійно супроводжує роботу одноплатного комп'ютера. Це важливо для задач енергоефективного планування, оскільки навантаження на процесор змінюється нерівномірно: періоди простою можуть чергуватися з короткими піками активності або тривалими обчислювальними процесами.

Основою алгоритму стали телеметричні дані, які надходять із попереднього програмного модуля. До них належать температура CPU, поточна частота процесора, рівень завантаження, часова мітка вимірювання та службова ознака коректності запису. Ці дані використано не окремо, а в поєднанні. Наприклад, високе завантаження процесора саме по собі ще не означає, що частоту потрібно знижувати. Якщо температура залишається в безпечному діапазоні, система може певний час підтримувати продуктивний режим. Водночас навіть помірне навантаження може потребувати обмеження частоти, якщо температура вже наблизилася до небажаної зони.

На першому етапі алгоритм перевіряє придатність отриманого телеметричного запису. Якщо значення температури, частоти або завантаження

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

відсутні, мають некоректний формат або виходять за очевидно неможливі межі, система не виконує різких керуючих дій. У такому випадку реалізовано утримання поточного режиму з фіксацією відповідного стану в журналі. Це дозволяє запобігти ситуації, коли випадкова помилка зчитування спричиняє необґрунтоване підвищення чи зниження частоти CPU. Така перевірка робить алгоритм стійкішим до короточасних збоїв і підвищує надійність роботи засобу.

Наступним етапом є аналіз завантаження CPU. У реалізованій логіці короткі випадкові піки не прирівнюються до повноцінного інтенсивного навантаження. Для цього використано оцінювання показників за кілька послідовних циклів. Якщо завантаження зросло лише на короткий момент, система не змінює режим різко. Якщо підвищене навантаження зберігається стабільно, алгоритм визначає його як значущий робочий стан. Це дозволяє зменшити кількість зайвих перемикань частоти, які могли б виникати через фонові процеси операційної системи.

Захисний режим застосовується тоді, коли температура перебуває в небажаній зоні або система швидко наближається до неї. У цьому стані пріоритетом стає стабілізація температури, а не максимальна швидкодія. Частота CPU знижується до безпечнішого рівня, а нове підвищення продуктивності блокується до моменту стабільного охолодження. Вихід із захисного режиму реалізовано з урахуванням гістерезису, тобто повернення до вищої частоти виконується не після першого незначного зниження температури, а після її стабільного утримання в безпечному діапазоні.

Гістерезис перемикання режимів став важливою частиною практичної реалізації алгоритму. Без нього система могла б занадто часто змінювати частоту на межі двох режимів. Наприклад, якщо температура коливається біля заданої межі, процесор міг би постійно переходити з продуктивного режиму до обмеженого й назад. Це створювало б нестабільну поведінку та зайві коливання енергоспоживання. У реалізованій логіці перехід до обмеження і повернення до продуктивнішого режиму мають різні умови, тому система працює плавніше.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

Окремо реалізовано очікування стабілізації після застосування керуючої дії. Після зміни частоти процесора алгоритм не приймає нове рішення миттєво. Він очікує наступних циклів телеметрії, щоб оцінити реальну реакцію системи. Це потрібно тому, що температура не змінюється одразу після зниження або підвищення частоти. Тепловий стан залежить від накопиченого нагріву процесора, радіатора та навколишніх умов. Очікування стабілізації дозволяє уникнути надмірного обмеження продуктивності та робить керування більш точним.

У практичній реалізації алгоритм поділено на кілька функціональних частин. Одна частина відповідає за отримання підготовленої телеметрії, друга - за аналіз температури та навантаження, третя - за визначення режиму системи, четверта - за вибір керуючої дії, а п'ята - за застосування зміни частоти та запис результатів у журнал. Такий поділ зробив програмну структуру зрозумілішою і дозволив окремо перевіряти кожен етап. Якщо під час тестування виникає помилка, її легше знайти саме в тому функціональному блоці, який відповідає за відповідну операцію.

Керування частотою CPU реалізовано через системні механізми операційної системи. Алгоритм не взаємодіє з апаратною частиною напряму, а використовує доступні інтерфейси ОС для зміни режиму процесора. Це знижує ризик некоректного втручання в роботу платформи та дає змогу зберегти сумісність зі штатними механізмами керування. У такій реалізації програмний засіб не замінює повністю стандартне керування, а доповнює його власною логікою, яка враховує температуру, навантаження та фактичний стан платформи.

Під час вибору керуючої дії алгоритм не застосовує різкі переходи між крайніми частотами. Якщо потрібно підвищити продуктивність, частота збільшується поступово. Якщо потрібно зменшити нагрів, частота також знижується поетапно. Така реалізація забезпечує плавну зміну режимів і не створює різких провалів швидкодії. Для одноплатного комп'ютера це особливо важливо, оскільки різка зміна частоти може дати короткочасний ефект, але не забезпечити стабільної поведінки під тривалим навантаженням.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62



й визначений режим системи та прийнята керуюча дія. Завдяки цьому під час аналізу результатів можна пояснити, чому в певний момент частоту підвищено, знижено або залишено без змін. Це робить перевірку роботи засобу більш прозорою, оскільки поведінка системи підтверджується не лише графіками температури чи частоти, а й записами про прийняті рішення.

Практична цінність реалізованого алгоритму полягає в тому, що він дозволяє керувати процесором не за фіксованим сценарієм, а відповідно до реального стану одноплатного комп'ютера. Якщо система працює в режимі простою, частота не утримується на завищеному рівні. Якщо з'являється корисне навантаження і температура залишається безпечною, процесор отримує можливість працювати продуктивніше. Якщо температура наближається до небажаної зони, алгоритм завчасно обмежує частоту та знижує ризик тротлінгу. У результаті засіб підтримує баланс між продуктивністю, температурою та енергоспоживанням.

### 3.4 Перевірка роботи засобу в різних режимах навантаження

Після реалізації програмного модуля збору телеметрії та алгоритму аналізу стану системи виконано перевірку роботи програмно-апаратного засобу в різних режимах навантаження. Цей етап став необхідним для підтвердження того, що розроблена система не лише зчитує показники одноплатного комп'ютера, а й коректно реагує на зміну обчислювального навантаження, температури та фактичної частоти процесора. У межах перевірки основну увагу зосереджено на тому, як засіб поводить себе під час простою, помірної активності, інтенсивного навантаження та наближення до температурно напруженого режиму.

Перевірку організовано у вигляді послідовного запуску кількох сценаріїв роботи одноплатного комп'ютера. Такий підхід дав змогу простежити не один окремий стан системи, а повну зміну режимів від мінімального навантаження до більш складних умов. Для кожного сценарію зафіксовано температуру CPU, поточну частоту процесора, рівень завантаження, визначений режим системи та

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		64

керуючу дію алгоритму. Завдяки цьому поведінку засобу оцінено не за суб'єктивним враженням, а на основі телеметричних записів, отриманих під час роботи.

На першому етапі перевірено роботу засобу в режимі простою. У цьому режимі одноплатний комп'ютер не виконував інтенсивних прикладних задач, а основне навантаження створювали лише фонові процеси операційної системи та сам модуль телеметрії. Такий сценарій потрібний для перевірки того, чи здатний алгоритм розпізнати відсутність активного навантаження та не утримувати процесор на завищеній частоті без потреби. У процесі перевірки система визначила спокійний режим роботи, а керуюча логіка залишила CPU в економному або близькому до економного стані.

У режимі простою температура процесора змінювалася повільно та не мала різких коливань. Це підтвердило, що модуль збору телеметрії не створює надмірного фонового навантаження і не спотворює результати власної роботи. Частота процесора в цьому сценарії не потребувала підвищення, оскільки завантаження CPU залишалось низьким. Така поведінка відповідає логіці енергоефективного планування: якщо система не виконує важких задач, зайва продуктивність не використовується, а отже, її підтримання лише збільшує енергоспоживання та теплове навантаження.

Другим етапом стала перевірка в режимі помірного навантаження. Для цього на одноплатному комп'ютері запущено задачі, які створювали стабільну, але не максимальну активність процесора. Такий сценарій наближений до реальної роботи системи, коли платформа одночасно виконує службові процеси, обробляє дані, записує журнали або підтримує роботу невеликого прикладного сервісу. У цьому режимі алгоритм мав визначити, що навантаження вже не є фоновим, але температура ще не створює загрози перегріву.

Окремо перевірено температурно напружений режим. Цей сценарій відображає ситуацію, коли температура ще не обов'язково досягла критичного значення, але вже наблизилася до небажаної зони або демонструє стійке зростання. Саме в такому стані стандартні політики операційної системи можуть

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

запізно реагувати на перегрів, оскільки вони часто орієнтуються переважно на поточне навантаження або вже встановлені температурні межі. Запропонований засіб у цьому випадку використав аналіз тренду температури й завчасно сформував обмежувальну дію.

У температурно напруженому режимі алгоритм надав пріоритет стабілізації теплового стану. Частота процесора знижувалася до безпечнішого рівня, а нове підвищення продуктивності відкладалося до моменту стабільного охолодження. У цьому сценарії важливу роль відіграв гістерезис перемикання режимів. Завдяки йому система не поверталася до вищої частоти одразу після незначного зниження температури. Повернення до продуктивнішого режиму виконувалося лише після того, як температура певний час утримувалася в безпечному діапазоні. Це зменшило кількість зайвих перемикань і зробило поведінку засобу більш стабільною.

Під час перевірки особливу увагу приділено відповідності реакції алгоритму фактичному стану системи. У спокійному режимі засіб не повинен безпідставно підвищувати частоту процесора. У помірному режимі він має підтримувати достатню швидкодію без зайвого перегріву. В інтенсивному режимі система повинна дозволити виконання ресурсомістких задач, але при цьому стежити за температурою. У температурно напруженому стані головним завданням стає завчасне обмеження частоти до моменту спрацювання апаратного захисту. Саме така послідовність реакцій підтверджує правильність побудованої логіки керування.

Результати журналювання показали, що засіб формує зрозумілу послідовність рішень. У записах фіксується не тільки температура, частота та завантаження CPU, а й визначений режим навантаження та керуюча дія. Це дало змогу простежити, чому в певний момент частоту залишено без змін, знижено або підвищено. Такий формат перевірки є важливим, оскільки він дозволяє пояснити поведінку системи на основі фактичних даних, а не лише за зовнішніми ознаками роботи пристрою.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

У режимі простою очікуваним результатом стало утримання економного режиму. Якщо процесор не виконує інтенсивних задач, високий рівень частоти не має практичного сенсу. Зафіксовані значення показали, що алгоритм не створює зайвих перемикачів і не намагається підвищувати продуктивність без стабільного навантаження. Це підтвердило правильність умови, за якою короткі фонові процеси не повинні переводити систему в продуктивний режим.

У режимі помірною навантаження перевірено здатність засобу підтримувати баланс між швидкістю та енергоспоживанням. Система не обмежувала частоту надмірно, оскільки температура залишалася прийнятною. Водночас алгоритм не допускав безпідставного переходу до максимального режиму. Така поведінка вважається коректною, оскільки помірні задачі потребують стабільної роботи, але не завжди потребують пікової продуктивності процесора.

Під час інтенсивного навантаження найбільш важливим стало спостереження за температурним трендом. На початку такого сценарію система підтримувала підвищений рівень продуктивності, оскільки процесор мав тепловий запас. Після появи стійкого нагріву алгоритм поступово знижував частоту або обмежував її подальше зростання. Це підтвердило, що засіб не реагує надто рано, але й не чекає моменту, коли платформа вже переходить до тротлінгу. У результаті робота процесора стала більш рівномірною.

У температурно напруженому режимі перевірено захисну поведінку алгоритму. Засіб коректно визначив, що подальше утримання високої частоти може призвести до небажаного перегріву, і перейшов до обмеження продуктивності. Важливо, що після зниження температури система не поверталася одразу до попереднього режиму. Це підтвердило роботу гістерезису та стабілізаційної затримки, які зменшують частоту перемикачів і допомагають уникнути повторного швидкого нагріву.

Практична перевірка також показала значення правильного вибору інтервалу збору телеметрії. Якщо інтервал надто короткий, журнал швидко заповнюється великою кількістю записів, а сам модуль моніторингу створює

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

додаткове навантаження. Якщо інтервал надто довгий, частина коротких змін може залишитися непоміченою. У межах перевірки використано помірний інтервал, який дав змогу бачити загальну динаміку нагріву та зміни частоти без надмірного впливу на роботу системи. Це забезпечило достатню деталізацію для подальшої оцінки результатів.

### 3.5 Аналіз отриманих результатів та оцінювання ефективності запропонованого підходу

Після перевірки роботи програмно-апаратного засобу в різних режимах навантаження виконано аналіз отриманих результатів і оцінювання ефективності запропонованого підходу. Цей етап став завершальною частиною практичної реалізації, оскільки саме на основі зібраних телеметричних даних можна оцінити, наскільки розроблений засіб впливає на поведінку одноплатного комп'ютера. Основну увагу приділено зміні температури CPU, фактичній частоті процесора, рівня завантаження та стабільності переходів між режимами роботи.

Оцінювання результатів виконано шляхом порівняння двох режимів роботи одноплатного комп'ютера. Першим став стандартний режим, у якому керування частотою процесора здійснюється штатними механізмами операційної системи. У цьому випадку система здебільшого реагує на поточне завантаження CPU та змінює частоту відповідно до універсальної політики продуктивності. Другим став режим із використанням запропонованого алгоритму енергоефективного планування задач, у якому додатково враховано температуру процесора, тренд її зміни, стабільність навантаження та потребу в плавному перемиканні частоти.

У стандартному режимі роботи операційна система швидко підвищує частоту процесора при появі навантаження. Така поведінка дозволяє забезпечити високу продуктивність на початку виконання задач, однак під час тривалої роботи вона поступово призводить до зростання температури. Для одноплатного

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

комп'ютера це має суттєве значення, оскільки компактна плата має обмежену площу відведення тепла, а система охолодження не завжди здатна швидко компенсувати підвищене тепловиділення. Унаслідок цього в стандартному режимі спостерігається швидше наближення до температурно напруженої зони.

У запропонованому режимі енергоефективного керування частота процесора змінюється більш плавно. Алгоритм не переводить CPU у максимально продуктивний режим лише через короткий сплеск навантаження, а перевіряє, чи зберігається активність протягом кількох циклів збору телеметрії. Завдяки цьому короточасні фонові процеси не викликають зайвих перемикачів. Якщо навантаження стає стійким, система підвищує частоту, але водночас контролює температуру та її напрям зміни. Такий підхід дозволяє зменшити кількість різких переходів між режимами й підтримати більш рівномірну роботу процесора.

Аналіз журналу телеметрії показав, що найбільш інформативними для оцінювання стали три показники: температура CPU, фактична частота процесора та визначений режим навантаження. Температура дала змогу оцінити теплову стабільність системи. Частота показала, як саме алгоритм впливає на продуктивність. Режим навантаження пояснив причину прийняття конкретної керуючої дії. Завдяки одночасному збереженню цих параметрів у журналі стало можливим не лише побачити результат, а й відтворити логіку роботи засобу в кожен момент часу.

У режимі простою запропонований підхід показав здатність утримувати процесор у нижчому діапазоні частот без погіршення роботи системи. Оскільки активне навантаження відсутнє, висока частота не дає практичної переваги, але збільшує фонове енергоспоживання. У цьому сценарії алгоритм коректно визначив спокійний режим і не виконував зайвого підвищення продуктивності. Це підтвердило, що засіб не реагує на короткі службові процеси як на повноцінне навантаження.

У режимі помірного навантаження ефективність підходу проявилася в підтриманні балансу між продуктивністю і температурою. Процесор не

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

обмежувався надмірно, тому виконання задач залишалось стабільним. Водночас частота не піднімалася до максимальної без реальної потреби. Такий режим є найбільш наближеним до щоденної роботи одноплатного комп'ютера, коли пристрій виконує кілька службових або прикладних задач, але не перебуває під постійним піковим навантаженням. Отримані результати показали, що запропонований алгоритм у такому сценарії працює найбільш рівномірно.

Під час інтенсивного навантаження головним критерієм стала здатність системи підтримувати продуктивність без неконтрольованого перегріву. У стандартному режимі процесор швидко виходить на високі частоти, що забезпечує добру швидкодію на початку виконання задачі, але поступово збільшує ризик температурного обмеження.

Оцінювання ефективності запропонованого підходу виконано за кількома практичними ознаками. Першою ознакою стала стабільність температури. Якщо температура не має різких стрибків і довше залишається в безпечному діапазоні, система працює передбачуваніше. Другою ознакою стала стабільність частоти CPU. Якщо частота не перемикається хаотично між мінімальним і максимальним значенням, прикладні задачі отримують рівномірніший режим виконання. Третьою ознакою стала логічність керуючих дій. Якщо кожне зниження або підвищення частоти можна пояснити даними журналу, алгоритм вважається прозорим і керованим.

За результатами перевірки встановлено, що запропонований алгоритм найкраще проявляє себе у тривалих сценаріях роботи. Саме в таких умовах стандартне швидке підвищення частоти може призводити до накопичення тепла й переходу до небажаного режиму. Розроблений засіб діє інакше: він не чекає критичного перегріву, а реагує на ознаки його наближення. Це дозволяє підтримувати більш рівномірний тепловий режим і зменшувати ймовірність різкого зниження продуктивності через захисні механізми платформи.

На рисунку 3.4 показано порівняння роботи одноплатного комп'ютера у стандартному та енергоефективному режимах, де відображено зміну

температури, частоти CPU та реакцію алгоритму під час переходу від простою до навантаження і подальшої стабілізації системи.

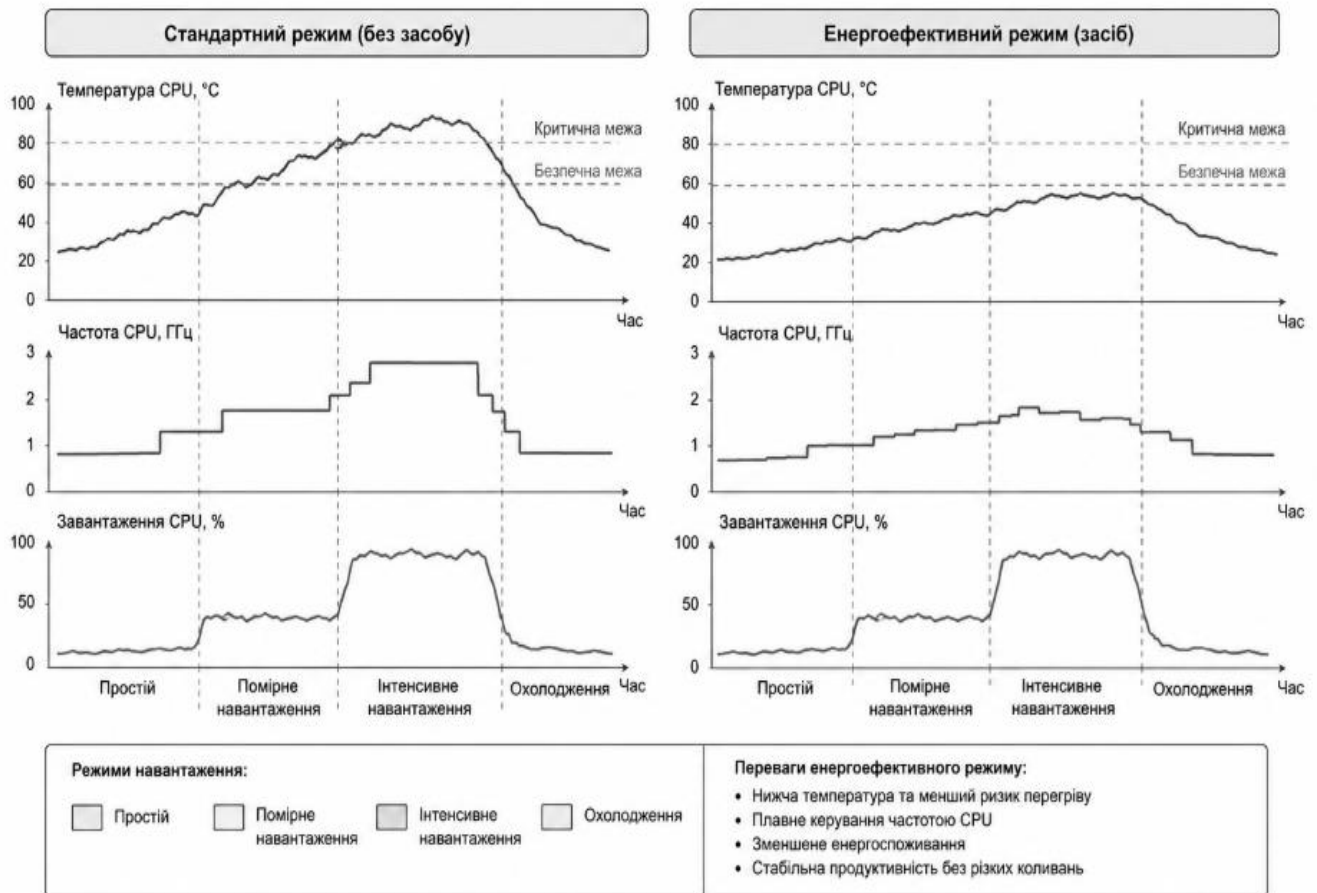


Рисунок 3.4 – Порівняння роботи одноплатного комп’ютера у стандартному та енергоефективному режимах

Водночас аналіз результатів показав, що запропонований підхід не завжди забезпечує максимальну швидкість у коротких обчислювальних задачах. Якщо задача триває дуже мало часу, стандартний режим може виконати її швидше за рахунок миттєвого переходу до високої частоти. Проте для цієї кваліфікаційної роботи головним критерієм є не пікова продуктивність в окремий короткий момент, а стабільність роботи, зменшення перегріву та кращий баланс між продуктивністю й енергоспоживанням. Саме за цими ознаками запропонований підхід показує практичну перевагу.

У третьому розділі кваліфікаційної роботи виконано практичну реалізацію та перевірку роботи програмно-апаратного засобу енергоефективного планування задач. Відповідно до структури пояснювальної записки, у цьому розділі розглянуто формування експериментального середовища, реалізацію модуля збору й журналювання телеметрії, реалізацію алгоритму аналізу стану системи та керування частотою CPU, перевірку засобу в різних режимах навантаження й оцінювання ефективності запропонованого підходу .

Практичну реалізацію розпочато з формування експериментального середовища на базі одноплатного комп'ютера. У цьому середовищі поєднано апаратну платформу, джерело живлення, систему охолодження, операційну систему, програмні модулі збору даних і засоби збереження результатів. Така побудова дала змогу перевірити роботу засобу в умовах, наближених до реального використання компактних обчислювальних платформ, де важливими залишаються стабільність, температура та енергоспоживання.

Практично реалізовано алгоритм аналізу стану системи та керування частотою CPU. У програмній логіці враховано коректність телеметричних даних, поточну температуру, тренд нагріву, завантаження процесора та фактичну частоту. На основі цих показників система визначає режим навантаження та формує керуючу дію. Частота процесора змінюється поступово, без різких переходів між мінімальним і максимальним значенням, що дозволяє зменшити коливання температури та стабілізувати роботу одноплатного комп'ютера.

Перевірку роботи засобу виконано в режимі простою, помірного навантаження, інтенсивного навантаження та температурно напруженого стану. У режимі простою система підтримує економніший режим роботи, оскільки процесор не потребує високої частоти. У режимі помірного навантаження забезпечується баланс між продуктивністю та тепловим станом. Під час інтенсивного навантаження алгоритм підтримує достатню швидкодію, але контролює температуру та реагує на стійкий тренд нагріву. У температурно напруженому режимі засіб переходить до обмеження частоти та очікує стабілізації перед поверненням до продуктивнішого стану.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

## ВИСНОВКИ

У кваліфікаційній роботі розглянуто питання створення програмно-апаратного засобу енергоефективного планування задач на одноплатному комп'ютері. Актуальність обраної теми пов'язана з тим, що одноплатні комп'ютери дедалі частіше використовуються як компактні обчислювальні платформи для моніторингу, автоматизації, навчальних стендів, периферійних обчислень і невеликих серверних сервісів.

Для таких пристроїв важливими є не лише продуктивність, а й стабільна температура, помірне енергоспоживання та здатність працювати тривалий час без перегріву. Саме тому в межах кваліфікаційної роботи увагу зосереджено на поєднанні збору телеметрії, аналізу стану системи та керування частотою процесора на основі зворотного зв'язку.

У першому розділі проаналізовано теоретичні аспекти енергоефективного виконання обчислювальних задач на одноплатних комп'ютерах. Розглянуто особливості таких платформ, їхню архітектуру, обмеження живлення, тепловий режим і роль процесора як основного елемента, що впливає на продуктивність та енергоспоживання.

Окрему увагу приділено тому, що одноплатний комп'ютер поєднує властивості вбудованої системи та повноцінного пристрою з операційною системою, багатозадачністю, файловою системою та стандартними засобами керування ресурсами. Це дало змогу обґрунтувати доцільність використання такої платформи для побудови програмно-апаратного засобу керування режимами роботи CPU.

У другому розділі сформовано архітектуру програмно-апаратного засобу енергоефективного планування задач. Запропоновано структуру, у якій одноплатний комп'ютер, Linux-подібна операційна система, модуль збору телеметрії, блок аналізу стану системи, регулятор частоти CPU та модуль журналювання працюють як єдина система.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

Архітектуру побудовано за принципом замкненого контуру керування: система зчитує параметри, аналізує їх, визначає режим навантаження, формує керуючу дію, застосовує її до процесора та знову перевіряє результат. Це дозволило перейти від звичайного спостереження за станом платформи до активного керування її режимом роботи.

У третьому розділі виконано практичну реалізацію та перевірку роботи програмно-апаратного засобу. Спочатку сформовано експериментальне середовище на базі одноплатного комп'ютера, у якому поєднано апаратну платформу, операційну систему, джерело живлення, систему охолодження, програмні модулі збору даних і засоби журналювання. Таке середовище дало змогу перевірити запропонований підхід у наближених до реальної експлуатації умовах, де процесор працює під різними рівнями навантаження, а температура змінюється поступово відповідно до теплової інерції платформи.

Мету кваліфікаційної роботи досягнуто. Створено програмно-апаратний засіб енергоефективного планування задач на одноплатному комп'ютері, який забезпечує збір телеметричних даних, аналіз стану системи, визначення режиму навантаження, керування частотою CPU та журналювання результатів. У роботі підтверджено, що використання телеметрії та зворотного зв'язку дозволяє підтримувати більш стабільний тепловий режим, зменшити кількість різких перемикань частоти та забезпечити кращий баланс між продуктивністю й енергоспоживанням. У підсумку запропонований засіб виступає практичним рішенням для підвищення передбачуваності роботи одноплатного комп'ютера під змінним навантаженням.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Zhang Y., Fu M., Wu X., Hua Y., Jiang S. Improving Restore Performance of Packed Datasets in Deduplication Systems via Reducing Persistent Fragmented Chunks. *IEEE Transactions on Parallel and Distributed Systems*. 2020. DOI: 10.1109/TPDS.2020.2972898.
2. Charyyev B., Arslan E., Gunes M. H. RIVA: Robust Integrity Verification Algorithm for High-Speed File Transfers. *IEEE Transactions on Parallel and Distributed Systems*. 2020. DOI: 10.1109/TPDS.2020.2966616.
3. Zhang D., Deng Y., Zhou Y., Zhu Y., Qin X. Improving the Performance of Deduplication-Based Backup Systems via Container Utilization Based Hot Fingerprint Entry Distilling. *ACM Transactions on Storage*. 2021. Vol. 17, No. 4. P. 1–23. DOI: 10.1145/3459626.
4. Yang R., Deng Y., Zhou Y., Huang P. Boosting the Restoring Performance of Deduplication Data by Classifying Backup Metadata. *ACM/IMS Transactions on Data Science*. 2021. Vol. 2, No. 2. P. 1–16. DOI: 10.1145/3437261.
5. Zou X., Xia W., Shilane P., Zhang H., Wang X. Building a High-Performance Fine-Grained Deduplication Framework for Backup Storage with High Deduplication Ratio. *2022 USENIX Annual Technical Conference (USENIX ATC 22)*. Carlsbad, 2022. P. 19–36. URL: <https://www.usenix.org/conference/atc22/presentation/zou>
6. Feng D. *Data Deduplication for High Performance Storage System*. Singapore : Springer, 2022. DOI: 10.1007/978-981-19-0112-6.
7. Wang J., Yang X., Lu Z. Cloud Data Integrity Verification Algorithm Based on Data Mining and Accounting Informatization. *Scientific Programming*. 2022. 4756899. DOI: 10.1155/2022/4756899.
8. Viji D., Revathy S. Hash-Indexing Block-Based Deduplication Algorithm for Reducing Storage in the Cloud. *Computer Systems Science and Engineering*. 2023. Vol. 46, No. 1. P. 27–42. DOI: 10.32604/csse.2023.030259.

									Арк.
									75
Зм.	Арк.	№ докум.	Підпис	Дата	КВРКІ.2301101.23.01.24 ПЗ				

9. Harasivka A., Lupenko A., Palaniza Y., Fryz M. Improve Data Backup Strategies with Machine Learning Predictive Analytics. *Proceedings of the 4th International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2024)*. Ternopil, 2024. P. 45–53. URL: <https://ceur-ws.org/Vol-3896/short1.pdf> (дата звернення: 25.02.2026).

10. Kesa D. M. Ensuring Resilience: Integrating IT Disaster Recovery into Corporate Governance Frameworks. *World Journal of Advanced Research and Reviews*. 2023. URL: <https://wjarr.com/sites/default/files/WJARR-2023-1166.pdf> (дата звернення: 25.02.2026).

11. An F., Ye J., Guo Z. A New Data Integrity Verification Scheme with Low Communication Cost. *Journal of Internet Technology*. 2023. Vol. 24, No. 2. P. 283–290. DOI: 10.53106/160792642023032402007.

12. Li P. F., Hua Y., Liu X. An Enhanced Physical-Locality Deduplication System for Backup Systems. *Journal of Computer Science and Technology*. 2024. DOI: 10.1007/s11390-023-2646-7.

13. Hamandawana P., Cho D.-J., Chung T.-S. Speed-Dedup: A New Deduplication Framework for Enhanced Performance and Reduced Overhead in Scale-Out Storage. *Electronics*. 2024. Vol. 13, No. 22. 4393. DOI: 10.3390/electronics13224393.

14. Goswami P., Faujdar N., Debnath S., Khan A. K., Singh G. Investigation on Storage Level Data Integrity Strategies in Cloud Computing: Classification, Security Obstructions, Challenges and Vulnerability. *Journal of Cloud Computing*. 2024. Vol. 13, No. 1. 45. DOI: 10.1186/s13677-024-00605-z.

15. Kavitha R., Shaik M. S., Swarnalatha N., Pujitha M., Hussaini S. A., Khan S., Ali S. Data Deduplication-Based Efficient Cloud Optimisation Technique: Optimizing Cloud Storage through Data Deduplication. *International Journal of Information Engineering and Electronic Business*. 2025. Vol. 17, No. 2. P. 129–146. DOI: 10.5815/ijieeb.2025.02.07.

16. Goel A. Leveraging Data Deduplication Approaches in the Cloud. *Procedia Computer Science*. 2025. URL:

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		76

<https://www.sciencedirect.com/science/article/pii/S1877050925012207> (дата звернення: 25.02.2026).

17. Werkman S. *Roadtested IT Recovery: Domains, Good Practices and Challenges in IT Recovery Testing*. Skövde : University of Skövde, 2025. URL: <https://www.diva-portal.org/smash/get/diva2%3A1983541/FULLTEXT01.pdf> (дата звернення: 25.02.2026).

18. Chen R. Improving Online Restore Performance of Backup Storage via Historical File Access Pattern. *Computers, Materials & Continua*. 2025. DOI: 10.32604/cmс.2025.068878.

19. Joshi A., Mahapatra R. P., Devarajan G. G. An Improved Mechanism to Maintain Data Integrity and Anomaly Detection in Cloud Storage. *Journal of Machine and Computing*. 2026. DOI: 10.53759/7669/jmc202606022.

20. National Institute of Standards and Technology. Ransomware Risk Management: A Cybersecurity Framework Profile. *NIST Interagency or Internal Report 8374*. 2022. DOI: 10.6028/NIST.IR.8374.

21. Cybersecurity and Infrastructure Security Agency. StopRansomware Guide. *CISA*. 2025. URL: <https://www.cisa.gov/sites/default/files/2025-03/StopRansomware-Guide%20508.pdf> (дата звернення: 25.02.2026).

22. Cybersecurity and Infrastructure Security Agency. Ransomware Guide. *CISA*. 2025. URL: <https://www.cisa.gov/stopransomware/ransomware-guide> (дата звернення: 25.02.2026).

23. Australian Cyber Security Centre. Ransomware Prevention Guide. *ACSC*. 2022. URL: [https://www.cyber.gov.au/sites/default/files/2023-03/ACSC\\_Ransomware\\_Prevention\\_Guide\\_05082022.pdf](https://www.cyber.gov.au/sites/default/files/2023-03/ACSC_Ransomware_Prevention_Guide_05082022.pdf) (дата звернення: 25.02.2026).

24. Canadian Centre for Cyber Security. ITSM.00.099 Ransomware Playbook. *Canadian Centre for Cyber Security*. 2021. URL: <https://www.cyber.gc.ca/sites/default/files/cyber/2021-12/itsm00099-ransomware-playbook-2021-final3-en.pdf> (дата звернення: 25.02.2026).

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		77

25. ENISA. ENISA Threat Landscape for Ransomware Attacks. *European Union Agency for Cybersecurity*. 2022. URL: <https://www.enisa.europa.eu/publications/enisa-threat-landscape-for-ransomware-attacks> (дата звернення: 25.02.2026).

26. ENISA. ENISA Threat Landscape 2023. *European Union Agency for Cybersecurity*. 2023. URL: <https://www.enisa.europa.eu/sites/default/files/publications/ENISA%20Threat%20Landscape%202023.pdf> (дата звернення: 25.02.2026).

27. Cohesity. Ransomware Readiness: An In-Depth Evaluation Guide. *Cohesity*. 2022. URL: <https://www.cohesity.com/content/dam/cohesity/resource-assets/guides/ransomware-readiness-guide-aws.pdf> (дата звернення: 25.02.2026).

28. Quorum. 2022 Ransomware Preparedness Guide. *Quorum*. 2022. URL: <https://d2ruqfpc8khp78.cloudfront.net/resources/Ransomware-Preparedness-Guide-2022-Quorum.pdf> (дата звернення: 25.02.2026).

29. Barracuda. The Smarter MSP's Guide to Ransomware. *Barracuda*. 2021. URL: [https://www.barracuda.com/content/dam/barracuda-msp/docs/resources/pdf/ebooks/EBK\\_Ransomware\\_Final.pdf](https://www.barracuda.com/content/dam/barracuda-msp/docs/resources/pdf/ebooks/EBK_Ransomware_Final.pdf) (дата звернення: 25.02.2026).

30. Bacula Systems. Ransomware Prevention Checklist. *Bacula Systems*. 2021. URL: <https://www.baculasystems.com/wp-content/uploads/2021/01/ransomware-prevention-checklist.pdf> (дата звернення: 25.02.2026).

31. Veeam Software. The Best Practices of Ransomware Recovery. *Veeam Software*. 2022. URL: [https://www.issummit.org/wp-content/uploads/2022/09/Veeam\\_Information-Security-Summit-6-Sep-2022-compressed.pdf](https://www.issummit.org/wp-content/uploads/2022/09/Veeam_Information-Security-Summit-6-Sep-2022-compressed.pdf) (дата звернення: 25.02.2026).

32. European Data Protection Board. Guidelines 01/2021 on Examples regarding Personal Data Breach Notification. *EDPB*. 2021. URL: [https://www.edpb.europa.eu/system/files/2022-01/edpb\\_guidelines\\_012021\\_pdbnotification\\_adopted\\_en.pdf](https://www.edpb.europa.eu/system/files/2022-01/edpb_guidelines_012021_pdbnotification_adopted_en.pdf) (дата звернення: 25.02.2026).

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		78

33. OpenZFS. zfs-send(8). *OpenZFS Documentation*. 2023. URL: <https://openzfs.github.io/openzfs-docs/man/master/8/zfs-send.8.html>

34. Peluso V., Rizzo R. G., Calimera A. Performance Profiling of Embedded ConvNets under Thermal-Aware DVFS. *Electronics*. 2019. Vol. 8, No. 12. 1423. DOI: 10.3390/electronics8121423.

35. Benoit-Cattin T., Velasco-Montero D., Fernández-Berni J. Impact of Thermal Throttling on Long-Term Visual Inference in a CPU-Based Edge Device. *Electronics*. 2020. Vol. 9, No. 12. 2106. DOI: 10.3390/electronics9122106.

36. Garcia A. M., Serpa M. S., Griebler D., Schepke C., Fernandes L. G. L., Navaux P. O. A. The Impact of CPU Frequency Scaling on Power Consumption of Computing Infrastructures. *Computational Science and Its Applications – ICCSA 2020. Lecture Notes in Computer Science*. 2020. Vol. 12254. P. 142–157. DOI: 10.1007/978-3-030-58817-5\_12.

37. Lambert J., Monahan R., Casey K. Power Consumption Profiling of a Lightweight Development Board: Sensing with the INA219 and Teensy 4.0 Microcontroller. *Electronics*. 2021. Vol. 10, No. 7. 775. DOI: 10.3390/electronics10070775.

38. Swarup S., Shakshuki E. M., Yasar A. Energy Efficient Task Scheduling in Fog Environment Using Deep Reinforcement Learning Approach. *Procedia Computer Science*. 2021. Vol. 191. P. 65–75. DOI: 10.1016/j.procs.2021.07.012.

39. Babchuk S., Humeniuk T., Romaniv I. T. Mathematical Models Productivity of Cluster System Based on Raspberry Pi 3B+. *Radio Electronics, Computer Science, Control*. 2021. No. 1. P. 46–56. DOI: 10.15588/1607-3274-2021-1-5.

40. Arroba P., Buyya R., Cárdenas R., Risco-Martín J. L., Moya J. M. Sustainable Edge Computing: Challenges and Future Directions. *Software: Practice and Experience*. 2024. Vol. 54, No. 11. P. 2272–2296. DOI: 10.1002/spe.3340.

41. Maseda T., Enes J., Expósito R. R., Touriño J. Automated Approach for Accurate CPU Power Modelling. *2024 IEEE International Conference on Cluster Computing (CLUSTER)*. 2024. P. 97–107. DOI: 10.1109/CLUSTER59578.2024.00016.

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		79

42. Ahmadi A., Abdelhafez H. A., Pattabiraman K., Ripeanu M. EdgeEngine: A Thermal-Aware Optimization Framework for Edge Inference. *Proceedings of the Eighth ACM/IEEE Symposium on Edge Computing*. 2023. P. 67–79. DOI: 10.1145/3583740.3626616.

43. Pyrhönen J., Ikonen J. Energy-Saving Techniques for Raspberry Pi. *2025 International Conference on Computer Systems and Technologies (CompSysTech)*. 2025. DOI: 10.1109/CompSysTech65493.2025.11136932.

44. Qiu B., Chen Y., Khooi X. Z., Song C. H., Chan M. C. Revisiting CPU Performance Scaling for Energy-Efficient Packet Processing Applications. *Proceedings of the ACM International Conference on Future and Sustainable Energy Systems*. 2025. P. 385–397. DOI: 10.1145/3679240.3734588.

45. Samadi N., Yadollahi F., Shah-Mansouri H. iDEAS: Intelligent DVFS for Energy-Efficient Task Scheduling in Mobile Devices With big.LITTLE Computing Architecture. *IEEE Access*. 2025. Vol. 13. P. 200711–200724. DOI: 10.1109/ACCESS.2025.3636995.

46. Minhas S. F., Shah M. H. Adaptive Filtering Technique for Mitigating Soil Mineralization Effects in Pulse-Induced Metallic Mine Detectors. *Measurement*. 2026. Vol. 266. 120473. DOI: 10.1016/j.measurement.2026.120473.

47. Baker B. та ін. A Survey of Short-Range Wireless Communication for Ultra-Low-Power Embedded Systems. *Journal of Low Power Electronics and Applications*. 2024. Vol. 14, No. 2. 27. DOI: 10.3390/jlpea14020027.

48. Bhaganagare S., Chavan S., Gavali S., Godase V. The Vocal Cortex of the Smart Home: A Review of ESP32-Based Control and Automation Architectures. *International Journal of Advanced Research in Science, Communication and Technology*. 2025. Vol. 5, No. 4. P. 1071–1094. DOI: 10.48175/IJARST-29460.

49. Li P. Improving the Restore Performance via Physical-Locality in Deduplication-Based Backup Systems. *Proceedings of ACM*. 2020. Pp. 341–355. DOI: 10.1145/3423211.3425691.

50. Raspberry Pi. URL: <https://arduino.ua/prod6505-raspberry-pi-5-4gb>

					КВРКІ.2301101.23.01.24 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		80







## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Валентин ГОМЕНЮК

**Співавтор:**

**Назва:** Програмно-технічний засіб керування дистанційним металошукачем на основі мікроконтролера ESP32

**Експерт:** Василь ЯЦКІВ

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 2.67%

**Коефіцієнт подібності 2:** 1.6%

**Мікропробіли:** 3

**Заміна букв:** 0

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2026-05-28 22:37:04.0

**Після аналізу Звіту подібності констатую наступне:**

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

**Обґрунтування:**

2026-05-29

Дата



Доцент Андрій Нічепорук

експерт

# Anti-Plagiarism (<http://ap.km.ua>) v-15.701

**Максимальне співпадіння з одним документом 1.0%**

**Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 8%**

ID: 272723 Назва: БКР Програмно-технічний засіб керування дистанційним металошукачем на основі мікроконтролера ESP32 Додано в БД: 2026-05-29 Автора: Валентин ГОМЕНЮК Керівники: Василь ЯЦКІВ Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	135365	999	2221 (2%)	29 (3%)

## Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

## РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Гоменюк Валентин Сергійович

Тема: Енергоефективне планування задач на одноплатному комп'ютері

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   71  

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є проектування, програмна реалізація та перевірка працездатності програмно-апаратного засобу енергоефективного планування задач на одноплатному комп'ютері. У роботі прийнято рішення використати одноплатний комп'ютер як центральну обчислювальну платформу, на якій виконується збір телеметричних даних, аналіз стану системи та керування режимами роботи процесора.

2. Висновок про відповідність роботи дипломному завданню: Кваліфікаційна робота повністю відповідає поставленому завданню. У роботі розкрито теоретичні аспекти енергоефективного виконання обчислювальних задач на одноплатних комп'ютерах, виконано розроблення програмно-апаратного засобу та проведено практичну перевірку його роботи.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі розглянуто особливості одноплатних комп'ютерів як платформи для прикладних і вбудованих обчислень, проаналізовано вплив навантаження та частоти процесора на енергоспоживання і тепловий режим. У другому розділі сформовано архітектуру програмно-апаратного засобу, обґрунтовано вибір апаратних і програмних компонентів, описано організацію збору телеметричних даних і розроблено логіку визначення режиму навантаження. У третьому розділі виконано практичну реалізацію програмного модуля збору та журналювання телеметрії, реалізовано алгоритм аналізу стану системи й керування частотою CPU, а також проведено перевірку роботи засобу в різних режимах навантаження.

4. Позитивні сторони роботи: Позитивною стороною кваліфікаційної роботи є її практична спрямованість. Запропонований засіб не обмежується звичайним моніторингом параметрів, а формує замкнений контур керування, у якому телеметричні дані використовуються для прийняття рішень щодо режиму роботи процесора. Важливою перевагою є врахування не лише завантаження CPU, а й температури, динаміки нагріву та поточної частоти.

5. Негативні сторони роботи: недостатня увага аналізу предметної частини. Недостатньо чітко описано складання програмно-технічного засобу.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно та має логічну структуру. Матеріал викладено послідовно: від аналізу предметної області до розроблення архітектури та практичної перевірки програмно-апаратного засобу. Графічна частина відповідає темі роботи та включає схему електричну функціональну, схему електричну принципову та блок-схему алгоритму програмного забезпечення.

7. Відгук про роботу в цілому: Кваліфікаційна робота виконана на достатньому технічному рівні.

8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: задовільно (D / 70)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

*Бедосмон Леонід Сергійович, зав.серії, ХЦУ*

“01” *червня* 2026 р.

 (підпис)

Зав. кафедри КІС  
д-р. філософії Ользі ПАВЛОВІЙ

Валентин ГОМЕНЮК

---

ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи КІ2с-23-1

### ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року

*Гоу*

Гоменюк В.С.

## РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

### КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Енергоефективне планування задач на одноплатному комп'ютері

Автор Валентин ГОМЕНЮК

Освітня програма Комп'ютерна інженерія та програмування

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: д.т.н., проф Яцків Василь Васильович

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

#### Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 2.67%; та системою Anti-Plagiarism складає 1% що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

01.06.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи

  
Підпис  
  
Підпис  
  
Підпис

Ольга ПАВЛОВА  
Ім'я, ПРІЗВИЩЕ

Андрій НІЧЕПОРУК  
Ім'я, ПРІЗВИЩЕ

Василь ЯЦКІВ  
Ім'я, ПРІЗВИЩЕ