

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

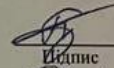
Галузь знань 12 - Інформаційні технології

Спеціальність 123 -Комп'ютерна інженерія

на тему «Метод симуляції енергоспоживання мікроконтролера»

КвРКІП.2302173.24.02.08 ПЗ

Виконав: студент 2 курсу, група КІ2м-23-1

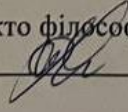
  
Підпис

Владислав БРИЦЬКИЙ  
Ім'я, прізвище

Керівник к.т.н., доцент  
Науковий ступінь, вчене звання

  
Підпис

Дмитро МЕДЗАТИЙ  
Ім'я, прізвище

До захисту допускаю:  
Зав. кафедри КІС, докто філософії, доцент  
Ольга ПАВЛОВА  
  
29 04 2025 р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 01 ” 09 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА**

Владиславу БРИЦЬКОМУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод симуляції енергоспоживання мікроконтролера

Керівник проекту (роботи) Дмитро МЕДЗАТИЙ, к.т.н., доцент

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

Затверджена наказом ректора університету від 08.01.2025 №8

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

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

Аналіз сучасних підходів до зниження енергоспоживання мікроконтролерів

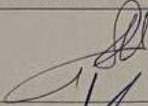
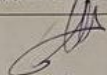


Огляд архітектури та режимів енергоспоживання мікроконтролера ATmega328P

Розробка симуляційної моделі в середовищі Proteus

Реалізація ПЗ для керування режимами енергозбереження

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

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КПС		
Антиплагиат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 01 » 09 2024р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики КвРМ з керівником	01.09.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2024	виконано
3	Робота над розділом 1 - аналіз відомих моделей, методів за темою; постановка задачі	01.11.2024	виконано
4	Робота над розділом 2 - розробка моделей для вирішення поставленої задачі	01.12.2024	виконано
5	Робота над науковою статтею	01.02.2025	виконано
6	Робота над розділом 3 - розробка методів для вирішення поставленої задачі	15.02.2025	виконано
7	Робота над розділом 4 - проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2025	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	виконано
9	Попередній захист ДРМ	29.04.2025	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2025	

Студент

  
Підпис

Владислав БРИЦЬКИЙ  
Ім'я, прізвище

Керівник роботи

  
Підпис

Дмитро МЕДЗАТИЙ  
Ім'я, прізвище

## РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Метод симуляції енергоспоживання мікроконтролера

Автор роботи: Брицький Владислав

Керівник роботи: Медзатий Дмитро

Пояснювальна записка: 73 с., 17 рис., 8 табл., 2 дод., 80 джерел.

Енергоспоживання, мікроконтролер, ATmega328P, Proteus, симуляція, енергоефективність

Об'єктом дослідження є процес енергоспоживання мікроконтролера вбудованих систем у різних режимах його роботи.

Предметом дослідження є методи зниження енергоспоживання мікроконтролера ATmega328P за допомогою застосування енергозберігаючих режимів та засобів моделювання в середовищі Proteus.

Метою кваліфікаційної роботи магістра є аналіз енергоспоживання мікроконтролера ATmega328P та розробка симуляційної моделі в середовищі Proteus для дослідження ефективності різних режимів його роботи з метою подальшої оптимізації енергозбереження.

Для розв'язання поставлених задач використовувалися методи структурного аналізу, моделювання в середовищі Proteus, експериментального дослідження енергоспоживання, порівняльного аналізу теоретичних і практичних результатів.

Наукова новизна отриманих результатів:

– набув подальшого розвитку метод симуляції енергоспоживання мікроконтролерів, який базується на моделюванні роботи пристрою у середовищі Proteus із врахуванням різних режимів роботи (Active, Sleep, Deep Sleep), а також із застосуванням змін тактової частоти та напруги живлення.

Наукова новизна полягає також у використанні віртуальних інструментів Proteus для моделювання та кількісної оцінки миттєвого споживання струму без необхідності фізичного прототипування, що значно спрощує попередній етап енергоаналізу мікроконтролерних систем.

На основі проведених досліджень розроблена архітектура і компоненти програмного забезпечення прикладного коду для управління режимами роботи мікроконтролера ATmega328P із реалізацією переходу в режим енергозбереження, а також схема в Proteus для аналізу енергоспоживання.

Практична значимість отриманих результатів полягає у можливості впровадження розробленої симуляційної моделі у навчальному процесі та при розробці енергоефективних вбудованих пристроїв на базі ATmega328P.

Розділ 1. Проведено огляд існуючих підходів до вимірювання та аналізу енергоспоживання мікроконтролерів. Визначено основні параметри, що впливають на споживання енергії: частота роботи, рівень напруги живлення, тривалість виконання програм, а також режими роботи (активний, standby, sleep, deep sleep). Поглиблено вивчено поняття динамічного та статичного енергоспоживання як критично важливих характеристик при оцінці ефективності мікроконтролерів. Також було охарактеризовано сучасні тенденції в енергоефективному проектуванні вбудованих систем.

Розділ 2. Проведено детальне дослідження особливостей моделювання енергоспоживання AVR-мікроконтролерів. Обґрунтовано вибір мікроконтролера ATmega32. Розглянуто вплив таких параметрів, як тактова частота, рівень напруги та тип вибраного режиму роботи на загальне енергоспоживання. Проаналізовано приклади практичного застосування AVR-мікроконтролерів у енергоефективних системах: автономних сенсорних вузлах, пристроях IoT, системах автоматизації.

Розділ 3. Розроблено метод симуляції енергоспоживання мікроконтролера з використанням програмного середовища Proteus. Обґрунтовано вибір середовища симуляції, описано архітектуру побудованої схеми, етапи налаштування параметрів живлення та запуску прошивки.

Розділ 4. Реалізовано повномасштабне симуляційне дослідження. Розроблено електронну модель з можливістю імітації перемикання режимів мікроконтролера.

У висновках підведено підсумки результатів роботи.

## ЗМІСТ

<b>СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....</b>	<b>5</b>
<b>ВСТУП.....</b>	<b>6</b>
<b>1 ТЕОРЕТИЧНІ ОСНОВИ ЕНЕРГОСПОЖИВАННЯ МІКРОКОНТРОЛЕРІВ .....</b>	<b>8</b>
1.1 Поняття енергоспоживання у мікроконтролерах .....	8
1.2 Методи вимірювання енергоспоживання .....	21
1.3 Постановка задачі.....	26
1.4 Висновки до першого розділу.....	26
<b>2 МОДЕЛЮВАННЯ AVR-МІКРОКОНТРОЛЕРІВ.....</b>	<b>27</b>
2.1 Модель енергоспоживання AVR-мікроконтролерів.....	28
2.2 Моделювання для AVR-мікроконтролерів за параметром енергоспоживання.....	38
2.3 Висновки до даного розділу .....	45
<b>3 МЕТОД СИМУЛЯЦІЇ ЕНЕРГОСПОЖИВАННЯ МІКРОКОНТРОЛЕРА .</b>	<b>47</b>
3.1 Модель симуляції енергоспоживання мікроконтролера .....	47
3.2 Метод симуляції енергоспоживання мікроконтролера .....	55
3.3 Висновки до даного розділу .....	63
<b>4 РЕАЛІЗАЦІЯ ТА СИМУЛЯЦІЯ МОДЕЛІ ЕНЕРГОСПОЖИВАННЯ .....</b>	<b>65</b>
4.1 Реалізація методу симуляції енергоспоживання мікроконтролера .....	65
4.2 Експериментальні дослідження та їх аналіз.....	72
4.3 Висновки до даного розділу.....	77
<b>ВИСНОВКИ .....</b>	<b>79</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....</b>	<b>80</b>

<b>ДОДАТОК А</b> Публікація .....	88
<b>ДОДАТОК Б</b> Програмна реалізація управління режимами роботи мікроконтролера ATMEGA328P.....	91

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IoT - інтернет речі

КП - керуючий пристрій

ПЗП - постійний запам'ятовуючий пристрій

ОЗП - оперативний запам'ятовуючий пристрій

АЦП - аналого-цифровий перетворювач

АЛП - арифметико-логічний пристрій

ПК - персональний комп'ютер

## ВСТУП

У сучасному світі стрімкий розвиток технологій вбудованих систем та Інтернету речей (IoT) вимагає підвищеної уваги до питань енергоефективності мікроконтролерів. Зростання кількості автономних пристроїв, що працюють на батарейному живленні, створює потребу в оптимізації енергоспоживання для забезпечення їх тривалої та стабільної роботи [1].

IoT охоплюють величезну кількість пристроїв, здатних збирати, обробляти та передавати дані. Однією з ключових вимог до таких пристроїв є енергоефективність, оскільки багато з них працюють від автономних джерел живлення, таких як батарейки або акумулятори.

Мікроконтролери стали основою для IoT-пристроїв завдяки своїм компактним розмірам, низькому енергоспоживанню та широкому функціоналу. Використання енергозберігаючих режимів у мікроконтролерах дозволяє значно продовжити тривалість автономної роботи пристроїв, що особливо важливо для систем, які не мають постійного доступу до джерела живлення.

Актуальність роботи полягає у зростаючій необхідності зниження енергоспоживання мікроконтролерів у вбудованих системах, особливо для автономних пристроїв, що працюють від батарей. Використання ефективних методів моделювання дозволяє оптимізувати споживану потужність ще на етапі розробки.

Метою кваліфікаційної роботи магістра є дослідження енергоспоживання мікроконтролера ATmega328P у різних режимах роботи та розробка методу його симуляції для оцінки ефективності енергозбереження.

Поставлена мета досягається розв'язанням таких основних завдань:

- 1) аналіз програмного забезпечення для симуляції мікроконтролерів та вибір оптимального середовища моделювання;
- 2) розробка алгоритму проведення симуляції енергоспоживання мікроконтролера;

3) проведення експериментальних досліджень у різних режимах роботи мікроконтролера;

4) аналіз отриманих результатів та визначення можливих шляхів оптимізації енергоспоживання.

Об'єктом дослідження є енергоспоживання мікроконтролера ATmega328P у різних режимах роботи, включаючи активний режим, режим сну (sleep mode) та глибокий сон (deep sleep mode).

Предметом дослідження є методи моделювання та аналізу енергоспоживання мікроконтролера ATmega328P у різних режимах роботи, а також способи його оптимізації для зменшення споживаної потужності.

Наукова новизна отриманих результатів - набув подальшого розвитку метод симуляції енергоспоживання мікроконтролерів, який базується на моделюванні роботи пристрою у середовищі Proteus із врахуванням різних режимів роботи (Active, Sleep, Deep Sleep), а також із застосуванням змін тактової частоти та напруги живлення.

Практична значимість отриманих результатів полягає в можливості використання розробленого методу симуляції для оцінки та оптимізації енергоспоживання мікроконтролерів у вбудованих системах. Це дозволить зменшити енергозатрати автономних пристроїв, підвищити їхню автономність та ефективність роботи.

Для розв'язання поставлених задач використовувалися методи комп'ютерного моделювання, аналізу енергоспоживання, експериментального дослідження, а також порівняльний аналіз отриманих даних у різних режимах роботи мікроконтролера.

За темою кваліфікаційної роботи опубліковано одну публікацію [81] у матеріалах конференції АПКН-2024 (Хмельницький національний університет).

# 1 ТЕОРЕТИЧНІ ОСНОВИ ЕНЕРГОСПОЖИВАННЯ МІКРОКОНТРОЛЕРІВ

## 1.1 Поняття енергоспоживання у мікроконтролерах

У сучасних цифрових системах, особливо у сфері IoT, мікроконтролери є основним обчислювальним елементом. Вони забезпечують роботу багатьох автономних пристроїв, функціонуючи в умовах обмеженого енергозабезпечення.

З розвитком технологій зростає попит на енергоефективні рішення, які дозволяють зменшити споживання електроенергії без втрати продуктивності. Розуміння внутрішньої структури мікроконтролера, особливостей його функціональних блоків та режимів роботи є ключовим для аналізу джерел енергоспоживання та пошуку шляхів його оптимізації.

Інтеграція мікроконтролерів у пристрої IoT дозволила створити компактні, гнучкі та масштабовані системи, що можуть працювати в автономному режимі тривалий час.

Водночас одним із ключових викликів у розробці таких систем є забезпечення мінімального енергоспоживання, особливо в умовах, коли пристрої розміщуються у віддалених або важкодоступних місцях і працюють від автономних джерел живлення - батарей або сонячних панелей.

Енергоефективність мікроконтролерів у такому контексті набуває першочергового значення. Їхня здатність працювати в різних енергозберігаючих режимах (наприклад, Sleep Mode, Idle Mode, Power-Down) дозволяє суттєво скоротити споживання енергії без шкоди для основної функціональності. Це досягається шляхом відключення несуттєвих модулів або зниження частоти тактового генератора в періоди неактивності пристрою.

Ефективне управління енергоспоживанням стало одним із ключових завдань у проєктуванні сучасних вбудованих систем, адже воно не лише продовжує термін служби пристроїв, але й знижує витрати на їх обслуговування [5].

Енергоспоживання мікроконтролерів залежить від багатьох факторів, включаючи архітектуру процесора, тактову частоту, робочу напругу, режими роботи (активний, енергозберігаючий, сплячий), а також ефективність програмного коду [2].

Крім того, енергоспоживання є важливим показником продуктивності вбудованих систем. Мікроконтролери забезпечують обчислювальні та обробні можливості для вбудованих систем і також повинні дотримуватися енергетичних обмежень.

Adegbiya та ін. [3] зазначають, що процесори IoT повинні бути оптимізовані для енергоефективності та конфігуровані для досягнення оптимального виконання, особливо з точки зору енергоефективності, оскільки енергоспоживання є інтегралом потужності в часі, потужність - не єдиний фактор впливу на енергоспоживання.

Як показали дослідження, наведені у статтях "Instruction Level Power Analysis and Optimization of Software" [78] та "Energy-Efficient Power Management Techniques" [2], оптимізація енергоспоживання досягається не лише зниженням потужності, але й скороченням часу виконання завдань та правильним використанням режимів енергозбереження .

Одним із важливих підходів до оптимізації енергоспоживання є моделювання та симуляція роботи мікроконтролерів за різних умов. Використання програмних інструментів, таких як Proteus та Logisim, дозволяє не лише спостерігати за роботою мікроконтролерів у різних режимах, але й оцінювати їх енергоспоживання без фізичного втручання в апаратну частину [63].

Додатково, апаратні вимірювання з використанням мультиметрів та лабораторних джерел живлення забезпечують точні дані для аналізу та оптимізації.

Актуальність дослідження також підкріплюється глобальною тенденцією до зменшення вуглецевого сліду та підвищення ефективності використання ресурсів. Дослідження показали, що правильний вибір режимів енергозбереження (Sleep,

Stop, Standby) та оптимізація динамічного масштабування частоти (DVFS) можуть значно зменшити енергоспоживання систем [22].

Результати дослідження можуть бути використані для підвищення енергоефективності пристроїв, що базуються на мікроконтролерах, у різних сферах застосування, включаючи промислову автоматизацію, медичні прилади та побутову електроніку.

Застосування енергоефективних мікроконтролерів охоплює широкий спектр пристроїв, які знаходять своє застосування у повсякденному житті, промисловості, медицині, сільському господарстві, безпеці тощо.

Приклади таких пристроїв є побутова електроніка та "розумний дім" - розумні термостати (наприклад, Google Nest) - автоматично регулюють температуру в приміщенні залежно від присутності людей або часу доби. Датчики руху - активують освітлення або сигналізацію лише при виявленні руху. Розумні розетки та лампи - відключають живлення, коли пристрої не використовуються, знижуючи енергоспоживання. Системи контролю освітлення - знижують яскравість у нічний час або за відсутності людей.

Також в медицині та охороні здоров'я - портативні медичні монітори (наприклад, пульсоксиметри, глюкометри, ЕКГ-монітори) - працюють від батарейок та оптимізовані для тривалої роботи без підзарядки. Носимі пристрої для пацієнтів - контролюють життєві показники та передають дані на медичні сервери;

В сільському господарстві - сенсори вологості ґрунту - вимірюють рівень вологості і передають дані для автоматичного керування зрошенням. Станції моніторингу погоди - фіксують температуру, вологість, атмосферний тиск, інтенсивність опадів тощо. GPS-мітки на тваринах - визначають місцезнаходження худоби та активність для виявлення хвороб або пошуку;

Транспорт і логістика - розумні трекери вантажу - визначають координати, вібрацію, перевернення, температуру товару. Датчики тиску в шинах - зменшують споживання пального та попереджають аварії. Системи моніторингу рівня палива - допомагають уникнути крадіжок і знижують експлуатаційні витрати.

Ще одним із прикладів є охорона та безпека - розумні камери спостереження з детекцією руху - активуються тільки при виявленні об'єкта. Датчики відкриття дверей і вікон - працюють від батарей і сигналізують про несанкціонований доступ. Портативні тривожні кнопки для людей похилого віку - активуються лише при натисканні або в разі падіння.

Широке застосування мікроконтролерів у таких системах вимагає глибокого розуміння їхньої внутрішньої структури та принципів роботи. Це дозволяє не лише оптимізувати програмну логіку, а й зменшити енергоспоживання шляхом раціонального використання ресурсів пристрою.

Мікроконтролери з однаковою архітектурою можуть відрізнитися набором функціональних вузлів і характером зв'язку між ними.

Будь-який мікроконтролер містить такі функціональні блоки як центральний процесор або процесорне ядро - це головний елемент, що виконує обчислювальні операції. Воно складається з керуючого пристрою (КП), який відповідає за послідовність виконання інструкцій, та операційного пристрою (ОП), який реалізує арифметичні, логічні та побітові операції.

Ядро також забезпечує взаємодію з регістрами, шинами даних та оперативною пам'яттю. Сучасні ядра можуть мати кілька рівнів оптимізації, наприклад, конвеєризацію, виконання інструкцій за один цикл, підтримку переривань тощо.

Наступним блоком є внутрішня пам'ять (РП, кеш-пам'ять, блоки ОЗП - використовується для тимчасового зберігання змінних і проміжних даних і ПЗП - у якій зберігається програмний код користувача);

Також сюди відносять інтерфейсний блок - забезпечує обмін інформацією між мікроконтролером та зовнішніми пристроями. До складу інтерфейсних засобів входять послідовні порти (UART, SPI, I<sup>2</sup>C), паралельні порти, USB, CAN, Ethernet тощо. Через ці порти здійснюється комунікація з іншими мікроконтролерами, комп'ютерами, сенсорами, модулями пам'яті, дисплеями, виконавчими механізмами тощо;

периферійні пристрої - це функціональні модулі, що розширюють можливості мікроконтролера без потреби в зовнішніх компонентах (таймерні модулі, аналого-цифрові перетворювачі, спеціалізовані контролери);

І різні допоміжні схеми - це апаратні компоненти, які підтримують стабільну роботу мікроконтролера та полегшують процес розробки (генератори тактових імпульсів, схеми налагодження і тестування, сторожовий таймер тощо).

Структурну схему сучасного мікроконтролера відображено на рисунку 1.1.

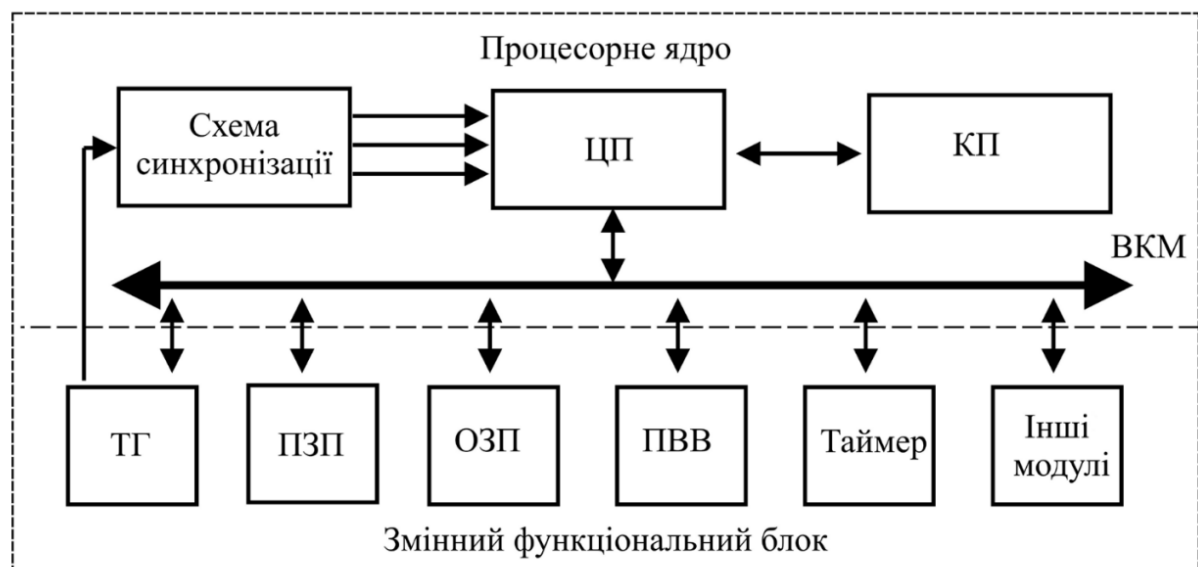


Рисунок 1.1 - Структурна схема сучасного мікроконтролера

Зв'язок між складовими частинами мікропроцесорної системи здійснюється за допомогою внутрішньої контрольної магістралі, яка є системою комунікації між основними блоками мікроконтролера. Вона складається з шини даних, шини адреси та шини керування.

Шина даних забезпечує передачу інформації між пристроями, шина адреси використовується для визначення джерела або одержувача даних, а шина керування синхронізує операції читання, запису та обміну сигналами керування.

Для того, щоб усі блоки мікроконтролера працювали злагоджено та узгоджено, їхня робота синхронізується за допомогою тактового генератора, який

формує імпульси заданої частоти, а також схеми синхронізації, яка координує обробку сигналів усередині пристрою.

Висока точність і стабільність тактового сигналу є критично важливими для забезпечення коректної роботи всіх функціональних блоків мікроконтролера.

Однією з найважливіших характеристик мікроконтролера є розрядність шини даних, тобто кількість бітів, які можуть передаватися або оброблятися одночасно. Від розрядності залежить продуктивність, обсяг адресованої пам'яті та енергоспоживання мікроконтролера. Залежно від цієї ознаки мікроконтролери поділяються на наступні класи:

1) 1-бітні - використовуються в дуже простих автоматизованих пристроях або в якості контролерів вузької спеціалізації (MC14500 фірми Motorola);

2) 4-бітні - застосовуються в побутовій електроніці, калькуляторах, простих іграшках та деяких сенсорах (Atmel MARC4, Winbond W742, NEC uPD75 тощо);

3) 8-бітні - найбільш поширений клас мікроконтролерів завдяки хорошему балансу між продуктивністю, ціною та енергоспоживанням. Застосовуються в більшості побутових пристроїв, датчиків, IoT-систем (Intel MCS-48, Intel MCS-51, Atmel ATtiny/ATmega/ATXmega, Microchip PIC12/16/18, Zilog Z86);

4) 16-бітні - мають вищу продуктивність і розширені можливості обробки даних. Застосовуються в системах реального часу, електроніці автомобілів, медичних приладах (Intel MCS-96, Texas Instruments MSP430, Motorola 68HC16, Fujitsu MB90, Infineon C16, Mitsubishi M16C, Microchip PIC24 тощо);

5) 32-бітні - найпотужніші в класі мікроконтролерів, підтримують складні алгоритми та обробку великої кількості інформації. Широко використовуються в промислових системах, мережевих пристроях, системах безпеки (Atmel ARM, Fujitsu MB91, NEC V850, NXP LPC2xxx тощо)

На рисунку 1.2 відображено класифікацію мікроконтролерів за призначенням.

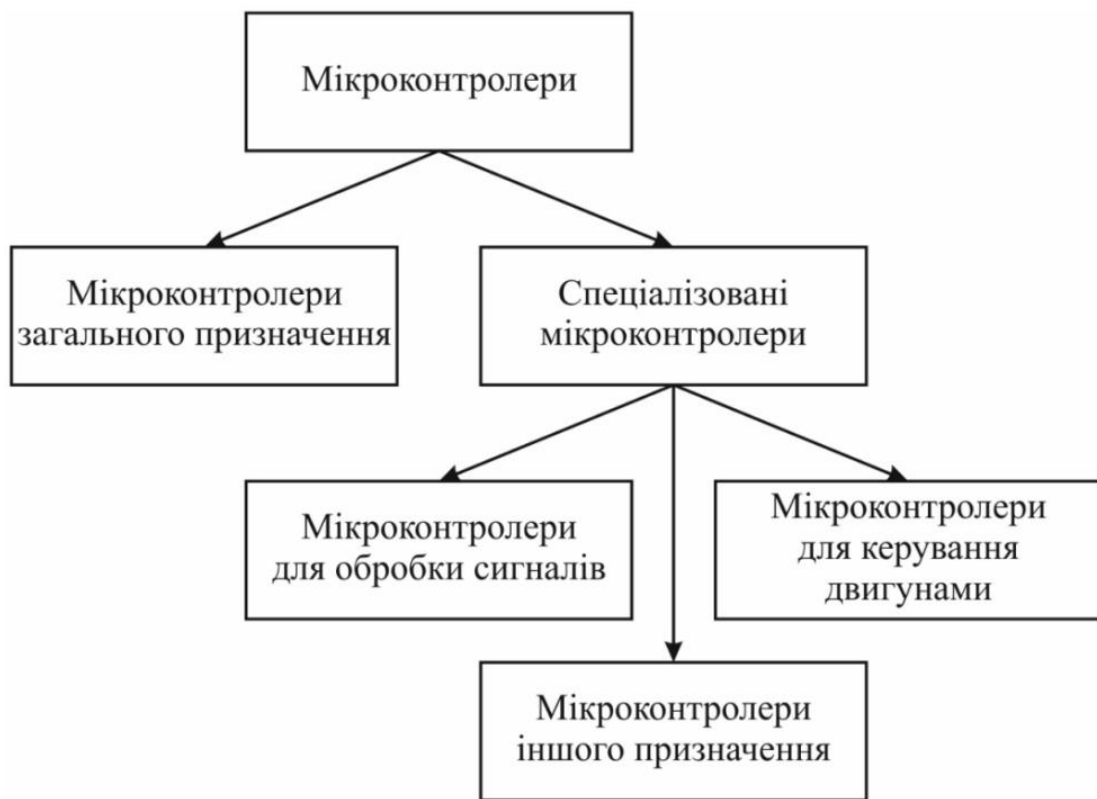


Рисунок 1.2 - Класифікація мікроконтролерів за призначенням

Енергоспоживання мікроконтролерів - це один із ключових параметрів, що визначає ефективність роботи пристроїв, їхню продуктивність, стабільність функціонування та придатність до використання у вбудованих системах.

У системах з обмеженим джерелом живлення, таких як пристрої на акумуляторах або енергозалежні бездротові сенсорні мережі, оптимальне управління енергоспоживанням набуває критичного значення. Від цього залежить тривалість автономної роботи, частота обслуговування та навіть надійність функціонування в реальному часі.

Особливо актуальним є питання енергозбереження у сферах, де від стабільності й автономності залежить людське життя - наприклад, у медичних імплантах або моніторингових системах життєвих показників.

Те саме стосується IoT-пристроїв, що часто працюють у важкодоступних умовах без постійного живлення. У подібних випадках мікроконтролери повинні бути не лише функціональними, а й максимально енергоефективними [1], [5].

На енергоспоживання мікроконтролерів впливає кілька факторів. По-перше, це архітектура процесора: різні типи, такі як RISC і CISC, мають різну енергоефективність при обробці команд. Зокрема, архітектура RISC вважається більш перспективною та поширеною завдяки оптимізованому виконанню інструкцій.

По-друге, значення має тактова частота - її збільшення підвищує продуктивність, але одночасно й енергоспоживання. Наприклад, у мікроконтролерах ATmega зменшення частоти з 16 МГц до 1 МГц дозволяє суттєво знизити споживання струму.

Також важливу роль відіграють режими роботи: активний, Sleep, Stop і Standby мають різний рівень енергоспоживання.

Крім того, ефективність програмного коду прямо впливає на час перебування мікроконтролера в активному стані - чим менше зайвих циклів, таймерів і затримок, тим краще, особливо для пристроїв із принципом “прокинувся-виконав-заснув”.

Нарешті, додаткові периферійні пристрої, як-от сенсори, дисплеї чи комунікаційні модулі, теж можуть істотно збільшувати загальне енергоспоживання системи.

Для кращого розуміння особливостей енергоспоживання різних мікроконтролерів важливо провести їх порівняльний аналіз.

Мікроконтролери Arduino Uno, Arduino Nano та STM32 є популярними прикладами, які демонструють різний рівень продуктивності, можливостей та енергоспоживання.

У таблиці 1.1 наведено порівняння енергоспоживання мікроконтролерів.

Таблиця 1.1 Порівняння енергоспоживання мікроконтролерів.

Параметр	Arduino Uno	Arduino Nano	STM32
Архітектура	8-біт AVR	8-біт AVR	32-біт ARM Cortex-M3
Тактова частота	16 МГц	16 МГц	До 72 МГц
Напруга живлення	5В	5В	3.3В
Енергоспоживання (активний режим)	~50 мА	~19 мА	~30 мА
Енергоспоживання (режим сну)	~0.5 мА	~0.2 мА	~0.2 мА

Як видно з таблиці 1.1, що Arduino Uno - найменш енергоефективний серед порівнюваних мікроконтролерів, але добре підходить для навчальних і простих проєктів. Arduino Nano - компактний і більш енергоефективний варіант з тими ж можливостями, що і Uno. STM32 - оптимальний вибір для проєктів, що вимагають високої продуктивності та мінімального енергоспоживання.

Енергоспоживання мікроконтролерів визначається інтегралом потужності за часом роботи пристрою: де - спожита енергія, - миттєва потужність у часі, - час виконання операцій. Таким чином, для зменшення енергоспоживання необхідно враховувати як потужність, так і час виконання завдань[12].

У статті Ниа та співавторів [13] акцентується увага на тому, що скорочення часу виконання операцій може бути настільки ж ефективним, як і зниження потужності.

Наприклад, збільшення тактової частоти процесора може зменшити час виконання задачі, що дозволяє економити енергію навіть за умов збільшення миттєвої потужності.

Енергоспоживання мікроконтролера складається з двох основних компонентів: динамічного та статичного.

Динамічне енергоспоживання виникає під час перемикання транзисторів у цифрових схемах і залежить від частоти роботи та квадрата робочої напруги - чим вони вищі, тим більше енергії споживається.

Статичне енергоспоживання, натомість, проявляється у стані спокою й зумовлене витоком струму в напівпровідникових елементах, навіть коли мікроконтролер не виконує жодних операцій.

Енергоспоживання мікроконтролерів є важливим показником, що визначає їхню ефективність та тривалість роботи, особливо в умовах автономного живлення від батарей [14].

Для оцінки цього показника застосовуються різноманітні теоретичні моделі та практичні методи вимірювання. Енергоспоживання визначається інтегралом потужності  $P(t)$  в часі:

$$E = \int_{t_0}^{t_1} P(t)dt, \quad (1.1)$$

де  $E$  - загальна енергія, вимірюється в джоулях (J);

$P(t)$  - миттєва потужність, яка змінюється з часом, вимірюється у ватах (W);

$t_0, t_1$  - початковий і кінцевий час спостереження.

Такі чинники як напруга та струм, тактова частота, різні режими роботи мікроконтролера та периферійні пристрої впливають на енергоспоживання мікроконтролера.

Наприклад, чим вища тактова частота, тим більше енергії споживається під час виконання задач, але час виконання скорочується. А щодо напруги і струму то зниження напруги/струму в режимах збереження енергії значно впливає на загальне енергоспоживання [16].

Розуміння того, як енергоспоживання залежить від таких параметрів, як напруга, струм, тактова частота та режими роботи, дозволяє розробникам створювати пристрої, які мають мінімальний вплив на джерела живлення, зокрема на батареї [16].

Важливими для енергоспоживання також є периферійні пристрої, які працюють незалежно від центрального процесора.

У статті [30] показано, що периферія може споживати суттєву частку енергії, навіть якщо центральний процесор знаходиться у режимі низького енергоспоживання.

Мікроконтролери зазвичай програмують мовою асемблера або C, хоча доступні компілятори для інших мов, а також вбудовані інтерпретатори BASIC і Forth.

Для налагодження програм використовують програмні симулятори (спеціальні програми для ПК, які імітують роботу мікроконтролера), внутрішньосхемні емулятори (електронні пристрої, які імітують мікроконтролер і замість нього можуть бути підключені до вбудованого пристрою, що розробляється), інтерфейс JTAG використовується інтерфейс JTAG.

Більшість сучасних мікроконтролерів підтримують кілька режимів роботи, що дає змогу оптимізувати енергоспоживання залежно від поточних завдань – активний режим, режим сну та режим глибокого сну.

У активному режимі спостерігається максимальне енергоспоживання, оскільки процесор активно виконує обчислення та обробку даних.

У режимах сну споживання значно зменшується - процесор перебуває в очікуванні або виконує мінімум операцій, залишаючись готовим до пробудження.

Найнижчий рівень енергоспоживання забезпечується в режимі глибокого сну (deep sleep), коли відключається більшість підсистем, а активними

залишаються лише найнеобхідніші компоненти для збереження стану або реагування на зовнішні події..

У статті [50] описано режими зниженого енергоспоживання для AVR-мікроконтролерів, зокрема використання внутрішніх RC-генераторів і можливість програмного переведення в режим сну для економії енергії.

У таблиці 1.2 наведено режими роботи мікроконтролерів

Таблиця 1.2 Режими роботи мікроконтролерів

Режим роботи	Споживана енергія (мВт)	Час реакції (мс)
Активний	100	0.1
Sleep	10	1
Deep Sleep	1	5

На рисунку 1.3 відображено AVR-мікроконтролер.

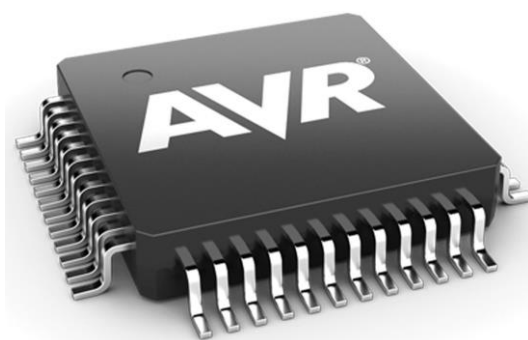


Рисунок 1.3 - AVR-мікроконтролер

Дані показують, що використання AVR-мікроконтролерів у режимі deep sleep дозволяє знизити енергоспоживання до 90% порівняно з активним режимом.

Приклад 1. Зниження тактової частоти у п'ять разів: у цьому випадку споживання зменшується приблизно на 50%. Це пояснюється зменшенням динамічного енергоспоживання, адже швидкість перемикавання транзисторів безпосередньо залежить від тактової частоти. Отже, при меншій частоті енергетичні втрати зменшуються.

Приклад 2. Перехід у режим *deep sleep*: у цьому режимі вимикаються всі периферійні компоненти, окрім найнеобхідніших. Внутрішній RC-генератор підтримує лише базові функції, що дозволяє знизити енергоспоживання до 10% від базового рівня активного режиму. Такий підхід ідеально підходить для пристроїв, які мають працювати тривалий час від батареї.

Отже, застосування енергозберігаючих режимів дозволяє значно продовжити час автономної роботи пристроїв.

Вибір відповідної тактової частоти та режиму роботи дозволяє оптимізувати баланс між продуктивністю та енергоспоживанням. А також зменшення енергоспоживання сприяє екологічній ефективності, оскільки потреба у заміні батарей або витратах енергії знижується [51].

У дослідженні Fezari, Mohamed аналізується енергоспоживання мікроконтролерів Arduino Uno та Nano у різних режимах роботи. Встановлено, що правильний вибір режиму роботи та налаштування тактової частоти може значно знизити споживання енергії, що особливо важливо для пристроїв із живленням від батарей [59].

В результаті дослідження мікроконтролера в активному режимі Arduino Uno споживає в середньому 50 мА при тактовій частоті 16 МГц. Arduino Nano показує трохи нижчий рівень споживання через меншу кількість активних компонентів.

У режимі сну обидва мікроконтролери знижують енергоспоживання на 70-80%, вимикаючи непотрібні периферійні пристрої. Arduino Nano досягає 5 мА у режимі *standby* [59].

Щодо режиму *deep sleep*: Arduino Nano споживає лише 0,1 мА, коли активним залишається лише RTC (Real-Time Clock). Arduino Uno показує вищі показники (0,5 мА) через складнішу архітектуру [59].

Таким чином, поняття енергоспоживання мікроконтролерів охоплює різні аспекти: архітектурні особливості, режими роботи, вибір оптимальної тактової частоти та використання технологій енергозбереження.

У наступному підрозділі буде розглянуто методи вимірювання енергоспоживання, які дозволяють оцінити ефективність використання мікроконтролерів у конкретних застосуваннях.

## 1.2 Методи вимірювання енергоспоживання

Енергоспоживання мікроконтролера є важливим параметром, який визначає час автономної роботи пристрою. Для оцінки енергоспоживання використовуються різні методи, які вимірюють як загальну споживану енергію, так і миттєву потужність.

Традиційні підходи до вимірювання енергоспоживання:

1) вимірювання за допомогою мультиметра: мультиметри є одним з найпоширеніших інструментів для вимірювання струму та напруги. Вимірювання проводиться шляхом підключення мультиметра до ланцюга живлення мікроконтролера.

На основі отриманих значень напруги і струму розраховується потужність за формулою: цей метод простий у використанні, але не дає даних про миттєву потужність або зміну енергоспоживання в часі;

2) логічний аналізатор із датчиком струму: більш точні вимірювання проводяться за допомогою спеціалізованих інструментів, таких як логічні аналізатори з датчиками струму. Вони реєструють зміни струму в режимі реального часу і надають дані для аналізу динаміки енергоспоживання. Це важливо для оцінки ефективності енергозберігаючих режимів;

3) програмні інструменти: середовища розробки, такі як AVR Studio, надають інструменти для моделювання енергоспоживання мікроконтролерів. Ці симулятори можуть допомогти на етапі проектування системи, оскільки вони дозволяють оцінити енергоспоживання без фізичного обладнання. Наприклад,

симулятори Proteus можуть аналізувати енергоспоживання в різних режимах роботи;

4) використання спеціалізованих вимірювальних модулів: сучасні рішення, такі як мікроконтролерні модулі з інтегрованими датчиками енергії (наприклад, INA219), можуть вимірювати струм, напругу та потужність з високою точністю. Ці модулі підключаються до мікроконтролера через інтерфейс I2C або SPI і дозволяють здійснювати безперервний моніторинг енергоспоживання.

У таблиці 1.3 наведено порівняння цих методів.

Таблиця 1.3. Порівняння традиційних методів вимірювання

Метод	Переваги	Недоліки
Мультиметр	Простота використання; доступність для початківців; дешевизна.	Відсутність даних про динаміку; потреба ручних розрахунків.
Логічний аналізатор	Висока точність вимірювань; можливість аналізу динамічних процесів; автоматизація збереження даних.	Висока вартість обладнання; складність налаштування та використання для новачків.
Програмні симуляції	Відсутність потреби у фізичному обладнанні; швидка оцінка на етапі проектування; візуалізація енергоспоживання.	Обмежена точність результатів; неможливість оцінити реальні фізичні фактори.
Спеціалізовані модулі	Висока точність вимірювань у реальному часі; легка інтеграція з іншими компонентами системи; зручність у тривалих моніторингах.	Висока вартість; необхідність додаткових налаштувань; залежність від програмного забезпечення.

Мікроконтролери є дуже складними пристроями у визначенні енергоспоживання. Це означає, що миттєве енергоспоживання під час роботи дуже високе, в кілька сотень разів вище, ніж енергоспоживання в стані спокою (в статичному режимі) [26].

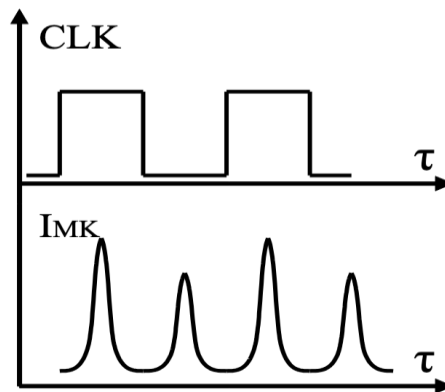


Рисунок 1.4 - Зміна миттєвого струму споживання мікроконтролера у відповідь на тактові імпульси

Тому звичайні методи вимірювання енергоспоживання не дуже точні при вимірюванні струму споживання мікроконтролера. При підключенні цифрового амперметра між джерелом живлення ланцюга живлення мікроконтролера і конденсатором вносяться похибки через неконтрольований обмін енергією між мікроконтролером і цим конденсатором.

Оцінка показала, що зміна напруги 100 мкВ на звичайному конденсаторі в ланцюзі живлення 470 мкФ (для цього роздільна здатність амперметра повинна бути 0,002%) недостатня для 20... 50, що відповідає енергії для виконання 50 команд [28].

Якщо цифровий амперметр розмістити безпосередньо між конденсатором фільтра живлення та мікроконтролером у ланцюгу живлення, він може суттєво вплинути на роботу пристрою через внесення паразитної індуктивності. Навіть при використанні сучасних малогабаритних вимірювальних приладів, внутрішні елементи амперметра можуть мати індуктивні властивості, які формують

швидкоплинні коливання або перешкоди у критичних точках схеми живлення мікроконтролера.

Оскільки більшість мікроконтролерів чутливі до стабільності та чистоти живлення, навіть незначна зміна в електричних параметрах ланцюга може призвести до збоїв у роботі, порушень виконання інструкцій або повної нестабільності в роботі програми.

Тому при побудові систем вимірювання енергоспоживання мікроконтролера необхідно уникати безпосереднього включення амперметра у розрив ланцюга живлення. Натомість слід застосовувати альтернативні підходи, які дозволяють неінвазивно або мінімально інвазивно оцінити параметри струму.

Одним із таких підходів є перетворення струму, що споживається мікроконтролером, у пропорційну напругу за допомогою шунтів або струмового дзеркала. Як було зазначено вище, основним елементом системи вимірювання енергії мікроконтролера, що споживається однією командою або міжкомандним переходом, є схема, яка перетворює струм, що споживається мікроконтролером, у напругу.

Найбільш досконалим є метод вимірювання миттєвої потужності мікроконтролера, запропонований у [26]. Схема пристрою, який реалізує цей метод, подана на риунку 1.5

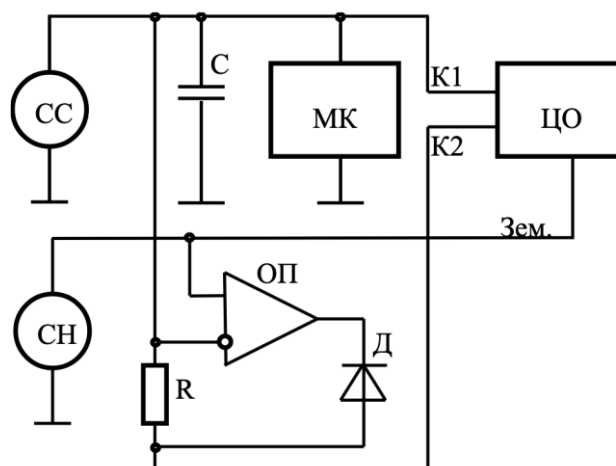


Рисунок 1.5 - Структурно-принципова схема пристрою вимірювання миттєвого струму споживання мікроконтролера

Основна ідея цього методу вимірювання миттєвого струму споживання мікроконтролера полягає, по-перше, у використанні конденсатора  $C$  в ланцюзі живлення мікроконтролера як вимірювального перетворювача для перетворення струму споживання мікроконтролера в напругу, а по-друге, у живлення від джерела струму стабілізатора, а не від джерела напруги [26].

Основними перевагами схеми, поданої на рисунку 1.5 є: робота досліджуваного мікроконтролера у штатному (нормальному) режимі. Це означає, що у колі живлення мікроконтролера є високочастотний конденсатор. Така схема виключає похибки вимірювання, викликані значними змінами напруги живлення мікроконтролера, а головне - його збоями.

Ще однією перевагою є те, що схема дає можливість строгого контролю балансу струмів, які протікають у колі живлення мікроконтролера.

І також є всі необхідні для роботи схеми зміни напруги на мікроконтролері не перевищують 50 мВ [3], тобто 1% від напруги живлення мікроконтролера.

Як видно з перелічених вище переваг схема забезпечує високу точність вимірювання та обчислення миттєвого струму та миттєвої потужності споживання при коректній експлуатації мікроконтролера (тобто при його роботі у нормальному, штатному режимі).

Однак цей метод має кілька недоліків. Перший з них - низька завадостійкість до звичайних шумів. Тому завадостійкі методи вимірювання енергоспоживання мікроконтролерів не повинні базуватися на цифрових осцилографах.

Другим недоліком інформаційно-вимірювальних систем є накопичення похибок при дослідженні енергоспоживання мікроконтролера при виконанні послідовності інструкцій. Ця проблема виникає при дослідженні енергоспоживання програми або фрагмента програми.

Слід відзначити, що метою такого дослідження наборів інструкцій, фрагментів програм та програм у цілому є оцінка результатів оптимізації програмного забезпечення. Тому таке накопичення похибки не дає можливості достовірно оцінити результати оптимізації [26-28].

### 1.3 Постановка задачі

Поставлена мета досягається розв'язанням таких основних завдань:

- 1) провести аналіз існуючих підходів до зниження енергоспоживання мікроконтролерів та методів його вимірювання;
- 2) вивчити архітектуру мікроконтролера ATmega328P та режими його роботи (Active, Sleep, Deep Sleep);
- 3) розробити концептуальну модель симуляції енергоспоживання в середовищі Proteus;
- 4) реалізувати схему симуляції з урахуванням різних режимів роботи та налаштувань тактової частоти;
- 5) провести експериментальні дослідження для аналізу та порівняння енергоспоживання в різних режимах;

### 1.4 Висновки до першого розділу

Було здійснено комплексний аналіз теоретичних аспектів енергоспоживання мікроконтролерів, що є критично важливим у контексті побудови автономних пристроїв. Розглянуто ключові фактори, які впливають на споживання енергії, включаючи архітектуру, тактову частоту, режими роботи та використання периферійних модулів.

Проаналізовано переваги застосування енергозберігаючих режимів. Показано, що значне зниження енергоспоживання може бути досягнуто за рахунок оптимізації програмного коду та використання відповідних режимів енергозбереження.

Розглянуто методи вимірювання енергоспоживання, що включають як апаратні засоби, так і програмні симулятори. Середовище Proteus, дозволяє здійснювати попередню оцінку енергоспоживання мікроконтролера з мінімальними витратами та без створення апаратної реалізації.

## 2 МОДЕЛЮВАННЯ AVR-МІКРОКОНТРОЛЕРІВ

У сучасному світі, де енергоефективність стає все більш важливою, моделювання та оцінка енергоспоживання вбудованих систем займають центральне місце в процесі проектування.

Зростаюча залежність від портативних пристроїв, Інтернету речей та інших цифрових технологій ставить перед інженерами завдання оптимізації енергоспоживання на всіх рівнях - від архітектури до окремих компонентів.

Для ефективного управління енергетичними ресурсами необхідно детально вивчити джерела живлення та їх взаємозалежності. Це включає аналіз різних режимів роботи мікроконтролерів, таких як активний, сплячий та режими зниженого споживання, а також вивчення впливу на енергоспоживання різних параметрів, таких як тактова частота і напруга живлення.

Оптимізація енергоспоживання у цифрових схемах є критично важливою метою проектування. Вона полягає у зменшенні споживаної потужності при збереженні або навіть покращенні функціональності та продуктивності системи.

Це має безпосередній вплив на тривалість роботи батареї, зниження експлуатаційних витрат і зменшення тепловиділення, що є особливо важливим для портативних пристроїв.

У цьому розділі детально розглядаються моделі енергоспоживання AVR-мікроконтролерів, аналізуючи їхню архітектуру, режими роботи та особливості периферійних модулів. Також обговорюються методи оцінки енергоспоживання, включаючи динамічні та статичні компоненти, а також вплив різних факторів на загальне споживання енергії.

Це дозволяє зрозуміти, як ефективно управляти енергетичними ресурсами в рамках вбудованих систем і досягати оптимального балансу між продуктивністю та енергоефективністю.

Таким чином, моделювання енергоспоживання не лише сприяє підвищенню ефективності роботи систем, але й допомагає розробникам створювати інноваційні рішення для сучасних викликів у сфері електроніки та автоматизації.

## 2.1 Модель енергоспоживання AVR-мікроконтролерів

Перші мікроконтролери AVR були створені в дослідницькому центрі Atmel у Норвегії групою інженерів, до складу якої входили Альф Боген та Вегард Воллан. Назва “AVR” виникла внаслідок ініціалів їх імен та посилання на архітектуру RISC. Перший мікроконтролер серії AVR, AT90S1200, був випущений у 1996-1997 роках.

Сьогодні мікроконтролери AVR вважаються одними з найкращих серед 8-бітних контролерів з архітектурою RISC за показниками «ціна - швидкодія-енергоспоживання». Обсяг їх продажу подвоюється щорічно.

Сфери застосування мікроконтролерів AVR надзвичайно різноманітні - від простих пристроїв до складних систем збору та обробки інформації, які використовуються в побуті та промисловості.

Широко застосовуються в енергоефективних проєктах завдяки їх здатності працювати в режимах зниженого енергоспоживання, що робить їх ідеальними для пристроїв, які потребують тривалої автономної роботи. Розглянемо декілька прикладів застосування AVR-мікроконтролерів у проєктах, де ефективно використання енергії є пріоритетом.

В епоху IoT автономні сенсори використовуються для збору різноманітних даних про навколишнє середовище. Застосування AVR-мікроконтролерів у таких пристроях допомагає досягти значної економії енергії.

Режими сну дозволяють мікроконтролеру переходити в стан мінімального енергоспоживання, коли активна робота не потрібна.

Циклічне пробудження та передача даних дозволяють оптимізувати енергоспоживання, активуючи сенсор лише на короткий час для збору та передачі інформації.

Можливість використання альтернативних джерел енергії, таких як сонячні панелі або термоелектричні генератори, робить такі сенсори повністю автономними та екологічними.

В автомобільній промисловості, де надійність і енергоефективність є критичними. Бортові системи діагностики (OBD-II) використовують AVR для моніторингу та діагностики стану автомобіля, при цьому мінімізуючи споживання енергії. Контролери освітлення можуть працювати в режимах зниженого енергоспоживання, автоматично регулюючи яскравість світла залежно від умов освітлення.

Датчики контролю тиску в шинах (TPMS) функціонують на батарейках з тривалим терміном служби завдяки енергоефективності AVR-мікроконтролерів.

У портативних медичних пристроях, де важлива тривала робота від батареї, AVR-мікроконтролери використовуються для забезпечення максимальної енергоефективності. Кардіомонітори та глюкометри використовують енергоефективні режими для зменшення частоти заряджання батарей.

Імплантовані сенсори, які вимагають мінімального споживання енергії для тривалої роботи в організмі, також використовують AVR-мікроконтролери.

Реалізація можливості пробудження мікроконтролера лише при необхідності зчитування показників дозволяє значно зменшити загальне енергоспоживання

AVR-мікроконтролери, завдяки своїм вбудованим інтерфейсам UART, SPI та I2C, ідеально підходять для бездротових систем зв'язку. Модулі віддаленого контролю з живленням від батарей, які використовуються для моніторингу та керування різними пристроями на відстані. Bluetooth Low Energy (BLE) пристрої, що забезпечують енергозберігаючі бездротові комунікації для передачі даних на невеликі відстані.

Радіочастотні системи моніторингу (RFID, ZigBee), які використовуються для ідентифікації та відстеження об'єктів з мінімальним енергоспоживанням.

Завдяки цим можливостям, AVR-мікроконтролери стають оптимальним вибором для проєктів, де ключовим фактором є енергоефективність, забезпечуючи тривалу та надійну роботу пристроїв у різноманітних умовах

Перший офіційний каталог мікроконтролерів компанії Atmel був випущений у 1997 році і містив мікроконтролери першого сімейства Classic. У

1999 році в другому випуску каталогу вже були представлені дані про мікроконтролери трьох сімейств: Classic, Mega та Tiny.

Сімейства мікроконтролерів AVR мають ряд спільних характеристик:

1) це 8-бітні (також доступні 16- і 32-бітні) RISC мікроконтролери загального призначення, що реалізують Гарвардську архітектуру з фізично та логічно розділеними шинами даних і адресними просторами для програмної та даних пам'яті;

2) оснащені вбудованою Flash-пам'яттю для програм, статичною SRAM для даних та енергонезалежною EEPROM. Збільшена ємність вбудованої пам'яті дозволяє використовувати мову високого рівня для розробки прикладних програм;

3) продуктивність мікроконтролера досягає 1 MIPS при частоті 1 МГц.;

4) мікроконтролери містять 32 регістри загального призначення, які формують файловий регістр. Ці регістри використовуються для виконання арифметичних і логічних операцій, що суттєво підвищує ефективність виконуваних програм;

5) AVR мікроконтролери реалізують однорівневий конвеєр для роботи з пам'яттю програм, що дозволяє одночасно виконувати поточну команду та вибирати наступну;

6) передбачена можливість як зовнішнього, так і внутрішнього програмування мікроконтролера;

7) пам'ять програм поділяється на два сектори: для прикладної програми (application sector) та для програми завантажувача (boot sector). Є можливість перерозподілу ємності цих сегментів, а сектор завантаження забезпечує функцію самопрограмування;

8) наявність спеціальних засобів (фюзесів) для програмування структури мікроконтролера.

До ресурсів мікроконтролерів зазвичай включають інформацію про фізичні характеристики корпусів, діапазони напруги живлення та робочі температури,

максимальну частоту роботи, кількість розрядів шини даних, а також обсяги пам'яті для програм, статичної та енергонезалежної пам'яті.

Також зазначається кількість ліній вводу/виводу та наводиться опис вбудованих контролерів, таких як таймери-лічильники, контролери RTC для ємнісних сенсорних панелей, а також периферійні контролери USART, SPI, TWI, USB, PWM, Ethernet і CAN.

У таблиці 1 відображені максимальні діапазони по ресурсам всіх AVR мікроконтролерів.

Таблиця 2.1. Максимальні діапазони для AVR-мікроконтролерів

№	Назва ресурсу	Значення, діапазон
1	Розрядність процесорного ядра (CPU)	8 / 16 / 32
2	Кількість виводів корпусу мікроконтролера	6 - 144
3	Максимальна робоча частота	1-84 МГц
4	Об'єм вбудованої пам'яті програм FLASH	0,5 - 512 кБайт
5	Об'єм статичної пам'яті даних SRAM	0,03 - 128 кБайт
6	Об'єм енергонезалежної пам'яті даних EEPROM	0 - 4096 Байт
7	Максимальна кількість ліній вводу/виводу	4 - 123
8	Напруга живлення	0,7 - 25 В
9	Діапазон робочих температур	-40 - +150 °С
10	Кількість каналів Touch Channels (контролер RTC)	0 - 56
11	Кількість таймерів лічильників (8 / 16 біт)	1 - 10
12	Сторожовий таймер (WDT)	1
13	Незалежний RTC генератор 32кГц	0 - 1
14	Аналоговий компаратор	1 - 8
15	Температурний сенсор	0 - 1
16	Розрядність аналогово - цифрового перетворювача	8 - 10 - 12

Продовження таблиці 2.1

17	Кількість каналів аналогово-цифрового перетворювача	4 - 28
18	Цифро аналоговий перетворювач DAC	0 - 4
19	Контролер UART	0 - 8
20	Контролер SPI	0 - 12
21	Контролер TWI	0 - 4
22	Контролер Ethernet	0 - 1
23	Контролер USB	0 -Device - OTG
24	Контролер CAN	0 - 2
25	Контролер PWM	0 - 36

AVR-мікроконтролери є одними з найбільш популярних завдяки своїй енергоефективності, простоті використання та широкому спектру застосувань. Модель енергоспоживання цих мікроконтролерів базується на аналізі їхньої архітектури, режимів роботи та особливостей периферійних модулів.

Основними параметрами, що впливають на енергоспоживання AVR-мікроконтролерів, є архітектура ядра, режими роботи, тактова частота та напруга живлення.

Використання 8-бітної архітектури забезпечує достатню продуктивність для виконання базових завдань при низькому енергоспоживанні. Ядро AVR виконує більшість інструкцій за мінімальну кількість тактових циклів, що дозволяє зменшити загальні енерговитрати.

Крім того, мікроконтролери цієї серії підтримують різні режими роботи - активний, сну та глибокого сну - що дає змогу гнучко керувати споживанням енергії відповідно до умов роботи системи.

Важливу роль також відіграє тактова частота: її збільшення підвищує продуктивність, однак призводить до зростання енергоспоживання, тоді як зниження - суттєво його скорочує.

Ще одним ключовим фактором є напруга живлення: AVR-мікроконтролери можуть працювати в широкому діапазоні від 1.8 В до 5.5 В. Зменшення напруги до мінімально допустимого рівня (наприклад, 3.3 В або 1.8 В замість 5 В) дозволяє значно зекономити енергію завдяки квадратичній залежності споживання струму від напруги, не втрачаючи при цьому функціональності для базових завдань.

Енергоспоживання мікроконтролерів можна розділити на два основні типи: динамічне та статичне енергоспоживання.

Динамічне енергоспоживання - це споживання, яке виникає внаслідок перемикань сигналів у схемах і залежить від тактової частоти, ємності навантаження та напруги живлення.

Динамічна потужність, зображена на рисунку 2.1, складається з потужності, що споживається схемами тактового генератора, потужності, що витрачається на з'єднання між блоками (де ємність навантаження залежить від типів і довжини проводів), і з логічної потужності, яка споживається функціональними блоками та пам'яттю.

Динамічна потужність може бути розрахована за наступною формулою (в ватах):

$$P = \alpha C_l V_{dd}^2 f , \quad (2.1)$$

де,  $\alpha$  - середня кількість переходів з 0 в 1 або з 1 в 0 за один тактовий цикл;

$C_l$  - ємність навантаження;

$V_{dd}$  - напруга живлення;

$f$  - частота такту

Ця формула дозволяє оцінити динамічну потужність, що є критично важливим для проектування енергоефективних систем на базі мікроконтролерів.

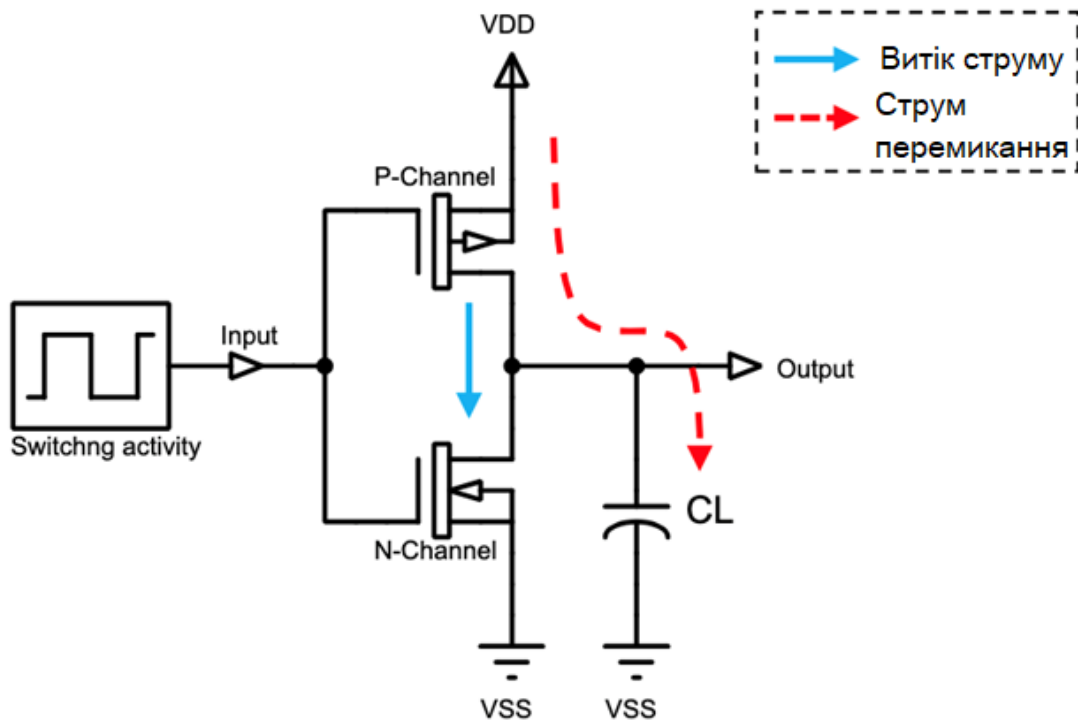


Рисунок 2.1 - Динамічна потужність

Статичне енергоспоживання - це споживання зумовлене витоковими струмами через транзистори навіть у режимах простою. Цей компонент є критично важливим для сучасних мікроконтролерів з малими техпроцесами.

Статична потужність, як показано на рисунку 2.2, виникає внаслідок витоків струму.

Для конкретного апаратного компонента вона залежить від стану джерела живлення (включене чи вимкнене, а також від напруги) і не залежить від виконуваних обчислень. Витік струму має тенденцію збільшуватися при зменшенні розміру транзисторів.

Оскільки підвищення температури кремнію збільшує його провідність, статична потужність також зростає з підвищенням температури.

Статичну потужність можна розрахувати за наступною формулою (в ватах):

$$P_{static} = V_{dd} \times I_{leakage} \quad (2.2)$$

де  $I_{leakage}$  - витік струму;

$V_{dd}$  - напруга живлення.

Ця формула дозволяє оцінити статичну потужність, яка є важливим параметром при проектуванні енергоефективних систем, оскільки вона може суттєво вплинути на загальне споживання енергії пристрою.

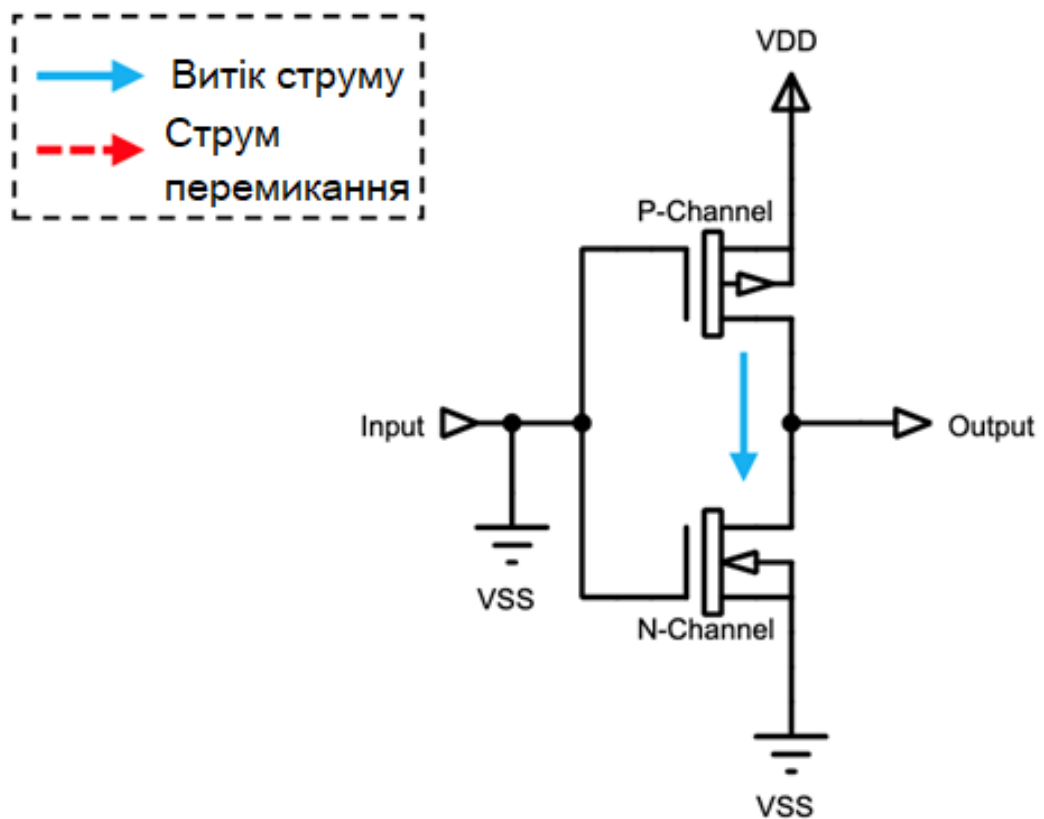


Рисунок 2.2 - Статична потужність

Теоретична модель енергоспоживання AVR-мікроконтролерів дозволяє проектувальникам оцінити ефективність використання енергії в системах, що базуються на цих мікроконтролерах.

Розуміння компонентів споживаної потужності допомагає оптимізувати проекти для досягнення максимальної енергоефективності.

Ця модель є важливим інструментом для розробників, які прагнуть зменшити енергоспоживання своїх пристроїв, особливо в умовах обмежених ресурсів живлення.

Для AVR-мікроконтролерів важливо дотримуватися балансу між цими двома компонентами. Наприклад, у режимі Active переважає динамічне енергоспоживання, тоді як у режимах Standby або Power-down основну роль відіграє статичний компонент.

Програмна модель AVR мікроконтролера, зображена на рисунку 2.3, представляє собою сукупність програмно доступних ресурсів, які використовуються наступним чином:

- 1) FLASH пам'ять програм: використовується для збереження робочої програми та програми завантаження мікроконтролера, розділена на два сектори: Application (для основної програми) та Boot (для програми завантаження);
- 2) SRAM: внутрішня та зовнішня статична пам'ять використовується для збереження динамічних змінних під час виконання програми;
- 3) EEPROM пам'ять: енергонезалежна пам'ять даних, призначена для зберігання енергонезалежних змінних, які потрібно зберігати навіть після вимкнення живлення;
- 4) регістра вводу/виводу: ці регістри забезпечують управління вбудованими ресурсами мікроконтролера, дозволяючи здійснювати взаємодію з зовнішніми пристроями;
- 5) регістровий файл: використовується для обробки інформації в арифметично-логічному пристрої (АЛП), що дозволяє виконувати різноманітні обчислення та маніпуляції з даними.

Ця програмна модель забезпечує ефективне управління ресурсами мікроконтролера і дозволяє реалізувати широкий спектр функцій у різних застосуваннях.

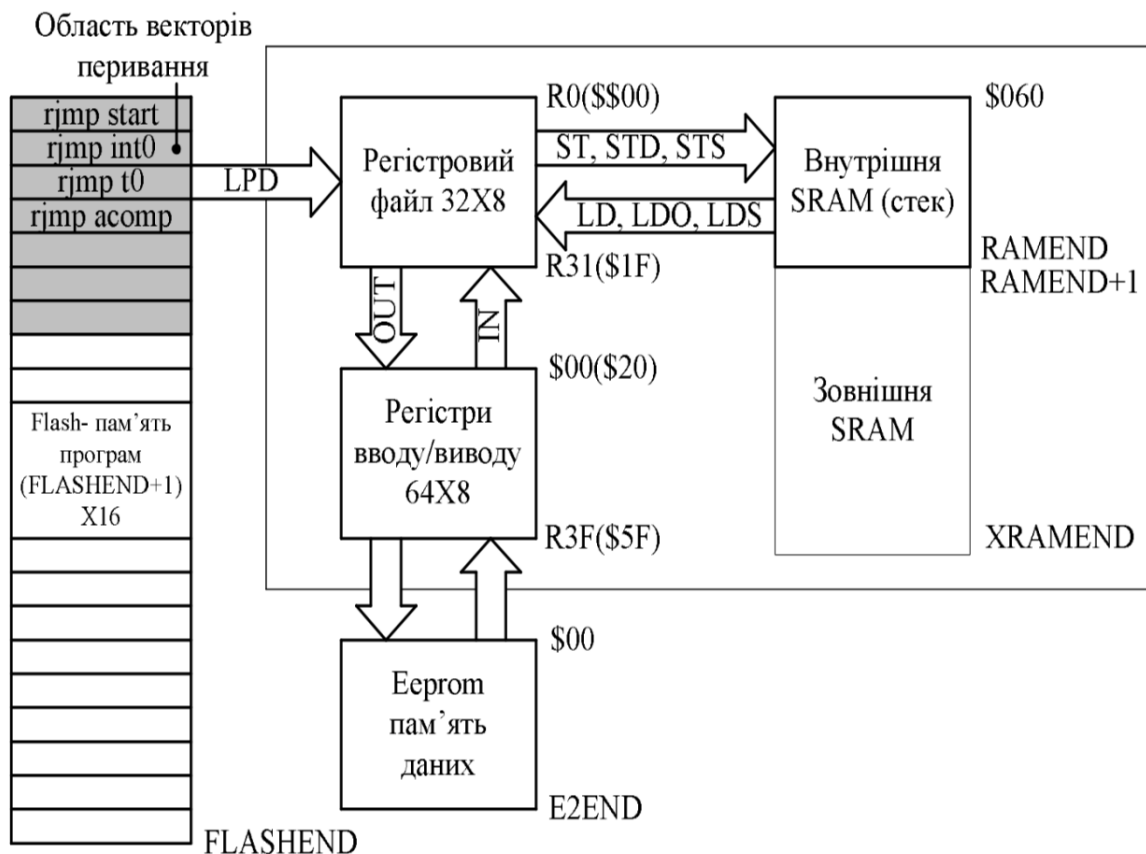


Рисунок 2.3 - Програмна модель AVR-мікроконтролерів

Енергоспоживання мікроконтролерів визначається низкою факторів, кожен з яких має свій вплив на загальну ефективність системи. Одним із ключових параметрів є частота роботи ядра: зі зростанням тактової частоти енергоспоживання збільшується майже лінійно, оскільки кількість перемикачів у цифрових схемах за одиницю часу також зростає. Для зменшення втрат енергії доцільно використовувати технології динамічного масштабування напруги та частоти (DVFS), що дозволяє знижувати частоту та напругу під час періодів низького навантаження без істотної втрати продуктивності.

Важливу роль відіграє і напруга живлення: вона має квадратичний вплив на динамічне споживання, тому її зниження, наприклад із 5 В до 3.3 В у AVR-мікроконтролерах, може майже вдвічі зменшити енергоспоживання. Проте при цьому важливо забезпечити стабільну роботу периферії та систем у цілому.

Значну частку енергії може споживати і периферія, зокрема UART, SPI, I2C - тому доцільно відключати непотрібні модулі в режимах зниженого енергоспоживання або використовувати драйвери з енергозберігаючими функціями.

Програмна оптимізація також суттєво впливає на енерговитрати: використання ефективних алгоритмів, уникнення зайвих циклів, оптимізація взаємодії з периферією (наприклад, через переривання замість опитування) дозволяє знизити навантаження на ядро та перевести мікроконтролер у стан очікування.

Крім того, слід враховувати вплив температури середовища - при її підвищенні зростають струми витоку, що збільшує статичне енергоспоживання. Щоб зменшити цей ефект, застосовують компоненти з низьким тепловиділенням або системи охолодження. Аналіз факторів впливу на енергоспоживання показує, що ефективне управління параметрами роботи мікроконтролера дозволяє суттєво знизити загальні витрати енергії.

Використання таких технологій, як DVFS, оптимізація периферійних модулів і програмного забезпечення, а також адаптація до умов роботи (наприклад, зниження напруги живлення), є ключовими інструментами для досягнення енергоефективності. Такий підхід особливо актуальний для автономних систем, де обмеження ресурсів живлення є критичним фактором.

## 2.2 Моделювання для AVR-мікроконтролерів за параметром енергоспоживання

Моделювання енергоспоживання вбудованих систем - це процес оцінки енергоспоживання системи, яка є комп'ютерною системою з визначеною функцією в рамках більшої електричної системи. Це моделювання виконується ще до фізичного створення пристрою, що дозволяє прогнозувати, скільки енергії буде споживати система за різних сценаріїв, таких як різні робочі навантаження або умови експлуатації.

Основною метою моделювання є виявлення потенційних проблем з енергоспоживанням і оптимізація роботи системи в межах заданого бюджету енергії.

Енергоспоживання мікроконтролерів визначається кількома ключовими параметрами, серед яких особливе значення мають напруга живлення, частота такту та режими роботи.

Напруга живлення безпосередньо впливає на потужність, яку споживає мікроконтролер: для AVR-пристроїв типовий діапазон становить від 1.8 до 5.5 В, і чим нижча напруга, тим менше енергії споживається, хоча при цьому важливо враховувати стабільність роботи системи.

Частота такту також відіграє важливу роль - вища частота забезпечує більшу продуктивність, але призводить до збільшення струму споживання. Наприклад, ATmega328P може працювати на частотах до 20 МГц, однак така швидкість супроводжується більшими енергозатратами.

Додатково, AVR-мікроконтролери підтримують кілька режимів роботи: активний режим, коли процесор виконує обчислення і споживає найбільше енергії, а також режими сну й зниженого споживання, які дозволяють суттєво зменшити витрати енергії в моменти бездіяльності системи.

Для симуляції енергоспоживання AVR-мікроконтролерів можна використовувати різні програмні засоби, одним з яких є Proteus - це програмне забезпечення дозволяє створювати електронні схеми та симулювати їх роботу в реальному часі, включаючи аналіз споживаної потужності.

Proteus є одним із найпотужніших програмних пакетів для моделювання електронних схем. Цей інструмент дозволяє створювати електронні схеми з використанням AVR-мікроконтролерів та периферійних пристроїв.

Симулювати роботу мікроконтролера в реальному часі, включаючи аналіз споживаної потужності а також спостерігати за змінами в параметрах енергоспоживання залежно від режиму роботи мікроконтролера (наприклад, активний режим, режим сну або Idle).

Proteus також підтримує інтеграцію з кодом, написаним у середовищах розробки, таких як AVR Studio або Arduino IDE, що дозволяє тестувати програму без необхідності завантаження її на фізичний пристрій.

Також популярним є AVR Studio - це середовище розробки для програмування AVR-мікроконтролерів також має можливості налагодження та симуляції. AVR Studio (нині відоме як Microchip Studio) є офіційним середовищем розробки для AVR-мікроконтролерів.

AVR Studio забезпечує емуляцію роботи AVR-мікроконтролерів, включаючи всі порти введення/виведення, таймери, АЦП та інші периферійні модулі. Також підтримує функцію симуляції, яка дозволяє аналізувати виконання програми та її вплив на енергоспоживання мікроконтролера.

Процес роботи з AVR Studio включає створення проекту, написання коду програми, компіляцію та запуск симуляції.

Наприклад, для налаштування симуляції потрібно вибрати мікроконтролер (наприклад, ATmega8535), встановити частоту такту (наприклад, 8 МГц) і запустити емуляцію.

Це дозволяє побачити роботу програми так, ніби вона вже записана у фізичний пристрій.

Для досягнення максимальної точності при моделюванні енергоспоживання рекомендується використовувати обидва інструменти в комбінації. Наприклад: у AVR Studio можна написати програму для мікроконтролера та провести її початкове налагодження.

У Proteus можна змоделювати фізичну схему з підключеними периферійними пристроями та оцінити загальне енергоспоживання системи. Такий підхід дозволяє врахувати як програмні аспекти (алгоритми роботи), так і апаратні (вплив периферії на споживану потужність).

Обидва інструменти - Proteus і AVR Studio - мають свої переваги та недоліки. Proteus є більш універсальним для моделювання електронних схем і симуляції їх роботи в реальному часі, тоді як AVR Studio спеціалізується на програмуванні та налагодженні мікроконтролерів AVR.

Вибір між ними залежить від конкретних потреб проекту: якщо необхідно створити віртуальну модель пристрою - краще використовувати Proteus; якщо ж акцент на написанні та тестуванні коду - оптимальним буде AVR Studio.

У таблиці 2.2 наведено порівняння даних програмних засобів.

Таблиця 2.2 Порівняння інструментів

Інструмент	Особливості	Переваги	Обмеження
Proteus	Моделювання схем і симуляція в реальному часі	Візуалізація роботи схем; підтримка різних периферійних пристроїв	Необхідність ліцензії для повного функціоналу
AVR Studio	Емуляція програмного забезпечення для AVR-мікроконтролерів	Глибока інтеграція з архітектурою AVR; детальне налагодження	Не підтримує моделювання фізичних схем

Процес симуляції енергоспоживання мікроконтролерної системи включає кілька послідовних етапів. Спочатку створюється електронна схема, яка містить мікроконтролер та необхідні периферійні пристрої - це основа для подальшого аналізу. Далі розробляється програмний код, що реалізує алгоритми роботи пристрою та враховує всі режими енергоспоживання, включаючи активний режим, сон та інші стани. Після цього запускається симуляція, яка дозволяє спостерігати за поведінкою системи та змінами у споживанні енергії залежно від обраного режиму. На завершальному етапі виконується збір даних: вимірюються струм і напруга в різні моменти часу, що дає змогу оцінити ефективність роботи пристрою та виявити можливості для подальшої оптимізації.

Моделювання енергоспоживання мікроконтролерних систем може здійснюватися за допомогою різних підходів, серед яких основними є аналітичне та імітаційне моделювання.

Аналітичне моделювання ґрунтується на використанні математичних рівнянь, які дозволяють оцінити споживання енергії на основі архітектури системи та технічних характеристик її компонентів. Такий підхід дає змогу швидко проаналізувати ефективність різних варіантів конфігурації без потреби у складних обчисленнях або повноцінній симуляції.

З іншого боку, імітаційне моделювання передбачає застосування програмного забезпечення для відтворення роботи системи в умовах наближених до реальних. Це дає змогу дослідити енергоспоживання за різних сценаріїв використання, враховуючи зміну навантаження, перехід між режимами роботи та взаємодію з периферією.

Обидва підходи можуть застосовуватись як окремо, так і у поєднанні для отримання більш точної та повної картини енергоспоживання системи.

Для імітації використовуються спеціалізовані програмні інструменти, такі як: Proteus та AVR Studio (Microchip Studio).

Суть процесу полягає у створенні віртуальної моделі мікроконтролера, налаштування параметрів (тактова частота, напруга, периферія) та проведення симуляції для оцінки споживання енергії в різних режимах.

Експериментальний метод - використовується лабораторне обладнання для точних вимірювань: мультиметри для вимірювання напруги й струму, осцилографи для аналізу часових характеристик, спеціалізовані модулі (наприклад, INA219) для реєстрації споживаної енергії.

Емпіричний метод - ґрунтується на зборі статистичних даних про енергоспоживання в реальних умовах. Він дозволяє отримати точні дані про споживання енергії в різних режимах роботи мікроконтролера, таких як Power Down, Idle та Active.

Наприклад, аналіз енергоспоживання ATmega328P у цих режимах може дати цінну інформацію про те, як оптимізувати використання енергії в конкретних застосуваннях.

Цей метод є особливо корисним, оскільки він базується на реальних вимірюваннях, а не на теоретичних розрахунках, що може забезпечити високу точність даних.

Гібридний метод - поєднує аналітичні моделі та результати симуляцій. Він дозволяє уточнити результати симуляції за рахунок експериментальних даних, що робить його особливо ефективним для комплексних систем із багатьма компонентами.

Наприклад, в системах, де використовуються кілька мікроконтролерів і периферійних пристроїв, гібридний підхід може допомогти більш точно оцінити загальне енергоспоживання та виявити можливості для оптимізації.

У таблиці 2.3 наведено порівняння методів моделювання.

Метод	Точність	Швидкість реалізації	Вартість	Приклад використання
Аналітичний	Помірна: залежить від правильності вибору параметрів моделі	Висока: потребує лише математичних розрахунків	Низька: не потребує спеціального обладнання	Початкове планування
Імітаційний	Висока: залежить від точності моделювання в програмі	Середня: потребує часу для налаштування параметрів у симуляторі	Середня: потрібні програмні інструменти	Оцінка роботи коду в різних режимах

Продовження таблиці 2.3

Експериментальний	Дуже висока: базується на реальних вимірюваннях	Низька: вимагає тривалого процесу вимірювань	Висока: потрібне лабораторне обладнання	Тестування готових пристроїв
Емпіричний	Помірна: залежить від кількості зібраних даних	Середня: вимагає аналізу вже існуючих систем	Середня: потрібен доступ до готових пристроїв	Оцінка роботи в реальних умовах
Гібридний	Висока: поєднання симуляції та вимірювань підвищує точність	Середня: потребує часу для інтеграції даних	Висока: комбінація програмного й апаратного забезпечення	Інтеграція складних систем

Моделювання енергоспоживання є критично важливим для вбудованих систем, оскільки воно допомагає забезпечити надійну роботу системи в межах її енергетичних обмежень.

Це також дозволяє виявити потенційні проблеми з енергоспоживанням на ранніх стадіях проектування, що може суттєво знизити витрати на розробку та реалізацію.

Завдяки моделюванню розробники можуть оптимізувати споживання енергії, скорочуючи загальні витрати або забезпечуючи, щоб система працювала в межах певного бюджету енергоспоживання. Це може включати зменшення активного споживання під час роботи системи шляхом вибору енергоефективних компонентів і алгоритмів.

Використання режимів сну та інших стратегій управління живленням для мінімізації енергоспоживання в періоди низької активності. Блоки керування живленням (PMU) відіграють важливу роль в управлінні температурою, відстежуючи та контролюючи температуру критичних компонентів, щоб запобігти перегріву та терморегулюванню;

І також аналіз впливу різних робочих навантажень на загальне споживання енергії, що дозволяє адаптувати систему до різних сценаріїв використання.

Таким чином, моделювання енергоспоживання сприяє не лише підвищенню ефективності роботи вбудованих систем, але й допомагає зберігати ресурси та знижувати витрати на експлуатацію. Зокрема, моделювання енергоспоживання кожного ядра CPU дозволяє регулювати режими енергоживлення, забезпечуючи безперервність сеансу зв'язку при відеоконференціях. Крім того, енергоефективність є критично важливою умовою при проектуванні вбудованої системи, щоб подовжити термін служби батареї та зменшити споживання енергії.

### 2.3 Висновки до даного розділу

Модель енергоспоживання AVR-мікроконтролерів дозволяє прогнозувати споживання енергії залежно від параметрів роботи та зовнішніх умов. Використання інструментів оптимізації, таких як clock gating та PRR, дозволяє суттєво знизити енергоспоживання. Така модель є основою для подальшого аналізу та оптимізації, що особливо важливо для систем із живленням від батарей.

Моделювання енергоспоживання AVR-мікроконтролерів є критично важливим інструментом для розробників, які прагнуть створювати ефективні та економічні рішення.

Завдяки використанню аналітичних та імітаційних моделей можна досягти значного зменшення споживаної потужності, що особливо важливо в умовах обмежених ресурсів живлення.

Для зниження енергоспоживання найкраще використовувати такі режими роботи:

1) режими сну та очікування: використання режимів сну на комп'ютерах, телевізорах та інших електроприладах дозволяє значно знизити споживання енергії, коли пристрої не використовуються. Це дозволяє уникнути витрат енергії в режимі очікування, що може становити до 10-15% від загального споживання;

2) економічні режими для побутових приладів: багато сучасних побутових приладів, таких як пральні та посудомийні машини, мають спеціальні економічні режими. Використання цих режимів під час повного завантаження дозволяє зменшити витрати електроенергії;

3) низький режим продуктивності: у смартфонах і деяких інших пристроях є функція низького режиму продуктивності, яка обмежує швидкість обробки даних і знижує енергоспоживання, не впливаючи на основні функції, такі як дзвінки або доступ до Інтернету;

4) автоматизація системи управління: впровадження "розумних" технологій у будівлях, які автоматично вимикають освітлення та регулюють системи опалення та кондиціонування відповідно до присутності людей, може суттєво знизити споживання електроенергії;

5) зниження напруги живлення: для деяких пристроїв зниження напруги живлення може привести до значної економії енергії без втрати функціональності.

Використання інструментів, таких як Proteus та AVR Studio, дозволяє проводити детальний аналіз енергоспоживання мікроконтролерів без фізичного втручання. Результати моделювання можуть бути використані для оптимізації проектів на основі AVR-мікроконтролерів, що критично важливо для IoT, медичних і портативних пристроїв.

Моделювання енергоспоживання AVR-мікроконтролерів демонструє, що: динамічне енергоспоживання залежить від напруги живлення та тактової частоти.

Статичне енергоспоживання має ключове значення в режимах енергозбереження, а режими Sleep і Deep Sleep значно знижують енергоспоживання, що робить їх оптимальними для автономних систем.

### 3 МЕТОД СИМУЛЯЦІЇ ЕНЕРГОСПОЖИВАННЯ МІКРОКОНТРОЛЕРА

Зростання вимог до ефективності та енергоефективності сучасних мікроконтролерів робить актуальним розвиток методів симуляції їхнього енергоспоживання.

Симуляція дозволяє передбачити поведінку системи під час різних умов роботи, оптимізувати її конструкцію та програмне забезпечення для зниження споживання енергії без фізичної реалізації прототипу.

Розробка й застосування таких моделей мають велике значення для створення з низьким споживанням енергії пристроїв у сфері IoT, носимих технологій тощо, де обмеження джерел живлення є критичним фактором проектування.

Крім того, використання симуляційних моделей дозволяє скоротити час розробки нових продуктів і підвищити їхню надійність шляхом попередньої оцінки впливу різних параметрів на споживання енергії.

У подальшому розвитку мікроелектроніки такі підходи стануть ще більш важливими через зростаючі вимоги до мобільності й автономності пристроїв. Симуляційні моделі також сприяють зниженню витрат на розробку шляхом зменшення кількості необхідних прототипів і експериментів з фізичними зразками.

#### 3.1 Модель симуляції енергоспоживання мікроконтролера

Модель симуляції енергоспоживання мікроконтролера базується на математичних рівняннях, які описують залежність споживання енергії від різних параметрів роботи мікроконтролера.

Основними факторами впливу є частота процесора, режим роботи (активний чи сплячий), а також тип виконуваних операцій.

Математично цю залежність можна виразити наступним чином:

$$E_{total} = P_{active} T_{active} + P_{sleep} T_{sleep} + P_{trans} N_{trans} \quad (3.1)$$

де,  $E_{total}$  - загальне споживання енергії;

$P_{active}$  - потужність, що споживається мікроконтролером у активному режимі;

$P_{sleep}$  - потужність у сплячому режимі;

$P_{trans}$  - потужність під час переходів між режимами;

$T_{active}$  - тривалість перебування у активному режимі;

$T_{sleep}$  - тривалість перебування у сплячому режимі;

$N_{trans}$  - кількість переходів між активним і сплячим станом

Для оцінки енергоспоживання мікроконтролерів найчастіше використовуються наступні математичні моделі: моделі лінійної регресії, нелінійні моделі і моделі динамічного та статичного споживання

Моделі лінійної регресії - описують залежність споживання енергії від параметрів, таких як тактова частота процесора, напруга живлення та активність периферійних пристроїв. Використовуються для прогнозування енергоспоживання та встановлення стандартів у технологічних системах;

Нелінійні моделі - дозволяють точніше описати енергоспоживання мікроконтролерів у складних умовах, таких як температурні зміни, змінне навантаження. Використовуються в реальних системах, де енергоспоживання нелінійно залежить від робочих параметрів;

Моделі динамічного та статичного споживання.

Динамічне споживання враховує роботу процесора та периферій під час виконання завдань, що визначається виразом:

$$P_{active} = C_{dyn} V^2 f \quad (3.2)$$

де,  $C_{dyn}$  - динамічна ємність перемикань;

$V$  - напруга живлення;

$f$  - тактова частота.

В активному режимі мікроконтролер виконує обчислення, обробку даних та керує периферією. Потужність у цьому режимі залежить від тактової частоти та навантаження.

Статичне споживання визначає втрати енергії в стані спокою через струми витоку:

$$P_{sleep} = I_{leak}V \quad (3.3)$$

де  $I_{leak}$  - виток струму;

У сплячому режимі процесор припиняє виконання команд, але деякі периферійні пристрої можуть залишатися активними.

Перемикання між активним і сплячим режимами також вимагає енергії через необхідність перезавантаження регістрів та стабілізації живлення. Потужність у цьому процесі залежить від кількості перемикачів:

$$E_{trans} = P_{trans}N_{trans} \quad (3.4)$$

Для використання цієї моделі потрібно вводити такі вхідні дані: характеристики апаратного забезпечення - тип процесора і його максимальна частота, режим роботи - час перебування у активному й сплячому стані та тип виконуваних операцій - вплив різних типів операцій на споживання енергії.

Вихідними даними будуть розрахункові значення загального споживання енергії за певний період часу або під час виконання певної програми.

Вибір цієї конкретної моделі обумовлений її простотою та адекватністю для багатьох практичних застосувань.

Ця модель дозволяє оцінити вплив різних факторів на загальне споживання енергії під час виконання завдань з різними вимогами до швидкості обробки інформації та часу бездіяльності системи.

Крім того, така модель може бути легко розширена шляхом включення додаткових факторів впливу для більш точної оцінки енергоспоживання сучасних багатоядерних процесорів чи спеціалізованих чипсетів.

Вона може бути застосована для оптимізації програмного забезпечення з метою зниження потреби в електроенергії при розробці пристроїв з обмеженими джерелами живлення (наприклад, носимих пристроїв або IoT-програм).

Крім того, вона може бути використана як основа для подальших досліджень щодо вдосконалення методології розрахунку й прогнозування потреби в електроенергії майбутніх поколінь енергоефективних систем.

Вплив параметрів на енергоспоживання. Енергоспоживання мікроконтролера визначається не тільки його режимами роботи, але й різними параметрами, які можуть значно впливати на загальний рівень витрат енергії. Основними такими параметрами є: тактова частота, напруга живлення, перемикання між режимами роботи та використання периферійних модулів.

Тактова частота - енергоспоживання прямо пропорційне частоті роботи мікроконтролера. Висока тактова частота збільшує кількість перемикань транзисторів, що призводить до підвищеного динамічного енергоспоживання. Використання технологій динамічного масштабування частоти (DVFS) може допомогти знизити витрати енергії, адаптуючи частоту до поточних потреб системи. Наприклад, ARM Cortex-M процесори активно використовують DVFS для зменшення енергоспоживання в режимах низького навантаження.

Напруга живлення - напруга живлення впливає на енергоспоживання квадратично: зменшення напруги призводить до значного зниження енергоспоживання, проте може обмежувати продуктивність мікроконтролера. Зниження напруги можливе при застосуванні сучасних енергозберігаючих схем.

Перемикання між режимами роботи - часті переходи між активним і сплячим режимами можуть викликати додаткові втрати енергії через необхідність перезапуску регістрів та стабілізації живлення. Оптимізація програмного коду дозволяє мінімізувати такі втрати.

Використання периферійних модулів - периферійні пристрої, такі як UART, SPI, I2C, можуть суттєво впливати на загальне енергоспоживання. Використання ефективних алгоритмів обробки даних і правильне керування периферією допомагають мінімізувати ці втрати.

Також є ще додаткові фактори впливу на енергоспоживання: температура середовища - зміни температури можуть впливати на характеристики транзисторів, що змінює витрати енергії. Підвищення температури може збільшувати струми витоку, що підвищує статичне енергоспоживання. При температурі понад 85°C струми витоку можуть зрости в кілька разів порівняно з номінальними умовами. Для вирішення цієї проблеми використовуються термостабільні матеріали та спеціальні схеми компенсації.

Вплив типу виконуваних операцій - арифметичні операції (наприклад, множення або ділення) зазвичай споживають більше енергії порівняно з логічними (AND, OR). Оптимізація коду шляхом використання ефективних інструкцій або апаратних прискорювачів може значно знизити витрати енергії. Наприклад, використання апаратного блоку для обробки чисел із плаваючою точкою (FPU) замість програмної емуляції або заміна складних арифметичних операцій простішими еквівалентами

Розробка ефективних алгоритмів керування режимами роботи - оптимізація програмного коду дозволяє мінімізувати кількість переходів між режимами роботи. Використання технологій, таких як динамічне масштабування напруги (DVFS), дозволяє адаптивно зменшувати енергоспоживання.

Перспективи застосування нових технологій - новітні мікроконтролери впроваджують енергоефективні архітектури та оптимізовані режими роботи. Використання технологій, таких як наноенергетичні схеми та ультранизьковольтні транзистори, може значно знизити енергоспоживання в майбутніх поколіннях мікроконтролерів.

Оскільки оцінка енергоспоживання мікроконтролерів залежить від великої кількості параметрів, її практична реалізація потребує використання спеціалізованих інструментів моделювання. Використання програмних

симуляторів дозволяє отримати точні результати без необхідності фізичного тестування пристроїв.

Основні середовища моделювання, які застосовуються для аналізу енергоспоживання мікроконтролерів є Proteus, Multisim, Matlab, LTspice

Proteus - найпотужніша система автоматизованого проектування, що дає змогу віртуально змоделювати роботу величезної кількості аналогових і цифрових пристроїв. Також це популярне середовище для схемотехнічного моделювання та емуляції мікроконтролерів. Середовище дозволяє завантажувати прошивку у віртуальний мікроконтролер і аналізувати його роботу без фізичного обладнання.

Програмний пакет Proteus VSM дає змогу зібрати схему будь-якого електронного пристрою і симулювати його роботу, виявляючи помилки, допущені на стадії проектування і трасування.

Програма складається з двох модулів. ISIS - редактор електронних схем з подальшою імітацією їхньої роботи. ARES - редактор друкованих плат, вбудованим редактором бібліотек і автоматичною системою розміщення компонентів на платі.

Proteus містить понад 6000 електронних компонентів з усіма довідковими даними, а також демонстраційні ознайомчі проекти. Програма має інструменти USBCONN і COMPIM, які дають змогу під'єднати віртуальний пристрій до портів USB і COM комп'ютера.

Можливості для симуляції енергоспоживання: використання вбудованого осцилографа та вимірювальних інструментів для аналізу електричних параметрів. Моделювання роботи периферійних пристроїв, що впливають на енергоспоживання. Використання аналізу живлення (Power Analysis) для оцінки споживаної потужності.

Розробка нових версій програмного забезпечення типу Proteus передбачає подальшу інтеграцію сучасних технологічних рішень щодо зниження споживання енергії та підвищення ефективності моделювання складних електронних систем.

Multisim - є популярним програмним середовищем для схемотехнічного моделювання та аналізу електронних схем. Воно належить компанії National

Instruments і широко використовується у світі проектування електронних пристроїв як у промисловості, так і в навчальних закладах.

Multisim дозволяє створювати й тестувати складні принципові схеми з використанням широкого набору бібліотечних компонентів, включаючи джерела живлення, резистори, конденсатори тощо.

Програма включає віртуальні контрольно-вимірювальні прилади (осцилографи, генератори сигналів тощо), які імітують реальні аналоги й дозволяють проводити симуляцію поведінки системи під різними умовами роботи. Є велика база електронних компонентів (понад 16 тисяч деталей) та підтримує можливість створення нових бібліотек.

Також Multisim включає моделювання мікроконтролерів різних типів (раніше відомий як MultiMCU), що робить його корисним інструментом для розробки систем на основі програмованих пристроїв.

Matlab - це високорівневе інтерпретоване середовище програмування та пакет прикладних програм для числового аналізу, розроблений компанією The MathWorks. Він широко використовується в наукових дослідженнях, інженерії та освіті для виконання математичних обчислень, моделювання систем і візуалізації даних.

Основні можливості Matlab: робота з матрицями, лінійна алгебра, статистика тощо. Вирішення диференціальних рівнянь і створення моделей поведінки складних систем. Побудова графіків (дво- і тривимірних) та анімаційних роликів для представлення результатів досліджень, а також дозволяє створювати математичні моделі поведінки систем, що включає аналіз динамічних процесів та оптимізацію параметрів для досягнення бажаних результатів. Програма надає потужні засоби для побудови графіків і анімаційних роликів, що допомагає візуалізувати дані й краще розуміти поведінку системи під різними умовами роботи

Переваги використання Matlab - простота використання завдяки високорівневій мові програмування та широкий спектр бібліотек і

інструментальних засобів для різних завдань від математичного моделювання до обробки сигналів.

До недоліки Matlab слід віднести високу вартість ліцензії для комерційного використання порівняно з іншими мовами програмування.

LTspice - це потужний і безкоштовний SPICE-симулятор для аналогових електронних схем, розроблений компанією Analog Devices (раніше Linear Technology). Він широко використовується в галузі радіочастотної електроніки, енергетики та інших дисциплінах для проектування й тестування складних схем.

LTspice є одним із найкращих інструментів для аналізу схем живлення мікроконтролерів, що включає аналіз стабільності джерел живлення та оптимізацію їхньої конструкції. Програма дозволяє створювати й тестувати моделі поведінки різних електронних компонентів під різними умовами роботи, що допомагає передбачити їхню роботу в реальних умовах.

Переваги використання LTspice: швидкість і точність симуляцій завдяки спеціалізованим моделям для комутаційних джерел живлення. Відкритість та відсутність обмежень, що робить його популярним серед інженерів та студентів.

Щодо недоліків то LTspice використовує свій власний синтаксис для опису моделей компонентів, що може створювати труднощі при імпорті моделей з інших симуляторів типу PSpice.

Середовища моделювання типу Proteus, Multisim, Matlab, LTspice та багато інших, відіграють важливу роль у процесі розробки енергоефективних систем на основі мікроконтролерів. Кожне з цих середовищ має свої переваги й особливості використання.

Використання цих інструментів дозволяє оптимізувати конструкцію систем на основі мікроконтролерів щодо мінімального споживання енергії ще на стадії проектування. Крім того, вони сприяють скороченню часу розробки нових продуктів шляхом попередньої оцінки впливу різних параметрів на загальне споживання енергії системи без витрат часу і ресурсів на створення прототипу в реальному світі.

### 3.2 Метод симуляції енергоспоживання мікроконтролера

Для реалізації симуляції енергоспоживання мікроконтролера було обрано мікроконтролер ATmega328P (зображений на рисунку 3.1), що є оптимальним рішенням завдяки своїй універсальності, доступності та широкій підтримці в інженерній спільноті.

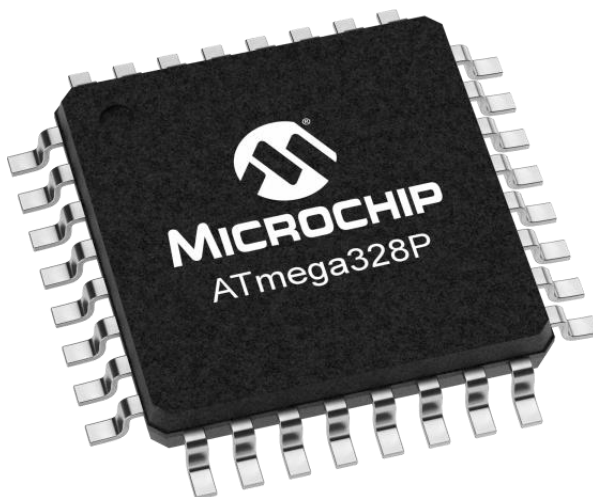


Рисунок 3.1 - мікроконтролер ATmega328P

ATmega328P - це 8-бітний мікроконтролер AVR, який поєднує в собі 32 КБ флеш-пам'яті, 2 КБ ОЗП та 1 КБ EEPROM, що робить його придатним для широкого спектру застосувань, від простих вбудованих систем до складніших пристроїв з мікропроцесорним керуванням.

Завдяки тактовій частоті до 20 МГц, ATmega328P забезпечує достатню обчислювальну потужність для реалізації алгоритмів керування та обробки даних у реальному часі. Крім того, наявність різноманітних інтерфейсів, таких як I2C, SPI та UART/USART, забезпечує гнучкість при підключенні до різних периферійних пристроїв та датчиків.

Варто відзначити, що ATmega328P є енергоефективним рішенням, що є важливим фактором при розробці автономних систем та пристроїв з живленням від батарей. Можливість роботи в широкому діапазоні напруг живлення (від 1,8

до 5,5 В) робить його придатним для використання в різних умовах та з різними джерелами живлення.

Крім того, АТmega328Р має вбудовані модулі, такі як таймери/лічильники, зовнішні переривання та Watchdog-таймер, що дозволяє реалізувати складні алгоритми керування та забезпечити надійну роботу системи в цілому.

Для проведення моделювання енергоспоживання мікроконтролера АТmega328Р у рамках магістерської роботи було обрано програмне середовище Proteus VSM. Це рішення зумовлене численними перевагами, які надає даний пакет програм для автоматизованого проектування електронних схем.

Proteus VSM, розроблений компанією Labcenter Electronics, є потужним інструментом для створення та моделювання електронних схем. Він включає два основних модулі: ISIS для проектування електронних схем та ARES для розробки друкованих плат.

Ця система дозволяє не лише створювати схеми, але й симулювати їх роботу в реальному часі, що є критично важливим для виявлення та усунення помилок на ранніх етапах проектування.

Однією з ключових особливостей Proteus є можливість створення електронних схем із реалістичним моделюванням роботи компонентів. Також є велика бібліотека компонентів, яка містить більше 6000 елементів, що дозволяє легко реалізувати різноманітні проекти

Алгоритм проведення симуляції. Алгоритм проведення симуляції роботи мікроконтролера АТmega328Р передбачає послідовне виконання кількох етапів, кожен з яких спрямований на моделювання роботи пристрою, аналіз його енергоспоживання та перевірку режимів роботи. Кроки, які необхідні для успішної симуляції:

- 1) створення схеми мікроконтролера - для початку необхідно створити схему, яка включає мікроконтролер АТmega328Р, з урахуванням його базових вимог до підключення.

Живлення - АТmega328Р працює від стабільної напруги 5V. Для цього використовується джерело живлення, наприклад, USB-порт або зовнішній

стабілізований блок живлення. Виводи VCC і AVCC мікроконтролера підключаються до джерела живлення через фільтруючі конденсатори (наприклад, 0,1 мкФ і 10 мкФ), які забезпечують стабільність напруги та зменшують рівень шумів.

Заземлення - виводи GND мікроконтролера підключаються до загального заземлення схеми.

Кварцовий генератор - для забезпечення стабільної тактової частоти роботи мікроконтролера використовується зовнішній кварцовий резонатор із частотою 16 МГц. Він підключається до ніжок XTAL1 та XTAL2 через два конденсатори (зазвичай по 22 пФ), які забезпечують правильну роботу генератора.

2) додавання світлодіода для перевірки активності - світлодіод (LED) використовується як індикатор активності мікроконтролера.

Один із портів вводу/виводу (наприклад, PORTB0) налаштовується як вихідний. Світлодіод підключається до цього порту через струмообмежувальний резистор (220-330 Ом), який запобігає перевищенню допустимого струму через світлодіод. При активації відповідного порту світлодіод вмикається, що дозволяє візуально спостерігати за станом роботи мікроконтролера.

3) включення кнопки для перемикання між режимами енергоспоживання - для реалізації можливості перемикання між різними режимами енергоспоживання додається кнопка.

Кнопка підключається до одного з портів вводу (наприклад, PORTD2) через підтягувальний резистор (10 кОм), який забезпечує стабільний рівень сигналу на вході при відсутності натискання. При натисканні на кнопку порт отримує сигнал низького рівня (логічний "0"), що програмно обробляється для зміни режиму роботи мікроконтролера (активний режим, сон або глибокий сон).

Для забезпечення високої точності результатів та адекватного відображення реальних умов експлуатації, необхідно правильно налаштувати ключові параметри мікроконтролера: робочу напругу та тактову частоту. Це дозволяє не лише симулювати правильну поведінку мікроконтролера, а й забезпечити сумісність із реальними пристроями на етапі тестування та розробки.

Встановлення робочої напруги у середовищі Proteus задається робоча напруга 5V для всіх компонентів схеми. Це відповідає реальним умовам експлуатації ATmega328P і дозволяє коректно моделювати його поведінку.

Вибір тактової частоти визначає швидкість виконання інструкцій мікроконтролером. Для ATmega328P встановлюється частота 16 МГц, що є стандартним значенням при використанні зовнішнього кварцу.

Наступний етап - це запуск текстових програм:

1) перевірка енергоспоживання в активному режимі - на цьому етапі завантажується тестова програма, яка реалізує блимання світлодіода (LED) як індикатора активності мікроконтролера.

Схема міститиме мікроконтролер ATmega328P, світлодіод, кварцовий резонатор на 16 МГц з конденсаторами по 22 пФ, а також кнопку скидання (Reset) з підтягувальним резистором 10 кОм.

Світлодіод вмикається та вимикається з певною періодичністю, що дозволить візуально спостерігати за активністю мікроконтролера. Після 10 циклів блимання світлодіода мікроконтролер переходить у режим сну (Sleep Mode);

2) вимірювання струму в режимі сну - у цьому режимі мікроконтролер знизить своє енергоспоживання до мінімуму.

Програма використовує бібліотеки управління живленням (наприклад, `avr/sleep.h`) для переведення мікроконтролера у режим сну після виконання 10 циклів блимання.

У режимі сну центральний процесор і більшість периферійних модулів вимикаються, що дозволяє зменшити споживаний струм;

3) вимірювання струму в режимі глибокого сну - у цьому режимі всі периферійні модулі відключаються повністю, що дозволить досягти найнижчого рівня споживаної енергії. Мікроконтролер може реагувати лише на зовнішні переривання - натискання кнопки скидання.

Цей етап дозволить не лише перевірити працездатність системи на базі ATmega328P, але й детально проаналізувати зміни в енергоспоживанні під час переходів між різними режимами роботи.

Для збору та аналізу даних буде використовуватись Proteus Virtual Instruments для вимірювання енергоспоживання мікроконтролера. Для аналізу струму використовується закон Ома:

$$I = \frac{U}{R} \quad (3.5)$$

де  $I$  - струм через резистор,

$U$  - напруга на резисторі,

$R$  - його опір.

Таким чином, вимірявши напругу на резисторах у схемі, можна обчислити струм, який споживає мікроконтролер у кожному режимі.

Після збору даних у середовищі симуляції результати порівнюються з теоретичними значеннями.

Використовуються технічні характеристики мікроконтролера ATmega328P (зазначені в документації), а також параметри схеми (напруга живлення, опори резисторів тощо). На основі цих даних обчислюється очікуване споживання струму для кожного режиму роботи.

Методи збору та аналізу даних дозволяють детально дослідити роботу мікроконтролера ATmega328P у різних режимах енергоспоживання.

Завдяки використанню інструментів симуляції Proteus можна точно виміряти струм і напругу без фізичного обладнання.

Порівняння експериментальних результатів із теоретичними розрахунками дає змогу оцінити ефективність роботи системи та знайти шляхи для її оптимізації.

При проведенні симуляції енергоспоживання мікроконтролера ATmega328P важливо враховувати можливі похибки та обмеження, які можуть вплинути на точність отриманих результатів.

Програмне середовище Proteus, як і будь-який інший інструмент моделювання, має певні спрощення, що можуть призвести до розбіжностей між віртуальними та реальними вимірюваннями.

Proteus не враховує нагрів компонентів, що може впливати на реальні виміри. Під час роботи електронних компонентів, таких як мікроконтролер, резистори та світлодіоди, відбувається нагрів. Зміна температури компонентів може впливати на їх електричні характеристики.

Опір резисторів може змінюватися залежно від температури. У реальних умовах нагрівання резистора призводить до збільшення його опору, що, в свою чергу, впливає на струм у ланцюзі.

Це явище описується температурним коефіцієнтом опору, який для металевих резисторів зазвичай становить приблизно  $+0.4\%$  на градус Цельсія. При підвищенні температури опір може зростати, що призводить до зменшення струму в ланцюзі і, відповідно, до зміни роботи всіх пов'язаних компонентів.

Ефективність світлодіода також залежить від температури. Нагрівання може знизити яскравість світлодіода та змінити його вольт-амперну характеристику.

Зокрема, при підвищенні температури зменшується напруга пробою світлодіода і збільшується струм витоку, що може призвести до перегріву і навіть виходу з ладу компонента. Це особливо критично для потужних світлодіодів, які використовуються в освітлювальних системах.

Мікроконтролер - температура кристала мікроконтролера впливає на його внутрішні параметри, такі як швидкість перемикання транзисторів та струми витоку.

З підвищенням температури швидкість перемикання може знижуватися, що призводить до затримок у виконанні програм та зниження загальної продуктивності системи.

Крім того, підвищена температура може викликати збільшення струму витоку в транзисторах, що негативно позначається на енергоспоживанні та стабільності роботи мікроконтролера.

Оскільки Proteus не моделює теплові процеси, результати симуляції можуть відрізнятися від реальних вимірювань, особливо при тривалій роботі схеми або при високих навантаженнях.

Енергоспоживання в реальних умовах визначається не лише характеристиками пристрою, але й низкою зовнішніх факторів, які не враховуються в симуляторі.

Зокрема, температурні умови здатні змінювати електричні властивості компонентів, що впливає на загальну витрату енергії. Наприклад, підвищення температури може спричинити зростання струмів витоку в транзисторах, що збільшує статичне енергоспоживання.

Крім того, важливу роль відіграє якість електронних компонентів: реальні елементи мають певні допуски, через що параметри, як-от опір або напруга світлодіодів, можуть відхилитися від номінальних значень.

Це, у свою чергу, впливає на точність функціонування схеми й її енергоспоживання. Застосування високоякісних компонентів із малим допуском допомагає знизити ці ризики.

Також суттєвим чинником є стабільність напруги живлення: її коливання здатні викликати зміну струмів у схемі, що безпосередньо впливає на загальну споживану енергію. Зокрема, нестабільне живлення може призвести до нерівномірної роботи мікроконтролера, спричинити появу помилок або збоїв у виконанні програмних інструкцій, а також вплинути на точність роботи аналогових компонентів, таких як АЦП чи компаратори.

У довгостроковій перспективі такі коливання можуть навіть скорочувати термін служби електронних компонентів через підвищене теплове навантаження або перевищення граничних параметрів. Для уникнення подібних негативних наслідків доцільно використовувати стабілізатори напруги - як лінійні, так і імпульсні, залежно від вимог до точності, ефективності та габаритів схеми. Вони забезпечують стабільне живлення незалежно від змін навантаження, вхідної напруги або зовнішніх умов, сприяючи більш надійній і прогнозованій роботі електронного пристрою.

У Proteus ці фактори не враховуються, тому симуляція відображає ідеалізовану модель, яка може відрізнитися від реальної поведінки пристрою.

У таблиці 3.1 наведено основні відмінності між симуляцією та реальними вимірюваннями

Таблиця 3.1 Відмінності між симуляцією та реальними вимірюваннями

Параметр	Симуляція в Proteus	Реальне вимірювання
Врахування нагріву	Не враховує	Температура впливає на роботу компонентів
Точність вимірювання	Висока але без похибок	Є похибки через якість компонентів
Час роботи батареї	Не враховує реальні умови	Може бути змінено через внутрішні втрати
Вплив паразитних елементів	Ідеальні моделі компонентів	Враховуються паразитні ємності та індуктивності
Фактичне енергоспоживання	Розраховане за ідеальними формулами	Може змінюватись через похибки компонентів
Вплив старіння компонентів	Не враховується	Електронні компоненти зношуються з часом

Програмна модель мікроконтролера не завжди точно відображає всі характеристики реального пристрою.

Програмна модель мікроконтролера, яка використовується в Proteus, є спрощеною копією реального пристрою. Вона може не враховувати деякі нюанси

поведінки реального мікроконтролера, такі як: внутрішні затримки та паразитні ємності та індуктивності

Внутрішні затримки - у реальному мікроконтролері існують внутрішні затримки, пов'язані з перемиканням транзисторів, передачею сигналів між різними блоками та виконанням інструкцій. Ці затримки можуть впливати на точний час виконання програми та, як наслідок, на енергоспоживання.

Паразитні ємності та індуктивності - реальні компоненти мають паразитні ємності та індуктивності, які можуть впливати на швидкість перехідних процесів та енергоспоживання. У симуляції ці ефекти зазвичай не враховуються.

Враховуючи ці обмеження, важливо розуміти, що результати симуляції будуть лише приблизною оцінкою енергоспоживання реального пристрою.

Для отримання більш точних даних необхідно проводити експериментальні вимірювання на фізичному прототипі.

### 3.3 Висновки до даного розділу

Симуляція енергоспоживання мікроконтролера є важливим етапом при розробці енергоефективних вбудованих систем.

Вона дозволяє ще на етапі проєктування оцінити вплив різних факторів на загальне споживання енергії, знайти оптимальні режими роботи та запропонувати методи його зниження.

Існує кілька підходів до моделювання енергоспоживання мікроконтролерів, і кожен з них має свої особливості та рівень точності залежно від глибини деталізації.

Одним із поширених методів є програмне моделювання, яке реалізується у таких середовищах, як Proteus, LTspice, Multisim або Matlab. Цей підхід дозволяє змоделювати електричну схему пристрою та оцінити, як різні параметри мікроконтролера впливають на загальне енергоспоживання. Хоча цей спосіб досить зручний і наочний, він не враховує всіх фізичних аспектів, таких як температурні коливання чи відхилення реальних компонентів.

Іншим методом є аналітичний розрахунок, який ґрунтується на технічній документації мікроконтролера. У цьому випадку враховуються типові значення струму споживання в різних режимах роботи, напруга живлення та частота тактування.

Такий підхід дозволяє оперативно отримати оцінку енергоспоживання без запуску симуляцій, але водночас ігнорує динамічні зміни, характерні для реального середовища експлуатації пристрою.

Найвищу точність забезпечують спеціалізовані апаратні засоби, такі як EnergyTrace або Power Profiler Kit. Вони дозволяють виконувати вимірювання струму та напруги в реальному часі безпосередньо на фізичному мікроконтролері.

Завдяки цьому вдається отримати максимально точну інформацію про споживану енергію в конкретних умовах. Проте цей метод потребує наявності відповідного обладнання та вже зібраного пристрою, що ускладнює його використання на етапі початкового проектування.

Аналіз даних, отриманих під час симуляції, дозволяє розробникам оптимізувати як програмне забезпечення, так і апаратну частину системи для зменшення енергоспоживання.

Цей процес включає вибір оптимальних алгоритмів, налаштування режимів енергозбереження та використання енергоефективних компонентів.

Отримані дані можна ефективно використовувати для розробки енергоефективних вбудованих систем шляхом оптимізації коду, правильного керування режимами мікроконтролера, а також налаштування апаратних параметрів.

Важливим аспектом є розробка стратегій управління живленням, які дозволяють автоматично переходити в режими зниженого споживання при відсутності активних задач.

Правильне використання цих методів дозволяє значно зменшити енергоспоживання пристроїв, продовжити час роботи від батареї, підвищити загальну ефективність системи та зменшити негативний вплив на навколишнє середовище.

## 4 РЕАЛІЗАЦІЯ ТА СИМУЛЯЦІЯ МОДЕЛІ ЕНЕРГОСПОЖИВАННЯ

### 4.1 Реалізація методу симуляції енергоспоживання мікроконтролера

Сучасні вбудовані системи часто працюють в умовах обмеженого енергопостачання — наприклад, у складі бездротових сенсорних мереж, пристроїв Інтернету речей (IoT) або автономних елементів керування. В таких випадках енергоефективність стає одним із ключових критеріїв при виборі апаратних та програмних рішень. Особливо це стосується мікроконтролерів, які відіграють центральну роль у побудові подібних систем. Вони відповідають за обробку даних, контроль периферії та організацію енергозберігаючих режимів, що дає змогу значно продовжити час автономної роботи пристрою без заміни або підзарядки батарей.

Крім того, вимоги до компактності та надійності таких систем зумовлюють необхідність попереднього аналізу та точного налаштування споживання енергії ще на етапі розробки. Саме тому все більшого значення набуває використання інструментів симуляції, які дають можливість моделювати поведінку пристрою в різних сценаріях експлуатації. Це дозволяє не лише передбачити навантаження на елемент живлення, але й своєчасно внести корективи в архітектуру схеми або алгоритм програми для досягнення бажаного балансу між продуктивністю та енергоефективністю.

Для ефективної оцінки споживання енергії та оптимізації роботи мікроконтролера доцільним є використання симуляційних методів, які дозволяють дослідити поведінку системи в різних режимах роботи без потреби у створенні фізичного прототипу. Це значно економить час і ресурси на етапі розробки.

У цьому розділі описується процес моделювання енергоспоживання мікроконтролера ATmega328P за допомогою середовища Proteus. Основною метою є аналіз споживання електроенергії мікроконтролером у різних режимах роботи, що дозволяє оптимізувати енергоефективність пристроїв. Такий підхід є

важливою складовою сучасного інженерного підходу до створення «розумної» електроніки, орієнтованої на економне використання енергоресурсів.

Для проведення симуляції використовується програмне середовище Proteus, яке дозволяє віртуально моделювати роботу мікроконтролерів у різних умовах. Середовище складається з двох основних модулів: ISIS для створення і симуляції електричних схем, і ARES для розробки друкованих плат.

Ця програма дозволяє точно відтворити поведінку мікроконтролера в різних режимах роботи, включаючи режими зниженого енергоспоживання.

Таким чином, реалізація методу симуляції енергоспоживання на базі середовища Proteus є ефективним інструментом для аналізу та оптимізації вбудованих систем ще на етапі розробки. Це значно скорочує час проектування, зменшує витрати на створення апаратних прототипів і підвищує надійність кінцевих рішень.

На рисунку 4.1 показана схема, використана для симуляції енергоспоживання мікроконтролера ATmega328P.

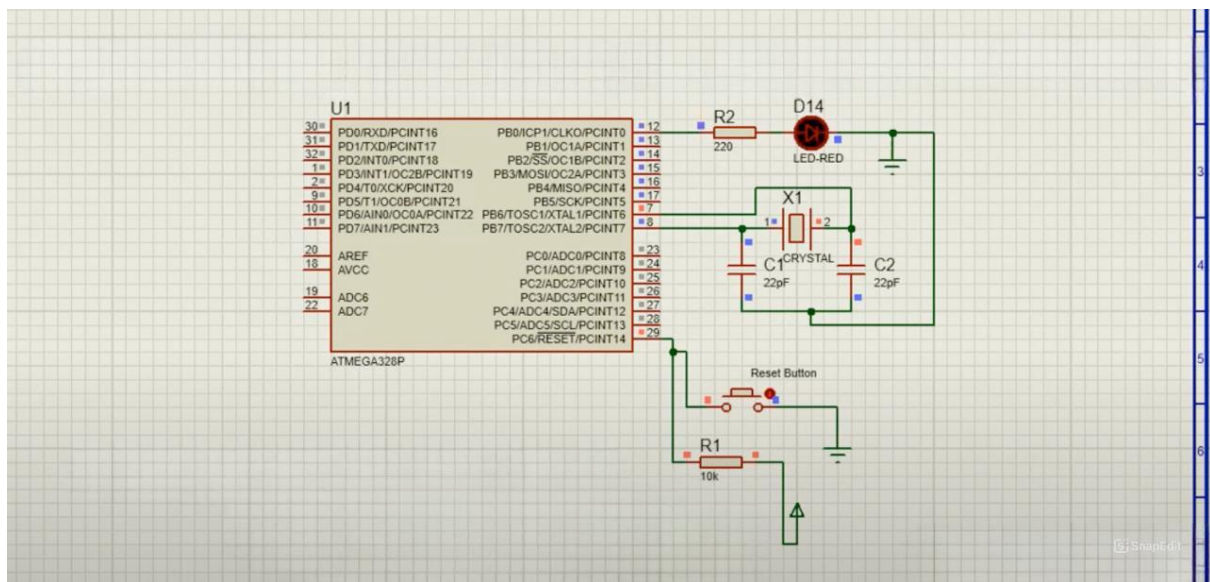


Рисунок 4.1 Схема вимірювання енергоспоживання мікроконтролера ATmega328P

Під час симуляції проводиться аналіз споживання енергії мікроконтролером у різних режимах роботи, включаючи активний режим, режим Sleep і режим Power Down.

Цей аналіз дозволяє визначити найбільш ефективні стратегії енергозбереження для конкретного застосування мікроконтролера

Схема складається з декількох ключових компонентів, які забезпечують роботу мікроконтролера ATmega328P у різних режимах. Компоненти схеми:

1) мікроконтролер ATmega328P (U1) - центральний компонент схеми, який виконує програмний код, зображений на рисунку 4.2. У цій схемі він використовується для керування світлодіодом та реагування на натискання кнопки скидання (Reset). Підтримує кілька режимів зниження енергоспоживання.

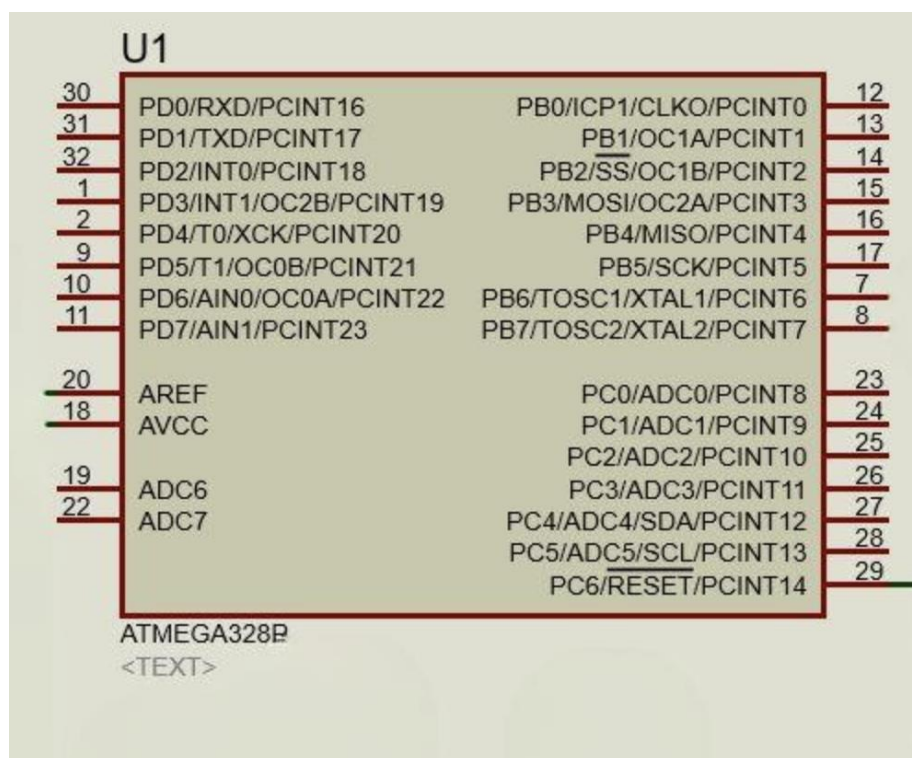


Рисунок 4.2 - Мікроконтролера ATmega328P

2) світлодіод LED-RED (D14) - вказує на роботу мікроконтролера. Підключений через резистор (R2), щоб обмежити струм і запобігти його перегоранню. Світлодіод загоряється, коли на відповідний вихід мікроконтролера подається високий рівень напруги. Зображений на рисунку 4.3

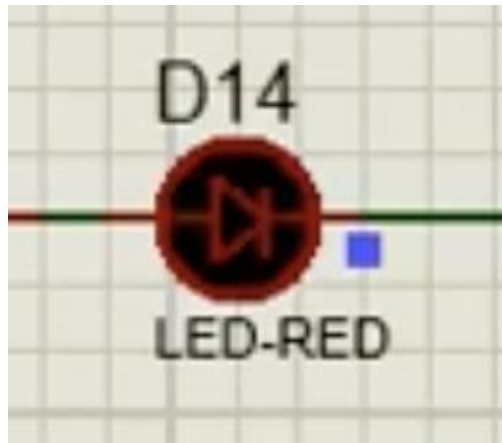


Рисунок 4.3 - Світлодіод LED-RED

3) резистори: R1 (10 кОм) - підтягуючий резистор для кнопки Reset. Використовується для утримання високоімпедансного стану на вході Reset мікроконтролера, щоб уникнути випадкових перезавантажень.

Коли кнопка натиснута, вона замикає контакт на землю, що призводить до скидання мікроконтролера. R2 (220 Ом) - Обмежувальний резистор для світлодіода. Обмежує струм через світлодіод, щоб не допустити його пошкодження. Резистор зображений на рисунку 4.4

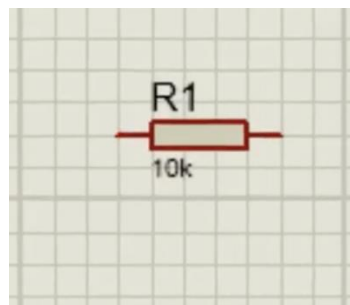


Рисунок 4.4 - Резистор

4) кварцовий генератор (X1) - CRYSTAL - забезпечує стабільну тактову частоту для мікроконтролера (найчастіше 16 МГц).

Використовується для коректної роботи програмних таймерів і точних обчислень часу. Зображений на рисунку 4.5

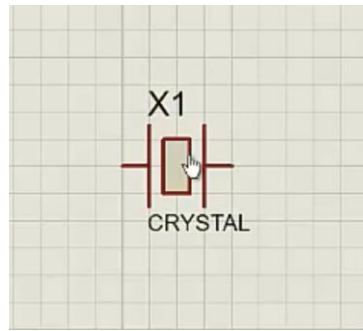


Рисунок 4.5 - Кварцовий генератор (X1)

5) конденсатори (C1, C2 - 22 пФ) - використовуються разом із кварцовим генератором для стабілізації його роботи та зменшення шумів. Допомагають покращити надійність роботи мікроконтролера на високих частотах. Конденсатори C1 та C2 зображені на рисунку 4.6

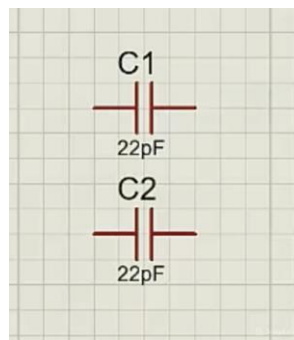


Рисунок 4.6 - Конденсатори C1, C2

б) кнопка Reset - використовується для перезапуску мікроконтролера. Коли натискається, вона підключає вхід Reset до землі, спричиняючи перезапуск програми. Кнопка зображена на рисунку 4.7



Рисунок 4.7 - Кнопка Reset

7) заземлення (GND) та живлення (VCC, AVCC) - GND (Ground) - загальна точка з'єднання для всіх компонентів схеми. VCC (5V) - Основне живлення мікроконтролера. AVCC - Живлення для аналогової частини мікроконтролера (використовується для ADC).

Принцип роботи схеми полягає в аналізі енергоспоживання мікроконтролера ATmega328P у різних режимах роботи. Після подачі живлення мікроконтролер запускається, використовуючи кварцовий резонатор 16 МГц, а конденсатори ємністю 22 пФ забезпечують його стабільну роботу. На початку експерименту проводиться вимірювання споживаного струму в стандартному режимі, коли мікроконтролер активно виконує програму. У цей момент світлодіод, підключений через резистор 220 Ом, періодично вмикається та вимикається, що дозволяє відстежити зміну стану виходів контролера. Через кілька секунд, відповідно до завантаженого алгоритму, мікроконтролер перемикається в енергозберігаючий режим Sleep Mode. У цьому стані його основні блоки відключаються, що значно зменшує рівень споживання електроенергії. На цьому етапі знову проводиться вимірювання струму, що дозволяє порівняти його значення з отриманими у звичайному режимі.

У разі потреби, симуляцію можна повторити, обравши інші енергозберігаючі режими роботи мікроконтролера, зокрема Idle Mode. У цьому режимі основне ядро процесора зупиняється, але периферійні модулі залишаються активними, що дозволяє зменшити енергоспоживання, зберігаючи при цьому функціональність, необхідну для обробки зовнішніх подій.

На початковому етапі запуску мікроконтролера важливо забезпечити правильне живлення - при подачі напруги на VCC (5 В) мікроконтролер автоматично починає виконання прошитої програми. Джерелом стабільної тактової частоти слугує кварцовий резонатор на 16 МГц (X1), який забезпечує надійну роботу завдяки своїм високочастотним коливанням. Два конденсатори, C1 і C2, номіналом по 22 пФ, підключені до резонатора, виконують функцію

стабілізації - знижують рівень шуму та підвищують точність генерації частоти, що особливо важливо для синхронної роботи мікросхеми.

Система управління світлодіодом реалізується просто: один із виводів мікроконтролера, наприклад PB0, підключається до аноду світлодіода через обмежувальний резистор на 220 Ом. При подачі високого логічного рівня на цей вивід світлодіод загоряється, а при переході в нульовий рівень - гасне.

Така схема дозволяє реалізувати візуальний контроль стану пристрою або відображення певних подій.

Окремо слід зазначити функціональність кнопки скидання (Reset). Вона підключається до відповідного виводу мікроконтролера через резистор номіналом 10 кОм, який утримує вхід у високому логічному рівні. При натисканні кнопки відбувається короткочасне замикання на землю, що викликає перезапуск програми. Це зручно, адже дозволяє оновити роботу мікроконтролера без вимикання живлення, особливо під час розробки та тестування.

Режими роботи мікроконтролера мають безпосередній вплив на його енергоспоживання. У активному режимі пристрій постійно виконує інструкції, використовуючи максимальну кількість енергії.

Однак перехід до енергозберігаючих режимів, таких як Sleep або Idle, дозволяє значно знизити споживання, тимчасово зупинивши ядро або окремі периферійні блоки.

Такий підхід особливо актуальний для автономних систем, де важливо забезпечити тривалий час роботи від батареї.

Таким чином, схема дозволяє наочно оцінити вплив програмних методів енергозбереження на реальне споживання мікроконтролера, а отримані результати можуть бути використані для оптимізації роботи пристрою в майбутніх розробках.

## 4.2 Експериментальні дослідження та їх аналіз

Метою даного експерименту є всебічна оцінка енергоспоживання мікроконтролера ATmega328P у різних режимах його роботи: активному режимі, режимі очікування та режимі сну.

Це дослідження дозволяє отримати кількісні показники споживання електроенергії у кожному з режимів, що є важливим при проектуванні енергоефективних автономних систем, зокрема IoT-пристроїв, бездротових сенсорних мереж, портативної електроніки тощо.

Отримані результати можуть бути використані для вибору оптимального режиму роботи мікроконтролера в задачах, де важливий мінімальний рівень енергоспоживання, таких як IoT-пристрої та бездротові сенсорні мережі. Для проведення експерименту використовувалося середовище Proteus, яке дозволяє виконувати моделювання електронних схем та вимірювання параметрів споживання енергії у реальному часі.

Вимірювання струму здійснювалося за допомогою вбудованого амперметра Proteus, що дозволяє точно визначити рівень споживання електроенергії мікроконтролером у різних режимах роботи.

На рисунку 4.8 представлено алгоритм вимірювання енергоспоживання мікроконтролера в різних режимах роботи. Цей процес включає в себе ініціалізацію системи, вимірювання струму в активному, енергозберігаючому та режимі очікування, а також збереження отриманих даних для подальшого аналізу.

У процесі налаштування симуляції були задані параметри, що максимально наближені до реальних умов експлуатації мікроконтролера ATmega328P. Зокрема, напруга живлення була встановлена на рівні 5 В - це типове значення для цього типу мікроконтролера, що забезпечує стабільну роботу цифрових і аналогових модулів. Тактова частота становила 16 МГц - стандартне значення, що дозволяє оцінити енергоспоживання при повному навантаженні процесора.

Для аналізу впливу периферійних пристроїв у схемі були задіяні світлодіодний індикатор (LED) та кнопка керування, які часто використовуються

у практичних застосуваннях і дозволяють наочно простежити зміну споживаної потужності в активному режимі.

Крім того, було реалізовано перехід мікроконтролера в енергозберігаючий режим Power-down, при якому всі другорядні модулі відключаються, що значно знижує загальне енергоспоживання пристрою. Такий підхід дозволяє проаналізувати динаміку споживання енергії залежно від режиму роботи мікроконтролера та активності зовнішніх компонентів.

Під час експерименту виконувалася реєстрація значень струму для кожного з досліджуваних режимів. Зібрані дані дозволяють оцінити ефективність енергозбереження та порівняти теоретичні розрахунки із симуляційними вимірюваннями.



Рисунок 4.8 - Блок-схема алгоритму вимірювання енергоспоживання мікроконтролера

У рамках проведення експерименту було проаналізовано три ключові режими роботи мікроконтролера ATmega328P, що дозволило оцінити вплив кожного з них на загальне енергоспоживання системи.

У першому випадку, мікроконтролер працював у повністю активному режимі, без жодних обмежень. У цьому стані процесор безперервно виконував програмний код, зокрема, реалізовував циклічне вмикання та вимикання світлодіода. Такий режим характерний високим споживанням енергії, адже всі внутрішні блоки контролера, включаючи тактовий генератор, знаходяться в активному стані.

Другий сценарій передбачав використання енергозберігаючого режиму Sleep Mode. Після завершення десяти повних циклів роботи, мікроконтролер автоматично переходив у сплячий режим, що дозволяє значно зменшити енергоспоживання. У цьому режимі відключаються всі модулі, які не є критичними для функціонування, тоді як основне ядро тимчасово призупиняє виконання інструкцій. Це забезпечує баланс між збереженням енергії та готовністю до відновлення роботи за подією (наприклад, натисканням кнопки або таймерним перериванням).

Третій режим, Idle Mode, відрізняється тим, що процесор повністю припиняє виконання коду, однак периферійні блоки - такі як таймери, UART або АЦП - залишаються активними. Це дозволяє зменшити споживання енергії при збереженні функціональності, необхідної для взаємодії із зовнішніми пристроями чи підтримки часових інтервалів.

Такий підхід до моделювання дозволяє не лише порівняти рівні енергоспоживання в різних режимах, а й зробити обґрунтований вибір найбільш оптимального режиму для конкретного застосування.

Вимірювання енергоспоживання проводилося за допомогою вбудованого амперметра в середовищі Proteus, який дозволяє отримати точні дані про споживаний струм у різних режимах роботи.

На початку експерименту проводилося вимірювання у звичайному режимі, після чого система автоматично переходила у режим енергозбереження, і здійснювалося повторне вимірювання.

В таблиці 4.1 наведено дані, які були отримані під час симуляції мікроконтролера ATmega328P щодо споживання струму.

Таблиця 4.1. Енергоспоживання ATmega328P у різних режимах

Режим роботи	Теоретичне споживання (мА)	Симуляційне споживання (мА)
Активний	15	14.8
Sleep Mode	1.2	1.1
Idle Mode	4.5	4.3

Результати симуляції демонструють, що реальне енергоспоживання в середовищі Proteus майже повністю відповідає теоретичним розрахункам. Відмінності в межах 1-2% можуть бути зумовлені особливостями внутрішньої реалізації моделі мікроконтролера в середовищі Proteus та незначними похибками вимірювання.

Незначні розбіжності між теоретичними та симуляційними даними можуть бути спричинені такими факторами: модельні спрощення в Proteus: симулятор не враховує деякі реальні ефекти, такі як паразитні струми та температурні впливи на характеристики компонентів; похибки вимірювання: хоча вбудований амперметр у Proteus є досить точним, він не ідеально відображає всі внутрішні процеси, що відбуваються в мікроконтролері; перехідні процеси: при перемиканні між режимами роботи можуть виникати короточасні стрибки струму, які не завжди враховуються в розрахунках.

Аналіз отриманих результатів показав, що найбільш енергоефективним режимом роботи є режим сну (Sleep Mode), у якому споживання струму знижується у 12-14 разів у порівнянні з активним режимом. Режим Idle Mode також забезпечує суттєве зменшення енергоспоживання, хоча рівень споживаного струму є вищим, ніж у Sleep Mode.

Для подальшого зниження енергоспоживання у вбудованих системах, що базуються на мікроконтролерах, доцільно впроваджувати низку стратегічних заходів. Однією з найефективніших практик є використання динамічного керування живленням, коли периферійні модулі вимикаються у періоди бездіяльності. Це дозволяє уникнути марного витрачання енергії на компоненти, які не задіяні у конкретний момент часу.

Не менш важливою є оптимізація програмної логіки. Зменшення кількості непотрібних або надлишкових обчислень у критичних сценаріях дозволяє знизити навантаження на процесор і, відповідно, його енергоспоживання. Ефективно побудований код із мінімальним часом активності значно підвищує загальну енергоефективність пристрою.

У випадках, коли це допускають електричні характеристики підключених компонентів, варто перейти на використання зниженого рівня напруги живлення - наприклад, 3.3 В замість традиційних 5 В. Такий підхід зменшує як струм, так і теплові втрати в системі.

Окрему увагу слід приділити застосуванню таймерів та систем переривань. Замість постійного опитування датчиків, яке створює зайве навантаження на центральне ядро мікроконтролера, краще використовувати подієво-орієнтовану архітектуру, де реакція на зміни в середовищі здійснюється лише при необхідності. Це не тільки знижує енергоспоживання, але й покращує загальну продуктивність системи.

У перспективі подальші дослідження можуть бути орієнтовані на розширення можливостей оптимізації енергоспоживання вбудованих систем. Одним із напрямів є вивчення впливу різних типів навантаження - зокрема сенсорів, дисплеїв або бездротових модулів зв'язку - на загальний рівень

споживаної енергії. Адже ці компоненти значно варіюють за характеристиками, і навіть невеликі зміни у підключеному навантаженні можуть впливати на енергетичний баланс усієї системи.

Ще одним важливим напрямом є розробка нових алгоритмів керування енергоспоживанням. Зокрема, можна досліджувати методи динамічного масштабування частоти, які дозволяють адаптивно змінювати тактову частоту мікроконтролера залежно від поточного навантаження або обчислювальної потреби. Це може забезпечити значну економію енергії без втрати функціональності, особливо в багатозадачних або періодично активних пристроях.

Також доцільно розглянути тестування альтернативних мікроконтролерів, які мають розширені енергозберігаючі можливості. Наприклад, серії STM32 або MSP430 відомі своєю гнучкою архітектурою та підтримкою широкого спектра енергозберігаючих режимів. Порівняльний аналіз їх ефективності в різних умовах дозволить обрати оптимальне апаратне рішення для конкретного проекту.

Ці напрями досліджень мають велике значення для створення автономних, енергоефективних пристроїв нового покоління.

Отримані результати можуть бути використані для подальшої оптимізації енергоспоживання мікроконтролерів у проектах, що потребують автономної роботи пристроїв на батарейному живленні.

#### 4.3 Висновки до даного розділу

У даному розділі було проведено моделювання енергоспоживання мікроконтролера ATmega328P у середовищі Proteus, що дозволило оцінити його роботу в різних режимах.

Було розглянуто три основні режими: активний, очікування (Idle Mode) та сон (Sleep Mode). В ході експерименту виконувалися вимірювання сили струму для кожного з режимів з метою порівняння теоретичних та симуляційних

результатів. Отримані результати підтвердили, що використання енергозберігаючих режимів дозволяє суттєво зменшити енергоспоживання.

У режимі Sleep Mode споживаний струм знизився більш ніж у 12 разів у порівнянні з активним режимом, що є критично важливим для автономних пристроїв з батарейним живленням.

Також було встановлено, що симуляційні вимірювання в середовищі Proteus мають високу точність і демонструють лише незначні відхилення від теоретичних значень. Це дозволяє використовувати подібний підхід для оцінки енергоспоживання мікроконтролерів без необхідності проведення реальних фізичних експериментів.

На основі аналізу можна зробити висновок, що для оптимізації енергоспоживання у системах на базі ATmega328P слід активно використовувати механізми зниження тактової частоти, вимикання непотрібних периферійних пристроїв та ефективно керування режимами сну.

Для покращення методики моделювання мікроконтролерів можна запропонувати кілька рекомендацій. По-перше, після симуляції в Proteus доцільно проводити експериментальні вимірювання на фізичних стендах, використовуючи мультиметри та осцилографи, для порівняння результатів з реальними даними.

По-друге, для точнішого аналізу енергоспоживання можна доповнити Proteus додатковими інструментами, такими як LTspice, MATLAB Simulink або Altium Designer, які забезпечують детальнішу модель електричних процесів.

Третім кроком є розширення симуляції енергоспоживання шляхом включення моделей батарей, джерел живлення та реальних навантажень, що зробить результат більш реалістичним. Врахування температурних змін також є важливим, тому додавання температурних коефіцієнтів допоможе коригувати отримані значення споживаного струму.

І, нарешті, комбінування апаратних вимірювань із програмним моделюванням дозволить отримати більш точні результати та оцінити можливі похибки симуляції.

## ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено метод симуляції енергоспоживання мікроконтролера ATmega328P з використанням віртуального моделювання в середовищі Proteus. Метод дозволяє оцінити ефективність різних режимів роботи пристрою без потреби створення фізичного прототипу. Отримано такі результати:

1) проаналізовано сучасні підходи до зниження енергоспоживання у мікроконтролерах, зокрема методи керування режимами роботи, зменшення тактової частоти та вимкнення непотрібних периферійних модулів;

2) розроблено концепцію симуляційного методу оцінки енергоспоживання мікроконтролера з урахуванням змін режимів роботи (Active, Sleep, Deep Sleep), тактової частоти та напруги живлення;

3) створено програмну модель симуляції у середовищі Proteus, що дозволяє аналізувати миттєве та середнє енергоспоживання без апаратного втручання;

4) проведено експериментальні дослідження, результати яких підтвердили ефективність розробленого методу: встановлено, що застосування режимів енергозбереження дозволяє знизити споживання струму до 90% у порівнянні з активним режимом.

Розроблений метод симуляції енергоспоживання показав свою ефективність і зручність у практичному використанні. Він дозволяє ще на етапі проектування оцінити вплив різних режимів роботи мікроконтролера на споживання енергії та прийняти оптимальні рішення для створення енергоефективних пристроїв. Отримані результати можуть бути корисними як для подальших досліджень, так і для реальних інженерних задач.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Dar K.S., Asif S., Islam A. Power Management and Green Computing: An Operating System Prospective. *Can. Int. J. Soc. Sci. Educ.* 2015. P.164-183.
2. Thakkar, A., Chaudhari, K., Shah M. A Comprehensive Survey on Energy-Efficient Power Management Techniques. *Procedia Comput. Sci.* 2020. P.1189-1199.
3. Adegbiya T., Rogacs A., Patel C., Gordon-Ross A. Microprocessor Optimizations for the Internet of Things: A Survey. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 2018.
4. Sun, L., Sheshadri K., Zheng W., Koutsonikolas D. Modeling WiFi Active Power/Energy Consumption in Smartphones. *In Proceedings of the 2014 IEEE 34th International Conference on Distributed Computing Systems, Madrid, Spain.* 2015.
5. Martinez B., Montón M., Vilajosana I. *The Power of Models: Modeling Power Consumption for IoT Devices.* *IEEE Sens. J.* 2015.
6. Yoon C., Lee S., Choi Y. Accurate power modeling of modern mobile application processors. *J. Syst. Archit.* 2017. P.17-31.
7. Alawnah S., Sagahyroon A. Modeling of smartphones' power using neural networks. 2017.
8. Anantha C., Denis C., Joyce C., Yogesh K. Next generation micro-power systems. *In Proceedings of the 2088 IEEE Symposium on VLSI Circuits, Honolulu, HI, USA.* June 2018. P. 2-5.
9. Gollakota S., Reynolds S., Smith .R., Wetherall J. The Emergence of RF-Powered Computing. *Computer.* 2020.
10. Gudan K., Chemishkian S., Hull J., Reynolds M., Thomas S. Feasibility of wireless sensors using ambient 2.4GHz RF energy. *In Proceedings of the 2012 IEEE SENSORS, Taipei, Taiwan.* October 2015.
11. Li C., Zhang W., Cho B. SolarCore: Solar energy driven multi-core architecture power management. *In Proceedings of the 2016 IEEE 17th International Symposium on High Performance Computer Architecture, San Antonio, TX, USA.* February 2016.

12. Raghunathan V., Alaskan A., Hsu J., Friedman J., Srivastava M. Design considerations for solar energy harvesting wireless embedded systems. *In Proceedings of the IPSN 2005 Fourth International Symposium on Information Processing in Sensor Networks, Boise, ID, USA*. April 2015. P. 64.
13. Hua S., Qu G. Approaching the maximum energy saving on embedded systems with multiple voltages. *In Proceedings of the IEEE/ACM International Conference on Computeraided Design, San Jose, CA*. November 2021.
14. Gheorghita V., Basten T., Corporaal H. Application Scenarios in Streaming-Oriented Embedded System Design. *In Proceedings of the International Symposium on System-on-Chip, Tampere, Finland*. 2016.
15. Choi K., Soma R., Pedram M. Fine-grained dynamic voltage and frequency scaling for precise energy and performance tradeoff based on the ratio of off-chip access to on-chip computation times. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* 2019. P.18-28.
16. Saewong S., Rajkumar R. Practical voltage-scaling for fixed-priority RT-systems. *In Proceedings of the 9th IEEE Real-Time and Embedded Technology and Applications Symposium, Toronto, ON, Canada*. 2023.
17. Quan G., Hu X. Energy efficient fixed-priority scheduling for real-time systems on variable voltage processors. *In Proceedings of the 38th Annual Design Automation Conference, Taipei, Taiwan*. October 2021.
18. Kan Y., Chan K., Tse H. Leveraging Performance and Power Savings for Embedded Systems Using Multiple Target Deadlines. *In Proceedings of the 2017 10th International Conference on Quality Software, Zhangjiajie, China*. 14-15 July 2017.
19. STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 Advanced Arm®-based 32-bit MCUs-Reference Manual. URL: [https://www.st.com/resource/en/reference\\_manual/dm00031020-stm32f405415-stm32f407417-stm32f427437-and-stm32f429439-advanced-armbased-32bit-mcus-stmicroelectronics.pdf](https://www.st.com/resource/en/reference_manual/dm00031020-stm32f405415-stm32f407417-stm32f427437-and-stm32f429439-advanced-armbased-32bit-mcus-stmicroelectronics.pdf) (дата звернення: 04.01.2025).

20. Hoeller A.S., Wanner L.F., Fröhlich A.A. A Hierarchical Approach for Power Management on Mobile Embedded Systems. *In Proceedings of the IFIP Working Conference on Distributed and Parallel Embedded Systems*. 2016.
21. Huang K., Santinelli L., Chen J., Thiele L., Buttazzo G.C. Adaptive power management for real-time event streams. *Asia and South Pacific Design Automation Conference (ASP-DAC)*. 2019.
22. Bhatti K., Belleudy C., Auguin M. Power Management in Real Time Embedded Systems through Online and Adaptive Interplay of DPM and DVFS Policies. *In Proceedings of the 2018 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*. 2018.
23. Niu L., Quan G. Reducing both dynamic and leakage energy consumption for hard real-time systems. *Compilers, Architecture, and Synthesis for Embedded Systems*. 2018.
24. Liang J.M., Chen J.J., Cheng H.H., Tseng Y.C. An energy-efficient sleep scheduling with QoS consideration in 3GPP LTE-advanced networks for Internet of things. *IEEE J. Emerging Sel. Topics Circuits Syst*. 2016. P.13-22.
25. Ahmad R.W., Gani A., Naveed A., Rodrigues J.J. A case and framework for code analysis-based smartphone application energy estimation. *Int. J. Commun. Syst*. 2016.
26. Device for Measuring Instant Current Values of CPU's Energy Consumption. *Proceedings of the fifth international workshop on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS'2009)*. 2018.
27. A. Borovyi, V. Konstantakos, V. Kochan. Analysis of Circuits for Measurement of Energy of Processing Units. *Proceedings of the fourth IEEE international workshop on Intelligent Data Acquisition and Advancing Computing Systems*. 2017.
28. Vivek Tiwari, Sharad Malik, Andrew Wolfe, Tien. Instruction level power analysis and optimization of software. *Journal of VLSI Signal Processing*. 2016.

29. Chen N. Research and Implementation of Low Power Technology for Embedded Real-Time Operating System. *Master Thesis, University of Electronic Science and Technology of China, Chengdu, China*, 2016.
30. Optimize Options (Using the GNU Compiler Collection (GCC)). URL: <https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html> (дата звернення: 07.01.2025).
31. Arm Compiler User Guide. URL: <https://developer.arm.com/documentation/100748/0610> (дата звернення: 07.01.2025).
32. Chung E.Y., Benini L., Micheli G.D. Source code transformation based on software cost analysis. *International Symposium on Systems Synthesis*, 2021.
33. Dzhagaryan A., Milenković A. Impact of thread and frequency scaling on performance and energy in modern multicores: A measurement-based study. *In Proceedings of the 2016 ACM Southeast Regional Conference*. 2016.
34. Kim W., Gupta M.S., Wei G., Brooks D. System level analysis of fast, per-core DVFS using on-chip switching regulators. *IEEE 14th International Symposium on High Performance Computer Architecture*. 2018.
35. Jiangwei H. Partitioning the Program into different regions using dynamic and Static Approach with kernel-Assisted in Power management for Embedded System. *IEEE International Conference on Information Reuse & Integration*. 2016.
36. Pinheiro D., Valentin E., Oliveira H., Barreto R. Inserting DVFS Code in Hard Real-Time System Tasks. *In Proceedings of the 2017 VII Brazilian Symposium on Computing Systems Engineering*. 2017. P. 23-30.
37. Kuehn P.J., Mashaly M. DVFS-Power Management and Performance Engineering of Data Center Server Clusters. *In Proceedings of the 2019 15th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. 2019. P. 91-98.
38. Wu H. An Automatic Energy Consumption Measuring Platform for Embedded Systems. *International Conference on Information Science and Control Engineering (ICISCE)*. 2019.

39. Wu H. *Two Designs of Automatic Embedded System Energy Consumption Measuring Platforms Using GPIO*. 2020.
40. Series 2280S Precision Measurement, Low Noise, Programmable DC Power Supplies Datasheet. URL: [https://download.tek.com/datasheet/1KW-50894-0\\_2280S\\_Datasheet\\_040219.pdf](https://download.tek.com/datasheet/1KW-50894-0_2280S_Datasheet_040219.pdf) (дата звернення: 07.01.2025).
41. Elecia White. *Making Embedded Systems: Design Patterns for Great Software*. 2016.
42. Žilvinas Nakutis. Embedded systems power consumption measurement methods overview. *MATAVIMAI*. 2019.
43. Luca Benini and Giovanni de Micheli. System-level power optimization: techniques and tools. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*. 2020.
44. Sparsh Mittal. A survey of techniques for improving energy efficiency in embedded computing systems. *International Journal of Computer Aided Engineering and Technology*. 2016.
45. Roman Lim, Lothar Thiele. Testbed assisted control flow tracing for wireless embedded systems. *In Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*. 2017. P. 180-158.
46. ARM. *CMSIS-Core, Cortex Microcontroller Software Interface Standard*. 2017.
47. Atmel Corporation. *Power Debugger User Guide*. 2016.
48. IAR Systems. *IAR Debug probes User Guide, seventh edition*. 2017.
49. T. Dirsch, G. Keding. *User Manual PowerScale*. Hitex Development Tools. December 2019.
50. Torkel Glad, Lennart Ljung. *Reglerteori, Flervariabla och olinjära metoder*. 2023.
51. Stefan Schürmans, Diandian Zhang, Dominik Auras, Rainer Leupers, Gerd Ascheid, Xiaotao Chen, Lun Wang. Creation of ESL power models for communication architectures using automatic calibration. *In Proceedings of the 50th Annual Design Automation Conference*. 2019.

52. Stefan Schürmans, Gereon Onnebrink, Rainer Leupers, Gerd Ascheid, Xiaotao Chen. Frequency-aware ESL power estimation for ARM Cortex-A9 using a black box processor model. *ACM Transactions on Embedded Computing Systems (TECS)*. 2016.
53. Seo H. Memory-efficient implementation of ultra-lightweight block cipher algorithm CHAM on low-end 8-bit AVR processors. *J. Korea Inst. Inf. Secur.* 2018.
54. Kim Y., Seo S.C. *An Efficient Implementation of AES on 8-Bit AVR-Based Sensor Nodes*. 2020.
55. Fredrik G., Lennart L., Mille M. *Signalbehandling*. 2017.
56. Amanda Nordhamn. Design and implementation of a demonstrator for a Bluetooth low energy based fleet service system for hand-held gardening and forestry products. 2016.
57. Anna L., Marika K., Nilsánu P. Praktisk im-plementering av fjärrstyrt. *Baserad på RTOS-teknik och Bluetooth low energy*. 2016.
58. Andrew Aziz, Ludwing Franquiz, Max Wijnbladh, Eric Wang(Yiqing), Maria Namaghi Rajabzadeh, Johan Snider, Sami Kaivonen, Adrian Amigues, Eirini Petraki. Design, development and deployment of an information-centric networking based solution for the internet-of-things. 2017.
59. Nie, Tingyuan, Lijian Zhou, Zhe-Ming Lu. *Power evaluation methods for data encryption algorithms*. 2016.
60. Srinath N. K. 8085 Microprocessor: Programming and Interfacing. 2018
61. Jiang, Jiahui, Tao Zhang, Daolian Chen. Analysis, design, and implementation of a differential power processing DMPPT with multiple buck-boost choppers for photovoltaic module. *IEEE Transactions on Power Electronics*. 2021.
62. Fezari, Mohamed, Ali Dahoud. *Integrated development environment IDE for Arduino*. 2018.
63. Burch Carl. Logisim: A graphical system for logic circuit design and simulation. *Journal on Educational Resources in Computing (JERIC)*. 2022.
64. Brewer E., Chrysanthos N., Adrian C., William E. Weihl. *Proteus: A high-performance parallel-architecture simulator*. 2020.

65. Hranov, Hristov, Hinov. *Using Logisim Digital Circuit Simulator for Examining Power Electronics Digital Control Systems*. 2020.
66. Asparuhova Katya, Daniela Shehova, Slavi Lyubomirov. *Using Proteus to support engineering student learning: Microcontroller-driven sensors case study*. 2018.
67. Charntaweekhun Kanis, Somkiat Wangsiripitak. *Visual programming using flowchart*. 2016.
68. Nie Tingyuan, Lijian Zhou, Zhe-Ming Lu. *Power evaluation methods for data encryption algorithms*. 2015
69. Shah A., Nasir H., Fayaz M., Lajis A., Shah A. A Review on Energy Consumption Optimization Techniques in IoT Based Smart Building Environments. *Information*. 2019.
70. Martinez B., Monton M., Vilajosana I., Prades, J. D. The Power of Models: Modeling Power Consumption for IoT Devices. *IEEE Sensors Journal*. 2015. P. 15.
71. Ateeq M., Ishmanov F., Afzal M., Naeem M. MultiParametric Analysis of Reliability and Energy Consumption in IoT: A Deep Learning Approach. *Sensors*. 2019. P. 309.
72. Bideh P. N., Sönnerup J., Hell M. Energy consumption for securing lightweight IoT protocols. *Proceedings of the 10th International Conference on the Internet of Things*. 2020.
73. Mocnej J., Miškuf M., Papcun P., Zolotová I. *Impact of Edge Computing Paradigm on Energy Consumption in IoT*. *IFAC-PapersOnLine*. 2018. P. 162-167.
74. Ahmed Z. E., Hasan M. K., Saeed R. A., Hassan,R., Islam S., Mokhtar R. A., Khan S., Akhtaruzzaman M. Optimizing Energy Consumption for Cloud Internet of Things. *Frontiers in Physics*. 2020. P. 8.
75. Sun H., Yu H., Fan G. Towards Energy and Time Efficient Resource Allocation in IoT-Fog-Cloud Environment. *Lecture Notes in Computer Science*. 2019. P. 387-393.
76. Kaur K., Garg S., Kaddoum G., Bou-Harb E., Choo K. A Big Data-Enabled Consolidated Framework for Energy Efficient Software Defined Data Centers in IoT Setups. *IEEE Transactions on Industrial Informatics*. 2020.

77. Fidrovska N., Slepuzhnikov E., Varchenko I. *Norwegian Journal of development of the International Science*. 2020.
78. V. Tiwari. *Instruction Level Power Analysis and Optimization of Software*. 2020.
79. S. Nikolaidis, N. Kavvadias, Th. Laopoulos. *Instruction-level power measurement methodology*. 2020.
80. S. Nikolaidis, N. Kavvadias, Th. Laopoulos, L. Bisdounis and S. Blonias. *International Workshop Power and Timing Modeling, Optimization and Simulation: proceedings*. 2023.
81. Брицький В.В., Медзатий Д.М.. Метод симуляції енергоспоживання мікроконтролера / Збірник наукових праць за матеріалами XVI Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2024». Хмельницький, 2024, С. 71-72.

## ДОДАТОК А

Публікація за результатами роботи

Сертифікат № 2024-050-1



Міністерство освіти і науки України  
Хмельницький національний університет

# СЕРТИФІКАТ



## Брицький Владислав Володимирович

учасник XVI Всеукраїнської науково-практичної конференції  
«Актуальні проблеми комп'ютерних наук АПКН-2024»  
24 години участі (0,8 ECTS credits)

Голова оргкомітету АПКН-2024

**Олег СИНЮК**

проректор Хмельницького національного  
університету з наукової роботи,  
доктор технічних наук, професор

м. Хмельницький  
15-16 листопада 2024

E-mail: [apkt.khnu@gmail.com](mailto:apkt.khnu@gmail.com)

УДК 004.4

Брицький В.В., Медзатий Д.М.

*Хмельницький національний університет***МЕТОД СИМУЛЯЦІЇ ЕНЕРГОСПОЖИВАННЯ МІКРОКОНТРОЛЕРА**

**Актуальність.** У сучасних умовах стрімкого розвитку мікроелектроніки та зростання кількості автономних пристроїв особливої уваги набуває проблема ефективного використання енергоресурсів. Мікроконтролери є основою багатьох вбудованих систем, які працюють на обмежених джерелах енергії, таких як акумулятори чи сонячні батареї. Однак, через складність архітектури та варіативність режимів роботи мікроконтролерів прогнозування їх енергоспоживання стає значним викликом [1]. Для оптимізації роботи таких пристроїв необхідно застосовувати методи симуляції, які дозволяють точно оцінити енергоспоживання на етапі розробки [2].

**Метою** даної роботи є розробка та впровадження методу симуляції енергоспоживання мікроконтролера, що дозволить точно оцінювати витрати енергії в різних режимах роботи, оптимізувати використання ресурсів та забезпечити ефективність автономних систем із мінімальними енергетичними втратами.

**Основні положення.**

Розробка методу симуляції енергоспоживання мікроконтролера ґрунтується на детальному аналізі апаратної архітектури та програмного забезпечення, які впливають на витрати енергії в різних режимах роботи. Цей метод дозволяє комплексно враховувати всі аспекти роботи мікроконтролера, включно з такими параметрами, як активне навантаження, рівні напруги, температурний режим і особливості використання периферійних пристроїв.

Симуляція передбачає використання програмного моделювання, яке дозволяє імітувати різні сценарії роботи мікроконтролера. Зокрема, алгоритми симуляції враховують специфіку виконання типових операцій, оптимізацій програмного коду, а також динамічне регулювання напруги та частоти (DVFS). Цей підхід дозволяє адаптувати роботу мікроконтролера до поточних потреб і зменшувати витрати енергії залежно від завантаженості системи.

Ще одним важливим аспектом є інтеграція моделей прогнозування енергоспоживання. Завдяки збору даних про попередню роботу пристрою можна визначити закономірності та прогнозувати витрати енергії на основі очікуваних

сценаріїв. Це дозволяє забезпечити максимальну енергоефективність, уникаючи надмірного споживання енергії в умовах низького навантаження.

**Висновки.** Розроблений метод симуляції енергоспоживання мікроконтролера є важливим кроком у напрямку оптимізації роботи сучасних вбудованих систем. Застосування методу дозволяє оцінювати енергетичні витрати на різних етапах розробки, що суттєво підвищує ефективність проектування автономних пристроїв. Важливість такого методу зумовлена постійним зростанням попиту на пристрої з низьким енергоспоживанням у сферах промисловості, медицини, транспорту та побуту. Використання розроблених підходів дозволяє не лише зменшити енергетичні витрати, але й продовжити термін служби автономних пристроїв. Це сприяє підвищенню їхньої надійності, функціональності та економічної доцільності.

#### **Перелік посилань**

1. Олексієнко М. С., "Методи моделювання енергоспоживання мікропроцесорних систем." 2022.
2. Бойко В. П., "Оптимізація режимів енергоспоживання вбудованих систем на основі мікроконтролерів." 2021.
3. Types and Applications of Microcontrollers. Доступ до ресурсу: <https://www.eit.edu.au/resources/types-and-applications-of-microcontrollers>.
4. Huang J., Patel K. Energy-Efficient Microcontroller Design: Methods and Applications. IEEE Transactions on Embedded Systems. 2019. (дата звернення: 18.11.2024).

## ДОДАТОК Б

### ПРОГРАМНА РЕАЛІЗАЦІЯ УПРАВЛІННЯ РЕЖИМАМИ РОБОТИ МІКРОКОНТРОЛЕРА АТМЕГА328Р ДЛЯ АНАЛІЗУ ЕНЕРГОСПОЖИВАННЯ

```
#include <avr/io.h>
#include <util/delay.h>
#include <avr/sleep.h>
void enter_sleep_mode() {
    set_sleep_mode(SLEEP_MODE_PWR_DOWN);
    sleep_enable();
    sleep_cpu();
}
int main(void) {
    DDRB |= (1 << PB0);
    DDRD &= ~(1 << PD2);
    PORTD |= (1 << PD2);
    for (int i = 0; i < 10; i++) {
        PORTB ^= (1 << PB0);
        _delay_ms(1000);
    }
    enter_sleep_mode();
    while (1);
}
```

**Anti-Plagiarism v-15.260 Educational****Максимальне співпадіння з одним документом 0.0%**

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 14%

ID: 229502 Назва: МКР Метод симуляції енергоспоживання мікроконтролера Додано в БД: 2025-04-15 Автора: Владислав Брицький Керівник: Дмитро Медзятий Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	112238	892	1569 (1%)	24 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

### 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: Владислав Брицький

Співавтор:

Назва: БРИЦЬКИЙ\_Метод симуляції енергоспоживання мікроконтролера

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:9.7%

Коефіцієнт подібності 2:3.3%

Мікропробіли: 5

Заміна букв: 12

Інтервали: 0

Білі знаки: 131

Дата створення звіту: 2025-04-15 13:29:39.0

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. 3У Про вищу освіту, пункт 3.1, Ст. 42. 3У Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-04-15

Дата



Доцент Андрій Нічепорук

експерт

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Владислав Брицький

Тема: Метод симуляції енергоспоживання мікроконтролера

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість листів креслень —; кількість сторінок записки 73

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано метод симуляції енергоспоживання мікроконтролера ATmega328P із використанням середовища Proteus.

2. Висновок про відповідність роботи дипломному завданню \_\_\_\_\_

Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено огляд існуючих підходів до вимірювання та аналізу енергоспоживання мікроконтролерів. Визначено основні параметри, що впливають на споживання енергії. У другому розділі проведено детальне дослідження особливостей моделювання енергоспоживання AVR-мікроконтролерів. Обґрунтовано вибір мікроконтролера ATmega32. У третьому розділі розроблено метод симуляції енергоспоживання мікроконтролера з використанням програмного середовища Proteus. У четвертому розділі розроблено електронну модель з можливістю імітації перемикання режимів мікроконтролера.

4. Позитивні сторони роботи: Запропонований метод симуляції енергоспоживання мікроконтролера в середовищі Proteus дозволяє на ранньому етапі розробки оцінити вплив різних режимів роботи на загальне енергоспоживання пристрою.

5. Негативні сторони роботи: Дослідження обмежуються лише мікроконтролером ATmega328P, що не дозволяє зробити узагальнені висновки для інших архітектур мікроконтролерів.

6. Оцінка графічного оформлення та пояснювальної записки роботи: -

7. Відгук про роботу в цілому: В загальному робота виконана на невисокому рівні.


8. Інші зауваження: -

9. Оцінка кваліфікаційної роботи магістра:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «задовільно» 3.00 (E)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) к.т.н., доцент, Федула М.В.

“ 1 травня ” \_\_\_\_\_ 2025р.



Завідувачу кафедри КПС  
доктору філософії, доценту  
Ользі ПАВЛОВІЙ

Брицького Владислава Володимировича

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2М-23-2

#### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

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

22 квітня 2025 року



**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ**  
**КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод симуляції енергоспоживання мікроконтролера

Автор: Брицький Владислав Володимирович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Медзатий Дмитро Миколайович, к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Виявлені у роботі запозичення є допустимими та не становлять плагіату з наступних причин:

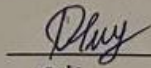
1) окремі співпадіння стосуються загальноживаних словосполучень або фраз, що підтверджується тим, що система вказує на збіг одного фрагмента з великою кількістю джерел;

2) зафіксовані системою випадки так званої модифікації тексту стосуються виключно поєднання латинських символів з українськими скороченнями в формулах, що не може вважатися зміною змісту;

3) частина списку використаних джерел було визначено як плагіат.

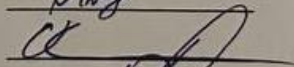
Загальний обсяг виявлених збігів, встановлений системою перевірки, становить менше ніж 9,72% і походить з відкритих інтернет-ресурсів та електронної бібліотеки. З огляду на вищезазначене, такий рівень запозичень є припустимим і відповідає вимогам до кваліфікаційної роботи.

Керівник роботи



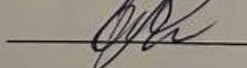
Дмитро МЕДЗАТИЙ

Гарант ОП



Олег САВЕНКО

Завідувач кафедри КІС



Ольга ПАВЛОВА