

УДК 629.783:004.7:621.3.051.025

doi: 10.32620/aktt.2024.3.10

І. Б. ТУРКІН, О. В. ЛЮБІМОВ, В. О. ЗАХАРЕНКО

*Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут», Україна*

## ОПТИМІЗАЦІЯ ЕНЕРГОСПОЖИВАННЯ БОРТОВОГО КОМП'ЮТЕРА CUBESAT ЗА УМОВ ОБМЕЖЕНЬ РЕАЛЬНОГО ЧАСУ

**Об'єкт** дослідження – енергоспоживання бортового комп'ютера наносупутника CubeSat. **Предметом** дослідження є модель енергоспоживання бортового комп'ютера та метод оптимізації його енергетичної ефективності. **Мета** роботи – розробити модель енергоспоживання бортового комп'ютера та метод оптимізації його енергетичної ефективності з наступною експериментальною перевіркою ефективності цього метода. **Завдання:** обґрунтувати доцільність пошуку нових методів оптимізації енергетичної ефективності бортового комп'ютера наносупутника CubeSat; виконати підготовку до проведення експерименту, а саме розробити модель енергоспоживання бортового комп'ютера та метод оптимізації його енергетичної ефективності, розробити план експерименту та методику вимірювань; провести експериментальні дослідження та представити основні результати експерименту; надати змістовну інтерпретацію отриманих експериментальних результатів; узагальнити висновки, сформулювати переваги та недоліки даної роботи, запропонувати напрямки подальших досліджень. **Висновки.** Енергоспоживання бортового комп'ютера наносупутника необхідно зменшувати апаратними та програмними засобами. У статті досліджено ефективність відомих методів енергозбереження для авторської обчислювальної платформи «Falco SBC 1.0». Такими методами є динамічне масштабування частоти (DFS), алгоритм race-to-dark (RTD) та комбінований алгоритму, який виявився найефективнішим. Результати ідеалізовані, оскільки не враховують накладні витрати ОС. Методологія може застосовуватись для оцінки інших платформ. Наступні дослідження будуть розглядати такі напрями: енергоспоживання в різних режимах енергозбереження процесора, накладні витрати багатопотокової операційної системи реального часу, керування живленням непроцесорних компонентів. Вирішення цих проблем залишається важливою областю сучасних наукових досліджень.

**Ключові слова:** CubeSat; наносупутник; ефективність; СБК; Falco; SSA; модель енергоспоживання; потужність; ATSAMV71; DFS; RTD.

### Вступ

#### 1.1. Концепція CubeSat

Наносупутники в стандарті CubeSat [1], перший з яких – CalPoly був запущений у 1999 році Каліфорнійським політехнічним університетом, стали дуже популярним способом дослідження ближнього космосу та низьких навколоземних орбіт (ННО). Використовуючи великі космічні ракети у якості носіїв і, як наслідок, дешеви́зну виведення на орбіту завдяки кількості виводимих одиниць, наносупутники CubeSat сьогодні використовуються академічними та науковими установами, комерційними та військовими фірмами. З початку програми на низькі навколоземні орбіти було запущено понад 2400 супутників [2].

Типовий CubeSat — це підклас наносупутників, зазвичай розміром від 1U до 12U, де 1U дорівнює 10x10x10 см<sup>3</sup>, хоча на сьогодні вже існують проекти

CubeSat 27U [3]. Концепція CubeSat полягає в тому, щоб усю авіоніку малих космічних кораблів, систему електроживлення, корисне навантаження, а за необхідності й рушійні двигуни, розмістити у внутрішній об'єм прямокутного паралелепіпеда, який складається з кубиків 1U. Зовнішня поверхня CubeSat зазвичай використовується для розміщення сонячних панелей, антен, оптичних та електромеханічних датчиків.

Наносупутники CubeSats швидко вийшли за межі своєї початкової ролі недорогих інструментів, які використовуються в освітньому процесі та демонструють можливості сучасних технологій. Сьогодні вони служать платформами для високоякісних наукових експериментів, комерційних застосувань [4, 5], є основою для угруповань (сузір'їв) супутників зв'язку [6], використовуються в військових цілях [7]. Ця тенденція підтримана прогресом у мініатюризації та вищим рівнем інтеграції апаратного забезпечення, зростанням обчислювальної потужно-

сті та застосування досліджень та інновацій протягом останньої чверті століття. Відкриваючи нові можливості, CubeSat розширює горизонти для дослідницького співтовариства компаній, науковців і національних космічних агентств, зокрема з метою підвищення надійності та ефективності технологій, а також запровадження комплексного підходу до процесів розробки, тестування та введення в експлуатацію [8].

### **1.2. Генерування та збереження електричної енергії як головні обмеження наносупутника**

Однією з головних проблем будь-якого супутника, а особливо наносупутника CubeSat, є його рівень наявної енергетичної забезпеченості, оскільки енергія – це найбільш дефіцитний ресурс протягом усього життєвого циклу супутника. Складність CubeSat зростає, для нього планується більше корисного навантаження, йому потрібно більше енергії, щоб мати можливість отримувати, обробляти та передавати на наземні станції результати роботи наукового обладнання, що складає його корисне навантаження.

Той факт, що на ННО є сотні супутників розміром 1U, означає, що типової потужності чотирьох сонячних панелей розміром 10x10 см достатньо для роботи супутника. Є навіть супутники 0,25U, які здатні виконувати свої місії з дуже обмеженою кількістю сонячної енергії. Однак, проаналізувавши популярність супутників різних розмірів, найпопулярнішим розміром залишається 3U, оскільки він забезпечує як можливість для розміщення складнішого корисного навантаження, так і більшу кількість отриманої енергії завдяки сонячним панелям.

Типовий рівень енергопостачання від сонячних панелей на освітленій ділянці орбіти для CubeSat типорозміром від 1U до 3U становить від 2 до 8 Вт, при цьому академічні супутники, безумовно, знаходяться на нижній межі цього діапазону. У той же час CubeSats, які займаються обробкою зображень і базуються на FPGA, можуть споживати до 12,5 Вт енергії [9]. Проблема балансу необхідної та доступної електричної потужності вимагає серйозного підходу до проектування з урахуванням енергетичних обмежень, а також перевірки та валідації всієї авіоники малого космічного корабля (small spacecraft avionics – SSA).

Головним ключовим компонентом є сама система електрозабезпечення (Electric power system – EPS), яка відповідає за перетворення сонячної енергії в електричну, збереження енергії та енергопостачання споживачів. Різні моделі

ефективності та управління EPS добре описані в комплексних дослідженнях [10 – 13] і загалом проєктні рішення достатньо оптимізовані. Через складність розробки, тестування, перевірки та валідації EPS більшість розробників CubeSat використовують готові комерційні продукти (Commercial off-the-shelf – COTS), тому надалі в статті питання енергозабезпечення не розглядатиметься, мова буде йти тільки про енергоспоживання бортового комп'ютера.

Типовий CubeSat є складною технічною системою, його складові споживають електричну енергію. Зазвичай на більшості плат CubeSat є власний мікроконтролер (micro-controller unit – MCU), але всі вони керуються з єдиного центру: бортового комп'ютера (On Board Computer – OBC). OBC відповідає за загальну злагоджену роботу електроніки та програмного забезпечення і, отже, є ключовим компонентом для отримання або втрати енергії.

Незалежно від схемотехнічного рішення OBC: це може бути декілька мікроконтролерів, або комбінація центрального процесора та FPGA [9], або, навіть, міні-комп'ютер на базі звичайного смартфона [14], всі OBC повинні бути енергетично ефективними, тобто споживати мінімум енергії під час роботи.

Існує кілька ключових підходів до того, як збалансовано використовувати енергію в OBC. Це і розумне планування обчислювальних завдань, і використання відомих змішаних апаратних і програмних методів забезпечення енерго-ефективності обчислень.

Оптимізація плану обчислень зосереджується на підході, який допомагає спланувати завдання так, щоб вони виконувались тоді, коли енергія доступна, тобто під час генерування енергії сонячними панелями на освітленій ділянці орбіти, або при достатньому рівні зарядженості вторинного джерела енергії – хімічної батареї. В цьому випадку добре працює динамічне планування за пріоритетами, коли частина завдань може бути призупинена до більш сприятливих умов. Багато з таких алгоритмів планування базуються на типовому плануванні завдань в операційних системах (ОС) на основі пріоритетів, як-от оцінка незадовільної пропускної спроможності системи (unmet system capacity – USC) [15]. Деякі алгоритми планування використовують навіть глибоке навчання з підкріпленням (deep reinforcement learning – DRL) або більш просунуті методи, що базуються на теорії графів, такі як метод гілок, відсікання та оцінювання [16]. Багато з цих алгоритмів сильно навантажені необхідністю вирішення завдань змішано-цілочисельного лінійного програмування (mixed-

integer linear programming – MILP) [17] і, отже, самі по собі досить вимогливі до ресурсів. Іншим недоліком більшості переглянутих підходів є бажання знайти загальне оптимальне рішення, яке не залежить від конкретної платформи та/або операційної системи.

Методи оптимізації енергоспоживання другого типу засновані на інтегрованому керуванні апаратно-програмним забезпеченням. Вони використовують різноманітні режими живлення MCU, стратегії керування синхронізацією та загальне інтелектуальне керування живленням периферійних пристроїв.

В огляді [18] наведено наступну класифікацію методів і технологій енергозбереження зазначеного вище типу.

1. Методи динамічного масштабування напруги та частоти (DVFS – Dynamic voltage and frequency scaling) та планування з урахуванням потужності.

2. Використання режимів низького енергоспоживання через динамічне керування живленням (DPM – Dynamic Power Management) [19].

3. Мікроархітектурні прийоми для збереження енергії в окремих компонентах, таких як пам'ять.

4. Багатопроцесорні архітектури бортового обчислювача, що дозволяє використовувати тільки необхідні обчислювальні ресурси [20].

5. Використання спеціалізованих процесорів, таких як DSP, GPU або FPGA.

### 1.3. Мета і завдання

Метою роботи є розробка моделі енергоспоживання бортового комп'ютера та методу оптимізації його енергетичної ефективності з наступною експериментальною перевіркою ефективності цього метода.

Завдання:

- в розділі 1 обґрунтувати доцільність пошуку нових методів оптимізації енергетичної ефективності бортового комп'ютера наносупутника CubeSat;

- в розділі 2 виконати підготовку до проведення експерименту, а саме розробити модель енергоспоживання бортового комп'ютера та метод оптимізації його енергетичної ефективності, розробити план експерименту та методику вимірювань;

- в розділі 3 провести експериментальні дослідження та представити основні результати експерименту;

- в розділі 4 надати змістовну інтерпретацію отриманих експериментальних результатів;

- в розділі 5 узагальнити висновки, сформулювати переваги та недоліки даної роботи, запропонувати напрямки подальших досліджень.

## 2. Модель енергоспоживання бортового комп'ютера та метод оптимізації його енергетичної ефективності

### 2.1. Бортовий комп'ютер на базі платформи CDHM “Falco SBC 1.0”

Щоби досягти енергетичної ефективності бортового комп'ютера, необхідно спроектувати та розробити методи, алгоритми та програмне забезпечення, що будуть керувати живленням бортового комп'ютера та периферійних пристроїв. Дуже рідко буває, коли вся обчислювальна потужність, тобто весь наявний машинний час, потрібна для підтримання всього необхідного обчислювального процесу місії, і, таким чином, слід розглядати оптимізацію потужності в періоди простою системи.

У мікроконтролері з операційною системою FreeRTOS керування в режимі дефіциту енергопостачання здійснюється за допомогою апаратного та програмного забезпечення [21]. Коли завдання очікує переривання або закінчення інтервалу часу, воно блокується. Якщо всі завдання переходять у заблокований стан, FreeRTOS виконає неактивне завдання, або завдання простою, з найнижчим пріоритетом. Тому, коли процесор неактивний, неактивне завдання може перевести процесор у режим енергозбереження. Цей механізм може бути корисним у деяких сценаріях, але якщо тактова частота занадто висока, процесор буде витрачати енергію та час, щоб увійти та вийти з режиму очікування. Тому економія електроенергії за допомогою цього механізму не принесе користі. Щоби покращити механізм енергозбереження була введена техніка холостого ходу за відсутності переривання по таймеру [22]. Ця техніка використовує механізм відстеження часу, щоб вимкнути джерело тактового сигналу генератора частоти на певний період часу, щоб перевести процесор у режим глибокого сну, доки не відбудеться зовнішнє переривання або переривання з боку ядра з вищим пріоритетом.

В роботі [8] описані корисні методи для вимірювання часу, витраченого на такі сервісні операції, як обробка переривань і затримки перемикавання потоків для віртуальних машин. Надалі вони будуть адаптовані для обчислювальної платформи на основі мікросхеми ATSAMV71Q21.

Пристрої на базі мікросхеми ATSAMV71Q21 мають три режими низького енергозбереження, які

можна обрати програмно: режими сну, очікування та резервного копіювання.

1. В режимі сну процесор зупиняється, і всі інші функції можуть працювати.

2. У режимі очікування всі таймери та функції зупиняються, але деякі периферійні пристрої можна налаштувати для пробудження системи на основі попередньо визначених умов. Ця функція під назвою «SleepWalking» виконує часткове асинхронне пробудження, таким чином дозволяючи процесору виходити з режиму сну лише тоді, коли це необхідно.

3. Режим резервного копіювання включає 32-розрядний малопотужний таймер реального часу (RTT), годинник реального часу (RTC) і логіку пробудження. Крім того, пристрій може відповідати найсуворішим вимогам у цьому режимі (Key-Off) при підтримці всього 1 КБ SRAM. До щоби оптимізувати енергоспоживання, система генератора тактової частоти була розроблена для підтримки різних частот для вибраних периферійних пристроїв. Крім того, тактові частоти процесора та інтерфейсних шин можна змінювати без впливу на процес виконання завдань, наприклад USB, U(S)ART, AFE та лічильника таймера [20].

## 2.2. Метод оптимізації енергетичної ефективності бортового комп'ютера

Припустимо, що для системи реального часу ми знаємо час, необхідний для виконання запланованих завдань у певний проміжок часу на максимальній тактовій частоті процесора. Тоді для економії енергії нам потрібно зробити вибір між застосуванням динамічного регулювання частоти або використанням енергозберігаючого режиму, або комбінації цих режимів енергозбереження.

Критерієм оптимізації є мінімальне енергоспоживання бортового комп'ютера при різних режимах енергозбереження:

$$P(\eta) = \min(P_{DFS}(\eta), P_{RTD}(\eta), P_{DFS+RTD}(\eta)), \quad (1)$$

де  $\eta$  – відносні витрати процесорного часу, достатні для забезпечення обмежень реального часу;  $P_{DFS}(\eta), P_{RTD}(\eta), P_{DFS+RTD}(\eta)$  – витрати енергії на роботу бортового комп'ютера при застосуванні таких підходів до енергозбереження:

– динамічне частотне масштабування (DFS) [23];

– алгоритм race-to-dark (RTD) (іноді Race-to-sleep) виконує завдання максимально швидко, щоб

процесор можна було перевести в сплячий режим, який мінімізує струм витоку [20];

– комбінований алгоритм (DFS+RTD), коли програмне забезпечення встановлює мінімальну тактову частоту після роботи на максимальній частоті перед переходом у сплячий режим.

## 2.3. План експерименту

Мета експерименту: Побудова математичних моделей для подальшого вирішення оптимізаційної задачі визначення енергоефективного режиму роботи бортового комп'ютера (1). Таким чином, в результаті експерименту необхідно визначити:

– залежності енергоспоживання бортового комп'ютера від тактової частоти та напруги живлення в активному режимі та режимі сну малого енергоспоживання:

$$P_{Active}(U, f), P_{Sleep}(U, f) \quad (2)$$

– залежність відносного коефіцієнту збільшення часу обчислення від частоти, який визначається через відношення часу, необхідного для виконання певної обчислювальної роботи при тактовій частоті процесора, до часу, витраченого на виконання тієї ж роботи на максимальній частоті:

$$k_T(f) = \frac{T(f)}{T(f_{max})}, \quad (3)$$

де  $T(f), T(f_{max})$  – час, витрачений на роботу на двох частотах.

У статтях [24, 25] викладено декілька суттєвих міркувань щодо залежності енергоспоживання процесора ( $P$ ) від напруги живлення та тактової частоти, основним з яких є наступне: енергоспоживання процесора прямо пропорційне квадрату напруги живлення і пропорційно тактовій частоті:

$$P = ACV^2f + VI_{leak}. \quad (4)$$

У лінійній моделі час ( $T$ ) для вирішення задачі на мікропроцесорі є пропорційним кількості машинних команд ( $n$ ) і обернено пропорційний добутку кількості необхідних циклів на команду ( $c$ ) на тактову частоту:

$$T(f) = \frac{n}{c} \cdot f^{-1}. \quad (5)$$

Відхилення від теоретичної моделі може спостерігатися через дію багатьох факторів, таких як затримка під час взаємодії з пам'яттю або зовнішніми пристроями через їх обмежену швидкість, роботу кеш-пам'яті тощо. В результаті фактичний показник ступеня ( $a_f$ ), отриманий з експериментальних даних, може відрізнятися від теоретичного, який дорівнює  $a_f = -1$ .

$$T(f) = T(f_{\max}) \left( \frac{f}{f_{\max}} \right)^{a_f} \Rightarrow k_T(f) = \left( \frac{f}{f_{\max}} \right)^{a_f}. \quad (6)$$

Експеримент сплановано як повнофакторний, тобто вимірювання проведено для всіх можливих комбінацій тактової частоти процесора –  $f$ , МГц і напруги живлення –  $V$ , В, які дозволені в технічній документації процесора [26]:

$$f \in \{30; 100; 200; 300\}, V \in \{1,8; 2,4; 3,0; 3,3\}.$$

#### 2.4. Техніка вимірювань

Експеримент проводився на платформі, яка виконувала лише одне циклічно повторюване завдання, в якості якого використане завдання сортування цілочисельного масиву за допомогою бульбашкового алгоритму. Після кожного циклу сортування інструкція Wait for Interrupt (WFI) переводить проце-

сор у режим сну; коли функції процесора зупинені, а системні годинники включені, споживання енергії в цьому режимі цілком залежить від роботи обладнання. Далі переривання за таймером викликає пробудження процесора та йде наступне виконання сортування.

Для вимірювання часу виконання алгоритму при різних комбінаціях частоти та напруги використана системна функція FreeRTOS.

Для вимірювання споживаної бортовим комп'ютером електроенергії, використовувалося високоточне стендове джерело живлення постійного струму Keithley 2281S, що гарантує точність вимірювань (похибка часового інтервалу – не гірше 15 мс, похибка потужності споживаної електроенергії – не гірше 0.0001 Вт).

### 3. Результати

Приклад отриманих первинних результатів для побудови залежності енергоспоживання бортового комп'ютера від тактової частоти та напруги живлення в активному режимі та режимі сну зображено на рис. 1. Первинні результати мають досить високу точність; похибки вимірювання інтервалів часу не перевищує 2%, а середньоквадратичне відхилення споживаної потужності не більше ніж 0,1%.

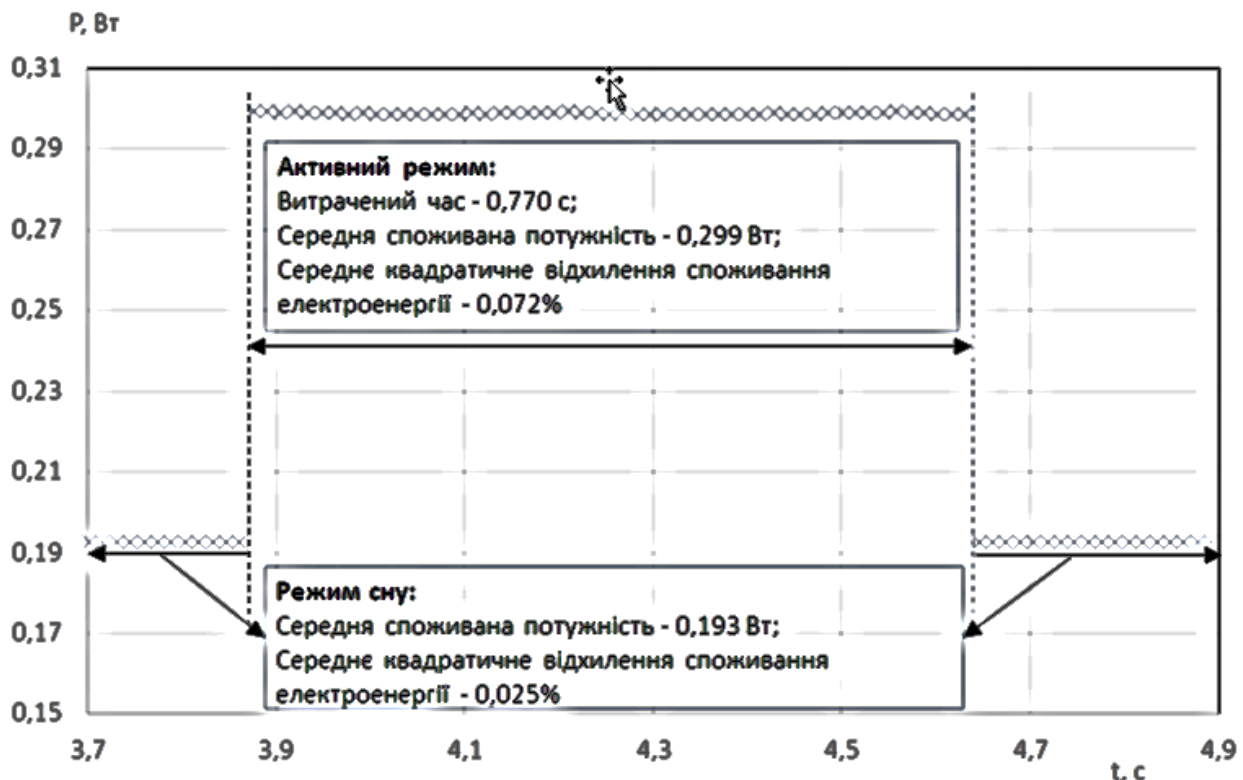


Рис. 1. Первинні експериментальні результати: вимірювання потужності для частоти 300 МГц, напруги 3,3 В

На відміну від рекомендованої моделі енергоспоживання процесора (4), кращої відповідності між результатами вимірювань і обчисленнями вдалось досягти за допомогою наступного рівняння (рис. 2, табл. 1) як для активного режиму, так і для режиму сну:

$$P(V, f) = a_1 Vf + a_2 V^2 + a_3 V. \quad (7)$$

Моделльні коефіцієнти  $a_1$ ,  $a_2$ ,  $a_3$  в (7) є різними для активного та сплячого режимів (табл. 1). Коефіцієнт детермінації ( $R^2$ ) – достатньо високий, щоби стверджувати відповідність моделі результатам вимірювань.

Час, необхідний для виконання конкретної обчислювальної роботи на тактовій частоті, відмінній від максимальної, потребує визначення лише одного коефіцієнта – показника  $a_f = -0,8178$  (рис. 3). Цей коефіцієнт не залежить від напруги живлення, тому кожна точка на графіку відповідає всім чотирьом можливим значенням напруги

$$\{1,8; 2,4; 3,0; 3,3\}.$$

#### 4. Аналіз отриманих результатів

Для зручності аналізу введемо в подальший розгляд коефіцієнт завантаженості процесора -  $\eta$ . В класичній постановці теорії систем реального часу [28] цей коефіцієнт визначається як:

$$\eta = \sum_{i=1}^N \frac{T_i}{D_i}, \quad (8)$$

де  $N$  – кількість завдань, виконуваних в системі реального часу;

$T_i, D_i$  – максимальний час, необхідний  $i$ -му завданню, та період повторення завдання.

Тоді при використанні кожного з режимів енергозбереження середня споживана потужність буде не більшою ніж при безперервній роботі процесора на максимальній частоті. Відносна частину збереженої енергії  $\mu$  для трьох режимів енергозбереження залежить від коефіцієнта завантаженості процесора –  $\eta$  та в випадку ідеального обчислювача розраховується просто:  $\mu = 1 - \eta$ .

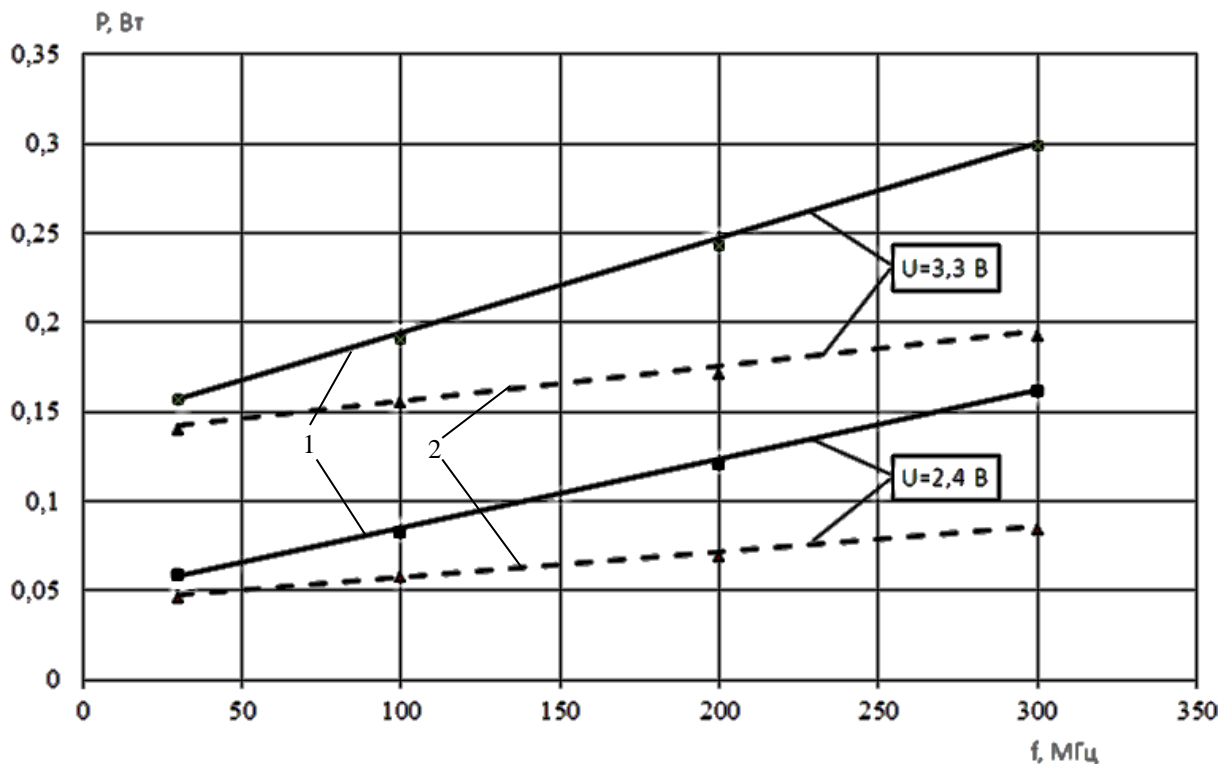


Рис. 2. Експериментальні дані та побудовані модельні розрахунки:  
(1 – активний режим, 2 – режим сну)

Таблиця 1

Моделльні коефіцієнти  $a_1$ ,  $a_2$ ,  $a_3$  та коефіцієнт детермінації –  $R^2$

	$a_1$	$a_2$	$a_3$	$R^2$
Активний режим	0,0001607	0,026005	-0,042979	0,99965
Сплячий режим	0,0000593564	0,025951038	-0,044252153	0,99927

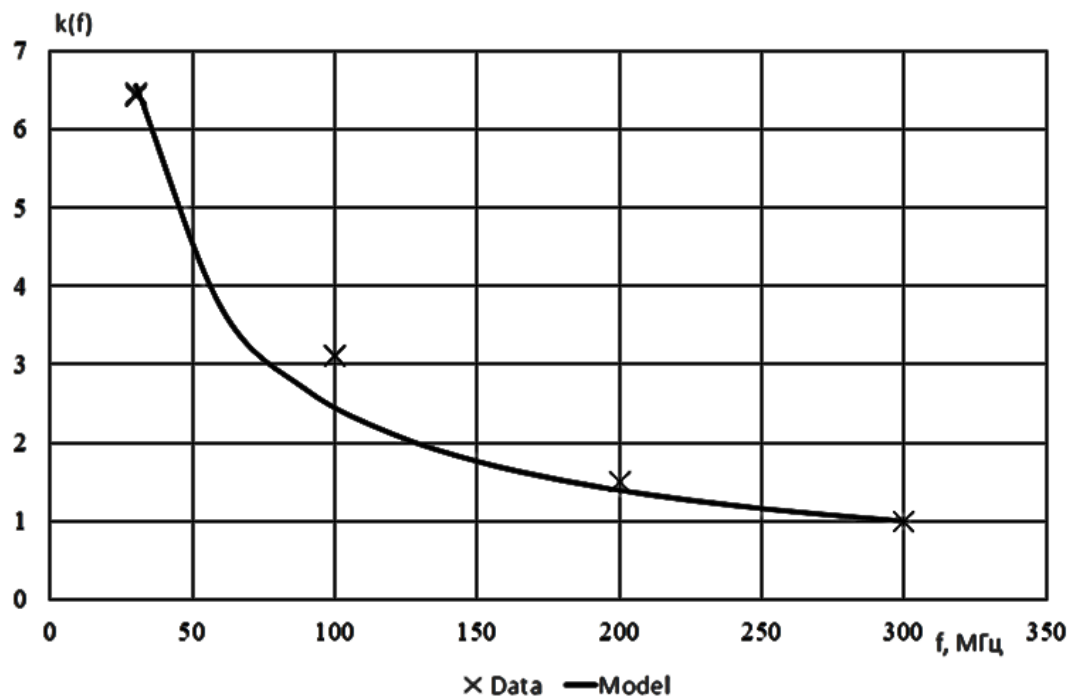


Рис. 3. Відносний коефіцієнт збільшення часу обчислення при зниженні тактової частоти процесора

Для реального обчислювача  $\mu$  можна обчислити через наступні рівняння, які використовують модельні залежності споживаної потужності від частоти та напруги:

$$\begin{aligned} \mu_{DFS}(\eta, V) &= 1 - \frac{P_{Active}(V, f_{max} \cdot \eta^{af})}{P_{Active}(V, f_{max})}, \\ \mu_{RTD}(\eta, V) &= 1 - \eta - \frac{(1 - \eta) \cdot P_{Sleep}(V, f_{max})}{P_{Active}(V, f_{max})}, \\ \mu_{DFS+RTD}(\eta, V) &= 1 - \eta - \frac{(1 - \eta) \cdot P_{Sleep}(V, f_{min})}{P_{Active}(V, f_{max})}. \end{aligned} \quad (9)$$

Основним результатом є оптимістична оцінка можливості енергозбереження за рахунок використання програмного керування режимами роботи комп'ютера, вказаних в підрозділі 2.2. Ця оцінка (рис. 4) є оптимістичною, оскільки не враховує витрати на роботу планувальника та диспетчера операційної системи. Водночас отримані кількісні результати є основою для оптимізації конструкції апаратних компонентів бортового комп'ютера. Отримані кількісні результати свідчать про те, що використання DFS є більш вигідним, ніж RTD, але необхідно враховувати, що використання DFS зменшить реактивність системи. Тому необхідно буде шукати певний компроміс між енергозбереженням і реактивністю системи.

Додатковим результатом роботи є запропонована та перевірена експериментальна методика виконання подібних оцінок та побудови моделі для подальшої оптимізації.

## 5. Висновки

Енергоспоживання бортового комп'ютера студентського наносупутника є критично важливим, його необхідно зменшувати за рахунок оптимізації, як апаратної, так і програмної частини.

У статті порівняно енергозберігаючі можливості конкретної обчислювальної платформи «Falco SBC 1.0» при використанні різних методів керування, а саме динамічного частотного масштабування (DFS), алгоритма gate-to-dark (RTD) і комбінованого алгоритму. Критерієм вибору алгоритму керування є мінімальне енергоспоживання за умов гарантованого виконання системних обмежень реального часу.

Було створено математичні моделі, щоб відобразити залежність споживання бортовим комп'ютером електроенергії від тактової частоти, напруги живлення та часу обчислення, необхідного для запланованих завдань. Ці моделі дозволили сформулювати та розв'язати задачу оптимізації кількісно. Режим DFS є найбільш ефективним рішенням з точки зору енергозбереження. Комбінований спосіб суттєво не відрізняється від DFS, але при цьому погіршується реактивність системи, тому необхідно шукати певний компроміс з урахуванням вимог конкретного застосування.

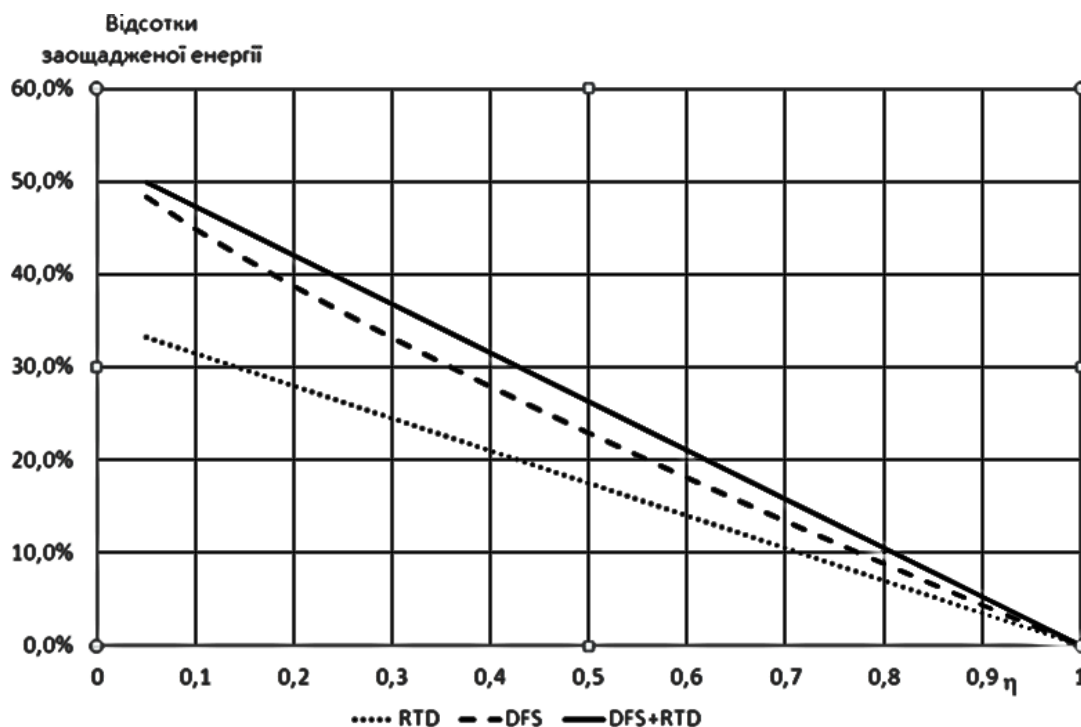


Рис. 4. Залежність збереженої бортовим комп'ютером енергії від  $\eta$  при використанні різних режимів енергозбереження

Недоліком роботи є те, що отримані результати є ідеалізованими, оскільки вони не враховують додаткові накладні витрати планувальника та диспетчера багатозадачної операційної системи.

Тим не менш, розроблена тут методологія для аналізу, моделювання, вимірювання та оптимізації енергоспоживання бортового комп'ютера може бути основою для подальшого використання при оцінці інших апаратних та програмних платформ.

У подальшому розвитку роботи ми плануємо три напрямки досліджень:

- перший – дослідження енергоспоживання комп'ютера в режимах роботи процесора, таких як очікування та резервне копіювання;

- другий і основний – визначення накладних витрат на роботу багатопоточної операційної системи реального часу, що є більш складним завданням. Прогнозувати поведінку системи в реальному часі дуже складно тому що завдання можуть виконуватися повільніше, ніж очікувалося і вони можуть аварійно завершувати роботу під час виконання або мати несподівані затримки;

- третій буде зосереджено на дослідженні гнучкого керування живленням непроецесорних компонентів, тобто пам'яті, комунікаційних інтерфейсів тощо.

Вирішення цих проблем у багатопоточних операційних системах реального часу залишається важливою областю сучасних досліджень.

**Внесок авторів:** основна концепція статті та методологія наукових питань – **І. Б. Туркін, О. В. Любімов**; формування проблеми оптимізації енергоспоживання бортового комп'ютера cubesat та шляхів її вирішення – **І. Б. Туркін, О. В. Любімов, В. О. Захаренко**; планування та проведення наукового експерименту – **О. В. Любімов**; огляд та аналіз англійських інформаційних джерел – **І. Б. Туркін**; форматування та підготовка статті до видання – **В. О. Захаренко**.

### Конфлікт інтересів

Автори заявляють, що немає конфлікту інтересів щодо цього дослідження, фінансового, особистого, авторського чи іншого, який міг би вплинути на дослідження та його результати, представлені в цій статті.

### Фінансування

Дослідження проводилося без фінансової підтримки.

### Доступність даних

Рукопис не має пов'язаних даних.

### Використання засобів штучного інтелекту

Автори підтверджують, що не використовували технології штучного інтелекту при створенні представленої роботи.



Усі автори прочитали та погодились з опублікованою версією рукопису.

## Література

1. *CubeSat Design Specification (1U – 12U). Rev 14.1. The CubeSat Program, Cal Poly SLO [Electronic resource]*. – 2022. – 34 p. – Available at: [https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/62193b7fc9e72e0053f00910/1645820809779/CDS+REV14\\_1+2022-02-09.pdf](https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/62193b7fc9e72e0053f00910/1645820809779/CDS+REV14_1+2022-02-09.pdf). (Accessed 28.04.2023).
2. Kulu, E. *World's largest database of nanosatellites, over 4100 nanosats and CubeSats [Electronic resource]* / E. Kulu // *Nanosats Database*. – Available at: <https://www.nanosats.eu/#figures>. (Accessed 29.11.2023).
3. *Space Microgrids: New Concepts on Electric Power Systems for Satellites [Text]* / A. Lashab, M. Yaqoob, Y. Terriche, J. C. Vasquez, & J. M. Guerrero // *IEEE Electrification Magazine*. – 2020. – Vol. 8, Iss. 4. – P. 8-19. DOI: 10.1109/MELE.2020.3026436.
4. *Towards the Thousandth CubeSat: A Statistical Overview [Text]* / T. Villela, C. A. Costa, A. M. Brandão, F. T. Bueno, & R. Leonardi // *International Journal of Aerospace Engineering*. – 2019. – Vol. 2019, Article No. 5063145. – 13 p. DOI: 10.1155/2019/5063145.
5. Swartwout, M. *CubeSat Database [Electronic resource]* / M. Swartwout // Saint Louis University. – Available at: <https://sites.google.com/a/slu.edu/swartwout/cubesat-database> (Accessed: 11.04.2023).
6. De, R. *Enabling Science with CubeSats – Trends and Prospects [Text]* / R. De, M. P. Abegaonkar, & A. Basu // *IEEE Journal on Miniaturization for Air and Space Systems*. – 2022. – Vol. 3, no. 4. – P. 221-231. DOI: 10.1109/JMASS.2022.3209897.
7. Cappelletti, C. 2 - *CubeSat missions and applications [Text]* / C. Cappelletti, & D. Robson // *Cubesat Handbook*. – Academic Press, 2021. – P. 53-65. DOI: 10.1016/B978-0-12-817884-3.00002-3.
8. *Agile Software Development Lifecycle and Containerization Technology for CubeSat Command and Data Handling Module Implementation [Text]* / O. Liubimov, I. Turkin, V. Pavlikov, L. Volobuyeva // *Computation*. – 2023. – Vol. 11, Iss. 9. – Article no. 182. DOI: 10.3390/computation11090182.
9. *Improving energy aware nanosatellite task scheduling by a branch-cut-and-price algorithm [Text]* / L. O. Seman, C. A. Rigo, E. Camponogara, P. Munari, & E. A. Bezerra // *Computers & Operations Research*. – 2023. – Vol. 158. – Article no. 106292. DOI: 10.1016/j.cor.2023.106292.
10. Arnold, S. S. *Energy budgeting for CubeSats with an integrated FPGA [Text]* / S. S. Arnold, R. Nuzaci, & A. Gordon-Ross // *IEEE Aerospace Conference, Big Sky, MT, USA, 2012*. – P. 1-14. DOI: 10.1109/AERO.2012.6187240.
11. *Hardware-in-the-loop simulation of an on-board energy-driven scheduling algorithm for CubeSats [Text]* / V. P. Bernardo, L. O. Seman, E. A. Bezerra, & B. F. Ribeiro // *IEEE Embedded Systems Letters*. – 2024. – Vol. 16, Iss. 1. – P. 69-72. DOI: 10.1109/LES.2023.3268575.
12. *Task scheduling for optimal power management and quality-of-service assurance in CubeSats [Text]* / C. A. Rigo, L. O. Seman, E. Camponogara, E. M. Filho, & E. A. Bezerra // *Acta Astronautica*. – 2021. – Vol. 179. – P. 550-560. DOI: 10.1016/j.actaastro.2020.11.016.
13. Gerards, M. E. T. *Optimal DPM and DVFS for frame-based real-time systems [Text]* / M. E. T. Gerards, & J. Kuper // *ACM Trans. Archit. Code Optim.* – 2013. – Vol. 9, No. 4. – P. 1-23. – Article no. 41. DOI: 10.1145/2400682.2400700.
14. *STRaND-1: The world's first smartphone nanosatellite [Text]* / C. Bridges, S. Kenyon, C. Underwood, & V. Lappas // *2nd International Conference on Space Technology, Athens, Greece, 2011*. – P. 1-3. DOI: 10.1109/ICSpT.2011.6064651.
15. Gueguen, C. *Fair energy efficient scheduler providing high system capacity for wireless networks [Text]* / C. Gueguen, & C. Merlhe // *SN Appl. Sci.* – 2020. – Vol. 2. – Article no. 2116. DOI: 10.1007/s42452-020-03965-8.
16. *Energy-driven scheduling algorithm for nanosatellite energy harvesting maximization [Text]* / L. K. Slongo, S. V. Martínez, B. V. B. Eiterer, T. G. Pereira, E. A. Bezerra, & K. V. Paiva // *Acta Astronautica*. – 2018. – Vol. 147. – P. 141-151. DOI: 10.1016/j.actaastro.2018.03.052.
17. Dobiáš, P. *Online fault tolerant energy-aware algorithm for CubeSats [Text]* / P. Dobiáš, E. Casseau, & O. Sinnen // *Sustainable Computing: Informatics and Systems*. – 2023. – Vol. 38. – Article no. 100853. DOI: 10.1016/j.suscom.2023.100853.
18. Mittal, S. *A survey of techniques for improving energy efficiency in embedded computing systems [Text]* / S. Mittal // *International Journal of Computer Aided Engineering and Technology*. – 2014. – Vol. 6, Iss. 4. – P. 440-459. DOI: 10.1504/IJCAET.2014.065419.
19. *Dynamic Voltage and Frequency Scaling as a Method for Reducing Energy Consumption in Ultra-Low-Power Embedded Systems [Text]* / J. Zidar, T. Matic, I. Aleksi, & Ž. Hocenski // *Electronics*. – 2024. – Vol. 13, no. 5. – Article no. 826. DOI: 10.3390/electronics13050826.
20. *A survey on system level energy optimisation for MPSoCs in IoT and consumer electronics [Text]* / H. Ali, U. U. Tariq, J. Hardy, X. Zhai, L. Lu, Y. Zheng, F.

Bensaali, A. Amira, K. Fatema, & N. Antonopoulos // *Comput. Sci. Rev.* – 2021. – Vol. 41. – Article no. 100416. DOI: 10.1016/j.cosrev.2021.100416.

21. Oliveira, G. *Scheduling and energy savings for small scale embedded FreeRTOS-based-real-time systems [Text]* / G. Oliveira, & G. Lima // *Des Autom Embed Syst.* – 2023. – Vol. 27. – P. 3-29. DOI: 10.1007/s10617-023-09267-7.

22. *Low Power Support: Tickless Idle Mode.* Available at: <https://www.freertos.org/low-power-tickless-rtos.html>. (Accessed 28.04.2023).

23. Liu, C. L. *Scheduling Algorithms for Multiprogramming in Hard Real-Time Environment [Text]* / C. L. Liu, & J. W. Layland // *Journal of the ACM.* – 1973. – Vol. 20, Iss. 1. – P. 46-61. DOI: 10.1145/321738.321743.

24. Craig, K. *Optimal power switch design for dynamic voltage scaling from high performance to sub-threshold operation [Text]* / K. Craig, Y. Shakhsher, & B. H. Calhoun // *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design, ISLPED '12, July 2012.* – P. 221-224. DOI: 10.1145/2333660.2333714.

25. Wolf, M. Chapter 5 - Processors and Systems [Text] / M. Wolf // *The Physics of Computing.* – Morgan Kaufmann, 2017. – P. 149-203. DOI: 10.1016/B978-0-12-809381-8.00005-5.

26. Atmel | *SMART ARM-based Flash MCU DATASHEET [Electronic resource].* – Available at: <https://en.sekorm.com/doc/1455047.html>. (Accessed 29.11.2023).

## References

1. *CubeSat Design Specification (1U – 12U). Rev 14.1. The CubeSat Program, Cal Poly SLO*, 2022. 34 p. Available at: [https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/62193b7fc9e72e0053f00910/1645820809779/CDS+REV14\\_1+2022-02-09.pdf](https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/62193b7fc9e72e0053f00910/1645820809779/CDS+REV14_1+2022-02-09.pdf). (Accessed 28.04.2023).

2. Kulu, E. World's largest database of nanosatellites, over 4100 nanosats and CubeSats. *Nanosats Database.* Available at: <https://www.nanosats.eu/#figures>. (Accessed 29.11.2023).

3. Lashab, A., Yaqoob, M., Terriche, Y., Vasquez, J. C., & Guerrero, J. M. Space Microgrids: New Concepts on Electric Power Systems for Satellites. *IEEE Electrification Magazine*, 2020, vol. 8, Iss. 4, pp. 8-19. DOI: 10.1109/MELE.2020.3026436.

4. Villela, T., Costa, C. A., Brandão, A. M., Bueno, F. T., & Leonardi, R. Towards the Thousandth CubeSat: A Statistical Overview. *International Journal of Aerospace Engineering*, 2019, vol. 2019, article no. 5063145. 13 p. DOI: 10.1155/2019/5063145.

5. Swartwout, M. *CubeSat Database.* Saint Louis University. Available at: <https://sites.google.com/a/slu.edu/swartwout/cubesat-database> (Accessed: 11.04.2023).

6. De, R., Abegaonkar, M. P., & Basu, A. Enabling Science with CubeSats – Trends and Prospects. *IEEE Journal on Miniaturization for Air and Space Systems*, 2022, vol. 3, no. 4, pp. 221-231. DOI: 10.1109/JMASS.2022.3209897.

7. Cappelletti, C., & Robson, D. 2 - *CubeSat missions and applications. Cubesat Handbook.* Academic Press, 2021, pp. 53-65. DOI: 10.1016/B978-0-12-817884-3.00002-3.

8. Liubimov, O., Turkin, I., Pavlikov, V., & Vologubeva, L. Agile Software Development Lifecycle and Containerization Technology for CubeSat Command and Data Handling Module Implementation. *Computation*, 2023, vol. 11, iss. 9, article no. 182. DOI: 10.3390/computation11090182.

9. Seman, L. O., Rigo, C. A., Camponogara, E., Munari, P., & Bezerra, E. A. Improving energy aware nanosatellite task scheduling by a branch-cut-and-price algorithm. *Computers & Operations Research*, 2023, vol. 158, article no. 106292. DOI: 10.1016/j.cor.2023.106292.

10. Arnold, S. S., Nuzzaci, R., & Gordon-Ross, A. Energy budgeting for CubeSats with an integrated FPGA. *IEEE Aerospace Conference, Big Sky, MT, USA*, 2012, pp. 1-14. DOI: 10.1109/AERO.2012.6187240.

11. Bernardo, V. P., Seman, L. O., Bezerra, E. A., & Ribeiro, B. F. Hardware-in-the-loop simulation of an on-board energy-driven scheduling algorithm for CubeSats. *IEEE Embedded Systems Letters*, 2024, vol. 16, iss. 1, pp. 69-72. DOI: 10.1109/LES.2023.3268575.

12. Rigo, C. A., Seman, L. O., Camponogara, E., Filho, E. M., & Bezerra, E. A. Task scheduling for optimal power management and quality-of-service assurance in CubeSats. *Acta Astronautica*, 2021, vol. 179, pp. 550-560. DOI: 10.1016/j.actaastro.2020.11.016.

13. Gerards, M. E. T., & Kuper, J. Optimal DPM and DVFS for frame-based real-time systems. *ACM Trans. Archit. Code Optim*, 2013, vol. 9, no. 4, pp. 1-23, article no. 41. DOI: 10.1145/2400682.2400700.

14. Bridges, C., Kenyon, S., Underwood, C., & Lappas, V. STRaND-1: The world's first smartphone nanosatellite. *2nd International Conference on Space Technology*, Athens, Greece, 2011, pp. 1-3. DOI: 10.1109/ICSpT.2011.6064651.

15. Gueguen, C., & Merlthe, C. Fair energy efficient scheduler providing high system capacity for wireless networks. *SN Appl. Sci.*, 2020, vol. 2, article no. 2116. DOI: 10.1007/s42452-020-03965-8.

16. Slongo, L. K., Martínez, S. V., Eiterer, B. V. B., Pereira, T. G., Bezerra, E. A., & Paiva, K. V. Energy-driven scheduling algorithm for nanosatellite energy harvesting maximization. *Acta Astronautica*, 2018, vol. 147, pp. 141-151. DOI: 10.1016/j.actaastro.2018.03.052.
17. Dobiáš, P., Casseau, E., & Sinnen, O. Online fault tolerant energy-aware algorithm for CubeSats. *Sustainable Computing: Informatics and Systems*, 2023, vol. 38, article no. 100853. DOI: 10.1016/j.suscom.2023.100853.
18. Mittal, S. A survey of techniques for improving energy efficiency in embedded computing systems. *International Journal of Computer Aided Engineering and Technology*, 2014, vol. 6, iss. 4, pp. 440-459. DOI: 10.1504/IJCAET.2014.065419.
19. Zidar, J., Matić, T., Aleksi, I., & Hocenski, Ž. Dynamic Voltage and Frequency Scaling as a Method for Reducing Energy Consumption in Ultra-Low-Power Embedded Systems. *Electronics*, 2024, vol. 13, no. 5, article no. 826. DOI: 10.3390/electronics13050826.
20. Ali, H., Tariq, U. U., Hardy, J., Zhai, X., Lu, L., Zheng, Y., Bensaali, F., Amira, A., Fatema, K., & Antonopoulos, N. A survey on system level energy optimisation for MPSoCs in IoT and consumer electronics. *Comput. Sci. Rev.*, 2021, vol. 41, article no. 100416. DOI: 10.1016/j.cosrev.2021.100416.
21. Oliveira, G., & Lima, G. Scheduling and energy savings for small scale embedded FreeRTOS-based-real-time systems. *Des Autom Embed Syst.*, 2023, vol. 27, pp. 3-29. DOI: 10.1007/s10617-023-09267-7.
22. *Low Power Support: Tickless Idle Mode*. Available at: <https://www.freertos.org/low-power-tickless-rtos.html>. (Accessed 28.04.2023).
23. Liu, C. L. J., & Layland, W. Scheduling Algorithms for Multiprogramming in Hard Real-Time Environment. *Journal of the ACM*, 1973, vol. 20, iss. 1, pp. 46-61. DOI: 10.1145/321738.321743.
24. Craig, K., Shakhsher, Y., & Calhoun, B. H. Optimal power switch design for dynamic voltage scaling from high performance to subthreshold operation. *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design, ISLPED '12*, July 2012, pp. 221-224. DOI: 10.1145/2333660.2333714.
25. Wolf, M. *Chapter 5 - Processors and Systems. The Physics of Computing*. Morgan Kaufmann, 2017, pp. 149-203. DOI: 10.1016/B978-0-12-809381-8.00005-5.
26. *Atmel | SMART ARM-based Flash MCU DATASHEET*. Available at: <https://en.sekorm.com/doc/1455047.html>. (Accessed 29.11.2023).

Надійшла до редакції 29.04.2024, прийнята до опублікування 15.06.2024

## OPTIMIZATION OF ENERGY CONSUMPTION OF THE CUBESAT ON-BOARD COMPUTER UNDER REAL-TIME LIMITATIONS

*Ihor Turkin, Oleksandr Lubimov,  
Volodymyr Zakharenko*

**The object** of this study was to study the energy consumption of the on-board computer of the CubeSat nanosatellite. The subject of this study is the on-board computer energy consumption model and its energy efficiency optimization method. **The purpose** of this work was to develop a model of an on-board computer's energy consumption and a method of optimizing its energy efficiency, followed by an experimental verification of the effectiveness of this method. **Task:** justify the feasibility of finding new methods for optimizing the energy efficiency of the on-board computer of the CubeSat nanosatellite; prepare for the experiment, namely, develop a model of the on-board computer's energy consumption and a method of optimizing its energy efficiency, develop an experiment plan, and conduct measurement methodology; conduct experimental research and present the main results of the experiment; provide a meaningful interpretation of the obtained experimental results; generalize the conclusions, formulate the advantages and disadvantages of this work, and propose directions for further research. **Conclusions.** The power consumption of the on-board computers of nanosatellites must be reduced by hardware and software means. This study examines the effectiveness of known energy-saving methods for the authors' "Falco SBC 1.0" computing platform. These methods include dynamic frequency scaling (DFS), the race-to-dark (RTD) algorithm, and the combined algorithm, which proved to be the most effective. The results were idealized because they did not consider OS overhead. The proposed methodology can be used to evaluate other platforms. The following studies will consider the following areas: energy consumption in various processor energy-saving modes, overhead costs of multi-threaded real-time operating systems, and power management of non-processor components. Solving these problems remains an important area of scientific research.

**Keywords:** CubeSat; nanosatellite; efficiency; SBC; Falco; SSA; energy consumption model; power; ATSAMV71; DFS; RTD.

**Туркін Ігор Борисович** – д-р техн. наук, проф., зав. каф. інженерії програмного забезпечення, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

**Любимов Олександр Вікторович** – аспірант каф. інженерії програмного забезпечення, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

**Захаренко Володимир Олександрович** – канд. техн. наук, доц., доц. каф. інженерії програмного забезпечення, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

**Ihor Turkin** – Doctor of Technical Sciences, Professor, Head of the Software Engineering Department, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine,  
e-mail: i.turkin@khai.edu, ORCID: 0000-0002-3986-4186.

**Oleksandr Liubimov** – PhD Student of the Software Engineering Department, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine,  
e-mail: oleksandr.liubimov@gmail.com, o.v.liubimov@khai.edu, ORCID: 0000-0003-3636-6939.

**Volodymyr Zakharenko** – Candidate of Technical Sciences, Associate Professor at the Department of Software Engineering, National Aerospace University, "Kharkiv Aviation Institute", Kharkiv, Ukraine,  
e-mail: v.zakharenko@khai.edu, ORCID: 0009-0003-1713-4558.