

УДК 004.415.53

ДОСЛІДЖЕННЯ ОБЛАСТІ ДАНИХ BIOS ТА РОЗРОБКА ПРОГРАМИ ДЛЯ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ ПРО АПАРАТНУ КОНФІГУРАЦІЮ

Турута Г.С.

e-mail: hlib.turuta@nure.ua

Науковий керівник – д.ф. Хаханов І.В.

Харківський національний університет радіоелектроніки, каф. АПОТ
м. Харків, Україна

The major goal of this thesis is to develop a program that displays data available for viewing and tracking in a computer's BIOS. In order to complete the qualification work, a study was conducted on the structure and principles of data storage displayed in the BIOS. Additionally, a program was developed to present accessible information about the computer's hardware configuration. The program was implemented using the C# programming language and the cross-platform Avalonia UI framework, providing a user-friendly and intuitive interface. The result of the work is an efficient solution for visualizing critical technical parameters of the system, which can be beneficial for both regular users and IT professionals.

У сучасних інформаційних технологіях важливу роль відіграє доступ до інформації про апаратну частину комп'ютера. Одним із основних джерел таких даних є BIOS (Basic Input/Output System) — низькорівнева прошивка, яка відповідає за ініціалізацію обладнання при запуску системи та надає базові функції для взаємодії між апаратним та програмним забезпеченням. Незважаючи на ключову роль BIOS, її інтерфейс часто є обмеженим у плані зручності користування та доступу до інформації. Дослідження в галузі програмного забезпечення дозволяють створювати інструменти, що забезпечують зручний доступ до цих даних без необхідності входу до BIOS.

Метою цієї роботи є розробка програми яка виводить данні які можливо подивитись та відслідковувати у BIOS комп'ютера. Програма реалізована мовою програмування C# із застосуванням кросплатформного фреймворку Avalonia UI, що забезпечує зручний та інтуїтивно зрозумілий інтерфейс користувача.

При запуску програми виконується ініціалізація, під час якої збирається вся статична інформація про комп'ютер за допомогою WMI (Windows Management Instrumentation). Для цього використовується бібліотека System.Management, яка надає доступ до системної інформації Windows. Змінні, що постійно оновлюються (температура, завантаження, швидкість вентиляторів), зчитуються з використанням бібліотеки LibreHardwareMonitor. Наприклад, для визначення температури графічного процесора використовуються сенсори типу Temperature, а для частот ядра і пам'яті – сенсори типу Clock. Архітектура програми дозволяє чітко

розділити логіку та інтерфейс, дотримуючись патерну MVVM (Model-View-ViewModel). Клас MainWindowViewModel виконує роль моделі представлення і взаємодіє із внутрішніми класами (CPU, GPU, RAM, MotherBoard, Storage), щоб отримати інформацію про відповідні компоненти системи.

Для оновлення динамічної інформації створюється окремий потік, який кожен секунду оновлює значення параметрів. Оновлення відбувається кожен секунду через виклик Task.Delay(1000). Для відфільтровування частих оновлень вхідного потоку значень використовується механізм debounce за допомогою бібліотеки Reactive Extensions (Rx).

Інформація про оперативну пам'ять (RAM) збирається через запит WMI до класу Win32_PhysicalMemory, який дозволяє отримати виробника, слот, швидкість та обсяг кожного модуля. Для розрахунку загального обсягу оперативної пам'яті комп'ютера використовується підсумовування обсягів усіх знайдених модулів:

$$V_{total} = \sum_{i=1}^n V_i,$$

де V_{total} – загальний обсяг оперативної пам'яті (Total Volume), V_i – обсяг i -го модуля пам'яті, n – загальна кількість встановлених модулів.

Таблиця 1 – Основні параметри моніторингу графічного адаптера (GPU)

Властивість	Тип даних	Опис властивості
Name	string	Назва графічного адаптера, отримана через WMI.
Temperature	string	Температура GPU у форматі "°C".
CoreClockSpeed	float	Тактова частота ядра графічного процесора у MHz.
CoreLoad	float	Завантаженість графічного процесора у відсотках.
Fans	List<string>	Список швидкостей обертання вентиляторів у форматі "RPM".

Для відображення параметрів, які змінюються з часом, розроблено динамічні графіки. Для відображення графіків використовується масив з 20 елементів для кожного параметра. При кожному оновленні нове значення

додається в кінець масиву, а найстаріше видаляється, завдяки чому відображається інформація за останні 20 секунд роботи.

У результаті виконання роботи було досліджено структуру та принципи зберігання даних які відображаються у BIOS, а також розроблено програму, яка дозволяє виводити інформацію про апаратну конфігурацію комп'ютера. Реалізовано функції моніторингу апаратного забезпечення, включаючи CPU, GPU, RAM, материнську плату та накопичувачі. Забезпечено динамічне оновлення даних у реальному часі з використанням бібліотеки LibreHardwareMonitor та WMI. Виконана робота підтверджує доцільність застосування сучасних технологій для розв'язання завдань моніторингу апаратної частини комп'ютера.

Список використаних джерел:

1. Avalonia Docs. Avalonia UI : вебсайт. URL: <https://docs.avaloniaui.net/docs/welcome>. (дата звернення: 10.03.2026).
2. LibreHardwareMonitor. GitHub : вебсайт. URL: <https://github.com/LibreHardwareMonitor/LibreHardwareMonitor>. (дата звернення: 10.03.2026).
3. System.Management WMI C#. Microsoft Learn : вебсайт. URL: <https://learn.microsoft.com/en-us/dotnet/api/system.management?view=net-8.0-pp>. (дата звернення: 10.03.2026)
4. MoRoC (Monitoring of Resources of Computer). GitHub : вебсайт. URL: <https://github.com/Arnaghad/MoRoC>. (дата звернення: 10.03.2026)