

# E3UI - An Easy-To-Use Windows Energy Consumption Analysis Tool

Jan-Hendrik Schneider (ID: 6121934)  
Daniel Chou Rainho (ID: 5336643)  
Filip Gunnarsson (ID: 5319412)

March 28, 2024

## Abstract

We present E3UI - a desktop application for Windows devices that allows users to monitor and analyze their energy consumption. Designed for battery-powered devices, E3UI uses the Windows Energy Estimation Engine (E3) [5] to provide detailed insights into the power usage of individual applications. By enabling users to track the energy impact of specific applications and make informed choices, E3UI plays a pivotal role in promoting sustainable digital habits.

## 1 Problem Statements

1. The complexity of existing energy monitoring tools, which often feature cluttered interfaces or rely solely on command-line interfaces (CLI), makes them intimidating and inaccessible for many users.
2. The lack of detailed carbon intensity and consumption data tied to specific applications leaves users without a clear understanding of the environmental impact of their digital activities.
3. Most energy monitoring solutions fail to provide application-specific power consumption data, hindering users' ability to identify and manage the energy use of individual applications effectively.

## 2 Introduction

Several energy consumption monitoring tools exist, each with its unique strengths and limitations:

- **Intel Power Gadget**<sup>1</sup> was a popular tool for monitoring CPU power consumption, providing detailed metrics specifically for the CPU. However, it's being discontinued, leaving a gap for users needing comprehensive energy monitoring solutions.
- **EnergiBridge**[8] offers a solution for analyzing overall system energy consumption but falls short in user accessibility. Its command-line interface (CLI) presents a steep learning curve, and the lack of a graphical user interface (GUI) makes it difficult for non-technical users to adopt.
- **Windows Task Manager** provides a basic level of energy consumption monitoring by indicating the relative energy impact of applications. However, this feature offers only a vague understanding of an application's energy use, without presenting actual consumption figures, limiting its utility for precise energy management.

E3UI [3] is a specialized software created for the Windows operating system, offering a detailed analysis of energy consumption by applications on battery-equipped devices.

Unlike traditional energy profilers, E3UI distinguishes itself by measuring the power impact of applications independently, allowing for a more targeted approach to energy savings.

Adding to the innovative applications of E3UI, another use case highlights its ease of use, making it accessible to a broader audience beyond those with technical expertise.

Unlike many other energy consumption analysis tools that require command line interface knowledge, E3UI offers a simple to use interface. This ensures that users can effortlessly monitor and analyze their energy consumption without any need for coding experience or technical background.

---

<sup>1</sup><https://www.intel.com/content/www/us/en/developer/articles/tool/power-gadget.html>

## 3 Methodology

### 3.1 Nudging

A key feature of our product is designed to gently encourage users to become more conscious of their energy consumption. We are confident that our application empowers users to make environmentally conscious choices, potentially reducing their overall carbon footprint.

Studies, such as the one noted in [6], suggest that the use of nudging techniques can effectively inspire shifts toward more sustainable habits. This is the rationale behind our development of a background application that offers users clear insights into their system’s energy use.

### 3.2 Enhancing Accessibility and Promoting Sustainability

E3UI simplifies energy data management through an intuitive interface, making it accessible to all types of users, from tech novices to the environmentally conscious. This facilitates energy awareness by removing the intimidation factor of technical tools.

The tool offers immediate feedback on energy consumption patterns, allowing for quick and informed decisions for optimizing application use and enhancing energy efficiency.

### 3.3 Calculating CO2 emissions

The energy estimation engine only provides energy consumption data in *mJ*. While reducing energy consumption remains an important goal, it is even more important to recognize that the carbon dioxide (CO2) emissions resulting from that energy consumption are of even greater environmental concern.

While energy consumption is an important metric, it only tells part of the story. The CO2 emissions associated with that consumption provide a more complete picture of environmental impact. To get an estimate of carbon emissions, the amount of kilowatt-hours (*kWh*) of electricity is often multiplied by a carbon intensity value in grams of carbon per kilowatt-hour produced. However, relying on a single, constant value to represent CO2 emissions per *kWh* is inaccurate. This is because the CO2 intensity of electricity generation varies significantly depending on the energy sources used in the local electricity grid. For example, electricity from coal-fired power plants has a high CO2 intensity, while electricity from renewable sources such as solar and wind have much lower CO2 emissions

[7]. Table 1 shows the carbon intensity values for common forms of electricity generation.

The variability of CO2 intensity is amplified by the growing share of renewable energy in the electricity mix. As the contribution of solar and wind energy increases, the CO2 intensity of the electricity grid becomes even more dynamic. Factors such as weather conditions, time of day, and other environmental variables mean that CO2 intensity can vary widely. To capture this variability and provide a more accurate estimate of CO2 emissions from electricity use, it is essential to consider the current state of the local electricity grid.

To address this need, there is already an API that provides the current carbon intensity of electricity generation around the world API [1]. However, this resource is not freely accessible. In our approach, we use the Energy Charts API from the Fraunhofer ISE Institute [2]. While this API does not directly offer CO2 intensity values, it provides detailed information on the current share of different electricity production types. By analyzing this data, we can calculate the CO2 intensity of electricity generation at any given moment. The current carbon intensity of the local grid  $CI_{grid}$  can be calculated based on the power generation  $P_i$  per generation type  $i$  and the carbon intensity per generation type  $CI_i$  as follows:

$$CI_{grid} = \frac{\sum_i P_i CI_i}{\sum_i P_i} \quad (1)$$

Generation Type	CO2 Emmissions per <i>kWh</i>
Coal	1001
Natural gas	486
Oil	840
Nuclear	13
Biomass	52
Photovoltaic	43
Geothermal	37
Hydropower	21
Wind	13

Table 1: Carbon intensity of electricity generation types [7]

Using a location API call [4], the tool is able to automatically detect the user’s location and then select the appropriate data from the Energy Charts API.

By incorporating real-time CO2 intensity data into our E3UI, users can gain a deeper understanding of the environmental impact of their computing activities. Not only can users assess the carbon emissions resulting from their use of applications, but they can also gain insight into the overall carbon emissions

of the power grid. This knowledge empowers users to make informed decisions about the timing of energy-intensive computational tasks, potentially opting for periods when the grid's CO2 intensity is lower.

## 4 Results

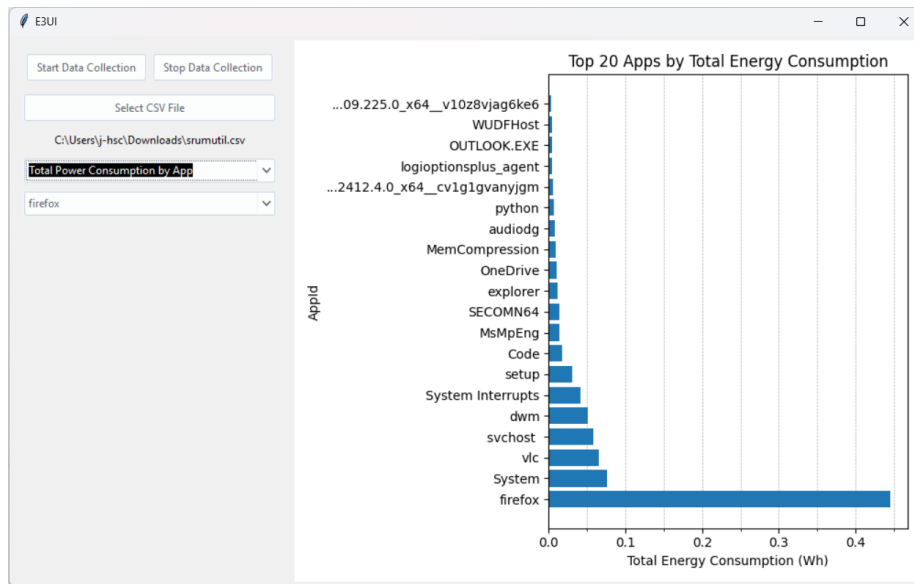


Figure 1: Total Power Consumption by App

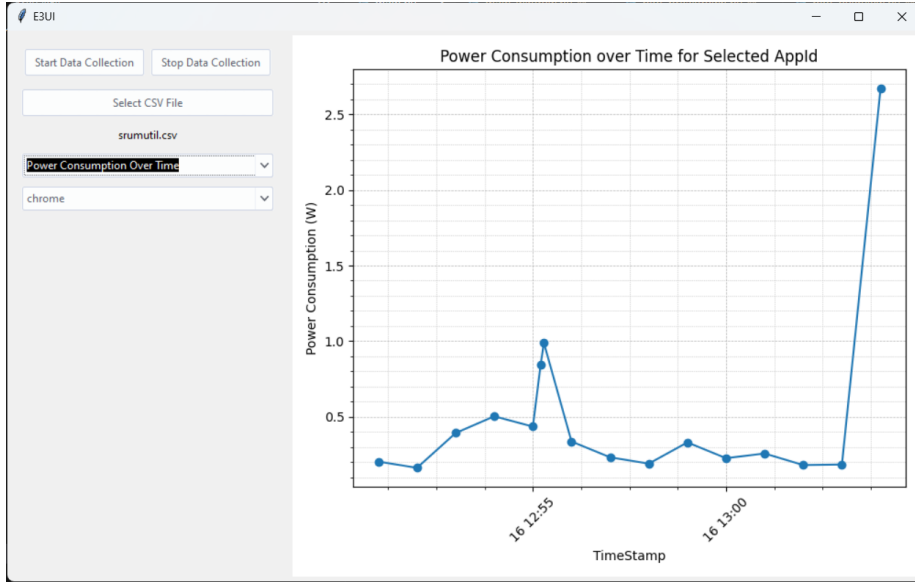


Figure 2: Total Power Consumption Over Time

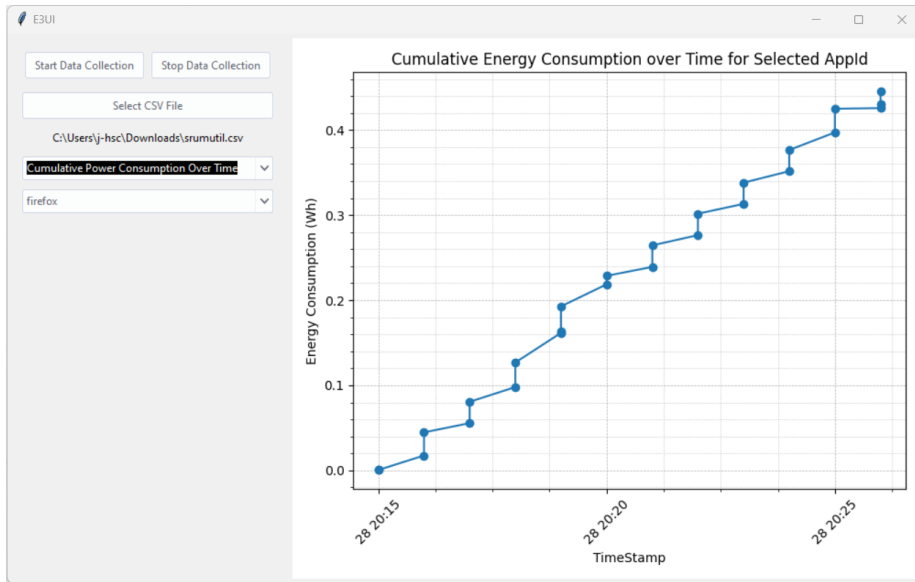


Figure 3: Cumulative Energy Consumption Over Time

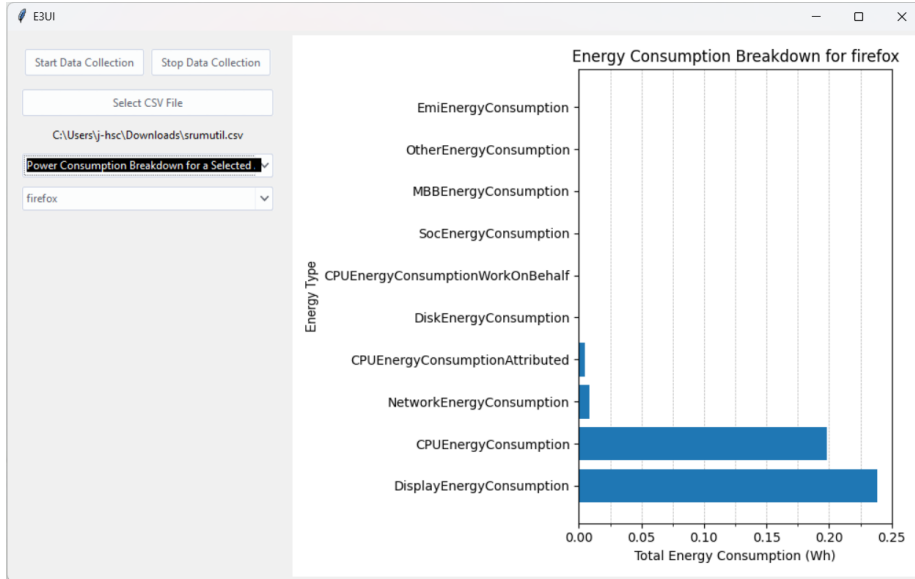


Figure 4: Energy Consumption Breakdown for System

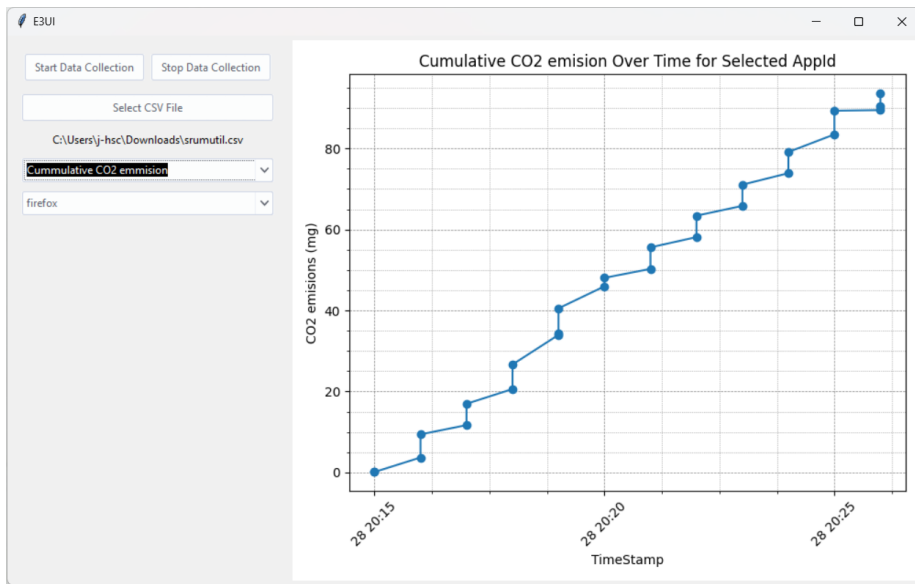


Figure 5: Cumulative Carbon Emissions

## 4.1 Graphs

The core functionality of our application allows users to initiate and terminate the monitoring process. The analysis results are presented through a series of graphs, each aimed at providing insights into different aspects of energy consumption. Below, we describe the findings from each type of graphical representation generated by our application.

- **Top 20 Energy-Consuming Applications:** A bar chart displays the applications that are the most energy-intensive, offering users a clear perspective on which applications contribute the most to their system's overall energy consumption.
- **Energy Consumption Sources Breakdown:** For any specified application, a bar chart breaks down the various sources of energy consumption (e.g., CPU, Display, Disk). This breakdown allows users to understand how an application allocates resources, enabling targeted optimization strategies for developers.
- **Temporal Energy Consumption Patterns:** Three line charts offer temporal views of consumption for a specified application, including power consumption over time, cumulative energy consumption over time, and carbon emissions over time. These charts provide a dynamic understanding of how application usage impacts energy consumption and environmental footprint across different timescales.

## 4.2 Conversion to Watts per Hour for Familiarity and Standardization

E3 measures power consumption in millijoules ( $mJ$ ) but converts this data into watts per hour ( $Wh$ ) for presentation. This decision is based on the widespread familiarity with the  $Wh$  unit, largely because it is the standard measurement used in utility bills. By aligning with this common standard, E3 ensures that its users can easily understand and relate to the energy consumption data presented.



## 5 Discussion

### 5.1 Dissemination

We prioritize accessibility and community engagement by making our solution open source. This approach not only facilitates transparency but also encourages contributions from the global community, enhancing the tool’s effectiveness and adaptability.

To ease the deployment and use of our software, we use *PyInstaller* to compile the project into a standalone executable file. This simplification means users need only to run the executable with administrative privileges, eliminating complex installation processes.

### 5.2 Limitations

Initially, we aspired to implement real-time energy consumption monitoring. However, the requirement for a secondary thread to run concurrently and execute commands at regular intervals introduced significant power usage, paradoxically counteracting our sustainability goals.

The integration with E3, our chosen tool for energy consumption analysis, further complicates real-time monitoring. E3 does not support incremental updates; it requires stopping and restarting the measurement process to capture new data, making continuous monitoring inefficient and challenging to architect within our project’s timeframe.

Furthermore, our solution is currently tailored to Windows devices equipped with batteries, a limitation inherited from E3’s operational constraints. Consequently, energy consumption on desktop computers, which are commonly used but do not feature batteries, falls outside our project’s scope. This limitation significantly narrows the range of devices on which our solution can be applied, potentially limiting its utility in broader sustainability efforts across different types of computing environments.

### 5.3 Future Work

A primary focus will be on overcoming the challenge of real-time energy consumption monitoring. Investigating more efficient multi-threading techniques could reduce the additional power consumption associated with real-time data collection. Additionally, exploring alternatives to E3 or enhancing its capabilities to support incremental updates without the need for process restarts would significantly improve usability and functionality.

Expanding the solution’s compatibility beyond Windows devices with batteries to include desktop computers and other operating systems represents another critical avenue for future work. This expansion could involve developing or integrating with new tools capable of measuring energy consumption across a broader array of hardware configurations. Such advancements would not only extend the applicability of our solution but also amplify its impact on promoting sustainability in software engineering.

By addressing these limitations, future iterations of the project can offer a more versatile and efficient tool, moving closer to our vision of widespread adoption and contribution to sustainable software practices.

## References

- [1] Electricity Maps API Documentation — [static.electricitymaps.com](https://static.electricitymaps.com/api/docs/index.html). <https://static.electricitymaps.com/api/docs/index.html>. [Accessed 28-03-2024].
- [2] Energy-Charts API — [api.energy-charts.info](https://api.energy-charts.info). <https://api.energy-charts.info/>. [Accessed 28-03-2024].
- [3] GitHub - daniel-chou-rainho/e3ui: Windows Energy Consumption Measurement — [github.com](https://github.com/daniel-chou-rainho/e3ui). <https://github.com/daniel-chou-rainho/e3ui>.
- [4] [ipinfo.io](https://ipinfo.io). <https://ipinfo.io/json>. [Accessed 28-03-2024].
- [5] Scott Chamberlin. Measuring your application power and carbon impact (part 1) - sustainable software, 2020.
- [6] Stephanie Mertens, Mario Herberz, Ulf J. J. Hahnel, and Tobias Brosch. The effectiveness of nudging: A meta-analysis of choice architecture interventions across behavioral domains. *Proceedings of the National Academy of Sciences*, 119(1):e2107346118, 2022.
- [7] Scott Nicholson and Garvin Heath. Life cycle greenhouse gas emissions from electricity generation: Update life cycle assessment of energy systems. 2012.

- [8] June Sallou, Luís Cruz, and Thomas Durieux. Energibridge: Empowering software sustainability through cross-platform energy measurement, 2023.