# Seminar Paper: Automatic Energy-Hotspot Detection and Elimination in Real-Time Deeply Embedded Systems

Andreas Gräfensteiner

## 1 ABSTRACT

The paper[2] delves into the crucial realm of energy efficiency within contemporary embedded systems. As pivotal components of modern technology, embedded systems permeate various aspects of daily life, from smartphones to healthcare infrastructure. The digitalization era underscores the urgency of prioritizing their energy efficiency, especially considering the prevalent reliance on battery power. This paper introduces an accessible and effective approach to tackle energy hotspots, areas within embedded systems that can be optimized to curtail energy consumption.

The exploration of energy hotspots involves categorizing them into three distinct types: Hotspot Tail, Hotspot Sleep, and Hotspot Active. Each type represents specific inefficiencies within the system, ranging from delays between code statements to suboptimal transitions between active and sleep states. The paper proposes a systematic algorithm to detect and mitigate these hotspots, offering tailored solutions for each category.

The significance of this paper lies in its practical applications, addressing challenges in real-world scenarios across various industries. By identifying and resolving energy hotspots in real time, the paper aims to enhance system performance, extend battery life, improve reliability, reduce costs, and minimize environmental impact. However, the feasibility of implementation depends on technological advancements and adaptability to diverse environments.

In conclusion, the paper provides valuable insights into the intricate landscape of energy management in deeply embedded systems. Its contributions lay the groundwork for future research and practical implementations, aiming to propel advancements in sustainable and energy-efficient embedded systems.

## 2 INTRODUCTION

Embedded systems hold a significant position in contemporary technology, serving as a fundamental cornerstone that deeply influences various facets of our daily lives. The prevalence of digitalization has led to their widespread utilization across the globe, infiltrating our smartphones, gaming consoles, wearables, household appliances, healthcare sector, and even the infrastructure of our power supply grids. This widespread integration emphasizes the critical need to prioritize their energy efficiency, particularly considering that most of these devices rely on battery power. This holds considerable significance in the present context, given that a substantial number of batteries currently operate on a single-use system, retaining a considerable amount of pollutants within their composition. Consequently, it has become increasingly crucial to minimize energy costs wherever feasible. In light of this necessity, the paper titled "Automatic Energy-Hotspot Detection and Elimination in Real-Time Deeply Embedded Systems"[2] offers an accessible and effective approach. This paper introduces methods for identifying and resolving what are termed as *energy hotspots*, areas within these systems that can be optimized to curtail their energy consumption.

## 3 ENERGY HOTSPOTS IN EMBEDDED SYSTEMS

Numerous scholarly articles endeavour to mitigate the energy costs associated with embedded systems through adjustments to the CPU's voltage, clock frequency, or enhancements to the underlying hardware infrastructure. Unfortunately, it is not always feasible to modify the hardware or firmware directly. Thus, Shekarisaz et al. have conceived an approach to analyse the program code itself for the purpose of further optimization. Through their research, they identified specific sections of code that exhibit notably higher energy consumption than others, giving rise to the concept of so-called *Energy Hotspots*. Optimization of these particular segments typically results in a discernible reduction in the overall energy expenditure of the system. Their paper "Automatic Energy-Hotspot Detection and Elimination in Real-Time Deeply Embedded Systems"[2] addresses this issue of energy consumption in deeply embedded systems. It proposes a method for identifying and mitigating energy hotspots in real time, aiming to improve energy efficiency in these systems. The approach involves a combination of hardware monitoring and software techniques to detect areas causing excessive energy usage. Once identified, the system applies optimization strategies to alleviate these hotspots, thereby enhancing overall energy efficiency without compromising performance. The paper focuses on real-time detection and mitigation of energy hotspots in deeply embedded systems, offering a practical solution to improve their energy efficiency.

### 3.1 Types of Energy Hotspots

Within the scope delineated by Shekarisaz et al., the term *energy hotspot* encapsulates designated areas or individual components intricately nestled within the intricate framework of deeply embedded systems, showcasing a conspicuous tendency towards markedly elevated energy consumption levels. These hotspots, positioned within the system's architecture, denote specific regions, operational functions, or discrete modules that conspicuously ingest and expend a notably substantial quantum of energy resources when juxtaposed against the prevailing energy consumption patterns observed throughout the system's operational matrix.

The paramount significance of identifying these enigmatic hotspots resides in their potential to exert an unequivocal influence upon the overarching energy efficiency metrics governing the embedded system's functionality. These hotspot entities yield an inherent capacity to instigate disruptive inefficiencies, precipitate heightened energy consumption thresholds, and decisively truncate the operational longevity of portable devices reliant on finite battery reserves.

An insightful facet discernible within the paper is the delineation of three distinct categories characterizing these hotspots. This categorical classification serves as an illuminating framework, permitting a granular understanding and systematic demarcation

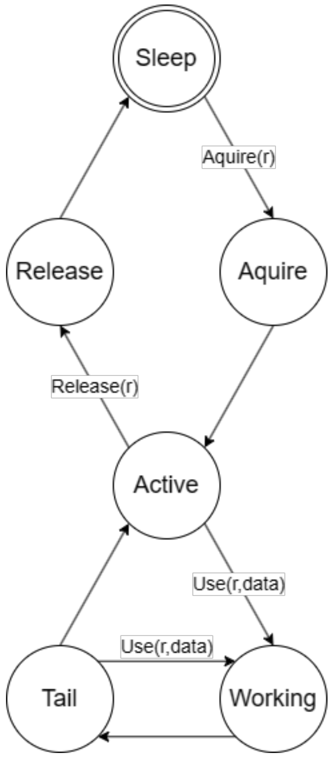


**Figure 1: State machine of r (simplified version from Fig 1 in the paper[2]) created with draw.io**

of these energy-intensive zones within the intricate labyrinth of deeply embedded systems. The classification empowers researchers and practitioners alike with a comprehensive taxonomy, paving the way for sophisticated analyses, targeted intervention strategies, and bespoke optimization methodologies tailored to address the divergent nuances encapsulated within each category of these influential energy hotspots.

Subsequently, we shall delve into the precise definitions of these influential energy hotspots.

*3.1.1 Hotspot$_{Tail}$:* This Hotspot specifically focuses on the energy inefficiency caused by a delay or time gap between two consecutive `Use(r, data)` statements within the program. After a significant time delay between these statements, there occurs a subsequent energy consumption referred to as $E_{\text{Tail}}$(Intvli). This energy inefficiency contributes to the "tail" phase of energy consumption, which occurs after the execution of the *i*-th `Use(r, data) statement`.

*3.1.2 Hotspot$_{Sleep}$:* This Hotspot in an embedded system represents an energy hotspot caused by the inefficiency in transitioning a module or resource between Active and Sleep states due to significant energy overheads associated with Release(r) and Acquire(r) transitions, especially when the duration of computational code between these transitions is shorter than the Lower Bound on Sleep Time (LBST), which is calculated:

$$LBST = \frac{E_{\text{Rel}} + E_{\text{Acq}}}{Pow_{\text{A}} - Pow_{\text{S}}}$$

*3.1.3 Hotspot$_{Active}$:* The identified Hotspot within embedded systems denotes a significant area where energy inefficiencies arise due to noticeable intervals between resource acquisition (Acquire) and subsequent utilization (Use), or alternatively, during resource utilization (Use) and subsequent release (Release). These discernible time gaps precipitate a wasteful scenario wherein resources remain needlessly active or idle for extended durations, surpassing their necessary operational timeframes. This prolonged idleness or prolonged activity leads to an unwarranted dissipation of energy reserves, impacting the system's overall efficiency.

## 3.2 Prevention of Energy Hotspots

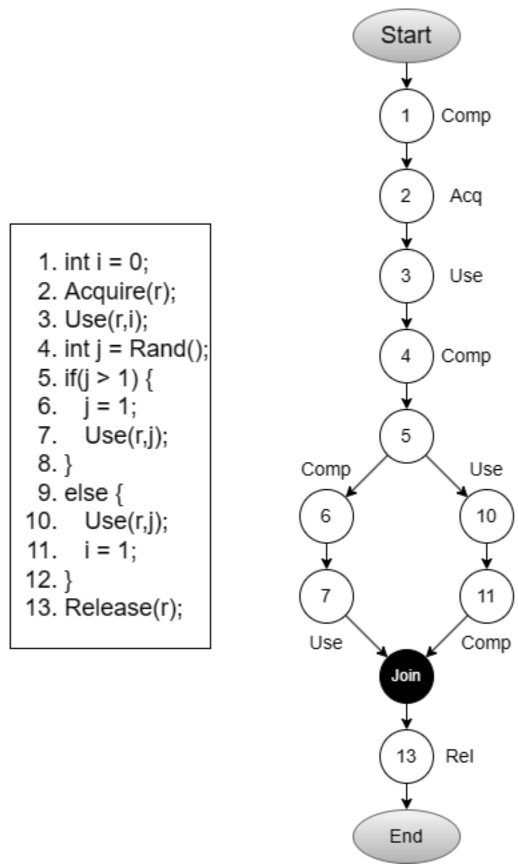


**Figure 2: A code example and the associated annotated MCFG from Fig 2 of the paper[2] created with draw.io**

Through a transformative process that involves the conversion of a specified program code into an altered representation known as a modified control flow graph (MCFG, illustrated in Figure 2), the authors introduce an intricately designed algorithm poised to identify and subsequently mitigate a substantial proportion of the prevalent hotspots. This innovative algorithm, encapsulated within the essence of simplicity, functions by meticulously traversing each node embedded within the comprehensive structure of the MCFG.

The algorithm's fundamental operation rests upon a nuanced classification system designed to segregate and categorize these nodes into three distinct predefined categories. This intricate classification schema forms the bedrock of the algorithm's functionality, enabling a systematic approach towards tackling the diverse manifestations of hotspots present within the MCFG. Post-classification, the algorithm proceeds to systematically apply designated solutions, tailoring its interventions in accordance with the nuanced characteristics encapsulated within each identified category.

This systematic methodology underpinning the algorithm's operation serves as a testament to its adaptive nature, designed to navigate through the multifaceted landscape of hotspots with dexterity and precision. By delineating tailored solutions and implementing targeted measures specific to the discerned categories, the algorithm exhibits an approach aimed at addressing the diverse array of energy-intensive areas within the MCFG. This deliberate and systematic algorithmic approach offers a promising trajectory towards mitigating the overarching impact of hotspots, envisaging a landscape where energy inefficiencies are methodically identified, categorized, and subsequently ameliorated with precision and efficacy.

Let us now turn our attention to the proposed solutions tailored for the distinct types of identified energy hotspots.

*3.2.1 $Hotspot_{Tail}$:* Resolving the $\text{Hotspot}_{\text{Tail}}$ is rather simple. The paper suggests to "move the ith Use(r, data) statement forward just beside the (i + 1)th Use(r, data) statement"[2], which would result in a complete elimination of the Hotspot. But due to time restrictions and possible dependencies between the Use blocks it is not always possible to resolve this Hotspot.

*3.2.2 $Hotspot_{Sleep}$:* The resolution strategies aim to eliminate or minimize the energy inefficiencies associated with $\text{Hotspot}_{\text{Sleep}}$ by optimizing the transition between Active and Sleep states. By identifying the optimal Sleep durations, adjusting code execution sequences, and potentially leveraging hardware optimizations, the paper proposes methods to mitigate the energy wastage caused by inefficient transitions between these states.

*3.2.3 $Hotspot_{Active}$:* The paper proposes a code fragment reordering strategy aiming to bring acquire nodes closer to their corresponding use nodes or to align use nodes with release nodes, reducing inefficiencies in resource utilization. Careful implementation is crucial to avoid creating a new $\text{Hotspot}_{\text{Sleep}}$. The proposed algorithm reexamines the relocated nodes to prevent the unintended creation of $\text{Hotspot}_{\text{Sleep}}$.

## 4 ASSESSMENT

Shekarisaz et al. intricately navigate the critical terrain of energy efficiency within deeply embedded systems, illuminating its core focus on the real-time detection and elimination of energy hotspots. This concerted effort is aimed at orchestrating a marked enhancement in the overall operational efficiency of these intricate systems. Delving comprehensively into multifaceted aspects, the paper meticulously dissects methods geared towards pinpointing areas of excessive energy consumption, categorizing them into distinct types of hotspots, notably $\text{Hotspot}_{\text{Tail}}$, $\text{Hotspot}_{\text{Sleep}}$, and $\text{Hotspot}_{\text{Active}}$.

Central to its discourse is a resounding emphasis on the imperative nature of real-time monitoring and ongoing analysis. The paper[2] underscores the significance of these practices in discerning hotspots as they manifest within the operational dynamics of the system. It navigates through strategic discussions surrounding automated hotspot detection strategies, leveraging algorithmic analyses to unveil and address these energy-intensive areas proactively.

Moreover, it does not merely identify issues but offers pragmatic solutions to combat and mitigate these hotspots, all while safeguarding the system's optimal performance. These solutions span an array of methodologies, encompassing code optimizations, strategic rearrangements, and dynamic power management strategies meticulously crafted to streamline and harmonize energy efficiency without compromising the system's operational efficacy.

A pivotal hallmark of the paper's[2] contribution rests in furnishing an automated framework adept at identifying and resolving energy hotspots within deeply embedded systems. It extends its gaze to practical applications, deftly addressing challenges such as the nuances of implementation hurdles, the delicate balance between energy efficiency and system performance, and the intricate handling of code dependencies. This comprehensive approach crystallizes its endeavours towards redefining the landscape of energy management within deeply embedded systems, offering a systematic roadmap for sustainable operational efficiency.

### 4.1 Usage

The relevance of this paper[2] lies in its focus on identifying and eliminating energy hotspots in real-time within deeply embedded systems. Energy hotspots refer to areas or components within a system that consumes excessive power, potentially leading to reduced battery life, overheating, or performance degradation. Detecting and mitigating these hotspots is pivotal to enhancing the overall efficiency and reliability of embedded systems.

- *Enhanced Performance:* Identifying and resolving energy hotspots can lead to better performance. By optimizing power consumption, systems can operate more efficiently, leading to improved responsiveness and overall functionality.

- *Extended Battery Life:* For devices relying on batteries, managing energy hotspots can significantly extend their operational lifespan. This is particularly important in applications like IoT devices, wearables, or remote sensors, where replacing or recharging batteries is impractical or inconvenient.

- *Improved Reliability:* Energy hotspots can cause system instability or failures. Identifying and addressing these hotspots can enhance the reliability and robustness of deeply embedded systems, ensuring consistent and uninterrupted functionality.

- *Cost Efficiency:* Minimizing excessive power consumption not only improves the device's operational efficiency but also reduces the overall cost of ownership by decreasing energy expenses and potentially delaying hardware upgrades or replacements.

- *Environmental Impact:* Energy-efficient embedded systems contribute to reducing overall energy consumption, which aligns with sustainability goals by minimizing environmental impact and reducing carbon footprints.

In summary, the paper[2] holds significance for enhancing the efficiency, reliability, and durability of deeply embedded systems by promptly detecting and addressing energy hotspots. Its relevance spans across diverse industries and applications reliant on these systems, offering pragmatic solutions for real-time hotspot identification and resolution. This approach fosters transformative improvements, promising heightened efficiency and longevity in deeply embedded systems, impacting various sectors and technological applications.

## 4.2 Feasibility

The feasibility of the paper's proposal depends on various factors related to implementation, technological advancements, and practical application considerations. These are the two most important aspects to consider:

- *Technological Advancements:* The feasibility of implementing automatic energy-hotspot detection and elimination relies heavily on the available technological advancements. If the proposed methods rely on cutting-edge hardware or software that is not widely available or too costly, practical implementation could be challenging. Given the recent nature of the paper, industries may require additional time to effectively incorporate its principles into their design practices. However, owing to its reliance primarily on the executable code within the system, the implementation process is anticipated to be relatively straightforward, facilitating seamless integration into pre-existing systems. The pivotal consideration, however, rests upon determining whether the proposed adjustment yields a sufficient energy-saving outcome to warrant the efforts and resources entailed in its implementation.

- *Adaptability and Flexibility:* Embedded systems often operate in diverse environments and conditions. The feasibility of the proposed solution depends on its adaptability to different scenarios, hardware configurations, and usage patterns without compromising detection accuracy or system performance. Similar to the aforementioned point, the crucial question here will be whether the effort involved justifies the energy savings. It is noteworthy that while the integration of the code is straightforward in the case of new productions, many pre-existing systems may not be designed for updates and, consequently, would not benefit from these alterations. The definitive market acceptance remains uncertain; however, the likelihood of success appears promising.

In conclusion, while the concept of automatic energy-hotspot detection and elimination in real-time deeply embedded systems holds significant promise for enhancing energy efficiency, its feasibility relies on overcoming technological and adaptability challenges. Advances in technology and innovative solutions addressing these challenges could increase the practicality and feasibility of implementing such systems in various real-world applications.

# 5 CONCLUSION

The paper "Automatic Energy-Hotspot Detection and Elimination in Real-Time Deeply Embedded Systems" represents a significant contribution to the field of embedded systems by addressing the critical issue of energy efficiency. Through its exploration of methods for detecting and mitigating energy hotspots in real-time, the paper offers valuable insights and strategies aimed at improving the overall energy efficiency of deeply embedded systems. The findings presented in the paper underscore the importance of real-time monitoring and analysis in identifying energy hotspots within embedded systems. By categorizing different types of hotspots such as Hotspot Tail, Hotspot Sleep, and Hotspot Active, the paper provides a comprehensive understanding of these inefficiencies. Furthermore, the paper proposes practical solutions to mitigate these hotspots without compromising system performance. These solutions involve code optimizations, dynamic power management, and strategies for minimizing idle periods or inefficient resource transitions. A key strength of this paper lies in its emphasis on the practical applications of these strategies across various industries. From IoT devices to automotive systems and industrial automation, the proposed methodologies offer promising avenues for improving energy efficiency in diverse embedded system applications. However, while the paper offers valuable insights and potential solutions, there remain practical challenges in implementing these strategies. Issues such as code dependencies, hardware limitations, and the complexity of real-world integration must be addressed for the proposed solutions to be effectively deployed. In conclusion, the paper serves as a foundational resource for understanding, detecting, and addressing energy hotspots in real-time deeply embedded systems. Its contributions lay the groundwork for future research and practical implementations aimed at achieving sustainable and energy-efficient embedded systems, thus contributing significantly to advancements in the field.

# REFERENCES

[1] JGraph Ltd. Draw.io a free online diagram software. URL https://app.diagrams.net.
[2] Mohsen Shekarisaz, Lothar Thiele, and Mehdi Kargahi. Automatic energy-hotspot detection and elimination in real-time deeply embedded systems.