Designing a General Purpose Development Platform for Energy-harvesting Applications

Nurani Saoda
University of Virginia
saoda@virginia.edu

Md Fazlay Rabbi Masum Billah
University of Virginia
masum@virginia.edu

Bradford Campbell
University of Virginia
bradjc@virginia.edu

ABSTRACT

Battery-less energy-harvesting systems have widened the landscape of Internet-of-Things (IoT) applications by taking computation to hard-to-reach places. Energy-harvesting sensors are perpetual, environment-friendly, cost-effective, and maintenance-free. Despite having such lucrative characteristics, battery-powered devices hold majority share of today's IoT market, since developing energy-harvesting applications require more expert knowledge, careful implementation, and rigorous debugging than applications with stable power. In this paper, we argue that development becomes easier, faster, efficient, and scalable with a standard, re-usable, general purpose platform that ensures the platform's versatility across various application with proper balance between abstraction and accessibility in hardware and software. Such platforms would provide flexibility across both hardware and software layers, at the same time, producing reliable performance. However, realizing this design point pose several research challenges that need to be identified and addressed. We identify the limitations in existing systems, articulate the challenges and provide guidelines for the community to work towards a general purpose platform that would enable new diversified battery-less applications in the future.

CCS CONCEPTS

• **Computer systems organization** → **Sensor networks**; **Embedded systems**.

KEYWORDS

Energy Harvesting Systems, Intermittent Computing, Development Platform

ACM Reference Format:
Nurani Saoda, Md Fazlay Rabbi Masum Billah, and Bradford Campbell. 2021. Designing a General Purpose Development Platform for Energy-harvesting Applications. In *The 19th ACM Conference on Embedded Networked Sensor Systems (SenSys'21), November 15–17, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3485730.3493366

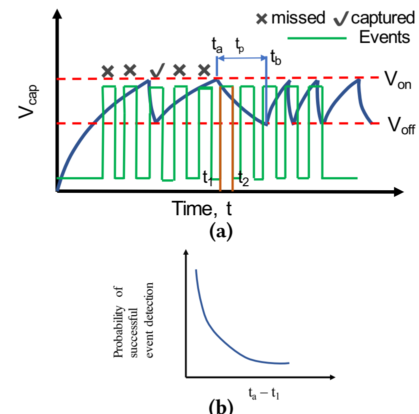

Figure 1: Uncertainty in successfully detecting an event in intermittently-powered systems. Intermittently-powered devices turn on once its capacitor reaches a minimum threshold and performs a routine task. Events that happen during recharging is missed compromising the resolution of the sensed data. (a) depicts a series of missed and captured events throughout capacitor life cycles. t_1, t_2 denote the start and end time of an event and t_p indicates the period of capacitor lifecycle. If energy availability and the event of interest does not coincide, the likelihood of detecting the event decreases as shown in (b).

1 INTRODUCTION

Devices that scavenge energy from environment and stores them momentarily in small capacitors has enabled zero-maintenance and life-long ubiquitous sensing [1, 2, 6]. From smart buildings to wearable health, from massive scale industry applications to academic research, such energy-harvesting devices have shown promising results in sensing, monitoring, and re-configuring, successfully replacing batteries and tethered power supplies. Looking back into the progress made in energy-harvesting systems over the last ten years, one can safely assume the trend will be only upward from now on.

However, developing applications without a stable power is more challenging than the ones with it. The energy-harvester's (e.g. light, kinetic, thermal, RFID) output power, optimum operating voltage varies depending on time, place, and the application's behaviour, which is difficult to characterize for all possible deployment scenarios. Without proper knowledge of the underlying energy dynamics of the system and how that impacts a sensor's working profile, developers end up designing systems that fail to achieve expected outcome. With unreliable power, programs are forced to restart in the middle of an execution, critical data are lost if not explicitly saved, interesting events are missed due to insufficient energy. All these factors make it difficult for an embedded developer to design hardwares and write codes for battery-free applications.

Analyzing the existing works in battery-less systems and from our own experience with developing energy-harvesting applications, we identify a polarizing gap between the extremities of two common design strategies. In one group of these design strategies [1, 3, 5, 9], systems are designed with a specific application

ENSsys, Workshop co-located with ACM SenSys'21, November 17, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9097-2/21/11…$15.00
https://doi.org/10.1145/3485730.3493366

goal in mind with a high degree of co-design in the software and hardware layers. Hardwares are fine-tuned and codes are optimized to work for a known use case. While these design points are simpler to build and achieve good performance, they fail to work in other application scenarios for which it has not been optimized for. On the other hand, another group of work [4, 10, 15] emphasize on developing more general platforms that hides the complexity of co-design from novice developers while at the same time, letting them chose their own peripherals. These systems make application development easier and provides flexibility, but now the developers have very limited access and control over the energy side. We claim that achieving a design point that balances between these two extreme points would further widen the boundary of today's battery-less application.

In this paper, we outline the guidelines to achieve such a design point in energy-harvesting application space that enables sufficient abstraction between the underlying energy complexity and the application, yet providing enough control to the developers by exposing feature-rich energy API and hardware interface. We conceptualize a design architecture that would provide hardware flexibility by letting the developers choose their own hardwares, at the same, a flexible runtime that selects the proper energy-optimization based on the application's behavioral pattern. The architecture integrates a discrete interface to make testing, debugging, and experimentation smooth, fast, and reliable across a variety of applications. We envision that the way platforms like TelosB [12] and Mica [11] revolutionized the research in the wireless sensor network, energy-harvesting research would also flourish further with more general prototyping platforms.

Next in the paper, we articulate the challenges in realizing such standardized, general platform and identify some crucial criteria that would enable such platforms.

2 WHY DEVELOPING A STANDARD HARDWARE PLATFORM IS DIFFICULT?

Though a sheer volume of promising battery-less applications have emerged over the last few years, current systems are often times carefully tailored to a specific set of applications, significantly limiting the usability and scope of such systems. The tight coupling between energy harvesting mechanism and application execution scales poorly, when either of these two parameters deviate from their anticipated behaviours. Following, we identify some crucial factors that pose research challenges towards developing a general purpose energy-harvesting platform.

Generality vs Performance. A battery-less intermittent node attempts to execute a series of tasks e.g., sensing, computation, communication over one or multiple power cycles. Existing systems leverage both hardware-controlled and software-based mechanisms to map peripheral activation or individual tasks according to the energy level of the capacitor. However, the more an application's execution is tightly integrated into its intermittent energy, the harder it gets to re-use that platform on a different application profile. Both energy availability and the occurrence of an external event are stochastic quantity and a slight deviation from their anticipated environment can cause the system to perform unreliably. Figure 1 captures the uncertainty in energy availability and how it affects the percentage of successfully captured events.

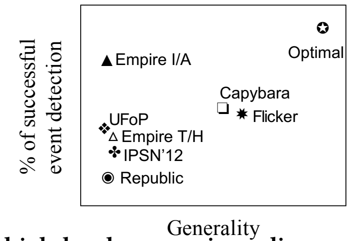

Figure 2: A high-level comparison diagram among some existing battery-less systems designed over past few years. A platform's capability to be used in designs outside their intended use (generality) and the anticipated performance (percentage of successful event detection) oppose each other in most of the cases.

Observing some of the existing platforms, we identify that a platform's ability to generalize across various application domains and ensure reliable performance outcome play at odds against each other. For example, Republic [1] starts with provisioning the capacitor with just enough energy to support its expected workload and intermittently captures vibration events while harvesting from the same event. The sensor fails to capture enough data to identify particular event of interest since the energy availability and the event of interest does not necessarily coincide. The paper solves the problem in Empire design by only activating the harvester when an interesting event occurs, however, making it only suitable for the specific use case. Federated energy platform [9] also suffered from static threshold based activation that does not scale well to different peripheral—a limitation which was later taken into account to design Flicker [10] platform. Flicker provides higher flexibility by providing a reconfigurable activation threshold at runtime. The platform also accommodates for various energy sources and peripheral using universal hardware interface. Capybara [4] activates a series of capacitors depending on a task's energy demand at runtime, providing a scope for generality in the system.

Figure 2 draws a high-level plot capturing the relation between generality and reliable performance and identifies the lack of works that achieves a balance between these two. We reason that finding a balance between these two metrics would be a cornerstone for designing general purpose platforms.

One might argue that the coupling between energy events and the sensed quantity makes sense for some applications. For example, light energy-harvesting sensors could only monitor humidity level of a space when there is actually light (therefore, humans) present, or a thermal energy-harvesting sensor could monitor the temperature of a running shower, only when someone is using hot water. However, we reason that such opportunities could be exploited when possible, but at the same time, a push for enabling sensing where no such co-relation exists is crucial for paving the way for innovation and robust energy-harvesting systems.

Dealing with Uncertainty. The uncertain behaviour of an energy-harvesting node accrues from the very nature of harvestable energy that is dependant on the kind of energy, time, and deployment location. For example, two solar cells with same electrical properties deployed at two nearby locations could operate at different points on its PV curve [13, 14]. Also, the unpredictability from an application's behaviour (e.g., re-transmissions due to packet loss, collisions) propagates back to the input tampering the energy-balance of the system. Even with precise modeling tools and energy emulators like [7, 8], modeling every possible cases is a futile attempt. This makes the implementation of a general technique difficult.

Lack of Re-usable Hardware and Software Interface. The inherent energy-application dependency has forced researchers to build hardwares that tightly combines energy-management circuitry with the rest of the system on a monolithic design due to its simplicity. Writing codes is also energy-dependent, which is often carefully implemented for an intended set of tasks. This rigidity limits a platform's ability to be re-used across diverse applications.

3 GUIDELINES FOR A GENERAL PURPOSE PLATFORM

The challenges explained in the previous section provides a few directions to explore solutions towards a flexible, standard platform. Here, we provide some guidelines that might benefit the energy-harvesting community.

Balance between Generality and Performance. Developing a platform that achieves optimum performance across all possible IoT applications might not be practical, if not impossible. On the other hand, platforms fine-tuned for a specific application scenario do not perform well outside its intended purpose. However, aiming to find out a middle-ground between the two extremities of the design space would push the boundary of battery-less design landscape and enable a plethora of new sensing. Generally, devices harvest from various energy sources and application workload ranges from periodic, event-driven, to long-running. An architecture that supports this diversity would allow hardware modules to be plugged and detached at will and offer a variety of runtime energy-management algorithm to choose from based on the nature of the application's workload.

Runtime Optimization. Systems based on static hardware-triggered activation threshold [1, 9, 14] can not perform optimally when deployed in unpredictable environments. Often times, the activation thresholds are made at design time, with an intended application profile in mind. In the case of plentiful energy, the energy utilization of such systems is extremely low since it can not lower its activation threshold to take advantage of the surplus energy. Translating the design to a new application requires hardware modification, otherwise produces poor performance. Instead, shifting to a runtime energy-management, prediction, optimization, adaptive task-scheduling offers tremendous flexibility, which a general purpose platform could leverage to ensure requirements from diverse application profiles.

Isolated Hardware and Software Interface. One of the major criteria for a standard energy-harvesting prototyping platform is to expose useful hardware interface and provide software APIs with enriched energy-related features. Such physical interface would allow access to necessary voltage channels, critical analog/digital signals, energy debugging channel or a data channel if necessary, hiding the low-level complex circuitry from the IoT application developer. Developers could leverage the software energy APIs built atop the platform for better runtime energy adaption. The key is to find out how to provide proper abstraction while exposing critical energy parameters. We envision that open source implementation of such platforms would encourage embedded developers and prototypists to take energy-harvesting applications to the next generation.

4 CONCLUSION

In order to make the battery-less vision of ubiquitous computing a reality, more and more novel energy-harvesting applications will be designed in both academia and industry domains. However, battery-free applications are yet to enter the mainstream sensing infrastructure. A standard, flexible, and general platform would enable efficient prototyping, testing, and experimentation paving the way to further innovations. In this paper, we identify a limiting gap between current design principles and provide guidelines for the community to bridge the gap which we believe will encourage more research in this direction.

5 ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their insights on improving this paper. This work is supported in part by the National Science Foundation under grant CNS-1823325, and the Strategic Investment Fund at the University of Virginia.

REFERENCES

[1] Mikhail Afanasov, Naveed Anwar Bhatti, Dennis Campagna, Giacomo Caslini, Fabio Massimo Centonze, Koustabh Dolui, Andrea Maioli, Erica Barone, Muhammad Hamad Alizai, Junaid Haroon Siddiqui, et al. 2020. Battery-less zero-maintenance embedded sensing at the mithræum of circus maximus. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 368–381.

[2] Bradford Campbell, Meghan Clark, Samuel DeBruin, Branden Ghena, Neal Jackson, Ye-Sheng Kuo, and Prabal Dutta. 2016. perpetual Sensing for the Built environment. *IEEE Pervasive Computing* 15, 4 (2016), 45–55.

[3] Bradford Campbell and Prabal Dutta. 2014. An energy-harvesting sensor architecture and toolkit for building monitoring and event detection. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. 100–109.

[4] Alexei Colin, Emily Ruppel, and Brandon Lucia. 2018. A reconfigurable energy storage architecture for energy-harvesting devices. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*. 767–781.

[5] Samuel DeBruin, Bradford Campbell, and Prabal Dutta. 2013. Monjolo: An energy-harvesting energy meter architecture. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. 1–14.

[6] EnOcean Self-powered IoT. 2021. https://www.enocean.com/en/products/.

[7] Matthew Furlong, Josiah Hester, Kevin Storer, and Jacob Sorber. 2016. Realistic simulation for tiny batteryless sensors. In *Proceedings of the 4th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*. 23–26.

[8] Kai Geissdoerfer, Mikołaj Chwalisz, and Marco Zimmerling. 2019. Shepherd: A portable testbed for the batteryless iot. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 83–95.

[9] Josiah Hester, Lanny Sitanayah, and Jacob Sorber. 2015. Tragedy of the coulombs: Federating energy storage for tiny, intermittently-powered sensors. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. 5–16.

[10] Josiah Hester and Jacob Sorber. 2017. Flicker: Rapid prototyping for the batteryless internet-of-things. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. 1–13.

[11] Jason L Hill and David E Culler. 2002. Mica: A wireless platform for deeply embedded networks. *IEEE micro* 22, 6 (2002), 12–24.

[12] Joseph Polastre, Robert Szewczyk, and David Culler. 2005. Telos: Enabling ultra-low power wireless research. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005*. IEEE, 364–369.

[13] Nurani Saoda and Bradford Campbell. 2019. No batteries needed: Providing physical context with energy-harvesting beacons. In *Proceedings of the 7th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems*. 15–21.

[14] Lohit Yerva, Brad Campbell, Apoorva Bansal, Thomas Schmid, and Prabal Dutta. 2012. Grafting energy-harvesting leaves onto the sensornet tree. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. 197–208.

[15] Ting Zhu, Ziguo Zhong, Yu Gu, Tian He, and Zhi-Li Zhang. 2009. Leakage-aware energy synchronization for wireless sensor networks. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*. 319–332.