Designing Batteryless Energy-harvesting Sensors for Sustainable Internet-of-Things

by

Nurani Saoda

A dissertation document submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in

Department of Electrical and Computer Engineering
University of Virginia
August, 2023

Doctoral Committee:

Dr. Bradford Campbell (Advisor), Assistant Professor, Computer Science
Dr. John Stankovic (Chair), Professor, Computer Science
Dr. Benton Calhoun, Professor, Electrical and Computer Engineering
Dr. Steven Bowers, Associate Professor, Electrical and Computer Engineering
Dr. Josiah Hester, Associate Professor, Georgia Institute of Technology

APPROVAL SHEET

This

Dissertation

is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Nurani Saoda

This Dissertation has been read and approved by the examing committee:

Advisor: Dr. Bradford Campbell

Advisor:

Committee Member: Dr. John A. Stankovic

Committee Member: Dr. Benton H. Calhoun

Committee Member: Dr. Steven M. Bowers

Committee Member: Dr. Josiah Hester

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Jennifer L. West, School of Engineering and Applied Science

August 2023

ABSTRACT

The Internet-of-Things holds the promise of realizing ubiquitous computing in its full potential. Sensors that work as the fundamental building blocks of the IoT have become an integral part of our everyday lives. They sense, compute, and communicate to monitor humans, pets, wildlife, marine life, plants, crops, buildings, factories, city infrastructures, and many others. As the network of computing devices continues to grow rampantly, in one or two decades, there will be a hundred sensors per person on earth. At this scale, sensors must be long-lived to curtail the intractable cost of maintenance and the negative environmental impact caused by short-lived batteries and outdated electronics.

This dissertation argues and establishes that perpetually-powered energy-harvesting devices, instead of battery-powered ones, are the key to enforcing a sustainable Internet-of-Things. Self-powered devices are perpetual, zero-maintenance, eco-friendly, and pervasively deployable. Together with sustainable power, we emphasize utilizing devices that are already installed in place to enable long-lasting design points through retrofitting and repurposing. However, the energy intermittency inherent in batteryless power supplies imposes two major challenges that limit the adoptability of energy-harvesting sensors: complexity in application design and highly unreliable service quality. To overcome these challenges, we introduce an energy supervisor architecture named ALTAIR, which abstracts the details of energy management from application software to simplify batteryless designs. Moreover, we propose PreFarad, a system architecture that isolates and prioritizes the sensor's energy requirement from the rest of the system components to improve the event detection accuracy of intermittent sensors. Additionally, we extend the functionality of an energy-harvesting power supply to enable two sustainable design points by incorporating new sensing capabilities on existing devices. RETROIOT

upgrades existing IoT devices with additional sensing features as well as an energy-harvesting power supply. Lastly, we demonstrate *SolarWalk* design point that transforms a photovoltaic energy-harvester to an accurate sensor. These systems significantly enhance the capabilities of today's energy-harvesting batteryless IoT sensors.

ACKNOWLEDGEMENTS

The last six years of my life have been perhaps, the most meaningful yet, the most challenging to navigate. But, the ending of this journey made it all worth it. Being a scientist was one of the few things I was fascinated by as a young kid, but, never had I imagined my path would lead to being one. I am forever thankful to the people who made it possible through their unwavering support.

If I had to mention only one person in this section of my dissertation, that would have been my supervisor, Brad Campbell. Brad is not only my PhD mentor but also an inspiration, a role model, and a magician scientist who happens to have the power to make impossible problems easier. I am grateful to have him as my mentor throughout the times when I was clueless to research from being an independent thinker. Thank you for believing in me when I was not able to see myself going far. Thank you for teaching me things as low-level as soldering a PCB to finding problems that are important to solve. Thanks for the conversations about silly things as well as the whiteboard dialogue and timeless brainstorming sessions. I always learnt valuable things from our conversations whether it be about technical or non-technical matters. It's rare to get to work with someone who is not only a uniquely brilliant researcher but also the most dedicated and patient to their students and I am fortunate that I was one of Brad's students.

I would like to greatly thank my committee members— John Stankovic, Ben Calhoun, Steven Bowers, and Josiah Hester, who provided genuine and invaluable thoughts, suggestions, and feedback to improve this thesis. Your insights and implications shaped this dissertation and I deeply thank you for your time. Ben and Josiah's work kept me inspired and motivated to pursue research in energy-harvesting systems. Their questions helped me think deeper about the broader implications and directions going forward. Jack's thought-inducing discussions and Steve's critical thinking helped me identify limitations and potential research directions to explore.

During these years, I was also fortunate to make some true friends who started as colleagues in the beginning and I am indebted to each one of them. Alex, I still have the veroboard that we made together when you first showed me hand-soldering and you suggested that I keep it to look back to remind me that progress starts with small things. Thanks for all the hardware hacking tricks and spending long hours in the lab with me trying to fix each other's boards. Wenpeng, you have been my go-to person for random conversations as well as thinking out loud to understand a new problem. Thank you for being the person I could trust for a 3am editing on a late-night submission. You made those nights doable and less stressful. Victor, you are an asset to a team and our conversations helped me see problems from different angles. Thank you for being so generous with time and being available no matter what. Nabeel, I found peace in our conversations and I am thankful for the surge of positive energy I absorbed from our interactions which sometimes was very much needed. Thank you for all the time you took to provide genuine feedback on a practice talk or a paper. Marhsall, thank you for being the person I could always turn to for help whether it's a quick feedback or a last minute practice talk. Giechao, thank you for making things funny whether it was about a federated learning discussion or a failed experiment, or a hopeless job search mission. Vish, thank you for making my last few days less stressful than they could have been with your thoughts and questions. Fateme, thanks for all your words of assurance and encouragement during the seemingly never-ending weeks. Tushar, Samy, Kai, It was a great pleasure to have you as teammates and thank you for whatever amount of time we had together. Lastly, Masum, you are my biggest motivation to keep going. Thank you for being there both inside and outside the lab when I needed the most. You are a rare scientist who goes beyond their way for the team. Thank you for everything from the first day till the day I defended.

I owe this dissertation to the amazing lady my mother and a great teacher, Farida Yasmin. We made it, Mom! I am indebted to my genius sisters, the other two Nuranis, who crafted my wings to fly high. Borpu, you are my ignition and rejuvenation and Prascovia, you are my incentive to make a difference through my work. I am also blessed to have some wonderful friends by my side who never hesitated to do their best for my survival. Kona and Naz, thank you for listening to my unreasonably long late night rantings. Thank you for not being disappointed when I struggled to make time for a phone call or a much-needed trip. Thanks to my friend, Israt, for being the most supportive roommate and not getting pissed off when my silly experiments broke down our apartment door or when I came home devastated after locking myself out of my laptop the night before my proposal exam.

Dedicated to ...

My Mom, Farida Yasmin

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Ubiquitous computing remarkably extends the reach of sensing by pushing computation into everyday physical objects. One key enabling factor is the growth of Internet-of-Things that envisions every *thing* to have something "smart" embedded into them. The result is computing at a tremendous scale extending from applications in industrial control, building management, citywide environmental monitoring, mass inhabitant (i.e. wildlife and marine life) tracking to implantables and wearables. Industry reports project that the Internet-of-Things (IoT) will have a trillion connected devices by the year 2035 [1], [2]. This scale of computing presents an inevitable challenge of *sustainability*. The energy and resource demands of IoT sensors are continuous, lifelong, and dynamic, however, sensors primarily rely on batteries to power them. Batteries are short-lived and hazardous [3]–[5]. These properties of a power source adversely impact the earth's sustainability as they fail to provide long-term operation and, once depleted, contributes to the enormous volume of harmful landfills. Moreover, battery chemicals like lithium, cobalt, and nickel are finite in reserve and are on the decline [6]. Another major source of IoT e-waste is obsolete electronics, which are hard to recycle and require pressing attention from the computing research community [7], [8].

Devices that harvest energy from the environment to support computation hold the promise to overcome the shortcomings of batteries and are a key to sustainable computing. In basic terms, a batteryless energy-harvesting power supply converts energy from external sources like light, thermal, kinetic, acoustic, chemical, and [9]–[14] to electrical form and stores the harvested energy temporarily in an energy storage like a capacitor. In this dissertation, I redesign and extend the role of an energy-harvesting power supply to introduce several sustainable design points that allow re-usability, retrofitting, and repurposability to support long-term sensing that resists device obsolescence.

1.1 Architectures for Batteryless Computing

Unlike battery-powered sensors, energy-harvesting devices do not have a steady supply of power as the amount of harvestable energy fluctuates with time, location, and type of energy sources. These devices operate under very limited, highly variable, and intermittent (occasionally present) power source, which is hard to model pre-deployment and difficult to predict during runtime [15]. For example, the output power of a AM1454 solar cell varies from $10\,\mu\mathrm{W}$ at $50\,\mathrm{lx}$ while fluorescent light (FL) to $50\,\mu\mathrm{W}$ at $200\,\mathrm{lx}$ FL [16], whereas, the illuminance level at indoors may vary from tens of lx to tens of klx. With a wrist-worn µTEG, harvested power varies from 50-400 µW at indoors and harvested power from RF sources range from a few µWs to tens of µWs[9], [17]. This is mostly insufficient to directly power the components of an IoT sensor: a microcontroller, radio, and a few peripherals. Instead, energy is buffered to a capacitor to drive the load. Depending on the operating principle, energy-harvesting device operation can be broadly categorized into two types: intermittent and energy-neutral operation. An intermittent batteryless sensor accumulates sufficient energy in a small capacitor, turns on once the voltage of the capacitor reaches a safe threshold, performs a chain of sense-compute-transmit tasks until the capacitor depletes, and turns off to recharge and repeat the power cycle. In the energy-neutral operation, the device buffers energy for long energy-draughts and duty-cycles its operation to ensure energy consumption does not exceed reserved energy. In both principles, the system experiences challenges due to limited and intermittent energy.

With an unsteady supply of energy, application operation becomes critically dependent on the current energy state and software becomes tightly energy-coupled. This dependency leads to designs that are monolithic, unscalable, and hard to implement. For example, IoT devices perform several application tasks (for example, sensing, computing, and transmitting) and task-specific decisions (for example, execution, postponement, and sequencing) that are directly dependent on the recent status of energy. This direct integration between energy states and application operation exacerbates the design complexity of energy-harvesting systems. Application developers now have to be an expert in both complex power management and IoT application development. Developers are not only burdened with low-level hardware complexity that comes with an energy-harvesting power supply, but also have to be aware of how the power supply hardware-level decisions may impact the application code. Furthermore, applications become highly platform-dependent, leaving little room for independent development and platform reusability. These limitations impede fast, parallel, and efficient development of energy-harvesting systems. To solve this challenge, I designed ALTAIR that relieves an application from implementing energy monitoring, prediction, and optimization by offloading these tasks to an energy supervisor. The supervisor communicates with the application through a set of APIs, accepts application task requirements, and implements an online reinforcement learning-based optimization algorithm to react to changing energy availability conditions in the post-deployment scenarios. Since the energy supervisor and the main application are separate modules of code, the application's task flow is not directly logically dependent on the outcomes of the supervisor, allowing ALTAIR to be modular, application-independent, and scalable. Additionally, energy supervisor can be modified (for example, adding new APIs or updating the energy management algorithm) without changing application code. Further, ALTAIR offloads the computation run by the energy-supervisor in a dedicated processing unit running in the power supply itself, which eliminates direct integration between the underlying energy-harvesting frontend. This provides a distinct physical interface between the energy-harvesting power supply hardware and the IoT sensor. ALTAIR architecture supports integration between a variety of sensors without re-designing the harvesting circuity, enabling reusable and sustainable design points.

Intermittent energy-harvesting sensors turn on momentarily in between power cycles, spending most of the time recharging their small energy buffer. Many IoT applications are event-based, where interesting events occur in stochastic manner. Therefore, batteryless intermittent sensors are highly unreliable to react to such events. On the contrary, battery-powered devices achieve high accuracy, as they typically remain waiting in sleep mode. For example, a displacement sensor attached to a window to detect intrusions needs to detect and transmit the event instantaneously. An intermittent displacement sensor will fail to report the intrusion event, severely compromising the reliability of the sensor. In low energy-harvesting situations, the sensor spends more time in recharging, rendering the service completely infeasible. To improve the reliability and responsiveness of intermittent sensing systems, I propose a new sensor architecture, PreFarad, that separates the sensing peripheral unit from the rest of device. This separation allows PreFarad to allocate a dedicated small capacitor that can be quickly recharged to power the decoupled sensing peripheral and store any event that may happen before a larger buffer can recharge and transmit the event. Since the average energy required to only detect an event is several times lower than energy-expensive tasks like transmitting a radio packet, the capacitor powering only the sensor is also significantly smaller and can recharge more quickly.

1.2 Sustainable Designs with EH Power Supply

The recent advancements in IoT have the potential to deliver long-term and massive-scale services demanded from the future of smart sensing. Sensors deployed on the trees planted on urban motorways can improve the tree health and city air quality by monitoring the mineral content and moisture level of the soil [18], IoT systems can be installed in remote agricultural farms to enable sustainable crop production [19], and sensors can control the ambient lighting of specific aisles or items in large infrastructures like museums or shopping malls. These sensors monitor infrastructures that last decades and so should the lifespan of the sensors. When devices expire prematurely at a massive scale, they contribute to an uncontrollable amount of e-waste. According to a 2020 UN report, the world produced 17.4 million metric ton e-waste from small IT equipment, the majority of which were discarded in waste bins before eventually thrown into landfills or incinerated [7]. Moreover, the heavy metals found in printed circuit boards including copper (Cu), tin (Sn), and lead (Pb) are toxic to human health and even with proper recycling only 30% of materials can be properly extracted [20]. To reduce the generation of e-waste, US General Services Administration (GSA) and Environmental Protection Agency (EPA) emphasize on the prevention of e-waste by encouraging continued use of electronic products [21], [22].

In reality, sensors lose their full utility way before the silicon wears out. Newer software and hardware keep forcing older devices to face untimely obsolescence. Sensors must adapt to accommodate new use cases to future-proof their service and improve operational longevity. In this dissertation, we argue that retrofitting existing installed devices with additional functionalities and repurposing existing device resources to incorporate new sensing can future-proof IoT installations. For example, sensors monitoring machine vibration in industries can be upgraded with a temperature sensor to alert for likely corrosion and grocery stores can upgrade the humidity sensors installed in shelves with a smart label reporting product expiration date. In another example, an occupancy sensor can also perform person identification. Empowering already deployed sensors with new functionalities and features will prevent untimely discontinuation of the sensor and eliminate the tremendous maintenance cost and labor of complete replacements. Moreover, upgraded services can benefit from the available system support without having to design everything from scratch. This motivated the design of RETROIOT that retrofits IoT deployments by encoding additional new data using the battery terminals of a device in a minimally invasive and low overhead method. The key insight is that some traditional open channels can be manipulated to send arbitrary data without any visibility into the device's hardware and software. One such data channel is the battery port which is simple, universal, and easily accessible and many devices sample their battery voltage and include the readings in their data packets. RETROIOT adds a simple encoder hardware to send new sensor data, metadata, or custom commands by modulating the battery voltage. The modulated data is later decoded at the cloud application. RETROIOT's encoder adds on to the ALTAIR's power supply supervisor to store any pre-programmed metadata and replace the sensor's battery with energy-harvesting upgrade.

Additionally, we propose a technique, *SolarWalk* that extends and repurposes the role of a photovoltaic (PV) energy harvester to perform occupant identification by embracing the noisy power supply jitters originated from human shadow. The key insight is that a person walking past a PV-powered sensor impacts the output voltage of the harvester and the resulting voltage pattern is a unique identifier of the person due to height, body shape, and gait differences in individuals. *SolarWalk* demonstrates a design point to utilize existing resources like the PV cell available on energy-harvesting sensors to learn about environmental context, without requiring completely new devices and hardware overhead. The concept of such PV-enabled sensing unlocks many applications beyond person identification including activity monitoring, occupant tracking, occupancy sensing, avoiding the bulky and invasive installations required of existing solutions [23]–[25].

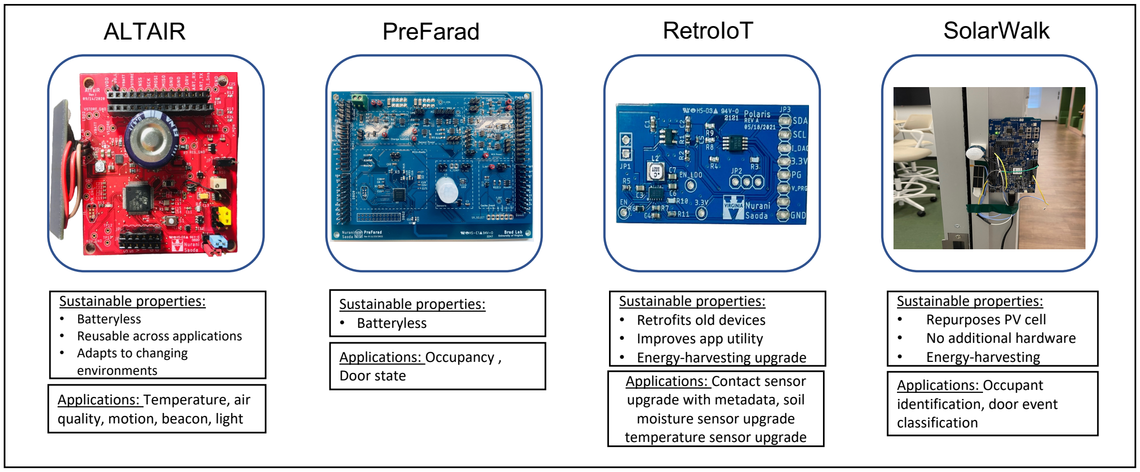


Figure 1.1: Platforms developed by this dissertation to support sustainable IoT systems. ALTAIR is an energy supervisor architecture that reduces batteryless design overhead, while learning to adapt to unseen energy conditions. PreFarad sensors are highly accurate event-driven intermittent batteryless sensors even under low energy condition. RETROIOT and *SolarWalk* are two repurposable design points that equip energy-harvesting sensors with new applications. These platforms enable a class of green batteryless devices for smart indoor sensing.

1.3 Supports for Sustainable IoT

To significantly improve their lifespan, IoT devices must accommodate two key properties: renewable source of power and capability to adapt to changing demands in future. To simplify batteryless application design and improve service reliability under energy intermittency, we introduce two design architectures, prototype the systems in custom hardware, and evaluate the performance in real world deployments. We focus on indoor light energy-harvesting and enable a genre of smart sensors with applications in smart indoor spaces like office, home, and labs. Additionally, we explore two new design points to upgrade deployed IoT systems with additional sensing capabilities from their original designs. The outcome is a class of self-powered green edge computers that support perpetual sustainable sensing (Figure 1.1).

1.4 Thesis Statement

Energy-harvesting sensors are powered from an intermittent and unpredictable energy source resulting in complex application design and unreliable sensing services. A design architecture that abstracts energy management decisions from an application's routine workload, yet exposes critical energy-related parameters at runtime, enables efficient energy-utilization and increases service availability in multi-sensor environments. Such a design method enables composability, provides re-usability, and introduces novel retrofitting applications with energy-harvesting functionality.

1.5 Summary of Contributions

We make the following contributions in this dissertation:

- We propose ALTAIR, a novel energy supervisor architecture for energy-harvesting applications that decouples energy optimization from an embedded application's task execution. Current energy-harvesting systems adopt a monolithic structure with tight interdependency between energy management and application, which increases complexity of application programming and limits portability to new applications. Instead, ALTAIR's energy supervisor allows independent, modular, and faster development. ALTAIR hides the low-level complexity of energy measurement and management from an application developer, yet offers critical energy parameters through the ALTAIR energy API. Instead of imposing additional resource overhead associated with the energy supervisor, the proposed system offloads computation to a separate processor. We develop ALTAIR hardware platform in a custom PCB, design six different IoT applications ranging from custom-made to COTS devices, and deploy the sensors in various indoor settings to analyze the performance of an online energy manager algorithm that learns to adjust application's dutycycle post-deployment. We discuss the approach in detail in Chapter 3.

- Intermittent batteryless sensors are extremely unreliable for event-driven IoT applications. The turn on rate of an intermittent sensors is controlled by the harvestable energy the sensor experiences in their installation point. In a low energy environment, the device spends most of its power cycle recharging before turning on for a very short period of time, which results in poor event detection accuracy. To improve the detection accuracy of intermittent batteryless systems, we design PreFarad that separates sensing from processing and transmission. The key insight is that the energy required to detect an event is significantly lower than combined energy requirement of processing and transmission. PreFarad adopts a small capacitor to mimic an "always-on" low-power subsystem that powers only the sensor. Since, a smaller capacitor can recharge faster, PreFarad's sensor subsystem becomes instantly available. This improves event detection accuracy over a common capacitor intermittent system. We develop PreFarad in a custom hardware platform and incorporate two event-based IoT applications: movement detection using PIR sensors and contact sensing with magnetic field sensors. We evaluate the performance of the proposed approach by deploying the sensors in two different indoor locations. Chapter 4 describes the overall approach and limitations of the system.

- We design, RETROIOT, a technique that retrofits deployed IoT devices with new functional features by modulating additional data through open data channels of the device. Commercial IoT devices are difficult to upgrade due to lack of transparency into their hardware and software, forcing them to become obsolete and get replaced as new demands arise. Through RETROIOT, we demonstrate that even a closed-source IoT system can be retrofitted to allow new sensing and data with simple add-ons without replacing existing installations. RETROIOT is motivated from the key observations that batteries have accessible ports in battery-powered IoT sensors and battery readings are periodically reported by the sensor as an indication to future replacements. We extend the upgrade to accommodate an energy-harvesting power supply which then completely replaces the battery and manages the limited available energy by dutycycling the device. We design two prototype encoder to implement RETROIOT and upgrade three COTS IoT sensors with new applications. We describe the technique in Chapter 5.

- We introduce, *SolarWalk*, a sustainable design point that performs accurate sensing using photovoltaic harvesters. *SolarWalk* light energy-harvesting sensors can distinguish occupants and their movement direction in a smart home by analyzing the noise introduced on the output voltage of the harvester as they walk past installation points. Current smart home occupant identification systems either require dedicated hardware resources or multiple devices, which limits flexible deployment. *SolarWalk* systems builds on PV cells that are widely used a power source in energy-harvesting sensors. We build a prototype hardware of *SolarWalk* and evaluate the identification accuracy on the data collected from five participants in two different setups. We elaborate the design in Chapter 6.

Chapter 2

Related Work

From smart buildings to wearable health, from massive-scale industry applications to academic research, energy-harvesting devices have shown promising results in sensing, monitoring, and re-configuring, replacing batteries and tethered power supplies. Battery-less sensing holds the key to future IoT which will eliminate cost and labor associated with limited lifetime batteries [26]–[28]. Looking back on 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. This progress has been enforced by the recent advancement in ultra-low power ultra-small microcontrollers, radios, MEMS sensors, and better operating systems informed by careful power management, efficient programming, and debug support. In this section, we discuss the landscape of energy-harvesting sensor design in terms of different hardware-software approaches, applications, and energy management and highlight research motivated by post-design reconfigurability.

2.1 Energy-harvesting Systems for IoT

2.1.1 Hardware-Software Approaches for Batteryless Design

Existing works for energy harvesting systems can be broadly categorized into two directions: Intermittent systems which perform operations whenever there is enough energy; Non-intermittent systems which store the harvested energy and implements a dynamic power management-based duty-cycling to maximize energy utilization. Intermittent systems incorporate small energy buffers to store energy temporarily and perform simple tasks when the stored energy reaches certain threshold until the buffer depletes. Gecko [29] and Monjolo [30] are such intermittent system that proposed the idea of measuring different phenomena like energy metering, water metering, door state sensing, occupancy by analyzing the activation frequency of the device. UFoP [31] presented the first federated energy storage approach for intermittently powered sensors. UFoP adopts static federated energy with static capacitor sizes and charging thresholds to achieve higher availability and energy efficiency than a centralized energy system. Flicker [32] improves the flexibility and efficiency of UFoP design and enables rapid prototyping for batteryless Internet of Things by modularizing multiple applications. Capybara [33] goes a step further, by dynamically resizing a bank of capacitosr to match the energy requirement by a task, which reduces cold start time as well as capacitor recharge time to support a given operation. In [34], authors proposed a new architecture and toolkit for energy harvesting systems, which masks the inevitable intermittency with a variety of trigger abstractions that activates the device for certain conditions. Signpost platform [35] is proposed as a generalized energy-harvesting platform for city-scale sensing using a shared backplane to interconnect and isolate each module, allowing energy to be used for particular module. Software based techniques introduce intermittent-aware programming models and compilers to resiliency in between power cycles by checkpointing program states [36]–[38].

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 [27], [30], [31], [34], systems are designed with a specific application 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 [32], [39], [40] 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 identify that achieving a design point that balances between these two extreme points would further widen the boundary of today's battery-less application.

2.1.2 Techniques for Energy Management

Reinforcement learning (RL) has already been adopted in providing a dynamic energy management for energy harvesting nodes. Hsu et al. [41] introduced a dynamic power manager for energy harvesting networks, where they use Q-learning algorithm on an agent to select within four different level of duty-cycles. Another paper provides dynamic power management in ensuring the requirement of throughput along with the battery's energy level [42], they also added penalties considering the condition of energy storage, which includes overcharging, deep-discharge, and depletion. Different from other approaches, Rioual et al. [43] focus on refining the reward function and discussed the choice of rewards in energy-harvesting IoT nodes. RL algorithms are also used to solve energy allocation problems in energy-harvesting systems. SARSA algorithm was introduced by Ortiz et al [44] to learn a power allocation policy in two-hop communications and maximize the throughput of a communication system. SARSA(λ) was also introduced to develop an adaptive power management algorithm for solar-energy-harvesting nodes [45]. Their reward function was designed based on the distance of energy neutrality, and trained the agent in an episode of 24 hours.

2.1.3 Event Detection with Intermittent Sensors

The stochastic nature of intermittent energy makes intermittent energy-harvesting sensors a poor choice for event-based IoT applications. Many sensor-based automation systems requires reacting to an asynchronous event accurately and in real-time. However, since intermittent sensors are momentarily on and usually spends a large fraction of their power cycle recharging the energy buffer, there success rate of event detection suffers

The SmartON platform in [46] uses on-device reinforcement learning and Ember platform in [47] uses off-device reinforcement learning to train the sensor to predict the occurrence of an event to wake up timely. These system rely on supercapacitors to buffer energy for several hours to several days. These approaches require computation heavy learning technique offline on large amount of historic data and frequent communication with the device itself which adds overhead. [31] partitions the energy buffer for each components of a sensor which improves the availability of the MCU and radio. However, the radio is powered by a relatively larger capacitor which may fail to detect an event on-demand.

[34] adopts event-based activation of sensors by powering from the side channel of energy associated with the event itself. Similarly, Empire [27] designed a platform to monitor structural vibration only activating the the sensor when the specific event of interest occurs. Both approaches rely on coupling between side channel of energy and the sensed quantity, making them highly application-specific.

2.2 Reconfigurable Design Space

Commercial IoT systems have been retrofitting legacy systems in condition monitoring, predictive maintenance, transparency in supply chain, etc. [48]–[51] over the past decades. In this section, we briefly discuss work related to enhancing existing systems or interfaces with new techniques.

One of the possible ways of retrofitting the IoT network is adding sensing capabilities by attaching extra sensors or exploiting the wireless medium. Penichet et al. presents passive sensor tags [52], where the IoT network can be augmented with a new sensor by placing a passive back-scatter sensor tag with the desired capability next to the already deployed devices. Since the proposed method lacks the media access control capabilities, it only demonstrates the prototype in low-density networks. LoRaBee [53] is presented as a LoRa to ZigBee cross-technology communication approach, which leverages the energy emission in the sub-1 GHz bands as the carrier to deliver information. LoRaBee tunes the LoRa's central carrier frequency and packet payload, so that a ZigBee device can decode the information carried by LoRa by sampling the RSS. This demonstrate a technique of backwards compatibility between existing devices and newer additions. RetroFab was introduced to provide an end-to-end design and fabrication environment to retrofit the hardware interface of legacy devices [54]. Another direction of retrofitting existing networks is to replace existing gateway with an generic gateway, whereas the devices itself remain unchanged, but the gateway would intercept their data stream at the next hop. Adding new sensors on generic gateways can add new sensing capabilities to the network-wide operation. iGateLink introduces a pluggable design to allow data from different sources that can be easily reused on edge without sending everything to the cloud [55]. This also speeds up the development of gateway applications. In real deployments, these approaches are not ideal for existing commercial IoT systems, due to a few reasons. The cost and complexity of recreating each layer is high. It is likely that existing devices cannot be changed since it requires specific software. Also, existing IoT platforms might be very rigid in the devices and the type of devices they support. Deploying an entirely new embedded-gateway-cloud system is another option. However, this approach is costly and does not leverage legacy systems, thus are not favorable for the end users. More traditional approaches of updating IoT sensors are over-the-programming, which mostly require a wired connection while reprogramming and a network-wide update assumes all the sensors in a deployment are same. Reconfigurable hardware platforms like FPGAs are emerging to accommodate evolvable computing. However, such platforms are yet to widely adopted like microcontrollers due to high energy requirement, cost, and lack of community support.

2.2.1 Exploiting Device Side Channels

The idea of augmenting versatile user interfaces of ubiquitous mobile devices have been explored in prior works. Kuo et al. designed HiJack [56] that exploits the exposed audio ports of mobile phone to encode additional data as well as harvest energy for operation. Nirjon et al. presented MusicalHeart [57] a wearable hardware platform to monitor the heart rate and activity level of the user which communicates the sensed data to the user mobile device using the audio jack of earphone. In our work, we focus on exploiting the battery voltage channel of smart IoT devices not only with the goal of adding sensor data, but also to eventually make the original device energy-harvesting and perpetual.

2.3 Repurposable Energy-harvesting Sensors

2.3.1 Harvester-enabled Sensing

Instrumenting the power sources of energy-harvesting devices to generate more expressive and meaningful data have been explored in several prior works. Monjolo presents an energy-harvesting power meter where the rate of harvesting energy by power source is leveraged to calculate how much current is consumed by the attached load. It demonstrates a case of redefining the harvester to produce more expressive data. While Monjolo exploits the recharge rate of an intermittent energy-harvesting node to infer energy-metering information, we demonstrate that even noisy solar cell voltage data can have context-rich data. In [34], authors infers the occupancy status of a room by simply inspecting if the node was able to harvest, wake up, and transmit a beacon. Since the node was able to harvest signifies, the room was lit and someone could be present. These works attempt to infer intuitive information exploiting the energy-harvester that comes as an inevitable design choice of self-powered systems.

2.3.2 Indoor Occupant Identification

Scalable and cost-efficient unobtrusive occupancy detection setups for smart buildings have been attempted previously in several works. Various types of sensors have been exploited for satisfying the myriad requirements of identification at diversified smart environments, which include RF-based [58]–[60], Ultrasonic-based [61] techniques or also collecting information from on-object sensors [62].

Utilizing cellular frequencies, BlueSentinel [58] has been proposed that can detect the number of users in a room and track them inside the building. Including iBeacon's location information and KNN as the classifier, BlueSentinel achieves accuracy near 83%. A similar methodology was deployed on a large scale in [59], encompassing diversified surroundings like office buildings and dormitories on a university campus. However, electricity and water consumption information was added to WiFi data over 4 weeks duration with occupancy varying from 0 to 550. Mean absolute percentage error exhibits that incorporation of multi-modal data to estimate the occupancy escalates detection accuracy. As height and weight combination is a unique feature for personalizing, non-intrusive occupant identification has been proposed by utilizing those features in [60]. This system takes into account 7 distinct features of a human being (including hand weight distance, bouncing pattern during walking etc.) for identification. After evaluating in multiple test beds, it has been demonstrated this system can detect a person with accuracy varying from 90%-100%. In [61], authors present MODES, which utilizes thermal and vibration information with an accuracy of *73%* and *84%* in high and low occupancy scenarios. However, all these techniques require heavy infrastructure, multiple device installations or carried devices which fails to achieve scalability.

In another branch of work, researchers have focused on utilizing on-object sensors to infer occupancy. For example, SenseTribute [62] collects personalized features from different on-object sensors such as accelerometers and gyroscopes installed on domestic utility products (refrigerator, towel dispenser etc.) to classify occupants. Since, different occupants interact with an object in different manners such as the pattern of knocking on a door, or opening a fridge, the vibration data collected from the attached sensors can be a unique personal attribute. SenseTribute achieves an identification accuracy of 74% and 96% for known and unknown training labels. MotionSync [63] proposes an approach to determine personalized energy consumption by occupants by finding the correlation between motion data from users' wearables and appliances. It classifies the appliances in five categories based on their interfaces to learn the interaction between user-appliance. We share similar motivations of these work to eliminate the need for infrastructure-heavy methods, rather exploit already existing sensors and augment them with richer capabilities.

Chapter 3

ALTAIR: Energy Supervisor for Energy-harvesting Systems

Converting a battery-powered application to energy-harvesting is not as straightforward as replacing the battery with a harvester. Harvestable energy is usually very limited, intermittent, and unpredictable which requires special hardware and software support to achieve useful operation [13], [36], [64], [65]. The operating principle of battery-less energy-harvesting applications can be broadly categorized into two approaches: intermittently-powered and energy-neutral. The first category of sensors harvest energy from the environment through solar, RF, thermal, and kinetic sources, store the energy momentarily in a capacitor, operate until the capacitor is depleted, and repeat this cycle continuously, while the latter store energy for future use and regulate the operational frequency of the sensor to ensure that the outgoing energy roughly matches the combined incoming and stored energy.

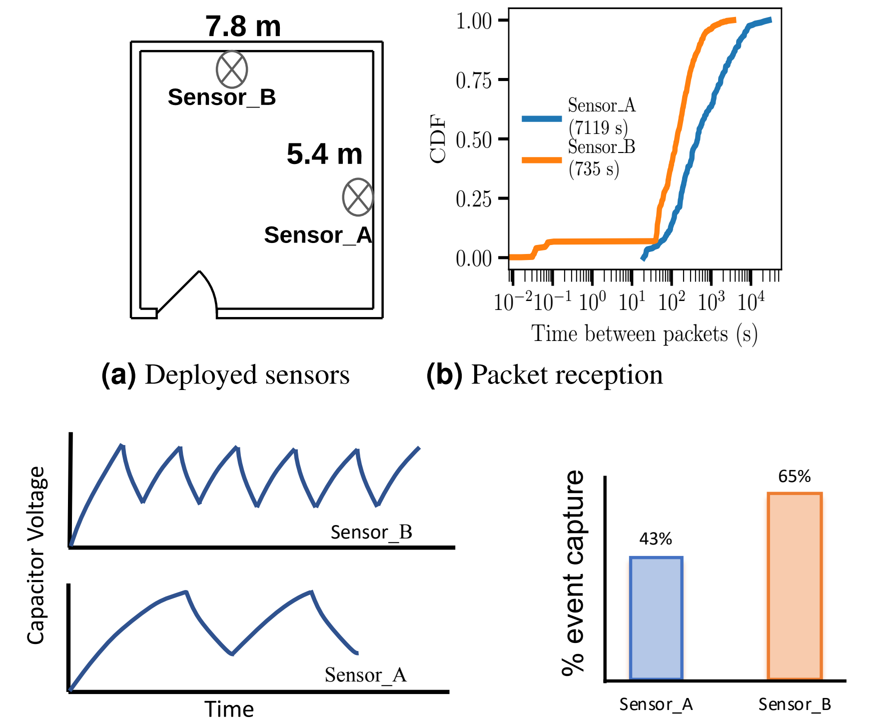
Various designs implement these techniques to realize energy-harvesting systems, including hardware-based [14], [31], [33], [66] and software-based solutions [36], [37], [67], [68]. In both cases, however, energy-harvesting systems typically consist of a single processor along with an energy-harvesting front-end and application peripherals, where the processor is responsible for both energy management tasks (i.e. tracking the amount of energy stored, controlling the wake-up time interval, turning on peripherals at specific voltage levels, etc.), and application-specific tasks (i.e. sampling, computation, and transmitting radio packets). While this monolithic architecture can be simple and efficient for the intended application, adopting these platforms to build new applications can be quite difficult due to tightly-coupled implementations of energy-management code and application code. The intertwined application and energy management requires the developer to be responsible for understanding not only how to manage energy and correctly implement the application, but also how the two halves might interact.

The coupled implementation of application tasks and energy-management in energy-harvesting limits scalability, increases complexity, and impedes efficient energy-harvesting system development. To address these limitations, in this work, we propose ALTAIR, a modular architecture for energy-harvesting system design that decouples energy management from application execution. ALTAIR offloads energy forecasting, allocation, measurement, and management to the power supply itself, therefore, the applications no longer have to integrate these tasks. With ALTAIR, application platforms can focus on the IoT task (as they would with a battery-based power supply), and the new "smart" power supply can make intelligent decisions about when the application should wake up, what operating mode it should be in, and how long it should stay active, based on its careful knowledge of the energy states. Since the energy-optimization algorithms and power supply are tightly coupled, they can be highly optimized, and must only be implemented once. Many application-level platforms can leverage the same power supply. Further, the energy supervisor can handle the uncertainty in energy-harvesting system deployment, relieving each application from needing to consider the range of potential deployment conditions it might face, and instead allowing the power supply to adapt to the local conditions post deployment.

By eliminating the tight coupling between energy state and application task ALTAIR reduces the high-degree of co-design between hardware-software for energy-harvesting applications, and achieves modularity for independent application development. We also propose a standard interface between the power supply and the sensor that can be re-used across multiple application platforms without requiring any significant changes. The re-usability and modularity are two crucial parameters for a platform to be general across different types of applications including both periodic and event-driven. We envision that ALTAIR is a step towards general platform for a wide range of battery-less applications.

3.1 System Design Challenges

Energy-harvesting devices must balance an unreliable source of energy with application-level goals. Coupling an application's task flow to an unreliable source of energy makes



(a) Deployed sensors **(b)** Packet reception

(c) Difference in sensor duty-cycles **(d)** Event detection comparison

Figure 3.1: Two energy-harvesting sensors in room a) transmit at a rate shown in b). Performance varies significantly indicating high energy variability of indoor solar energy. Different duty cycles in c) result in different event detection percentage in d).

energy-harvesting systems difficult to develop and debug, and can result in poor performance. Often, the application's task i.e., sensing, computing, or transmitting, is carefully mapped to the recent energy state of the energy storage. This tight integration between an application's task flow and energy availability significantly limits today's battery-less systems in several ways.

Suboptimal performance. With a high degree of energy-application coupling, an application's execution becomes highly energy-dependent. With unreliable energy, the application needs to perform complex software checkpointing techniques to ensure forward progress, which is not always guaranteed. Application programs can enter an endless inactive loop [37], [69], producing suboptimal performance. The complexity, uncertainty, and software overhead induced in intermittent computing indicate a need for alternative approaches to design energy-harvesting systems.

Runtime energy optimization. When an application's task execution is directly mapped to its energy status, this mapping is often performed at design time and is not optimized or re-evaluated during runtime. Decisions made at design time fail to scale post deployment. Since the nature of harvestable energy is time, space, and source dependent, modeling accurate energy states for all possible scenarios apriori is non-trivial. Figure 3.1 shows two co-located intermittently-powered solar energy-harvesting nodes that both transmit a radio packet each time their capacitor reaches a certain voltage. Though deployed in relatively similar environments, the harvesting rate of the sensors varies quite significantly resulting in different throughput and availability, which is hard to model at design time. Non-linear device parameters are another source of stochasticity in energy-harvesting design. For example, two sensors deployed nearby and powered by the same type of PV cell could operate at different points on its PV curve at a given time and therefore, produce different output power. Different output power results in different capacitor recharge times. Both of these two relations are stochastic and non-linear and fixed design time decisions produce suboptimal performance in post-deployment phases indicating the importance of runtime energy modeling.

Impedes development. Developing applications with unstable power requires more expertise, development time, and rigorous testing and debugging than with reliable power. With the application's behavior being energy-coupled, developers have to carefully implement everything from the low-level energy-harvesting hardware circuitry to writing optimized code within the system's limited energy budget. This creates a large burden on an IoT application developer. Moreover, finding the optimal design strategy often takes multiple design-test-deployment cycles. Successful and smooth battery-less development requires a well-balance between providing enough abstraction as well as control into the underlying energy optimization mechanism [70].

This combination of challenges suggests that a different design architecture for energy-harvesting is required.

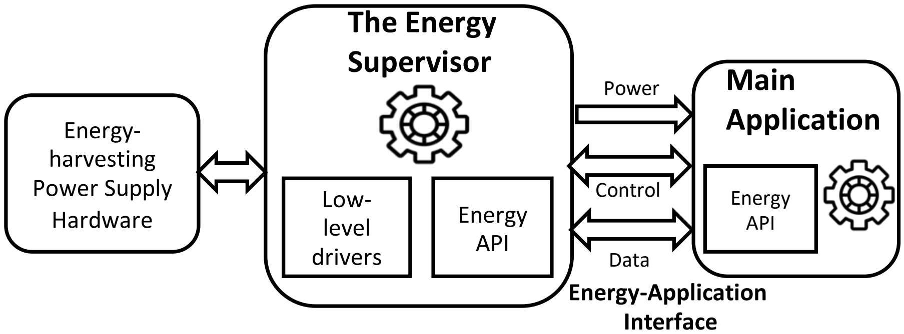


Figure 3.2: Overview of ALTAIR energy supervisor architecture.

3.2 Overview of ALTAIR

We propose ALTAIR, a new energy-management architecture for energy-harvesting applications that decouples energy related decisions from an embedded application's task execution. This separation introduces an abstraction layer between the application and power management which enables independent, modular, and faster design of both subsystems. ALTAIR hides the low-level complexity of energy measurement and management from an application developer, while exposing critical energy parameters through the ALTAIR energy API.

Figure 4.4 depicts the high-level overview of the ALTAIR energy supervisor architecture. The design consists of three core components: the energy supervisor, the energy-application interface, and the main application. The energy supervisor monitors the energy states of the storage along with load energy consumption and determines the optimal duty-cycle to achieve energy-neutral operation within the limited energy budget. The supervisor works as a wrapper function that implements power supply functionality and an interface to facilitate calls between the supervisor functions and main application. The main application implements the application specific tasks of an IoT sensor such as sampling, computation, and data communication, and makes call into the energy supervisor using the interface. The energy-application interface handles requests from the main application, defines the function-specific input/output parameters, and ensures reliable data communication. Algorithm 1 outlines how the application and the supervisor can interact. The function MAIN invokes ENERGY_SUPERVISOR specifying application requirements $(p_1, p_2, ..)$ to receive the rate at which a task is performed. Instead of tying an application's task with the specific energy status of the storage as done in many battery-less applications, the main application offloads the decision to determine an optimal wake-up rate of the sensor to the energy supervisor. This way, the dependence between the energy supervisor and the application is reduced.

3.2.1 Enabled Properties

ALTAIR enables several desired properties of energy-harvesting system design that traditional implementations often cannot. It introduces a general, reusable, and reliable application-power supply interface for energy-harvesting applications and achieves independent and modular design. Since the energy supervisor and the main application are separate modules of code and the application's task flow is not directly logically dependent on the outcomes of the supervisor, development can be performed in a parallel fashion. This decoupling also simplifies adding new APIs to the energy-supervisor and new functionality in the application. A standard interface between the energy-harvesting power supply hardware and the IoT sensor enables integrating a variety of sensors with a single power supply without re-designing the harvesting circuity or energy management logic, enabling reusability and scalability of the platform. Also, since the application does not interact with the underlying energy-harvesting power supply hardware, the IoT application developer does not need to implement power-supply specific drivers in the application code. Moreover, though we propose ALTAIR for energy-harvesting applications, the general architecture can be adopted in battery-powered IoT and mobile applications as well as for advanced power optimization.

3.3 ALTAIR System Design

An IoT application interfaces with the energy supervisor of ALTAIR to maximize its energy utilization. In this section, we discuss the core components of the architecture and how they interact. We also investigate the design choices to understand the trade-offs in the design space.

Algorithm 1

```
function ENERGY_SUPERVISOR (p_1, p_2, .., p_n)
    return action_rate
function APP_ROUTINE (rate) // application task code
    return
function MAIN
    After each t_period {
        rate = ENERGY_SUPERVISOR (p_1, p_2, .., p_n)
        APP_ROUTINE (rate) }
```

3.3.1 Design Space Trade-off

We note that the isolation between the energy management and application sub-blocks proposed by ALTAIR can be implemented in both software and hardware. In software, this isolation would be possible by delegating the energy management portion in a separate module with the implementation of appropriate interface functions accessed by the main application. In the hardware version, the energy management functionality could be executed in a separate core or a processor with dedicated hardware resources. We identify some crucial factors when choosing between these various design points. While implementing ALTAIR as a software component would provide the desired logic detanglement and independent code development, we advocate for the hardware version of ALTAIR design to take advantage of several benefits.

Minimal resource conflict. Today's IoT devices are extremely resource-constrained due to their size and power restrictions, yet, they are expected to perform a diverse range of processing-intensive applications. Such applications include critical real-time processing, multi-radio wireless communications, and even running machine learning inferences. Typically these computation-intensive tasks are handled in real-time by a low-end microcontroller causing significant burden on the shared memory and CPU bandwidth. Adding an online energy management algorithm would exacerbate these concerns. Instead, we leverage an ultra-low power microcontroller with dedicated clock, memory, and I/O bandwidth to execute the energy supervisor in parallel with the application.

Decoupling in the power domain. Using hardware isolation and adding additional hardware components to the system might impose an additional energy cost in an energy-harvesting application. However, we argue that the average energy overhead can actually be reduced by leveraging a lower power core than the main application. As these two

Table 3.1: List of ALTAIR APIs.

Energy Supervisor	**Main Application**
c_param_t *get_critical_parameters()*	dc_t *get_optimal_dutycycle()*
list_param_t *get_app_list()*	double *get_current_energy_status()*
mode_param_t *get_power_modes()*	int *get_update_period()*
	model_array_t *get_energy_model()*

cores are decoupled in the power domain and they can be turned on/off independently, one can reduce the overall energy cost. This architecture has been implemented by silicon vendors in many low power dual-core processors [71], [72]. Furthermore, the energy-management core can be further power-optimized with the recent growth of ultra-low power chip technology.

Reusability and generality A hardware implementation of ALTAIR accelerates the development phase and reduces developer effort by providing modularity and reusability across multiple applications. To promote reusability, we adopt the hardware-accelerated software energy management of ALTAIR and implement the energy supervisor in a lower power microcontroller taking inspiration from the ARM's big.LITTLE technology [73] that leverages a smaller lower power core to enable power optimization. In the evaluation, we test the performance with a variety of IoT sensors and demonstrate the composability and generality of the platform. This enables future embedded designers to rapidly develop their own applications while adopting energy-harvesting functionality.

3.3.2 The Energy Supervisor

The energy supervisor of ALTAIR handles the tasks of energy management, prediction, and allocation, and makes decisions independently from the application logic. To accomplish this, the energy supervisor has two key components. First, the supervisor interacts with an energy-harvesting front-end to collect useful information about the harvesting

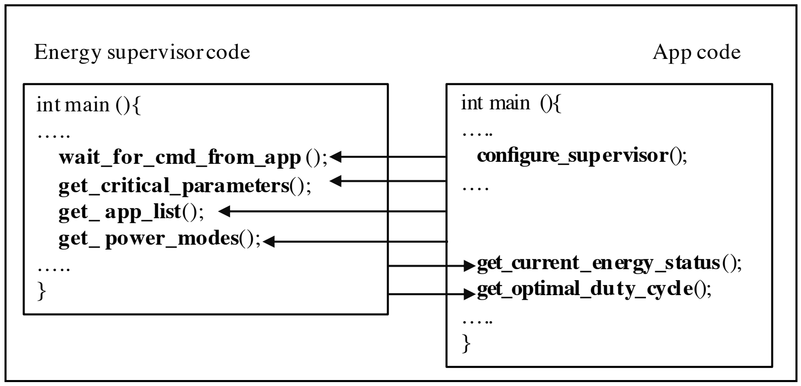


Figure 3.3: Example workflow diagram between the application and energy supervisor. The direction of the arrow specifies the direction of API calls.

conditions. This information includes the average input power, the charging rate of the storage, and instantaneous and average stored energy. The energy-harvesting front-end typically accommodates an energy-harvester (e.g. solar, RF, thermal, or piezoelectricity), a charge controller, and an energy storage (e.g. capacitor). Second, the supervisor implements the dynamic power management scheme and the interface presented to the main application. For dynamic energy management, the application can specify the parameters (i.e., duty-cycle) to be optimized and an optimization algorithm among the supported ones. The supervisor can also inform the application about which operating mode the application peripherals should be running in, or the recommended order of priorities for multiple applications.

The supervisor makes power management decisions by keeping track of system's past experience and predicting future expected energy incomes. Learning and adapting the optimization parameters at runtime, as opposed to fixed design time or datasheet parameters, makes the energy supervisor more robust to real-world deployment conditions. The supervisor attempts to support any type of application workload. However, as the underlying hardware can only buffer a finite amount of energy, the average energy consumption of the application must be below the maximum buffered energy.

3.3.3 Energy-Application Interface

The energy-application interface enables the abstraction layer between the main application and the energy supervisor module. It facilitates communication between the energy supervisor and the main application by implementing a set of useful APIs. This standard interface enables updates and improvements to the energy supervisor and any optimization algorithms without requiring direct changes in the application.

ALTAIR Energy API. Table 3.1 shows the list of available APIs provided by ALTAIR. The energy supervisor calls *get_critical_parameters*, *get_app_list*, and *get_power_modes* to acquire application or device specific information. These are fixed configuration parameters of the application that are not expected to change at runtime. *get_critical_parameters* returns an array of permitted duty-cycles of the running application, according to which the energy supervisor optimizes for long term energy neutrality, and which energy optimization algorithm from the supported ones to use. Currently, the platform implements three duty-cycling mechanisms (described in Section 3.5.2). To understand how energy is being spent, *get_app_list* provides the list of energy-atomic operations performed by the application. Energy-atomic operations are categorized into sampling a sensor, computing and analysing the sampled data, transmitting data, or receiving data. Each of these operations is associated with a unique operation ID. The application specifies the required operating power modes using *get_power_modes*. ALTAIR saves this information into the non-volatile memory of the energy supervisor to eliminate the need to repeat the APIs calls after a power failure.

On the application side, ALTAIR provides another four API, namely *get_current_energy-_status*, *get_optimal_dutycycle*, *get_update_period*, and *get_energy_model*. *get_optimal_dutycycle* returns the calculated optimal duty-cycle which is one of the values specified by *get_critical-_parameters* and the power modes of each operation. The application performs sensor sampling, computation, and communication at this optimal rate and enters sleep mode in between operations. The *get_update_period* returns at what interval the application should check for the updated duty-cycle. This depends on how variable the incoming energy profile of the device is (defaults to 15 minutes). The *get_current_energy_status* and *get_energy_model* offer finer insight into the system's energy status. By calling these, the application receives the current stored energy on the capacitor and the current numeric input values used by the duty-cycle algorithm to calculate the duty-cycle, respectively.

Hardware Energy Interface. The hardware energy interface consists of the hardware abstraction layer that configures the hardware interface between the supervisor and the application. Each API call is executed by a set of hardware signals and a data communication channel. The interface consists of voltage, control, and data channel as shown in Figure 4.4. The data channel enables a synchronous communication channel between two processors where the application processor provides the clock signal. When the application processor makes a call into the API functions, it sends an interrupt signal to the energy processor. The energy processor uses the interrupt to configure the communication hardware and initiate data transfer. The energy API calls described in the previous section are translated into data packets. The first byte of energy API packet encapsulates header information specifying the intended API call and a read/write bit, and the next two bytes specify the message length. To invoke the energy supervisor to call an API, the main application sends a write request and an API call from the application is sent as a read request. Both processors avoid sending a new request if there is any previous unresolved or pending request. We also keep a timeout timer to avoid a communication deadlock.

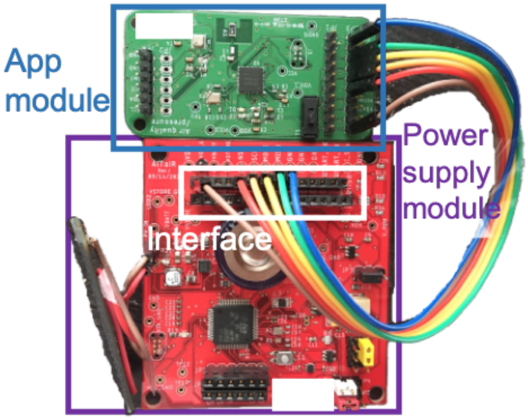
Figure 3.3 shows a flow diagram between the energy supervisor and the application code using ALTAIR energy API. Upon startup, the main application uses the *configure_supervisor* to send write requests and prompt the energy supervisor to call the next three functions for configuration. *get_current_energy_status* and *get_energy_model* is called at any time application, while, the *get_optimal_dutycycle* is invoked according to *get_update_period*.

3.3.4 The Main Application

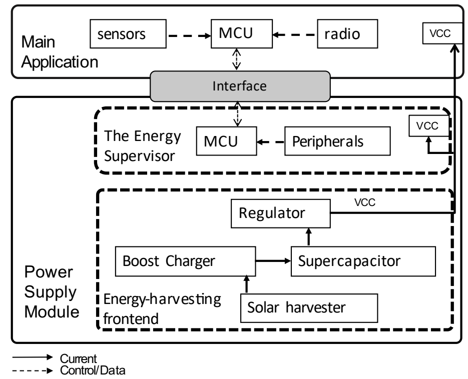
The main application is a piece of software that performs the typical workload of an IoT sensor, i.e. sampling, computing, processing, and transmitting.

3.4 Implementation

We implement the ALTAIR energy-harvesting power supply module in a custom PCB.



(a) ALTAIR Power supply board interfacing with a sensor.



(b) Hardware block diagram.

Figure 3.4: The ALTAIR hardware platform consists of a power supply module that implements the energy supervisor and a discrete power supply application interface that can be plugged in directly with an external application.

3.4.1 Hardware Components

The ALTAIR hardware consists of two primary modules: a power system module and an external application module. The power system module implements the energy supervisor, low level energy-harvesting hardware, and the hardware interface between the energy supervisor and the main application. The main application is representative of a typical IoT sensing application that is powered through the power supply interface.

Power supply module. The power supply module of ALTAIR hardware accommodates an energy-harvesting front-end and a companion microcontroller that implements the energy supervisor software. Figure 3.4 shows the power supply module and the block diagram of the core components.

An ultra-low power battery charger IC SPV1050 charges the supercapacitor from a solar or TEG harvester until it reaches $3.1\,\mathrm{V}$. A nano-power boost regulator MAX17222 with $>70\%$ efficiency at $10\,\mu\mathrm{A}$ of input current regulates the supercapacitor voltage after its voltage reaches 2 V. The platform currently uses a monocrystalline IXYS solar cell as the harvester and a 470 mF supercapacitor with an ESR value of $25\,\Omega$ as electrical storage. We size the capacitor empirically to ensure that it can supply the highest system peak current.

The energy supervisor uses an ultra-low power 32-bit ARM Cortex-M0+ with a 8 kB of SRAM and 64 kB of flash with different low power modes. The power supply consists of a current-sense amplifier MAX9634 to keep track of the load energy consumption. A nano-power power gating IC TPL5110 with reconfigurable time interval allows the MCU to duty cycle the application in hardware with minimal calibration. The MCU leverages a digital potentiometer to dynamically reconfigure the time interval according to the calculated duty cycle.

The interface. The interface of the power supply module provides two voltage rails of $3.3\,\mathrm{V}$ and $1.8\,\mathrm{V}$, one duty-cycled voltage rail, capacitor voltage output. We use SPI to exchange information between the two microcontrollers and one GPIO to trigger interrupts. For debugging and evaluation, the interface exposes a UART channel that can be used to log the instantaneous capacitor voltage state and current measurement channel.

Application module. The application module of ALTAIR platform is an externally attached sensor. We implement an air quality and pressure sensor board as a part of the platform.

3.4.2 Energy Supervisor Implementation

We implement an example energy supervisor to show how the architecture can be leveraged to optimize the duty-cycle of the connected application. With the dedicated hardware resources of the energy supervisor microcontroller, processing-intensive on-device energy optimization can be implemented without imposing significant resource conflict on the application microcontroller. One of the useful properties of the energy supervisor is its capability to learn to behave optimally post deployment without explicitly modeling the harvesting environment pre-deployment. To demonstrate this, we implement an on-device energy supervisor using reinforcement learning. Reinforcement learning has shown promising results as a power management technique since it can enable the sensor node to learn to adjust its duty cycle in a completely unknown environment [45], [74], [75]. The RL-based energy supervisor reacts to changes in available energy to update an application's operation, in this implementation, the rate of sending packets to report an event. The goal of the algorithm is to maximize the application sensing rate while

Algorithm 2 RL Algorithm for Energy Management

```
Initialize S, A, Q(s,a) = 0, α, γ, ϵ , δ
while true do
    for each episode do
        s ← Sample current states
        a ← Choose current action from s using ϵ-greedy policy
        wait for a duration of t_wait
        for each step of the episode do
            Perform action a for the duration of t_step
            wait for a duration of t_wait
            s′ ← Sample next states
            r ← reward (s′, a) according to equation (3)
            a′ ← Choose next action using ϵ-greedy policy
            Q(s,a) ← Q(s,a) + α ∗ [r + γ ∗ (Q(s′), a′) − Q(s,a)]
            ϵ ← ϵ − δ
            s ← s′
            a ← a′
```

avoiding critical energy depletion.

At a given time, the energy-harvesting node acts as an agent in different states ($s_t \in S$) corresponding to the available stored energy, incoming energy, and energy consumed by the load. The environment in this scenario consists of the stochastic harvestable energy source and the randomness inherent in the sensor hardware. The node interacts with the environment in time-slotted episodes by selecting a sensing rate ($a_t \in A$), and receives feedback in the form of reward ($R : S \times A \to R$). Through a series of such interactions with the environment, the agent finds its optimal policy ($\pi*$) to select future actions.

RL algorithm. We define the state space for the algorithm to capture the energy profile of the system. At a given time-step t_k of an episode, the energy supervisor collects all the following state information,

$$S = \{e_{st}(t_k), e_{in}(t_k), e_{load}(t_k)\} \tag{3.1}$$

where $e_{st}(t_k)$, $e_{in}(t_k)$, and $e_{load}(t_k)$ denotes the supercapacitor voltage at t_k, average input energy, and the load energy consumption during t_k. These parameters are indicative of the system's overall energy dynamics for which the supervisor finds an optimal action for the sensor. We consider a 24-hour long episode with a time step of 20 minutes.

The action space consists of a set of discrete sensing rate,

$$A = [r_{min}, ..., r_{max}] \tag{3.2}$$

where r_{min} and r_{max} are the minimum and maximum rate for the application. At each time step t_k, the agent selects an action $a(t_k) \in A$ according to the underlying policy. The goal of the reward function is to inspire the agent to choose the actions that maximize the sensing rate of the application and maintains minimum required energy on the energy storage. To model the reward function we adapt the reward function proposed by Aoudia, et al. [74] as follows:

$$R = (e_{st} - e_{min})/(e_{max} - e_{min}) * a(t_k) \tag{3.3}$$

We assign a negative reward of -400 if capacitor voltage falls below the minimum required voltage level of $2.0\,\mathrm{V}$. We choose this number so that the maximum cumulative reward over an episode does not exceed the negative reward. Algorithm 2 lists the pseudocode showing how we implement the SARSA reinforcement learning technique [76] to calculate the optimum duty cycle of an application.

Parameter setup. Though states and actions are continuous functions, we discretize those to restrict the size of Q-matrix. The discrete action space is A = [1,2,3,4,5] s, which denotes the time between two consecutive tasks. We set $\lambda = .99, \gamma = .8, \lambda_{min} = .1, \delta = .001, \alpha = .1$ after explicit testing. To enable faster convergence, we ensure that the learned Q-table is saved before a power failure happens by polling the capacitor voltage in the background.

3.5 Evaluation

To evaluate the ALTAIR design, we investigate the usability of the energy supervisor architecture and develop a set of different IoT applications. To demonstrate the versatility of the architecture, we run the applications using different energy supervisor algorithms and compare their performance. We tested the platforms across four categories of IoT hardware and evaluated how well these applications perform in terms of event generation frequency for periodic sensing and percentage of accurate detection for event-based

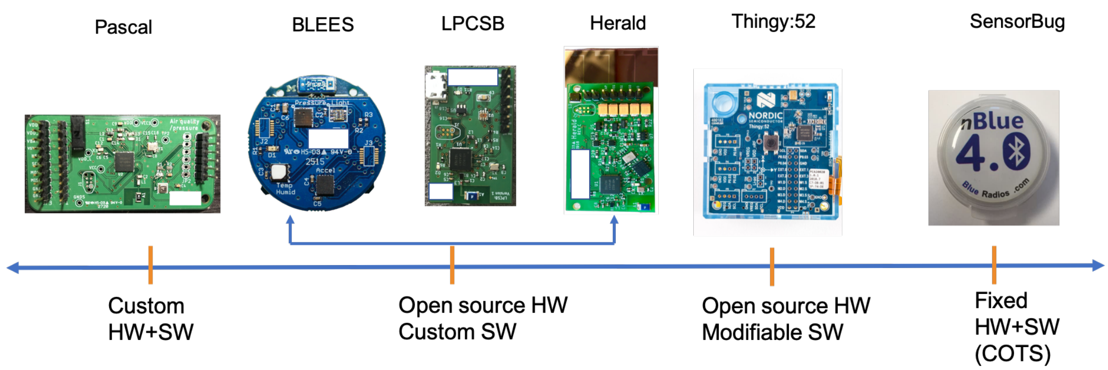


Figure 3.5: Spectrum of IoT sensors on a scale of hardware and software flexibility. The left-most category has maximum flexibility, whereas to the right-most has fixed hardware and software. We evaluate the ALTAIR platform with different points on this scale to demonstrate generality.

applications. We integrated six sensors with the ALTAIR hardware platform. We also explore the performance of the reinforcement learning based energy supervisor to understand how well the system adapts in terms of cumulative active time and reactivity—an inherent feature of the energy supervisor that shows the online adaptability of the system in post deployment situations.

3.5.1 Methodology

Categorizing existing IoT devices. ALTAIR uses its standard hardware and software interface to enable different applications. To test the usability of the ALTAIR power supply interface, we broadly categorize existing IoT devices into four groups based on the hardware and software interface exposed by the device: 1) sensors that are custom built specifically to use with ALTAIR platform ensuring ideal interfacing, 2) sensors with open source hardware and optimized applications, 3) sensors that have available hardware design with somewhat modifiable software stacks, 4) off-the-shelf sensors with non-modifiable hardware and software. This spectrum is shown in Figure 3.5. Of these four groups, the first group of sensors is best suited for use with ALTAIR. However, embedded software developers typically use the second and third categories of sensors.

We select six IoT sensors from these four categories to perform our experiments. These sensors are 1) a Pascal sensor board that monitors ambient air quality and pres-

Test platforms	Processor	Peak current (mA)	Default power supply	Available interface
Pascal	Cortex-M4 nRF52840	13.6	flexible	power supply, SPI
BLEES	Cortex-M0 nRF51822	15	Non-rechargable battery	power supply
Herald	Cortex-M0 nRF51822	14.8	Intermittently powered	power supply
LPCSB	Cortex-M0 nRF51822	14.6	USB-powered	power supply, I2C
Nordic Thingy:52	Cortex-M0 nRF52832	10	Rechargable battery	power supply, I2C, SPI, MOSFET drivers, IO
SensorBug	BR-LE4.0-S3A	17	Non-rechargable battery	power supply

Table 3.2: Specifications of test applications.

sure (category 1), 2) the BLEES platform [77] that senses temperature, humidity, light, pressure, and movement, (category 2), 3) the LPCSB [78], an ambient light sensor that categorizes natural light from sunlight, (category 2), 4) Herald, an intermittently-powered energy harvesting Bluetooth Low Energy (BLE) beacon [79] (category 2), 5) the Nordic Thingy:52 [80], a multi-sensor prototyping platform (category 3), and 6) the Sensor-Bug [81], a BLE beacon for smart home monitoring with temperature, light, and acceleration sensors (category 4). While BLEES, LPCSB, Herald hardware have limited hardware interfaces, the Pascal and Thingy platform includes a relatively richer interface with ports for communication including I2C and SPI. For the devices that do not have a data channel or open software that we can reprogram, we use the duty-cycled voltage terminal of the power supply interface to turn on/off the sensor according to the calculated duty-cycle. This exhibits the benefit of using the hardware version of the energy supervisor as discussed in Section 3.3.1.

The selected devices are designed to work on different powering options including rechargeable/non-rechargeable batteries, constant power, and intermittent source of energy. Also, these sensors use different application microcontrollers and their energy consumption varies. Section 3.5.1 lists the characteristics of these hardware platforms. ALTAIR's strength lies in its ability to take a battery-powered sensor and convert it to a self-powered energy-harvesting device. We envision that this will pave the way to many future battery-less applications.

Interfacing with ALTAIR. To interface with ALTAIR, we simply deactivate the default power supply of the sensor and jumper the power rails and SPI channel to the ALTAIR power supply. In the Thingy:52 board, we connect the voltage rails bypassing the battery monitoring circuitry. The application uses the energy API library at runtime to

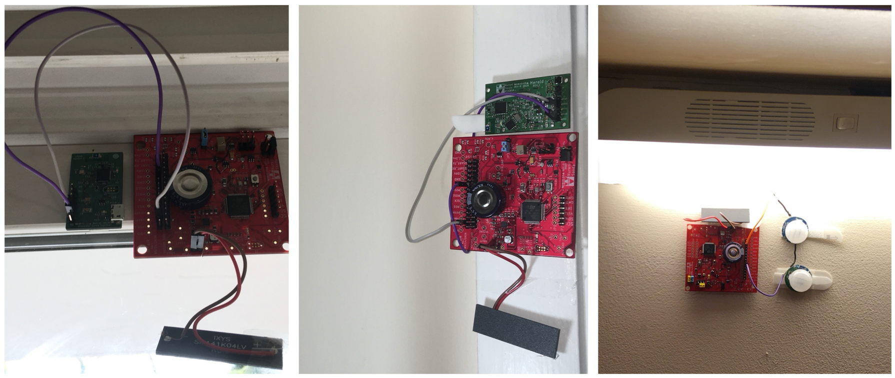


Figure 3.6: ALTAIR device deployments.

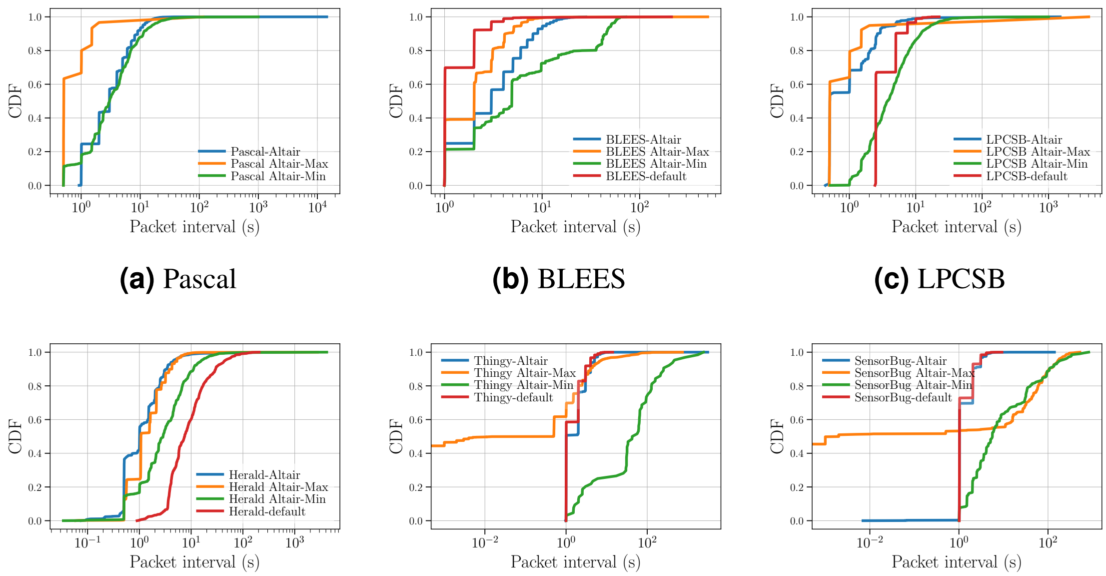
interface with the energy supervisor. The application developer implements the mapping between the API and their corresponding request id as an initial configuration for both the application and energy supervisor.

Sensing applications. We consider periodic and event-based sensing tasks from the above four categories to understand how well the adaptive power management algorithm captures useful events. The sensors use Bluetooth Low Energy (BLE) radio to report events. An always-on BLE receiver scans for advertisement packets and advertisements are sent with short intervals in between, in the range of milliseconds to a few seconds.

Deployment. Our deployment scenario consists of four different indoor locations in a building space that are exposed to variable light levels across different times of a day: on three walls, on a desk, a door, and a window. Figure 5.11 shows some of the deployed devices. A gateway device collects the BLE packets sent by the deployed sensors and logs them for post-processing. We train the energy supervisor reinforcement learning agent before beginning the data collection unless specified otherwise.

3.5.2 Energy Supervisor Performance

Event frequency. In this section, we compare the performance of the six test sensors in terms of the captured event frequency with respect to their default power source and different variants of the energy supervisor running on the power supply. The different variants of the energy supervisors are: Altair that runs the energy supervisor as discussed in Section 3.4.2, Altair-Min which always chooses the minimum duty cycle, hence max-



(a) Pascal **(b)** BLEES **(c)** LPCSB

(d) Herald **(e)** Thingy **(f)** SensorBug

Figure 3.7: Performance of different sensors when optimized by different variants of the energy supervisor and their default power supply. The ALTAIR energy supervisor implements reinforcement learning to choose between a set of transmission intervals. BLEES, LPCSB, and Thingy:52 sensors using ALTAIR produce a similar distribution of packet frequencies as the continuously powered version. For intermittently-powered Herald beacons however, ALTAIR produces denser packet distribution.

imum delay between packets (5s), and Altair-Max which chooses the minimum delay between packets (1s). We evaluate the cumulative distribution function (CDF) of the time between packets received by the receiver. Figure 3.7 compares the results. The time between two consecutive samples is a helpful parameter to understand overall how responsive the system is to an external event. The denser the samples, the more likely is the system to report critical events.

The sensor workload consists of taking a sample and reporting the data in BLE packet. When powered with the default supply, we program the BLEES, LPCSB, Thingy:52 sensor to send a BLE packet with the sensor data every second and SensorBug has a preprogrammed advertising interval of 1636 ms. For the herald beacon, however, the rate at which a packet is sent is proportional to its rate of harvesting energy. When connected to

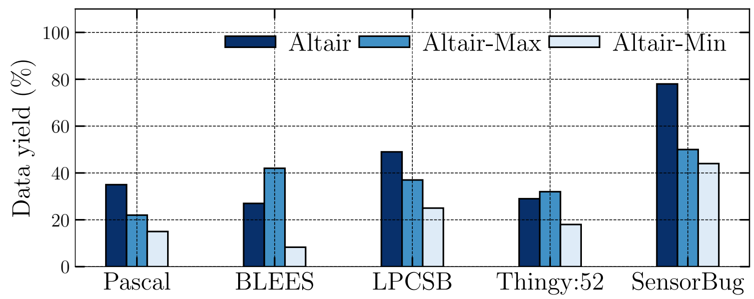


Figure 3.8: The percentage data yield of each sensor normalized to their default power supply. The ALTAIR energy supervisor produces better data yield than the Altair-max variant that always selects the high sampling rate.

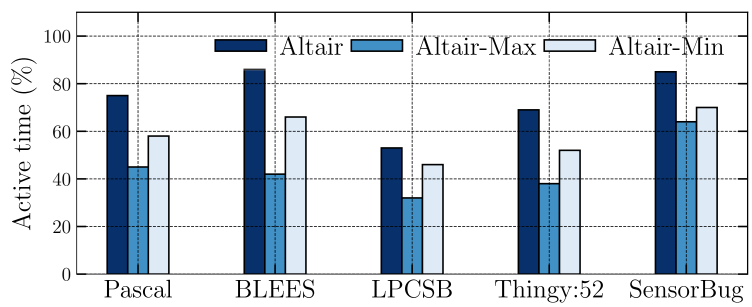


Figure 3.9: Percentage active time comparison across different energy supervisors. Active time denotes the percentage of time within an interval the sensor was continuously transmitting data. Altair outperforms the other variants.

the ALTAIR power supply, the sensors dynamically change the packet sent rate reacting to the changes in available energy.

We observe from the distributions of packet intervals in Figure 3.7 that for BLEES, LPCSB, and Thingy:52 sensors, the distribution curve of ALTAIR and the default power supply follow closely, and the 95th percentile of the inter-packet times are within ten seconds. The SensorBug, in contrast, achieves 111 s. The packet interval distribution of SensorBug with ALTAIR follows similar pattern as the default power, however, it undergoes longer occasional power outages due to its relatively high peak current (Section 3.5.1). ALTAIR achieves overall higher captured event frequency than Altair-Min, the intermittent power supply, but worse than Altair-Max. For the intermittently-powered herald

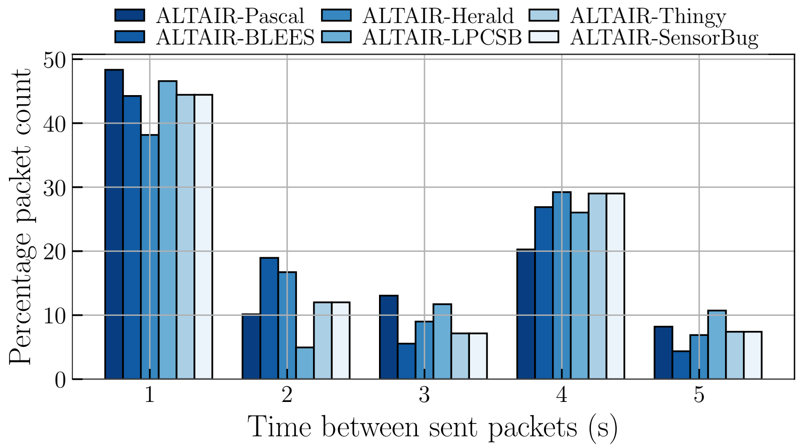


Figure 3.10: Packet distribution with ALTAIR. Sensors with ALTAIR opportunistically choose between five allowable rates, prioritizing the higher rate.

beacons, the time between consecutive samples is directly affected by the availability of harvestable energy and charge time of the storage capacitors resulting in larger delays. ALTAIR system however masks the irregularity of energy by storing it in a sufficiently sized capacitor and ensures samples are collected evenly at the desired rate. According to Figure 3.7, Herald achieves 10× higher captured event frequency with ALTAIR than with its intermittent power supply.

ALTAIR produces better percentage data yield and active time than both baselines as shown in Figure 3.8 and Figure 3.9, as ALTAIR optimizes for better sensing rate and fewer power failures. The percentage data yield signifies the amount of produced data normalized with respect to constant power sources and the percentage active time denotes the time in a fixed time interval for how long the sensor was active.

Figure 3.10 compares the distribution of inter-sample times of the sent packets. ALTAIR distributes the sample rate among the allowable rates reactively based on the decision of the energy supervisor. The RL agent chooses more and more actions that sample packets at a high rate when there is an energy surplus and relaxes the rate when the system is likely to see a power outage. The distribution shows that for all the sensors more than 35% of the total samples have a rate of one sample per second. Figure 3.11 shows the distribution for the default power supply. In the case of intermittently-powered systems,

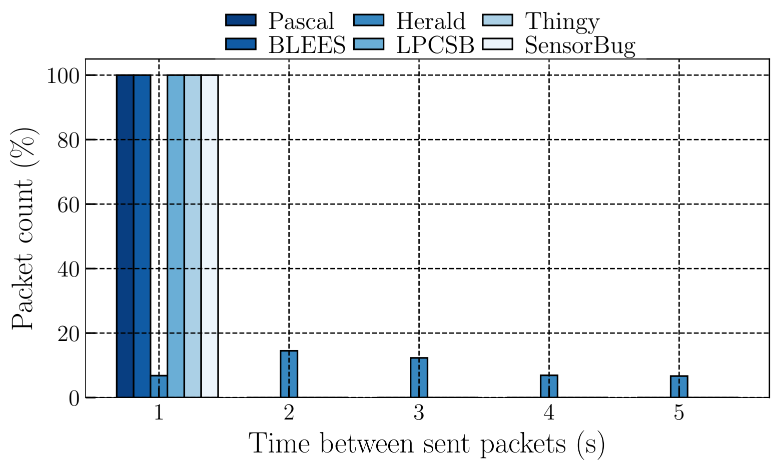


Figure 3.11: Packet distribution with the default power source.

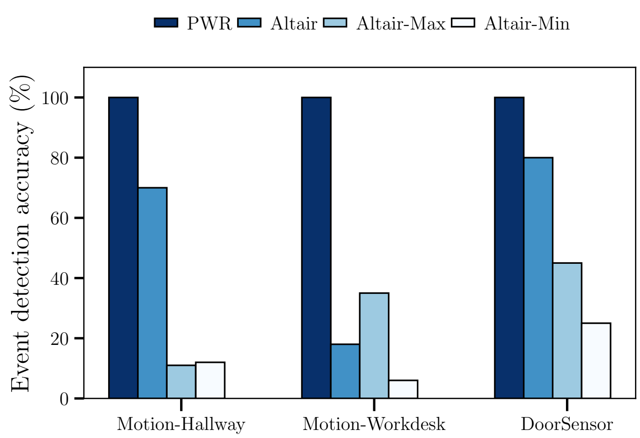


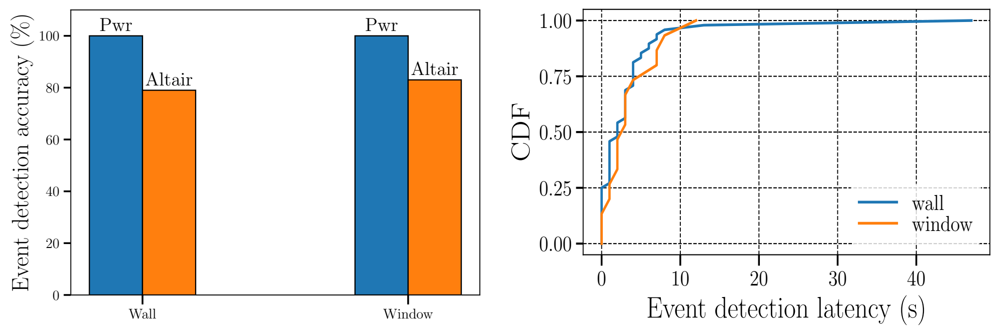
Figure 3.12: Event detection accuracy for time critical applications.

the samples are more sporadic and the sensor is spending majority of the time in charging the energy storage. Such systems are likely to miss events than ALTAIR that prioritizes higher sample rates when possible.

Event detection accuracy. To investigate how well applications can detect external events with ALTAIR, we classify event-based applications into two categories: time critical and non-time critical. For the time critical scenario, detecting an event should be instantaneous (i.e., less than a few seconds) since some external agent might need to react that event, for example, door sensors and motion-based light switch. For the non-time critical applications, detecting an event in a reasonable time interval is sufficient, for example, temperature sensors for HVAC systems. We deployed one BLEES sensor to detect door events, one to detect motion in two different locations and one Thingy:52 to detect temperature events.

We connected one BLEES board with the ALTAIR power supply and deployed it on a door to detect each time the door has been opened or closed, and two of them in a hallway and on a desk to detect movements for one week. With the default power supply, when the sensor gets an interrupt due to an event, BLEES wakes up to report the event. When connected with ALTAIR, the power supply processor fully controls the turn on/off the BLEES application processor. For the Thingy:52 board, the sensor is configured to go to the sleep mode and wake up when an event happens and report that event only if the capacitor has sufficient voltage. We chose to detect motions in two different locations to emulate two real-life scenarios: spaces that are usually lit most of the time of a day like a hallway, and spaces that have sporadic light exposure and sensing and harvesting is likely to happen simultaneously such as at a desk. To ensure we have enough data for statistical reasoning, we expedited the data collection process at the end by manually generating events as capturing organic events takes significant time. We used a constantly powered version of the sensors to collect the ground truth for events. We compared the performance of ALTAIR with two variants: ALTAIR-Min that always chooses minimum duty-cycle and ALTAIR-Max that selects maximum duty-cycle.

Figure 3.12 shows the percentage of correctly detected events and compares the result across three power management algorithms in three of the deployment scenarios. We find that sensors with ALTAIR achieves 70% and 80% detection accuracy in the hallway and on the door respectively, higher than the other two variants. This happens since ALTAIR spreads out the system active time by optimally choosing the duty-cycle and is likely to capture events correctly, whereas, ALTAIR-Max sees frequent power failure events and ALTAIR-Min misses events for spending much time in time between wake-ups. However, in the work-desk space ALTAIR-min detects more events than ALTAIR as it aggressively selects higher sampling rate. This signifies that careful decisions should



(a) Temperature event detection accuracy. **(b)** Non-time critical event detection latency CDF.

Figure 3.13: Event detection using ALTAIR.

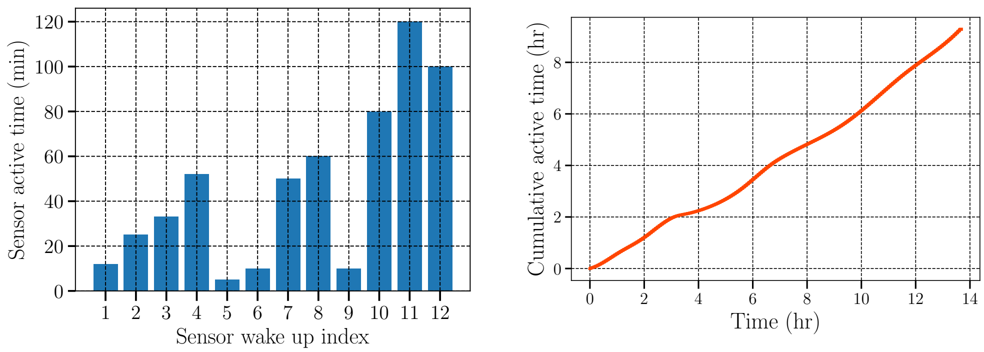
be made for applications where the event of interest can happen before the device can harvest enough energy. In such scenarios, predicting such events beforehand can improve detection accuracy. We plan to investigate such cases for future study.

As a candidate of non-time critical event detection, we deployed one Thingy:52 to monitor the temperature of a home in two different locations: on a window and on an indoor wall. We analyze how many times the sensor can correctly report when the temperature falls below 76°F or exceeds 79°F (selected according to the comfort level of the occupants). Figure 3.13(a) shows that with ALTAIR the device reports 79% and 83% of the events accurately. To determine the latency between the event has occurred and successfully reported, we show the CDF of detection latency in Figure 3.13(b). We find that the 95th percentile latency remains within 12 s.

3.5.3 RL Supervisor Robustness

System active time. ALTAIR uses a $470\,\mathrm{mF}$ supercapacitor as an energy-reservoir of the system. The larger the size of the capacitor, the more time it takes to recharge after a power failure. In this section, we aim to analyze the active time of a sensor connected to ALTAIR. We define the duration of the time a sensor samples continuously before exhausting its energy buffer as the active time.

To evaluate how much time the system spends in recharging the capacitor in a dynamic



(a) **(b)**

Figure 3.14: When moved to a new environment, the system increases its activity as it learns the new harvesting conditions.

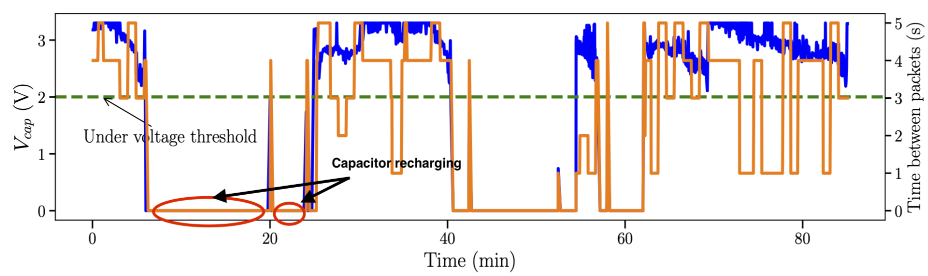
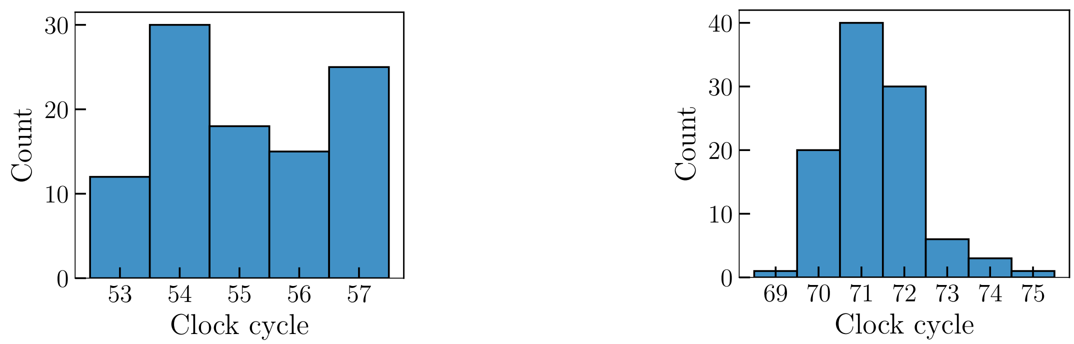


Figure 3.15: With time, the energy supervisor learns to avoid power failure by adjusting the time between samples, though experiences a few power failures at the beginning. The blue trace plots the instantaneous capacitor voltage, and the orange corresponds to the to the time between packets.

energy environment, we moved the Thingy:52 sensor from its original window position to a wall. Figure 3.14(a) shows the active time of the sensor during each progressive power cycle. After being exposed to a new environment with a different harvesting scenario, at first the system explores to find the optimal set of actions that avoids power failure. The system active time progressively increases as it sees less power failures with occasional dips. Figure 3.14(b) shows the cumulative active time of the sensor.

Reactivity. In this section, we analyze how the energy supervisor reactively changes the rate responding to the available energy. A sensor that runs at a constant duty-cycle



(a) Wake-up delay **(b)** Interrupt service delay

Figure 3.16: The histogram of the delay in servicing the message request by the energy supervisor in clock cycles.

suffers from multiple consequences: 1) in case of an energy surplus, the system underperforms by not sampling more, and 2) in case of an energy drought, the system runs the risk of frequent power failures by not backing off. Figure 5.19 shows how ALTAIR adjusts the time between samples reacting to the capacitor voltage. We set the episode interval as 2 min for this experiment. In the beginning, the system experiences frequent power failures around 8, 15, 22 and 25 minutes, spends significant time in power failure, but learns to adjust the time between samples allowing the sensor to sleep. A falling capacitor voltage results in an increase in the time between samples and a steady or rising capacitor voltage encourages frequent samples. Throughout this experiment, the harvester was kept under a stable harvesting environment which ensures that the capacitor voltage was only the system variable. By vary its rate of operation, the system incurs 50% fewer power failures with an increased availability of 44%.

3.5.4 Energy Supervisor Responsiveness

As the energy supervisor processor receives the energy API request from the main processor through a hardware GPIO interrupt, we investigate the number of clock cycles needed to serve the interrupt. We characterize the delay to wake up the energy supervisor from a low power sleep and the delay to respond to an interrupt while performing its routine task. We show the histogram of delays of 100 interrupts in Figure 3.16(a) and Figure 3.16(b), respectively. Though the delay in terms of clock cycle varies, the distribution shows the

Table 3.3: Power draw overhead of ALTAIR.

Component	Active current	Sleep current
MCU STM32L010R8	585 µA@16Mhz	4.7 µA
Charger SPV1050	2.6 µA	1 nA
Current Sensor Max9634	1 µA	1 nA
Power Gating TPL5110	35 nA	N/A
All components	7.8 mA	94.5 µA

delay can be bounded within a few clock cycles.

3.5.5 Energy Overhead

Using the ALTAIR platform does come with an energy overhead. However, while implementing the platform, we chose components with low power options. Table 3.3 lists the active and sleep current of the used components. The average active power draw of the board is 7.8 mA and the quiescent power draw is 94.5 µA. We notice that the significant energy overhead comes from the ADC polling to observe the system energy as ADC reading over one second costs 24.3 µJ. This overhead can be reduced by polling the ADC less frequently.

3.6 Discussion

Partial decoupling. Though ALTAIR reduces the logical dependency between energy management and application tasks, both subsystems are required to have a knowledge of the expected information from each other. Since IoT sensors are typically small systems with a handful of running applications, we expect the ALTAIR architecture is sufficient. However, for large scale embedded systems full decoupling may be needed.

Vast heterogeneity of IoT applications. Though we believe that ALTAIR is a stepping stone in the direction of a "general-purpose" energy-harvesting power system suited for IoT sensing applications, the spectrum of sensing is broad in terms of energy cost and time-sensitiveness. Applications that are susceptible to occasional power failures might require back-up source of energy such as rechargeable batteries [33]. In such a case, the RL manager might reduce the negative reward, if the backup energy source is available.

Energy storage size. Though an over-provisioned energy reservoir can mask unstable available energy and eliminate the need for complex software support, bigger capacitors suffer from higher leakage, prolonged cold-start phase, and longer recharge times.

Limited harvester support. Current ALTAIR platform only has support for harvesting energy using solar and TEG harvesters.

Enabling new techniques. We believe that faster testing and development plays an important factor when designing novel energy-harvesting applications and ALTAIR attempts to lower the barrier to entry. We recognize that there is a lack of prototyping platform for energy-harvesting application and this work will attract researchers to build and test new software and hardware techniques for better power management.

3.7 Conclusion

Managing energy is critical for energy-harvesting systems, and this burden has been foisted on the IoT application software with only limited support from the energy-related hardware. This chapter argues that ad-hoc and implementation-specific interfaces between applications and power supplies constrain the development of energy-harvesting devices, and that a new MCU-power supply interface is critical for restoring proper layering to these systems. In this paper, we introduce such a system that isolates the energy management decisions from a sensor's workload, and provides a simple interface for adding new applications to the system. By strictly separating energy-management from device operation, we believe we can lower the bar for developing energy-harvesting systems, helping to realize a fully batteryless IoT.

Chapter 4

PreFarad: Event Detection with Intermittent Sensors

The ability to harvest energy from environment empowers small computing devices to be pervasively deployed in most indoor and remote outdoor spaces for long-term sensing. However, the amount of harvestable energy can be critically low and volatile, where the sensor is deployed, which limits the availability of the sensors [10], [12], [69], [82].

Intermittent energy-harvesting devices store energy in a sufficiently-sized capacitor. When the capacitor is charged to the operating voltage threshold, the microcontroller turns on and performs sense-process-transmit tasks. However, because the average input power is typically significantly less than the average power draw of the device, the capacitor is depleted within a few milliseconds. When the the capacitor is depleted to a critical threshold, the MCU and peripherals completely turn off and the capacitor recharges and repeats the power cycle (Section 4.1). When the harvestable energy is plenty, the device can charge quickly and when energy is low, the device spends most of the time recharging. Our experiments with indoor light energy-harvesting BLE beacons find that the recharge time can range from a few hundreds of milliseconds to tens of minutes for a sensor [15]. The intermittent operation of energy-harvesting devices make stochastic event detection immensely challenging. An intermittent sensor has a higher rate of missed events than a battery-powered sensor that is always available.

To enable reliable event detection, in this chapter we present PreFarad, an intermittent battery-less event detection system, that partitions the energy required to sense an event of interest from the system's cumulative energy budget. PreFarad dedicates a decoupled energy buffer for the sensor peripheral responsible for event monitoring and a separate energy buffer for the rest of components to handle event data (i.e., process and transmit). We utilize the insight that the average energy requirement to detect an event is several times lower than transmitting a radio packet to report the information to a base station.

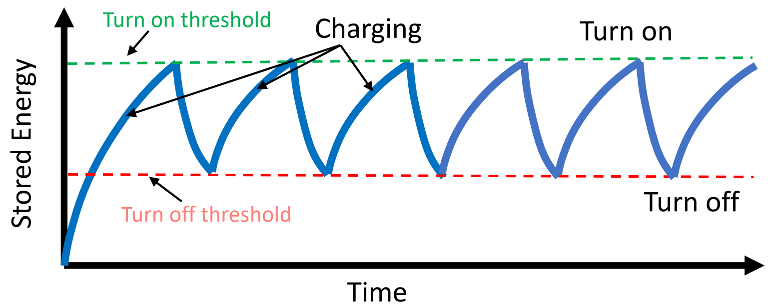


Figure 4.1: Intermittent energy-harvesting devices harvest and store energy in a capacitor to sustain their operation. Once the buffered energy reaches the turn on threshold, the device activates and performs operation. Since the average power of the harvester is significantly lower the device power draw, the buffer depletes quickly, effectively turning off the device and allowing the buffer to recharge.

This enables the event capacitor to be considerably small enough to quickly recharge and sample an event, significantly improving device responsiveness and event detection rate. PreFarad prioritizes charging the event capacitor and as more energy becomes available, a larger buffer activates the MCU to process and report the event. While the transmission capacitor charges up, PreFarad must cache any event that happens during this interval.

4.1 Event Detection with Intermittent Energy

To report an event, an IoT device must instantaneously react to an event before it finishes. To successfully report an event, the capacitor of an intermittently-powered device must have the minimum energy to turn on the device at the time the event occurs. If that condition is not met, the event is missed and the reliability and sensor data quality of the system is compromised. This happens when the capacitor is recharging after a discharge from the previous power cycle or from a fully depleted stage after a long energy drought. Low energy-harvesting conditions worsens the reliability of these sensors.

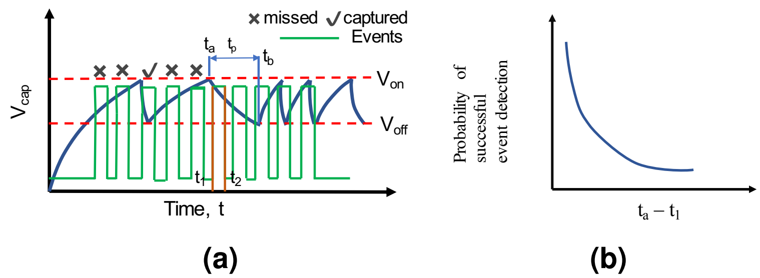


Figure 4.2: Uncertainty in successfully detecting an event in intermittently-powered systems. Intermittently-powered devices turn on once its capacitor reaches a minimum threshold and performs a routine task. Events that happen during recharging is missed compromising reliability of service. (a) depicts 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 life-cycle. If energy availability and the event of interest does not coincide, the likelihood of detecting the event decreases as shown in (b).

4.1.1 Unpredictable Energy Demand

Many IoT sensors are event-based, meaning, sensors respond to a particular change in the environment and notify the user off the trigger or turn on an actuator. For example, movement triggers from a PIR sensor control the lights of a room, contact status from a magnetic sensor to report door open/close events , or an accelerometer attached to a garage door opener. Reliable sensors must detect each of these events and report the data instantaneously. Battery-powered and wire-powered devices perform well for these applications, as they have steady power supply and are always available to react. However, intermittent sensors do not have continuous availability as they frequently recharge their storage, which makes them highly unreliable (Figure 4.2). For light energy at indoors, low illuminance level and occasional unavailability in light in controlled spaces exacerbates the challenge. Sensors may spend a few seconds to several minutes in between power cycles. Many IoT events are hard-to-predict, fast, and time-sensitive that are particularly challenging to detect under intermittency. The sensor must accumulate sufficient energy to turn on before the event finishes. We particularly focus on these types of event in this paper.

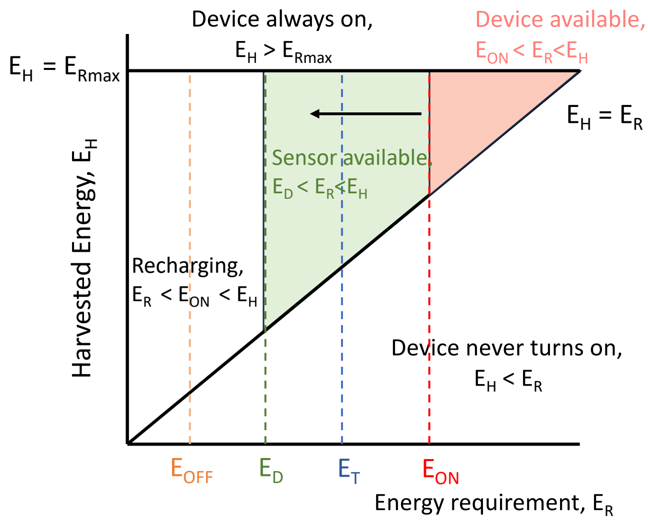


Figure 4.3: Traditional energy-harvesting based sensor devices need to reach an energy level of E_{on} to able to sense because of the tight coupling between the sensor and MCU+Radio energy distribution, as such the energy barrier is too high (denoted by red dotted lines). However, if the sensing element had its energy requirement decoupled from the MCU+Radio unit, the energy barrier for just sensing goes down (denoted by the green dotted lines).

4.1.2 Task Energy Requirement

Current intermittent sensors adopt a single capacitor to drive all the components (i.e., MCU, peripherals, radio) and execute the software application. The capacitor is selected to be large enough to buffer sufficient energy to complete a single chain of sense-compute-transmit during each active period. However, sensing external event (for example, change in the surrounding magnetic field for a contact sensor), perform computation from the sensor data if any, and transmitting the data over a radio packet (BLE, Wifi, LoRa) have different energy requirements. Typically radio transmissions are more energy-expensive than collecting sensor samples. For example, a EKMC PIR motion sensors consume a peak current of $100\,\mu A$ during detection and a hall-effect magnetic contact sensor draw a $3.2\,mA$ peak current while conversion [83], [84], which is less than half of the peak transmission current consumption of $9.2\,mA$ by ultra-low power BLE SoC nRF523832.

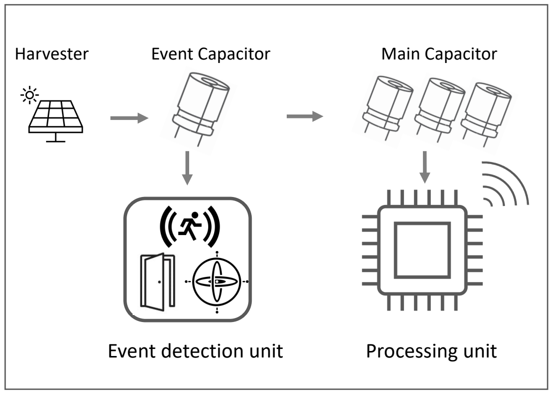


Figure 4.4: Real-time event detection on intermittent batteryless sensors is challenging due to their unique nature of operation. PreFarad proposes an architecture that prioritizes and separates the sensing peripheral dedicated for event detection from the rest of the component by allowing it to power from a smaller energy buffer. This decoupling in the energy buffer from the main capacitor enables the event subsystem to be more available and accurate.

4.1.3 Capacitor Transient Response

A capacitor's recharge time is proportional to input power of the energy-harvester and is inversely proportional to the capacitance of the capacitor. When powered by two solar cells with identical dimensions and electrical rating, a smaller capacitor will charge faster than a larger one. Moreover, capacitor's charge retention time is related to the leakage current (DCL) and larger capacitor has higher DCL.

4.2 System Design

Based on the observations and limitations mentioned in the last section, we propose PreFarad, a sensor architecture that improves the reliability of event-based intermittent devices.

4.2.1 Overview

Instead of powering the device load from a common capacitor, PreFarad decouples the energy bucket of the event detection unit from the total energy reserve. Figure shows the overview of the proposed system. A small capacitor (C_{Evt}) powers the sensor, its peripherals, and a low power memory unit that stores and holds the event. The rest of the component including the MCU, radio, and other peripherals is powered from a separate energy bucket (C_{Main}), which is larger and therefore, slower to respond. The memory component of the detection unit saves the event until the main capacitor finishes recharging and activates the MCU.

4.2.2 Design Principles

Section 4.1.2 draws a conceptual relation between the harvested energy (E_H) and energy demand(E_R) of a typical IoT device. In the region where, $E_{ON} < E_R < E_H$ device is on and "available". Here, E_{ON} is the total energy requirement in a power cycle. Typically, $E_{ON} = E_D + E_P + E_T + E_{OFF}$, where E_D is the energy required to detect an external event of interest, E_P is for handling the event, E_T represents the energy required to transmit a radio packet, and E_{OFF} corresponds to quiescent power draw. In $E_R < E_{ON} < E_H$, the device is inactive and recharging. Instead of waiting to accumulate E_{ON} on the capacitor, if the sensing unit can turn on at E_D, we would be able to move the available region to further left, improving the responsiveness of the system. The lower the E_D, the more responsive and accurate the system becomes to detect an event, as C_{Evt} spends less time recharging. As E_D is always smaller than E_{ON}, this decoupling is useful. By isolating detection from the core processing component, PreFarad utilizes more energy and is more likely to respond and detect an event. PreFarad is more effective to detect events that occur almost randomly, hard to predict and last less than a fraction of seconds to a few seconds. Additionally, isolating the sensing unit also allows C_{Main} to activate processing unit faster than a common capacitor system.

4.2.3 System Architecture

PreFarad architecture consists of three major units: the energy-harvesting unit, the detection unit, and the processing unit. Figure 4.5 denotes the block diagram of the system.

Energy-harvesting Unit. PreFarad converts light energy to electrical energy using a photovoltaic harvester and stores it in capacitors. To prevent the capacitors to be charged beyond their maximum rated voltage, a zener diode protects from overvoltage in high

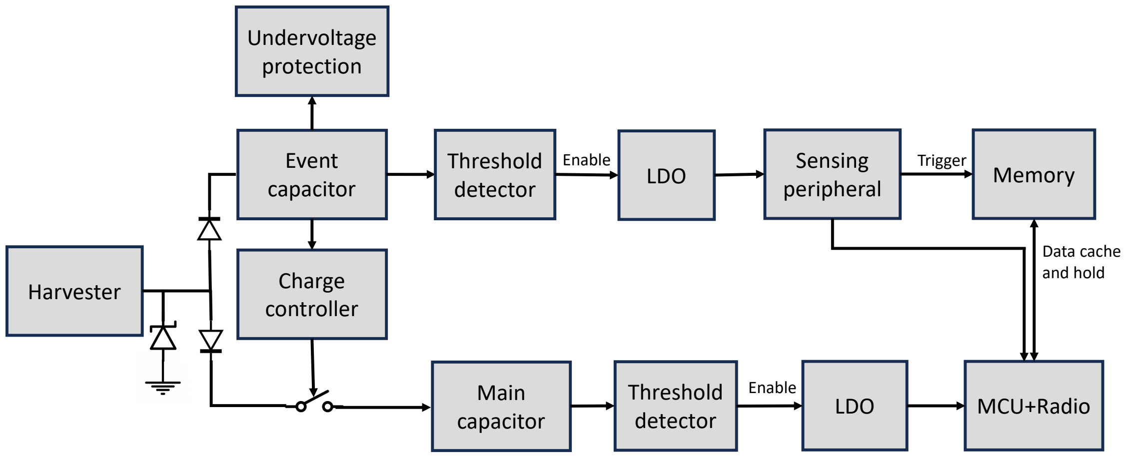
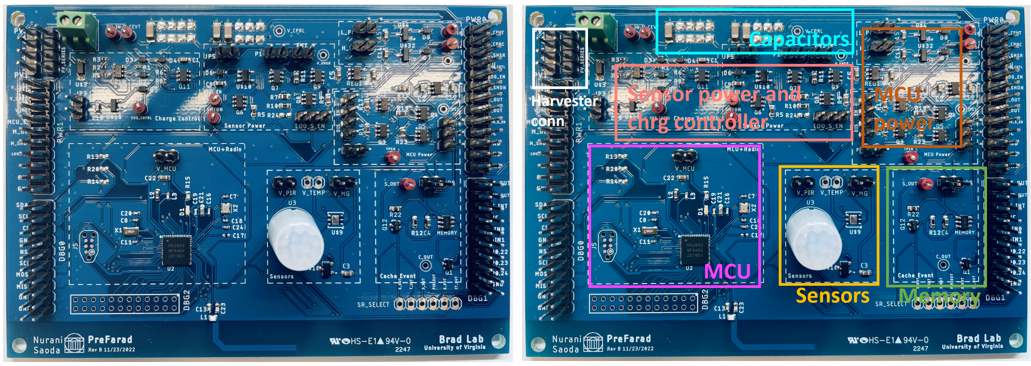


Figure 4.5: System design of PreFarad.

energy scenarios. The energy-harvester charges the event capacitor first before power flows through rest of the circuit.

The Detection Unit. The detection unit is powered from the event capacitor (C_{Evt}). The major component of the detection units are the sensor and a memory circuit. Once the capacitor charges to a certain level, it provides power to the sensor that detects an event. The sensor stays activated until the capacitor discharges below a voltage level. Once the discharge voltage level is reached, the sensor is turned off allowing the event capacitor to recharge. While activated, the event sensor monitors external environment and sets a digital output when an event occurs. Since the event might finish before the main capacitor can recharge, the detection unit must cache any event the sensor detects. Each time the output of the sensor toggles high, memory circuit sets its cache output. The cache output is held for a certain period of time before it resets. The events in between the cache sets and resets are ignored. The hold time of the memory allows the main capacitor some time to recharge sufficiently to activate the MCU. PreFarad prioritizes charging the event capacitor over the main capacitor. Instead of following a sequence of sense-process-transmit tasks, the sensing procedure is pre-accomplished before the MCU turns on. The detection unit also incorporates a charge controller that activates the main capacitor.

The Processing Unit. The processing unit of PreFarad consists of a microcontroller, radio, and other optional peripherals. It is powered by a larger main capacitor (C_{Main}) sufficient enough to read the cached data, perform data processing if required, and transmit



(a) (b)

Figure 4.6: 4.5cm x 3.1cm PreFarad custom hardware platform.

the event to notify the user or activate an actuator. When the event capacitor is sufficiently charged, the charge controller connects the main capacitor to the harvester. The main capacitor then turns on the rest of the systems. Upon turning on, the MCU samples the data cache line and the sensor output transmits the data periodically until the main capacitor exhausts and turns off the MCU. The MCU also resets the memory after handling the data. The main capacitor is provisioned at design-time to support at least one radio transmission each power cycle.

4.3 Implementation

We implement PreFarad in a custom PCB as shown in Figure 4.6. The dimension of the board is 4.5cm x 3.1cm.

The platform uses an AM1522 amorphous silicon indoor photovoltaic cell to harvest energy [16]. The PV cell has a rated open circuit voltage, short circuit current is 3.1 V and $62.2\,\mu A$ with a maximum power point voltage of 2.6 V. We use a 6.2 V BZT52 zener diode as an overvoltage protection [85]. The energy is stored in MLCC capacitors. We use a capacitance of $100\,\mu F$ for the event capacitor and $400\,\mu F$ main capacitor. A 2.1V S-1009 series super-low current voltage threshold detector enables the sensor [86]. A 2.3V MAX809 voltage supervisor along a MOSFET switch constitutes the charge controller that starts charging the main capacitor once the event capacitor reaches 2.3V. Two nano-power ultra-low quiescent current low-dropout LDO TPS7A03 are used to regulate the capacitor voltages separately. We use the TPL5110 nano-power timer reset IC in manual MOSFET mode to hold the sensor output for 8 s.

The platform has an ultra-low power nRF52832 SoC with BLE radio as a compute core [87]. Currently the platform has two event-based sensors: a passive infrared (PIR) human motion sensor with 5m range [83] and a hall-effect magnetic sensor [84].

4.4 Evaluation

To evaluate PreFarad, our goal is to determine how successful the system is to detect events of interest in event-triggered IoT applications. We design several event-based applications, deploy the sensors in real world indoor spaces, and analyze their performance. We answer the following questions to evaluate the accuracy and robustness of the proposed approach:

- What is the percentage of event detection accuracy of achieved by PreFarad? How does the performance compare with the other approach?
- How does the detection accuracy vary within different sensing applications?
- How is the availability of PreFarad sensors compared to other approaches?
- What is the latency of reporting an event?

4.4.1 Experimental Setup

Applications. We design two indoor event-based sensing applications that represent unpredictable and momentary events. One application is proximity sensing using magnetic hall-effect sensors to monitor the status of doors, shelves, or cabinets to determine whether they are closed or opened. These type of contact sensors are normally closed indicating an usual state of the attached object. When the distance between the magnet and the sensor increases due to opening a door, the sensor switches its output from low to high and the trigger is cached by PreFarad. The opening of a door is typically random and lasts only for a few seconds before the sensor restores. PreFarad door sensors detect when the door is opened and transmits the alert to a gateway. Besides contact sensing, we design an occupancy using PIR sensors that detects human movement. When no movement is detected within its range, the sensor output remains low. Upon detecting a movement, the sensor triggers to high and the event is cached for PreFarad's processing unit to turn on and report the movement. Human presence and movement in a room is usually random and the duration of movements vary from very short lasting only a few seconds to hundreds of seconds depending on the number of persons present in a space. If the movement is not correctly and timely detected, the actuators activated by the movement will fail to respond (for example, automated lighting, smart trash cans). To configure a board as a door sensor, we only activate the respective sensor and connect the sensor output to the memory unit by connecting and disconnecting jumper connections. Sensors transmit BLE advertisements periodically to report the sensor readings.

Baseline and Ground truth. To collect the ground truths of the events regarding when the event occurred and how long the event last, we use the same sensors powered from batteries. We remove the energy-harvesting frontend and capacitors from PreFarad board, and connect two AA batteries to supply a 3.0V to power all the components of the board. The battery-powered boards have the same MCU and sensor peripherals as the PreFarad ones, samples the sensor reading periodically , and transmits the reading BLE advertisement every second. We also implement a version of the board where all components are powered from a common capacitor to represent a traditional intermittent sensor. We empirically provision the capacitor with a capacitance of $600\,\mu\mathrm{F}$ to support all the computation. This version of the board do not implement the memory unit of the PreFarad. The MCU turns on when the capacitor voltage reaches 2.6V, samples the sensor output, transmits the data as BLE advertisements.

Data Collection. We collect the data transmitted by PreFarad sensors, common capacitor board, and the battery-powered sensors using an always-on BLE receiver [88]. We timestamp each received packet at the receiver and analyze the packets to compare the performance of different systems. We deploy all the door sensors on our lab door and install the occupancy sensors a lab desk.

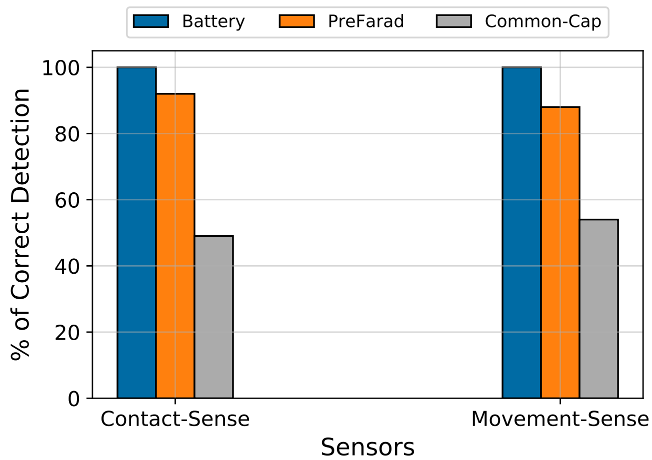


Figure 4.7: PreFarad outperforms the common capacitor approach and achieves a detection accuracy of 92% and 88% for two different event-based sensors.

4.4.2 Event Detection Accuracy

In this section, we evaluate and compare the percentage of correct event detection of each systems. Here, an event denotes a deviation from the default state of the environment: for the door sensors, when a door is opened and for the occupancy sensors, when a movement occurs. An event is correctly detected if the sensor transmits a packet notifying about the particular event. For PreFarad sensors, they will fail to report an event if it starts and finishes before the event capacitor recharges or the event finishes and the data cache times out before the main capacitor can charge up. The common capacitor sensors will fail to detect an event if the event finishes before the capacitor can charge up. Since the battery-powered sensors have continuous power, they report majority of the event unless some packets are dropped or missed due to wireless channel interference or receiver error. We deploy three versions of door sensors on a door and manually generate 90 door open events by entering and exiting through the door. All sensors are triggered by the same door movement. Figure 4.8(b) plots the histogram of duration of time the door was open and Figure 4.8(a) plots the histogram showing the interval between each open events. As shown is Figure 4.7, PreFarad achieves an accuracy of 92%, while the common-cap sensor have 49% accuracy. Since PreFarad caches the event for eight seconds after the event finishes, it detects event that finishes before an activation. According to Figure 4.8(b), the

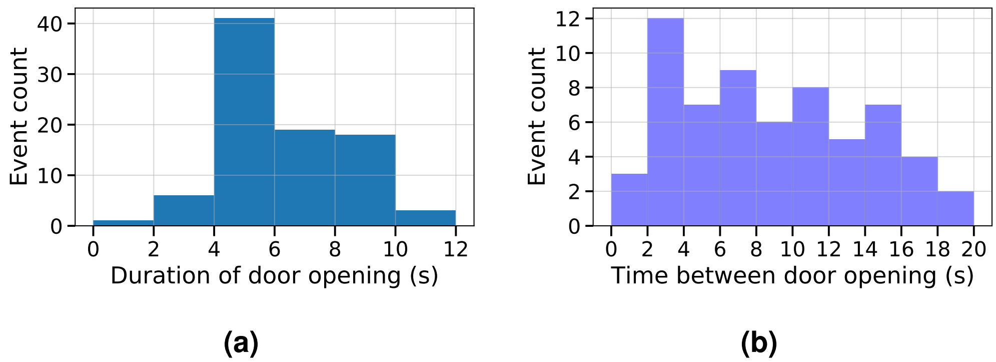


Figure 4.8: This figure plots the timing statistics of door sensors deployed to detect real door opening events. a) shows the distribution of the number of events corresponding to the duration the door was open while someone entered or exited the room. b) plots the histogram of time in between the door events.

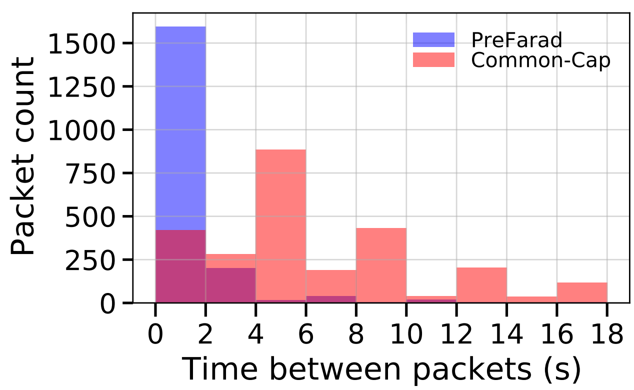


Figure 4.9: The figure plots the distribution of packet counts in terms of interval between two packets. PreFarad generates more frequent transmissions than the common-cap.

duration of door opening can be as short as 2-6 seconds, whereas the common-cap system experiences recharge durations more than 6 seconds. Figure 4.9 shows the distribution of number of packets against the time between two consecutive packets. To detect occupancy, we record 133 movements and plot the accuracy in Figure 4.7. PreFarad detects 88% of the total events and common-cap detects only 54%.

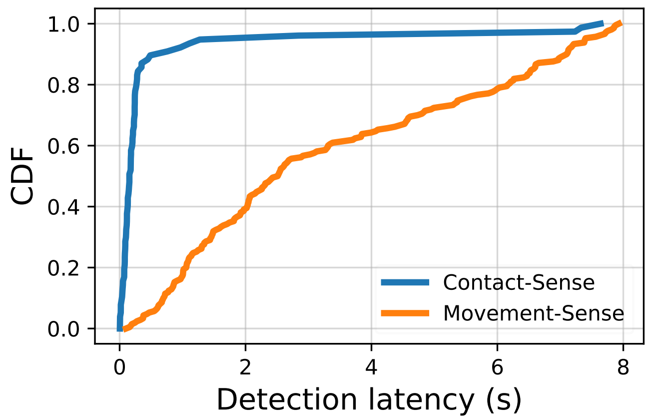


Figure 4.10: We plot the cumulative distribution function of the time between an event occurred and it was reported by PreFarad sensors.

4.4.3 Activation Frequency

Figure 4.9 shows the distribution of number of packets versus the interval between two consecutive packets. We see that PreFarad generates denser packets compared to common-cap, making it more available. This happens because energy is stored in two buckets, allowing the main capacitor to be smaller and experience quicker activations.

4.4.4 Detection Latency

To evaluate the latency between the occurrence of an event and when it was reported, we plot the cumulative distribution function (CDF) of the detection latency in Figure 4.10. We find that the 95th percentile of detection latency for the contact sensor and the occupancy sensor are 1.2 s and 7.4 s respectively.

4.5 Discussions and Limitations

In this section, we identify several limitations of the proposed limitation has room for improvement in the future work.

Event occurence time. In our current prototype, the data cache and hold line holds an event trigger signal for eight seconds. When the application turns on and samples the cache line, it has no way of determining exactly when last event was triggered. The application is only aware that the event happened within the last eight seconds. For simple sense and send application, this is sufficient. However, if the event was detected after an unacceptable amount of delay, it may be useful for the sensor to skip the transmission and save energy for re-trial.

Back-to-back events. If two consecutive events happen before the MCU activates, the platform currently can not distinguish between the events. We plan to incorporate this functionality in future revisions.

Data caching. Currently the memory unit is only able to hold the sensor output as long as the event capacitor has sufficient energy. The unit is volatile, meaning, if the power goes off the output of the memory also resets. In our experience, this happens very occasionally. We are currently working on a version which will store the data even is the event capacitor is discharged. Additionally, if the MCU spends more than eight seconds recharging due to the energy being very scarce, the sensor will fail to detect the event. Since caching the data for a long period of time also comes with energy overhead and may completely deplete the event capacitor, we considered the trade-off between energy overhead and detection accuracy.

4.6 Conclusion

In this chapter, we present PreFarad to improve the reliability of intermittent batteryless sensors in event-based IoT applications. PreFarad divides the energy storage of the sensor in two dedicated buffer and exploit the responsiveness of a smaller capacitor to capture more events. The system outperforms common storage intermittent sensors by a large margin and demonstrates that energy-harvesting sensors have the promise to be adopted for real world event detection.

Chapter 5

RetroIoT: Retrofitting IoT Deployments

Commercial Internet of Things (IoT) deployments are mostly closed-source systems that offer little to no flexibility to modify the hardware and software of the end devices. Once deployed, retrofitting such systems to an upgraded functionality requires replacing all the devices, which can be extremely time and cost prohibitive. For example, consider a scenario, where a user has installed a video doorbell camera in their home that streams video from the front gate to their app when a movement detected. The whole installation process requires buying and installing the sensor, a device-specific gateway from the vendor, and paying for a cloud subscription for storage and better data analytics. Later, if the user wants to add an additional trigger besides movement to their video doorbell, for example, responding to a loud noise at the frontyard, they would have to replace the current device and in the worse case, completely switch to a different manufacturer. End users cannot generally leverage deployed infrastructure to add their own sensors or custom data. This severely limits the longevity of devices as the functionality of the devices become obsolete.

Our key observation is that many IoT devices report their battery voltage in addition to their sensor data. This enables notifying the user when the battery must be replaced, but most of the battery reports are effectively unused. We claim this channel can be used to encode completely new information beyond the device's initial intended application. We propose RetroIoT, an approach that replaces a standard battery with a "programmable" battery that can control its own voltage output and encode additional information into the battery voltage level. Later, the voltage readings can be retrieved from the cloud and decoded, and a new data channel is introduced without modifying the existing IoT devices beyond just replacing the batteries. As we show, the battery voltage channel of IoT devices can be repurposed using oblivious devices, prototype new capabilities, as

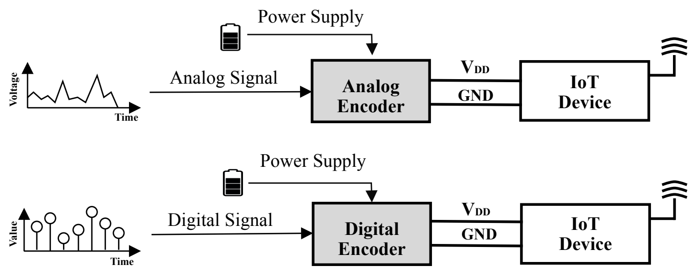


Figure 5.1: Overview of RETROIOT. Many IoT devices sample and report their battery voltage, and by simply swapping the battery these devices can be repurposed to encode additional useful information. This retrofitting gives users new control to capture new data, upgrade to energy-harvesting, or strategically deactivate sensitive sensors.

well as upgraded to adopt energy-harvesting with proper duty-cycling. This technique can enhance existing devices to improve sustainability and privacy without waiting on manufacturers to produce battery-free or privacy-first IoT devices.

5.1 RETROIOT System Overview

We introduce RETROIOT, an approach to modulate additional information using the battery terminals of an IoT device. RETROIOT replaces a conventional battery with a programmable voltage controller and augments the IoT device with desired functionality. It interoperates with many existing systems that already have functional hardware, software, and network infrastructure. Figure 6.4 shows the high-level overview of RETROIOT. Any analog and digital input data is mapped into the acceptable input voltage range of the attached IoT device. The main block of RETROIOT is the signal encoder that implements the modulation of symbols on the battery voltage channel from raw signal values. The encoder output includes the encoded voltage as well as provides power to the IoT device as the conventional battery would.

RETROIOT enables users, hobbyists, and IoT developers to take advantage of the existing infrastructure of closed source commercial devices, without requiring them to build the whole stack from scratch. This way, RETROIOT promotes re-usability and faster system development, and benefits deployments that require efficient and low-impact upgrades. It demonstrates a new design point for modifying existing IoT systems. RETROIOT is not a universal replacement for IoT redesign, but represents an option for applications where the value of a new data channel with existing devices is high and the limitations of increased power draw, power supply design effort, and limited data rate are acceptable. This is particularly true in cases where the alternative would require hardware and software updates in the IoT's gateway and server infrastructure. We elaborate on this further in Section 5.7.

5.2 Design Challenges

5.2.1 Minimal Modifications

To make retrofitting legacy IoT systems viable, integrating new capabilities into the existing IoT hardware and software infrastructure must require minimal changes. For instance, modifying the IoT device's software or wireless protocol is likely infeasible. We therefore assume a solution cannot require modifying the device's code, tweaking hardware settings, changing radio communication parameters, or introducing new networked devices. To meet this requirement, we only require replacing the battery with a new device, and as batteries are typically intended to be user-serviceable, this is a non-invasive option. However, we assume the IoT device sends the raw battery voltage values to the cloud for further processing, and the battery voltage information is retrievable by applications.

5.2.2 Power Supply Constraints

Replacing the battery with additional circuitry imposes several challenges. First, the retrofit must not interrupt the IoT device's normal operation. That is, the supply voltage and current from the encoder output must be within the expected range for the device's original primary battery. For example, an IoT device originally operating with 2 AAA batteries expects a voltage between 2.7 V and 3.3 V. This constraints the voltage range available to encode information. Second, the current draw of the legacy device is considered unknown, and a device with a high dynamic range of current draw can affect the output of the retrofit device. For example, if the expected encoded voltage is set to 3.27 V, this should be stable whether the device's current consumption is 1 mA or 20 mA. Further, as the retrofit device must replace the energy supply, it must be able to output the expected voltage regardless of the voltage of its own underlying energy supply. In addition to these short-term conditions, the energy consumption overhead introduced by the encoder must be minimized to not unduly shorten the IoT device's operating time.

5.2.3 Battery Reading Resolution

The retrofit device can optimize the resolution and accuracy of its programmable voltage supply connected to the IoT device. However, due to the minimal modification constraint, the retrofit is still constrained by the battery voltage monitoring circuitry and software used on the IoT device. For example, if the IoT device expects a maximum of a 3.3 V supply, and uses a 12 bit ADC to collect battery voltage readings, the voltage resolution of these readings is $3.3/(2^{12}-1) = 0.806$ mV. That results in approximately 372 distinguishable voltage levels between 3.0 V and 3.3 V. This implies we could theoretically encode 8 bits of data by assigning voltage levels to the numbers 0-256. In practice, there is no standard for how should IoT devices report battery level. Manufacturers choose their own ADC resolution and the decimal precision of the battery voltage with which it can be retrieved from the cloud. Some also use battery level percentages instead of the actual battery voltage. According to our experience, it is common for IoT devices to report battery voltage readings with resolutions between 1 and 10 mV [89]–[91]. In addition to limited resolution, other battery voltage reading limitations are signal noise, nonlinearity, and offsets. We explore some of these challenges in more depth as well as an error mitigation strategy in Section 5.3.

5.2.4 Data Synchronization

From the analysis in Section 5.2.3, a single voltage reading can transmit a single byte of data. Sending more information will require using multiple battery voltage readings. However, this requires the retrofit device to loosely synchronize with the IoT device to set the voltage every time the device samples the battery voltage. Otherwise the same voltage value could be sent multiple times, or a value could be missed.

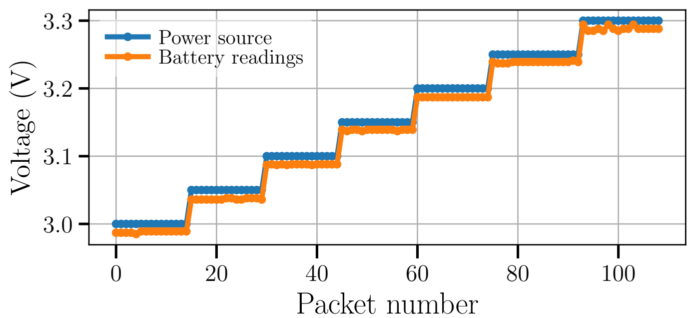


Figure 5.2: Battery voltage readings sent by a LoRa IoT device to the cloud via a LoRa gateway. This demonstrates the feasibility of encoding data in the battery voltage readings.

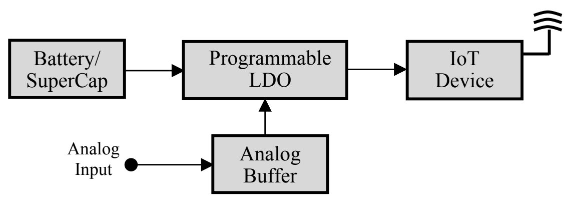


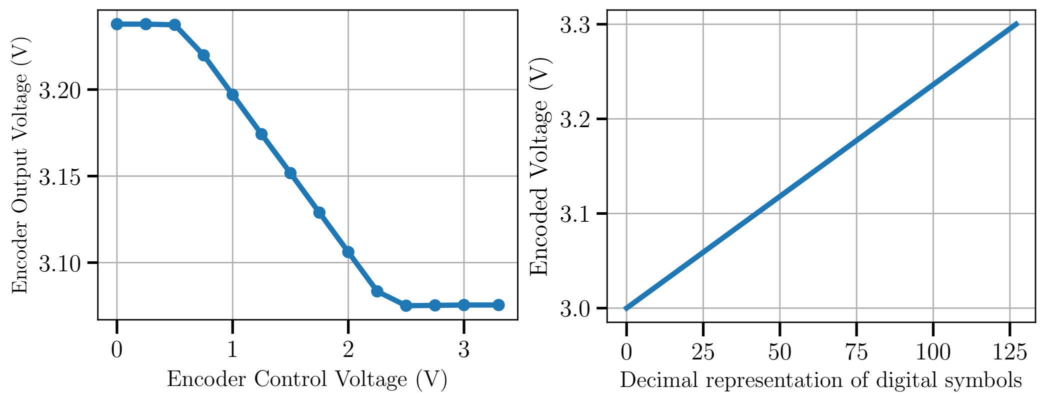
Figure 5.3: Analog encoder hardware design.

5.2.5 Recovering Transmitted Data

Once data has been encoded and the IoT device has (unknowingly) transmitted the data to its cloud backend, a processing algorithm must be able to recover the transmitted data successfully. This includes understanding how to divide the stream of transmitted voltage readings into the intended packets of data. We propose one possible and simple solution where two reserved symbols are used as a flag to signal the beginning and end of a multi-symbol message. These symbols will also be used in a decoding error mitigation step, discussed in more detail in Section 5.3.3.

5.3 RetroIoT Encoder Design

In this section, we design an approach to use the battery voltage channel to send analog readings and digital symbols over legacy IoT devices' network infrastructure.



(a) Analog encoder transfer function. **(b)** Digital encoder transfer function.

Figure 5.4: Transfer functions of the encoder.

5.3.1 Voltage Encoding Feasibility

As a proof of concept demonstration of the proposed approach, we attach a bench top voltage supply to the 3.3 V power rail of a LoRa IoT device and verify if we can receive the programmed voltage levels from the cloud. The device uses an ADC resolution of 12 bits. Figure 5.2 shows the battery readings collected in the cloud via the LoRa network against the ground truth input values. The close match suggests this approach is feasible.

5.3.2 Analog Encoder Design

The analog encoder directly translates an analog signal to a voltage suitable for the battery monitoring circuitry. The encoder accepts an input voltage ranging from 0 V to 3.3 V and adjusts the output of a low-dropout regulator (LDO) between 3.0 V and 3.3 V. The block diagram of the circuit is depicted in Figure 5.3. An analog buffer connects to an operational amplifier in a follower configuration to isolate the input signal source and the encoder control circuit. As the control voltage increases, the current flowing through the feedback resistor decreases, reducing the output voltage. Figure 5.4(a) represents the relationship between the control voltage and output voltage for the analog voltage encoder circuit. Although the usable control voltage range in this circuit is between 0.5 V and 2.3 V, this can be adjusted by adding appropriate gains and offsets with op-amp based analog circuits. The measured bias current for this circuit is 0.19 mA without any load

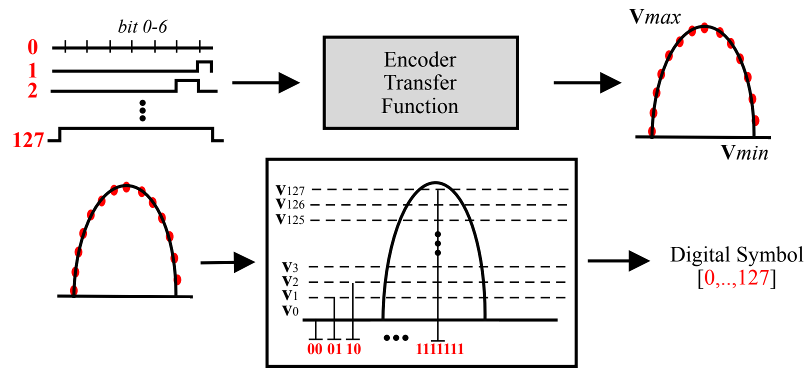


Figure 5.5: This diagram shows how the digital data is encoded (top) and decoded (bottom). The encoder converts the 7-bit digital symbol into a battery voltage value within v_{min} and v_{max}. The decoder function translates the encoded battery voltage back to a digital symbol.

connected to the encoder regulated output.

This simple analog voltage encoder supports directly connecting an analog input, for example an analog sensor, creating an easy-to-use option for retrofitting using the battery voltage monitoring channel. Section 5.6.4.

5.3.3 Digital Encoder Design

Using a purely analog input voltage reduces complexity, but limits the amount and type of data that can be transmitted using this channel. To show how arbitrary data can be transferred, we describe a technique to encode digital data into a range of battery voltage values and how to decode the received battery voltage to retrieve the sent information.

Data Encoding-Decoding. First, the data to be transmitted must be converted to a series of symbols. We select 7-bit values to represent the symbols based on the capabilities of our DAC device and the voltage range available to encode information. With a 7-bit representation, there exist 128 unique digital symbols each translating into a distinguishable voltage level. We define the difference between two consecutive encoded voltage as the resolution of the encoding v_{rs}. For instance, the first encoded voltage can be calculated as $v_1 = v_0 + v_{rs}$. Voltages v_0 and v_{127} are respectively the minimum (v_{min}) and maximum (v_{max}) encoded voltages. Therefore, the n-th encoded voltage level can be denoted as $v_n = v_0 + nv_{rs}$, where v_n is n-th voltage level. Both the resolution of the encoding volt-

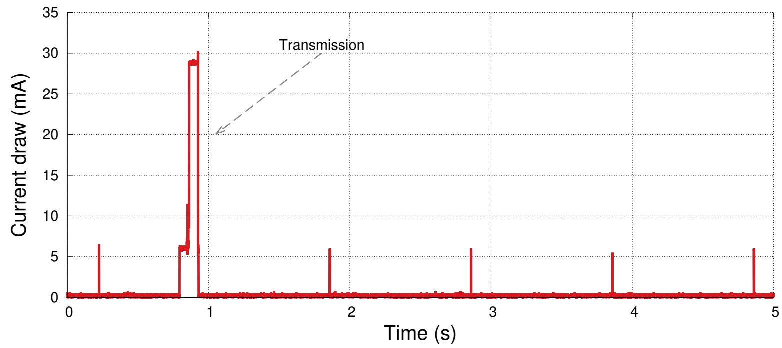


Figure 5.6: Current draw profile of a LoRa sensor [89] showing a distinct radio transmission spike.

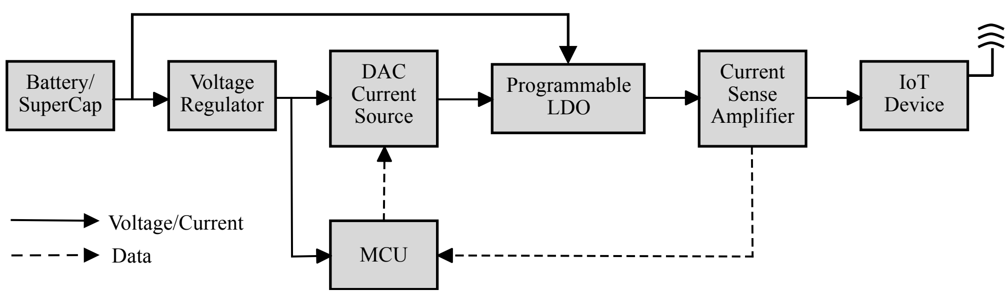


Figure 5.7: Block diagram of the digital encoder hardware design.

age and the IoT's reported battery reading resolution affects how well the symbols can be retrieved by a cloud application. In Figure 5.4(b) we plot the relation between encoded voltage between 3.0-3.3 V and decimal representation of the corresponding symbols.

To decode the information on the cloud application, the received battery voltage levels must be converted to symbols and then properly interpreted. A voltage level v is decoded as a unique symbol n if it satisfies $(v_n - \frac{v_{rs}}{2} < v < v_n + \frac{v_{rs}}{2})$. Figure 5.5 shows a block diagram of this process.

Data Synchronization. To ensure that the symbol to be transmitted is encoded approximately right before the device transmits a radio packet, the encoder needs to learn the device's transmission schedule. This is essential to support packets of data spread over multiple transmissions. We propose achieving this synchronization by measuring the current draw of the legacy IoT device, and observing spikes in the current trace. As battery powered devices must minimize their current draw, wireless transmissions will likely result in distinct spikes in the current trace as shown in Figure 5.6. The retrofit device can then update the voltage value every time it detects a transmission event. As a majority of the IoT sensing applications are fairly periodic, the battery voltage encoder observes the device's current draw and measures the time difference between two consecutive peaks resulting from a radio communication to determine the transmission interval. Then, it uses this interval to determine when to encode the next symbol.

Hardware Design. Figure 5.7 depicts the block diagram of the hardware design of the digital encoder. An I^2C-controlled digital-to-analog converter (DAC) current sink/source IC adjusts its output current across 128 values to produce a variable output voltage signal. The variable output current of the DAC is injected into the feedback node of a voltage divider that feeds into an adjustable output voltage low dropout regulator. The DAC current source programs the LDO regulator to be configured at one of the 127 voltage output levels. We use an ultra-low power MCU to send symbols through the I^2C interface of the current DAC. The MCU uses a current-sense amplifier to monitor the IoT device's current draw and calculates the transmission interval of the device.

5.3.4 Decoding Error Mitigation

By running encoding and decoding experiments as depicted in Figure 5.2, we identify that the difference between the encoded power source voltage and the IoT's battery reading voltage can be modeled as the sum of a constant and a linear term, representing offset error sources from the voltage encoder and the IoT device's ADC. To mitigate these errors in the symbol decoding process, we estimate the encoded power source voltage before decoding the received symbol. To estimate the encoded power source voltage from the IoT device's battery readings, we perform a linear interpolation using the maximum and minimum IoT device's battery readings (v_{bmax} and v_{bmin} respectively), obtained from setting the encoded power source voltage to v_0 and v_{127}, respectively. Equation (5.1) shows how we estimate the encoded power source voltage v_{pwr} from the IoT device's battery voltage reading v_b.

$$v_{pwr} = v_0 + (v_b - v_{bmin}) * \frac{(v_{127} - v_0)}{(v_{bmax} - v_{bmin})} \tag{5.1}$$

We assume the maximum and minimum encoded voltages (v_0 and v_{127}) are special

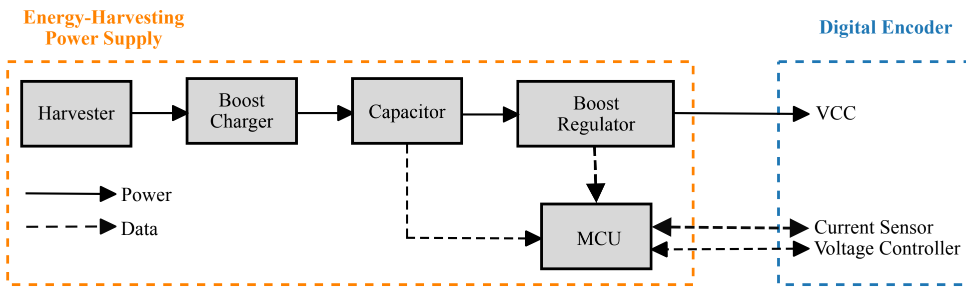


Figure 5.8: Energy-harvesting power supply module interfacing with the digital encoder.

encoded voltage levels used only for calibration purposes (0 and 127 are then reserved symbols), while also periodically reporting them so they can be later used in the v_{pwr} estimation and decoding process. This approach results in reduced decoded bit error at the cost of decreased bandwidth due to the use of reserved symbols and special calibration messages as we will evaluate in Section 5.6.2.

5.4 Energy-Harvesting Retrofitting

One of the promising applications of upgrading a deployed IoT system is to replace batteries with energy-harvesting power supplies. However, successful energy-harvesting systems must adapt their execution based on available energy. A device designed with a reliable source of energy (e.g. a battery) will not have the programming or included logic to adjust its own operation based on the current harvesting conditions. In this section, we show how the retrofitting approach can address this challenge.

We start by replacing the battery with an energy-harvesting power supply that connects to the existing power and ground terminals, as shown in Figure 5.8. The new power supply replaces the energy store with a supercapacitor that is recharged with a harvester. A second stage voltage regulator regulates the capacitor voltage and supplies a constant voltage to the rest of the system.

In ideal harvesting conditions simply doing this swap would be sufficient. However, the available harvestable energy may not be sufficient to recharge the capacitor at the rate the legacy IoT device requires. To address this, we integrate a small microcontroller into the replacement power supply. The MCU observes the state of charge of the storage

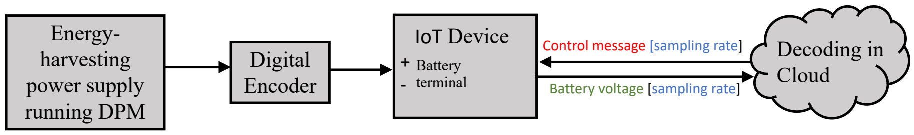


Figure 5.9: The retrofitting energy-harvesting power supply runs a dynamic power management algorithm locally and encodes the updated sensor sampling rate in the battery voltage. The sensor is then re-configured by the cloud control message to adjust device behavior.

element and the incoming harvested energy. If it detects a shortfall, it configures the IoT device to reduce its operation to conserve energy. Since the new power supply is only connected via the old battery terminals, it cannot do this directly.

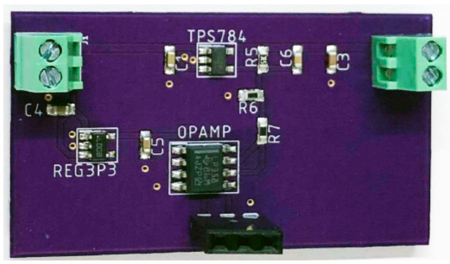
Instead, we leverage the underused battery voltage channel. To adjust the device's duty-cycle, the MCU creates a message by encrypting the recommended duty cycle in the battery voltage and transmits it to the cloud. An application hosted in the cloud receives the device's message and then tries to alter the device's operation to match the available energy constraints. This process is illustrated in Figure 5.9. IoT devices often times allow some degree of re-configuration through cloud APIs, particularly related to update rates. For example, control messages may be able to set the sampling period [90] or the keep-alive interval [89]. The cloud application uses one of the existing methods to send a control signal to the device to adjust the operation of the legacy IoT device.

Our proposed dynamic power management algorithm is shown in Algorithm 2. At each interval t_p, the power supply checks the gradient of storage voltage and if the gradient is either zero or has a positive value, the cloud is instructed to increase the sampling frequency of the sensor, and vice versa.

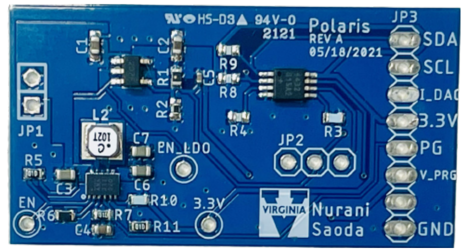
5.5 Implementation

We implement the RETROIOT encoder and power supply designs using prototype PCBs.

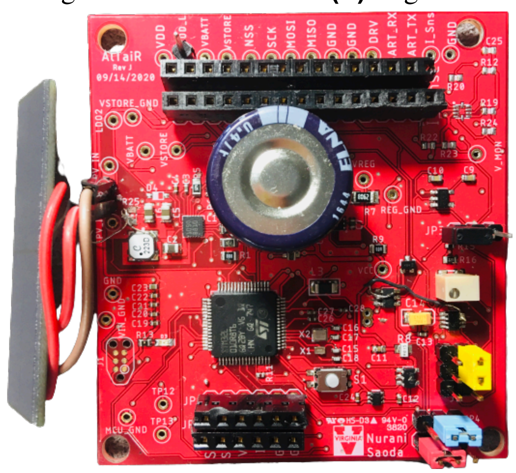
Analog Voltage Encoder. The analog voltage encoder is based on the low-dropout (LDO) regulator TPS784 [92] to adjust the output voltage. It uses the low power operational amplifier LP358 [93] to implement the buffer circuit for the analog input voltage. A LP2980 [94] LDO regulator with a fixed 3.3 V output voltage powers the operational



(a) Analog voltage encoder



(b) Digital voltage encoder



(c) Energy-harvesting power supply

Figure 5.10: Prototype voltage encoder circuit boards and energy-harvesting power supply board.

amplifier. Figure 5.10(a) shows the prototype.

Digital Voltage Encoder. The digital voltage encoder board uses a Maxim Integrated DS4432 [95] DAC current source/sink amplifier. The current output of the IC can be controlled by I^2C commands to set a a variable output voltage of a LDO. We integrate a Texas Instrument TPS784 [92] as the LDO with an output voltage accuracy of $\pm.75\%$. The board also accommodates a Monolithic Power MPQ28164 [96] buck-boost switching voltage regulator with an efficiency above 85% at input voltage of 3.3 V that supplies voltage to the components. The assembled PCB is 4.3 cm by 2.3 cm. Figure 5.10(b) shows the hardware.

Power Supply. We adopt the ALTAIR [97] hardware platform as the energy-harvesting power supply. Figure 5.10(c) shows the PCB of the energy-harvesting add-on module.

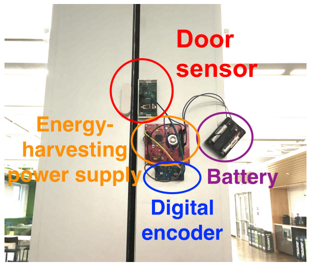
The energy-harvesting power supply board accommodates an energy-harvesting front-end and an ultra-low power MCU to monitor the device current draw and send the appropriate commands to the digital encoder circuit. An ultra-low power battery charger boost converter SPV1050 [98] charges a supercapacitor from a solar or TEG harvester until it reaches 3.1 V. A nano-power boost regulator MAX17222 [99] with $>70\%$ efficiency at $10\,\mu\mathrm{A}$ of input current regulates the supercapcitor voltage after its voltage reaches 2 V. We use monocrystalline IXYS solar cell as the harvester and a 470 mF supercapacitor with an ESR value of $25\,\Omega$ as electrical storage. We adopt an ultra-low power 32-bit ARM Cortex-M0+ STM32 [100] MCU to implement the dynamic sampling rate algorithm as explained in Section 5.4. The MCU monitors the load current draw by sampling a MAX9634 [101] current amplifier.

5.6 Evaluation

To evaluate our system, we explore how accurately and reliably information can be retrieved from the voltage encoder through the battery voltage channel. We perform an extensive study to investigate the battery voltage channel characteristics in terms of voltage error, percentage of bit error per packet, and percentage of correctly decoded packets. We build two applications using commercial IoT devices to encode custom digital metadata, one application to retrofit with energy-harvesting, and one application to transmit readings from an analog sensor. We demonstrate how the proposed technique can help retrofit existing devices and how a functional end-to-end system can be built just by accessing the battery voltage terminal of the device.

5.6.1 Methodology

Experimental Setup. To investigate the battery voltage channel characteristics (Section 5.6.2), we connect the analog and digital encoder boards with a STMicroelectronics LoRaWan discovery kit [102]. The LoRaWan device measures readings from an attached MS5837-30BA [103] pressure sensor and transmits the encoded readings in the sampled battery voltage information twice every minute. We disconnect any power source from the discovery kit and replace it with the programmable voltage encoders by directly con-



(a) On door

(b) On cabinet

Figure 5.11: We deploy RETROIOT with a door sensor in different locations. Picture corresponds to two of the deployment scenario.

necting it to the 3.3 V rail. We sample battery voltage with ADC resolutions of either 12, 10 or 8 bits and the reported voltage readings at the cloud have 1 mV resolution.

Retrofitted Devices and Applications. We retrofit two commercial LoRa sensing devices with upgraded functionality: 1) a door event sensor [89] and 2) a soil moisture sensor [90]. We upgrade the LoRaWan door sensor with an analog TMP37 [104] temperature sensor and a location metadata tag. We upgrade the soil moisture sensor with the solar energy-harvesting power supply. The goal of the sensor add-on experiment is to evaluate the fire extinguisher application scenario described in **??** by using temperature readings as an alarm to indicate unusual storage conditions. We artificially heated the sensor to simulate changes in ambient temperatures that would trigger the alarm. The door sensor sends a radio packet every minute with a door open/close event along with the battery voltage reading. The soil moisture sensor, by default, sends a reading every ten minutes. For these devices, the battery voltage is reported with 1 mV resolution. We also upgrade one off-the-shelf BLE temperature and humidity sensor [91] with long digital metadata representing a 32-bit timestamp value. This sensor reports battery voltage up to 10 mV resolution at approximately each hour.

Cloud Application. The LoRa IoT devices are connected to The Things Network gateways [105] and the messages are received and stored by a TTN application with storage and MQTT integration. For the door sensor applications, a Python script downloads the messages from the storage integration of the TTN application and then decodes the battery values. For the soil moisture sensor, a Python script running a MQTT client application connects to the TTN application's MQTT broker, then receives and decodes the

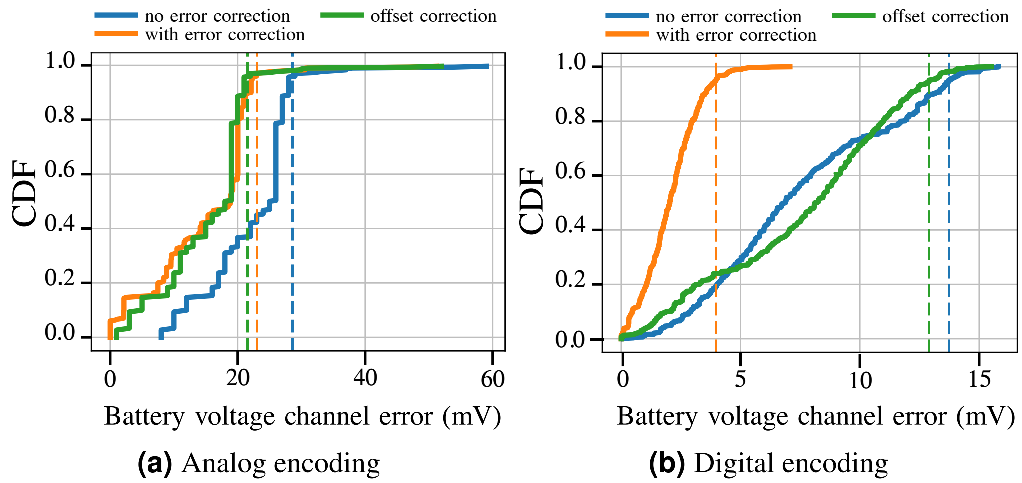


Figure 5.12: CDF of the error in received battery voltage. The channel error is significantly reduced after calibration using the proposed error correction technique. The dash lines correspond to 95th percentile error values.

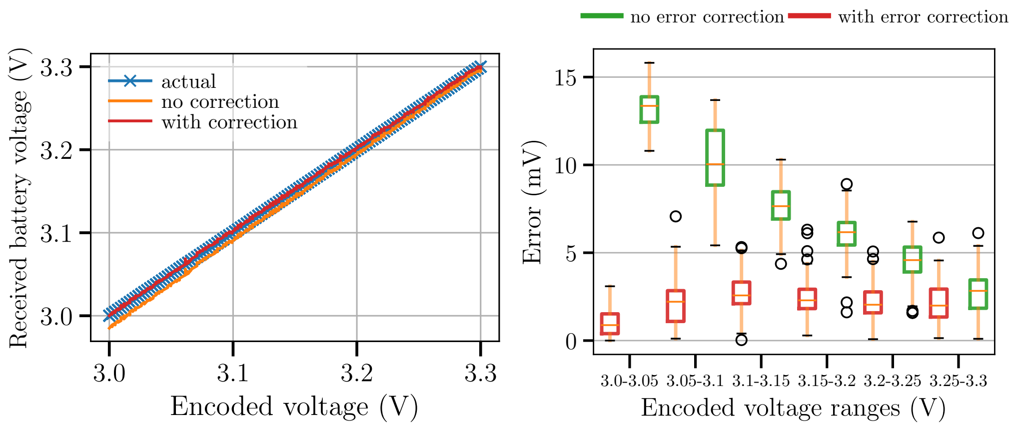
sensor's messages to obtain the energy-harvesting retrofit commands. The Python script then sends the appropriate downlink command to update the wake up period of the LoRa IoT device. For the BLE sensor, the manufacturer provides a cloud API that allows sensor data and battery voltage information to be downloaded by our Python script.

Deployments and Experiments. We deploy the door event sensor with location metadata add-on functionality at four different locations: on a door, a cabinet, a fridge, and a drawer. Figure 5.11 shows the deployment.

5.6.2 Battery Voltage Channel Characteristics

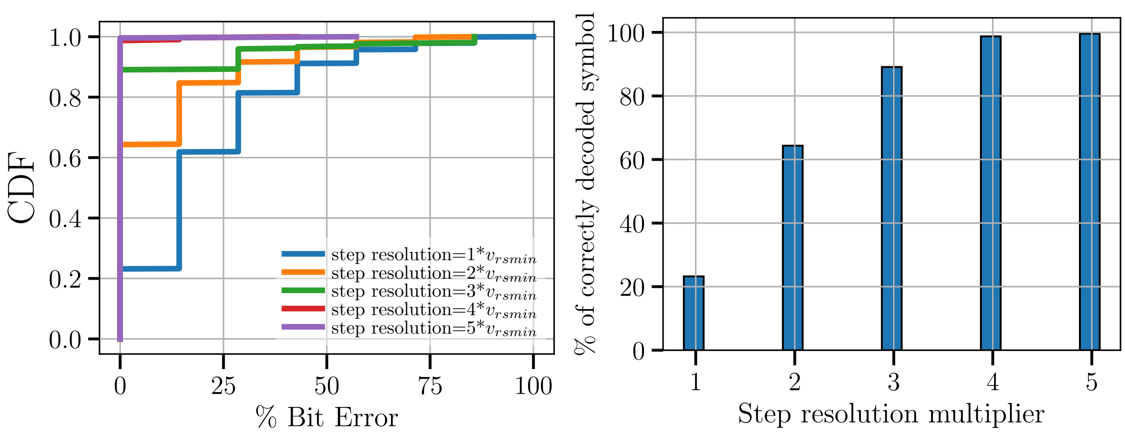
In this section, we evaluate the error induced in the battery voltage channel and how the resolution of the channel affects successful decoding of information encoded in the battery voltage readings. Understanding these metrics is essential for further developments using such channels.

Received Voltage Error. To estimate the difference between the battery voltage sent from the voltage encoder and the battery voltage received at the cloud, we sweep the encoded voltage from the minimum (3.0 V) and maximum (3.3 V) values, report the value over a LoRa radio packet using the STMicroelectronics LoRaWan board. We collect two



(a) **(b)**

Figure 5.13: The efficacy of the error correction technique on the battery voltage readings. After applying the error correction, the received voltage values match better with the actual encoded voltage.



(a) **(b)**

Figure 5.14: a) shows how the percentage bit error improves as we increase the step resolution of the voltage encoder. With step resolution, $5 * v_{rs} = 11.81$ mV, we can correctly decode 99% of the sent symbols.

samples per minute for five minutes at each voltage level. After retrieving the battery voltages, we perform error correction on the data using Equation (5.1) as described in Section 5.3.3. We also perform a simple offset correction using just one of the two reserved symbols. We measure the CDF of error in the battery voltage, denoted by the difference between encoded voltage and received voltage values and analyze the results with and without error correction and the offset correction technique. Figure 5.12(a) and Figure 5.12(b) show the CDF of the errors while encoding analog and digital data, respectively. Battery voltages in IoT devices usually have a limited acceptable operating range below which the device is turned off. With more error induced in the battery voltage channel, the bandwidth of information that we can successfully decode decreases. The 95th percentile of the error is 28.42 mV for the analog data and 13.71 mV for the digital data without any error correction. With correction, the error can be bounded within 20.91 mV and 3.96 mV. Figure 5.13(a) shows the shift in voltage values after the error calibration on the digital data, which significantly reduces channel error. We further break down the error values across the whole spectrum of the voltage levels and show the variation in Figure 5.13(b).

Successful Decoding vs Encoder Resolution. The digital voltage encoder encodes a 7-bit data into the battery voltage. For a packet to be correctly decoded, the voltage error should be within the voltage difference corresponding to two symbols. The bandwidth of the channel is proportional to the number of achievable voltage levels. To evaluate how many bits per packet are incorrectly decoded, we analyze the CDF of percentage bit error with increasing step resolution (v_{rs}) starting from the minimum step resolution of the encoder at 2.36 mV. As shown in Figure 5.14(a), we observe that we can successfully decode 99% bits with a step resolution of $5 * v_{rs}$= 11.81 mV. Figure 5.14(b) shows the percentage of symbols that are correctly decoded across different encoder resolution. For this experiment, we perform the error calibration before the analysis.

5.6.3 Hardware Variation Effect

We quantify the errors produced as an artifact of the hardware imperfections of the encoder itself and the IoT device. Specifically, we consider the variation in the encoder output voltage and variations in the reported battery voltage by the retrofitted IoT device due to different ADC sampling resolutions. Due to component variations, we expect the encoder output voltage to be slightly different across different boards. In Figure 5.15(a), we show the programmed output voltage of the encoder for three different boards as we perform a full voltage sweep. Though none of the encoder outputs violates the linearity of the transfer curve, encoders two and three have larger shift in between their transfer

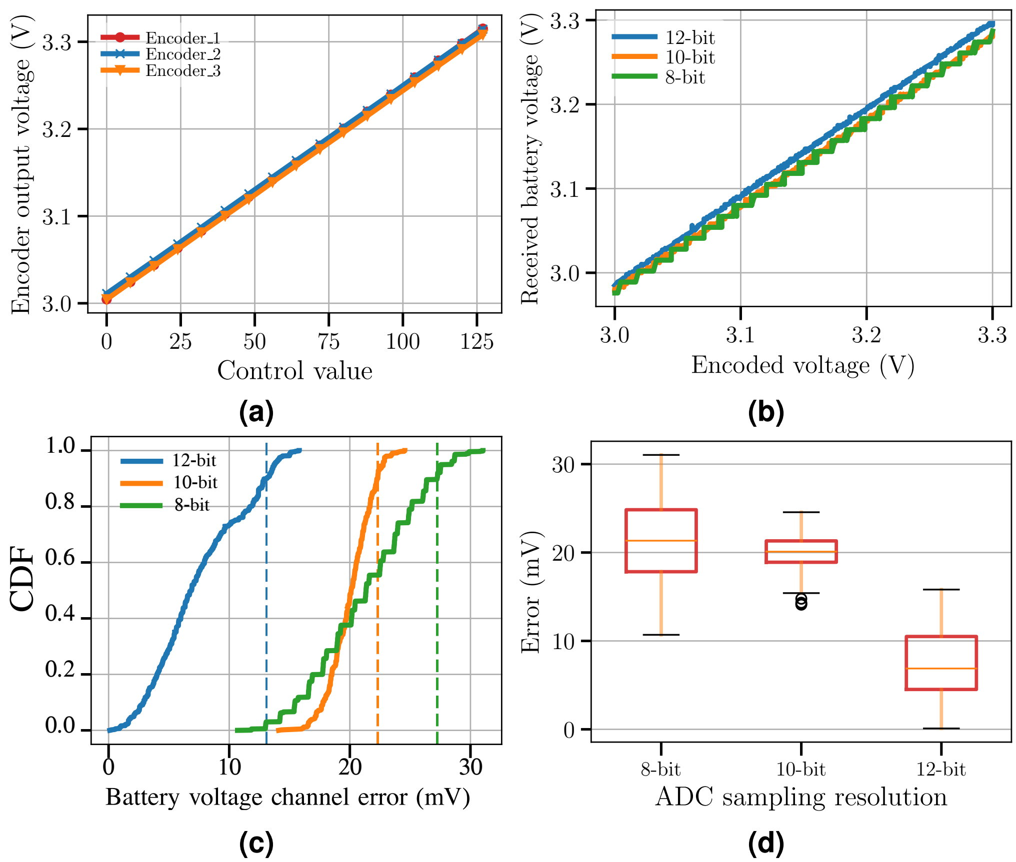


Figure 5.15: Understanding the effect of different sources of error due to hardware limitations of the design. a) captures the difference in encoder output voltage of three different boards. b) shows that with lower ADC resolution, the number of distinguished voltage levels is reduced, which compromises the bandwidth of the channel. c) and d) characterize the distribution of end-to-end channel error.

curve than encoder one and three.

Next, we quantify the variation in the battery voltage readings sampled by the LoRaWAN IoT device [102] with different ADC resolution. Different MCUs in the device can come with different resolutions among which 8, 10, and 12-bit are well-supported by most devices. In Figure 5.15(b), we see that higher ADC resolution allows us to achieve higher encoding resolution, while lower 8-bit resolution compromises the necessary voltage levels. The error distribution with different ADC resolution in Figure 5.15(c) and Figure 5.15(d) show that the maximum error can in fact be reduced by more than two

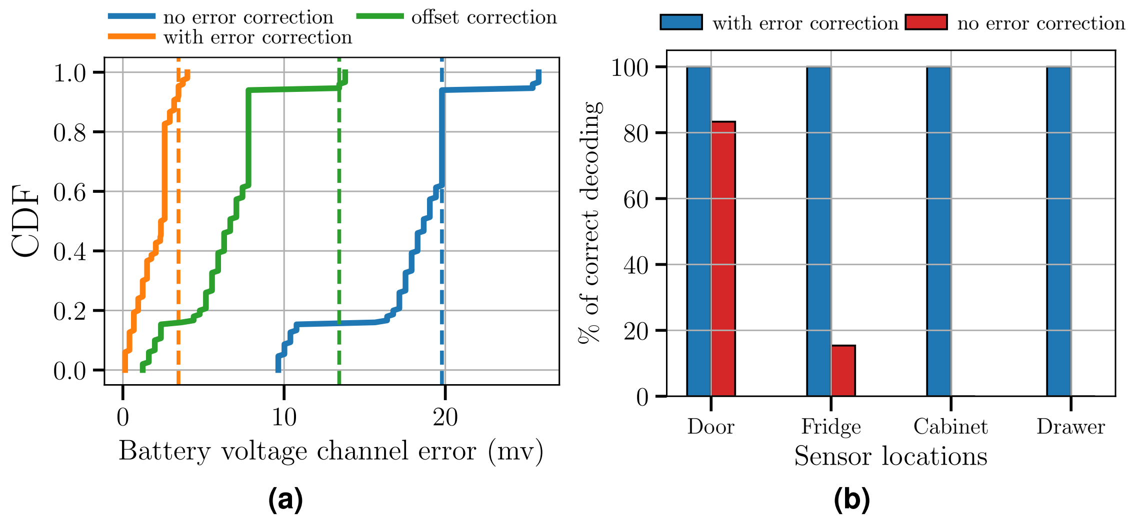


Figure 5.16: Results from the door sensor metadata application. We can successfully decode all 12 unique metadata after error mitigation.

times with 12-bit resolution.

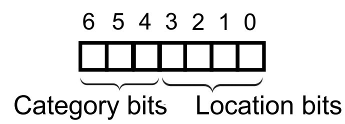
5.6.4 Real World Applications

We augment two COTS door event sensors with temperature sensing functionality and deployment specific metadata, one COTS BLE sensor with a timestamp metadata, and one soil monitoring sensor with a light energy-harvesting power supply. We investigate how accurately the encoded sensor data can be retrieved and report our findings in this section.

Temperature Monitoring. For this application we attach a TMP37 temperature sensor to the analog encoder board and evaluate the sensor input voltage at normal operations. We measure the sensor output voltage as 0.55 V (equivalent of 27.5 °C) and measured the encoder regulator voltage output as 3.237 V, while the cloud application indicated a battery voltage of 3.246 V. Heating the temperature sensor raised its output voltage by about 1 V (equivalent of 50 °C), decreasing the encoder regulator voltage output to 3.1965 V, while the cloud application reported 3.198 V. This experiment demonstrated that the analog encoder regulator is capable of translating the temperature sensor readings into detectable alarm events at the cloud application with the threshold temperature being

```
{
door_sensor:
  _metadata:
        {
        category: "cabinet"
        location: "living_room"
        }
}
```

(a)



(b)

Figure 5.17: Example of door sensor tag metadata in a). Figure b) shows how the 7-bit digital symbol can encode location and category information.

around 30 °C.

Digital Metadata Transmitting. We enhance the door sensor device with a simple digital metadata tag that informs what type of event it reports and where it is deployed. Figure 5.11 shows two installations. The digital metadata encoded in the battery voltage is unique for each sensor in a deployment area. Figure 5.17 shows an example of how the information can be encoded in a symbol. We categorize the sensor into four types based on the equipment it is monitoring: door, fridge, cabinet, and drawer and assign three location string for each of them based on which room they are located. These types of tags are useful for in smart home monitoring applications where the number of deployed sensors are only a handful. In total, we encode 12 unique symbols each representing different installed sensors.

We observe from Figure 5.16(b) that without proper error mitigation some of the metadata are not decoded correctly due to the high channel error shown in Figure 5.16(a). But after calibration, we could decode the metadata correctly for all of the samples. We could achieve this accuracy as the symbols are spread enough over the encoded voltage range. Though we can only encode 128 unique metadata tags, one can overcome the data bandwidth limitation by chaining multiple symbols together as we show in our BLE sensor experiment.

Multi-Symbol Metadata Transmitting. We upgrade a commercial BLE temperature and humidity sensor with an encoded 32-bit metadata message representing a unix timestamp. To transmit this message, we use ten symbols which are updated every hour.

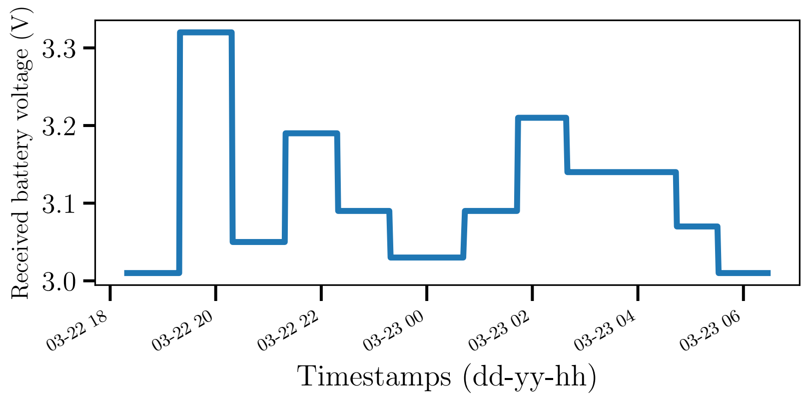


Figure 5.18: Detected voltage levels of the multi-symbol metadata transmission. Two reserved symbols equivalent of the maximum and minimum encoded voltages are used to mitigate decoding errors and as a flag for message start and end. The following encoded voltages represent the 32-bit unix timestamp for “2022-03-22 16:22:49”.

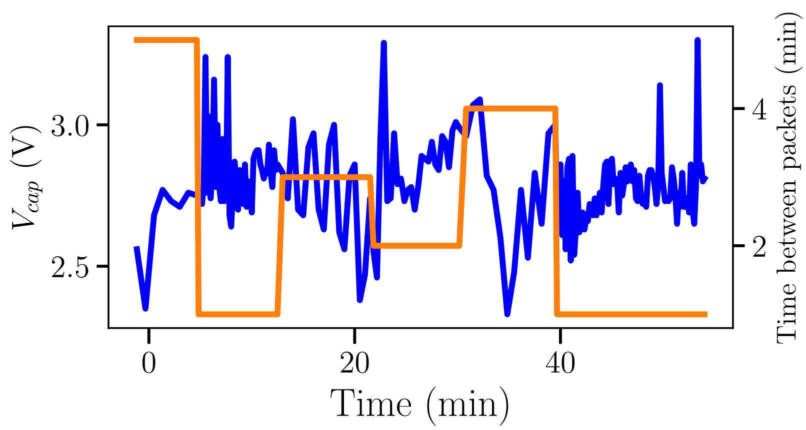


Figure 5.19: The dynamic sampling rate controlled by the energy-harvesting power supply with changing capacitor voltage over time.

The first two symbols of the message are the reserved maximum and minimum voltages provided by the encoder regulator, used for decoding error mitigation as explained in Section 5.3.3. The following eight encoded voltages are 4-bit symbols representing subsections of the 32-bit unix timestamp. We show in Figure 5.18 the sequence of battery voltage readings for the timestamp corresponding to “2022-03-22 16:22:49”.

Replacing Battery with Energy-harvesting. As a demonstration of how the battery voltage channel can be leveraged to convert a battery-powered sensor to an energy-harvesting one, we replace the battery of the soil monitoring sensor and plug in our energy-harvesting power supply. We implement Algorithm 2 in the power supply board and report the optimized rate to the cloud in the battery voltage reading. The sensor period is updated every ten minutes. Figure 5.19 shows the instantaneous capacitor voltage and calculated sensor period.

5.6.5 Energy Overhead

As the encoder and power supply design result in an energy overhead, we evaluate the overall power draw of each module.

The Analog Voltage Encoder Module. We measured a total standby current of $190\,\mu A$ for our analog module prototype without any sensor load. Adding the temperature sensor TMP37 resulted in a total current consumption of $212\,\mu A$.

The Digital Voltage Encoder Module. We measured a total standby current of $518\,\mu A$ for our digital module prototype without any device connected to the I2C interface.

Energy Harvesting Retrofit. The retrofit module consisting of a power management IC and a low power microcontroller has a quiescent current of about $95\,\mu A$. The digital encoder module together with the energy harvesting retrofit consumes a total of about $613\,\mu A$.

Prototype Limitations. Our proof of concept design goal is to demonstrate how the control of the battery terminals voltage can be used as a new data channel, focusing on data encoding and recovery steps. Our prototype is not optimized to achieve minimal power consumption, and as such its standby power consumption can be too high for some battery-powered applications. Achieving lower standby current is possible by replacing the low-dropout regulator with a more efficient voltage converter and disabling unnecessary circuits while the sensor is in sleep mode. When retrofitting an IoT device to use energy-harvesting, the harvester should be selected with proper consideration of the energy consumption overhead.

5.7 Discussion

Our prototype demonstrates the feasibility of augmenting existing IoT deployments with new data streams, and here we discuss some limitations, remaining challenges, and potential mitigations.

Increased Power Draw. Adding a controllable power supply and new sensors to an IoT device increases its overall power draw, and if the retrofitted device retains its original battery capacity, the IoT device will require more frequent maintenance interventions to replace batteries. To mitigate this maintenance overhead, the designer can adopt larger battery capacity in the retrofit power supply module or adopt an energy-harvesting solution compatible with the retrofitted IoT device energy requirements.

Data Channel Bandwidth. Our retrofit approach is constrained by the battery level reporting choices made by IoT device manufacturers, restricting the maximum achievable data bandwidth for a given application. For example, the Decentlab's soil humidity LoRaWAN sensor [90] reports its battery voltage with every uplink data message as a four-digit integer value with millivolt resolution (typically 2100 to 3300 mV), while the Seed Studio's LoRaWAN CO2 sensor [106] reports its percentage battery level after every 10 uplink data messages as a three-digit integer value (0 to 100%). While only very low throughput might be achievable under some IoT platforms, it can still be of great value to applications, for instance to enable alarm features or to support IoT deployment management by encoding a batch number or expiration date over multiple transmission as we demonstrated in Section 5.6.4.

Hardware Heterogeneity. Different hardware platforms may have different acceptable voltage ranges and resolutions for their battery voltage monitors. This essentially alters the data channel for the retrofit device. To accommodate this, a programmable range selector can be added to change the voltage output range. Also, using fewer voltage values could help with resilience at the expense of datarate.

Cloud API Access. We rely on the cloud API to retrieve the encoded battery voltage. For some signals, like the on-off of a button, this is likely readily available. But the battery voltage readings, may not be exposed through an API, either only used locally by the application provider or exposed only through a "battery low" alert. This limits the channels that can be used for this approach, or requires further consideration of the cloud-provided API when considering how the data to communicate is encoded. For example, a battery low alert could still be used as a low data rate channel.

Lossy Channels. The retrofit data channel may be constructed on top of a lossy underlying channel, and therefore data symbols can be lost. If the receiver is expecting to use multiple symbols to decode a packet, the protocol must handle the potential lossiness. Many standard data communication techniques could be used, including checksums and packet headers with length values.

Retrofit Synchronization. To synchronize the voltage encoder with the unmodified sensor we detect its sampling interval and only output new voltage readings before we expect the sensor to take its next reading. However, if the sensor is event-based, it may not follow a regular pattern when sending battery voltage state. This would hinder the ability to send packets of data without missing or duplicating symbols. One workaround is updating the voltage output only after a detected current spike, however, this would lead to an unpredictable datarate and perhaps stale data if events are infrequent. Some sensors both detect events and have a periodic transmission (such as a heartbeat packet), and a future version of this work could attempt to identify the regularly spaced packets and only transmit using those.

Another challenge related to our synchronization approach is that sensor devices also increase their power draw during receive mode, what could be falsely identified as a triggering event. However current peaks tend to be significantly lower for receiving modes, so the retrofit module controller can learn the IoT operation pattern and only use the highest current peaks as trigger events.

Another potential opportunity is the coupling between the energy harvesting rate of the devices in Section 5.4 and the datarate of the channel. More favorable harvesting conditions could lead to a better performing channel as the sensor is able to transmit more often. This increased performance may enable a secondary use of the channel and change how the energy-harvesting optimization algorithm works.

Attack Potential. The ability to send data through the battery voltage channel, and that many devices are designed with user serviceable batteries, suggests that a possible attack vector is surreptitiously replacing the battery in the target IoT device with a "smart battery" that is controlling its own voltage output to exfiltrate data without any visual signs of tampering. The attacker would still need to be able to access the data once it is sent to the cloud, but the end-to-end attack may be feasible in conjunction with another vulnerability. Further analysis is required to understand the extent of this possible issue and future safeguards.

5.8 Conclusion

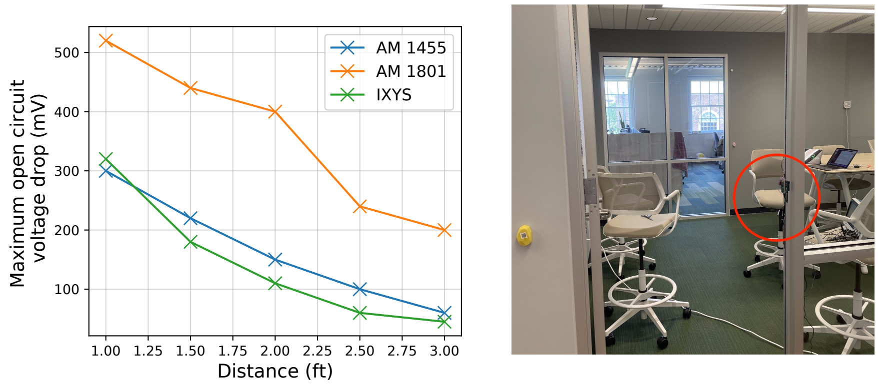
As IoT deployments grow larger in scale, designs and techniques that build on the existing device and network infrastructures can unlock many new applications and capabilities. Such design technique can not only enhance the functionality of existing systems, but also can significantly reduce the design time, developer overhead, maintenance cost, and immature device obsolescence. In this chapter, we introduce one such technique that encodes information in the battery voltage enabling end-to-end communication, which otherwise just provides insight-less battery voltage information. We envision that this can lead to future explorations of other interesting underused channels in IoT deployments. Further, providing open and configurable channels can increase the solution flexibility and usefulness of new IoT devices and infrastructure. Open analog and digital ports and cloud API support to retrieve acquired data enable future users to customize IoT platforms for their own need at reduced cost and design effort.

Chapter 6

SolarWalk: Sensing with Photovoltaic Harvesters

Energy-harvesters convert other forms of energy available in the environment to electrical energy to power small sensors in-situ. These sensors are flexible to be deployed in remote and hard-to-reach spaces which also makes them more susceptible to experience power supply fluctuations. The output voltage of an energy-harvester becomes noisy or changes abruptly due to transient source of energy. For example, fundamentally, the output voltage of a photovoltaic harvester is proportional to the level of illuminance at its cell surface. The harvester's voltage changes as people turn on/off light source or objects obscure the direct path of light. One interesting observation is that human movement is a potential cause of power supply glitches. This lead to one key insight that changes in the illuminance level caused by a particular external activity creates an imprint on the voltage pattern of the PV cell, which can carry information. For example, the voltage jitters caused by a person walking past a PV-powered sensor is, in fact, a unique identifying feature of that person due to height, body shape, and gait differences in individuals. Interestingly, the pattern also has a directional property.

Based on these observations, we present *SolarWalk*, that distinguishes persons in a smart home by exploiting the voltage side-channel of PV harvesters. *SolarWalk* takes a crucial step to establish that a noisy harvester can actually perform as a good sensor. Further, such sensors can potentially enable many other sensing applications where the harvester reacts an external activity. The potential behind re-purposing the energy harvester of these devices to function beyond just a power source is tremendous. With *SolarWalk*, an energy-harvesting PIR [107] or a door status sensor [108] is not only able to sense the presence of a person but also can determine who the event is associated with just by inspecting the ripples in the harvester voltage. By enabling this, *SolarWalk* further augments the capabilities of energy-harvesting sensors.



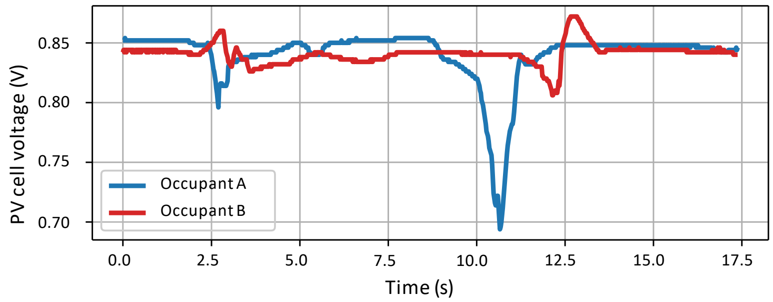
(a) **(b)**

Figure 6.1: a) A PV cell's open circuit voltage drops to different levels as someone walks at different distance from the solar cell's surface. b) Experimental setup with PV cell mounted on a office doorframe.

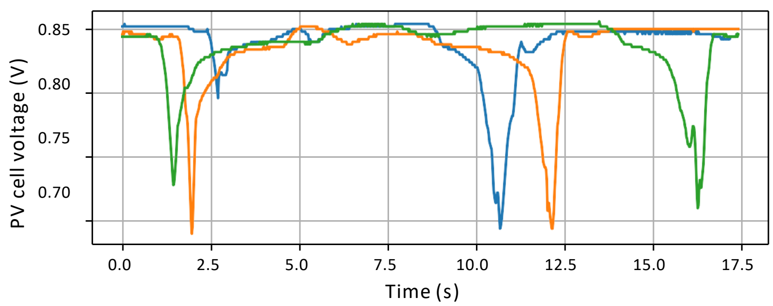
6.1 Occupant Identification using Photovoltaics

Indoor light energy-harvesting sensors typically harvest energy using one or multiple small photovolatic cells. These solar cells are usually optimized for a specific range of wavelength associated with indoor lighting conditions and the open circuit output voltage is proportional to the light intensity of its surrounding. During normal operation, the light intensity of indoor spaces changes steadily throughout the day until the light is turned off. The light intensity of the surrounding, however, undergoes a rapid change when someone passes nearby and is reflected in the output voltage of the solar cell. The maximum open circuit voltage drop induced on the solar cell decreases as the shadow of the person diminishes. Figure 6.1(a) shows the maximum voltage drop in three different indoor solar cells (both amorphous and monocrystalline) [109], [110] as someone walks by at different distances from the solar cell surface. Maximum voltage drop occurs when the person stands right in front of the cell completely blocking majority of the light exposure on the surface.

To better understand the characteristics of the open circuit output voltage when someone walks by within a few feet, we installed an IXYS indoor PV cell [109] on a door frame, halfway above the ground as shown in Figure 6.1(b). The solar cell surface is placed orthogonal to the floor. Since occupants walk in a narrow passage in spaces like



(a) Voltage fluctuations of occupant A and B are different from each other.

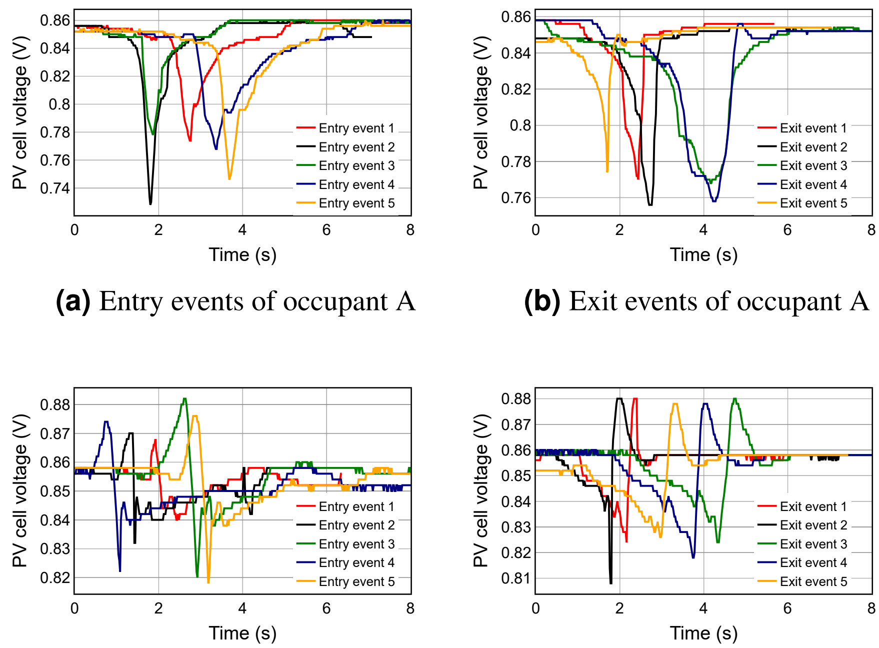


(b) Voltage fluctuations of the same occupant have similar shape.

Figure 6.2: This figure shows how the output voltage of the solar cell mounted on a doorframe ripples as different occupants pass through the door. The maximum voltage drop and the duration of voltage fluctuations vary differently for occupant A and B . On the other hand, these characteristics remain consistent over multiple trials by the same person.

doorways and hallways, such places are best suited for this study. We record the voltage traces as we enter and exit through the door. From Figure 6.2(a), we find that, for two different persons the voltage traces have different amplitude over time. Voltage drops as the person obscures the surface of the cell and restores itself as the person walks away. The amplitude of the ripple voltage is related to the height of the person and time length of the ripple is associated with someone's gait or walking style. However, Figure 6.2(b) shows that the shadow pattern is similar for the same occupant. This indicates that shadow pattern of a person as observed by a solar cell can be a characteristic feature for occupant identification.

Moreover, the pattern for different entry and exit events are distinguishable and can



(a) Entry events of occupant A **(b)** Exit events of occupant A

(c) Entry events of occupant B **(d)** Exit events of occupant B

Figure 6.3: From a) and b), we see that occupant A's entry and exit patterns are distinguishable. The patterns associated with the same type of event is similar. Since during entry and exit, the light is obstructed in similar but reverse direction, the entry and exit patterns tend to mirror each other. c) and d) show that occupant B's entry and exit patterns are distinguishable. It is significantly different from occupant A's pattern.

be used to determine if occupant A entered or exited the room to turn on/off any device in that room. Figure 6.3 show output voltage fluctuations for entry and exit events. We observe the shape of the entry and exit events shape tend mirror to each other and have an opposing skewed tail, indicating a sense of direction associated with the events. As the person enters the room, they do not obscure majority of the surface area until they reach the door frame plane which is orthogonal to the solar cell surface and continues blocking the light as they move away. However, for the exit event, voltage begins to decrease earlier than the person reaches the door frame. This happens mostly due to the brighter source of the light coming from inside the room. Typically rooms are brighter than hallways because of multiple light sources. This particular voltage pattern phenomena is a good indicator to determine from which direction the person crossed the solar cell.

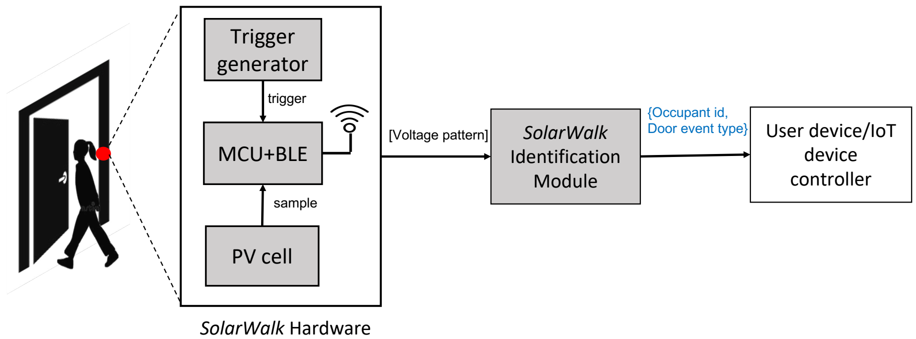


Figure 6.4: Overview of *SolarWalk* design. The photovoltaic harvester's output voltage attached to an indoor light energy-harvesting sensor fluctuates differently as different occupants of a home passes by. *SolarWalk* leverages this voltage fluctuations as an unique attribute to differentiate occupants.

Motivated by these observations, in this paper, we aim to design the proposed system named *SolarWalk*, that can identify persons using tiny, non-invasive solar cells.

6.2 System Design

6.2.1 Overview of *SolarWalk*

SolarWalk identifies occupants in smart homes by analyzing their associated distinct voltage patterns, reflected on a solar energy harvester as they walk in close proximity. *SolarWalk* design consists of two major components: *SolarWalk* hardware and *SolarWalk* identification module as shown in Figure 6.4. *SolarWalk* hardware records voltage traces from the PV cell as the event of interests occur and the identification module employs a pre-trained machine learning classifier trained from the data collected in the same physical environment.

The hardware module consists of an external trigger generator that notifies a microcontroller of a possible walking event. The microcontroller starts recording the door event until it finishes. Once the voltage trace is recorded, the MCU communicates the data over BLE to the *SolarWalk* identification module. *SolarWalk* identification module determines the identity label of who the door event is associated with and what type of door event it is (i.e., entry or exit from the room). We train the identification module with historic data

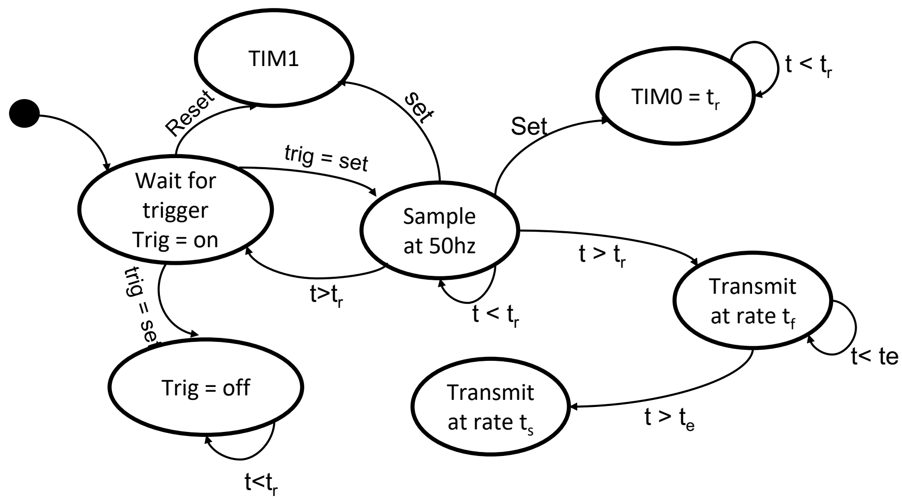


Figure 6.5: State machine representation of *SolarWalk* device's workflow

containing both walk and no-walk events. During the training phase, the identification module relies on labeled voltage data with an occupant identifier and the type of event.

SolarWalk elevates the capability of an ordinary photovoltaic harvester by introducing the concept of meaningful power supply fluctuation. With *SolarWalk* we envision that, existing battery-less devices could be repurposed to do more than their usual sensing and these sensors could be crowdsourced to enable zone-specific data-fusioning.

6.2.2 *SolarWalk* Hardware

SolarWalk relies on the mobility of an occupant to record how someone's shadow pattern impacts the voltage generation. However, continuous sampling of solar cell voltage at the required frequency is energy-expensive, even when carefully duty-cycled. *SolarWalk* overcomes this challenge by incorporating an external trigger sensor to initiate voltage sampling. Since entering and exiting through a door in a home are not high-frequency events, the average energy-overhead can be kept significantly low.

Figure 6.5 shows the state machine of the software that runs on *SolarWalk* devices. The MCU waits for the trigger in low power mode with trigger enabled. Once the trigger is set ($trig = set$), the MCU starts sampling the solar cell at 50 Hz. We empirically determine the required sampling rate throughout our data collection study. The system

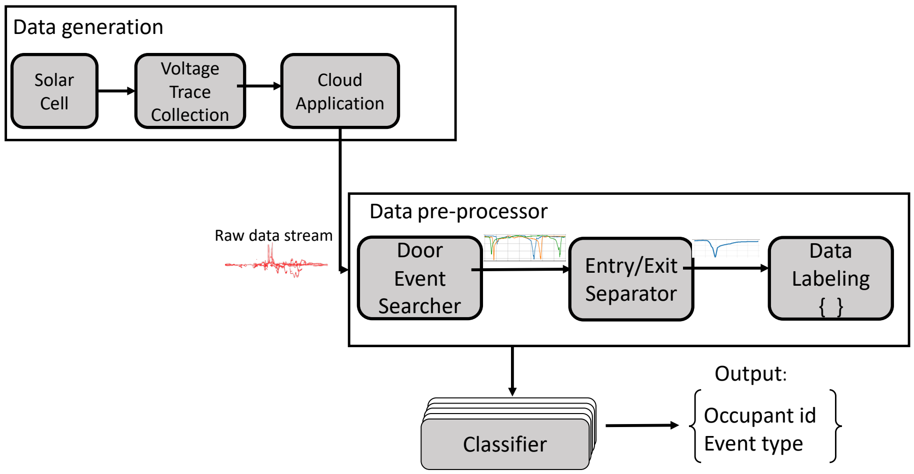


Figure 6.6: Block diagram of *SolarWalk* identification module

needs to keep recording for the entire duration of the event and sets a timer ($TIM0 = t_r$) to stop sampling. The external trigger is turned off during an ongoing sampling to prevent further triggers while the event is being recorded and turned on once sampling is finished. Upon finishing sampling, the MCU transmits the data over BLE advertisements with t_f rate. The MCU also keeps track of how long it has been passed since the last trigger happened and if it is greater than t_e (TIM1), it lowers the advertisement rate to t_s to conserve more energy.

As we discussed in Section 6.1, the shadow pattern of a person diminishes with increasing distance from the solar cell surface. Though adoption of multiple solar cells could provide us wider range, we refrain from this design choice to make *SolarWalk* hardware unobtrusively fit in indoor spaces within a reasonable form factor. Moreover, solar energy-harvesting devices usually are extremely low-power devices and a majority of them incorporate at least one PV cell.

6.2.3 *SolarWalk* Identification Module

The identification module of *SolarWalk* system runs on a gateway receiver or an edge device and collects data from the hardware to perform the identification process using supervised learning techniques. We train the identification module with historic data using KNN supervised machine learning technique.

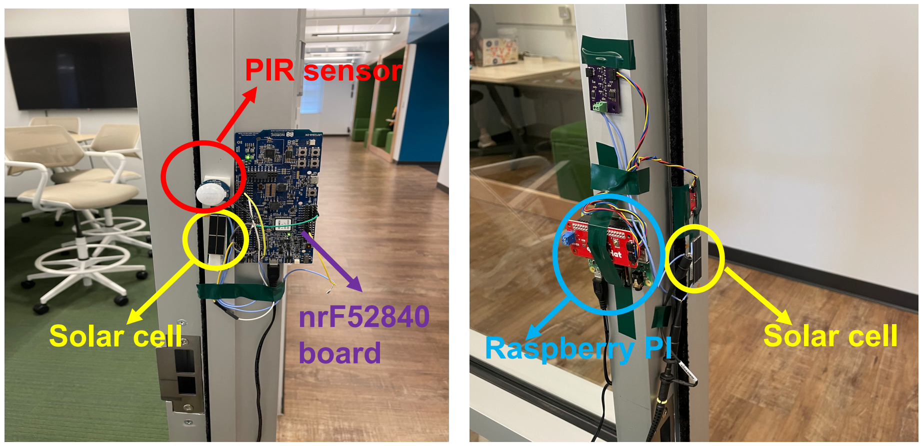
Figure 6.6 shows an overview of *SolarWalk*'s identification module's training phase. The training phase consists of major blocks: data generation, data pre-processor, and classifier. The data generation block accommodates a voltage trace collection module connected to a solar cell and sends the data over a cloud application for pre-processing. To be able to differentiate between steady state voltage fluctuations and actual events of interest, the system trains the classifier with data samples containing both walk and no-walk events. The data generation block generates a robust dataset that captures the output voltage profile of the solar cell throughout different times of the day. The data pre-processor block receives a stream of data containing door entry and exit events for multiple occupants. In the pre-processing phase, the system separates door walk events from no-walks events. This process, however, is not required in the deployment phase since the identification module only receives walk events from *SolarWalk*'s hardware device. It also filters and labels entry and exit traces with user-provided label. From the time series data of door events, the pre-processor labels each occupant's entry and exit sequence. The entry and exit sequence of voltage samples are then fed into the classifier along with the occupant id label. The classifier outputs the result in terms of occupant label and the type of event.

6.3 Implementation

In this section, we discuss the implementation of *SolarWalk* hardware and the data collection platform in the training phase of *SolarWalk* identification module.

***SolarWalk* hardware prototype.** We use a PIR sensor as the trigger generator to detect movement in the doorway. We incorporate a Panasonic AMN41121 [111] which can detect movement within 5m range with a 50° horizontal angle field of detection. We run *SolarWalk* software in the nRF52840 development kit [112]. The development kit accommodates a Cortex M4 processor SoC with BLE 5 radio. The MCU is connected with a IXYS SMLD121H04L monocrystalline solar cell with a 22% efficiency. The solar cell is optimized to be used for both indoor and outdoor applications. The rate open circuit voltage is 2.52V with a short circuit current of 50 mA. The dimension of the solar cell is 43x14 mm. Figure 6.7(a) depicts the hardware prototype implementation.

At runtime, the MCU samples the solar cell at a 50hz rate using one of the internal



(a) *SolarWalk* hardware prototype **(b)** Data collection hardware platform

Figure 6.7: *SolarWalk* prototype implementation

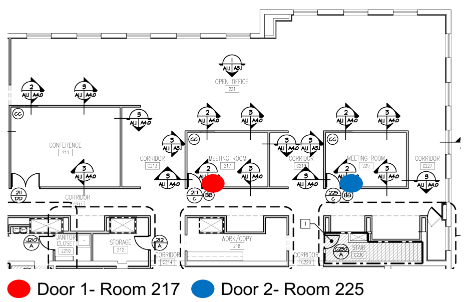


Figure 6.8: Floor plan showing the installed sensors on two doors of two different rooms.

ADC channels for $t_r = 6$ s. We set this value to capture the whole entry or exit event. We determine this value to be the maximum duration of any door event by analyzing the data collected during the training phase.

Data collection module. During the training phase, we adopt a data acquisition platform [113]consisting of a Raspberry Pi model 3A+. It connects a custom breakout board containing an ADS1015 analog to digital converter and to a Sparfun breakout board containing a VEML6030 illuminance sensor over I2C interface. This platform is configured to sample open-circuit voltage of a IXYS SMLD121H04L monocrystalline solar cell at a rate of 50 samples per second and also to record illuminance readings as a baseline for the data acquisition conditions. The ADS1015 gain stage was configured to 8, resulting in a full-scale resolution of ±4.096 Volts, and a 2 millivolt least significant bit size. The platform records and streams data using MQTT protocol to a cloud-hosted database, so we can later use the recorded data to train and evaluate our classifier models. Figure 6.7(b) shows the set up of the data collection module.

6.4 Evaluation

To evaluate *SolarWalk*, our goal is to answer how accurately the solar cell voltage trace performs as an attribute to identify occupants in a 5-person household. We base our experiments on real-world study to evaluate the performance of *SolarWalk* identification system. We explore how the identification accuracy is impacted by i) different systems parameter: the number of occupants, different classification methods, ii) environmental parameters: doors from different rooms, different times of a day, iii) physical attributes: different occupants and their heights. Another interesting feature of *SolarWalk* classifier is the ability to distinguish between two types of door events: entry and exit and we analyze how accurately the system can distinguish between these events.

6.4.1 Methodology

Experimental setup: We perform our data collection study by installing the *SolarWalk* data collection platform on two different doors of two middle rooms in our lab building. The width of both doorways is three feet. Figure 6.8 shows the floor plan including the installation points. Figure 6.7(b) depicts one of the setups. We install the device halfway above the floor on the doorframe to cover an optimal range of occupant height. The lower the position of the solar cell, the more likely the shadow of a person is going to impact the voltage. However, since solar energy-harvesting sensors usually should be placed as close as possible to the light source, we chose the midway to be the optimum point for deployment. We also deploy a working *SolarWalk* hardware on one of the doors to demonstrate the functionality and proof-of-concept implementation of the design(Figure 6.7(a)).

Data collection procedure: Our study involves five different occupants from different body shape in terms of height and girth. We collected 900 door entry and exit events from five participants as they walked through the door. Four participants walk a 100 times through each of the doors and one participants walk 100 times through one door. We

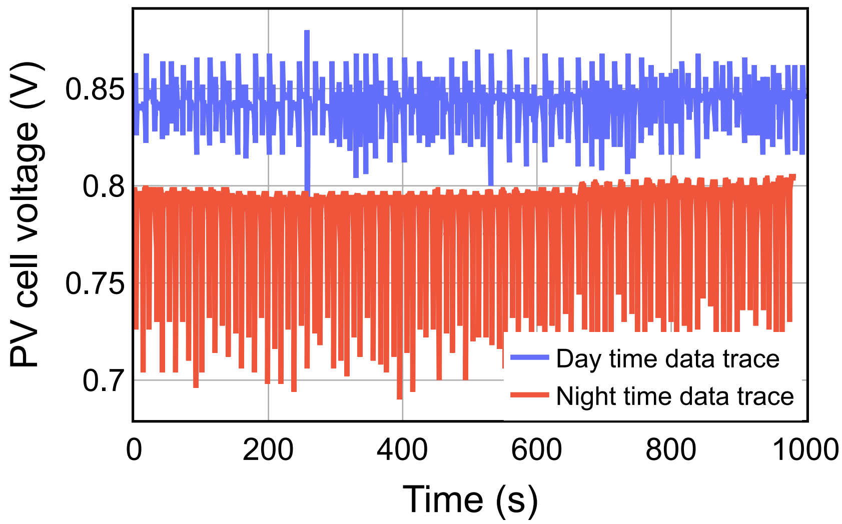
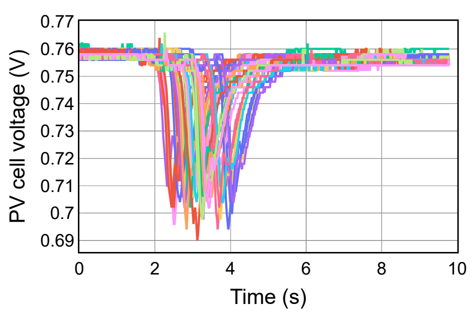
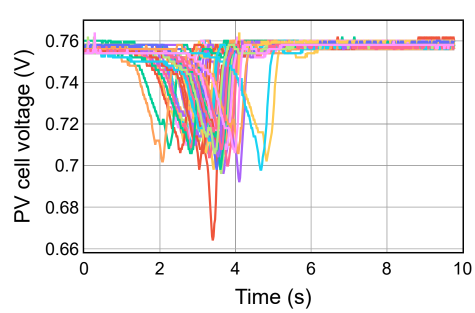


Figure 6.9: Voltage trace of a participant during day and night time. Open circuit voltage of solar cell changes throughout the day and can have impact on model performance. *SolarWalk* dataset includes traces from both day and night.



(a) Solar voltage trace of 50 room entry events of a single participant.



(b) Solar voltage trace of 50 room exit events of a single participant.

Figure 6.10: Data collection step of *SolarWalk* involves each participant walking through the door every 10 seconds. However, a noticeable change in solar cell voltage pattern is observed in the first six seconds, which contains 300 voltage samples. Thus, the dimension of the input feature of our machine learning model is 1×300.

collected 50 room entry samples and 50 room exit samples. Each walk spans ten seconds.

We performed the data collection throughout different hours of the day including both day and night time to build a robust dataset, since the shadow pattern and the open circuit voltage of the solar cell is expected to change throughout the day. Figure 6.9 illustrates the solar trace of one participant's walk event during day time and night time. Each trace in this figure consists of 100 events (room entry or room exit) that lasts 1,000 seconds.

Data preprocessing procedure: Once we collected room entry and exit voltage traces from participants, we analyzed each trace carefully to identify the trigger point of the solar cell. Figure 6.10(a) illustrates 50 solar cell traces of one participant's entry event. We notice that, although each event spans for 10 seconds, a noticeable change in voltage pattern happens in the first six seconds. A similar outcome can be noticed in exit events Figure 6.10(b). As such, during training and testing our machine learning models, we have taken traces from the first six seconds. As our prototype collects data at 50Hz sampling rate, a single entry event or exit event contains 6×50 samples. Thus, as an input feature our ML models take 300 voltage readings.

Machine learning models: To evaluate *SolarWalk*, we implemented three supervised classifier algorithms: K-Nearest Neighbor (KNN) classifier, Random Forest, and Decision Tree. In our evaluation, the KNN classifier contains six neighbors. On the other hand, the Random Forest classifier consists of 10 trees and uses entropy as the loss function. We performed 10-fold cross-validation while training and testing each model.

6.4.2 Overall System Performance

In this section, our goal is to evaluate how accurately the system can identify different occupants and distinguish between two different door activity. Results show that, our KNN-based classifier can accurately detect the identity of occupants on average 87% of the time in a 5-person home and on average 95% of the time in a 2-person small home. We also explore the performance of two other supervised learning method: decision tree [114], random forest [115] for comparison. Figure 6.11 the how percentage identification accuracy changes with an increasing number of occupants across different classification methods. The plot shows the distribution over 10 trials. From the result, we find that the percentage of accuracy drops from 99% for one occupant to 88% for five occupants, denoting a 13% point decrease. This represents that the solar cell shadow feature is a new accurate physical attribute for the occupant identification for homes with less than 5

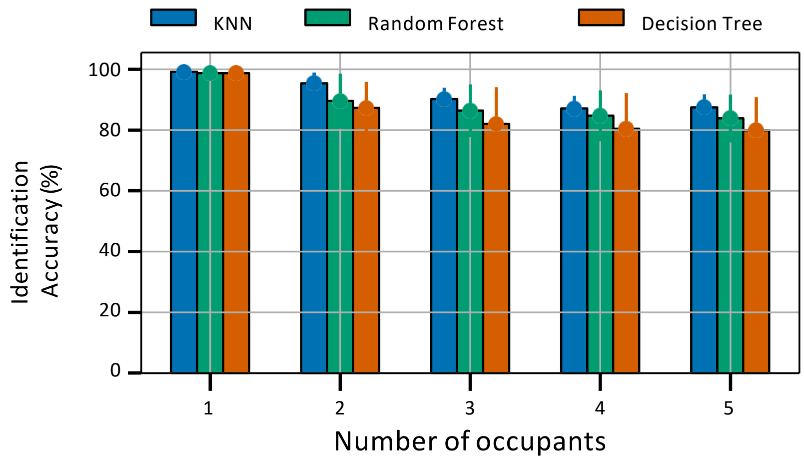


Figure 6.11: The figure shows that the occupant identification accuracy continues to drop as we increase the number of occupants. With five occupants *SolarWalk*'s KKN classifier achieves 88% accuracy.

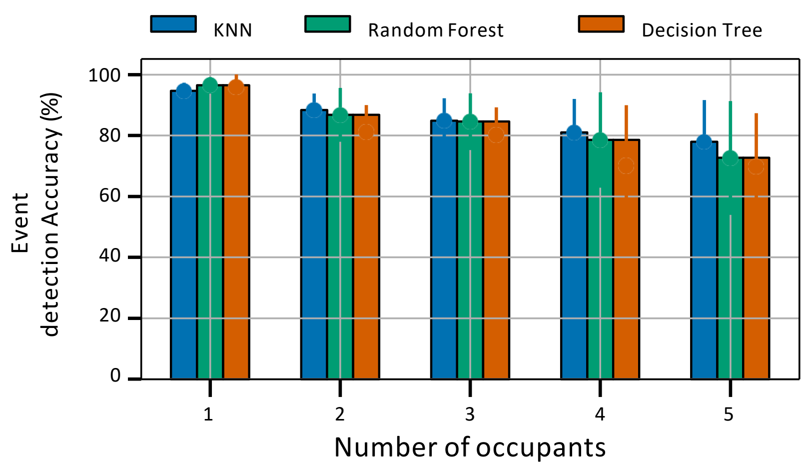
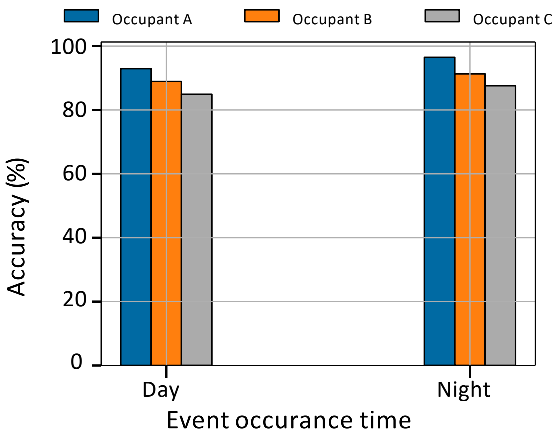


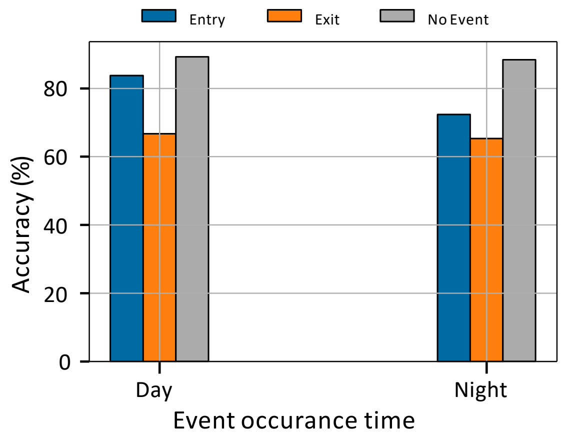
Figure 6.12: Here, we show how the type of event detection accuracy changes with increasing number of occupants. *SolarWalk* classifier can on average accurately identify between door entry and exit events 77% of time.

people. However, as the demographic increases, the system might fail to perform acceptably and more robust learning techniques i.e., reinforcement learning is needed for high occupancy spaces such as offices or classrooms.

To determine if *SolarWalk* can differentiate between a door entry and exit event, we measure the event detection accuracy as we vary the number of occupants. Figure 6.12



(a) Occupant identification accuracy



(b) Type of event detection accuracy

Figure 6.13: These plots show the effect of different times in a day on the system's accuracy. Since the steady state voltage of the solar cell undergoes variation due to different illuminance levels throughout the day, the voltage pattern's DC component shifts. Yet, system performance stays similar with a slightly higher accuracy for night events.

shows that, on average *SolarWalk* can correctly differentiate between entry and exit events with a probability of .77 for five people. For two persons, it can detect events with an accuracy of 88%.

6.4.3 Environmental Effect

Since a photovoltaic's energy conversion efficiency is dependent on a number of factors including the spectrum of exposed light and illuminance of the surface, its open circuit voltage varies throughout different indoor spaces and hours of the day. Therefore, the shadow pattern of a person is different in multiple doors. However, it should still preserve characteristics to be distinguishable from another person. In this section, we explore how *SolarWalk* performs during the day vs night and the performance among two doors.

In Figure 6.13 we show the impact of different hours of the days has on the identification accuracy of three participants and event detection accuracy. For all three occupants, we evaluate the results from the data collected during two different time durations of the day. The day time voltage readings are collected within 12 pm to 4 pm and night time readings are collected after 8 pm. From Figure 6.13(a) we find that, for all three partic-

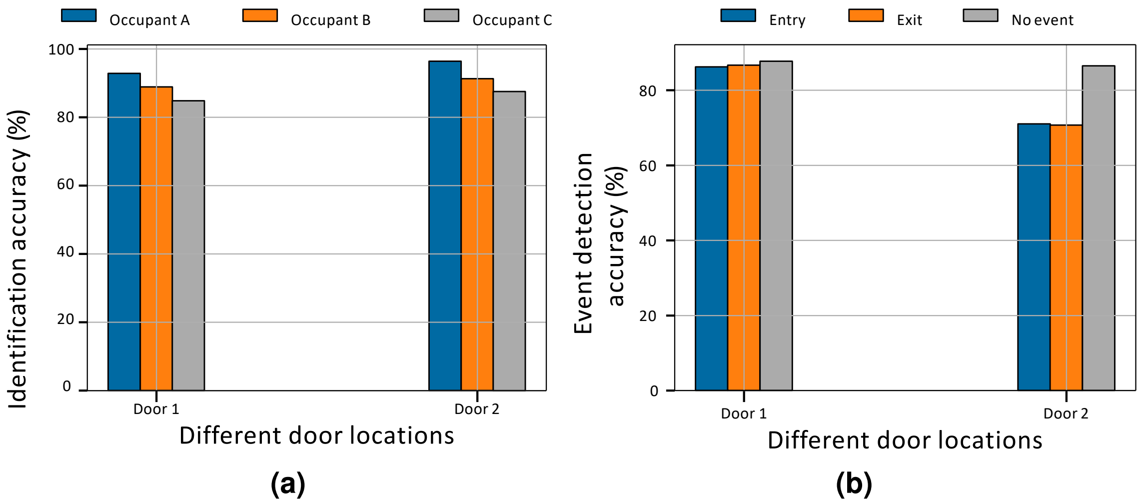


Figure 6.14: a) *SolarWalk*'s identification accuracy remains similar for multiple occupants over two deployment locations. b) Event detection accuracy achieved at different doors. The accuracy of Door 2 is at least 15% lower for entry and exit events than Door 1. For the direction of movement, location seems to play an important role.

ipants, the identification accuracy at night time is higher than day time by 3.5%, 2.3%, and 2.7%. This slight difference happens due to indoor spaces getting illuminated by natural night during day time. Therefore, someone's shadow makes the surface of the solar cell less illuminated during night than day and results in a larger voltage drop. We verify this observation from the attached illuminance sensor in the data collection module. The event detection accuracy stays similar for the exit and no event scenario as shown in Figure 6.13(b), but entry event accuracy drops by 11.4%. This could happen, since while entering through the door, as opposed to, exiting to the hallway, the brighter illuminance of the room light plays a role to distort the shape of the pattern. To summarize, event detection accuracy is affected more by the brightness variation throughout the day than occupant identification.

Figure 6.14(a) shows if the classifier performs in a similar manner in terms of occupant identification accuracy on two different doors. The identification accuracy for different occupants stays within a difference of 3.6% between the doors. However, the accuracy of individual occupants drops from overall single occupant identification accuracy because we only consider data from one location for this scenario. Figure 6.14(b) shows the event accuracy. We find that Door 1 achieves higher accuracy than Door 2 for

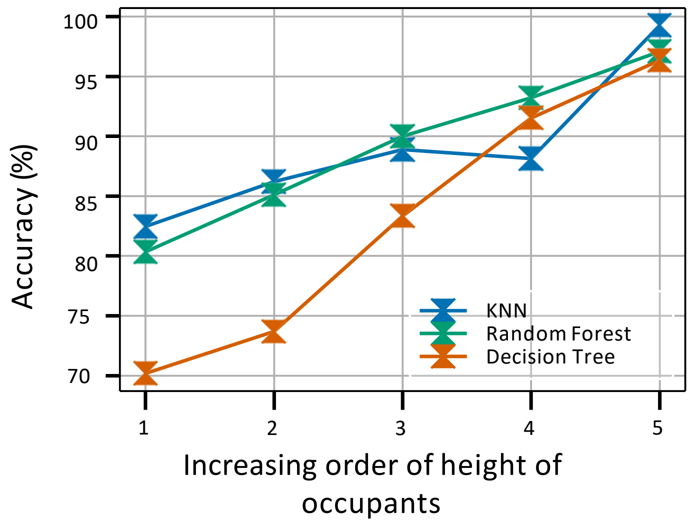


Figure 6.15: This figure plots the identification accuracy of the model with increasing occupant height. Occupant's height plays as an important factor for the system's identification accuracy. The taller height produces more distinguishable shadow pattern.

all event types, which denotes that deployment location matters more for event detection type. Since the pattern of entry and exit events tend to change more depending on the position of source of light.

6.4.4 Sensitivity to Physical Attributes

A individual's body shape features plays role in the shadow formation [116], [117]. In order to understand how *SolarWalk* performs across individual occupants, we analyze the system's occupant identification accuracy and event detection accuracy for each participants.

Figure 6.15 shows individual occupant's identification accuracy in the increasing order of height. Identification accuracy improves with increasing height, which is expected. The maximum and minimum identification accuracy is 99% and 82% respectively. Figure 6.16 shows event detection accuracy for different occupants. We find that occupant E achieves a maximum of 100% and 85.7% accuracy for entry and exit event detection respectively. All of the participants achieve an entry accuracy of more than 80%, however, exit events see less accuracy. This matches our findings from previous sections.

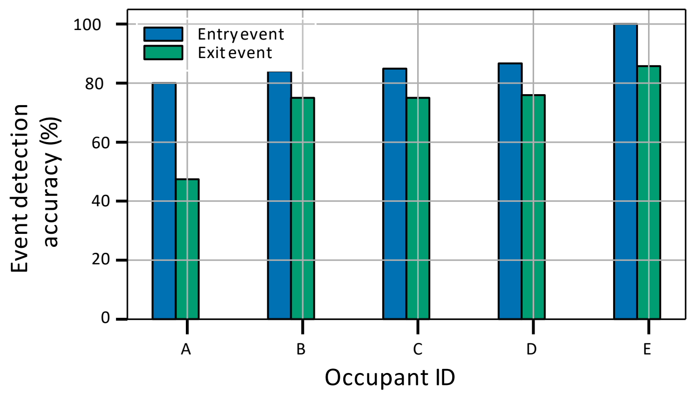


Figure 6.16: Event detection accuracy of an individual participant. Entry events are likely to be detected more accurately than exit events.

6.5 Discussions

In this section, we highlight some remaining challenges and potential research directions.

Larger demographic and deployment conditions. Since *SolarWalk* relies on the shadow feature of a person, if two people in a home have similar body shape, the system might fail to distinguish them. A natural following step to build upon our initial results and make our technique more robust is to collect data representing a wider range of demographic and deployment spaces. For instance, recruiting participants with different and similar body sizes and likeness and with different gait patterns could support a more comprehensive evaluation of the proposed approach, since it would be based on a larger demographic. Though, we argue that *SolarWalk* enables accurate person identification in an average-size smart home, a possible dimension to explore is deploying the system at locations with a wider range of illumination conditions as to cover a larger set of possible real-world scenarios. For instance, evaluating the system with different combinations of natural and artificial light sources could better capture specific deployment conditions and increase the deployability of the system more than on just doors.

Potential applications. As we demonstrated in this work, a single and brief voltage time series recording from a solar cell can potentially carry enough information to classify occupants with reasonable accuracy, what can be even more valuable for smart building applications is to create ”dynamic” spaces according to the occupant’s preferences and needs. We can imagine *SolarWalk* to incorporate multiple solar-powered sensors present in a space and the spatio-temporal correlation between these sensors' solar cell voltage readings can provide even richer information, not only allowing better occupant classification but also enabling other potential applications such as activity recognition, or monitor the zones inside a shopping complex or museum to analyze which items get more attention or track complex usage patterns of smart building spaces. These envisioned applications also come with a number of challenges, for instance, deciding optimal strategies to process and exchange information between energy-constrained battery-less sensors, and the development of machine learning models that provide best results given the sensors' limited energy and computation capability.

Incorporating new occupants. Currently, *SolarWalk* does not incorporate any policy to handle data from unknown users. However, a realistic scenario would be able to update the model if the occupant situation in a home changes over time. In this case, online learning-based techniques such as reinforcement learning can be adopted to increase the robustness of the system.

Data labeling. An important challenge to using the supervised learning technique as the ones used in our work is the need to label the data with ground truths. While controlled experiments can be used to collect labeled data, they are time-consuming and needed to be repeated for each set of conditions (e.g if the illumination source changes). One possible alternative approach to collect labeled data is to use an user's interaction with another smart device in the room. For instance, if an occupant walks into a room and their cellphone connects to a voice assistant device, the system can use the logged id to label the data previously collected by the *SolarWalk* device to the respective occupant.

6.6 Conclusion

Future smart buildings will be a lot more personalized, greener, and full a of large network of nearly-invisible devices. To enable such a vision, one crucial step is to design systems that are aware of their contextual cues, yet simple, unobtrusive, and, installation-friendly. As a forward step, in this chapter, we introduce *SolarWalk* to enable occupant-specific personalized control by sensing the voltage perturbance of photovoltaic energy-harvester. *SolarWalk* not only demonstrates a novel and accurate non-invasive, infrastructure-free occupant identification system, but also introduces the concept of empowering the power sources of battery-less energy-harvesting applications with meaningful contextual data. We believe innovation flourishes more rapidly when systems build on existing resources, which would otherwise just be wasted. We envisage this work would enable more interesting applications in the field of smart building research.

Chapter 7

Conclusion and Future Directions

Ubiquitous computers in the form of sensors have played a crucial role to collect important data from their surroundings enabling many revolutionary applications in wearables, smart health, industrial monitoring, autonomous vehicles, smart cities, smart agriculture, wildlife preservation and many others. Sensors deployed in smart farming increase crop production over manual farming, sensors embedded in wearables combat a global pandemic by predicting early onset and preventing disease spread, sensors worn by endangered animals protect and preserve wildlife—all of these sensors push for sustainable lives on the planet earth. It is quite intelligible to predict that the upward scale of computing will continue to expand and sensors will solve more difficult problems posing humanity as advancements in hardware and software accelerate.

However, like any other technology, the IoT has its caveats too as the massive global scale production of devices eventually become an Internet of Trash. One major source of these e-waste are short-lived batteries that power majority of small IoT devices in the use today. Another impending challenge is forced device obsolescence as a result of fast technology progress, which exacerbates the e-waste problem. In this dissertation, we specifically focus on these challenges to enable long-term sensing with long-lived sensors. Our approach ditches batteries for energy-harvesting power supplies that can support computation perpetually which leads to a net-zero carbon footprint over the sensor's end-to-end lifecycle. However, the adoption of energy-harvesting sensors is currently very limited in the commercial IoT space due to two inherent challenges: design overhead due to an unreliable power supply and unreliable sensing services due to an intermittent energy source. ALTAIR's energy supervisor exposes useful APIs to the application providing enough abstraction as well as options for visibility and control over the energy optimization. To reduce the design overhead of energy-harvesting sensors, we designed ALTAIR which relieves an IoT application developer off complex energy management by abstracting power management and hardware-level power supply interactions. This architecture also accommodates an online energy optimization algorithm that adapts to the changing energy availability after deployment instead of being constrained to design-time training. With PreFarad, we attempted to design a class of accurate intermittent sensors for IoT event detection. PreFarad sensors achieve improved accuracy to detect stochastic events over the baseline design approach. Stochastic events are particularly challenging to detect with intermittent device availability. PreFarad solves this challenge by introducing a new design that prioritize detection over processing and event transmission. We evaluate the system with real indoor event monitoring applications like occupancy and door state sensing in real locations to demonstrate the efficacy of the solution.

ALTAIR takes a step to reduce the burden associated with managing an unpredictable and limited source of energy. However, this also brings several interesting questions: how much abstraction is useful for developers with varying level of expertise: beginner, familiar, and expert? How to find a trade-off between abstraction and visibility as applications become more complex than what ALTAIR handled? For example, an energy-harvesting tinyML sensor may perform training, inference, computation, and even offloading [118]–[120]. ALTAIR's API can be expanded to incorporate these tasks. How would the role of energy supervisor evolve with the presence of an operating system? For instance, the energy supervisor could perform better energy accounting per application and implement predictive scheduling on behalf of the OS.

An intermittent sensor's data generation capacity is limited by the rate of its power cycles. On the other hand, many edge applications are now becoming increasingly data-driven. PreFarad improved the availability of intermittent sensors significantly, but still performs less accurate than a battery-powered device. While incorporating multiple intermittent sensors may help further improve the availability and accuracy, and achieve continuous sensing, several trade-offs need to be carefully understood. More devices increase network collisions resulting in high packet drops or duplicate results, eventually wasting system energy. Furthermore, scaling deployment increases the consumption of unnecessary electronics. How can we achieve a target frequency of data by desynchronizing the power cycles of a network of sensors? How do we determine the target frequency of the sensed data to begin with, as this impacts the accuracy of the overall application? How do we ensure the optimal number of sensors while maximizing data frequency and minimizing simultaneous wakeups? Another future direction is exploring how data-driven edge applications can be achieved with reasonable latency when data packets are intermittent. Many edge-based applications collect and coordinate data from multiple devices [121], [122] as well as multimodal sensors [123], [124] for activity monitoring, human tracking, occupancy prediction where the latency of inference is critical. The unpredictable and compromised packet rate of intermittent sensor would compromise the performance in such applications.

To mitigate the worsening impact of computing on the environment as a result of premature discontinuation of services, this dissertation emphasizes on empowering already existing computing resources to introduce added utility to the systems though additional sensing services. RETROIOT demonstrates how we can exploit battery replacements of IoT devices to modulate new data steams into the channel and eventually upgrade to an optimized energy-harvesting powers supply that dutycycles sensor operation. Simply by replacing the battery with a smart battery or an encoder energy-harvesting power supply, the upgrade can utilize the already existing infrastructure including the gateway, cloud, and front-end app services simultaneously eliminating the need of complete replacement of the old device and possibly the entire infrastructure. Taking it a step further, *SolarWalk* demonstrates how purely software methods can expand the role of a photovoltaic harvester to perform sensing as well as energy harvesting. Instead of adding dedicated sensors or additional peripherals, *SolarWalk* performs person identification by analyzing the power supply voltage noise introduced by human shadow.

It is crucial to think about how we should design sensors today that stay up-to-date for decades passing the test of time. However, advancement in low power and low footprint hardwares and efficient software features makes device obsolescence seemingly unavoidable. One common way to future-proof devices is reprogramming new applications into the flash memory using over-the-air firmware upgrades. However, remote firmware upgrades usually require a steady power source on the reprogrammed device to ensure memory consistency. Establishing secure and successful firmware update is challenging as intermittent sensors may experience power outages multiple times for up to several minutes during a long communication involving a few kilobytes to hundreds of kilobytes. Moreover, saving program states in non-volatile memory (NVM) to resume operation after a power failure should be properly enforced by attack-proof hardware or energy-efficient software encryption mechanism. States in NVM becomes vulnerable if an attacker has physical access to the device [125]. Network-wide upgrade also becomes more complicated if sensors in a network are more heterogeneous (for instance, different applications, supported radios, memory budgets). How can we establish secure and reliable firmware upgrades with limited intermittent energy? Even with software upgrades, hardware-enforced limitations curtails a device's performance. For example, the SoCs available only a few years ago are now obsolete [126] because the BLE radio and software stack do not have the advanced features (Direction finding, range improvement, concurrent peripherals) like the newer SoCs [112] have. If an application requires the advanced features, the complete sensor needs to be replaced with the upgraded SoCs. Newer MCUs also have more memory, better CPU, and multiple cores to handle concurrency with lower power consumption and reduced footprint. Moreover, as hardwares become more integrated like SoCs, SiPs, the harder it becomes to replace a specific component. One promising research direction is to embrace reconfigurable hardware architectures like FPGAs and processor architectures like RISC-V for edge applications powered by harvested energy.

While green computing has been explored in the context of large scale datacenter-based applications, the concept is still maturing in the context of IoT application domain. We require design principles both in hardware and software and meaningful metrics to ensure sustainability without compromising service quality. We need to rethink materials, hardware designs, network protocols, cloud services to ensure a complete stack of Design-for-Environment (DfE). How can we design materials, components, and hardware that reduces the overall carbon starting from the production throughout the lifetime to after the end-of-life? How to ensure we achieve carbon neutrality or possibly carbon negativity for a specific deployment? How should we write software, application interfaces, and gateway applications to process more data in the edge or close to the edge instead of offloading to high carbon footprint cloud servers? A holistic approach combining research effort from different domains in hardware, power electronics, software engineering, operating systems, networking, and cloud computing is essential to investigate how we can realize the entire stack to manage devices at scale and maintain longevity.

To conclude, with the total number of IoT devices already exceeding the population of earth a few years ago and billions of more devices expected to market in the next decade, we must rethink the way we design and develop applications and systems to prevent device obsolescence for a long-lived IoT. This requires combined research effort informed and enforced by sustainable components across a range of disciplines including hardware design, software engineering, and network protocols. This dissertation takes several crucial steps toward that effort.

References

[1] P. Sparks, "The route to a trillion devices," *White Paper, ARM*, 2017.

[2] IoT Analytics, https://iot-analytics.com/number-connected-iot-devices/.

[3] J. Hester and J. Sorber, "The future of sensing is batteryless, intermittent, and awesome," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, 2017, pp. 1–6.

[4] K. A. Severson, P. M. Attia, N. Jin, *et al.*, "Data-driven prediction of battery cycle life before capacity degradation," *Nature Energy*, vol. 4, no. 5, pp. 383–391, 2019.

[5] D. H. P. Kang, M. Chen, and O. A. Ogunseitan, "Potential environmental and human health impacts of rechargeable lithium batteries in electronic waste," *Environmental science & technology*, vol. 47, no. 10, pp. 5495–5503, 2013.

[6] S&P Global, Lithium supply is set to triple by 2025. Will it be enough? https://www.spglobal.com/en/research-insights/articles/lithium-supply-is-set-to-triple-by-2025-will-it-be-enough.

[7] *The global e-waste monitor 2020*, https://ewastemonitor.info/wp-content/uploads/2020/11/GEM_2020_def_july1_low.pdf, 2020.

[8] IEEE Spectrum, *The internet of trash: Iot has a looming e-waste problem*, https://spectrum.ieee.org/the-internet-of-trash-iot-has-a-looming-ewaste-problem.

[9] D. Ma, G. Lan, M. Hassan, W. Hu, and S. K. Das, "Sensing, computing, and communications for energy harvesting iots: A survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1222–1250, 2019.

[10] P. Martin, Z. Charbiwala, and M. Srivastava, "Doubledip: Leveraging thermoelectric harvesting for low power monitoring of sporadic water use," in *Proceedings of the 10th ACM conference on embedded network sensor systems*, 2012, pp. 225–238.

[11] J. Madden, G. Marcano, S. Taylor, P. Pannuto, and C. Josephson, "Hardware to enable large-scale deployment and observation of soil microbial fuel cells," 2022.

[12] M. Gorlatova, J. Sarik, G. Grebla, M. Cong, I. Kymissis, and G. Zussman, "Movers and shakers: Kinetic energy harvesting for the internet of things," in *The 2014 ACM international conference on Measurement and modeling of computer systems*, 2014, pp. 407–419.

[13] M. Buettner, B. Greenstein, and D. Wetherall, "Dewdrop: An energy-aware runtime for computational rfid," in *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'11, Boston, MA: USENIX Association, 2011, pp. 197–210.

[14] A. Colin, E. Ruppel, and B. Lucia, "A reconfigurable energy storage architecture for energy-harvesting devices," *SIGPLAN Not.*, vol. 53, no. 2, pp. 767–781, Mar. 2018.

[15] N. Saoda and B. Campbell, "No batteries needed: Providing physical context with energy-harvesting beacons," in *Proceedings of the 7th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems*, 2019, pp. 15–21.

[16] Panasonic Battery, *AM1454*, `https://www.mouser.com/datasheet/2/315/panasonic_AM-1454CA-1197022.pdf`, 2021.

[17] M. Thielen, L. Sigrist, M. Magno, C. Hierold, and L. Benini, "Human body heat for powering wearable devices: From thermal energy to application," *Energy conversion and management*, vol. 131, pp. 44–54, 2017.

[18] M. S. Wong, Y. T. C. Kwok, H. Li, K. W. Hui, and J. Sun, "Tree monitoring system for urban tree management," 2021.

[19] D. Vasisht, Z. Kapetanovic, J. Won, *et al.*, "Farmbeats: An iot platform for data-driven agriculture.," in *NSDI*, vol. 17, 2017, pp. 515–529.

[20] M. Manikkampatti Palanisamy, V. R. Myneni, B. Gudeta, and S. Komarabathina, "Toxic metal recovery from waste printed circuit boards: A review of advanced approaches for sustainable treatment methodology," *Advances in Materials Science and Engineering*, vol. 2022, 2022.

[21] United States Environmental Protection Agency (EPA), *Basic information about electronics stewardship*, https://www.epa.gov/smm-electronics/basic-information-about-electronics-stewardship, 2023.

[22] U. G. S. A. (GSA), https://www.gsa.gov/governmentwide-initiatives/federal-highperformance-green-buildings/resource-library/integrative-strategies/ewaste-and-electronic-stewardship, 2022.

[23] H. Xue, W. Jiang, C. Miao, *et al.*, "Deepmv: Multi-view deep learning for device-free human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–26, 2020.

[24] V. Radu and M. Henne, "Vision2sensor: Knowledge transfer across sensing modalities for human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 3, pp. 1–21, 2019.

[25] A. Das, E. S. Kolvig-Raun, F. C. Sangogboye, and M. B. Kjærgaard, "Occu-track: Occupant presence sensing and trajectory detection using non-intrusive sensors in buildings," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 512–524.

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

[27] M. Afanasov, N. A. Bhatti, D. Campagna, *et al.*, "Battery-less zero-maintenance embedded sensing at the mithræum of circus maximus," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 368–381.

[28] B. Campbell, M. Clark, S. DeBruin, *et al.*, "Perpetual sensing for the built environment," *IEEE Pervasive Computing*, vol. 15, no. 4, pp. 45–55, 2016.

[29] L. Yerva, B. Campbell, A. Bansal, T. Schmid, and P. Dutta, "Grafting energy-harvesting leaves onto the sensornet tree," in *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, 2012, pp. 197–208.

[30] S. DeBruin, B. Campbell, and P. Dutta, "Monjolo: An energy-harvesting energy meter architecture," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, 2013, pp. 1–14.

[31] J. Hester, L. Sitanayah, and J. Sorber, "Tragedy of the coulombs: Federating energy storage for tiny, intermittently-powered sensors," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, 2015, pp. 5–16.

[32] J. Hester and J. Sorber, "Flicker: Rapid prototyping for the batteryless internet-of-things," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, 2017, pp. 1–13.

[33] N. Jackson, J. Adkins, and P. Dutta, "Capacity over capacitance for reliable energy harvesting sensors," in *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, 2019, pp. 193–204.

[34] B. Campbell, B. Ghena, and P. Dutta, "Energy-harvesting thermoelectric sensing for unobtrusive water and appliance metering," in *Proceedings of the 2nd International Workshop on Energy Neutral Sensing Systems*, 2014, pp. 7–12.

[35] J. Adkins, B. Ghena, N. Jackson, *et al.*, "The signpost platform for city-scale sensing," in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, IEEE, 2018, pp. 188–199.

[36] A. Colin and B. Lucia, "Chain: Tasks and channels for reliable intermittent programs," in *Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications*, ser. OOPSLA 2016, Amsterdam, Netherlands: Association for Computing Machinery, 2016, pp. 514–530, ISBN: 9781450344449.

[37] B. Ransford, J. Sorber, and K. Fu, "Mementos: System support for long-running computation on rfid-scale devices," ser. ASPLOS XVI, Newport Beach, California, USA: Association for Computing Machinery, 2011, pp. 159–170, ISBN: 9781450302661.

[38] J. Hester, K. Storer, and J. Sorber, "Timely execution on intermittently powered batteryless sensors," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '17, Delft, Netherlands: Association for Computing Machinery, 2017, ISBN: 9781450354592.

[39] A. Colin, E. Ruppel, and B. Lucia, "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*, 2018, pp. 767–781.

[40] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z.-L. Zhang, "Leakage-aware energy synchronization for wireless sensor networks," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, 2009, pp. 319–332.

[41] R. C. Hsu, C.-T. Liu, and W.-M. Lee, "Reinforcement learning-based dynamic power management for energy harvesting wireless sensor network," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Springer, 2009, pp. 399–408.

[42] R. C. Hsu, C.-T. Liu, and H.-L. Wang, "A reinforcement learning-based tod provisioning dynamic power management for sustainable operation of energy harvesting wireless sensor node," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 2, pp. 181–191, 2014.

[43] Y. Rioual, Y. Le Moullec, J. Laurent, M. I. Khan, and J.-P. Diguet, "Reward function evaluation in a reinforcement learning approach for energy management," in *2018 16th Biennial Baltic Electronics Conference (BEC)*, IEEE, 2018, pp. 1–4.

[44] A. Ortiz, H. Al-Shatri, X. Li, T. Weber, and A. Klein, "Reinforcement learning for energy harvesting decode-and-forward two-hop communications," *IEEE Transactions on green communications and networking*, vol. 1, no. 3, pp. 309–319, 2017.

[45] S. Shresthamali, M. Kondo, and H. Nakamura, "Adaptive power management in solar energy harvesting sensor node using reinforcement learning," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 5s, pp. 1–21, 2017.

[46] Y. Luo and S. Nirjon, "Smarton: Just-in-time active event detection on energy harvesting systems," in *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, IEEE, 2021, pp. 35–44.

[47] F. Fraternali, B. Balaji, D. Sengupta, D. Hong, and R. K. Gupta, "Ember: Energy management of batteryless event detection sensors with deep reinforcement learning," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 503–516.

[48] W. D. Stephenson, *Strategic Retrofitting: Older Machines Get an IIoT Update*, `https://www.industryweek.com/technology-and-iiot/article/21125857/strategic-retrofitting-older-machines-get-an-iiot-update`, 2019.

[49] Bosch-Connectivity, *3 Examples of How to Retrofit IoT Sensor Devices*, `https://www.bosch-connectivity.com/newsroom/blog/3-examples-of-how-to-retrofit-iot-sensor-devices/`, 2020.

[50] B. V. Guerreiro, R. G. Lins, J. Sun, and R. Schmitt, "Definition of smart retrofitting: First steps for a company to deploy aspects of industry 4.0," in *Advances in Manufacturing*, Springer, 2018, pp. 161–170.

[51] T. Lins and R. A. R. Oliveira, "Cyber-physical production systems retrofitting in context of industry 4.0," *Computers & industrial engineering*, vol. 139, p. 106 193, 2020.

[52] C. Pérez-Penichet, F. Hermans, A. Varshney, and T. Voigt, "Augmenting iot networks with backscatter-enabled passive sensor tags," in *Proceedings of the 3rd Workshop on Hot Topics in Wireless*, 2016, pp. 23–27.

[53] J. Shi, D. Mu, and M. Sha, "Lorabee: Cross-technology communication from lora to zigbee via payload encoding," in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, IEEE, 2019, pp. 1–11.

[54] R. Ramakers, F. Anderson, T. Grossman, and G. Fitzmaurice, "Retrofab: A design tool for retrofitting physical interfaces using actuators, sensors and 3d printing," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 409–419.

[55] R. Mancini, S. Tuli, T. Cucinotta, and R. Buyya, "Igatelink: A gateway library for linking iot, edge, fog, and cloud computing environments," in *Intelligent and Cloud Computing*, Springer, 2021, pp. 11–19.

[56] Y.-S. Kuo, S. Verma, T. Schmid, and P. Dutta, “Hijacking power and bandwidth from the mobile phone’s audio interface,” in *Proceedings of the First ACM Symposium on Computing for Development*, 2010, pp. 1–10.

[57] S. Nirjon, R. F. Dickerson, Q. Li, *et al.*, “Musicalheart: A hearty way of listening to music,” in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, 2012, pp. 43–56.

[58] G. Conte, M. De Marchi, A. A. Nacci, V. Rana, D. Sciuto, *et al.*, “Bluesentinel: A first approach using ibeacon for an energy efficient occupancy detection system.,” in *BuildSys@ SenSys*, Citeseer, 2014, pp. 11–19.

[59] A. K. Das, P. H. Pathak, J. Jee, C.-N. Chuah, and P. Mohapatra, “Non-intrusive multi-modal estimation of building occupancy,” in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, 2017, pp. 1–14.

[60] N. Khalil, D. Benhaddou, O. Gnawali, and J. Subhlok, “Nonintrusive occupant identification by sensing body shape and movement,” in *Proceedings of the 3rd ACM international conference on systems for energy-efficient built environments*, 2016, pp. 1–10.

[61] H. Rajabi, Z. Hu, X. Ding, S. Pan, W. Du, and A. Cerpa, “Modes: Multi-sensor occupancy data-driven estimation system for smart buildings,” in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, 2022, pp. 228–239.

[62] J. Han, S. Pan, M. K. Sinha, H. Y. Noh, P. Zhang, and P. Tague, “Sensetribute: Smart home occupant identification via fusion across on-object sensing devices,” in *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*, 2017, pp. 1–10.

[63] J. Jee, A. K. Das, P. H. Pathak, and P. Mohapatra, “Motionsync: Personal energy analytics through motion tags and wearable sensing,” in *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, 2016, pp. 65–74.

[64] J. Hester, L. Sitanayah, and J. Sorber, “Tragedy of the coulombs: Federating energy storage for tiny, intermittently-powered sensors,” in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’15,

Seoul, South Korea: Association for Computing Machinery, 2015, pp. 5–16, ISBN: 9781450336314.

[65] L. Yerva, B. Campbell, A. Bansal, T. Schmid, and P. Dutta, “Grafting energy-harvesting leaves onto the sensornet tree,” in *2012 ACM/IEEE 11th International Conference on Information Processing in Sensor Networks (IPSN)*, 2012, pp. 197–207.

[66] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z.-L. Zhang, “Leakage-aware energy synchronization for wireless sensor networks,” in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’09, Kraków, Poland: Association for Computing Machinery, 2009, pp. 319–332, ISBN: 9781605585666.

[67] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, “Power management in energy harvesting sensor networks,” *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, 32–es, Sep. 2007.

[68] B. Buchli, F. Sutton, J. Beutel, and L. Thiele, “Dynamic power management for long-term energy neutral operation of solar energy harvesting systems,” in *Proceedings of the 12th ACM conference on embedded network sensor systems*, 2014, pp. 31–45.

[69] B. Lucia, V. Balaji, A. Colin, K. Maeng, and E. Ruppel, “Intermittent computing: Challenges and opportunities,” in *2nd Summit on Advances in Programming Languages (SNAPL 2017)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[70] N. Saoda, M. F. R. M. Billah, and B. Campbell, “Designing a general purpose development platform for energy-harvesting applications,” in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 581–583.

[71] STMicroelectronics, *stm32wb10cc*, https://www.st.com/resource/en/datasheet/stm32wb10cc.pdf.

[72] Texas Instruments, *CC2640*, https://www.ti.com/product/CC2640#design-development.

[73] Arm technologies, https://www.arm.com/technologies/big-little.

[74] F. Ait Aoudia, M. Gautier, and O. Berder, "Rlman: An energy manager based on reinforcement learning for energy harvesting wireless sensor networks," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 2, pp. 408–417, 2018.

[75] R. C. Hsu, C. Liu, and H. Wang, "A reinforcement learning-based tod provisioning dynamic power management for sustainable operation of energy harvesting wireless sensor node," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 2, pp. 181–191, 2014.

[76] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018, ISBN: 0262039249.

[77] J. Adkins, B. Campbell, S. DeBruin, *et al.*, "Demo: Michigan's iot toolkit," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15, Seoul, South Korea: Association for Computing Machinery, 2015, pp. 485–486, ISBN: 9781450336314.

[78] A. Sarris, "Lpcsb: A device to distinguish between natural and artificial light(afrikaans text)," AAI0662099, M.S. thesis, 2020.

[79] N. Saoda and B. Campbell, "No batteries needed: Providing physical context with energy-harvesting beacons," in *Proceedings of the 7th International Workshop on Energy Harvesting amp; Energy-Neutral Sensing Systems*, ser. ENSsys'19, New York, NY, USA: Association for Computing Machinery, 2019, pp. 15–21, ISBN: 9781450370103.

[80] Nordic Semiconductor, *Nordic Thingy:52*, https://www.nordicsemi.com/Software-and-tools/Prototyping-platforms/Nordic-Thingy-52#infotabs, 2018.

[81] BLEHome, http://www.blehome.com/sensorbug.htm, 2017.

[82] B. Ransford, J. Sorber, and K. Fu, "Mementos: System support for long-running computation on rfid-scale devices," in *Proceedings of the sixteenth international conference on Architectural support for programming languages and operating systems*, 2011, pp. 159–170.

[83] *Panasonic PIR motion sensors*, https://api.pim.na.industrial.panasonic.com/file_stream/main/fileversion/250456.

[84] Silicon Labs, *Si720x Switch/Latch Hall Effect Magnetic Position Sensor*, https://www.silabs.com/documents/public/data-sheets/si720x-data-sheet.pdf, 2020.

[85] BZT-52, *Surface mount silicon zener diode*, https://www.panjit.com.tw/upload/datasheet/BZT52-B2V4-AU_SERIES.pdf, 2017.

[86] Maxim Integrated, *0.27 μA CURRENT CONSUMPTION VOLTAGE DETECTOR*, https://www.ablic.com/en/doc/datasheet/voltage_detector/S1009_E.pdf, 2017.

[87] Nordic Semiconductor, *nRF52832 Product Specification*, https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.4.pdf.

[88] Samsung ARTIK 530/530s Module, https://www.mouser.com/pdfdocs/530530sds.pdf, 2017.

[89] Dragino, *LoraWan Door Sensor*, https://www.dragino.com/downloads/downloads/LoRa_End_Node/LDS01/Datasheet_LDS01_Door_Sensor.pdf, 2021.

[90] Decentlab, *Soil moisture and temperature profile sensor*, https://cdn.decentlab.com/download/datasheets/Decentlab-DL-SMTP-datasheet.pdf, 2018.

[91] HT1 Temperature and Humidity Sensor, *SensorPush*, https://www.sensorpush.com/products/p/ht1.

[92] Texas Instruments, *TPS78401*, https://www.ti.com/store/ti/en/p/product/?p=TPS78401DBVR, 2020.

[93] Texas Instruments, *LP358*, https://www.ti.com/product/LP358, 2020.

[94] Texas Instruments, *LP2980*, https://www.ti.com/product/LP2980-N, 2020.

[95] Maxim Integrated, *DS4432*, https://datasheets.maximintegrated.com/en/ds/DS4432.pdf, 2008.

[96] Monolithic Power, *MPQ28164*, https://www.monolithicpower.com/en/mpq28164.html, 2018.

[97] N. Saoda, W. Wang, M. F. R. M. Billah, and B. Campbell, "An energy supervisor architecture for energy-harvesting applications," in *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, IEEE, 2022, pp. 323–336.

[98] STMicroelectronics, *SPV1050*, https://www.st.com/en/power-management/spv1050.html, 2018.

[99] Maxim Integrated, *MAX17222*, https://datasheets.maximintegrated.com/en/ds/MAX17220-MAX17225.pdf, 2020.

[100] STMicroelectronics, *STM32L010R8*, https://www.st.com/resource/en/datasheet/stm32l010r8.pdf, 2019.

[101] Maxim Integrated, *MAX9634*, https://datasheets.maximintegrated.com/en/ds/MAX9634.pdf, 2020.

[102] STMicroelectronics, *LoraWan Discovery Kit*, https://www.st.com/resource/en/data_brief/b-l072z-lrwan1.pdf, 2019.

[103] TE Connectivity, *MS5837-30BA*, https://www.te.com/, 2019.

[104] Analog Devices, *TMP37*, https://www.analog.com/en/products/tmp37.html, 2015.

[105] The Things Network, https://www.thethingsnetwork.org/.

[106] Seeed Studio, *SenseCAP Wireless CO2 Sensor*, https://www.seeedstudio.com/LoRaWAN-CO2-Sensor-US915-p-4308.html, 2018.

[107] Desk Occupancy Sensor, *EnOcean*, https://www.enocean.com/en/product/occupancy-sensor-sub-desk-eosd/.

[108] Magnet Contact Sensor, *EnOcean*, https://www.enocean.com/en/product/magnet-contact-sensor-emcs-oem/.

[109] IXOLAR High Efficiency SolarMD., *IXYS*, https://ixapps.ixys.com/DataSheet/SLMD121H04L_Nov16.pdf, 2016.

[110] Panasonic Amorphous Solar cell, https://www.endrich.com/sixcms/media.php/2308/Catalogue20Sanyo20Amorton.490768.pdf.

[111] PIR motion sensor, *Panasonic Electronics*, https://www.mouser.com/datasheet/2/315/panasonic_amn1_2_4-1196943.pdf.

[112] Nordic Semiconductor, *nrF52840 Dev Kit*, https://www.nordicsemi.com/Products/Development-hardware/nrf52840-dk, 2019.

[113] V. A. L. Sobral, J. Lach, J. L. Goodall, and B. Campbell, "Thermal energy harvesting profiles in residential settings," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 520–523.

[114] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[115] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[116] M. Shinzaki, Y. Iwashita, R. Kurazume, and K. Ogawara, "Gait-based person identification method using shadow biometrics for robustness to changes in the walking direction," in *2015 IEEE Winter Conference on Applications of Computer Vision*, IEEE, 2015, pp. 670–677.

[117] K. Prabin, "Theory of shadow," *Himalayan Physics*, vol. 1, pp. 101–105, 2010.

[118] M. Giordano, P. Mayer, and M. Magno, "A battery-free long-range wireless smart camera for face detection," in *Proceedings of the 8th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, 2020, pp. 29–35.

[119] S. Lee, B. Islam, Y. Luo, and S. Nirjon, "Intermittent learning: On-device machine learning on intermittently powered system," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 4, pp. 1–30, 2019.

[120] M. Nardello, H. Desai, D. Brunelli, and B. Lucia, "Camaroptera: A batteryless long-range remote visual sensing system," in *Proceedings of the 7th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems*, 2019, pp. 8–14.

[121] M. F. R. M. Billah, N. Saoda, J. Gao, and B. Campbell, "Ble can see: A reinforcement learning approach for rf-based indoor occupancy detection," in *Proceedings of the 20th international conference on information processing in sensor networks (co-located with CPS-IoT week 2021)*, 2021, pp. 132–147.

[122] T. Routh, N. Saoda, M. F. R. M. Billah, and B. Campbell, "Low cost light source identification in real world settings," in *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, IEEE, 2022, pp. 226–234.

[123] X. Ouyang, X. Shuai, J. Zhou, *et al.*, "Cosmo: Contrastive fusion learning with small data for multimodal human activity recognition," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 2022, pp. 324–337.

[124] T. Li, J. Huang, E. Risinger, and D. Ganesan, "Low-latency speculative inference on distributed multi-modal data streams," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, 2021, pp. 67–80.

[125] A. S. Krishnan and P. Schaumont, "Exploiting security vulnerabilities in intermittent computing," in *Security, Privacy, and Applied Cryptography Engineering: 8th International Conference, SPACE 2018, Kanpur, India, December 15-19, 2018, Proceedings 8*, Springer, 2018, pp. 104–124.

[126] Nordic Semiconductor, *nRF51822 Product Specification*, `https://infocenter.nordicsemi.com/pdf/nRF51822_PS_v3.1.pdf`.