Research Statement

Josiah D. Hester
School of Computing
Clemson University

My research enables longterm sustainable sensing on tiny, energy harvesting, batteryless computing platforms. Batteries are the single greatest threat to the vision of a sustainable Internet of Things. They are expensive, bulky, hazardous, and wear out after a few years (even rechargeables). Replacing and disposing of billions or trillions of dead batteries per year would be expensive and irresponsible. By leaving the batteries behind and surviving off energy harvested from the environment, tiny intermittently powered computers can monitor objects in hard to reach places **maintenance free for decades.** Batteryless sensing will revolutionize computing and open up new application domains from infrastructure monitoring and wildlife tracking to wearables and healthcare.

Batteryless sensing is fundamentally different from conventional computing. These devices challenge one of the most basic assumptions of computing—a stable power supply. Energy harvesting is inconsistent and energy storage is scarce, so batteryless sensors *will* lose power intermittently. Programmers must figure out how to tie together fragmented execution opportunities to get useful work done *each time a sensor is resurrected.*

Research Goals: My goal is to make sophisticated batteryless sensing practical and usable by system designers across application domains. I explore and develop new hardware designs, software techniques, tools, and programming abstractions so that developers can easily design, debug, and deploy intricate batteryless applications that work in spite of frequent power failures.

Research Methods: I enjoy designing and deploying real systems, whenever possible, in order to keep my experiments both relevant and realistic. Batteryless computing does not have good datasets or standard evaluation tools. Indeed, developing evaluation tools for batteryless sensors research is an integral part of my dissertation work. Building real systems and tools—including software and hardware artifacts—and evaluating them with real data and deployments, form the core of my research. Each of these deployment experiences, prototypes, and datasets provide real world evidence of the efficacy of my research; and can be used by others to improve and test their science.

Themes: I focus on fundamental questions of *sustainable computing*, and *energy efficiency*. I believe that my work will enable sustainable, responsible computing. My intention is to **lay the groundwork for the future of sensing.** In this future devices are small, cheap, and are useful for the entire lifetime of the things they monitor. These devices will be easy to program, straightforward to test, and will be deployed confidently. I intend to push the boundaries of sensing, with a focus on relieving the cognitive burden on programmers to the point where even amateurs can confidently program, test, and deploy, batteryless sensors.

Languages, Platforms, and Tools for Batteryless Sensing

Batteryless, energy harvesting sensing presents critical challenges not encountered by tethered or battery-powered sensing. Firstly, batteryless devices operate in unpredictable environments, where voltages vary and power failures can occur at any time—often devices are in failure for hours. Secondly, a device's behavior effects the amount of energy they can harvest—meaning small changes in tasks can drastically change

harvester efficiency. Thirdly, the programming interfaces of batteryless device are ill-defined and non-intuitive; most developers have trouble anticipating the problems inherent with an intermittent power supply.

My work on batteryless sensing has touched on multiple layers of the system stack to address these problems. Specifically my dissertation work 1) developed tools for repeatable experimentation with batteryless computers, 2) designed a hardware platform to simplify task scheduling and protect against failures, and 3) created a language and runtime that captures the temporal attributes of sensor data on intermittently executing sensing platforms. Each of these pieces seek to simplify the development process; enabling more robust, long lived computation, on sustainable, energy harvesting sensors.

1. Evaluating and testing batteryless sensors is incredibly difficult. Without a repeatable way to compare different algorithms, hardware platforms, and energy harvesters, batteryless sensing would have no rigorous scientific backing. I developed the Ekho emulator **[Best paper, SenSys'14]** to fill this toolchain gap. Ekho allows developers to record, and then playback energy harvesting environments in a realistic and repeatable way. This realism comes from the observation that the behavior of a batteryless device has a measurable effect on the amount of energy that can be harvested. Behavioral changes of a sensor device—going from low power mode to broadcasting on the radio, sensing the temperature, or simple processing—cause the voltage on the harvester to change, which alters the harvesting efficiency. No other tool captured this, because of the assumption that all sensing was performed on battery powered devices with a stable supply voltage and large reserves of energy. Using Ekho, developers could finally compare different configurations realistically inside the lab.

2. Conventional ideas about sensing hardware are detrimental to batteryless sensors. Untethered sensing devices generally rely on a *single power supply*, a battery or large capacitor. When designing batteryless sensors that are powered by harvested energy, this conventional approach to energy storage results in devices that charge slowly, are less available, and fail frequently—often in the middle of a task. This happens because of the unique physical characteristics of small energy storage— 1) the larger the capacitor the longer it takes to charge to a usable voltage 2) different hardware peripherals (such as radios, sensors, actuators) have different usable voltage ranges and 3) tasks are coupled to a single energy store, meaning any one peripheral could starve the rest, including the processor. This tragedy of the commons (or coulombs) necessitated a different approach to energy storage. Based on these observations, I looked into federating the energy supply. I showed that by using multiple independently charged capacitors that were each assigned to a single peripheral developers could eliminate the weakness of the commons approach. I introduced the concept of Federated Energy, developed a hardware prototype, and then deployed the prototype in a greenhouse monitoring application **[SenSys'15].** Devices equipped with Federated Energy failed less, completed more tasks, were more available for user tasks, and even harvested more energy than the conventional centralized approach.

3. Prevailing programming models don't translate well to the intermittent computing model. Despite these advances in tooling and hardware, developers still find *programming* batteryless devices very challenging and non-intuitive. Composing programs for batteryless, intermittently powered devices requires in-depth knowledge of hardware behaviors. Worse, the idea that a power failure can happen between any two lines of code causes a logic crises. These problems demand developers spend most of their time protecting against failures, instead of focusing on sensing goals, defining tasks, and generating data.
The Mayfly Language and Runtime[1] is a response to this intuition gap. The Mayfly language is a graph based, declarative programming model that allows developers to focus on application goals, outline timing requirements for data collection, and list task dependancies. The Mayfly runtime takes these tasks and

[1] Paper in submission co-authored by K. Storer, and J. Sorber.

constraints, and creates a schedule at compile time that can be executed on the sensor node. This scheduler manages timekeeping across power failures, and uses Federated Energy to handle tasks energy requirements. Devices programmed with Mayfly will be deployed on RFID powered WISPs, in a greenhouse monitoring application, and on a batteryless wearable platform. I also conducted a user study that revealed that participants had trouble visualizing program execution and keeping sensor data from expiring when programming in traditional Embedded-C, and were able to reason about tasks better with Mayfly, accomplishing more in the same amount of time, and having a more enjoyable experience.

Future Research Directions

My dissertation work has laid the groundwork for what is a rich and exciting new field of study. My long term research goal is to continue to develop the ideas around sustainable computing and sensing, advancing the vision of the Internet-of-Things for the benefit of science and society. In each of the following future directions, building and deploying real systems for emerging applications will be the focus—and the main form of establishing scientific validity. The following research questions are core to the future of sustainable, batteryless sensing and computing, each has significant and exciting unsolved problems.

What new hardware, software, and infrastructure can help batteryless sensing go mainstream? The batteryless sensing toolchain has matured to the point that we have simulation, emulation, and some debugging tools. However, the cognitive burden of designing and testing applications that effectively handle intermittent power is still high. The toolchain lacks definitive tools such as testbeds, network simulators, and general hardware platforms. Execution on batteryless devices depends on random environmental factors, causing operation to be probabilistic. In spite of this, developers need to better understand how the code they write will execute in deployment. I envision two tools that can facilitate user understanding and confidence in deployment. First, the idea of energy aware code coverage—meaning developers can automate the generation of energy environments that will exercise the full program functionality. Second, the idea of a hardware debugging tool that completely reproduces the energy environment, data traces, and allows non-invasive step debugging and tracing of batteryless sensors. Additionally I see the development of generalized hardware platforms with built in Ekho support as fundamental to success. I plan on developing this platform, and back it with a web based infrastructure that brings together thousands of energy harvesting environments recorded by researchers in the field using Ekho, shared across institutions and research labs.

What new applications would benefit from sustainable, batteryless sensing? Deploying new and exciting applications for batteryless computing keeps my research relevant and grounded. I plan on expanding the realms of battery-free sensing past RFID applications, to include wearables, occupancy sensing, wildlife tracking, greenhouse monitoring, and infrastructure support. These new applications necessitate collaboration with domain scientists in other fields such as health and biology. I believe this will create a positive dialog about the future of sensing, and further inform my work. In the short term I plan to adapt my work on the Amulet: an mHealth wearable platform **[SenSys'16]** to support battery free operation for health and fitness applications like sleep monitoring, and step counting.

In summary, battery-free sensing promises to revolutionize science and society. My dissertation work has only scratched the surface of this deeply challenging, deeply interesting new field. My future research will continue to push the boundaries of sensing, and attempt to tackle some of the most pressing problems of the Internet-of-Things. My work on responsible, sustainable sensing has the potential for positive impacts and collaborations with many areas, including health services and patient care, commercial and consumer applications, wildlife conservation, industrial and infrastructure management and monitoring, and many other fields. I am excited to start building this future of sustainable sensing.