Ekho: Bridging the Gap Between Simulation and Reality in Tiny Energy-Harvesting Sensors

Hong Zhang, Mastooreh Salajegheh, Kevin Fu, Jacob Sorber

Dept. of Computer Science
University of Massachusetts Amherst
{zhangh,negin,kevinfu}@cs.umass.edu

Institute for Security, Tech., and Society
Dartmouth College
jacob.m.sorber@dartmouth.edu

ABSTRACT

Harvested energy makes long-term maintenance-free sensor deployments possible; however, as devices shrink in order to accommodate new applications, tightening energy budgets and increasing power supply volatility leaves system designers poorly equipped to predict how their devices will behave when deployed.

This paper describes the design and initial FPGA-based implementation of Ekho, a tool that records and emulates energy harvesting conditions, in order to support realistic and repeatable testing and experimentation. Ekho uses the abstraction of I-V curves—curves that describe harvesting current with respect to supply voltage—to accurately represent harvesting conditions, and supports a range of harvesting technologies. An early prototype emulates I-V curves with 0.1mA accuracy, and responds in $4.4\mu s$ to changes in energy conditions.

1. INTRODUCTION

Energy harvesting is key to the success of many ubiquitous sensing applications. By relying on energy harvested from a variety of sources—solar [10,8], kinetic [11], RF [13,1], and even other devices [9]—low power sensing devices collect valuable data perpetually, while requiring little or no maintenance.

Designing and deploying energy-harvesting sensors is challenging since harvested energy is highly variable, often scarce, and unpredictable. Conditions at runtime are often vastly different from those in the lab.

As devices shrink to accommodate new applications, volatility increases. Sensor platforms as small as a few mm^3 have been built [2]; however, energy storage technologies have not kept pace. Some devices, like *compu-tational RFIDs* (CRFID) [1, 13], store only enough energy to run for a few seconds, in tiny capacitors. As energy storage shrinks, a device's supply voltage becomes more sensitive to changes in energy harvesting and device behavior. Conversely, both power consumption and power harvested depend on supply voltage. In the face of these dependencies, system designers are often unable to predict how a device will perform in the wild. Deploying sensor nodes without being able to predict the behavior of a node would result in long debugging cycles and re-deployments.

A few existing simulators predict the energy consumption [5, 15, 13] and energy harvesting [6] of low-power sensors; however, realistic simulation requires an accurate model of every hardware component in the system, which leads to simulators that are limited to a single hardware platform. Many tools [5, 15] simulate only the processor and assume a fixed supply voltage. Trace-driven simulators [6] allow designers to replay real scenarios. However, simulating the behavior of hardware under the collected traces is complicated. The simulator needs to know the characteristics and the power profile of the hardware platform.

In contrast to simulated approaches, this paper argues in favor of emulating energy harvesting conditions. An energy harvesting emulator combine's many of the benefits of both simulation and deployment—replaying harvesting traces from real environments, in order to provide repeatable testing and experimentation. Other advantages of emulation include simplicity and flexibility—hardware or software changes to the device under test do not require changes to the emulator.

Our contributions include (1) a general framework for emulating energy harvesting for low-power sensors, (2) a preliminary design of Ekho, a tool for recording and emulating energy conditions, and (3) initial accuracy and sensitivity results from an early-stage FPGA-based implementation of Ekho.

2. REPEATABLE HARVESTING

Many technologies convert environmental energy into electrical energy that can then power embedded devices.
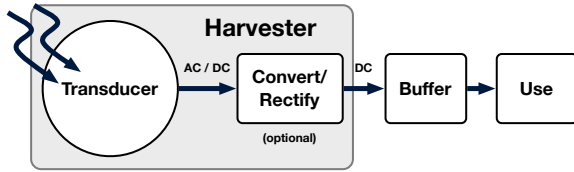
Figure 1: The general structure for an energy-harvesting device. Environmental energy is converted into either AC or DC power, which may be amplified and/or rectified before being stored in a battery or capacitor for use by the devices components.

Energy harvesting makes it possible to deploy sensors in areas where wired power is either cost-prohibitive or unavailable and where large batteries would be either too expensive or cumbersome.

Each form of environmental energy is converted to electricity differently, but all energy harvesting devices share the same basic structure (Figure 1). A transducer converts environmental energy into either AC or DC power. AC power is converted (or rectified) to DC power and voltages may need to be amplified (using a charge-pump) to a usable level. In this paper, we refer to the transducer and amplifier/rectifier stages as the *energy harvester*. Once harvested, energy is stored in a buffer—a battery or capacitor—and then used by the processor, sensors, clock, radio, and other components.

Small self-powered devices differ from traditional computing systems in several important ways. When wired power or a sufficiently large battery is available, computing devices are powered by a *voltage source*—a power supply that provides a near-constant voltage, regardless of current draw. In contrast, stable voltages are rare in RFID-scale devices. As energy buffers shrink and small capacitors replace batteries, fluctuations in system voltage are magnified, as energy harvesting conditions and power consumption vary over time.

Voltage fluctuations dramatically impact performance. System components (CPU, radio, sensors) behave differently at different voltages. When energy is scarce, a low supply voltage may render one component useless or unpredictable, while others are unaffected. Sensor accuracy and per-component power consumption may also vary with the supply voltage. Some components (like Flash memory) can be made more robust to low-voltage conditions [14], and voltage regulators can mitigate some of these effects [4,3]; however, regulators consume energy—the amount depends on the supply voltage—that could otherwise be used for computation and sensing.

Energy harvesting performance also depends on the device's supply voltage. The power delivered by any energy harvester to a battery or capacitor can be characterized by an I-V curve, as shown for a solar panel in Figure 2, that describes how the current (I) flowing from
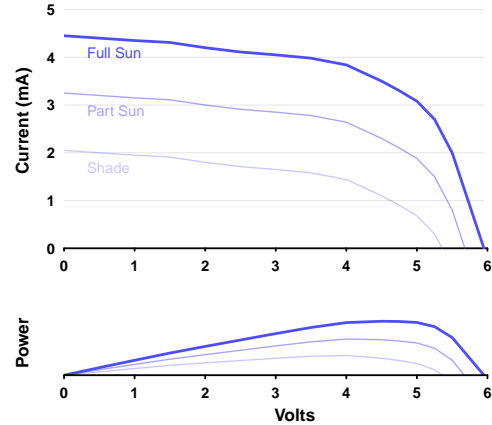

Figure 2: A solar panel I-V curve. As solar irradiance increases or decreases the shape of the curve changes. At any given time, the power harvested by the solar panel depends on both the capacitor voltage and the irradiance.

the harvester changes depending on the device's supply voltage (V). The shape of the I-V curve changes as solar irradiance (W/m^2) on the panel increases or decreases. The power (P), computed using Joule's law ($P = IV$), is also shown. Using the full-sun I-V curve from Figure 2, a device with a supply voltage of 5V will harvest energy *twice* as fast as a device charged to only 2V.

Power-supply fluctuations compounded with unpredictable variations in energy availability make designing small self-powered devices challenging. Specifically, this volatility makes it difficult for designers to predict a device performance when deployed or to compare two possible software approaches or hardware configurations.

Replaying Power: A naive approach to making energy harvesting reproducible is to measure the harvested power as the device executes, and then replay the collected power trace, in a simulator. This approach has been used in other harvester-powered mobile systems [16,13], and our initial efforts to emulate energy harvesting conditions, focused on replaying power traces.

Emulating a power trace in hardware is simple and straightforward, but only accurate for devices with a stable supply voltage. The voltage of a rechargeable battery is reasonably stable—typically varying by less than half a volt—between 15% and 85% of a full charge; however, when the battery is almost full or nearly empty, replaying harvested power will result in inaccuracies, over- or under-estimating the energy that would be harvested in a deployed scenario.

These inaccuracies are much worse for capacitor-based devices. A capacitor's voltage depends on its stored energy and its capacitance, and when small capacitors are used—like those on the UMass Moo ($10\mu F$) [17]—the voltage may change as much as 1-2V in a 100ms. The
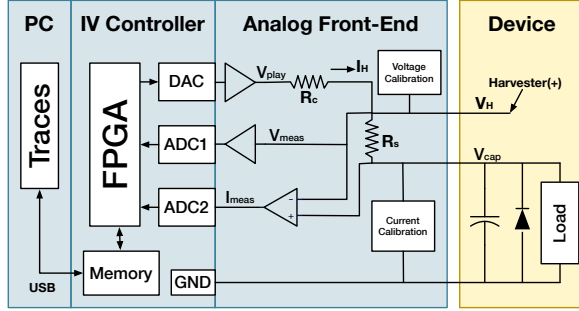
Figure 3: Ekho consists of three components: an analog front-end facilitates accurate current and voltage measurements and controllable power ouput; a high-speed I-V controller emulates an I-V curve by adjusting the harvesting current, I_H, as V_{cap} changes; and a desktop updates the I-V controller based on the recorded traces.

energy harvested by these devices, when deployed, depends heavily on *both* the energy conditions (solar irradiance or RF energy) and the device's behavior. Consequently, comparing two different programs running with a single power trace will likely produce results that differ significantly from reality.

Replaying I-V Curves: While harvested power depends on both a device's environment and it's behavior, the harvester's I-V curve depends only on the environment. In our initial measurements of solar, RF, and kinetic energy harvesters, as well as devices that parasitically harvest energy from other mobile devices [9], each energy harvester can be accurately characterized by an I-V curve. This generality and load-independence make I-V curves an ideal abstraction of energy harvesting conditions.

3. THE EKHO EMULATOR

In order to make energy harvesting reproducible when designing and testing low-power devices, we propose Ekho, a tool, that allows system designers to record and emulate energy harvesting conditions. Our design of Ekho is guided by the following goals:

Harvester independence. Instead of recording and replaying a series of power readings, Ekho represents changing energy harvesting conditions as a series of I-V curves. Unlike simulators, Ekho is not tied to any particular type of harvested energy since I-V curves are generally applicable to all common energy-harvesting technologies.

Load independence. Unlike simulators whose accuracy depends on a detailed and accurate model of a device's energy consumption, an emulator, like Ekho can be paired with any set of processors, sensors, and other circuit components. Using Ekho, system designers are free to make frequent hardware and software adjustments during development and testing without notifying Ekho.

Recording and replay accuracy. Energy harvesting conditions can experience large variations over the course of a second. Ekho should reproduce these fluctuations accurately, ideally down to a few μA. Even more important, Ekho should provide accurate comparisons between two different hardware or software configurations under the same harvesting conditions.

Responsiveness. Small capacitor-based sensor devices experience rapid voltage changes. During emulation, Ekho is designed to respond as quickly as in few μs to these changes, making Ekho applicable to sensors with extremely small energy storage (e.g. $10\mu F$).

Ekho Overview. Ekho has three functional components, shown in Figure 3. The first, an analog front-end, enables precise measurements of system voltage and harvesting current as well as accurate delivery of power. The second component (the I-V controller) records and replays the current I-V curve. Finally, temporal I-V curve changes are stored, and sent to the I-V controller by a USB-connected PC. Ekho requires little from the attached device—only an energy harvester (for collecting I-V curves), and an energy buffer (e.g. battery or capacitor) for powering microcontrollers, sensors, and any other device components that require energy. Ekho is designed to operate in two modes: *capture* and *emulate*. The following sections describe these basic operations.

3.1 Capturing harvesting conditions

In capture mode, Ekho records an energy harvesting trace, consisting of time-series of I-V curves, that can later be replayed in order to realistically compare different hardware and software approaches.

One approach to recording I-V curves is to use sensors (e.g., solar irradiance meter) and an accurate model of the harvesting technology to determine the appropriate I-V curve at each point in time. This will result in accurate harvesting traces if the sensor-to-I-V characteristics of the energy-harvesting technology is well-understood.

With Ekho, we take a more general approach to characterize I-V curves without using energy-source specific sensors. Using the design in Figure 3, we can measure a single point on the I-V curve—the capacitor voltage, V_{cap}, and the harvesting current, I_H—at any point in time. If the system knows which energy source (e.g., solar, RF) is in use, a single point may be enough to estimate the correct I-V curve. When either the energy source is unknown or higher accuracy is required, Ekho uses an optional synthetic load that dynamically changes, rapidly varying V_{cap}, in order to quickly explore as much of the I-V curve as possible.

3.2 Emulating energy environments

Replaying a captured trace with Ekho is straightforward. At any point in time, the appropriate I-V curve is sent to the I-V controller. The I-V controller measures

the capacitor voltage, $V_{cap} = V_{meas} - I_H R_s$, and determines the appropriate harvesting current, I_H, from the I-V curve stored in its memory. The correct harvesting current is produced by setting the output voltage, $V_{play} = V_{meas} + I_H R_c = V_{cap} + I_H (R_c + R_s)$. The DAC has finite precision (12 bits) and so our choice of resistor, R_c, determines the range of currents that can be replayed.

The accuracy of this approach, depends on how fast it is performed. Setting V_{play} changes the harvesting current, I_H, which combined with the load, causes the capacitor voltage V_{cap} to fluctuate. Consequently, the system's choice of V_{play} is often wrong almost immediately after it is set and must quickly be recomputed.

In order to minimize the delay between updates to V_{play}, the I-V Controller uses an FPGA instead of a low-cost microcontroller to rapidly compute updates based on voltage and current measurements.

Updates to the I-V curve do not need to be as quick as the I-V curve-based current updates (second delays are acceptable). So, these updates and other trace management functions are handled by a USB-attached PC.

3.3 Prototype

In order to verify our design, we have built an initial prototype, based on Figure 3, consisting of a PC running basic trace-management software, an FPGA-based I-V controller, and a custom analog front end. On the PC, Xilinx ISE is used to program the FPGA firmware and we have written a C# program that communicates I-V curve data to and from the I-V controller's memory.

The I-V controller core is a Digilent Spartan FPGA board, with two 14-bit ADC channels, which digitize the buffered V_{meas} and the I_{meas}, respectively. I-V computations are handled by the FPGA and I-V curve data is stored in flash memory. When emulating, the board's 12-bit DAC is used to generate an appropriate V_{play}.

The analog front end is implemented as a custom printed circuit board, which scales/shifts V_{meas} and I_H to match the input range of the ADCs (0.4-2.9V) and amplifies the DAC output to allow a full range of 0-8.25V for V_{play}. A low-tolerance sense resistor ($R_s = 5\Omega$) allows accurate current measurements, and custom circuitry has been added for precisely calibrating both voltage and current measurements in order to reduce gain and resistance errors introduced by the buffers and amplifiers.

3.4 Initial Results

Our initial experiments focus on Ekho's ability to accurately replay a single solar I-V curve (the full-sun curve in Figure 2) to a UMass Moo [17], a batteryless RFID-scale sensing device with a $10\mu F$ capacitor. In order to induce rapid variations in load current, the device periodically blinks an LED.

Accuracy: During the experiment, we measured the replay current at each voltage and compared it to the I-V curve being emulated. When the LED was on the differ-ence between the replayed current and the target current was $170\mu A$. When the LED was off, the error measured up to $70\mu A$. While this indicates a need for more precise calibration and noise elimination, it is close to our target accuracy, and is sufficient for many devices.

Sensitivity: In addition to the accuracy of our prototype, we are also interested in the sensitivity or responsiveness of the system to sudden voltage changes. To this end, we measured the time lag between changes in V_{meas} and Ekho's response in the form of an updated V_{play} to be $4.4\mu s$. For a response time of $4.4\mu s$, and a maximum current load of 10 mA, and with a ideal tuning, Ekho will be able to achieve 50 μA accuracy for devices with capacitor size of as small as $2.5\mu F$.

We also measured the time delay of individual tasks that make up Ekho's $4.4\mu s$ total latency. Converting signals from analog-to-digital (ADC) and the reverse (DAC) dominate the delay, requiring as much as $1.5\mu s$ for each conversion. Signals take over $1\mu s$ to propagate across the analog front-end, and computing V_{play} updates based on the current I-V curve takes just under $100ns$. In order to further reduce Ekho's latency, we are currently working on performing ADC and DAC conversions in parallel and finding lower-latency analog components.

4. RELATED WORK

Ekho builds on a number of related projects focused on making energy consumption and harvesting reproducible for small embedded devices.

Hardware-based power metering tools for low-power sensors [7] and storage systems [12] have made it possible to accurately characterize energy consumption patterns. Using simulation tools for low-power devices, like PowerTOSSIM [15], system designers can accurately predict energy consumption on specific platforms.

In more closely related work, Gummeson et al. [6] built a trace-driven simulator that models and predicts CRFID behavior. This simulator predicts harvesting and device behavior, but it is limited to solar-powered platforms and a single specific hardware platform.

5. FUTURE WORK & CONCLUSIONS

This paper describes an initial design and implementation of Ekho, a tool for recording and emulating I-V curve-based energy-harvesting traces. Ekho is built to ease the testing and debugging of RFID-scale embedded devices that rely on environmental energy. Accuracy and latency measurements from our initial prototype, during emulation indicate that Ekho is responsive enough to accommodate a wide range of sensor devices, though improved calibration is needed.

Moving forward, we plan to complete the implementation and evaluation of Ekho. Capturing I-V curves is still under testing and needs to be validated for multiple

harvesting technologies. Efforts are under way to improve accuracy through adding automatic calibration algorithm into I-V controller firmware, better tuning and component selection. Finally, we plan to expand our evaluation of Ekho using irradiance measurements in order to provide accurate ground truth.

Ekho promises to help system designers manage the complexity of energy harvesting and supply voltage fluctuations. As embedded sensors continue to find new applications that require small form factors and long life, we expect tools like Ekho to play a critical role.

Acknowledgments

6. REFERENCES

[1] BUETTNER, M., GREENSTEIN, B., AND WETHERALL, D. Dewdrop: An energy-aware task scheduler for computational RFID. In *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation* (2011).

[2] CHEN, G., FOJTIK, M., KIM, D., FICK, D., PARK, J., SEOK, M., CHEN, M.-T., FOO, Z., SYLVESTER, D., AND BLAAUW, D. Millimeter-scale nearly perpetual sensor system with stacked battery and solar cells. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* (2010).

[3] CHEN, G., GHAED, H., HAQUE, R. M., WIECKOWSKI, M., KIM, Y., KIM, G., FICK, D., KIM, D., SEOK, M., WISE, K., BLAAUW, D., AND SYLVESTER, D. A cubic-millimeter energy-autonomous wireless intraocular pressure monitor. *IEEE International Solid-State Circuits Conference* (2011).

[4] DOMS, I., MERKEN, P., MERTENS, R., AND VAN HOOF, C. Capacitive power-management circuit for micropower thermoelectric generators with a 2.1μW controller. *IEEE International Solid-State Circuits Conference* (2008).

[5] ERIKSSON, J., DUNKELS, A., FINNE, N., ÖSTERLIND, F., AND VOIGT, T. MSPsim—an extensible simulator for MSP430-equipped sensor boards. In *Proceedings of the 4th European Conference on Wireless Sensor Networks (EWSN), Poster/Demo session* (2007), Springer.

[6] GUMMESON, J., CLARK, S. S., FU, K., AND GANESAN, D. On the limits of effective hybrid micro-energy harvesting on mobile CRFID sensors. In *Proceedings of the 8th international conference on Mobile systems, applications, and services* (2010), MobiSys, ACM.

[7] JIANG, X., DUTTA, P., CULLER, D., AND STOICA, I. Micro power meter for energy monitoring of wireless sensor networks at scale. *Proceedings of the 6th International Conference on Information Processing in Sensor Networks* (2007).

[8] KANSAL, A., HSU, J., ZAHEDI, S., AND SRIVASTAVA, M. B. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems* (2006).

[9] KUO, Y.-S., VERMA, S., SCHMID, T., AND DUTTA, P. Hijacking power and bandwidth from the mobile phone's audio interface. In *Proceedings of the 1st Annual Symposium on Computing for Development (DEV)* (2010).

[10] LIN, K., HSU, J., ZAHEDI, S., LEE, D. C., FRIEDMAN, J., KANSAL, A., RAGHUNATHAN, V., AND SRIVASTAVA, M. B. Heliomote: Enabling long-lived sensor networks through solar energy harvesting. In *Proceedings of ACM Sensys* (2005).

[11] MENINGER, S., MUR-MIRANDA, J. O., AMIRTHARAJAH, R., CHANDRAKASAN, A., AND LANG, J. H. Vibration-to-electric energy conversion. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems 9*, 1 (2001).

[12] PEEK, D., AND FLINN, J. Drive-thru: Fast, accurate evaluation of storage power management. *USENIX 2005 Annual Technical Conference*.

[13] RANSFORD, B., SORBER, J., AND FU, K. Mementos: System support for long-running computation on RFID-scale devices. In *Proceedings of the 16th Intl. Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)* (2011).

[14] SALAJEGHEH, M., WANG, Y., FU, K., JIANG, A. A., AND LEARNED-MILLER, E. Exploiting half-wits: Smarter storage for low-power devices. In *Proc. 9th USENIX Conference on File and Storage Technologies (FAST)* (2011).

[15] SHNAYDER, V., HEMPSTEAD, M., CHEN, B.-R., AND WELSH, M. PowerTOSSIM: Efficient power simulation for tinyos applications. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)* (2004).

[16] SORBER, J., KOSTADINOV, A., GARBER, M., BRENNAN, M., CORNER, M. D., AND BERGER, E. D. Eon: A Language and Runtime System for Perpetual Systems. In *Proc. ACM SenSys* (2007).

[17] ZHANG, H., GUMMESON, J., RANSFORD, B., AND FU, K. Moo: A batteryless computational RFID and sensing platform. Tech. Rep. UM-CS-2011-020, UMass Amherst Department of Computer Science, June 2011.