# Making energy-harvesting computers reliable

October 28 2016, by Krista Burns



Credit: Carnegie Mellon University Electrical and Computer Engineering

A revolutionary and emerging class of energy-harvesting computer systems require neither a battery nor a power outlet to operate, instead operating by harvesting energy from their environment. While radio waves, solar energy, heat, and vibrations have the ability to power devices, harvested energy sources are weak leading to an "intermittent execution", with periodic power failures and unreliable behavior.

Brandon Lucia, an assistant professor of electrical and computer engineering at Carnegie Mellon University, and his Ph.D. student Alexei Colin created the first [programming language](#) designed to build reliable software for intermittent, energy-harvesting computers. Colin will present the work at the [2016 SPLASH conference](#) in Amsterdam, Netherlands, on November 3<sup>rd</sup>.

"Energy is not always available in the environment for a device to harvest," explains Lucia. "Intermittent operation makes it difficult to build applications because existing software programming languages—and programmers themselves—assume that energy is a continuously available resource."

The innovative new programming language, called [Chain](#), asks an application developer to define a set of computational tasks that compute and exchange data through a novel way of manipulating the computer's memory, called a channel. Chain guarantees that tasks execute correctly despite arbitrary power failures.

"When power is not continuously available, power failures disrupt the software's execution, often leading to unrecoverable errors," says Lucia. "Chain solves this problem by requiring [computational tasks](#) in the program to use a novel channel-based memory abstraction that ensures tasks complete without error."

Channel-based memory is the key to Chain's ability to avoid software errors: regardless of when power fails, channels ensure that a computational task always has an intact version of the data it needs when power resumes. Restarting a Chain program after a failure has virtually zero time cost because Chain does not rely on an expensive, conventional approach, like memory checkpointing. The extreme scarcity of energy makes efficient restarting essential for energy-harvesting applications including IoT devices and implantable or ingestible medical devices.

"Chain provides important reliability guarantees in a familiar and flexible programming interface that is well-positioned to be the foundation for today's and future energy-harvesting applications," says Lucia.

Lucia, Colin, and Dr. Alanson Sample, a collaborator at Disney Research Pittsburgh, worked together to push Chain into real-world deployment; early next year, in cooperation with nano-satellite company KickSat, software written in Chain will run on-board two tiny, postage stamp-sized satellites in a low-earth orbit of Earth. Once in orbit, these satellites will use tiny solar panels to harvest solar energy, powering them to collect and process sensor data and send information back to earth. While satellites are typically powered by solar energy, these satellites will be the first with the strong software correctness guarantees furnished by Chain ensuring continuous, reliable operation.

"The potential benefit of reliable energy-harvesting computer systems is far-reaching," says Lucia. "Small satellites are proliferating and the space industry itself is expanding. If we can guarantee that even tiny, energy-harvesting satellites operate without interruption, we can make it easier to conduct other scientific research in space. Further out, we may even see future applications like extraterrestrial natural resource discovery relying on this technology."

  **More information:** Learn more about Professor Lucia's research group here.

Read the researcher's paper "Chain: Tasks and Channels for Reliable Intermittent Programs" here.

Provided by Carnegie Mellon University Electrical and Computer

Engineering