1. Please describe the research you've developed so far.

Our project wants to address energy efficiency in the Internet of Things (IoT) through rigorous use of standby. Our initial focus is on low-power resource-constrained embedded microcontrollers, which are simpler architectures and also constitute most of the IoT. Our approach is to provide transparent standby mechanisms at the programming language level in order to scale to all applications.

The bulk of our research is grounded on the design of Céu [1], a new reactive programming language, which I have been working for the past 8 years. The language relies on the synchronous concurrency model, which trades power for reliability and has a simpler model of time that suits most requirements of IoT applications. In this model, all reactions to the external world are guaranteed to be computed in bounded time, ensuring that applications always reach an idle state amenable to standby mode.

Since the grant notification at the end of last year, our immediate goal was to reach, as fast as possible, a working prototype of the core research idea. We worked towards a new version of Céu that would provide automatic standby for all applications:

A programmer writes an IoT application unaware of standby or energy constraints.

The language puts the IoT device to sleep automatically whenever possible in the most efficient standby mode.

In parallel, we worked towards a formal semantics of Céu and a mathematical proof that all valid programs react to an external event in bounded time, with bounded memory, and deterministic behavior. This proof is an important result because it shows that applications always reach an idle state amenable to standby. The formalization of Céu already appeared on my PhD thesis [2], but the proofs for the properties are new results from this year.

By the end of February was the submission deadline for LCTES'18, the main annual venue that links programming languages and embedded systems. Fortunately, we had enough time to submit a full paper with the proofs [3] and a work-in-progress paper describing our approach for transparent standby at the language level [4], both of which were accepted to the conference.

Between March and May, we invested most of the time on engineering efforts on the prototype. We refactored and optimized the language implementation and also worked on new device drivers and microcontrollers. In particular, we built an RF (radio) module driver in Céu from scratch, which also requires an SPI driver. A reliable software infrastructure for radio communication is fundamental for the development of IoT applications.

In the last months, with all basic pieces set, we started working on full IoT applications. At this moment, we have more concrete evaluation scenarios that show significant energy savings due to standby in the microcontroller and other peripherals.

We are also working on an educational programming environment based on Céu targeting early undergraduate students. The idea is to introduce physical computing and IoT early in the computer science curriculum. Our hypothesis is that the dedicated vocabulary of Céu to deal with the external world (e.g., events and concurrency) will permit that students accomplish simple but complete IoT projects. We also want to attract new students to our research area over the long term, which will be important for the future of the project. We plan to offer a 3-week course open for all undergraduate students on each semester, starting next year.

References:

F. Sant'anna and others. The design and implementation of the synchronous language Céu. ACM Trans. Embed. Comput. Syst., 16(4):98:1–98:26, July 2017.

F. Sant'Anna. Safe System-level Concurrency on Resource-Constrained Nodes with Céu. PhD thesis, PUC–Rio, 2013.

Rodrigo C. M. Santos and others. A memory-bounded, deterministic and terminating semantics for the synchronous programming language Céu. In LCTES'18. ACM, New York, NY, USA, 1-18.

F. Sant'anna and others. Transparent standby for low-Power, resource-constrained embedded systems: a programming language-based approach (short WIP paper). In LCTES'2018. ACM, New York, NY, USA 94.

2. Have your objectives changed? If so, please explain.

The main objective remains the same: Energy efficiency for IoT software in the large through automatic standby at the programming language level.

We are also following the methodology of the original proposal closely:

IoT Hardware Infrastructure: Arduino-based embedded platforms (AVR/ATmega328p and ARM/Cortex-M0).

IoT Software Infrastructure: Rewriting device drivers and interrupt service routines in Céu.

IoT Applications: RF module and real-world IoT applications.

As mentioned in the previous section, we are opening a new front on education which is aligned with our objectives but was not originally planned.

3. Please list any preliminary results/outcomes.

We implemented a new version of Céu with support for interrupts and power management:

We added support for a variety of drivers and two microcontrollers for Céu-Arduino:

We have an early prototype of our educational programming environment:

Our main quantitative results so far are the first full-application experiments showing significant energy savings ranging from 20% to 99%. The table below shows the energy consumption (in mA) of five applications originally in C/Arduino which we rewrote in Céu:

The first case illustrates how an "Empty" program in Arduino remains 100% of the time awake while an equivalent in Céu sleeps in the most efficient standby mode. The "Blink" example in Céu relies on timer standby, which is not the most efficient, but still consumes less energy than its Arduino counterpart. The "Radio" example in Céu can turn off the radio periodically, showing significant savings while waiting (3.0 mA). The "Protocol" example illustrates a scenario for a busy node in a mesh network, which keeps the radio on 100% of the time.

The results are preliminary and should be taken with a grain of salt. Even tough the chosen applications are from third parties, they are not real-world scenarios. In particular, the versions in Arduino do not attempt to take advantage of standby modes. Nevertheless, the use of standby modes in Arduino is unusual and challenging since it is very application dependent and must be programmed explicitly.

4. Have you faced any difficulties or challenges in developing your project so far? If so, please explain.

Hiring research students has been the most difficult task so far. We originally planned to spend around 80% on research students, but we now estimate this number to be around 50% by the end of the project.

In particular, we wanted a professional embedded engineer early in the project for the heavy-lifting work on drivers and platforms that serve as the basis of the IoT applications. However, since the grant became available only on April,

we missed some opportunities and could not hire such professional. At the end, the outcome was actually favorable since I overestimated the importance of an experienced domain-specific engineer at an early stage. We could redirect part of the budget to the presentation on the international conference, which was not initially planned.

We still have difficulties in finding good students to work in the area of embedded systems. As a new professor in my University, I was not associated with any graduate program until mid March, when I joined the Electrical Engineering program. The next semester I will offer a course on Embedded Software, which did not exist in the graduate program. This course and our new front on the educational programming environment are important to bring new students. Nevertheless, we got our first long-term committed student last May: She is now joining the MSc program under my supervision and will probably stay in this project for the next two years.

5. Please list any findings you've generated so far.

IoT applications with the typical sense-broadcast-sleep loop pattern can be put in standby mode most of the time and save energy significantly (up to over 99%).

A specialized programming language, with appropriate vocabulary to deal with events from the real world, can be adopted in the whole development cycle, from device drivers up to the application.

This programming language can automate the process to detect when an application can sleep, and also which sleep mode to use.

6. Please indicate any information that you consider relevant for your final evaluation.

7. What are the next steps in your project and how much of your original plan do you expect to have achieved by the time you submit your final report (in February 2019)?

Our next steps are focused on three fronts:

Evaluate realistic IoT applications:

From the original plan: "In order to evaluate the gains in energy efficiency with the proposed infrastructure, we will need to evaluate the consumption of realistic applications." This is the last step and probably the most laborious, since it requires a mix of hardware, software, and physical deployment. As a canonical example, to monitor the temperature of a building, we need to deploy IoT devices in several floors and rooms of the building. We also need to build the device with sensors and RF communication, and develop the software. Finally, we need some third-party software to compare, and measure the energy consumption in the evaluation.

Introduce an optional IoT course for undergraduate students:

We already have a student working on the educational programming environment for a couple of months and we expect to start this course during the first semester of the next year.

Work towards formal verification of program specifications:

Céu has strong safety properties that apply to all programs as discussed in our last paper (i.e., termination, determinism, and memory boundedness). We want to extend this work with automatic proofs for even stronger properties, but which apply to specific programs. As an example, suppose we want to prove that a given program satisfies a specification related to energy consumption as follows: "After broadcasting the value of a sensor, the device must sleep for one hour". Our goal is to take a specification written in some higher-level language, take a program written in Céu, and automatically verify whether the program satisfies the specification or not.

Item 1 will concentrate most of our efforts until February, and would complete our intended research for the first year. The applications would validate our software infrastructure and would allow us to submit a paper to a journal with high impact in our subfield (embedded software). We believe that we can complete our original plan and submit this paper by the end of February.

Items 2 and 3 were not in our original plan for the first year, and actually target medium and long-term goals. Item 2 is related to the usability and adoption of Céu and is important to build a local community around the language. Item 3 is in very early stages and is related to what we can know (and automatically prove) about a program before it executes. We believe that static verification of specifications towards program correctness (including energy efficiency) will have a high impact also outside of our subfield.

8. Por favor, descreva como tem sido o uso dos recursos via Funarbe (em português).

A comunicação com os funcionários é suficientemente rápida, eles em geral têm boas respostas e conhecimento sobre as dúvidas e demandas, e sempre tentam resolver os problemas proativamente.

O meu projeto requisitou R$70000, mas descontando 6% da Funarbe e 5% da UERJ, sobram R$62300. Já foram gastos em torno de R$31000 (incluindo setembro) divididos da seguinte forma:

No momento, temos dois alunos de iniciação, um pós-doc, e uma aluna de mestrado. Segue a tabela prevista para o pagamento de pessoal:

Com essa previsão, ainda há um restante de R$10000 que será usado para um congresso internacional em novembro caso um trabalho já submetido seja aceito.