INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Laboratoire CITI / INRIA

Research internship

Title: **Delay-Tolerant Device Drivers for Transiently-Powered Systems**

Location: CITI Lab, INSA-Lyon/INRIA

Keywords: Embedded Systems, Bare-Metal Programming, Memory Management

Contact: `guillaume.salagnac@insa-lyon.fr` — `http://perso.citi-lab.fr/gsalagnac`

Context

The general context for this project the Internet of Things, i.e. a vision where we are gradually surrounded by intelligent, connected objects. Today the most obvious IoT device is probably the smartphone which we all carry in our pockets. Technologically speaking, a smartphone is a full-featured computer with abundant processing power and wireless connectivity. But all these features come at a significant cost in terms of energy: every few days the battery runs out and must be recharged.

In contrast, some IoT systems are designed to operate without maintenance in long-term deployments. In these scenarios energy management is a critical concern and impacts many aspects of the design process. For reasons of cost reduction, volume constraints, or even just to ensure and indefinite lifetime, some systems don't rely on battery power but harvest their energy in the environment. For instance, contactless smart cards are entirely powered by the host terminal. In the academic litterature these platforms are known as Transiently-Powered Systems.

Writing software for a TPS is a lot more challenging than traditional computer programming. First, TPS plaftorms typically have very limited computing resources: a slow CPU, a few kilobytes of unprotected memory, no hardware accelerators. Second, running on harvested energy means that the system will suffer frequent power failures, typically many times per second. In current industrial practice, the application developer has to take all these contraints into account when designing and writing their program. However there are several research efforts intended to make this task easier, thanks to both hardware-based and software-based approaches. Our group at CITI Lab is involved in designing and studying operating-system level mechanisms for such systems.

Goals

In the past, we have already designed and developped an operational TPS demonstrator called Sytare. First, the student will have to get familiar with the hardware architecture of the system (MCSP430FR5739, CC2500), the build toolchain (GCC, LD, make, python), as well as the embedded codebase (OS, application, device drivers).

Then, the main goal of this work is to design, implement, and study new mechanisms at the interface between application and drivers, in particular delay-tolerant driver calls. Sytare is able to detect when a power failure happens in the middle of a hardware access (e.g. sending a radio packet). When power comes back on, we ensure that the whole request is not just resumed but *retried* entirely. But it would be smarter to detect, in advance, when a hardware access is going to fail. In that case, the request at hand can be postponed and the remaining energy can be spent doing something more useful and thus improve the overall performance of the system.

Depending on the pace of progress of the student, several other research directions are to be explored in Sytare: multi-application support, smart sleep functions, selective syscall wrappers, etc.