Demo Abstract: Rebooting the Embedded System

Amit Levy[§], Bradford Campbell[†], Branden Ghena[†], Shane Leonard[§], Pat Pannuto[†], Philip Levis[§], and Prabal Dutta[†]

[§]Stanford University
{levya,shanel,pal}@stanford.edu

[†]University of Michigan
{brghena,bradjc,ppannuto,prabal}@umich.edu

ABSTRACT

For the last fifteen years, research explored the hardware, software, sensing, communication abstractions, languages, and protocols that could make networks of small, embedded devices—motes—sample and report data for long periods of time while unattended. Today, the application and technological landscapes have shifted, introducing new requirements and new capabilities. Hardware has evolved past 8 and 16 bit microcontrollers: there are now 32 bit processors with lower energy budgets and greater computing capability. New wireless link layers have emerged, creating protocols that support direct interaction with users, but introduce novel limitations that systems must consider. Programming language advances have led to the ability to write system kernels that guarantee safety and reliability while maintaining low overhead. The time has come to look beyond optimizing networks of motes. We look towards new technologies such as Bluetooth Low Energy, Cortex M processors, and capable multi-process operating systems, with new application spaces such as personal area networks, and new capabilities and requirements in security and privacy to inform contemporary hardware and software platforms. It is time for a new, open experimental platform in this post-mote era.

1. INTRODUCTION

In the early 2000s, a flurry of hardware platforms and supporting software empowered the research community to investigate and explore wireless sensor networks and their applications. Many projects today still use these "mote"-class devices to research systems software, low-power networking, and application design.

Technology has progressed a great deal in the intervening years. 32-bit microcontrollers (MCUs) with many more peripherals and capabilities have sleep currents competitive with mote-class microcontrollers. Low-power wireless connectivity is no longer confined to closed world of ZigBee. Bluetooth Low Energy allows human-centric devices like smartphones to interact with wireless sensor networks directly. Sensors are more diverse and orders of magnitude more energy efficient and precise.

Simultaneously, applications have become much richer. Applications in early sensor network research focused on fixed rate, long-term sensing, guiding a research agenda of ultra-low power operation and robust multi-hop networking. Today, personal area networks (PANs) tether to phones and interact with proximity networks such as iBeacon. Building area networks share knowledge, like occupancy, among security, HVAC, and lighting control. In addition, the rise of "maker culture" and their platforms and communities (e.g. Arduino [1]) has led to a level of diversity and accessibility that early research platforms simply could not provide. We have an explosion of new applications and developers, each with new and challenging requirements. We have reached a turning point in hardware, enabling new devices and operational models. It is time for a new platform: a new OS and a family of hardware devices to explore and research embedded networked systems in the post-mote era.

1.1 Hardware Evolution

Recent generations of 32-bit low-power ARM MCUs (the Cortex-M series) are significantly more powerful than the mote-class MCUs of yester-year. For example, the Atmel SAM4L [3] runs at 48 MHz and integrates encryption accelerators, 16 DMA channels, and numerous communication bus controllers such as UART, SPI, I^2C, and USB, with a sleep current as low as 2 µA with full RAM retention. These MCUs also include hardware protection mechanisms, such as ARM's Memory Protection Unit (MPU). This evolution presents an opportunity for systems design to take advantage of these enhanced features to optimize energy consumption, increase functionality, and provide better security guarantees.

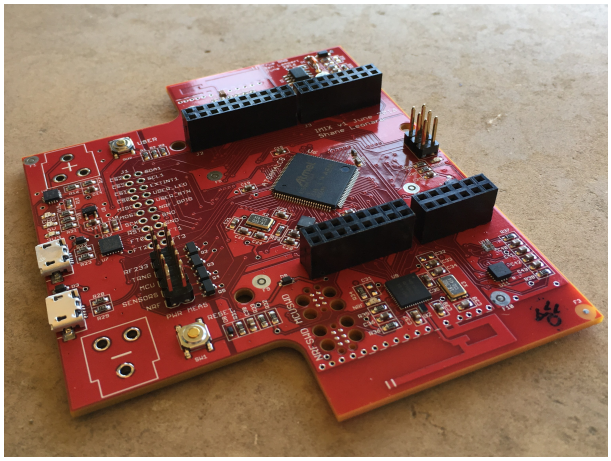
1.2 Application Evolution

Modern sensor network applications are transitioning from the lab to end-users. These applications span a variety of environments, from personal devices like health trackers, to home and building automation, and even campus-wide or city-scale sensing. They are increasingly mobile, relying on ubiquitous wireless protocols like Bluetooth Low Energy to communicate opportunistically with end-users. They store private or sensitive data, like health information or encryption keys. Finally, their applications can no longer be written and flashed just once for the lifetime of the device. Systems today expect in-the-field updates and support for third-party extensions and applications.

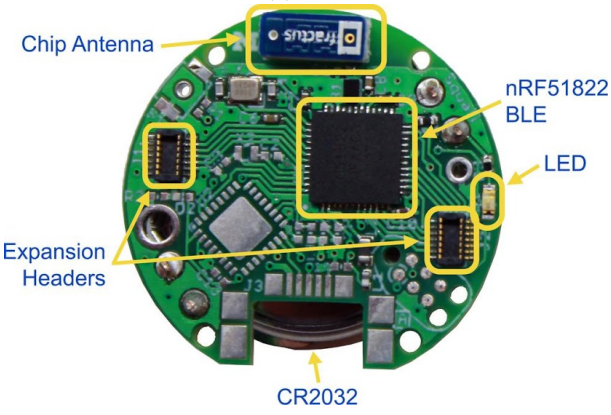
SenSys '16 November 14-16, 2016, Stanford, CA, USA

ACM ISBN 978-1-4503-4263-6/16/11.

DOI: http://dx.doi.org/10.1145/2994551.2996539



(a) Imix



(b) Squall

Figure 1: The Imix and Squall are development platforms for next generation sensor networks. We demonstrate an end-to-end application on these platforms using a new, multi-tenant embedded operating system called Tock.

2. DEMO

We demonstrate an end-to-end application built using a new multi-tasking embedded operating system on two development platforms. The platforms include a small, Bluetooth Low Energy sensor tag extensible with a variety of sensor shields, as well as a larger development platform with a powerful but low-power MCU, Bluetooth Low Energy, IEEE 802.15.4 radios, a number of on-board sensors, and an Arduino-compatible header for expansion.

2.1 Squall

Squall is a price and area optimized platform for massive dissemination and large-scale prototyping of Bluetooth Low-Energy sensing applications. It is a tag-sized board incorporating a Nordic nRF51822 [4]—an ARM Cortex-M0 with integrated Bluetooth Low Energy—, with optional USB and rechargable battery. The Squall includes headers allowing extension with a variety of shields, such as environmental sensing, motion detection, and passive audio loudness monitoring.

2.2 Imix

Imix is a mass-production version of the Firestorm [2], adding an external high-entropy true random number generator. It includes both an Atmel SAM4L and a Nordic nRF51 Bluetooth Low Energy MCU. In addition, the board incorporates an RF233 802.15.4 radio, an accelerometer, and light and temperature sensors. Imix includes Arduino compatible headers allowing it to be extended with any Arduino shield.

Imix enables researchers to explore novel embedded applications using both BLE and 802.15.4 wireless networks. Moreover, Imix allows power measurement of individual components, allowing high-fidelity evaluation of systems and applications.

2.3 Tock Embedded Operating System

Embedded operating systems have traditionally been limited to libraries that abstract hardware and implement common utilities. These systems provide only limited mechanisms, if any, to ensure the safety of drivers or isolate applications. Instead, developers must assume that all code is equally trustworthy and bug free. As embedded systems strive to provide additional features, developers draw on third-party source code for libraries, drivers and applications.

Tock is a safe, multitasking operating system for memory constrained devices. Tock is written in Rust, a type-safe systems language with no runtime or garbage collector. Tock uses the Rust type system to enforce safety of components, called capsules, in a single- threaded event-driven kernel. In addition, Tock uses remaining memory to support processes written in any language. To support safe event-driven code that responds to requests from processes, Tock introduces two new abstractions: memory containers and memory grants.

2.4 Application

We demonstrate a sensor network application in which Imix and Squall boards are deployed with Tock. The Squalls are distributed to participants and advertise their presense over Bluetooth Low Energy to Imix boards spread around the room. The Imix boards connect over an 802.15.4 network and report presence information to a central hub.

3. REFERENCES

[1] Arduino. http://www.arduino.cc/. Accessed 19-August-2016.

[2] M. P. Andersen, G. Fierro, and D. E. Culler. Enabling synergy in iot: Platform to service and beyond. In *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 1–12, April 2016.

[3] Atmel. SAM4L Cortex-M4 MCUs. http://www.atmel.com/microsite/sam4l/default.aspx. Accessed 19-August-2016.

[4] Nordic Semiconductor. NRF51822. https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF51822. Accessed 19-August-2016.