Demo Abstract: Hamilton - A Cost-Effective, Low Power Networked Sensor for Indoor Environment Monitoring

Michael P Andersen
Computer Science Division
University of California, Berkeley
m.andersen@cs.berkeley.edu

Hyung-Sin Kim
Computer Science Division
University of California, Berkeley
hs.kim@cs.berkeley.edu

David E. Culler
Computer Science Division
University of California, Berkeley
culler@cs.berkeley.edu

ABSTRACT

Operating buildings can be challenging, especially with poor instrumentation of the indoor environment. There are several wireless sensor platforms on the market but most are too difficult to deploy en-masse, requiring the end-user to program devices, or manage infrastructure. Many rely on smart-phones and do not work when unattended. The Hamilton wireless sensor node is a full-stack solution providing a low-cost and low-power high-resolution sensor that operates for more than five years on a battery, along with all the cloud infrastructure required to interact with the data. It is pre-programmed and ready to use, but the firmware can be easily modified by using the standard C language.

CCS CONCEPTS

• **Computer systems organization** → **Sensor networks**; • **Networks** → *Network experimentation*; Transport protocols;

KEYWORDS

Sensor Network, Low Power Network, Transport Layer, Link Layer, Transmission Control Protocol, IEEE 802.15.4

ACM Reference format:

Michael P Andersen, Hyung-Sin Kim, and David E. Culler. 2017. Demo Abstract: Hamilton - A Cost-Effective, Low Power Networked Sensor for Indoor Environment Monitoring. In *Proceedings of ACM conference, Delft, The Netherlands, November 6–8 2017 (BuildSys'17),* 2 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

Deployment of a large number of embedded networked sensors in a building has the potential to provide useful indoor data more easily and at a finer granularity. For a large deployment in practice, a networked sensor should be easy-to-use, low-cost, and low-power. Recently, low power but highly capable Cortex-M SoCs (System-on-Chip) have become available on the market, permitting a low-cost highly-integrated wireless sensor design. RIOT-OS [3], a recent low power embedded OS, provides a multi-threading and standard C language-based firmware development environment, making embedded programming easier than using conventional embedded OSs [4, 7]. Development of fast time-series database and plotting techniques enables easy storage and visualization of sensor data through the cloud [1].

This demo presents Hamilton, a networked sensor for indoor monitoring. It is an easy-to-use, low-cost, and low-power full stack solution, which incorporates the aforementioned latest technologies. It has a default firmware so that a user can obtain various sensor data once she turns on a Hamilton device.

BuildSys'17, November 6–8 2017, Delft, The Netherlands

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

2 RELATED WORK

Many networked sensors have been developed over the last two decades. Each networked sensor has its own design goal while reflecting the latest hardware technology at the time. TelosB [8] is one of the most popular networked sensors, which has been benchmarked in a number of studies. It integrates every component (an IEEE 802.15.4 radio, a 16-bit MCU, sensors, USB interface) on a single board with careful power gating, which makes it a low power, easy-to-use, and robust networked sensor. Epic [5] also uses a 16-bit MCU and an IEEE 802.15.4 radio but takes a different approach, called building block approach, to facilitate all of prototype, pilot, and production phases.

After Cortex family was introduced, networked sensors started trying to use 32-bit CPUs for faster processing and more memory space. Egs and Opal [6], designed for medical and environment monitoring applications, respectively, use Cortex-M3. This work shows that compared to TelosB, a 32-bit platform can save both time and power when doing a complex computation, but its current consumption in the idle mode is much higher. Wandstem [9] shows that a 32-bit (Cortex-M3) board can provide lower idle current than TelosB. In addition, Firestorm [2] reveals that a 32-bit (Cortex-M4) networked sensor can be applied to the maker space by providing abundant extension pins like Arduino, both IEEE 802.15.4 and BLE radios, and extremely low power.

3 HARDWARE

In view of real application, a key missing aspect in the previous work is a thorough cost analysis including all the manufacturing process. In addition, an extensible board design with many I/O pins is good for embedded developers to try various things, but not for real users who want to apply networked sensors directly without much labor. Lastly, the previous work focuses on a hardware device rather than a full stack solution, which leaves significant amount of additional work to actually see desirable data from the screen.

To fill the gap, our design approaches for Hamilton are (1) adopting the latest low power and highly integrated off-the-shelf components, such as SoC, integrated balun, and chip antenna, for low assembly cost, (2) including useful sensors for indoor monitoring,

Figure 1: The Hamilton-7C wireless sensor

Type	Sensor	Precision
Air Temperature	HDC1080	0.01 C
Air Humidity		0.1 %
Radiant Temperature	TMP006	-
Illuminance	ADPS 9007	1 Lux
3 Ax. Acceleration	FXOS8700	0.1 mG
3 Ax. Magnetometer		20 nT
Motion	EKMA PIR	-

Table 1: The sensors on the Hamilton-H7C

(3) integrating all components onto a single board with power gating. This results in a small low-cost networked sensor with a 32-bit SoC.

Specifically, the Hamilton platform is built around an Atmel SAMR21 SoC, a 32-bit Cortex M0+ processor with ample resources (32kB SRAM, 256kB Flash) and an IEEE 802.15.4 radio. The Hamilton-7C (pictured in Figure 1) has an extensive sensor pack (Table 1). These sensors allow the unit to report illuminance, operative temperature (by combining air temperature, radiant temperature and air humidity with a model for unobserved parameters) and even orientation. The sensors also know when they have been tampered with, by leveraging the accelerometer, so that analysis algorithms can know which data to discard. The Hamiltons use a CR123A Lithium battery, mounted on the reverse of the board, and for the models without PIR (the Hamilton-3C) are expected to last 5 years on a battery when reporting sensor data every 20 seconds.

The boards ship pre-programmed and ready to use with our cloud services. But for developers, the boards feature a Tag-Connect bed-of-nails programming port so that custom firmware can be loaded on the device. For easy programming low power firmware, we support RIOT-OS [3], a standard C-based preemptive multi-threading OS developed for Internet of Things (IoT).

4 CLOUD SERVICES

A key challenge in wireless sensor networks (WSNs), and IoT in general, is how to store and process the data in a cost-effective yet reliable manner. If the wireless sensors are cheap but the requisite servers to store the data are expensive, then the total cost of ownership remains prohibitive. It is tempting to use a single cheap server to store the data, but then you will invariably suffer from data loss and service outages due to a lack of redundancy. Similarly, many of the people who stand to benefit most from WSNs (such as building operators) do not have the System Administrator or Dev Ops resources to securely maintain high-availability services, even if the capital expenses were overcome. If a system requires constant attention to keep it operating, the effort will soon outpace the benefit and the sensors will remain unused.

With Hamilton we hope to resolve these problems using economies of scale. By providing a high-availability highly-reliable cloud back-end to multiple users of the platform, the costs can be spread across all users, reducing the costs for everyone. Similarly, the devices ship already pre-programmed and pre-commissioned with encryption keys, so that they can be deployed straight out of the box without any embedded development expertise, yet remain highly secure. Unlike most commercial products, however, all the hardware and firmware is completely open-source, so if researchers want to tweak and improve the design, they can.

The gateways transport the data to the cloud back-end which provides data storage, online visualization and a rich API for integrating with analysis software such as Tableau, MATLAB or Python. It can also integrate with control software, so that building-level operating parameters can be dynamically adjusted to track fine-grained occupant-level metrics.

5 DEMO

We will show a live sensor network, showing the sensors themselves as well as the network infrastructure. Participants are welcome to interact with the sensors. There will be a display showing the cloud back-end and participants will be able to visualize the data from the sensors, watching how their interactions reflect in real-time. We will also demonstrate the API, showing integration with analysis software and helping participants evaluate the Hamilton open platform for their use cases.

The Hamilton project is the culmination of years of academic research in WSNs, and it is now time to bring the technology into widespread use.

REFERENCES

[1] Michael P Andersen and David E Culler. 2016. BTrDB: optimizing storage system design for timeseries processing. In *Proceedings of the 14th USENIX Conference on File and Storage Technologies (FAST 16)*.

[2] Michael P Andersen, Gabe Fierro, and David E Culler. 2016. System design for a synergistic, low power mote/BLE embedded platform. In *Information Processing in Sensor Networks (IPSN), 2016 15th ACM/IEEE International Conference on*. IEEE, 1–12.

[3] Emmanuel Baccelli, Oliver Hahm, Mesut Gunes, Matthias Wahlisch, and Thomas C Schmidt. 2013. RIOT OS: Towards an OS for the Internet of Things. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*. IEEE, 79–80.

[4] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. 2004. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. IEEE, 455–462.

[5] Prabal Dutta, Jay Taneja, Jaein Jeong, Xiaofan Jiang, and David Culler. 2008. A building block approach to sensornet systems. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 267–280.

[6] JeongGil Ko, Kevin Klues, Christian Richter, Wanja Hofer, Branislav Kusy, Michael Bruenig, Thomas Schmid, Qiang Wang, Prabal Dutta, and Andreas Terzis. 2012. Low Power or High Performance? A Tradeoff Whose Time Has Come (and Nearly Gone).. In *EWSN*. Springer, 98–114.

[7] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, et al. 2005. TinyOS: An operating system for sensor networks. *Ambient intelligence* 35 (2005), 115–148.

[8] Joseph Polastre, Robert Szewczyk, and David Culler. 2005. Telos: enabling ultra-low power wireless research. In *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press, 48.

[9] Federico Terraneo, Alberto Leva, and William Fornaciari. 2016. A High-Performance, Energy-Efficient Node for a Wide Range of WSN Applications.. In *EWSN*. 241–242.