Poster abstract: Building Adaptable Sensor Networks with Sensor Cubes

H. Diall, [1] K. Raja,[1] I. Daskalopoulos, [1] S. Hailes, [1] T. Torfs,[2] C. Van Hoof[2] and G. Roussos[3]
[1]University College London, UK
[2]IMEC, Integrated Systems Department, Belgium
[3]Birkbeck College, University of London, UK
{d.diall,k.raja,i.daskalopoulos,s.hailes}@cs.ucl.ac.uk, {torfst,vanhoofc}@imec.be, g.roussos@bbk.ac.uk

ABSTRACT
In this poster and demonstration we present the Sensor Cube platform, an ultra-compact, modular and power-efficient way to build sensor networks. The design, implementation and the complete technical details of this platform have been discussed elsewhere. Here, we aim to showcase the platform in action and argue in support of its features. Sensor Cubes are developed using two ingredients: a stackable hardware design recently developed by the human++ research initiative at IMEC, and a Tiny OS based operating environment. The hardware measures less than a cubic centimetre when configured with standard environmental monitoring capability and integrated coplanar antenna. The operating and software development environment is derived from Tiny OS, which has been modified to meet the hardware requirements of the Sensor Cube. In particular, we have introduced a power-aware, reliable, ALOHA-type MAC protocol that closely matches the operational characteristics of the low-power radio chip. In this poster and demonstration we will show the Senor Cubes executing standard data harvesting tasks.

1. INTRODUCTION
Over the past two years a variety of wireless sensor node platforms have emerged each offering particular benefits in building sensor networks. In this poster we introduce the Sensor Cube platform, which we believe distinguishes itself by providing novel opportunities for the construction of modular, ultra-compact and energy-efficient sensor networks. Indeed, Sensor Cubes offer increased flexibility in systems design and implementation compared against previous sensor network platforms. In this poster and demonstration, we present Sensor Cubes in action and aim to showcase their unique features.

The technical details of Sensor Cubes have been detailed in a series of papers [1, 4, 7] and here we concentrate on their effective and efficient operation in a practical situation. It is noteworthy that the platform hardware has been developed within the ongoing human++ research initiative at IMEC, Belgium, which is primarily focused on the development of body sensor networks. The addition of a fully functional runtime based on Tiny OS [2] and associated development tools offer the opportunity for the general-purpose use of this platform in additional application areas. The demonstration is based on the use of the popular Tiny OS Surge application.

2. HARDWARE PLATFORM
Sensor Cubes are built on the hardware platform recently developed at IMEC [1, 7] which offers two distinct advantages over the current state-of-the-art: first, it provides for an ultra-compact design including an integrated coplanar antenna that allows for very low power consumption; and second, it supports pluggable modules that allow for the physical reconfiguration of nodes to include only the functionality required for a particular application.

Figure 1. The Sensor Cube hardware platform developed by IMEC: Layers from top: Radio, Microcontroller, Power and Sensor Module. A 2€ coin provides a size reference.

The currently available hardware modules of the Sensor Cube platform include the micro-controller, radio communication, power and sensor. The prototype implementation, which will be used in the demonstration, features these functional blocks implemented as printed circuit boards of size 14x14mm2, plugged together to form a four-layer stack (Figure 1). This connector-based implementation is 18mm in height but an alternative design is also available, whereby solder-ball interconnections are used instead thus reducing the stack height to only 10mm (Figure 2).

At the top of the stack sits the radio layer consisting of the Nordic nRF2401 2.4GHz wireless transceiver chip [3] and an integrated antenna. The second (micro-controller) layer incorporates the Texas Instruments MSP430 micro-controller [6], with digital input/output, 12-bit analogue-to-digital converter, universal synchronous-asynchronous receiver/transmitter, and clock system and timers.

Below the micro-controller are the power management and sensing layers: the power management layer is designed so that in addition to batteries it can receive power from an energy-harvesting device (including, but not limited to, solar cells and vibration scavengers). The available sensing equipment used for this poster and demonstration is a Sensirion SHT15 board incorporating a temperature/humidity sensor and a Light-Dependent Resistor [5].

Of particular relevance to this work are the power-efficient operational characteristics of the radio module, including its transmit and receive power consumption, its built-in power-saving modes and its relatively high bit-rates for data transfer

(250kbps transmit and 1Mbps receive). In particular, the radio module supports the following modes of operation: transmit mode (13mA average at 0dBm output power); receive mode (23mA average for both channels on); configuration mode (12uA average); stand-by mode (12uA average); and power down mode (400nA average).

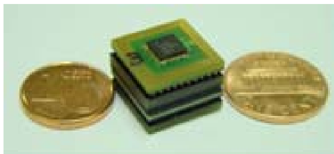

Figure 2. The Sensor Cube alternative hardware implementation using solder ball interconnect technology.

The overall modular design of Sensor Cubes allows for considerably increased adaptability when supporting a variety of application scenarios:

- When a large geographic area must be covered with a low-density sensor network it may be necessary to replace the radio layer to optimise overall performance.
- In cases where high data rates and complex signal processing functions are required, a more powerful digital signal processor could be employed in an alternative microcontroller layer.
- In cases where particular specialized sensors and associated sensor electronics are required (for example chemical, electrical or biosensors) they could be accommodated within this design on a separate module.
- In cases where power beyond that provided by the battery is necessary, or when a full power management system with scavenging is needed, such components could and have been developed and added as separate modules.

3. SOFTWARE PLATFORM

While assembling a fully functioning software platform has been a critical milestone for the system, actual operational experience indicated early on that it was necessary to re-engineer the standard Tiny OS network protocol stack to achieve appropriate levels of performance. The process of porting Tiny OS to the Sensor Cubes has been detailed elsewhere [4] and here we only briefly present the operation of the energy-efficient MAC protocol, which was introduced to match the operational characteristics of the Nordic chip.

The use of Nordic radio chip in particular imposes a number of limitations, notably due to lack of a high speed clock source – necessary to substantially reduce power consumption – which prevents the use of its so-called Direct mode and thus restricts the ability to control its radio efficiently. Furthermore, the maximum frame size in the alternative Shockburst mode is limited to 32 bytes, which must accommodate both the control header and the payload. Using a combination of the two modes available is not an option, as the radio configuration word cannot be modified during operation. Nevertheless, in Shockburst mode the radio component can concurrently operate two separate channels, a capability we exploited to decouple data transmission from the transmission and receipt of acknowledgements.

The implemented MAC design employs the payload section of the ShockBurst frame to encapsulate Tiny OS Active Message packets constructed by the OS. This leaves only 20 bytes per packet for application payload but this should be enough for the majority of applications. In case where packets require additional space it is their responsibility to manage fragmentation.

The use of the Shockburst mode as the principal mode of communications implies that two addresses are encapsulated in every data packet: the Shockburst broadcast address and a node-specific address within the AM header. This approach allows us to combine the performance advantages of Shockburst while at the same time retaining the capability to address data to specific nodes, thus maintaining unicast semantics.

A second feature of the MAC protocol is the fact that the receiver employs a set duty cycle that switches the radio between stand-by and receive mode at regular intervals so as to reduce energy consumption. The percentage of time the radio stays in each mode can and should be controlled by the application and defined at development time. A detailed discussion of the implications of various choices of duty cycle in the context of a specific empirical study is detailed in [4].

Finally, the last design decision has been against employing a carrier sense-based approach but rather to use a simple Aloha-based protocol instead. This decision was made on the basis that collisions are rare in the case of the application scenarios considered, either as a result of low node density or low transmission rates. Given that our principal concern has been to optimise for energy-efficiency, the higher energy cost required by the carrier sense approach is prohibitively high given the frequency of situations for which it would actually be useful. Nevertheless, it is still necessary to address the problem of collisions whenever they arise and also to deal with the cases in which the receiver is in stand-by when required to receive and hence unable to do so. To address both of these issues, link level acknowledgements and retransmissions are used whereby the transmitter waits for an acknowledgement for a fixed (but configurable by the application) period. If no acknowledgement is received then the packet is retransmitted. Note that since the transmitter cannot distinguish packets that are lost due to collision, and those lost because the receiver was in stand-by mode, it resends its packets until it receives an acknowledgement or reaches the Maximum Retransmission count (which is also configurable by applications).

4. REFERENCES

[1] K. Baert, B. Gyselinckx, T. Torfs, V. Leonov, F. Yazicioglu, S. Brebels, S. Donnay, J. Vanfleteren, C. Van Hoof, “Technologies for highly miniaturized autonomous sensor networks”, *1st Int. Wor. Adv. Sensors and Interfaces* (IWASI), Bari, Italy, April, 2005.

[2] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, A. Woo, E. Brewer and D. Culler, “The Emergence of Networking Abstractions and Techniques in TinyOS,” *Proc.* NSDI 2004, San Fransisco, California, USA, 29-31 March, 2004.

[3] Nordic Semiconductor, “nRF2401A Single Chip 2.4 GHz Radio Transceiver”.

[4] K. Raja, I. Daskalopoulos, H. Diall, S. Hailes, T. Torfs, C. Van Hoof and G. Roussos, “Sensor Cubes: A Modular, Ultra-Compact, Power-Aware Platform for Sensor Networks”, *IPSN SPOTS 2006*, submitted for publication.

[5] Sensirion, “SHT1x/SHT7x Humidity & Temperature Sensor”.

[6] Texas Instruments, “ MSP430x1xx Family User’s Guide”

[7] T. Torfs, C. Van Hoof, S. Sanders, C. Winters, and S. Brebels, “Wireless network of autonomous environmental sensors”, *IEEE Sensors Conference*, Vienna, Austria, April, 2004.